

Swalbe.jl: A lattice Boltzmann solver for thin film hydrodynamics

Stefan Zitz^{*1, 2}, Manuel Zellhöfer¹, Andrea Scagliarini^{3, 4}, and Jens Harting^{1, 2}

¹ Helmholtz Institute Erlangen-Nürnberg for Renewable Energy, Erlangen, Germany ² Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany ³ Consiglio Nazionale delle Ricerche (CNR), Roma, Italy ⁴ INFN, sezione Roma "Tor Vergata," Roma, Italy

DOI: [10.21105/joss.03728](https://doi.org/10.21105/joss.03728)

Software

- Review [↗](#)
- Repository [↗](#)
- Archive [↗](#)

Editor: [David P. Sanders](#) [↗](#)

Submitted: 12 September 2021¹³

Published: 28 November 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Small amounts of liquid deposited on a substrate are an everyday phenomenon. From a theoretical point of view this represents a modelling challenge, due to the multiple scales involved: from the molecular interactions among the three phases (solid substrate, liquid film and surrounding gas) to the hydrodynamic flows. An efficient way to deal with this problem is via the thin-film equation:

$$\partial_t h = \nabla \cdot (M(h) \nabla p), \quad (1)$$

where h is the film thickness, $M(h)$ is a thickness dependent mobility and p is the pressure inside the film. Solving the thin film equation directly is however a difficult task as it is a fourth order degenerate PDE ([Becker et al., 2003](#)). `Swalbe.jl` approaches the problem from a different angle. Instead of directly solving the thin film equation we use a novel method based on a class lattice Boltzmann models ([Krüger et al., 2016](#)), originally developed to simulate shallow water flows ([Salmon, 1999](#)). This allows us to benefit from the simplicity of the lattice Boltzmann algorithm which makes it straightforward to parallelize the code and run it on accelerator devices. Choosing appropriate forces it is possible to simulate complex problems. Among them is the dewetting of a patterned substrates as shown in Fig. 1. It is as well possible to simulate low contact angle droplets out of equilibrium to probe relaxation experiments, e.g. the Cox-Voinov or Tanner's law ([Bonn et al., 2009](#)). Due to a disjoining pressure model for the three phase contact line droplets can not only relax towards their equilibrium they can slide as well ([Zitz et al., 2019](#)). All of this can be coupled with thermal fluctuations to study the stochastic thin film equation ([Shah et al., 2019](#)).

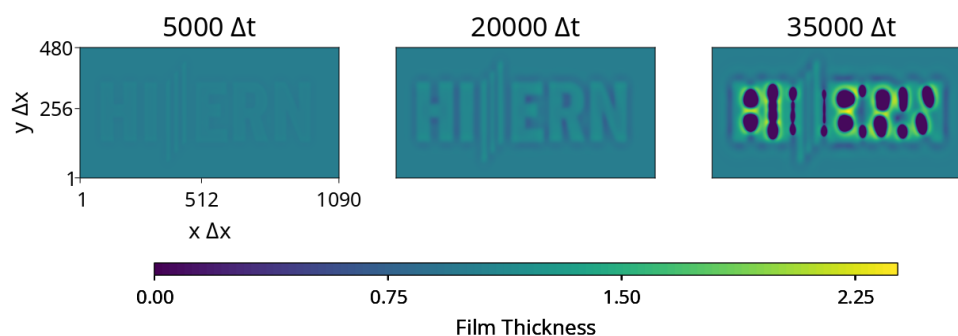


Figure 1: Dewetting simulation on a patterned substrate, letters have a higher wettability than the rest of the substrate.

*corresponding author

28 Statement of need

29 Swalbe is written in Julia ([Bezanson et al., 2017](#)) and developed for a script your exp
30 eriment workflow. For that reason an experiment is composed of three steps. First, the
31 initial problem is defined by setting the system size and other input parameters, stored in
32 a custom type. Followed by the lattice Boltzmann time loop where different force terms
33 allow for different dynamics. To note here however is that some forces are mandatory for
34 every experiment. This is on the one hand the friction with the substrate, the slip that helps
35 regularizing the contact line and on the other hand the capillary- or filmpressure that accounts
36 for the correct wetting behavior.

37 After the time loop has ended an IO step can be included to store the data in files or to
38 generate a plot. The package is written in pure Julia, therefore it can be easily coupled with
39 other packages from the Julia ecosystem such as Plots.jl ([Brelhoff, 2021](#)) to visualize data
40 and JLD2.jl ([Kornblith et al., 2017](#)) to store data in HDF5 format. Of course one future
41 development goal is to interact with SciML environment to pair modelling with ML.

42 Swalbe.jl is designed to approach two problems, first being the modelling itself and second
43 the applicability to large system sizes. Ideas can be implemented quickly and tested with
44 a two dimensional system, which is discretized in a single horizontal direction and offers a
45 second computed dimension for the thickness. The hardware requirements to run these two
46 dimensional simulations are comparably low and depending on the number of lattice Boltzmann
47 iterations ranging from seconds to at most an hour on a single Core of a modern CPU. After
48 testing it is possible to scale up and simulate the same or other problems in three dimensions
49 with a slightly more complex discretization. Keeping the simulation time low is archived by
50 using a Nvidia GPU. Most functions are written in a generic style and can be executed both
51 on a CPU or GPU. For the GPU usage the high-level API of CUDA.jl ([Besard et al., 2018](#),
52 [2019](#)) is used, mostly CuArrays.

53 An older version of the numerical model (written in C++) has been tested and used for
54 thin film simulations in previous publications ([Zitz et al., 2019, 2021](#)). While there is a
55 small performance decrease when moving from C++ with OpenACC to Julia, the benefits of
56 usability, straightforward documentation and automated testing outweighs this issue. There
57 are many thin film problems the authors will investigate in the future with Swalbe.jl. Among
58 those are switchable substrate and their influence on a dewetting thin film, or the influence
59 of thermal fluctuations on the coalescence of droplets.

60 State of the field

61 In the context of computational fluid dynamics low Reynolds number flows and especially thin
62 film flows are a comparably small subsection. Therefore numerical tools that deal exclusively
63 with the thin film problem are sparse. Two packages for simulations of thin film hydrodynamics
64 are **ThinViscoelasticFilms** and **stochastic_thin_films** ([Lin et al., 2019](#); [Nesic, 2017](#)). The
65 core components are written in Fortran and at least the later package can be used according
66 to BSD-2 license. Documentation however is only available through code comments and a
67 short readme, leaving the user little guidance.

68 That said, the thin film equation is a fourth-order parabolic equation and can of course be
69 solved with appropriate numerical schemes. Some of these schemes can be found in the refs.
70 ([Becker et al., 2003](#); [Diez et al., 2000](#); [Engelnkemper et al., 2016](#); [Peschka et al., 2019](#)).
71 Upon contacting the authors it should be possible to have access to a working version of the
72 described approach.

73 Wilczek et al. used **DUNE** to study the dynamics of an ensemble of sliding drops in ref.
74 ([Wilczek et al., 2017](#)). DUNE is a software suite written in C++ that solves partial differential

equations with a grid based approach (Sander, 2020). Therefore DUNE is not limited to the problem of thin film flows, interested readers may visit the project's home page.

Another open source package with similar functionality that is used to solve thin film problems is **oomph-lib**. oomph-lib uses both *Fortran* as well *C* components to solve differential equations. The library's emphasis however are fluid dynamic problems as can be seen from refs. (Heil et al., 2015; Pihler-Puzović et al., 2013, 2015). Of course, similar to DUNE, its capabilities are not limited to thin film problems.

Given the nature of the thin film problem one can, as well, use classical Navier-Stokes solvers with appropriate initial and boundary conditions. What comes to mind here is for example **OpenFOAM** a widely used open source CFD software with an active community. Another example utilizing a Navier-Stokes solver would be the **basilisk** software library, which is written in C and is the successor of **GERRIS**.

Lattice Boltzmann solvers offer another category to approximate the Navier-Stokes equation. Starting point of this method is not the Navier-Stokes equation but the Boltzmann equation. Using the Chapman-Enskog expansion (Chapman & Cowling, 1939; Enskog, 1917), it can be shown that the resulting system of equations recovers to the Navier-Stokes equation. The method is straightforward to implement and several small to large projects can be found with OSI-approved license. To name just a few examples: **waLBerla**, **openLB** or some smaller project **STLBM**.

Proprietary software, e.g. **COMSOL** can as well be used to simulate thin film dynamics. Wedershoven et al. used COMSOL to study the rupture of a thin film due to laser irradiation (Wedershoven et al., 2014). Berendsen et al. from the same group simulated the dynamics an impinging air jet has on a thin film using COMSOL (Berendsen et al., 2012).

With the exclusion of **ThinViscoelasticFilms** every above mentioned project has a much wider purpose than *just* solving the thin film equation. However due to the generality of most libraries it can become quite complex to set up a simulation for a thin film problem. Especially concerning the Navier-Stokes solvers one uses a *sledge hammer to crack a nut*.

Use Case

An interesting problem in the domain of thin liquid films is the coalescence of sessile droplets, see references (Aarts et al., 2005; Eddi et al., 2013; Eggers et al., 1999; Hernández-Sánchez et al., 2012). The underlying idea is that two droplets placed on a hydrophilic substrate in close contact to each other will coalesce into a single droplet to minimize their surface area and therefore energy. The dynamics of this process can be explained using a self-similarity solution of the thin film equation. In fact, that the bridge height, the point that connects the two droplets, has to grow with a power law. We now show how to perform that simulation with the help of the **Swalbe.jl** package. The goal will be to observe a growth of the bridge height as

$$h_0(t) = kt^\alpha, \quad (2)$$

where the exponent α should be $2/3$ (Pawar et al., 2019; Sui et al., 2013). This experiment has been performed using the *Pluto* notebook **Drop_coal.jl**.

Just to outline the most important steps towards the simulation:

1. Define the initial conditions of the simulation: **Swalbe.two_droplets**
2. Define a function that performs the experiment: **run_drop_coal()**
3. Run the experiment with varying parameters:

```
for in enumerate(s)
    # System parameter, =50 can still considered small to medium slippage (
```

```
sys = Swalbe.SysConst_1D(L=1024, Tmax=2000000, =50.0, = [2])
# The experiment
data_merge[:, :, [1]] = run_drop_coal(sys, r=sphere_rad, r=sphere_rad)
end
```

118 4. Display the results:

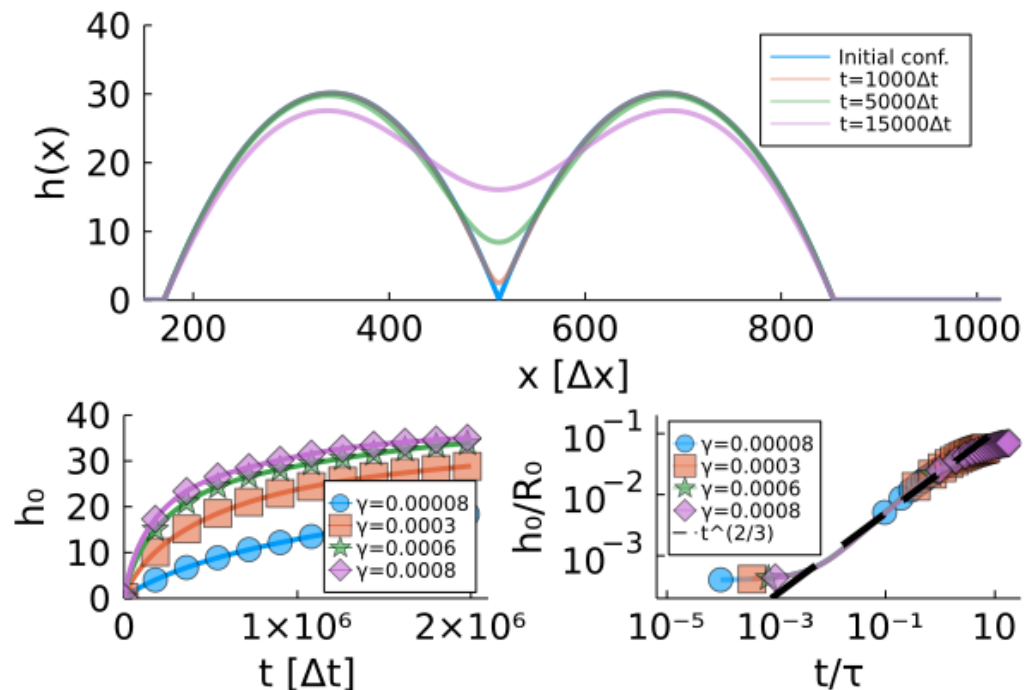


Figure 2: Coalescence of sessile droplets on a partially wetting substrate. The upper panel shows the time evolution for a single experiment (lowest surface tension). In the lower left panel we plot the evolution of the bridge height for the four different surface tensions. To the right we normalize the data with characteristic quantities and show that the bridge height grows as $\propto t^{2/3}$.

119 Acknowledgements

120 The authors acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG)
121 within the priority program SPP2171 “Dynamic Wetting of Flexible, Adaptive, and Switchable
122 Substrates”, within project HA-4382/11.

123 References

- 124 Aarts, D. G. A. L., Lekkerkerker, H. N. W., Guo, H., Wegdam, G. H., & Bonn, D. (2005).
125 Hydrodynamics of droplet coalescence. *Phys. Rev. Lett.*, 95, 164503. <https://doi.org/10.1103/PhysRevLett.95.164503>
126
127 Becker, J., Grün, G., Seemann, R., Mantz, H., Jacobs, K., Mecke, K. R., & Blossey, R. (2003).
128 Complex dewetting scenarios captured by thin-film models. *Nature Materials*, 2(1), 59.
129 Berendsen, C. W. J., Zeegers, J. C. H., Kruis, G. C. F. L., Riepen, M., & Darhuber, A. A.
130 (2012). Rupture of thin liquid films induced by impinging air-jets. *Langmuir*, 28(26),
131 9977–9985. <https://doi.org/10.1021/la301353f>

- 132 Besard, T., Churavy, V., Edelman, A., & De Sutter, B. (2019). Rapid software prototyping for
133 heterogeneous and distributed platforms. *Advances in Engineering Software*, 132, 29–46.
- 134 Besard, T., Foket, C., & De Sutter, B. (2018). Effective extensible programming: Unleashing
135 Julia on GPUs. *IEEE Transactions on Parallel and Distributed Systems*. <https://doi.org/10.1109/TPDS.2018.2872064>
136
- 137 Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to
138 numerical computing. *SIAM Review*, 59(1), 65–98. <https://doi.org/10.1137/141000671>
- 139 Bonn, D., Eggers, J., Indekeu, J., Meunier, J., & Rolley, E. (2009). Wetting and spreading.
140 *Rev. Mod. Phys.*, 81, 739–805. <https://doi.org/10.1103/RevModPhys.81.739>
- 141 Breloff, T. (2021). *Plots.jl*. <https://doi.org/10.5281/zenodo.4907285>
- 142 Chapman, S., & Cowling, T. G. (1939). *The mathematical theory of non-uniform gases*.
143 Cambridge University Press.
- 144 Diez, J. A., Kondic, L., & Bertozzi, A. (2000). Global models for moving contact lines. *Phys.*
145 *Rev. E*, 63, 011208. <https://doi.org/10.1103/PhysRevE.63.011208>
- 146 Eddi, A., Winkels, K. G., & Snoeijer, J. H. (2013). Influence of droplet geometry on the
147 coalescence of low viscosity drops. *Phys. Rev. Lett.*, 111, 144502. <https://doi.org/10.1103/PhysRevLett.111.144502>
148
- 149 Eggers, J., Lister, J. R., & Stone, H. A. (1999). Coalescence of liquid drops. *Journal of Fluid*
150 *Mechanics*, 401, 293–310.
- 151 Engelinkemper, S., Wilczek, M., Gurevich, S. V., & Thiele, U. (2016). Morphological tran-
152 sitions of sliding drops: Dynamics and bifurcations. *Phys. Rev. Fluids*, 1, 073901.
153 <https://doi.org/10.1103/PhysRevFluids.1.073901>
- 154 Enskog, D. (1917). *Kinetische theorie der vorgänge in mässig verdünnten gasen*. PhD Thesis.
- 155 Heil, M., Hazel, A. L., Duprat, C., & Stone, H. (2015). Flow in flexible/collapsible tubes.
156 *Fluid-Structure Interactions in Low-Reynolds-Number Flows* (ed. C. Duprat & H. Stone),
157 280–312.
- 158 Hernández-Sánchez, J. F., Lubbers, L. A., Eddi, A., & Snoeijer, J. H. (2012). Symmetric
159 and asymmetric coalescence of drops on a substrate. *Phys. Rev. Lett.*, 109, 184502.
160 <https://doi.org/10.1103/PhysRevLett.109.184502>
- 161 Kornblith, S., Thaney, K., & Hahnel, M. (2017). JLD2. In *GitHub repository*. GitHub.
162 <https://github.com/JuliaIO/JLD2.jl>
- 163 Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & Viggien, E. M. (2016).
164 *The lattice Boltzmann method: Principles and practice*. Springer. ISBN: 978-3-319-
165 44647-9
- 166 Lin, T. S., Murisic, N., Mema, E., Mahady, K., Seric, I., Lam, M., Barra, V., & Allaire,
167 R. (2019). ThinViscoelasticFilms. In *GitHub repository*. GitHub. <https://github.com/valeriabarra/ThinViscoelasticFilms>
168
- 169 Nesic, S. (2017). Stochastic thin films. In *GitHub repository*. GitHub. https://github.com/snesic/stochastic_thin_films
170
- 171 Pawar, N. D., Bahga, S. S., Kale, S. R., & Kondaraju, S. (2019). Symmetric and asymmetric
172 coalescence of droplets on a solid surface in the inertia-dominated regime. *Physics of*
173 *Fluids*, 31(9), 092106. <https://doi.org/10.1063/1.5119014>
- 174 Peschka, D., Haefner, S., Marquant, L., Jacobs, K., Münch, A., & Wagner, B. (2019).
175 Signatures of slip in dewetting polymer films. *Proceedings of the National Academy of*
176 *Sciences*, 116(19), 9275–9284. <https://doi.org/10.1073/pnas.1820487116>

- 177 Pihler-Puzović, D., Juel, A., Peng, G. G., Lister, J. R., & Heil, M. (2015). Displacement flows
178 under elastic membranes. Part 1. Experiments and direct numerical simulations. *Journal*
179 *of Fluid Mechanics*, 784, 487–511.
- 180 Pihler-Puzović, D., Périllat, R., Russell, M., Juel, A., & Heil, M. (2013). Modelling the sup-
181 pression of viscous fingering in elastic-walled hele-shaw cells. *Journal of Fluid Mechanics*,
182 731, 162–183.
- 183 Salmon, R. (1999). The lattice Boltzmann method as a basis for ocean circulation modeling.
184 *J. Mar. Res.*, 57(3), 503–535. <https://doi.org/doi:10.1357/002224099764805174>
- 185 Sander, O. (2020). *DUNE—the distributed and unified numerics environment* (Vol. 140).
186 Springer Nature.
- 187 Shah, M. S., Steijn, V. van, Kleijn, C. R., & Kreutzer, M. T. (2019). Thermal fluctuations
188 in capillary thinning of thin liquid films. *Journal of Fluid Mechanics*, 876, 1090–1107.
189 <https://doi.org/10.1017/jfm.2019.595>
- 190 Sui, Y., Maglio, M., Spelt, P. D. M., Legendre, D., & Ding, H. (2013). Inertial coalescence
191 of droplets on a partially wetting substrate. *Physics of Fluids*, 25(10), 101701. <https://doi.org/10.1063/1.4824108>
- 193 Wedershoven, H. M. J. M., Berendsen, C. W. J., Zeegers, J. C. H., & Darhuber, A. A. (2014).
194 Infrared laser induced rupture of thin liquid films on stationary substrates. *Applied Physics*
195 *Letters*, 104(5), 054101. <https://doi.org/10.1063/1.4863318>
- 196 Wilczek, M., Tewes, W., Engelnkemper, S., Gurevich, S. V., & Thiele, U. (2017). Sliding
197 drops: Ensemble statistics from single drop bifurcations. *Phys. Rev. Lett.*, 119, 204501.
198 <https://doi.org/10.1103/PhysRevLett.119.204501>
- 199 Zitz, S., Scagliarini, A., & Harting, J. (2021). Lattice Boltzmann simulations of stochastic
200 thin film dewetting. *Phys. Rev. E*, 104, 034801. <https://doi.org/10.1103/PhysRevE.104.034801>
- 202 Zitz, S., Scagliarini, A., Maddu, S., Darhuber, A. A., & Harting, J. (2019). Lattice Boltzmann
203 method for thin-liquid-film hydrodynamics. *Phys. Rev. E*, 100, 033313. <https://doi.org/10.1103/PhysRevE.100.033313>
- 204