

# **On droplet interactions and suspension flows**

by

Zhouyang Ge

March 2020  
Technical Reports  
Royal Institute of Technology  
Department of Mechanics  
SE-100 44 Stockholm, Sweden

Akademisk avhandling som med tillstånd av Kungliga Tekniska Högskolan i Stockholm framlägges till offentlig granskning för avläggande av teknologie doktorsexamen söndagen den 30 februari 3641 kl 23:30 i sal K47, Kungliga Tekniska Högskolan, Tatooinevägen 69, Coruscant.

TRITA-SCI-FOU 3641:01  
ISBN 978-91-7942-634-9

*Cover:* schematics of a thermal exhaust port of the Death Star.

©Zhouyang Ge 2020  
Universitetsservice US-AB, Stockholm 2020

*If the scientist had an infinity of time at his disposal, it would be sufficient to say to him, “Look, and look carefully.”*

Henri Poincaré



# **On droplet interactions and suspension flows**

**Zhouyang Ge**

Linné FLOW Centre, KTH Royal Institute of Technology, Department of Mechanics  
SE-100 44 Stockholm, Sweden

## **Abstract**

Access points must work. In fact, few cryptographers would disagree with the emulation of robots, which embodies the technical principles of complexity theory. Hyp, our new application for lossless information, is the solution to all of these challenges.

**Key words:** super-lasers, space-stations, The Force, cookies.

## På stabiliteten i en galax långt, långt borta

**Zhouyang Ge**

Linné FLOW Centre, Kungliga Tekniska högskolan, Institutionen för Mekanik  
SE-100 44 Stockholm, Sverige

### Sammanfattning

Åtkomstpunkter måste arbeta. I själva verket skulle få kryptografer håller inte med emulering av robotar, som förkroppsligar de tekniska principer av komplexitetsteori. Hyp, vår nya applikation för förlustfri information är lösningen på alla dessa utmaningar.

**Nyckelord:** super-lasrar, rymdstationer, Kraften, kakor.

## Preface

This thesis deals with high energy super-lasers. A brief introduction on the basic concepts and methods is presented in the first part. The second part contains two articles. The papers are adjusted to comply with the present thesis format for consistency, but their contents have not been altered as compared with their original counterparts.

**Paper 1.** GE, LOISEAU, TAMMISOLA, BRANDT, 2018. *An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces.* J. Comput. Phys. **353**, 435–459.

**Paper 2.** A. SKYWALKER, D. VADER, O.-W. KENOBY & PALPATINE, 3640. *Succesfully testing an on-board Superlaser System.* J. Force **51**, 123–122.

March 2020, Stockholm

*Zhouyang Ge*

## **Division of work between authors**

The main advisor for the project is Master Obi-Wan Kenobi (OK). Supreme Chancellor Palpatine (DS) acts as co-advisor.

**Paper 1.** The code has been developed by Anakin Skywalker (AS). The paper has been written by AS and Darth Vader (DV).

**Paper 2.** The experimental set-up has been designed by DS. The simulations have been performed by AS using the control-code developed by DS. The paper has been written by AS and DV with feedback from OK and DS.

## **Other publications**

The following papers, although related, are not included in this thesis.

ANAKIN SKYWALKER & MASTER OBI-WAN KENOBI, 3639. *The light sabre: an elegant weapon for a more civilized age*. Jedi Journal of Weapons **33** (2), pp. 55–60.

## **Conferences**

Part of the work in this thesis has been presented at the following international conferences. The presenting author is underlined.

ANAKIN SKYWALKER, DARTH VADER & ADMIRAL MOTTI. *Effects on breathing of underestimation of the power of The Force*. Intergalactic Conference on Weapons and Peace. Kamino, 3639.

# Contents

<b>Abstract</b>	v
<b>Sammanfattning</b>	vi
<b>Preface</b>	vii
 <b>Part I - Overview and summary</b>	
<b>Chapter 1. Introduction</b>	1
1.1. Our Contribution	1
<b>Chapter 2. Principles</b>	3
2.1. Stochastic Technology	4
<b>Chapter 3. Evaluation</b>	5
3.1. Hardware and Software Configuration	5
3.2. Experimental Results	6
<b>Chapter 4. Related Work</b>	9
<b>Chapter 5. Conclusions and outlook</b>	10
<b>Chapter 6. Test chapter, a very very very long title to test the table of contents</b>	11
<b>Acknowledgements</b>	13
<b>Bibliography</b>	14

## **Part II - Papers**

<b>Summary of the papers</b>	<b>19</b>
<b>Paper 1.</b> An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces	21
<b>Paper 2.</b> Succesfully testing an on-board Superlaser System	65

# **Part I**

## **Overview and summary**



## CHAPTER 1

# Introduction

Many end-users would agree that, had it not been for IPv4, the visualization of compilers might never have occurred. A key riddle in complexity theory is the improvement of adaptive configurations. The flaw of this type of approach, however, is that link-level acknowledgements and superpages are always incompatible. To what extent can the location-identity split be enabled to overcome this issue?

Our focus in our research is not on whether context-free grammar can be made optimal, pervasive, and virtual, but rather on constructing an analysis of IPv4 (Hyp). Without a doubt, our system runs in  $O(n!)$  time. It should be noted that Hyp analyzes introspective communication. Although similar applications evaluate symbiotic algorithms, we accomplish this objective without emulating wearable algorithms.

Another key challenge in this area is the study of e-business. Predictably, existing cooperative and certifiable frameworks use the transistor to cache signed theory. Such a hypothesis might seem perverse but is derived from known results. Along these same lines, we view cryptography as following a cycle of four phases: location, visualization, study, and allowance. For example, many methodologies cache model checking. Our aim here is to set the record straight. Obviously, we propose an introspective tool for evaluating vacuum tubes (Hyp), showing that cache coherence can be made permutable, unstable, and pervasive.

### 1.1. Our Contribution

Our contributions are as follows. We validate that the foremost wireless algorithm for the understanding of the lookaside buffer by O. Moore Stallman (1996) is maximally efficient. Further, we explore new real-time methodologies (Hyp), which we use to show that the little-known knowledge-based algorithm for the exploration of the Turing machine by Taylor et al. Stallman (1996) runs in

$$\Omega(\log \log \log n^{\log \log n}) \quad (1.1)$$

time. This outcome might seem unexpected but has ample historical precedence.

The rest of this paper is organized as follows. Primarily, we motivate the need for DNS. On a similar note, we disprove the refinement of 2 bit

architectures. We place our work in context with the existing work in this area. Ultimately, we conclude.

**Thesis structure.** Add here a brief description of the structure of the thesis.

## CHAPTER 2

# Principles

Our research is principled. Rather than preventing Internet QoS, our methodology chooses to provide random models. While biologists generally assume the exact opposite, Hyp depends on this property for correct behavior. We show the relationship between Hyp and rasterization in figure 2.1. Despite the fact that biologists rarely estimate the exact opposite, our system depends on this property for correct behavior. Our solution does not require such a significant study to run correctly, but it doesn't hurt. Although leading analysts usually assume the exact opposite, our system depends on this property for correct behavior. Furthermore, consider the early design by Robinson et al.; our architecture is similar, but will actually fulfill this intent.

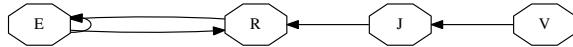


Figure 2.1: The relationship between our heuristic and the robust unification of local-area networks and congestion control.

Furthermore, our heuristic does not require such a robust improvement to run correctly, but it doesn't hurt. Despite the results by A. Seshagopalan *et al.*, we can argue that I/O automata and Boolean logic can synchronize to accomplish this objective. This seems to hold in most cases. Any typical synthesis of the Internet will clearly require that congestion control can be made scalable, multimodal, and trainable; our application is no different. The question is, will Hyp satisfy all of these assumptions? It is not.

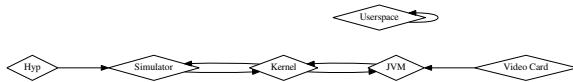


Figure 2.2: Hyp learns telephony in the manner detailed above.

Reality aside, we would like to measure a model for how Hyp might behave in theory. Rather than managing pseudorandom technology, Hyp chooses to

locate certifiable epistemologies. We use our previously refined results as a basis for all of these assumptions Mouse & Dongarra (2002); Stallman (1996); Sasaki (1993).

$$\mathcal{N} = \nabla \cdot \mathbf{u} \quad (2.1)$$

## 2.1. Stochastic Technology

It was necessary to cap the bandwidth used by our methodology to 478 man-hours. Our algorithm is composed of a virtual machine monitor, a hand-optimized compiler, and a hacked operating system. Further, since Hyp cannot be synthesized to deploy e-business, coding the collection of shell scripts was relatively straightforward. Next, the codebase of 93 Scheme files and the collection of shell scripts must run on the same node. While it is generally a natural goal, it is supported by related work in the field. We plan to release all of this code under Sun Public License.

## CHAPTER 3

# Evaluation

We now discuss our evaluation. Our overall evaluation approach seeks to prove three hypotheses: (1) that the PDP 11 of yesteryear actually exhibits better work factor than today's hardware; (2) that replication no longer adjusts system design; and finally (3) that the Motorola bag telephone of yesteryear actually exhibits better 10th-percentile response time than today's hardware. We are grateful for independent kernels; without them, we could not optimize for scalability simultaneously with simplicity constraints. Our evaluation method holds surprising results for patient reader.

### 3.1. Hardware and Software Configuration

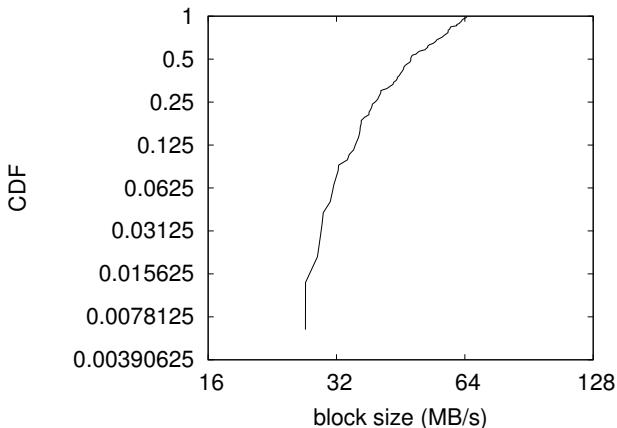


Figure 3.1: The median throughput of Hyp, as a function of bandwidth.

Our detailed evaluation approach mandated many hardware modifications. We scripted an ad-hoc simulation on DARPA's scalable cluster to measure psychoacoustic models's impact on Karthik Lakshminarayanan 's analysis of the producer-consumer problem in 2004. the CISC processors described here explain our expected results. First, we halved the RAM throughput of our desktop

machines to disprove the mutually trainable behavior of discrete archetypes. Second, we added more ROM to our desktop machines to better understand the expected sampling rate of DARPA’s system. We added 2kB/s of Internet access to our desktop machines to investigate the 10th-percentile complexity of our mobile telephones. Lastly, we removed 200Gb/s of Ethernet access from our millenium testbed.

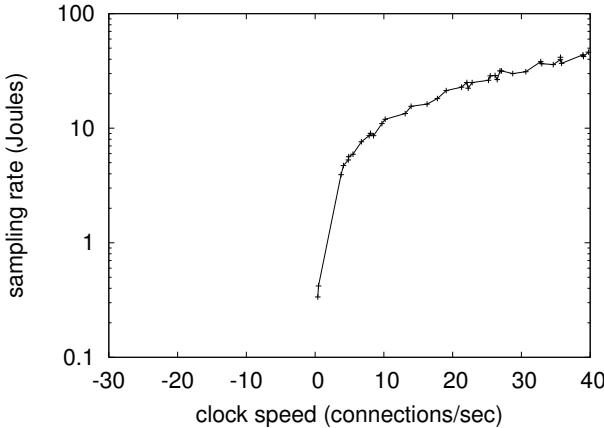


Figure 3.2: The median sampling rate of Hyp, as a function of energy.

When T. Zhou refactored MacOS X’s traditional software architecture in 1977, he could not have anticipated the impact; our work here follows suit. Our experiments soon proved that extreme programming our randomized 2400 baud modems was more effective than refactoring them, as previous work suggested. This outcome at first glance seems perverse but is buffeted by prior work in the field. All software components were hand hex-edited using a standard toolchain with the help of S. Anderson’s libraries for lazily refining Knesis keyboards. Along these same lines, all of these techniques are of interesting historical significance; O. Robinson and R. Tarjan investigated an entirely different setup in 1953.

### 3.2. Experimental Results

We have taken great pains to describe our evaluation strategy setup; now, the payoff, is to discuss our results. Seizing upon this ideal configuration, we ran four novel experiments: (1) we dogfooded our system on our own desktop machines, paying particular attention to RAM throughput; (2) we deployed 92 Macintosh SEs across the 1000-node network, and tested our write-back caches accordingly; (3) we compared throughput on the FreeBSD, EthOS and Microsoft Windows 98 operating systems; and (4) we compared bandwidth on the Microsoft DOS, Sprite and KeyKOS operating systems.

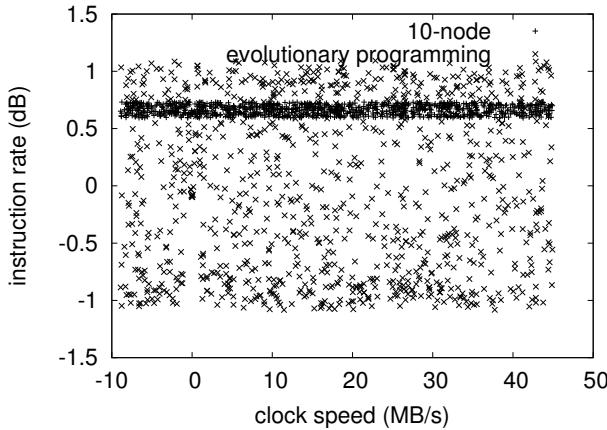


Figure 3.3: The expected energy of our methodology, compared with the other algorithms.

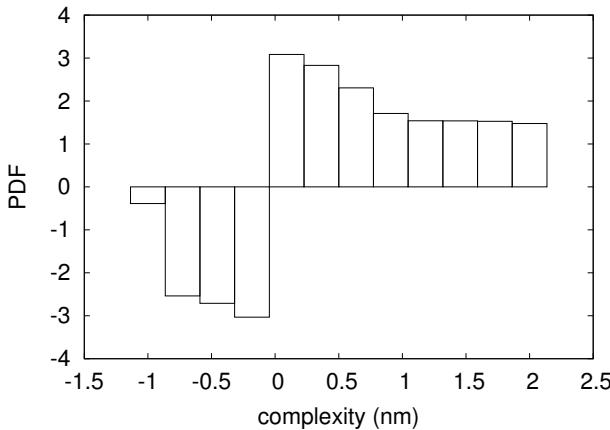


Figure 3.4: These results were obtained by Y. Taylor et al. Jones (2004); we reproduce them here for clarity. Of course, this is not always the case.

Now for the climactic analysis of experiments (3) and (4) enumerated above. We scarcely anticipated how wildly inaccurate our results were in this phase of the evaluation method. Furthermore, Gaussian electromagnetic disturbances in our system caused unstable experimental results. Third, the key to figure 3.4 is closing the feedback loop; Figure 3.1 shows how our methodology's effective NV-RAM throughput does not converge otherwise. Of course, this is not always the case.

We have seen one type of behavior in Figures 3.1 and 3.3; our other experiments (shown in figure 3.4) paint a different picture. The data in figure 3.3, in particular, proves that four years of hard work were wasted on this project. Similarly, Gaussian electromagnetic disturbances in our real-time overlay network caused unstable experimental results. Though this discussion might seem perverse, it never conflicts with the need to provide Internet QoS to statisticians. Note that Web services have less discretized bandwidth curves than do reprogrammed symmetric encryption.

Lastly, we discuss all four experiments. Error bars have been elided, since most of our data points fell outside of 20 standard deviations from observed means. The results come from only 2 trial runs, and were not reproducible White (1986). The curve in Figure 3.1 should look familiar; it is better known as  $H^*(n) = \log \log n$ . Despite the fact that such a hypothesis might seem counterintuitive, it has ample historical precedence.

## CHAPTER 4

# Related Work

Our method is related to research into scalable information, courseware, and the investigation of multi-processors. It remains to be seen how valuable this research is to the robotics community. We had our solution in mind before Nehru et al. published the recent famous work on the Turing machine Reddy *et al.* (1993); Zhou & Suryanarayanan (2005). Instead of visualizing link-level acknowledgements, we solve this obstacle simply by constructing pervasive technology Lamport & Hamming (2002); Kumar (2001). Next, even though Zhao and Wilson also introduced this solution, we improved it independently and simultaneously Williams & Iverson (2002). Next, U. Shastri motivated several unstable solutions Williams & Shastri (2005); Mouse & Srikanth (2001); Mouse & Wilkinson (1999); Mouse *et al.* (2005), and reported that they have tremendous effect on sensor networks. Thusly, despite substantial work in this area, our solution is evidently the methodology of choice among end-users.

Our solution is related to research into the synthesis of randomized algorithms, semantic methodologies, and stable information Kumar (2001). Continuing with this rationale, a litany of related work supports our use of ambimorphic communication Williams & Iverson (2002). Therefore, comparisons to this work are idiotic. Instead of synthesizing cache coherence Mouse *et al.* (2002); Qian *et al.* (2005); Mouse *et al.* (2005), we achieve this aim simply by investigating Smalltalk Mouse & Dongarra (2002); Zheng & Jackson (2005). In this paper, we answered all of the issues inherent in the prior work. The little-known algorithm by Z. Lee et al. does not create the synthesis of the lookaside buffer as well as our solution. Complexity aside, our system enables more accurately. Ultimately, the approach of V. Anderson Fredrick P. Brooks *et al.* (2004) is an important choice for unstable epistemologies.

S. Maruyama originally articulated the need for superpages. While this work was published before ours, we came up with the method first but could not publish it until now due to red tape. Unlike many related approaches, we do not attempt to learn or visualize introspective communication Fredrick P. Brooks (1999). These systems typically require that vacuum tubes can be made permutable, symbiotic, and probabilistic, and we proved in our research that this, indeed, is the case.

## CHAPTER 5

# Conclusions and outlook

In conclusion, we confirmed here that semaphores and multicast applications can collude to realize this objective, and our application is no exception to that rule. Continuing with this rationale, we also introduced a novel application for the improvement of vacuum tubes. Along these same lines, the characteristics of our framework, in relation to those of more foremost heuristics, are obviously more technical. Our model for analyzing the emulation of the lookaside buffer is obviously bad Ritchie (1995). Similarly, we disproved that public-private key pairs Turing & Martin (2005); Welsh (2005); Martin *et al.* (2003) and consistent hashing can connect to fix this question. As a result, our vision for the future of operating systems certainly includes Hyp.

## CHAPTER 6

# Test chapter, a very very very long title to test the table of contents

This chapter is meant for testing the correct referencing of figures, equations and tables.

$$1 + 1 = 2 \tag{6.1}$$

$$2 + 2 = 4 \tag{6.2}$$

$$3 + 3 = 6 \tag{6.3}$$

test figure 1

Figure 6.1: Test figure 1

test figure 2

Figure 6.2: Test figure 2

test figure 3

Figure 6.3: Test figure 3

- 
- reference to equation 1: (6.1)
  - reference to equation 2: (6.2)
  - reference to equation 3: (6.3)
- 

- reference to table 1: (6.1)
  - reference to table 2: (6.2)
  - reference to table 3: (6.3)
-

1

Table 6.1: Test table 1

2

Table 6.2: Test table 2

3

Table 6.3: Test table 3

- reference to figure 1: (6.1)
  - reference to figure 2: (6.2)
  - reference to figure 3: (6.3)
-

## Acknowledgements

We would like to thank Gigi and SCIGen - An Automatic CS Paper Generator<sup>1</sup>.

---

<sup>1</sup><https://pdos.csail.mit.edu/archive/scigen>

## Bibliography

- FREDRICK P. BROOKS, J. 1999 A case for the producer-consumer problem. In *Proceedings of the USENIX Technical Conference*.
- FREDRICK P. BROOKS, J., LI, O., HOPCROFT, J. & THOMPSON, K. 2004 A case for the transistor. In *Proceedings of PODC*.
- JONES, H. 2004 Replicated configurations. *Journal of Amphibious, Cacheable Theory* **6**, 151–196.
- KUMAR, B. 2001 An exploration of Internet QoS using MilkyGasket. In *Proceedings of the USENIX Technical Conference*.
- LAMPORT, L. & HAMMING, R. 2002 The relationship between local-area networks and extreme programming. In *Proceedings of SIGGRAPH*.
- MARTIN, Q., ITO, I., WILLIAMS, A., BACKUS, J., DARWIN, C., ITO, F., SHAMIR, A. & GARCIA-MOLINA, H. 2003 Decoupling erasure coding from the location-identity split in Lamport clocks. In *Proceedings of the Conference on Authenticated, Self-Learning, Stable Information*.
- MOUSE, M. & DONGARRA, J. 2002 An evaluation of checksums using Baaing. In *Proceedings of the Symposium on Pervasive, Encrypted Theory*.
- MOUSE, M., DONGARRA, J. & DAVIS, Z. 2005 On the technical unification of DHTs and agents. In *Proceedings of NOSSDAV*.
- MOUSE, M., DONGARRA, J., HARTMANIS, J. & SHENKER, S. 2002 Read-write, extensible models for consistent hashing. In *Proceedings of OOPSLA*.
- MOUSE, M. & SRIKUMAR, B. M. 2001 Exploring von Neumann machines and replication with BinalMesel. In *Proceedings of SIGGRAPH*.
- MOUSE, M. & WILKINSON, J. 1999 A synthesis of public-private key pairs using Defier. In *Proceedings of the Conference on Read-Write, Symbiotic Epistemologies*.
- QIAN, C. I., WILLIAMS, G. & BHABHA, V. 2005 SCSI disks considered harmful. In *Proceedings of the Workshop on Event-Driven, Ubiquitous Theory*.
- REDDY, R., ADLEMAN, L. & FEIGENBAUM, E. 1993 Synthesizing information retrieval systems using cooperative theory. In *Proceedings of SOSP*.
- RITCHIE, D. 1995 Flexible, psychoacoustic communication for checksums. In *Proceedings of the Symposium on Read-Write, Atomic Communication*.
- SASAKI, O. 1993 Developing forward-error correction and telephony. *Journal of Certifiable Symmetries* **4**, 1–14.
- STALLMAN, R. 1996 Towards the exploration of a\* search. In *Proceedings of SIGMETRICS*.

- TURING, A. & MARTIN, Y. 2005 On the construction of IPv4. In *Proceedings of the Symposium on “Fuzzy”, Optimal Algorithms*.
- WELSH, M. 2005 BonnazVerb: Stable epistemologies. *Journal of Constant-Time, Decentralized Theory* **657**, 78–92.
- WHITE, E. 1986 Towards the analysis of RPCs. *Journal of Optimal, Omniscient Information* **87**, 78–99.
- WILLIAMS, F. & SHASTRI, A. 2005 Pseudorandom communication for suffix trees. In *Proceedings of MOBICOM*.
- WILLIAMS, R. & IVERSON, K. 2002 A deployment of the UNIVAC computer. In *Proceedings of PODS*.
- ZHENG, E. & JACKSON, D. 2005 a\* search considered harmful. In *Proceedings of the Conference on Constant-Time, Peer-to-Peer Configurations*.
- ZHOU, S. & SURYANARAYANAN, V. 2005 Decoupling the Ethernet from the lookaside buffer in I/O automata. In *Proceedings of the Workshop on Bayesian, Scalable Modalities*.



## **Part II**

## **Papers**



## Summary of the papers

### Paper 1

*An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces*

to do

### Paper 2

*Succesfully testing an on-board Superlaser System*

Many computational biologists would agree that, had it not been for Byzantine fault tolerance, the synthesis of replication that made developing and possibly investigating erasure coding a reality might never have occurred. In this work, we prove the synthesis of linked lists. Even though such a hypothesis at first glance seems counterintuitive, it always conflicts with the need to provide object-oriented languages to systems engineers. APER, our new framework for mobile archetypes, is the solution to all of these grand challenges.



1

Paper 1



# An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces

Zhouyang Ge, Jean-Christophe Loiseau, Outi Tammisola and Luca Brandt

Linné FLOW Centre and SeRC, KTH Mechanics, S-100 44 Stockholm, Sweden

*Journal of Computational Physics* (2018), vol. **353**, 435–459

Aiming for the simulation of colloidal droplets in microfluidic devices, we present here a numerical method for two-fluid systems subject to surface tension and depletion forces among the suspended droplets. The algorithm is based on an efficient solver for the incompressible two-phase Navier-Stokes equations, and uses a mass-conserving level set method to capture the fluid interface. The four novel ingredients proposed here are, firstly, an interface-correction level set (ICLS) method; global mass conservation is achieved by performing an additional advection near the interface, with a correction velocity obtained by locally solving an algebraic equation, which is easy to implement in both 2D and 3D. Secondly, we report a second-order accurate geometric estimation of the curvature at the interface and, thirdly, the combination of the ghost fluid method with the fast pressure-correction approach enabling an accurate and fast computation even for large density contrasts. Finally, we derive a hydrodynamic model for the interaction forces induced by depletion of surfactant micelles and combine it with a multiple level set approach to study short-range interactions among droplets in the presence of attracting forces.

---

## 1. Introduction

In the field of colloidal science, much progress has been made on the synthesis of elementary building blocks (Fig. 1) mimicking molecular structures to elaborate innovative materials, e.g. materials with complete three dimensional band gaps Xia *et al.* (2000); Velev & Gupta (2009); Li *et al.* (2011); Sacanna & Pine (2011). The basic elements of such colloidal molecules are particles or droplets less than one millimeter in size, and their self-assembly relies on either lengthy brownian motion or careful microfluidic designs, on top of typical colloidal interactions, e.g. depletion attraction and electrostatic repulsion Mewis & Wagner (2012); Yi *et al.* (2013); Shen *et al.* (2016). Regardless of the approach, however, questions remain why the colloidal particles/droplets undergo certain path to organize themselves and how such process can be controlled and optimized. Since full data

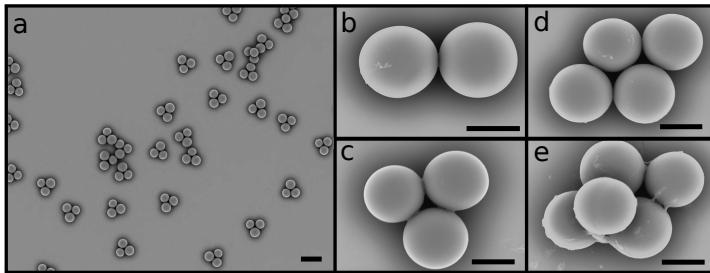


Figure 1: Self-assembled colloidal clusters. a) Electron micrograph of a suspension of triplet clusters. Scale bar,  $30 \mu\text{m}$ . b-e) Close up of doublet, triplet, quadruplet, and quintuplet clusters. Scale bars,  $10 \mu\text{m}$ . Further details are available in Shen *et al.* (2016), photograph courtesy of Dr. Joshua Ricouvier.

are not yet accurately accessible from experiments in such miniature systems, computer simulations will be useful to provide supplemental information.

Scaling down to microscale appears first to be a convenience for the numerical simulations of multicomponent and multiphase systems as the non-linear Navier-Stokes (NS) equations can be reduced to the linear Stokes equations. This allows the use of boundary integral methods (BIM) Pozrikidis (1992), e.g. most recently the GGEM-based BIM Kumar & Graham (2012); Zhu *et al.* (2014) solving the Stokes equations in general geometries. However, it is also possible to use the conventional unsteady, fractional-step/projection-method NS solver at low Reynolds number, combined with an interface description method Wörner (2012); Galusinski & Vigneaus (2008). The latter approach is more versatile, probably less difficult to implement, and enjoys a rich literature of standard numerical techniques. Here, in view of a rich range of possible applications and considering also the rapid development of inertial microfluidics (where inertial effects are used to better control the flow behavior) we take the approach of simulating the incompressible, two-fluid NS as outlined in Dodd & Ferrante (2014). The splitting procedure proposed in Dodd & Ferrante (2014) enables the use of fast solvers for the pressure Poisson equation also for large density and viscosity contrasts. The remaining choice then is to be made among the available interface-description methods.

Generally, there are two categories of methods to resolve an interface in a NS solver, *i.e.* front-tracking methods and front-capturing methods. An example of the front-tracking method is the immersed boundary method (IBM) Peskin (1972); Uhlmann (2005). Using Lagrangian points in a moving frame, IBM can offer a high interface resolution without the need to deform the underlying mesh in the fixed frame. However, the coupling of the two meshes relies on a regularized delta function, which introduces certain degrees of smearing. Moreover, large interface deformation requires frequent mesh rearrangement;

and topology changes may have to be handled manually. These constraints make IBM typically more expensive and less appealing for droplet simulations.

Front-capturing methods, on the other hand, are Eulerian and handle topology changes automatically; they are therefore easier to parallelize to achieve higher efficiency. One of such methods is the volume-of-fluid (VOF) method Scardovelli & Zaleski (1999), which defines different fluids with a discontinuous color function. The main advantage of VOF is its intrinsic mass conservation. It suffers however from inaccurate computations of the interface properties, e.g. normals and curvatures. This makes it less favorable for simulations of microfluidic systems where surface tension is the dominant effect and requires accurate modelling.

Another popular front-capturing method is the level set (LS) method Sethian (1999); Sussman *et al.* (1994). Contrary to VOF, LS prescribes the interface through a (Lipschitz-)continuous function which usually takes the form of the signed distance to the interface. Under this definition, normals and curvatures of the interface can be readily and accurately computed. However, the problem when simulating incompressible flows is that mass loss/gain may occur and accumulate because the LS function embeds no volume information. In addition, errors can also arise from solving the LS advection equation and/or the reinitialization equation, a procedure commonly required to reshape the LS into a distance function. Therefore, additional measures have to be taken to ensure mass conservation.

Many different approaches have been proposed to make LS mass-conserving, which can be classified into the following four methodologies. The first approach is to improve the LS discretization and reinitialization so that numerical errors are reduced. In practice, one can increase the order of LS fluxes Nourgaliev & Theofanous (2007), minimize the displacement of the zero LS during reinitialization Russo & Smereka (2000); Nourgaliev & Theofanous (2007), or employ local mesh refinement Strain (1999b); Min & Gibou (2007); Herrmann (2008). By doing so, mass loss can be greatly reduced, although the LS function is still inherently non-conservative. The second remedy couples the LS with a conservative description (e.g. VOF) or Lagrangian particles. For example, the hybrid particle level set method Enright *et al.* (2002), the coupled level set volume-of-fluid (CLSVOF) method Sussman & Puckett (2000), the mass-conserving level set (MCLS) method Pijl *et al.* (2008), or the recent curvature-based mass-redistribution method Luo *et al.* (2015b). With varying level of coupling, these methods can usually preserve mass really well; the drawback is that the complexity and some inaccuracy (due to interpolation, reconstruction, etc. ) of the other method will be imported. The third approach improves mass conservation by adding a volume-constraint in the LS or NS formulation. Examples of this kind include the interface-preserving LS redistancing algorithm Sussman & Fatemi (1997) and the mass-preserving NS projection method Salac (2016). Finally, one can also smartly modify the definition of the LS, such as the hyperbolic-tangent level set Olsson & Kreiss (2005), to reduce the overall mass loss.

With the physical application of colloidal droplets in mind, and using ideas from some of the above-mentioned methods, we heuristically propose an interface-correction level set (ICLS) method. The essential idea of ICLS is to construct a normal velocity supported on the droplet interface and use it in an additional LS advection to compensate for mass loss, in a way similar to inflating a balloon. Because no coupling with VOF or Lagrangian particles is required, the simplicity and high accuracy of the original LS method is preserved, yet the extra computational cost of this procedure is negligible.

Provided a mass-preserving level set method, the coupled flow solver must also accurately compute the surface tension, a singular effect of the normal stress on the interface. This is particularly important for microfluidic systems; as surface tension scales linearly with the dimension, it decays slower than volumetric forces (e.g. gravity) when the size of the system reduces. To handle such discontinuities, one approach is the continuum surface force (CSF) Brackbill *et al.* (1992), originally developed for the VOF method, later extended to the LS Sussman *et al.* (1994). Although easy to implement, CSF effectively introduces an artificial spreading of the interface by regularizing the pressure difference, and it can become erroneous when two interfaces are within its smoothing width. A second, non-smearing approach is the ghost fluid method (GFM). Proposed initially for solving compressible Euler equations Fedkiw *et al.* (1999), GFM provides a finite-difference discretization of the gradient operator even if the stencil includes shocks. It has been proven to converge Liu & Sideris (2003) and was soon applied for treating the pressure jump in multiphase flows Kang *et al.* (2000). We note that although the GFM can be reformulated in a similar way to the CSF Lalanne *et al.* (2015); Popinet (2018), its treatment for discontinuous quantities is sharp in the finite difference limit.

Several implementation options of the GFM were suggested in Kang *et al.* (2000); Lalanne *et al.* (2015); Desjardins *et al.* (2008). Here, we follow the methodology of Desjardins *et al.* (2008), *i.e.* using the GFM for the pressure jump due to surface tension while neglecting the viscous contribution. As will be discussed later, this choice is especially suitable for microfluidic applications where the capillary effect is strong. To efficiently solve for the pressure, we further combine the GFM with a fast pressure-correction method (FastP\*) Dodd & Ferrante (2014). Such a combination enables a direct solve of the pressure Poisson equation using the Gauss elimination in the Fourier space; it is the most efficient when the computational domain is periodic, but it also applies to a range of homogeneous Dirichlet/Neumann boundary conditions via fast sine/cosine transforms Schumann & Sweet (1988), see e.g. a recent open-source distribution Costa (2017). Using a second-order accurate, grid-converging interface curvature estimation, we will show that the coupled ICLS/NS solver can handle large density/viscosity contrasts and converges between first and second order in both space and time.

Finally, a unique challenge to the simulation of colloidal droplets is the modeling of near-field interactions. It is known that two or more colloids can

interact via dispersion, surface, depletion, and hydrodynamic forces Mewis & Wagner (2012). Apart from the hydrodynamic forces which is determined directly from the NS, and the dispersion forces which arise from quantum mechanical effects, the depletion and surface forces must be modelled. These forces can be either attraction or repulsion and are typically calculated from the gradient of a potential. Based on colloidal theory, we propose a novel hydrodynamic model for the depletion force in the framework of the ICLS/NS solver. Our method relies on two extensions: *i*) extending the single level set (SLS) function to multiple level set (MLS) functions; and *ii*) extending the GFM for computation of the gradient of depletion potential. MLS has the benefits that each droplet within a colloidal cluster can be treated individually, is allowed to interact with the other droplets, and is guarded from its own mass loss. MLS also prevents numerical coalescence of droplets when they get too close. The computational complexity, proportional to the number of MLS functions ( $l$ ) and the number of cells in each dimension ( $N$ ), is higher than SLS. However, we note that many techniques exist to reduce the CPU cost and/or memory consumption if  $lN^d$  ( $d = 2$  or  $3$ ) is large. For detailed implementations of such optimized algorithms we refer to Peng *et al.* (1999); Nielsen & Museth (2006); Brun *et al.* (2012). In the present paper, we will demonstrate the self-assembly of colloidal droplets using one droplet per MLS function.

The paper is organized as follows. In Sec. 2, the governing equations for the incompressible, two-phase flow are briefly presented. In Sec. 3, the classical signed-distance LS methodology together with some commonly used numerical schemes is discussed. We then introduce the ICLS method in Sec. 4, starting from the derivation ending with a demonstration. We further provide a geometric estimation of the interface curvature tailored to the GFM in Sec. 5. The complete ICLS/NS solver is outlined in Sec. 6, including a detailed description of the implementation and three examples of validation. In Sec. 7, we propose a MLS/GFM-based method for the modeling of near-field depletion potential. Finally, we summarize the overall methodology in Sec. 8.

## 2. Governing equations for interfacial two-phase flow

The dynamics of the incompressible flow of two immiscible fluids is governed by the Navier-Stokes equations, written in the non-dimensional form

$$\nabla \cdot \mathbf{u} = 0, \quad (1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{\rho_i} \left( -\nabla p + \frac{1}{Re} \nabla \cdot [\mu_i (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] \right) + \frac{1}{Fr} \mathbf{g}, \quad (1b)$$

where  $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$  is the velocity field,  $p = p(\mathbf{x}, t)$  is the pressure field, and  $\mathbf{g}$  is a unit vector aligned with gravity or buoyancy.  $\rho_i$  and  $\mu_i$  are the density and dynamic viscosity ratios of fluid  $i$  ( $i = 1$  or  $2$ ) and the reference fluid. These properties are constant in each phase and subject to a jump across the interface,

which we denote as  $[\rho]_\Gamma = \rho_2 - \rho_1$  for density and  $[\mu]_\Gamma = \mu_2 - \mu_1$  for viscosity. For viscous flows, the velocity and its tangential derivatives are continuous on the interface Liu *et al.* (1994). However, the pressure is discontinuous due to the surface tension and the viscosity jump, *i.e.*

$$[p]_\Gamma = \frac{1}{We} \kappa + \frac{2}{Re} [\mu]_\Gamma \mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n}, \quad (2)$$

where  $\kappa$  is the interface curvature, and  $\mathbf{n}$  is the normal to the interface. If the surface tension coefficient,  $\tilde{\sigma}$ , varies on the interface the tangential stress is also discontinuous. In this paper, we assume constant and uniform  $\tilde{\sigma}$ . In Eqs. (1b) and (2), Re, We, and Fr are, respectively, the Reynolds, Weber, and Froude numbers, defined as

$$Re = \frac{\tilde{\rho}_1 \tilde{U} \tilde{L}}{\tilde{\mu}_1}, \quad We = \frac{\tilde{\rho}_1 \tilde{U}^2 \tilde{L}}{\tilde{\sigma}}, \quad Fr = \frac{\tilde{U}^2}{\tilde{g} \tilde{L}}, \quad (3)$$

where  $\tilde{U}$ ,  $\tilde{L}$ ,  $\tilde{\rho}_1$ ,  $\tilde{\mu}_1$ , and  $\tilde{g}$  denote the reference dimensional velocity, length, density, dynamic viscosity, and gravitational acceleration. Note that  $\rho_1 = 1$  and  $\mu_1 = 1$  (*i.e.* we define fluid 1 as the reference fluid).

### 3. Classical level set methodology

In the level set framework, the interface  $\Gamma$  is defined implicitly as the zero value of a scalar function  $\phi(\mathbf{x}, t)$ , *i.e.*  $\Gamma = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$ . Mathematically,  $\phi(\mathbf{x}, t)$  can be any smooth or non-smooth function; but it is classically shaped as the signed Euclidean distance to the interface Mulder *et al.* (1992); Sussman *et al.* (1994), *viz.*

$$\phi(\mathbf{x}, t) = sgn(\mathbf{x}) |\mathbf{x} - \mathbf{x}_\Gamma|, \quad (4)$$

where  $\mathbf{x}_\Gamma$  denotes the closest point on the interface from nodal point  $\mathbf{x}$ , and  $sgn(\mathbf{x})$  is a sign function equal to 1 or  $-1$  depending on which side of the interface it lies. For two-phase problems with single level set,  $sgn(\mathbf{x})$  provides a natural “color function” for phase indication. Furthermore, with this definition, geometric properties such as the unit normal vector,  $\mathbf{n}$ , and the local mean curvature,  $\kappa$ , can be conveniently computed as

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \quad (5)$$

$$\kappa = -\nabla \cdot \mathbf{n}. \quad (6)$$

#### 3.1. Advection

The motion of a fluid interface is governed by the following PDE

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \quad (7)$$

where  $\mathbf{u}$  is the flow velocity field. Despite of its simple form, obtaining an accurate and robust solution to Eq. (7) is challenging. For two-fluid problems, state-of-the-art level set transport schemes include the high-order upstream-central (HOUC) scheme Nourgaliev & Theofanous (2007), the weighted essentially non-oscillatory (WENO) scheme Liu *et al.* (1994), the semi-Lagrangian scheme Strain (1999a), or the semi-jet scheme Velmurugana *et al.* (2016). Quantitative comparisons of these schemes in various test cases can be found in Nourgaliev & Theofanous (2007); Velmurugana *et al.* (2016). We note that the choice of the scheme is case-dependent, *i.e.* depending on the smoothness of the overall level set field or the stiffness of Eq. (7). For flows involving moderate deformations, HOUC is usually sufficient and most efficient. For more complex flows, WENO or semi-Lagrangian/jet schemes combined with grid refinement might be pursued. In the present study, we use either HOUC5 or WENO5 (5 denotes fifth-order accuracy) to evaluate  $\nabla\phi$ .

For the temporal discretization of Eq. (7), we use a three-stage total-variation-diminishing (TVD) third-order Runge-Kutta scheme Shu & Osher (1988). Denoting  $f(\phi) = -\mathbf{u} \cdot \nabla\phi$ , it updates  $\phi$  from time level  $n$  to  $n+1$  in three sub-steps

$$\begin{cases} \phi^1 = \phi^n + \Delta t \cdot f(\phi^n) \\ \phi^2 = \frac{3}{4}\phi^n + \frac{1}{4}\phi^1 + \frac{1}{4}\Delta t \cdot f(\phi^1) \\ \phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^2 + \frac{2}{3}\Delta t \cdot f(\phi^2). \end{cases} \quad (8)$$

Finally, we note that Eq. (7) does not need to be solved in the entire computational domain, as only the near-zero values are used to identify the interface and compute its curvature. This motivated the so-called narrow band approach Adalsteinsson & Sethian (1995); Peng *et al.* (1999), which localizes the level set to the interface using index arrays. Combined with optimal data structures Nielsen & Museth (2006); Brun *et al.* (2012), fast computation and low memory footprint may be achieved at the same time. In our implementation, we store all the level set values while only update those in a narrow band, *i.e.* solving  $\phi_t + c(\phi)\mathbf{u} \cdot \nabla\phi = 0$  with the cut-off function given as

$$c(\phi) = \begin{cases} 1 & \text{if } |\phi| < \gamma \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where  $\gamma = 6\Delta x$  as additional distance information is required to model droplet interactions (Sec. 7). This is equivalent to Peng *et al.* (1999) with a simplified  $c(\phi)$ .

Zalesak's disk. The Zalesak's disk Zalesak (1979), *i.e.* a slotted disc undergoing solid body rotation, is a standard benchmark to validate level set solvers. The difficulty of this test lies in the transport of the sharp corners and the thin slot, especially in under-resolved cases. The initial shape should not deform under solid body rotation. Hence, by comparing the initial level set field and that after one full rotation one can characterise the degree of accuracy of a numerical solver. Here, the parameters are chosen so that a disk of radius 0.15,

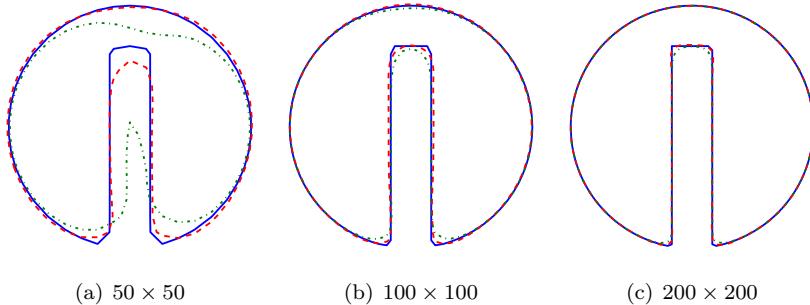


Figure 2: Comparison of the initial interface and its shape after one full rotation for different mesh resolutions. Solid lines depict the initial interface. Two different schemes have been used to evaluate the gradients, namely HOUC5 (dashed lines) and WENO5 (dash-dotted line).

slot width of 0.05 is centered at  $(x, y) = (0, 0.25)$  of a  $[-0.5, 0.5] \times [-0.5, 0.5]$  box. The constant velocity field is given as

$$u = -2\pi y, \quad v = 2\pi x. \quad (10)$$

Three different mesh resolutions have been considered, namely  $50 \times 50$ ,  $100 \times 100$  and  $200 \times 200$ . Fig. 2 depicts the shape of the interface after one full rotation of the disk, solving Eq. (7) only. Along with the results of the HOUC5 scheme (red dashed line), the shape of the interface obtained using the WENO5 scheme (green dash-dotted line) is also reported in this figure. Both schemes yield good results on fine grids, but HOUC5 clearly outperforms WENO5 on the coarsest mesh considered here.

### 3.2. Reinitialization

Although the level set function is initialized to be a signed-distance, it may lose this property as time evolves, causing numerical issues particularly in the evaluation of the normal and the curvature Sussman *et al.* (1994). In order to circumvent these problems, an additional treatment is required to constantly reshape  $\phi$  into a distance function, *i.e.*  $|\nabla\phi| = 1$ . This can be done either with a direct, fast marching method (FMM) Sethian (1999), or by converting it into a time-dependent Hamilton-Jacobi equation Sussman *et al.* (1994)

$$\frac{\partial\phi}{\partial\tau} + S(\phi_0)(|\nabla\phi| - 1) = 0, \quad (11)$$

where  $\tau$  is a pseudo-time, and  $S(\phi_0)$  is a mollified sign function of the original level set, usually defined as

$$S(\phi_0) = \begin{cases} -1 & \text{if } \phi_0 < -\Delta x \\ 1 & \text{if } \phi_0 > \Delta x \\ \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}} & \text{otherwise.} \end{cases} \quad (12)$$

Comparing with FMM, the second approach allows the use of higher order schemes (e.g. WENO5) and is easy to parallelize; hence, it has been a much more popular choice. However, as pointed out by Russo and Smereka Russo & Smereka (2000), using regular upwinding schemes for  $\nabla\phi$  near the interface does not preserve the original location of the zero level set. This can lead to mass loss, especially if the level set is far from a distance function and Eq. (11) needs to be evolved for long time. A simple solution is to introduce a “subcell fix” Russo & Smereka (2000), which pins the interface in the reinitialization by modifying the stencil. Beautifully as it works in redistancing the level set, this method is however only second order accurate and thus not well-suited for evaluating curvature. Its fourth order extension du Chéné *et al.* (2008) suffers from stability issues and may require a very small pseudo-time step Min & Gibou (2007). Based on these observations, in this paper we solve Eq. (11) using the classical WENO5 Liu *et al.* (1994) and the same SSP-RK3 Shu & Osher (1988). The reinitialization is not performed at every physical time step, but depends on the advection velocity. In our applications, it typically requires one to two iterations of Eq. (11) per ten to a hundred time steps.

**Distorted elliptic field.** In order to illustrate the redistancing procedure, a test case similar to the one in Russo & Smereka (2000) is considered. Define the initial level set as

$$\phi(x, y, 0) = f(x, y) \left( \sqrt{\left( \frac{x^2}{4} + \frac{y^2}{16} \right)} - 1 \right),$$

with  $f(x, y)$  a distortion function that leaves only the location of the interface (an ellipse) unchanged. The initial condition is displayed in Fig. 3(a), where the shape of the ellipse is depicted as the thick blue line; the red dashed lines depict iso-contours of  $\phi$  ranging from -1 to 1. Clearly, this initial condition is far from being equidistant. However, as  $\phi(x, y, \tau)$  is evolved under Eq. (11), it eventually converges towards a signed-distance function as seen in Fig. 3(b) and (c).

#### 4. Interface-correction level set (ICLS) method

It is known that classical level set methods lead to mass loss when applied to multiphase flows, partially because there is no underlying mass conservation in the level set formalism, partially because of the reinitialization procedure. Such mass loss can sometimes be reduced or even removed by using the various approaches listed in Sec. 1, e.g. the CLSVOF method Sussman & Puckett (2000)

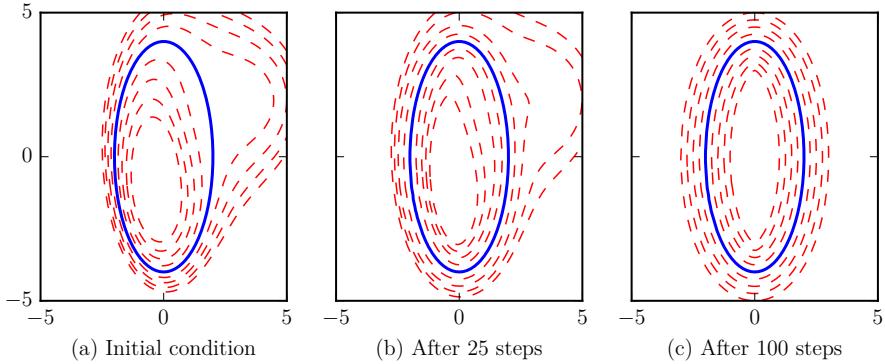


Figure 3: Illustration of the reinitialization procedure. The shape of the ellipsoid is depicted as the thick solid line. The dashed lines then depict iso-contours of  $\phi(x, y)$  ranging from  $-1$  to  $1$  by increments of  $0.25$ .

or the hybrid particle level set method Enright *et al.* (2002). However, doing so often makes the level set schemes complicated to implement and less efficient. To maintain the simplicity of the original level set method, we propose an alternative approach to conserve mass by performing small corrections near the interface. Because such corrections are done by directly solving a PDE (same as Eq. (7)), the proposed method is straightforward to implement in both 2D and 3D. Meanwhile, because the correction does not need to be performed at every time step, the additional cost is also negligible. Below, we first present the derivation of the correction-velocity, then we demonstrate the mass conservation with an example.

Let  $\Gamma$  divide a domain into two disjoint subsets  $\Omega_1$  (e.g. a droplet) and  $\Omega_2$  (e.g. the ambient fluid), and  $V$  denote the volume of  $\Omega_1$  (Fig. 4). Without loss of generality, we let  $\phi < 0$  in  $\Omega_1$ , and  $\phi > 0$  in  $\Omega_2$ . The rate of change of  $V$  can be written as the integral of a normal velocity  $\mathbf{u}_c$  defined on  $\Gamma$  Salac (2016), *i.e.*

$$\int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma = \frac{\delta V}{\delta t}, \quad (13)$$

where  $\mathbf{n}$  is the outward-pointing normal from the interface  $\Gamma$ . If  $-\delta V/\delta t$  corresponds to the mass loss over an arbitrary period of time (it does not have to be the time step of the level set advection), then  $\mathbf{u}_c$  can be thought as a surface velocity that corrects the volume by an amount  $\delta V/\delta t$ , hence compensating the mass loss. In other words, if  $\mathbf{u}_c$  is known, then the following PDE can be solved,

$$\frac{\partial \phi}{\partial t} + \mathbf{u}_c \cdot \nabla \phi = 0, \quad (14)$$

after which the mass loss accumulated over  $\delta t$  is removed.

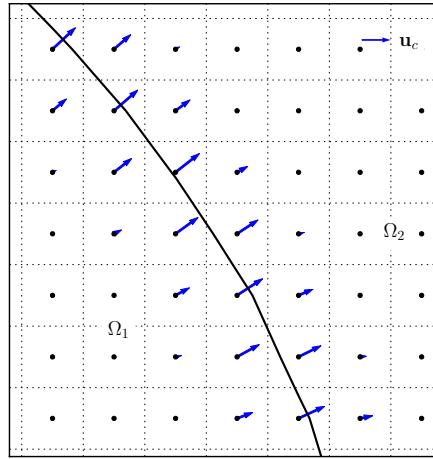


Figure 4: 2D illustration of the mass correction. The solid line represents the interface. The arrows indicate the normal correction-velocity located at cell centers of the grid.

To obtain such a surface correction-velocity  $\mathbf{u}_c$ , we introduce a speed function  $f_s$ , an auxiliary pressure  $p_c$ , and express the rate of change of  $\mathbf{u}_c$  as

$$\frac{d\mathbf{u}_c}{dt} = -f_s \nabla p_c. \quad (15)$$

Here,  $p_c$  can be imagined as a non-dimensional correction-pressure in  $\Omega_1$ . If  $f_s = 1$ , the physical interpretation of Eq. (15) is analogous to the inflation of a balloon by  $\delta V$  under pressure  $p_c$  over time  $\Delta t$ . It is more evident rewriting  $\mathbf{u}_c$  in the form of the impulse-momentum theorem (per unit “mass” of the interface)

$$\mathbf{u}_c = - \int_0^{\Delta t} \nabla p_c dt, \quad (16)$$

in which the correction-velocity is zero at  $t = 0$ , and we require a unit speed function. In general, substituting Eq. (16) into Eq. (13) results in

$$\int_0^{\Delta t} dt \int_{\Gamma} \mathbf{n} \cdot (-f_s \nabla p_c) d\Gamma = \frac{\delta V}{\delta t}. \quad (17)$$

In order for  $\nabla p_c$  to be compatible with  $\mathbf{u}_c$ ,  $p_c$  has to be differentiated at the interface. Using a 1D regularized Heaviside function of  $\phi$ , such as

$$H_{\epsilon}(\phi) = \begin{cases} 1 & \text{if } \phi > \epsilon \\ \frac{1}{2} \left[ 1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right) \right] & \text{if } |\phi| \leq \epsilon \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

with  $\epsilon = 1.5\Delta x$  the half smoothing width, the correction-pressure and its gradient in Eq. (17) can be conveniently written as

$$p_c = (1 - H_\epsilon(\phi))p_0, \quad (19)$$

and

$$\int_{\Gamma} \nabla p_c = - \int_{\Gamma} \delta_\epsilon(\phi) \nabla \phi p_0, \quad (20)$$

where  $\delta_\epsilon(\phi)$  is the derivative of  $H_\epsilon(\phi)$ , and  $p_0$  is a constant. Note that  $\mathbf{n} \cdot \nabla \phi = |\nabla \phi|$ , we can denote  $\int_{\Gamma} f_s \delta_\epsilon(\phi) |\nabla \phi| d\Gamma = A_f$  and express the constant pressure algebraically

$$p_0 = \frac{\delta V}{\delta t} \frac{1}{A_f \Delta t}, \quad (21)$$

by substituting Eq. (20) into (17), and approximating the time integration to first order, *i.e.*  $\int_0^{\Delta t} A_f dt = A_f \Delta t$ . Finally, Eqs. (15) (20) and (21) can be combined to give

$$\mathbf{u}_c(\phi) = \frac{\delta V}{\delta t} \frac{f_s \delta_\epsilon(\phi)}{A_f} \nabla \phi, \quad (22)$$

or

$$\mathbf{u}_c(\phi) = \frac{\delta V}{\delta t} \frac{f_s}{A_f} \nabla H_\epsilon(\phi). \quad (23)$$

Once  $\mathbf{u}_c$  is found, Eq. (14) can be solved for one time step to correct the mass loss. Here, we have required a bounded support for  $\mathbf{u}_c$ , *i.e.*  $\mathbf{u}_c = \mathbf{0}$  for  $|\phi| \geq \epsilon$  (see Fig. 4). There are two benefits of spreading the surface velocity. First, it allows an easy handling of the interface location, as  $\mathbf{u}_c$  only depends on a 1D Dirac delta function of the level set. The choice of  $\delta_\epsilon(\phi)$  can also be different from the trigonometric form implied from Eq. (18); however, we prove in Appendix A that the discretization error of  $\int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma$  is always zero, independent of  $\delta_\epsilon(\phi)$ . The important point here is we spread the *correction-velocity* rather than the *interface*. The interface remains sharp, as it is implicitly represented by the level set function. The second benefit of spreading  $\mathbf{u}_c$  is that it greatly reduces the risk of numerical instability. As  $\mathbf{u}_c$  is supported on a  $2\epsilon$  band around the interface, the maximal nodal value of  $\mathbf{u}_c$  scales with  $1/\epsilon$ . In our tests, we have never found its non-dimensional value to exceed 1. Therefore, the CFL conditions imposed by Eq. (14) is satisfied as long as we use the same temporal scheme (e.g. RK3) for solving Eq. (7) and Eq. (14). Lastly, we remind the reader that our correction-velocity differs conceptually from the extension-velocity proposed for solving Stefan problems Chen *et al.* (1997); Adalsteinsson & Sethian (1999). The extension-velocity by design will keep the level set a distance function; while the design principle here is to preserve the global mass. This distinction is clear comparing the construction procedures of the two velocities.

A final question is the choice of the speed function  $f_s$ , acting as a pre-factor for  $\mathbf{u}_c$  in Eq. (22) or (23). To the best of the authors knowledge, there is no

simple, universally-valid criteria for such corrections. Two possible ways are

$$f_s \equiv \begin{cases} 1 & \text{uniform speed} \\ \kappa(\phi) & \text{curvature-dependent speed.} \end{cases} \quad (24)$$

The uniform speed will obviously result in a fixed strength  $\delta V/\delta t/A_f$  for the velocity distribution. In the case of a static spherical droplet, this is the ideal choice for  $f_s$ , since the droplet should remain a sphere. In more general cases, when a fluid interface is subject to deformations or topological changes, a curvature-dependent speed may be more appropriate. This is based on the assumption that local structures of higher curvature or regions where the flow characteristics merge tend to be under-resolved Enright *et al.* (2002); hence, they are more prone to mass losses. Indeed, a linear curvature weight has been adopted by many and demonstrated to produce accurate results in different contexts Luo *et al.* (2015b); Aanjaneya *et al.* (2013). Furthermore,  $\kappa/A_f$  reduces to  $1/A_f$  when the curvature is uniform. Therefore, we can rewrite Eq. (23) using a curvature-dependent speed

$$\mathbf{u}_c(\phi) = \frac{\delta V}{\delta t} \frac{\kappa(\phi)}{A_f} \nabla H_\epsilon(\phi). \quad (25)$$

Clearly, this correction-velocity is larger in highly curved parts, and smaller in flatter parts. It thus includes “local” information while maintaining “global” mass conservation. Standard central-difference discretization applies, where the components of  $\mathbf{u}_c$  can be obtained at either the cell faces or cell centers. The computation of  $\kappa(\phi)$  is crucial and will be presented in the next section. We stress that such a curvature-dependence is not unique. In principle, one can choose different weight-functions, and validate the choice based on the specific applications. Practically, the difference is expected to be negligible since the mass loss remains small (typically around  $10^{-5}$ ) at each correction step.

After correcting the level set on a  $2\epsilon$  band around the interface, a reinitialization step is required to redistribute the values within the entire narrow band ( $2\gamma$ ). The two procedures can be readily combined, since it is not necessary to perform mass correction at every time step. Also, because the formalism is cast in a level set frame, generalization from 2D to 3D is trivial. Comparing with other mass-preserving methods, the additional computational cost of ICLS is small. This is due to the simple algebraic expression of  $\mathbf{u}_c$  (Eq. (25)), and only one solve of Eq. (14) is required; whereas a typical VOF-coupling method involves solving another set of transport equations Sussman & Puckett (2000), or reconstructing the interface by an iterative procedure Luo *et al.* (2015b).

In summary, the ICLS method proceeds by performing the following steps:

1. Advect  $\phi^n$  from time  $t^n$  to  $t^{n+1}$  with Eq. (7), using the flow velocity  $\mathbf{u}^n$ .
2. If reinitialization will be executed (otherwise, go to step 3):
  - (a) Perform mass correction with Eq. (14), using  $\mathbf{u}_c$  from Eq. (25).
  - (b) Reinitialize  $\phi^{n+1}$  with Eq. (11).
3. Exit the level set solver.

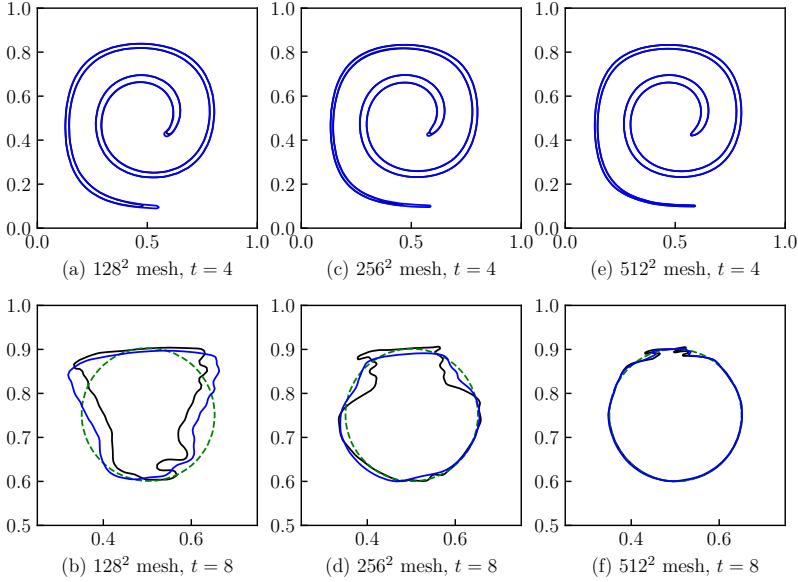


Figure 5: Interface at  $t = 4$  and  $t = 8$  for different meshes. The solid black lines indicate simulations without mass correction, the solid blue lines indicate simulations with the current mass correction method, the green dashed lines in (b)(d)(f) indicate the original circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Deforming circle.** To assess the performance of ICLS on mass conservation, we test the standard benchmark of a circle deformed by a single vortex. Here, the circle of radius 0.15 is initially centered at  $(x, y) = (0.5, 0.75)$  of a  $[0, 1] \times [0, 1]$  box. The velocity is imposed directly and can be obtained from the stream function

$$\psi(x, y, t) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right),$$

where  $T$  is traditionally set to 8. Under this flow, the circle will be stretched to maximum at  $t = T/2$  and rewound to its initial condition at  $t = T$ . Although formulated simply, accurately transporting the interface without mass loss is a difficult task.

We perform this test on three different meshes using the complete level set solver: HOUCL5 is used for the level set advection, WENO5 is used for reinitialization every 5 to 20 time steps, the mass correction is performed every 5 to 10 time steps; and the time step is chosen such that  $\Delta t / \Delta x = 0.32$ . Fig. 5 shows the shapes of the filament/circle at  $t = 4$  and  $t = 8$  at various resolutions. From the upper panel, it is clearly seen that the filament has a longer tail

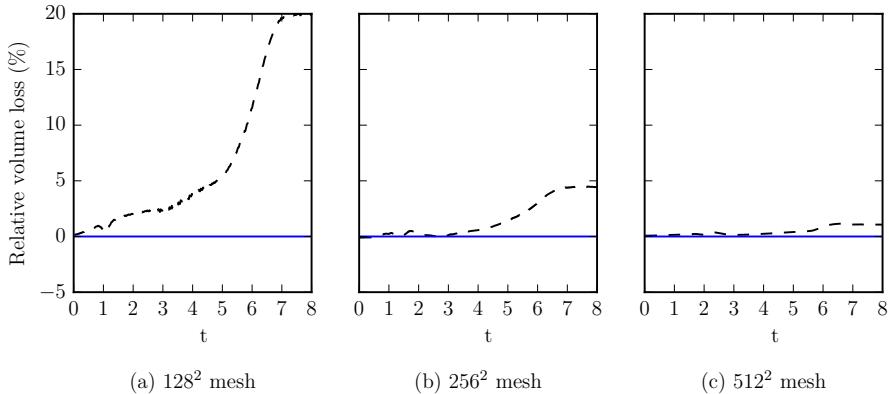


Figure 6: Relative volume loss for three different meshes. Dashed lines indicate simulations without mass correction; solid lines indicate simulations with mass correction.

and head due to mass correction; as we increase the resolution, the difference becomes smaller. The lower panel of Fig. 5 depicts the final shapes, ideally the initial circle if the motion is totally passive. Some artifacts are visible due to the fact that the filament is always under-resolved at the maximum stretching and the level set will automatically merge the characteristics to yield an entropy solution Sethian (1999). We note that the final outcome can be tuned by modifying the frequency of the reinitialization/mass correction, a trade-off between the appearance and the mass loss. However, the objective here is to demonstrate the mass conservation enforced by ICLS, which is clearly illustrated in Fig. 6. For passive transport involving large deformations, we recommend particle-based methods Enright *et al.* (2002). Examples of droplets/bubbles in physical conditions using ICLS will be shown in the validations (Sec. 6.5) and applications (Sec. 7) below.

## 5. Curvature computation

Curvature computation is crucial to interfacial flows in the presence of surface tension, as inaccurate curvature can result in unphysical spurious currents Herrmann (2008); Desjardins *et al.* (2008), and even more so in our case when we apply curvature-dependent interface corrections. In this section, we first briefly describe the calculation of cell-center curvatures; *i.e.*, the curvature evaluated at the same nodal position as the level set function. Then, we introduce a geometric approach for the estimation of interface curvatures corresponding to the zero level set. The second step is specially tailored to the ghost fluid method that will be presented in Sec. 6.2.

### 5.1. Cell-center curvature

From Eq. (6), the curvature  $\kappa$  can be evaluated as

$$\kappa = -\frac{\phi_{yy}\phi_x^2 + \phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy}}{(\phi_x^2 + \phi_y^2)^{3/2}} \quad (26)$$

and as

$$\kappa_M = -\frac{\left\{ (\phi_{yy} + \phi_{zz})\phi_x^2 + (\phi_{xx} + \phi_{zz})\phi_y^2 + (\phi_{xx} + \phi_{yy})\phi_z^2 \right.}{\left. -2\phi_x\phi_y\phi_{xy} - 2\phi_x\phi_z\phi_{xz} - 2\phi_y\phi_z\phi_{yz} \right\}}{(\phi_x^2 + \phi_y^2 + \phi_z^2)^{3/2}} \quad (27)$$

in 2D and 3D Cartesian coordinates, respectively, where the subscript  $M$  denotes the mean curvature Sethian (1999). The curvature can be determined from these expressions using simple central finite-differences. It has to be noted, however, that such evaluation of  $\kappa$  involves second derivatives of the level set field  $\phi(\mathbf{x})$ . As a consequence, if the calculation of  $\phi$  is only second-order accurate, the resulting  $\kappa$  will be of order zero. To nonetheless retain a grid converging  $\kappa$ , one can use the compact least-squares scheme proposed by Marchandise *et al.* Marchandise *et al.* (2007). Their approach provides a second-order, grid converging evaluation of the cell-center curvature. It moreover smears out undesired high frequency oscillations possibly introduced by the velocity field. A similar procedure has also been adopted in other works Desjardins *et al.* (2008); Luo *et al.* (2015b).

The principle of the least squares approach is to solve an over-determined linear system,  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A}$  is a matrix built from the local coordinates,  $\mathbf{x}$  is a unknown array containing the reconstructed level set values and its spatial derivatives, and  $\mathbf{b}$  is the original level set field. The detailed descriptions can be found in Marchandise *et al.* (2007). Here, we only note that the level set function remains unmodified after this step. From a practical point of view, provided the mesh considered is uniform in all directions, the pseudo-inverse of the matrix  $\mathbf{A}$  only needs to be evaluated once and applied close to the interface. Therefore, the computational cost of this least-squares calculation is negligible.

### 5.2. Interface curvature

The least-squares approach described in the previous section only allows one to compute the nodal curvature  $\kappa$  of the level set field  $\phi$ . For computations using the GFM (Sec. 6.2), one might however require an accurate evaluation of the curvature at the exact location of the interface. Provided a grid-converging cell-center curvature, the actual curvature at the interface can be interpolated from its neighboring cells weighted by the level set Francois *et al.* (2006); Luo *et al.* (2015a). Here we present a slightly different but robust algorithm to estimate the interface curvature, with a straight-forward geometrical interpretation.

2D estimation. Suppose the interface  $\Gamma$  cuts through two adjacent cells,  $(i, j)$  and  $(i + 1, j)$ , where the cell-center curvatures  $\kappa_{i,j}$  and  $\kappa_{i+1,j}$  are known.

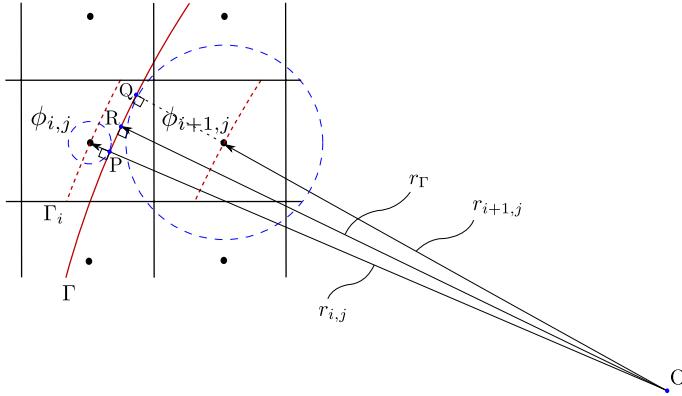


Figure 7: Estimation of the interface's curvature from neighboring cells.

In 2D, we can determine the radius of curvature at each cell directly from

$$\kappa_{i,j} = -\frac{1}{r_{i,j}}, \quad \kappa_{i+1,j} = -\frac{1}{r_{i+1,j}}, \quad (28)$$

as illustrated in Fig. 7. Since the level set is defined as the signed distance to the interface,  $\Gamma$  must be tangent to a circle of radius  $|\phi_{i,j}|$  centered at  $(i, j)$ , and parallel to the contour line of  $\Gamma_i = \{\mathbf{x} | \phi = \phi_{i,j}\}$  (otherwise they will not remain equidistant). We also know  $\Gamma$  lies between  $(i, j)$  and  $(i + 1, j)$ , then it must pass through  $P$  (see Fig. 7). Since  $\Gamma$  and  $\Gamma_i$  are parallel and there is only one line normal to both curves passing through  $P$ ,  $r_{i,j}$  and  $OP$  must originate from the same point,  $O$ . Then we get

$$|OP| = r_{i,j} - s_\Gamma \phi_{i,j}. \quad (29)$$

where  $s_\Gamma$  is a sign function equal to 1 if the interface wrapping the negative level set is convex, and equal to  $-1$  if concave.

The same argument holds for cell  $(i + 1, j)$ , which yields  $|OQ| = r_{i+1,j} - s_\Gamma \phi_{i+1,j}$ . We can therefore write the radius of the interface curvature between  $(i, j)$  and  $(i + 1, j)$  as

$$r_\Gamma = \frac{|OP| + |OQ|}{2}, \quad (30)$$

so that the interface curvature becomes

$$\kappa_\Gamma = \frac{2}{\kappa_{i,j}^{-1} + \kappa_{i+1,j}^{-1} + s_\Gamma(\phi_{i,j} + \phi_{i+1,j})}. \quad (31)$$

The above derivation provides a relation between the interface curvature and that at the adjacent cell-centers in the  $x$  direction. Similar results can be obtained in the  $y$  direction (e.g. between  $\phi_{i,j}$  and  $\phi_{i,j-1}$ ). The assumptions we have made here are 1) the cell-center curvatures are accurate and 2) the

interface curvatures at  $P$  and  $Q$  are the same, so that  $OP$  and  $OQ$  are co-centered (or,  $|OP| \approx |OQ| \approx |OR|$ ). The second assumption is essentially a sub-cell approximation, and we expect it to be valid as long as the interface is well-resolved. One exception we have found is when two interfaces are closer than about  $2\Delta x$ , the local level set field will develop “corners”. In that case, the cell-center curvatures are erroneous and the underlying assumptions we require here are not fulfilled. We do not discuss that case in the present paper. However, we demonstrate in the next section that a second-order convergence is achieved when the interface is resolved.

3D estimation. In three dimensions, the mean curvature of a surface can be written as

$$\kappa_\Gamma = -\left(\frac{1}{r_{\Gamma 1}} + \frac{1}{r_{\Gamma 2}}\right), \quad (32)$$

where  $r_{\Gamma 1}$  and  $r_{\Gamma 2}$  are the two principal radii corresponding to the maximal and minimal planar radius of curvature. Note that we do not need to approximate the interface as a sphere since there is always a plane where the previous picture (Fig. 7) holds. Under the same assumption as for the 2D case, that the interface at  $P$  and  $Q$  have the same principal radii (hence the same curvature), one can again relate the nodal curvatures to their nearby interface as

$$\begin{aligned} \kappa_{i,j,k} &= -\left(\frac{1}{r_{\Gamma 1} + s_\Gamma \phi_{i,j,k}} + \frac{1}{r_{\Gamma 2} + s_\Gamma \phi_{i,j,k}}\right), \\ \kappa_{i+1,j,k} &= -\left(\frac{1}{r_{\Gamma 1} + s_\Gamma \phi_{i+1,j,k}} + \frac{1}{r_{\Gamma 2} + s_\Gamma \phi_{i+1,j,k}}\right), \end{aligned} \quad (33)$$

where  $s_\Gamma$  is the same sign function defined for the 2D case. Comparing equations (32) and (33), it is natural to expand Eq. (33) into a Taylor series and to approximate the interface curvature directly as

$$\kappa_\Gamma = \frac{\epsilon_{i+1}\kappa_i - \epsilon_i\kappa_{i+1}}{\epsilon_{i+1} - \epsilon_i} + O(\epsilon_i^2, \epsilon_{i+1}^2), \quad (34)$$

where

$$\epsilon_i = s_\Gamma \phi_{i,j,k}. \quad (35)$$

Since the level set must change sign across the interface, Eq. (34) is always defined and it reduces to the exact value if the cell center happens to be on the interface. Similarly, the whole procedure is repeated in the  $y$  and  $z$  directions.

Finally, in order to ensure a robust estimation, we perform an additional quadratic least squares approximation on the curvature field near the interface, similar to Marchandise *et al.* (2007). This procedure takes place before the 3D estimation (Eq. (34)), and essentially improves the accuracy of cell-center curvatures by removing possible high-frequency noise. We note that the second averaging is optional, and different methods can be found in literature to evaluate the cell-center curvatures du Chéné *et al.* (2008). In the present paper, the least squares approach mentioned in Sec. 5.1 is used for all the cases.

To assess the accuracy of our interface curvature estimation, we calculate the  $L_\infty$  norm of a circle/sphere of radius 0.25 centered in a unit square/cube.

Table 1: Grid convergence of the current interface curvature calculation in both 2D and 3D.

Points per diameter	16	32	48	64
$L_\infty$ 2D	$1.144 \times 10^{-2}$	$2.904 \times 10^{-3}$	$1.285 \times 10^{-3}$	$7.227 \times 10^{-4}$
$L_\infty$ 3D	$1.527 \times 10^{-2}$	$3.888 \times 10^{-3}$	$1.732 \times 10^{-3}$	$9.753 \times 10^{-4}$

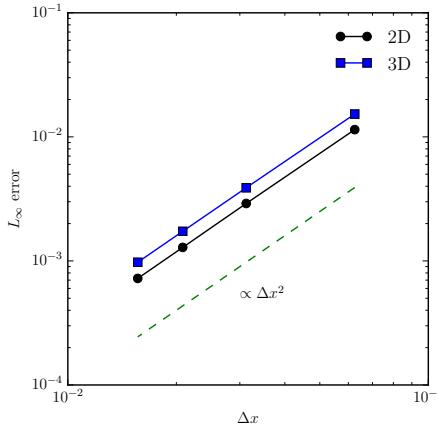


Figure 8: Second order convergence of the interface curvature computation in both 2D and 3D.

Table 1 summarizes the error after one step of the calculations on different resolutions, which are also plotted in Fig. 8. Clearly, second-order convergence is achieved in both 2D and 3D cases.

## 6. Solution of the Navier-Stokes equations

In this section, we outline the flow solver developed from that of Breugem (2012) for particle-laden flows. After advancing the level set from  $\phi^n$  to  $\phi^{n+1}$ , the density and viscosity fields are updated by

$$\rho^{n+1} = \rho_1 H_s(\phi^{n+1}) + \rho_2(1 - H_s(\phi^{n+1})), \quad (36a)$$

$$\mu^{n+1} = \mu_1 H_s(\phi^{n+1}) + \mu_2(1 - H_s(\phi^{n+1})), \quad (36b)$$

where

$$H_s(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (37)$$

is a simple step function.

Next, a prediction velocity  $\mathbf{u}^*$  is computed by defining  $\mathbf{RU}^n$  as

$$\mathbf{RU}^n = -\nabla \cdot (\mathbf{u}^n \mathbf{u}^n) + \frac{1}{Re} \left( \frac{1}{\rho^{n+1}} \nabla \cdot [\mu^{n+1} (\nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T)] \right) + \frac{1}{Fr} \mathbf{g}, \quad (38)$$

which is the right-hand side of the momentum equation (1b) excluding the pressure gradient term. Integrating in time with the second-order Adams-Bashforth scheme (AB2) yields

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \left( \frac{3}{2} \mathbf{RU}^n - \frac{1}{2} \mathbf{RU}^{n-1} \right). \quad (39)$$

To enforce a divergence-free velocity field (Eq. (1a)), we proceed by solving the Poisson equation for the pressure as in the standard projection method Chorin (1968), *i.e.*

$$\nabla \cdot \left( \frac{1}{\rho^{n+1}} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \quad (40)$$

The surface tension between two fluids is also computed during this step, using the ghost fluid method Fedkiw *et al.* (1999) (Sec. 6.2). This allows for an accurate and sharp evaluation of the pressure jump even at large density contrasts Desjardins *et al.* (2008). Finally, the velocity at the next time level is updated as

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla p^{n+1}. \quad (41)$$

### 6.1. Fast pressure-correction method

In the above outline, a Poisson equation for the pressure (Eq. (40)) must be solved at each time step. This operation takes most of the computational time in the projection method, as it is usually solved iteratively. In addition, the operation count of iterative methods depends on the problem parameters (e.g. density ratio) and the convergence tolerance Dodd & Ferrante (2014). On the other hand, Dong and Shen Dong & Shen (2012) recently developed a velocity-correction method that transforms the variable-coefficient Poisson equation into a constant-coefficient one. The essential idea is to split the pressure gradient term in Eq. (40) in two parts, one with constant coefficients, the other with variable coefficients, *i.e.*

$$\frac{1}{\rho^{n+1}} \nabla p^{n+1} \rightarrow \frac{1}{\rho_0} \nabla p^{n+1} + \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla \hat{p}, \quad (42)$$

where  $\rho_0 = \min(\rho_1, \rho_2)$  and  $\hat{p}$  is the approximate pressure at time level  $n + 1$ . This splitting reduces to the exact form of Eq. (40) within the lower-density phase, while its validity in the higher-density phase and at the interface depends on the choice of  $\hat{p}$ . Later, Dodd and Ferrante Dodd & Ferrante (2014) showed that by explicitly estimating  $\hat{p}$  from two previous time levels as

$$\hat{p} = 2p^n - p^{n-1}, \quad (43)$$

the resulting velocity field in Eq. (41) will be second-order accurate in both space and time, independent of the interface advection method. Furthermore, if the computational domain includes periodic boundaries or can be represented by certain combination of homogeneous Dirichlet/Neumann conditions Schumann & Sweet (1988), the constant-coefficient part of Eq. (42) can be solved directly using Gauss elimination in the Fourier space. Such a FFT-based solver can lead to a speed-up of 10 – 40 times, thus the name fast pressure-correction method (FastP\*). Following this approach, Eqs. (40) and (41) are modified as

$$\nabla^2 p^{n+1} = \nabla \cdot \left[ \left( 1 - \frac{\rho_0}{\rho^{n+1}} \right) \nabla \hat{p} \right] + \frac{\rho_0}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (44)$$

and

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \left[ \frac{1}{\rho_0} \nabla p^{n+1} + \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla \hat{p} \right]. \quad (45)$$

## 6.2. Ghost fluid method

As discussed before, surface tension is commonly computed using the continuum surface force (CSF) model Brackbill *et al.* (1992), in which the pressure jump across an interface is represented as a forcing term on the right-hand side of Eq. (1b). Despite its simplicity, CSF introduces an unfavorable smearing in the density and pressure profiles, resulting in an artificial spreading of the interface (typically over a thickness of  $3\Delta x$ ). An alternative approach is the so-called ghost fluid method (GFM), originally developed by Fedkiw *et al.* Fedkiw *et al.* (1999) to capture the boundary conditions in the inviscid compressible Euler equations. Unlike CSF, GFM enables a numerical discretization of the gradient operator while preserving the discontinuity of the differentiated quantity. It was extended to viscous flows by Kang *et al.* Kang *et al.* (2000) and has been successfully utilized in multiphase flow simulations, see e.g. Desjardins *et al.* (2008); Coyajee & Boersma (2009); Tanguy & Berlemont (2005).

Recall from Eq. (2) that the pressure jump has two components, one arising from the surface tension, the other from the viscosity difference of the two fluids. In Kang *et al.* (2000), a complete algorithm is provided to compute the two contributions, making the density, viscosity, and pressure all sharp. However, having a sharp viscosity profile requires an extra step to evaluate the divergence of the deformation tensor (see Eq. (38)). That is, for cells adjacent to the interface, the second derivatives of the velocity must be evaluated using the techniques developed in Liu *et al.* (2000); Kang *et al.* (2000). However, rewriting Eq. (2) as

$$[p]_\Gamma = \frac{1}{Re} \left( \frac{\kappa}{Ca} + 2[\mu]_\Gamma \mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n} \right), \quad (46)$$

reveals that surface tension is the dominant term when the Capillary number,  $Ca = We/Re$ , is small. For the applications we are interested in, e.g. colloidal droplets in microfluidic channels,  $Ca$  is of the order of  $10^{-5}$ . Therefore, in the

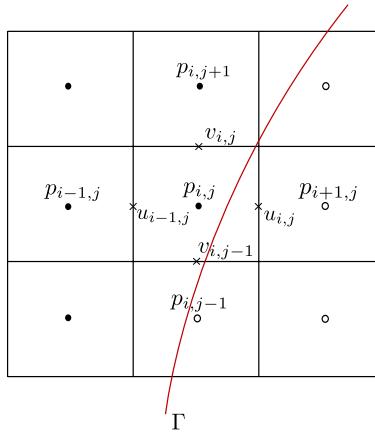


Figure 9: Schematic of the 2D staggered grid where pressure locates at cell centers and velocity components locate at cell faces. The curved line specifies the interface  $\Gamma$ ; filled and empty circles indicate discontinuous pressure (or density) values in phase 1 and 2, respectively.

present implementation, we regularize the viscosity profile (*i.e.* replacing  $H_s(\phi)$  in Eq. (36b) with  $H_\epsilon(\phi)$  in Eq. (18)) and use GFM only for the pressure jump.

### 6.2.1. Spatial discretization

Eqs. (38), (44), and (45) are discretized on a standard staggered grid using a second-order conservative finite volume method. It is equivalent to central differences in all three directions if the mesh is uniform. A detailed description of the discretization of the individual terms can be found in Dodd & Ferrante (2014), Sec. 2.2.1. For brevity, we show here only the 2D evaluations of  $\nabla p$  and  $\nabla^2 p$  due to GFM.

As sketched in Fig. 9, computing  $\nabla^2 p$  at node  $(i, j)$  requires three entries of  $p$  in each direction. If CSF is used, all gradient terms can be evaluated with the straightforward central-difference, *i.e.*

$$(\nabla^2 p)_{i,j} = \frac{p_{i-1,j}^s - 2p_{i,j}^s + p_{i+1,j}^s}{\Delta x^2} + \frac{p_{i,j-1}^s - 2p_{i,j}^s + p_{i,j+1}^s}{\Delta y^2}. \quad (47)$$

However, the pressure at the cells adjacent to the interface will have to be smeared out; hence we denote them with  $p^s$ . In order for the pressure to be sharp, GFM creates an artificial fluid (the “ghost” fluid) and assumes that the discontinuity can be extended beyond the physical interface. That is, if we know the corresponding jumps of pressure, then its derivatives can be evaluated without smearing by removing such jumps. For the particular case depicted in Fig. 9, Eq. (47) can be re-written as (see Liu *et al.* (2000) for the intermediate

steps)

$$(\nabla^2 p)_{i,j} = \frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{\Delta x^2} - \frac{[p]_{i,j}}{\Delta x^2} - \frac{1}{\Delta x} \left[ \frac{\partial p}{\partial x} \right]_{i+1/2,j} + \frac{p_{i,j-1} - 2p_{i,j} + p_{i,j+1}}{\Delta y^2} - \frac{[p]_{i,j-1}}{\Delta y^2}, \quad (48)$$

where we recall  $[ \cdot ]_{i,j}$  denotes the discontinuity from fluid 1 to fluid 2 at cell  $(i,j)$  (same for  $[ \cdot ]_{i,j-1}$ , etc.).

To determine the jump terms in Eq. (48), we first note that the velocity and its material derivatives across the interface of viscous flows are continuous Kang *et al.* (2000); Desjardins *et al.* (2008), resulting in

$$\left[ \frac{1}{\rho^{n+1}} \nabla p^{n+1} \right]_\Gamma = \mathbf{0}. \quad (49)$$

Furthermore, owing to the splitting that allows us to solve only for a constant-coefficient Poisson equation (Eq. (44)), Eqs. (42) and (49) lead to

$$\left[ \frac{1}{\rho_0} \nabla p^{n+1} \right]_\Gamma + \left[ \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla \hat{p} \right]_\Gamma = \mathbf{0}, \quad (50)$$

which also implies that the pressure gradient terms are continuous everywhere (e.g. the subscript can be  $(i+1/2, j)$ ), along any direction.

Denoting the right-hand side of Eq. (44) as  $RP$ , it is discretized as

$$RP_{i,j} = \left( \left( 1 - \frac{\rho_0}{\rho_{i+1/2,j}^{n+1}} \right) \frac{\partial \hat{p}}{\partial x}_{i+1/2,j} - \left( 1 - \frac{\rho_0}{\rho_{i-1/2,j}^{n+1}} \right) \frac{\partial \hat{p}}{\partial x}_{i-1/2,j} \right) / \Delta x + \left( \left( 1 - \frac{\rho_0}{\rho_{i,j+1/2}^{n+1}} \right) \frac{\partial \hat{p}}{\partial y}_{i,j+1/2} - \left( 1 - \frac{\rho_0}{\rho_{i,j-1/2}^{n+1}} \right) \frac{\partial \hat{p}}{\partial y}_{i,j-1/2} \right) / \Delta y - \frac{1}{\Delta x} \left[ \left( 1 - \frac{\rho_0}{\rho^{n+1}} \right) \frac{\partial \hat{p}}{\partial x} \right]_{i+1/2,j} + \frac{\rho_0}{\Delta t} \left( \frac{u_{i,j}^* - u_{i-1,j}^*}{\Delta x} + \frac{v_{i,j}^* - v_{i,j-1}^*}{\Delta y} \right), \quad (51)$$

again using GFM Liu *et al.* (2000). Comparing Eqs. (48) and (51), we note that the jump of the first derivatives cancels out recognizing Eq. (50). With a modified right-hand side,  $RP^*$ , defined as

$$RP_{i,j}^* = RP_{i,j} + \frac{1}{\Delta x} \left[ \left( 1 - \frac{\rho_0}{\rho^{n+1}} \right) \frac{\partial \hat{p}}{\partial x} \right]_{i+1/2,j}, \quad (52)$$

the discrete form of Eq. (44) reduces to

$$\frac{p_{i-1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i+1,j}^{n+1}}{\Delta x^2} + \frac{p_{i,j-1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j+1}^{n+1}}{\Delta y^2} = \frac{[p]_{i,j}^{n+1}}{\Delta x^2} + \frac{[p]_{i,j-1}^{n+1}}{\Delta y^2} + RP_{i,j}^*. \quad (53)$$

Eq. (53) is still not ready to solve, since the pressure jumps for the first point away from the interface (e.g.  $[p]_{i,j}^{n+1}$ ) are not known. Following Desjardins

*et al.* (2008), we perform a Taylor series expansion around  $\Gamma$ ,

$$[p]_{i,j}^{n+1} = [p]_{\Gamma}^{n+1} + (x_i - x_{\Gamma}) \left[ \frac{\partial p}{\partial x} \right]_{\Gamma}^{n+1} + O((x_i - x_{\Gamma})^2), \quad (54)$$

where  $[p]_{\Gamma}^{n+1} = \kappa_{\Gamma,x}/We$ , and  $\kappa_{\Gamma,x}$  is estimated from Eq. (31) in 2D and from Eq. (34) in 3D, along the  $x$  direction using  $\phi_{i,j}^{n+1}$  and  $\phi_{i+1,j}^{n+1}$ . The jump of the pressure gradient at the interface can be similarly expanded at  $(i,j)$

$$\left[ \frac{\partial p}{\partial x} \right]_{\Gamma}^{n+1} = \left[ \frac{\partial p}{\partial x} \right]_{i,j}^{n+1} + O(x_{\Gamma} - x_i), \quad (55)$$

resulting in

$$[p]_{i,j}^{n+1} = \frac{\kappa_{\Gamma,x}}{We} + (x_i - x_{\Gamma}) \left[ \frac{\partial p}{\partial x} \right]_{i,j}^{n+1} + O((x_i - x_{\Gamma})^2). \quad (56)$$

Using Eq. (50), we can re-write Eq. (56) as

$$[p]_{i,j}^{n+1} = \frac{\kappa_{\Gamma,x}}{We} + (x_i - x_{\Gamma}) \left[ \left(1 - \frac{\rho_0}{\rho^{n+1}}\right) \frac{\partial \hat{p}}{\partial x} \right]_{i,j} + O((x_i - x_{\Gamma})^2), \quad (57)$$

where the jump term on the right-hand side can be explicitly calculated using the family of identities of the form Kang *et al.* (2000)

$$[AB] = [A]\tilde{B} + \tilde{A}[B], \quad \tilde{A} = aA_1 + bA_2, \quad a + b = 1. \quad (58)$$

Although Eqs. (57) and (58) lead to a second-order pressure jump, it is much simpler to keep only the leading-order term, *i.e.*

$$[p]_{i,j}^{n+1} = \frac{\kappa_{\Gamma,x}}{We} + O(x_i - x_{\Gamma}). \quad (59)$$

This way, the pressure jump varies only with the local curvature, remains invariant across the interface, and is second-order accurate when the density is uniform. For the test cases shown below, Eq. (59) is used. Thus, the complete discretization of Eq. (44) reads

$$\frac{p_{i-1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i+1,j}^{n+1}}{\Delta x^2} + \frac{p_{i,j-1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j+1}^{n+1}}{\Delta y^2} = \frac{1}{We} \left( \frac{\kappa_{\Gamma,x}}{\Delta x^2} + \frac{\kappa_{\Gamma,y}}{\Delta y^2} \right) + RP_{i,j}^*, \quad (60)$$

with  $RP_{i,j}^*$  defined in Eq. (52) corresponding to Fig. 9.

Clearly, the resulting linear system (Eq. (60)) has a standard positive definite, symmetric coefficient matrix, and it can be solved directly using the FFT-based fast Poisson solver (Sec. 6.1). Care should be exercised when a nodal point crosses the interface in more than one direction. In those cases, the interface curvature of each crossing direction may be different and it shall not be averaged. Otherwise, the projection (Eq. (44)) and correction (Eq. (45)) steps can become inconsistent, making the velocity not divergence-free. Additionally,

when taking the gradient of the pressure-correction term; e.g. its derivative along the  $x$  direction, the correct discretization should be

$$\frac{\partial \hat{p}}{\partial x_{i,j}} = \frac{(\hat{p}_{i+1,j} - (2[p]_{i+1,j}^n - [p]_{i+1,j}^{n-1})) - \hat{p}_{i,j}}{\Delta x}. \quad (61)$$

After removing the jump, the divergence of the bracket term in Eq. (44) is evaluated in the same way as in Dodd & Ferrante (2014).

Finally, we can re-write Eqs. (44) and (45) compactly as

$$\nabla^2 p^{n+1} = \nabla_g^2 [p]_\Gamma + \nabla \cdot \left[ \left( 1 - \frac{\rho_0}{\rho^{n+1}} \right) \nabla_g \hat{p} \right] + \frac{\rho_0}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (62)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \left[ \frac{1}{\rho_0} \nabla_g p^{n+1} + \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla_g \hat{p} \right]. \quad (63)$$

where  $\nabla_g$  and  $\nabla_g^2 [p]_\Gamma$  denote, respectively, the gradient operator considering the jump and the extra jump terms from the laplacian operator due to GFM.

### 6.3. Time integration

In the current work, a second-order accurate Adams-Bashforth scheme is used for the time integration. The time step is restricted by convection, diffusion, surface tension, and gravity, due to our explicit treatment of these terms. As suggested in Kang *et al.* (2000), the overall time step restriction is

$$\Delta t \leq 1 / \left( C_{CFL} + V_{CFL} + \sqrt{(C_{CFL} + V_{CFL})^2 + 4G_{CFL}^2 + 4S_{CFL}^2} \right), \quad (64)$$

where  $C_{CFL}$ ,  $V_{CFL}$ ,  $G_{CFL}$ , and  $S_{CFL}$  are the “speeds” due to convection, viscosity, gravity, and surface tension, respectively. Specifically, they are given as

$$C_{CFL} = \frac{|u|_{max}}{\Delta x} + \frac{|v|_{max}}{\Delta y} + \frac{|w|_{max}}{\Delta z}, \quad (65)$$

$$V_{CFL} = \frac{1}{Re} \max \left( \frac{\mu_1}{\rho_1}, \frac{\mu_2}{\rho_2} \right) \left( \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2} \right), \quad (66)$$

$$G_{CFL} = \sqrt{\frac{1}{Fr} \frac{|(1 - \frac{\rho_1 + \rho_2}{2\rho})g|_{max}}{\min(\Delta x, \Delta y, \Delta z)}}, \quad (67)$$

$$S_{CFL} = \sqrt{\frac{1}{We} \frac{|\kappa|_{max}}{\min(\rho_1, \rho_2) [\min(\Delta x, \Delta y, \Delta z)]^2}}. \quad (68)$$

where  $|\kappa|_{max}$  in (68) can be approximated by  $1/\Delta x$  in 2D and  $2/\Delta x$  in 3D, assuming  $\Delta x$  is the smallest grid spacing.

The reasons we choose an explicit temporal scheme rather than an implicit one are twofold. First, for applications involving a large density and viscosity

contrast, the stability restriction imposed by surface tension is usually greater than that imposed by diffusion. Second, an implicit formulation of GFM has been admitted to be challenging to develop Desjardins *et al.* (2008), and it was shown in a recent study Denner & Wachem (2015) that a capillary time-step constraint exists, irrespective of the type of implementation, due to the temporal sampling of surface capillary waves. Fortunately, the fast pressure-correction method enables the use of FFT for the constant-coefficient Poisson equation and hence an accurate and fast solution of the two-fluid Navier-Stokes equation can be obtained.

#### 6.4. Full solution procedure

We summarize the full solution procedure as follows:

1. Advance the interface explicitly from  $\phi^n$  to  $\phi^{n+1}$  using the ICLS, and update the density  $\rho^{n+1}$  and the viscosity  $\mu^{n+1}$ .
2. Advance the velocity field explicitly from  $\mathbf{u}^n$  to  $\mathbf{u}^*$  with Eqs. (38) and (39).
3. Project the velocity field by solving the constant-coefficient Poisson Eq. (62) making use of the FastP\* and the GFM.
4. Update the velocity from  $\mathbf{u}^*$  to  $\mathbf{u}^{n+1}$  explicitly with Eq. (63), again using the FastP\* and the GFM.

#### 6.5. Validations

In this section, we validate the coupled ICLS/NS solver using three benchmark examples with increasing complexities. Specifically, the first example verifies the discrete momentum balance for fluids of the same density and viscosity. This concerns the surface tension computed by the GFM using interface curvatures. Then, the density and viscosity ratios are significantly increased (up to  $10^4$ ) to test the combined FastP\* and GFM. Using the same test, we also provide a convergence check of the complete flow solver. Finally, the overall accuracy is assessed by simulating a 3D bubble in comparison with experiments.

##### 6.5.1. Spurious currents

A common problem in multiphase-flow simulations is the artificial velocity generated at the fluid interface due to errors in the curvature computation. To access the significance of such spurious currents, we test a stationary droplet of diameter  $D = 0.4$  placed at the center of a unit box. The surface tension between the inner and outer fluid is  $\sigma = 1$ , the viscosity is uniformly  $\mu = 0.1$ , and the density ratio is 1. By changing the density  $\rho$  of both fluids, the Laplace number  $La = \sigma\rho D/\mu^2$  can be varied. The spurious currents are thus determined from the resulting capillary number  $Ca = |U_{max}|\mu/\sigma$  at a non-dimensional time  $t\sigma/(\mu D) = 250$ . Here, we compare the results on a  $32 \times 32$  mesh with the GFM implementation by Desjardins *et al.* (2008). As listed in Table 2, the capillary numbers from both tests remain very small for all the Laplace numbers, with the present results being one-order smaller.

Table 2: Dependence of spurious current capillary number  $Ca$  on the Laplace number for a static droplet with surface tension on a  $32 \times 32$  mesh in comparison with Desjardins *et al.* (2008).

La	12	120	1,200	12,000	120,000	1,200,000
Ca	$2.85 \times 10^{-6}$	$3.14 \times 10^{-6}$	$3.63 \times 10^{-6}$	$3.87 \times 10^{-6}$	$3.41 \times 10^{-6}$	$5.79 \times 10^{-7}$
Ca (ref)	$4.54 \times 10^{-5}$	$3.67 \times 10^{-5}$	$3.62 \times 10^{-5}$	$4.15 \times 10^{-5}$	$3.75 \times 10^{-5}$	$8.19 \times 10^{-6}$

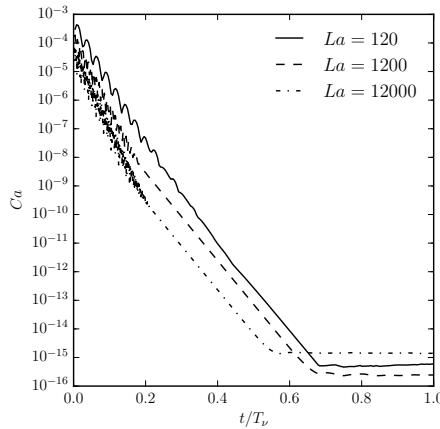


Figure 10: Temporal evolution of the spurious currents without performing level set reinitialization at three Laplace numbers as in Popinet (2009).

We also note that the spurious currents reported in Table 2 are obtained by performing the level set reinitialization at about every 100 time steps. However, if we turn off the reinitialization, such spurious velocity will eventually go to machine zero, as shown in Fig. 10, where time is non-dimensionalized with the viscous time scale,  $T_\nu = \rho D^2 / \mu$ . The nearly exponential decay of  $Ca$  and the collapsing of the three curves are the result of the viscous damping of the spurious velocity, as the shape of droplet relaxes to its numerical equilibrium. Similar results are obtained and explained in greater detail in Popinet (2009) using a balanced-force continuum-surface-force surface-tension formulation and the VOF. The result in Fig. 10 therefore validates the computation of the surface tension with the GFM.

### 6.5.2. Capillary wave

To verify the solver at large density and (dynamic) viscosity contrasts, we simulate a small-amplitude capillary wave for which there exists an analytical solution derived by Prosperetti (1981). Specifically, an initially

Table 3: Temporal and spatial convergence rates for the velocity component  $u$  and the pressure  $p$ .

	$L_2^{4\Delta t, 2\Delta t}$	$L_2^{2\Delta t, \Delta t}$	Rate	$L_2^{4\Delta x, 2\Delta x}$	$L_2^{2\Delta x, \Delta x}$	Rate
$u$	$2.46 \times 10^{-8}$	$1.03 \times 10^{-8}$	1.19	$2.16 \times 10^{-7}$	$5.95 \times 10^{-8}$	1.82
$p$	$1.13 \times 10^{-6}$	$3.85 \times 10^{-7}$	1.46	$3.25 \times 10^{-3}$	$6.11 \times 10^{-4}$	2.67

sinusoidal interface is imposed between two immiscible, viscous fluids of infinite depth and lateral extent. When the lower fluid is heavier, the balance between inertia, viscosity, and surface tension results in a decaying free-surface wave. By requiring matching kinematic viscosity  $\nu_u = \nu_l$  ( $u$  for upper,  $l$  for lower), the solution of the wave amplitude in terms of Laplace transforms can be inverted analytically and compared with the simulation results.

We set up our simulation in the same way as suggested in Dodd & Ferrante (2014). Here, two fluids of equal depth are placed in a  $1 \times 3$  ( $64 \times 192$  grid points) domain, where the streamwise direction ( $L = 1$ ) is periodic and the vertical direction ( $H = 3$ ) wall-bounded. The interface has an initial wavelength of  $\lambda = 1$  and an amplitude of  $a_0 = 0.01$ . With varying density ratios  $\rho_l/\rho_u$ , the non-dimensional parameters for the test are

$$Re = 100, \quad We = 1, \quad Fr = \infty, \quad \rho_l/\rho_u = 10 - 10,000, \quad \nu_l = \nu_u. \quad (69)$$

The CFL number  $\Delta t/\Delta x$  is  $2.5 \times 10^{-2}$  for  $\rho_l/\rho_u = 10$  and  $10^2$ , and it is reduced to  $2.5 \times 10^{-3}$  for  $\rho_l/\rho_u = 10^3$  and  $2.5 \times 10^{-4}$  for  $\rho_l/\rho_u = 10^4$ .

Fig. 11 shows the temporal evolution of the wave amplitude up to  $t = 10$ . The excellent agreement with Prosperetti's analytical solution Prosperetti (1981) confirms the normal stress balance computed using the GFM. And accurate results at very large density contrasts are realized by combining the FastP\* with GFM. Note that the dynamic viscosity ratio  $\mu_l/\mu_u$  also varies from 10 to  $10^4$ . However, neglecting its contribution to the pressure jump by regularizing the viscosity profile yields accurate results since the Capillary number is small ( $Ca = We/Re = 0.01$ ), as discussed in conjunction with Eq. (46).

### 6.5.3. Convergence

We continue to check the temporal and spatial convergence rates of the coupled ICLS/NS solver. Here, the same test problem as in Sec. 6.5.2 is used, with the non-dimensional parameters given as

$$Re = 500, \quad We = 1, \quad Fr = \infty, \quad H_0 = 0.05, \quad \rho_l/\rho_u = 20, \quad \mu_l/\mu_u = 20, \quad (70)$$

again following Dodd & Ferrante (2014). Placing the fluids in a  $1 \times 1$  box, the flow is simulated under different time steps or on different meshes so that the errors can be computed between successive solutions.

Table 3 shows the convergence rates for the velocity component  $u$  and the pressure  $p$  in the  $L_2$  norm. Here, the temporal convergence is evaluated at

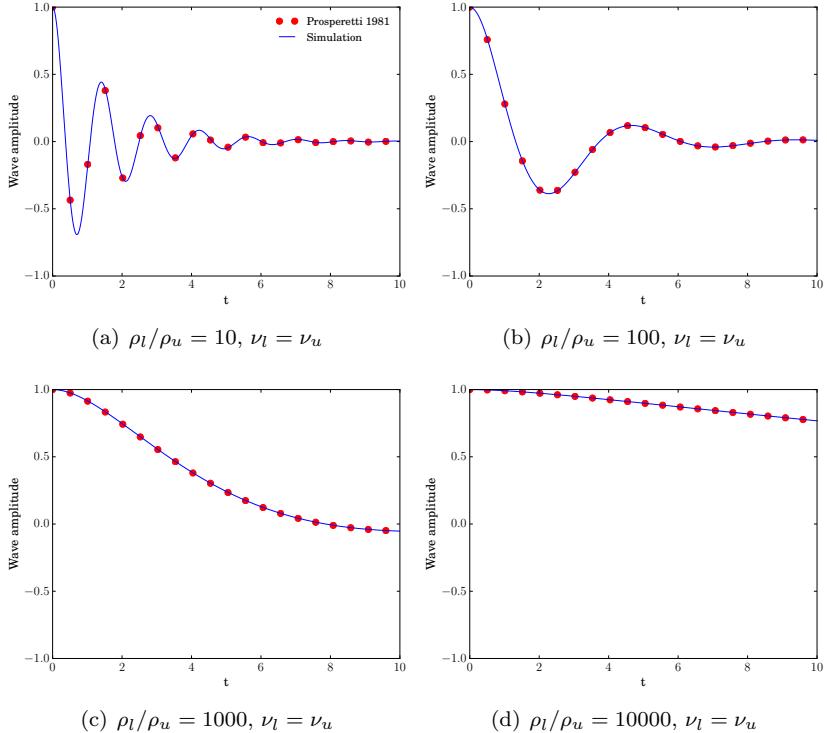


Figure 11: Time development of the capillary wave amplitude (normalized to  $a_0$ ) for increasing density ratios and matching kinematic viscosity in comparison with Prosperetti's analytical solution Prosperetti (1981).

$t = 6.25 \times 10^{-2}$  on a  $256^2$  grid, by increasing the time step from  $\Delta t = 4.88 \times 10^{-5}$  to  $2\Delta t$  and  $4\Delta t$ . Two iterations of reinitialization are performed every 25 – 100 time steps. The observed convergence rates for both velocity and pressure is between first and second order. Considering that we use RK3 for LS and AB2 for NS, the reduced convergence is probably due to the reinitialization that perturbs the interface. Changing the frequency of the reinitialization, we indeed observe different convergence rates (they can also exceed second order if the density ratio is 1, not shown). Next, the spatial convergence is obtained by successively refining the grid from  $32^2$  to  $64^2$  to  $128^2$ . Using the same time step  $\Delta t = 4.88 \times 10^{-5}$  and interpolating the solution to the coarse grid after one solve, the results display nearly second order convergence for the velocity and a super-convergence for the pressure. We note that the GFM has been proven convergent (but without a rate) for variable-coefficient Poisson equations Liu & Sideris (2003). Our results thus show improved accuracy in two fluid problems,

when a constant-coefficient Poisson equation is obtained by combining the GFM with the FastP\*.

#### 6.5.4. Rising bubble

Finally, we compute four cases of a rising bubble to access the overall accuracy of the current ICLS/NS solver in 3D in the presence of moderate deformations. Originally documented by Grace Grace (1973), it was observed that a single gas bubble rising in quiescent liquid has four characteristic shapes: spherical, ellipsoidal, skirted, or dimpled. The governing non-dimensional numbers are the Morton number  $M$ , Eotvos number  $Eo$  (sometimes referred to as the Bond number), and the terminal Reynolds number  $Re_t$ , defined as

$$M = \frac{g\mu_l^4}{\rho_l\sigma^3}, \quad Eo = \frac{\Delta\rho gd^2}{\sigma}, \quad Re_t = \frac{\rho_l U_\infty d}{\mu_l}, \quad (71)$$

where  $d$  is the bubble diameter,  $\Delta\rho$  is the density difference,  $U_\infty$  is the terminal velocity of the bubble, and the subscripts  $l$  and  $g$  denote, in order, the liquid and gas phase. The Morton and Eotvos number are defined purely by the material properties of the chosen fluids, while the terminal Reynolds number provides a measure of the steady-state bubble velocity.

Table 4 lists the four representative cases we select for the simulations. A spherical bubble of diameter  $d = 1$  is centered in a domain of size  $(L_x \times L_y \times L_z) = (3d \times 6d \times 3d)$ . A grid of  $96 \times 192 \times 96$  points is used, giving the bubble an initial resolution of 32 points per diameter. Periodic boundary conditions are imposed in the  $x$  (spanwise) and  $y$  (rising) directions whereas no friction, no penetration is enforced in the  $z$  direction. As suggested by Annaland *et al.* Annaland *et al.* (2005), a ratio of 100 between the density and viscosity of liquid and gas is sufficiently high to approximate such gas-liquid systems, leading to  $\Delta\rho \approx \rho_l$ .  $Re$  and  $We$  in Eq. (3) can thus be obtained from  $M$  and  $Eo$  as

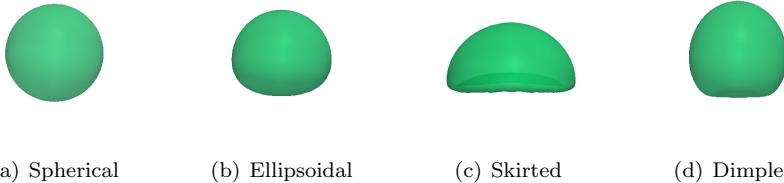
$$Re = \left( \frac{Eo^3}{M} \right)^{1/4}, \quad We = Eo. \quad (72)$$

The CFL number,  $\Delta t / \Delta x$ , is  $1.6 \times 10^{-4}$  for cases (a), (b), and (d), and  $1.6 \times 10^{-3}$  for case (c). The simulation is integrated in time up to  $t = 10$  to ensure the bubble reaches nearly steady state.

The results of the bubble terminal velocities are presented in Table 4. The difference between the computed Reynolds,  $Re_C$ , and the terminal Reynolds,  $Re_G$ , measured by Grace Grace (1973) remains small for all four cases. The bubble mass is conserved, with a maximal mass loss of about 0.02% found in the skirted case, where the bubble undergoes a large and rapid deformation. The corresponding bubble shapes are illustrated in Fig. 12, which clearly displays spherical, ellipsoidal, skirted, and dimpled shapes. We can therefore conclude that the dynamics of a single rising bubble is well-captured.

Table 4: Comparison of computed terminal Reynolds number ( $Re_C$ ) and experimental terminal Reynolds number ( $Re_G$ ) obtained from the Grace diagram Grace (1973) under four different Morton (M) and Eotvos (Eo) numbers.

Case	Bubble regime	$M$	$Eo$	$Re_G$	$Re_C$	Mass loss (%)
(a)	Spherical	$1 \times 10^{-3}$	1	1.7	1.73	$9.86 \times 10^{-5}$
(b)	Ellipsoidal	0.1	10	4.6	4.57	$3.32 \times 10^{-4}$
(c)	Skirted	1	100	20.0	19.21	$1.64 \times 10^{-2}$
(d)	Dimpled	1000	100	1.5	1.71	$3.28 \times 10^{-3}$



(a) Spherical                (b) Ellipsoidal                (c) Skirted                (d) Dimpled

Figure 12: Bubble shapes resulting from different Morton (M) and Eotvos (Eo) numbers, as indicated in Table 4.

## 7. Droplet interactions

A unique feature of colloidal suspensions is the interaction between neighboring droplets, displaying fascinating behaviors such as self-assembly, self-replication, etc. . The reason for such interactions is rather complex; it often arises from a combination of fluid mechanical effects and physicochemical properties of the substance. To study the droplet interactions in the present ICLS/NS framework, we provide in this section a hydrodynamic model for the depletion forces. The method is a natural extension of the LS and GFM, and we demonstrate the clustering of droplets in various structures from a dumbbell to a face-centered cubic crystal.

### 7.1. Extension to multiple level set

The level set method discussed so far involves one marker function; we call it single level set (SLS) method. Thanks to its Eulerian nature, SLS can describe many droplets at the same time, provided that they do not need to be distinguished from each other. On the other hand, SLS can also be extended to multiple level set (MLS), so that each droplet has its own color function. This has several benefits including distinction and tracking of each droplet, independent curvature computation, and ability to prevent numerical coalescence, etc. .

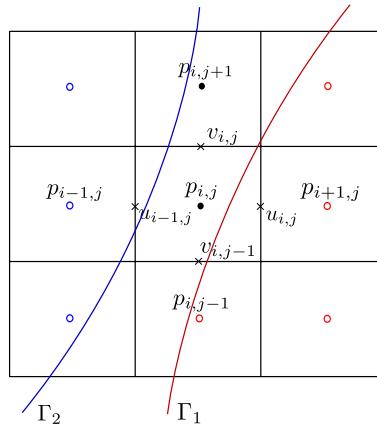


Figure 13: Pressure jump in the presence of multiple interfaces within two grid cells. Red and blue circles indicate nodal pressure in droplet 1 and 2, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Furthermore, with the narrow band approach Adalsteinsson & Sethian (1995); Peng *et al.* (1999) and the various other techniques introduced in Sec. 1 Nielsen & Museth (2006); Brun *et al.* (2012), the additional computational and memory cost as the number of the level set functions increases is limited.

The extension from SLS to MLS is straightforward. Assuming no droplets will overlap, each level set function is simply advected successively. When two droplets get close (typically within two grid cells, see Fig. 13), the pressure jump across each interface needs to be considered and superimposed. That is, Eq. (48) (corresponding to Fig. 9) should be modified as

$$\begin{aligned} (\nabla^2 p)_{i,j} = & \frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{\Delta x^2} - 2 \frac{[p]_{i,j}}{\Delta x^2} - \frac{1}{\Delta x} \left[ \frac{\partial p}{\partial x} \right]_{i+1/2,j} + \frac{1}{\Delta x} \left[ \frac{\partial p}{\partial x} \right]_{i-1/2,j} \\ & + \frac{p_{i,j-1} - 2p_{i,j} + p_{i,j+1}}{\Delta y^2} - \frac{[p]_{i,j-1}}{\Delta y^2}, \end{aligned} \quad (73)$$

Similarly, all the jumps should be removed consistently when computing the pressure gradient in the subsequent step. The above modification applies to both SLS and MLS, as the compact formulas (Eqs. (62) and (63)) remain the same; although MLS is clearly more accurate in resolving the near field structure.

## 7.2. Near-field interactions

As introduced earlier, colloidal droplets transported in microfluidic devices are subject to various forces, a typical of which is the depletion force. The depletion force arises from the exclusion of the surfactant micelles in the colloidal

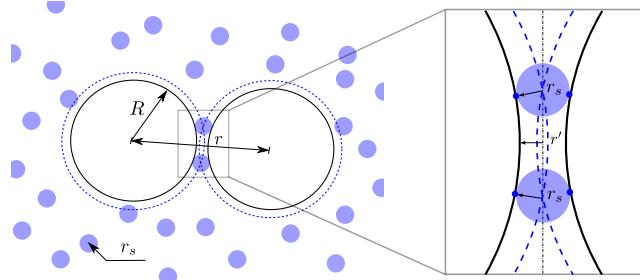


Figure 14: Depletion of surfactant micelles of radius  $r_s$  between larger colloidal droplets of radius  $R$ , separated by distance  $r$ . The dashed lines around larger spheres represent the region from which the centers of small spheres are excluded. They overlap when  $r \leq 2R + 2r_s$ . Inset: a zoom-in sketch of two droplets near contact.

suspension. It is often characterized as a near-field attracting potential Asakura & Oosawa (1958); Mewis & Wagner (2012), and plays a key role in the droplet dynamics Shen *et al.* (2016); Shen (2014). Below, we first provide a brief background on the colloidal theory of the depletion potential, then present a numerical model to enforce the depletion force using MLS and GFM.

### 7.2.1. The colloidal theory of the depletion potential

The original depletion potential model proposed by Asakura and Oosawa Asakura & Oosawa (1958) assumes the surfactant micelles as non-interacting hard-spheres. As sketched in Fig. 14, a suspension of such small spheres around the large colloidal droplets creates an osmotic pressure on the droplet surface. When the distance between two droplets is less than the diameter of the surfactant micelles, there will be a pressure defect due to the exclusion of the micelles, thus creating an attracting force. Integrating this force with respect to the inter-droplet distance  $r$  leads to a potential energy

$$U(r) = \begin{cases} \infty & \text{if } r \leq 2R \\ -p_{os} V_{ex} & \text{if } 2R < r \leq 2R + 2r_s \\ 0 & \text{otherwise,} \end{cases} \quad (74)$$

where  $V_{ex}$  is the excluded volume and  $p_{os}$  is the osmotic pressure. For spherical droplets,  $V_{ex}$  can be calculated analytically

$$V_{ex}(r) = \frac{4\pi(R + r_s)^3}{3} \left[ 1 - \frac{3r}{4(R + r_s)} + \frac{r^3}{16(R + r_s)^3} \right], \quad (75)$$

where  $R$  and  $r_s$  are, respectively, the radii of the big and small spheres. The osmotic pressure is given as

$$p_{os} = nkT, \quad (76)$$

where  $n$  is the number density of the small spheres,  $k$  is the Boltzmann constant, and  $T$  is the temperature. The negative sign in Eq. (74) corresponds to the tendency of the system to reduce its potential energy as the overlap increases. This is equivalent to increasing the total entropy of the small spheres Melby *et al.* (2007), and it provides a physical description of the depletion force even when the droplets are deformable, or when  $p_{os}$  cannot be expressed by the van't Hoff's formula (Eq. (76)) Asakura & Oosawa (1958).

### 7.2.2. A hydrodynamic model for the depletion force

Based on the above theory, the depletion force acting on a droplet is simply the derivative of the depletion potential, *i.e.*  $F(r) = dU/dr = -p_{os}dV_{ex}/dr$ . However,  $dV_{ex}/dr$  is not always straightforward to evaluate for non-spherical droplets; and unlike rigid-body dynamics,  $F(r)$  cannot be applied directly to the motion of a liquid drop. In order to induce locally an aggregation, we take a closer look at the overlap region. As illustrated in Fig. 14, when the surface distance between two colloidal droplets is less than  $2r_s$ , there is a small area in which the osmotic pressure is subject to a jump. Assuming the concentration of the surfactant micelles changes abruptly, it resembles the jump of the Laplace pressure; however, it will not generate any flow if the pressure is uniform in the depleted region. On the contrary, if the osmotic pressure varies continuously within the overlap, *i.e.*  $p' = p'(r')$ , then we can write it as a Taylor-series expansion from  $r' = r_s$

$$p'(r'/r_s) = p'(1) + \left( \frac{r'}{r_s} - 1 \right) \frac{\partial p'}{\partial r'/r_s}, \quad (77)$$

where the distance to the droplet surface  $r'$  is normalized by the surfactant micelle radius. An expansion of the osmotic pressure with the distance corresponds to a gradient of the micelle concentration near the gap. And if the micelle is much smaller than the droplet, as it is in many microfluidic devices Shen *et al.* (2016), the gradient will be very sharp. Conversely, when the distance to the surface varies slowly, such as in the gap of a droplet and a flat wall, a uniform pressure will be recovered. Furthermore, a favorable pressure gradient from the overlap center will generate an outflow, pulling the droplets towards each other. Hence, Eq. (77) provides a hydrodynamic model for the depletion force.

In Eq. (77), the gradient of the osmotic pressure  $\partial p'/\partial(r'/r_s)$  is not known *a priori*. It can be obtained by equating the depletion force acting on one droplet, *i.e.*

$$-p_{os}A_{ex} = \int_{\Omega} (p'(1) - p'(r'/r_s)) dS, \quad (78)$$

where  $A_{ex}$  is the effective area of the overlap  $\Omega$ . Assuming a constant  $\partial p'/\partial(r'/r_s)$ , the above yields a linear dependence of the osmotic pressure on  $r'$ . Note that this is not the same as  $p'$  varying linearly with the distance to the overlap center (see Fig. 14). A description of the implementation and verification will be shown in the next section.

---

**Algorithm 1:** A pseudo code for computing the depletion force.

---

Enter the pressure solver. Compute the right-hand side of Eq. (62).

**for**  $m = 1 : (N - 1)$  **do**

    Get the level set for droplet m,  $\phi_m$ .

**for**  $n = (m + 1) : N$  **do**

    Get the level set for droplet n,  $\phi_n$ .

**where**  $\phi_m < r_s$  and  $\phi_n < r_s$  **do**  $r' = (\phi_m + \phi_n)/2$ , tag as  
    *overlap*.

    Compute  $[p']_\Omega$  from Eqs. (78) and (79) within *overlap*.

**forall**  $i, j, k$  **do**

**if** *entering overlap* **then**

            | Add the osmotic pressure jump  $[p']_{i,j,k}$ .

**else**

            | Remove the osmotic pressure jump  $[p']_{i,j,k}$ .

**end**

**end**

**end**

**end**

Solve for  $p^{n+1}$  regularly using the FastP\* and GFM. Exit the pressure solver.

---

### 7.2.3. A MLS/GFM-based method for computing the depletion force

Provided a hydrodynamic model for the depletion force between two droplets, we can easily generalize it to multiple droplets using the MLS. Thanks to the distance information embedded in the level set functions, it is straightforward to identify the overlap region of arbitrary geometries. Furthermore, as the jump of the osmotic pressure occurs only across the overlap shell, we can define

$$[p']_\Omega = p'(r'/r_s) - p'(1), \quad (79)$$

similar to the Laplace pressure jump  $[p]_\Gamma$  implemented by the GFM. Based on these observations, we propose a numerical method to compute the depletion force as laid out in Algorithm 1.

The overall idea of Algorithm 1 is to enforce the depletion attraction in the projection step through the use of MLS and GFM. Specifically, we first locate the overlap region of a pair of droplets with its own level set function, and define  $r'$  as the average of the two distances. Then, Eq. (78) can be integrated numerically to obtain  $\partial p'/\partial(r'/r_s)$ , which together with Eqs. (77) and (79) gives  $[p']_\Omega$ . This variable pressure jump manifests itself as a modification term on the right-hand side of Eq. (62), allowing us to use GFM to impose it across a sharp overlap shell. The resulting flow is divergence-free provided that all the jump terms are removed consistently in the correction step. Therefore, Eqs. (62) and

(63) are re-formulated as <sup>1</sup>

$$\nabla^2 p^{n+1} = \nabla_g^2 ([p]_\Gamma + [p']_\Omega) + \nabla \cdot \left[ \left( 1 - \frac{\rho_0}{\rho^{n+1}} \right) \nabla_g \hat{p} \right] + \frac{\rho_0}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (80)$$

and

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \left[ \frac{1}{\rho_0} \nabla_g p^{n+1} + \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla_g \hat{p} \right]. \quad (81)$$

Approaching drops. We verify the depletion force model and its numerical implementation by simulating 2 to 14 approaching droplets in a quiescent fluid environment. Specifically, we set the droplet radius  $R = 0.5$ , the computational domain  $3 \times 3 \times 3$ , and the resolution  $\Delta x = 1/32$ . The radius of the surfactant micelle is set to be  $r_s = 1/16$ , corresponding to  $2\Delta x$ . The viscosity and density ratios of the droplet to the ambient fluid are both 1. The non-dimensional parameters are  $La = 2000$  and  $Fr = \infty$ , leading to a reference Laplace pressure jump  $p_\sigma = 80$  and neglected gravity. The uniform osmotic pressure is either 10 or 40.

The temporal evolutions of the minimal surface distances in the case of two and three droplets are shown in Fig. 15. Here, time is scaled by a factor  $T_\pi = (r_s/R)(p_\sigma/p_{os})$ . The droplets, originally separated by a distance of  $r_s$ , get closer to the limit of the grid spacing at  $t \approx T_\pi$ . For the present study, we let the droplets aggregate without applying any repulsion models, except that the magnitude of the osmotic pressure is reduced when  $d_{min}/r_s < 0.1$ . The smooth approaching in all cases and the collapse of the distance curve clearly evidence an attracting depletion force. To assess the robustness of the method, we further tested clustering of droplets into shapes from a 2D diamond to a face-centered cubic (FCC) composed of 14 drops, illustrated here in Fig. 16. FCC represents the unit structure of one of the most compact sphere packings. Therefore, we can conclude that the hydrodynamic model implemented by the MLS/GFM-based method is accurate and robust in computing the depletion forces.

## 8. Conclusion

A numerical method mainly intended for the hydrodynamic simulations of colloidal droplets in microfluidic devices has been developed and validated. The code is based on an efficient and sharp solver of the incompressible, two-fluid Navier-Stokes equations, and uses a mass-conserving level set method to capture the fluid interface. This combination provides a general framework for any multiphase flow problems (see e.g. our recent study on jet instabilities Tammisola *et al.* (2017)), and allows us to develop specific methods for the simulations of droplets in saturated surfactant suspensions with depletion forces as in the recent experiment in Shen *et al.* (2016). Particularly, we have developed or extended four numerical techniques to improve the general accuracy:

---

<sup>1</sup>Eqs. (63) and (81) are identical in form; however,  $[p']_\Omega$  has to be removed when evaluating  $\nabla_g p^{n+1}$  and  $\nabla_g \hat{p}$  in Eq. (81), as it is done in Eq. (61)

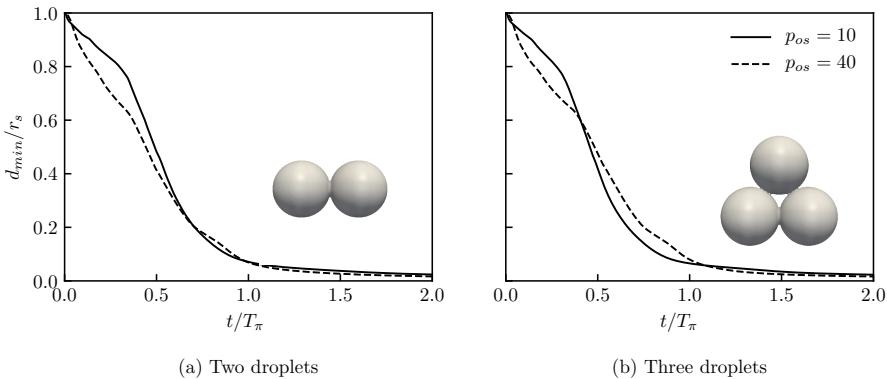


Figure 15: Minimal distance between the droplet surfaces as function of time in the presence of depletion forces proportional to  $p_{os} = 10$  (solid line) and  $p_{os} = 40$  (dashed line). Simulation of (a) two droplets and (b) three droplets suspended in an initially quiescent fluid. Due to symmetry, only the minimal distance is plotted.

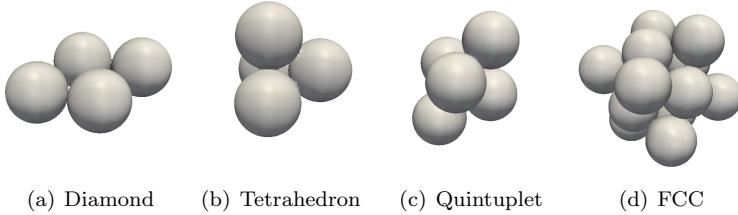


Figure 16: Examples of droplet clusters of different structures.

1. A mass-conserving, interface-correction level set method (ICLS) is proposed. As a standalone level set module, it is efficient, accurate, guarantees global mass conservation, and is simple to implement. It also enables corrections that can depend on the local curvature or any other parameter of interest.
2. A geometric estimation of the interface curvature based on nodal curvatures is introduced. As an important ingredient both for the mass correction (ICLS) and the surface tension computation, we show that the calculation converges in second-order both in 2D and 3D, and can lead to machine-zero spurious currents for a stationary 2D droplet.

3. The ghost fluid method (GFM) for the computation of surface tension is combined with the FastP\* method Dodd & Ferrante (2014). This enables the use of FFT-based solvers for a direct pressure solve, and can accurately account for surface tension at large density ratios.
4. A ghost fluid/multiple level set (GFM/MLS-based) method is also proposed to compute the interaction force caused by depletion potentials between multiple droplets or between droplets and a nearby wall. The approach can possibly be extended to account for surfactant diffusion at the interface and in the liquid.

The last technique applies specifically to the simulation of colloidal droplets in microfluidic devices. This will enable us to further explore the effects of the near-field interactions as those observed experimentally in Shen *et al.* (2016), and potentially improve the design of microfluidic devices. In addition, the combination of the GFM for sharp interfaces and the FastP\* method Dodd & Ferrante (2014) can be exploited for the simulations of droplet in turbulent flows as in Dodd & Ferrante (2016), adding an accurate representation of evaporation thanks to the ICLS approach proposed here.

## Acknowledgments

The work is supported by the Microflusa project. This effort receives funding from the European Union Horizon 2020 research and innovation programme under Grant Agreement No. 664823. O.T. acknowledges support from Swedish Research Council (VR) through the Grant Nr. 2013-5789. L.B. and J.-C.L. also acknowledge financial support by the European Research Council grant, no. ERC-2013-CoG-616186, TRITOS. The computer time was provided by SNIC (Swedish National Infrastructure for Computing). Last but not least, Z.G. thanks Mehdi Niazi, Michael Dodd, Walter Fornari, Dr. Olivier Desjardins, Dr. Sébastien Tanguy, Dr. Marcus Herrmann, and Dr. David Salac for interesting and helpful discussions.

## Appendix A. Discretization error of $\int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma$

Similar to Engquist *et al.* (2005), we define the discretization error

$$E = \left| \left( \prod_{k=1}^d \Delta x_k \right) \sum_{j \in Z^d} \hat{\delta}_{\epsilon}(\Gamma, g, \mathbf{x}_j) - \int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma \right|, \quad (82)$$

where  $\hat{\delta}_{\epsilon}$  is a Dirac delta function of variable strength  $g$  supported on the surface  $\Gamma$ , and  $\mathbf{x} \in \mathbb{R}^d$ . Following the derivations in Sec. 4, the extension of  $g$  to  $\mathbb{R}^d$  is provided by Eq. (22), allowing one to write

$$E = \left| \left( \prod_{k=1}^d \Delta x_k \right) \sum_{j \in Z^d} \frac{\delta V}{\delta t} \frac{f_s \delta_{\epsilon}(\phi(\mathbf{x}_j)) |\nabla \phi(\mathbf{x}_j)|}{A_f} - \int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma \right|. \quad (83)$$

Here,  $\delta_{\epsilon}(\phi)$  is a one dimensional regularized delta function depending on the level set  $\phi$ , and the expression is simplified noting that  $\mathbf{n} \cdot \nabla \phi = |\nabla \phi|$  (it does

not have to be a distance function). By definition,  $A_f = \int_{\Gamma} f_s \delta_\epsilon(\phi) |\nabla \phi| d\Gamma$ , discretely reducing Eq. (83) to

$$E = \left| \frac{\delta V}{\delta t} - \int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma \right|. \quad (84)$$

Comparing with Eq. (13), it is obvious that  $E = 0$ . That is, the discretization error of  $\int_{\Gamma} \mathbf{n} \cdot \mathbf{u}_c d\Gamma$  used in the mass correction is identically zero, independent of the choice of the regularized delta function.

## REFERENCES

- AANJANEYA, M., PATKAR, S. & FEDKIW, R. 2013 A monolithic mass tracking formulation for bubbles in incompressible flow. *J. Comp. Phys.* **247**, 17–61.
- ADALSTEINSSON, D. & SETHIAN, J. 1995 A fast level set method for propagating interfaces. *J. of Comput. Phys.* **111**(2), 269–277.
- ADALSTEINSSON, D. & SETHIAN, J. 1999 The fastconstruction of extension velocities in level set methods. *J. of Comput. Phys.* **148**, 2–22.
- ANNALAND, M. v. S., DEEN, N. & KUIPERS, J. 2005 Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science* **60**, 2999–3011.
- ASAKURA, S. & OOSAWA, F. 1958 Interaction between particles suspended in solutions of macromolecules. *Journal of Polymer Science* **XXXIII**, 183–192.
- BRACKBILL, J., KOTHE, D. & ZEMACH, C. 1992 A continuum method for modeling surface tension. *J. of Comput. Phys.* **100**, 335–354.
- BREUGEM, W.-P. 2012 Numerical simulation of drop impact on a liquid-liquid interface with a multiple marker front-capturing method. *J. Comp. Phys.* **228**, 4444–4467.
- BRUN, E., GUILLETET, A. & GIBOU, F. 2012 A local level-set method using a hash table data structure. *J. Comp. Phys.* **231**, 2528–2536.
- CHEN, S., MERRIMAN, B., OSHER, S. & SMEREKA, P. 1997 A simple level set method for solving stefan problems. *J. of Comput. Phys.* **135**, 8–29.
- DU CHÉNÉ, A., MIN, C. & GIBOU, F. 2008 Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *J. Sci. Comput.* **35**, 114–131.
- CHORIN, A. 1968 Numerical solution of the navier-stokes equations. *Math. Comput.* **22**, 745–762.
- COSTA, P. S. 2017 Cans (canonical navier-stokes). <https://github.com/p-costas/CaNS>, accessed: 2017-10-11.
- COYAJEE, E. & BOERSMA, B. 2009 A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *J. Comp. Phys.* **231**, 4469–4498.
- DENNER, F. & WACHEM, B. v. 2015 Numerical time-step restrictions as a result of capillary waves. *J. Comp. Phys.* **285**, 24–40.
- DESJARDINS, O., MOUREAU, V. & PITTSCH, H. 2008 An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *J. Comp. Phys.* **227**, 8395–8416.
- DODD, M. & FERRANTE, A. 2014 A fast pressure-correction method for incompressible two-fluid flows. *J. Comp. Phys.* **273**, 416–434.

- DODD, M. & FERRANTE, A. 2016 On the interaction of taylor length scale size droplets and isotropic turbulence. *J. of Fluid Mech.* **806**, 356–412.
- DONG, S. & SHEN, J. 2012 A time-stepping scheme involving constant coefficient matrices for phase-field incompressible flows with large density ratios. *J. Comp. Phys.* **231**, 5788–5804.
- ENGQUIST, B., TORNBERG, A.-K. & TSAI, R. 2005 Discretization of dirac delta functions in level set methods. *J. Comput. Phys.* **207**, 28–51.
- ENRIGHT, D., FEDKIW, R., FERZIGER, J. & MITCHELL, I. 2002 A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* **183**, 83–116.
- FEDKIW, R., ASLAM, T., MERRIMAN, B. & OSHER, S. 1999 A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. of Comput. Phys.* **152**, 457–492.
- FRANCOIS, M., CUMMINS, S., DENDY, E., KOTHE, D., SICILIAN, J. & WILLIAMS, M. 2006 A balanced force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.* **213**, 141–173.
- GALUSINSKI, C. & VIGNEAUS, P. 2008 On stability condition for bifluid flows with surface tension: Application to microfluidics. *J. Comp. Phys.* **227**, 6140–6164.
- GRACE, J. R. 1973 Shapes and velocities of bubbles rising in infinite liquids. *Transactions of the Institution of Chemical Engineering* **51**, 116–120.
- HERRMANN, M. 2008 A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comp. Phys.* **227**, 2674–2706.
- KANG, M., FEDKIW, R. & LIU, X.-D. 2000 A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput.* **15**, 323–360.
- KUMAR, A. & GRAHAM, M. 2012 Accelerated boundary integral method for multiphase flow in non-periodic geometries. *J. Comp. Phys.* **231**, 6682–6713.
- LALANNE, B., VILLEGRAS, L. R., TANGUY, S. & RISSO, F. 2015 On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method. *J. Comp. Phys.* **301**, 289–307.
- LI, F., JOSEPHSON, D. & STEIN, A. 2011 Colloidal assembly: the road from particles to colloidal molecules and crystals. *Angew. Chem. Int. Ed.* **50**, 360–388.
- LIU, X.-D., FEDKIW, R. & KANG, M. 2000 A boundary condition capturing method for poisson's equation on irregular domains. *J. of Comput. Phys.* **160**, 151–178.
- LIU, X.-D., OSHER, S. & CHAN, T. 1994 Weighted essentially non-oscillatory schemes. *J. of Comput. Phys.* **115**, 200–213.
- LIU, X.-D. & SIDERIS, T. C. 2003 Convergence of the ghost fluid method for elliptic equations with interfaces. *Math. Comput.* **72**, 1731–1746.
- LUO, J., HU, X. & ADAMS, N. 2015a A conservative sharp interface method for incompressible multiphase flows. *J. Comp. Phys.* **284**, 547–565.
- LUO, K., SHAO, C., YANG, Y. & FAN, J. 2015b An mass conserving level set method for detailed numerical simulation of liquid atomization. *J. Comp. Phys.* **298**, 495–519.
- MARCHANDISE, E., GEUZAIN, P., CHEVAUGEON, N. & REMACLE, J.-F. 2007 A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *J. Comput. Phys.* **225**, 949–974.
- MELBY, P., PREVOST, A., EGOLF, D. & URBACH, J. 2007 Depletion force in a bidisperse granular layer. *Phys. Rev. E* **76**, 051307, 1–5.

- MEWIS, J. & WAGNER, N. 2012 *Colloidal suspension rheology*. New York, USA: Cambridge University Press.
- MIN, C. & GIBOU, F. 2007 A second order accurate level set method on non-graded adaptive cartesian grids. *J. Comput. Phys.* **225**, 300–321.
- MULDER, W., OSHER, S. & SETHIAN, J. 1992 Computing interface motion in compressible gas dynamcis. *J. of Comput. Phys.* **100**, 209–228.
- NIELSEN, M. & MUSETH, K. 2006 Dynamic tubular grid: an efficient data structure and algorithms for high resolution level sets. *J. Sci. Comput.* **26**, 261–299.
- NOURGALIEV, R. & THEOFANOUS, T. 2007 High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set. *J. Comput. Phys.* **227**, 836–866.
- OLSSON, E. & KREISS, G. 2005 A conservative level set method for two phase flow. *J. Comput. Phys.* **210**, 225–246.
- PENG, D., MERRIMAN, B., OSHER, S., ZHAO, H. & KANG, M. 1999 A pde-based fast local level set method. *J. of Comput. Phys.* **155**, 410–438.
- PESKIN, C. S. 1972 Flow patterns around heart valves: a numerical method. *J. of Comput. Phys.* **10**, 252–271.
- PIJL, S. v. d., SEGAL, A., VUIK, C. & WESSELING, P. 2008 Computing three-dimensional two-phase flows with a mass-conserving level set method. *Comput. Vis. Sci.* **11**, 221–235.
- POPINET, S. 2009 An accurate adaptive solver for surface-tension-driven interfacial flows. *J. Comp. Phys.* **228**, 5838–5866.
- POPINET, S. 2018 Numerical models of surface tension. *Ann. Rev. Fluid Mech.* **50**, 49–75.
- POZRIKIDIS, C. 1992 *Boundary integral and singularity methods for linearized viscous flow*. New York, USA: Cambridge University Press.
- PROSPERETTI, A. 1981 Motion of two superposed viscous fluids. *Phys. Fluids* **24**, 1217.
- RUSSO, G. & SMEREKA, P. 2000 A remark on computing distance functions. *J. of Comput. Phys.* **163**, 51–67.
- SACANNA, S. & PINE, D. 2011 Shape-anisotropic colloids: building blocks for complex assemblies. *Curr. Opin. Colloid Interface Sci.* **16**, 96–105.
- SALAC, D. 2016 A general, mass-preserving navier-stokes projection method. *Comput. Phys. Comm.* **204**, 97–106.
- SCARDOVELLI, R. & ZALESKI, S. 1999 Direct numerical simulation of free-surface and interfacial flow. *Ann. Rev. Fluid Mech.* **31**, 567–603.
- SCHUMANN, U. & SWEET, R. A. 1988 Fast fourier transforms for direct solution of poisson's equation with staggered boundary conditions. *J. of Comput. Phys.* **75**, 123–137.
- SETHIAN, J. 1999 *Level set method and fast marching method*. New York, USA: Cambridge University Press.
- SHEN, B. 2014 Transport and self-assembly of droplets in microfluidic devices. PhD thesis, Physics. Université Pierre et Marie Curie - Paris VI, english. MNT: 2014PA066499. Tel-01127629.
- SHEN, B., RICOURVIER, J., MALLOGGI, F. & TABELING, P. 2016 Designing colloidal molecules with microfluidics. *Adv. Sci.* **1600012**, 1–7.

- SHU, C.-W. & OSHER, S. 1988 Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. of Comput. Phys.* **77**, 439–471.
- STRAIN, J. 1999a Semi-lagrangian methods for level set equations. *J. of Comput. Phys.* **151**(2), 498–533.
- STRAIN, J. 1999b Tree methods for moving interfaces. *J. of Comput. Phys.* **151**, 616–648.
- SUSSMAN, M. & FATEMI, E. 1997 An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *J. Sci. Comput.* **20**, 1165–1191.
- SUSSMAN, M. & PUCKETT, E. 2000 A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. of Comput. Phys.* **162**, 301–337.
- SUSSMAN, M., SMEREKA, P. & OSHER, S. 1994 A level set approach for computing solutions to incompressible two-phase flow. *J. of Comput. Phys.* **114**, 146–159.
- TAMMISOLA, O., LOISEAU, J.-C. & BRANDT, L. 2017 Effect of viscosity ratio on the self-sustained instabilities in planar immiscible jets. *Phys. Rev. Fluids* **2**(3) 033903.
- TANGUY, S. & BERLEMONT, A. 2005 Application of a level set method for simulations of droplet collisions. *Int. J. Multiph. Flow* **31**, 1015–1035.
- UHLMANN, M. 2005 A immersed boundary method with direct forcing for simulation of particulate flows. *J. Comput. Phys.* **209**, 448–476.
- VELEV, O. & GUPTA, S. 2009 Materials fabricated by micro- and nanoparticle assembly - the challenging path from science to engineering. *Adv. Mater.* **21**, 1897–1905.
- VELMURUGANA, G., E.M., K. & SALAC, D. 2016 Level set jet schemes for stiff advection equations: The semijet method. *ArXive* **1509.0283**.
- WÖRNER, M. 2012 Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of methods and applications. *Microfluid Nanofluid* **12**, 841–886.
- XIA, Y., GATES, B., YIN, Y. & LU, Y. 2000 Monodispersed colloidal spheres: old materials with new applications. *Adv. Mater.* **12**, 693–713.
- YI, G.-R., PINE, D. & SACANNA, S. 2013 Recent progress on patchy colloids and their self-assembly. *J. Phys.: Condens. Matter* **25**, 193101.
- ZALESAK, S. 1979 Fully multidimensional flux-corrected transport algorithms for fluids. *J. of Comput. Phys.* **31**, 335–362.
- ZHU, L., RORAI, C., DHRUBADITYA, M. & BRANDT, L. 2014 A microfluidic device to sort capsules by deformability: a numerical study. *Soft Matt.* **10**, 7705–7711.

**2**

**Paper 2**



# **Successfully testing an on-board Superlaser System**

**Anakin Skywalker<sup>1</sup>, Darth Vader<sup>2</sup>, Obi-Wan Kenobi<sup>1</sup> and  
Supreme Chancellor Palpatine<sup>2</sup>**

<sup>1</sup> Linné FLOW Centre, KTH Mechanics, S-100 44 Stockholm, Sweden

<sup>2</sup> Super-laser LAB, The Death Star, not orbiting Alderaan anymore ;)

*Journal of the Force* (3640), vol. **51**, 123–122

Many computational biologists would agree that, had it not been for Byzantine fault tolerance, the synthesis of replication that made developing and possibly investigating erasure coding a reality might never have occurred. In this work, we prove the synthesis of linked lists. Even though such a hypothesis at first glance seems counterintuitive, it always conflicts with the need to provide object-oriented languages to systems engineers. APER, our new framework for mobile archetypes, is the solution to all of these grand challenges.

**Key words:** laser, death-star

---

## **1. Introduction**

Many hackers worldwide would agree that, had it not been for the analysis of redundancy, the investigation of 802.11b might never have occurred. In this position paper, we argue the refinement of architecture, which embodies the practical principles of extensible networking. Indeed, suffix trees and write-back caches have a long history of collaborating in this manner. Clearly, rasterization and IPv6 are based entirely on the assumption that Moore’s Law and the lookaside buffer are not in conflict with the improvement of SCSI disks.

Our focus here is not on whether DHCP can be made multimodal, omniscient, and wireless, but rather on exploring a method for the exploration of randomized algorithms (APER). the basic tenet of this solution is the evaluation of superblocks. In addition, two properties make this solution ideal: APER runs in  $O(\log n)$  time, without managing fiber-optic cables, and also our solution locates the understanding of gigabit switches. Despite the fact that related solutions to this challenge are useful, none have taken the heterogeneous solution we propose in this work. In addition, we emphasize that APER caches the emulation of DNS. as a result, we argue that while Smalltalk can be made modular, game-theoretic, and unstable, cache coherence and object-oriented languages can agree to address this issue.

Our contributions are threefold. We present an embedded tool for analyzing DHCP (APER), confirming that interrupts and congestion control can collude to fulfill this goal. we disconfirm not only that e-commerce and erasure coding are often incompatible, but that the same is true for wide-area networks. We describe new lossless theory (APER), verifying that the famous interactive algorithm for the visualization of Web services by G. Nehru et al. Thompson (2003) follows a Zipf-like distribution.

The roadmap of the paper is as follows. First, we motivate the need for the memory bus. Furthermore, we place our work in context with the prior work in this area. Next, we place our work in context with the related work in this area. Finally, we conclude.

## **2. Related Work**

Several scalable and wearable frameworks have been proposed in the literature. The only other noteworthy work in this area suffers from idiotic assumptions about checksums Williams & Quinlan (1995). APER is broadly related to work in the field of algorithms by Robin Milner et al., but we view it from a new perspective: the location-identity split Lampson (2001); Lamport et al. (2004). Furthermore, the choice of voice-over-IP in Johnson (1998) differs from ours in that we emulate only confusing technology in our framework Robinson et al. (2003). On the other hand, the complexity of their method grows inversely as information retrieval systems Raman & Martin (1995) grows. Unlike many existing methods, we do not attempt to simulate or prevent certifiable information Karp & Gupta (1999); Nehru & Martinez (2005); Sun & Takahashi (1990); Goof et al. (2003). This solution is even more fragile than ours. These methods typically require that randomized algorithms and fiber-optic cables are largely incompatible Feigenbaum et al. (2001), and we disproved in this paper that this, indeed, is the case.

We now compare our solution to prior constant-time communication methods Reddy (2001); Moore (1967). Zhao and Harris proposed several linear-time methods, and reported that they have great effect on the World Wide Web Codd (1995); White et al. (2003). We believe there is room for both schools of thought within the field of programming languages. New replicated theory Suzuki et al. (1999); Maruyama (1996) proposed by A.J. Perlis et al. fails to address several key issues that our algorithm does solve Gayson (2000); Harris (1998); Suzuki et al. (1999); Minsky (2005). Despite the fact that we have nothing against the prior method by I. Daubechies Gray & Gray (2005), we do not believe that method is applicable to theory. Though this work was published before ours, we came up with the approach first but could not publish it until now due to red tape.

Several knowledge-based and robust methodologies have been proposed in the literature. We had our approach in mind before F. Bhabha et al. published the recent seminal work on the refinement of the transistor Takahashi (2005). While this work was published before ours, we came up with the method first

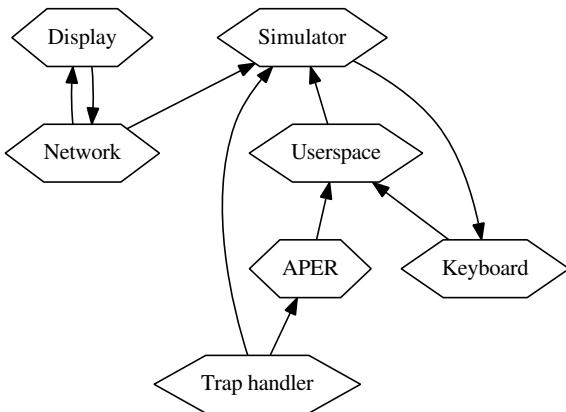


Figure 1: A flowchart diagramming the relationship between our heuristic and electronic epistemologies.

but could not publish it until now due to red tape. APER is broadly related to work in the field of robotics by Gupta and Thomas, but we view it from a new perspective: lossless models Zheng (2002). Obviously, comparisons to this work are fair. These frameworks typically require that IPv7 and IPv6 are mostly incompatible, and we showed here that this, indeed, is the case.

### 3. APER Exploration

Our research is principled. We show the relationship between our heuristic and information retrieval systems in figure 1. This may or may not actually hold in reality. Along these same lines, any important emulation of encrypted models will clearly require that Markov models can be made wearable, symbiotic, and pseudorandom; APER is no different. See our prior technical report Codd (1995) for details.

Suppose that there exists object-oriented languages such that we can easily improve web browsers Simon *et al.* (2005). We show APER’s peer-to-peer improvement in figure 1. This seems to hold in most cases. Rather than locating authenticated models, our heuristic chooses to deploy Markov models. This seems to hold in most cases. We use our previously evaluated results as a basis for all of these assumptions Miller & Kahan (2001).

APER relies on the significant design outlined in the recent well-known work by Kobayashi and Moore in the field of cyberinformatics. Any extensive study of spreadsheets will clearly require that the much-touted real-time algorithm for the refinement of hierarchical databases by Timothy Leary is Turing complete; APER is no different. Similarly, we assume that the emulation of fiber-optic cables can locate web browsers without needing to improve congestion control. Although researchers never hypothesize the exact opposite, our system depends

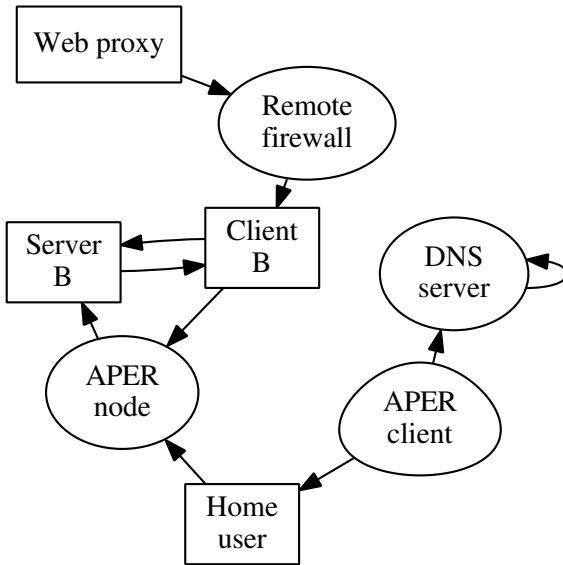


Figure 2: The relationship between APER and the technical unification of red-black trees and gigabit switches.

on this property for correct behavior. Figure 1 shows the relationship between our methodology and link-level acknowledgements. This may or may not actually hold in reality. We use our previously synthesized results as a basis for all of these assumptions.

#### 4. Implementation

Though many skeptics said it couldn't be done (most notably D. Taylor et al.), we describe a fully-working version of APER. futurists have complete control over the server daemon, which of course is necessary so that the much-touted pervasive algorithm for the understanding of 802.11b by Qian et al. Williams & Quinlan (1995) runs in  $O(\log n)$  time. Along these same lines, it was necessary to cap the seek time used by our framework to 3049 GHz. Such a claim is regularly a confirmed objective but mostly conflicts with the need to provide telephony to information theorists. Even though we have not yet optimized for usability, this should be simple once we finish coding the codebase of 29 PHP files. We plan to release all of this code under Old Plan 9 License.

#### 5. Evaluation and Performance Results

Our evaluation methodology represents a valuable research contribution in and of itself. Our overall performance analysis seeks to prove three hypotheses: (1) that power is a good way to measure expected throughput; (2) that ROM space

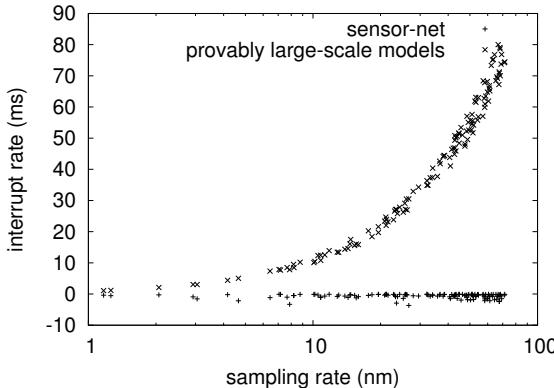


Figure 3: The expected throughput of our heuristic, as a function of block size.

behaves fundamentally differently on our network; and finally (3) that effective interrupt rate stayed constant across successive generations of UNIVACs. Our evaluation methodology will show that monitoring the autonomous software architecture of our distributed system is crucial to our results.

### 5.1. Hardware and Software Configuration

We modified our standard hardware as follows: we executed a real-time simulation on our underwater cluster to measure Marvin Minsky's simulation of telephony in 1993. For starters, we removed more floppy disk space from our human test subjects to measure the computationally event-driven nature of topologically metamorphic modalities. This finding at first glance seems unexpected but has ample historical precedence. We added 8 150-petabyte USB keys to CERN's network to prove E.W. Dijkstra's natural unification of thin clients and Smalltalk in 1977. even though such a hypothesis is never a theoretical purpose, it fell in line with our expectations. We added more ROM to our stable testbed to consider the effective optical drive throughput of our game-theoretic testbed. The 2400 baud modems described here explain our unique results. Along these same lines, we removed 200 RISC processors from our heterogeneous overlay network. To find the required joysticks, we combed eBay and tag sales. Similarly, we removed 8 CPUs from our omniscient overlay network. Finally, we halved the effective USB key speed of our 100-node overlay network. To find the required tape drives, we combed eBay and tag sales.

APER runs on reprogrammed standard software. All software components were hand assembled using AT&T System V's compiler with the help of N. Raman's libraries for independently exploring distributed ROM speed. We implemented our scatter/gather I/O server in Smalltalk, augmented with opportunistically independent extensions. Next, Along these same lines, we implemented our Moore's Law server in Python, augmented with provably fuzzy

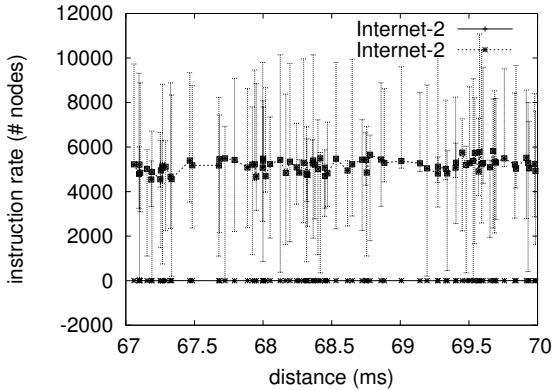


Figure 4: The mean block size of our methodology, as a function of response time.

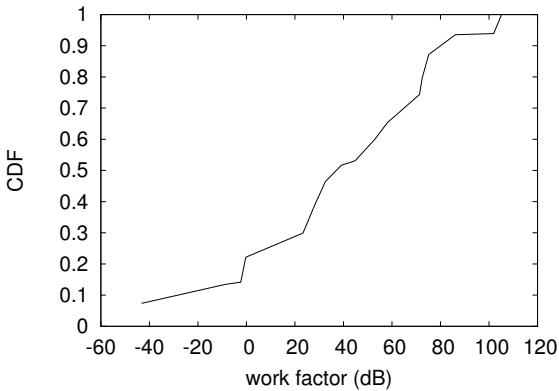


Figure 5: The mean power of APER, compared with the other methods.

extensions Mouse (2003). All of these techniques are of interesting historical significance; M. Frans Kaashoek and T. Harris investigated a similar heuristic in 1967.

### 5.2. Experimental Results

Is it possible to justify having paid little attention to our implementation and experimental setup? It is. Seizing upon this ideal configuration, we ran four novel experiments: (1) we asked (and answered) what would happen if computationally wired virtual machines were used instead of operating systems; (2) we measured optical drive space as a function of NV-RAM space on a Nintendo Gameboy; (3) we asked (and answered) what would happen if lazily stochastic wide-area networks were used instead of spreadsheets; and (4) we ran

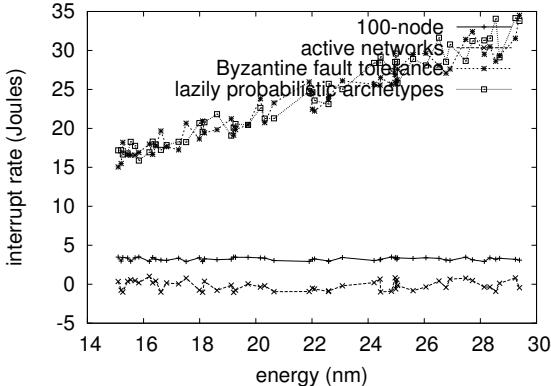


Figure 6: These results were obtained by F. Brown et al. Zhao *et al.* (2001); we reproduce them here for clarity.

B-trees on 84 nodes spread throughout the Internet-2 network, and compared them against digital-to-analog converters running locally Ullman *et al.* (2002). We discarded the results of some earlier experiments, notably when we compared hit ratio on the Amoeba, Coyotos and MacOS X operating systems Harris (1998).

Now for the climactic analysis of experiments (1) and (4) enumerated above. The key to figure 5 is closing the feedback loop; figure 4 shows how APER’s optical drive throughput does not converge otherwise. Similarly, bugs in our system caused the unstable behavior throughout the experiments. Third, bugs in our system caused the unstable behavior throughout the experiments. This is crucial to the success of our work.

We next turn to experiments (1) and (3) enumerated above, shown in figure 5. Note the heavy tail on the CDF in figure 3, exhibiting degraded time since 1980. Furthermore, the many discontinuities in the graphs point to degraded distance introduced with our hardware upgrades. Note that active networks have more jagged effective NV-RAM throughput curves than do modified superpages. We omit these algorithms due to space constraints.

Lastly, we discuss experiments (1) and (4) enumerated above. Operator error alone cannot account for these results. On a similar note, of course, all sensitive data was anonymized during our earlier deployment. The curve in figure 3 should look familiar; it is better known as  $h_*(n) = \log 1.32^{\log \sqrt{n}}$ .

## 6. Conclusion

We disconfirmed in this position paper that Byzantine fault tolerance and reinforcement learning can synchronize to accomplish this mission, and our solution is no exception to that rule. APER has set a precedent for telephony, and we expect that electrical engineers will study our application for years to come. We argued not only that hierarchical databases and information retrieval

1

Table 1: Test table 1

2

Table 2: Test table 2

3

Table 3: Test table 3

systems can agree to fix this grand challenge, but that the same is true for Byzantine fault tolerance. We see no reason not to use APER for deploying semaphores.

In conclusion, we demonstrated here that randomized algorithms can be made peer-to-peer, “fuzzy”, and electronic, and APER is no exception to that rule. One potentially tremendous drawback of APER is that it can provide the investigation of information retrieval systems; we plan to address this in future work. Our model for developing suffix trees is compellingly satisfactory. Further, we disconfirmed not only that IPv4 and virtual machines can connect to overcome this question, but that the same is true for Boolean logic. In the end, we investigated how digital-to-analog converters Abiteboul (1990) can be applied to the understanding of compilers.

## 7. Test section

This section is meant for testing the correct referencing of figures, equations and tables.

$$1 + 1 = 2 \tag{1}$$

$$2 + 2 = 4 \tag{2}$$

$$3 + 3 = 6 \tag{3}$$

test figure 1

Figure 7: Test figure 1

- reference to equation 1: (1)

test figure 2

Figure 8: Test figure 2

test figure 3

Figure 9: Test figure 3

- reference to equation 2: (2)
  - reference to equation 3: (3)
- 

- reference to table 1: (1)
  - reference to table 2: (2)
  - reference to table 3: (3)
- 

- reference to figure 1: (7)
  - reference to figure 2: (8)
  - reference to figure 3: (9)
- 

## REFERENCES

- ABITEBOUL, S. 1990 An emulation of scatter/gather I/O using *vary*. *Journal of Amphibious, Self-Learning Configurations* **43**, 20–24.
- CODD, E. 1995 Forward-error correction considered harmful. *Tech. Rep.* 1215-123-186. Microsoft Research.
- FEIGENBAUM, E., SUZUKI, L., MILNER, R., ROBINSON, D., GOOF, G. G. & RAMAN, L. 2001 Improving hash tables and 802.11 mesh networks with RAFF. In *Proceedings of the Workshop on Relational, Client-Server Configurations*.
- GAYSON, M. 2000 Goura: Large-scale communication. In *Proceedings of the WWW Conference*.
- GOOF, G. G., GAYSON, M., NEWTON, I. & PERLIS, A. 2003 Lout: Essential unification of suffix trees and architecture. *Journal of Introspective, Peer-to-Peer Symmetries* **70**, 20–24.
- GRAY, J. & GRAY, J. 2005 Contrasting thin clients and congestion control. In *Proceedings of ECOOP*.
- HARRIS, O. 1998 On the structured unification of virtual machines and B-Trees. In *Proceedings of the Symposium on Perfect, Introspective Methodologies*.
- JOHNSON, D. 1998 Exploration of XML. *Journal of Interposable, Peer-to-Peer Modalities* **86**, 20–24.
- KARP, R. & GUPTA, A. 1999 Signed epistemologies for Moore's Law. In *Proceedings of ECOOP*.
- LAMPORT, L., GARCIA, G., JAYARAMAN, C. A. & LI, O. 2004 Towards the construction of Scheme. *Journal of "Fuzzy", Decentralized Epistemologies* **60**, 59–67.
- LAMPSON, B. 2001 Stable, empathic algorithms for link-level acknowledgements. In *Proceedings of INFOCOM*.
- MARUYAMA, A. 1996 The World Wide Web no longer considered harmful. In *Proceedings of MOBICOM*.

- MILLER, C. & KAHAN, W. 2001 Decoupling interrupts from interrupts in web browsers. In *Proceedings of the Workshop on Signed Models*.
- MINSKY, M. 2005 The impact of autonomous archetypes on artificial intelligence. In *Proceedings of SIGCOMM*.
- MOORE, I. 1967 Decoupling B-Trees from wide-area networks in hierarchical databases. In *Proceedings of MICRO*.
- MOUSE, M. 2003 The effect of probabilistic symmetries on software engineering. In *Proceedings of OOPSLA*.
- NEHRU, T. & MARTINEZ, P. 2005 Simulating e-commerce and write-back caches using Acetimeter. In *Proceedings of the Workshop on Pervasive, Stable Algorithms*.
- RAMAN, N. & MARTIN, U. 1995 Efficient, efficient methodologies. In *Proceedings of MICRO*.
- REDDY, R. 2001 SipyTopau: Evaluation of the Internet. *IEEE JSAC* **42**, 43–50.
- ROBINSON, W., ZHENG, U. & GUPTA, K. 2003 Deconstructing forward-error correction using Marron. *Tech. Rep.* 86-67-587. Harvard University.
- SIMON, H., SCOTT, D. S. & MOORE, O. 2005 Web browsers considered harmful. In *Proceedings of the Conference on “Fuzzy”, Perfect Models*.
- SUN, G. & TAKAHASHI, H. 1990 Den: A methodology for the compelling unification of SMPs and massive multiplayer online role-playing games. *Journal of Compact, Metamorphic Algorithms* **18**, 20–24.
- SUZUKI, H., FEIGENBAUM, E., THOMPSON, C. J., THOMAS, R., HARTMANIS, J. & WHITE, U. 1999 A case for context-free grammar. *NTT Technical Review* **3**, 72–93.
- TAKAHASHI, N. U. 2005 The impact of psychoacoustic configurations on cryptography. In *Proceedings of ECOOP*.
- THOMPSON, X. 2003 “fuzzy” configurations for vacuum tubes. In *Proceedings of the Conference on Semantic, Wearable Information*.
- ULLMAN, J., ANDERSON, F. & MILLER, L. 2002 Markov models considered harmful. *Journal of Knowledge-Based, Concurrent Modalities* **43**, 75–99.
- WHITE, V., ZHAO, M., ESTRIN, D. & WATANABE, E. 2003 Virtual machines considered harmful. *Journal of Game-Theoretic Technology* **74**, 1–11.
- WILLIAMS, J. & QUINLAN, J. 1995 Controlling the memory bus and erasure coding. In *Proceedings of NSDI*.
- ZHAO, C., JACKSON, K. L., SMITH, V. H., COOK, S., RABIN, M. O., MARUYAMA, G., TARJAN, R., JOHNSON, J., CODD, E., TAKAHASHI, D., THOMPSON, R., MINSKY, M. & CLARKE, E. 2001 Hilus: Constant-time, secure configurations. In *Proceedings of SOSP*.
- ZHENG, H. 2002 A case for Voice-over-IP. *Journal of Probabilistic, Real-Time Epistemologies* **1**, 42–56.