







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

Stefan Zitz ^{1,2,3}✉, Manuel Zellhöfer¹, Andrea Scagliarini ^{4,5}, and Jens Harting ^{1,3}

¹ Helmholtz Institute Erlangen-Nürnberg for Renewable Energy, Erlangen, Germany ² Department of Science and Environment, Roskilde University, Roskilde, Denmark ³ Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany ⁴ Consiglio Nazionale delle Ricerche (CNR), Roma, Italy ⁵ INFN, sezione Roma “Tor Vergata”, Roma, Italy ✉ Corresponding author

DOI: [10.21105/joss.04312](https://doi.org/10.21105/joss.04312)

Software

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

Editor: [Arfon Smith](#)  

Reviewers:

- [@mbernaschi](#)
- [@Inacquaroli](#)

Submitted: 12 September 2021

Published: 05 September 2022

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/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 vapor) to the hydrodynamic flows. An efficient way to deal with this multiscale problem is 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 at the liquid-vapor interface. Solving the thin film equation directly is a difficult task, because it is a fourth order degenerate PDE ([Becker et al., 2003](#)). `Swalbe.jl` approaches this 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 approach serves two benefits, on the one hand the ease of implementation where the lattice Boltzmann method essentially comprises of two steps: *collision* and *streaming*. On the other hand due to the simple algorithm a straightforward approach 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 [Figure 1](#). Beyond films, low contact angle droplets can be studied and compared to 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](#); [Zitz et al., 2021](#)).

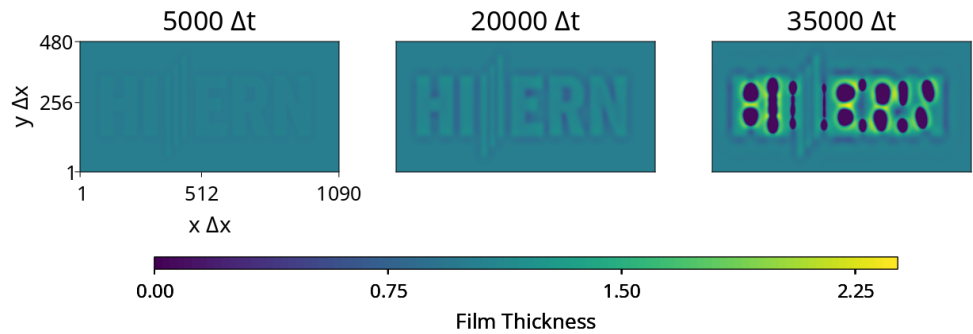


Figure 1: Dewetting simulation on a patterned substrate, letters have a higher wettability than the rest of the substrate.

Statement of need

Swalbe is written in Julia (Bezanson et al., 2017) and developed for a script your experiment workflow. For that reason an experiment is composed of three steps. First, the initial problem is defined by setting the system size and other input parameters, stored in a custom type. Followed by the lattice Boltzmann time loop where different force terms allow for different dynamics. Some forces, however, are mandatory for every experiment. This is, on the one hand, the friction with the substrate that helps regularizing the contact line (slippage) and, on the other hand, the capillary- or film pressure that accounts for the correct wetting behavior (contact angle).

After the time loop has ended an IO step can be included to write data to files or create plots. The package is written in pure Julia, therefore it can be easily coupled with other packages from the Julia ecosystem such as Plots.jl (Brelhoff, 2021) to visualize data and JLD2.jl (Kornblith et al., 2017) to store data in HDF5 format. One future development goal is to interact with the SciML environment to pair modelling with machine learning.

Swalbe.jl is designed to approach two problems, first being the modelling itself and second the applicability to large scale simulations.

Ideas can be implemented and tested quickly with a two dimensional system, which is discretized in a single horizontal direction and offers a second computed dimension for the thickness. The hardware requirements to run these two dimensional simulations are comparably low and depending on the number of lattice Boltzmann iterations ranging from seconds to at most an hour on a single core of a modern CPU. After testing it is possible to scale up and simulate the same or other problems in three dimensions with a slightly more complex discretization. Keeping the simulation time low is archived by using a Nvidia GPU. Most functions are written in a generic style and can be executed both on a CPU or GPU. For the GPU usage the high-level API of CUDA.jl (Besard et al., 2018, 2019) is used.

An older version of the numerical model (written in C++) has been tested and used for thin film simulations in previous publications (Zitz et al., 2019, 2021). While there is a small performance decrease when moving from C++ with OpenACC to Julia, the benefits of usability, straightforward documentation and automated testing outweighs this issue. In the future to authors plan to study switchable substrate coupled to a dewetting thin film, or thermal fluctuations in coalescing droplets using Swalbe.jl.

State of the field

In the context of computational fluid dynamics, low Reynolds number flows and especially thin film flows are a comparably small subsection. Therefore numerical tools that deal exclusively

with the thin film problem are sparse. Two packages for simulations of thin film hydrodynamics are **ThinViscoelasticFilms** and **stochastic_thin_films** (Lin et al., 2019; Nesic, 2017). The core components are written in Fortran and at least the later package can be used according to BSD-2 license. Documentation however is only available through code comments and a short readme, leaving the user little guidance.

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

Wilczek et al. used **DUNE** to study the dynamics of an ensemble of sliding drops in ref. (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. This mesoscopic approach is build on 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**, **PALABOS** or some smaller projects **STLBM**, **TLBfind** (explicitly written for GPU use).

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. Setting up a simulation for a thin film problem can therefore become fairly complex due to the generality these solvers offer. Especially concerning the Navier-Stokes solvers one uses a *sledge hammer to crack a nut*.

Use Case

In the domain of thin liquid films coalescence of sessile droplets is a broadly studied problem, see references (Aarts et al., 2005; Eddi et al., 2013; Eggers et al., 1999; Hernández-Sánchez et al., 2012). Two barely touching droplets on a hydrophilic substrate will coalesce into a single droplet to minimize their surface energy. From theoretical arguments we know that the bridge height, the point of minimal thickness between the two droplets, follows a power law. We now show how to perform that simulation with the help of **Swalbe.jl**.

The goal of this is to observe a growth of the bridge height h_0 given by

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

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  $\gamma$  in enumerate( $\gamma$ s)
    # System parameter,  $\delta=50$  can still considered small to medium slippage ( $\delta \approx \max(x)$ )
    sys = Swalbe.SysConst_1D(L=1024, Tmax=20000000,  $\delta=50.0$ ,  $\gamma=\gamma[2]$ )
    # The experiment
    data_merge[:, :,  $\gamma[1]$ ] = run_drop_coal(sys, r[]=sphere_rad, r[]=sphere_rad)
end
```

4. Display the results:

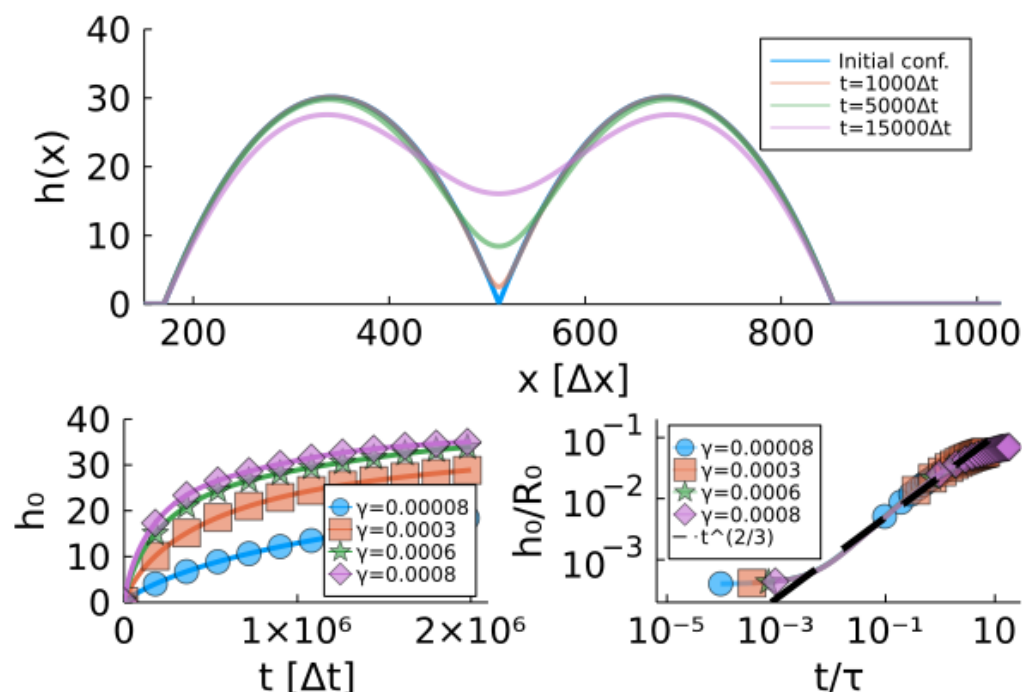


Figure 2: Coalescence of sessile droplets on a partially wetting substrate. The upper panel shows the time evolution for a single experiment (lowest surface tension). In the lower left panel we plot the evolution of the bridge height for the four different surface tensions. To the right we normalize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

Author Contribution Statement

The authors confirm contribution to the paper as follows: S. Z. wrote the code, performed the simulations and set up the package. M. Z. contributed to the continuous integration as well as the documentation. A. S. derived the mathematical approximations on which the packages is based on and J. H. served in an advisory role in the design and the execution of the research.

Acknowledgements

S. Zitz, M. Zellhöfer and J. Harting acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) within the priority program SPP2171 “Dynamic Wetting of Flexible, Adaptive, and Switchable Substrates”, under the project HA-4382/11.

References

- Aarts, D. G. A. L., Lekkerkerker, H. N. W., Guo, H., Wegdam, G. H., & Bonn, D. (2005). Hydrodynamics of droplet coalescence. *Phys. Rev. Lett.*, *95*, 164503. <https://doi.org/10.1103/PhysRevLett.95.164503>
- Becker, J., Grün, G., Seemann, R., Mantz, H., Jacobs, K., Mecke, K. R., & Blossey, R. (2003). Complex dewetting scenarios captured by thin-film models. *Nature Materials*, *2*(1), 59. <https://doi.org/10.1038/nmat788>
- Berendsen, C. W. J., Zeegers, J. C. H., Kruis, G. C. F. L., Riepen, M., & Darhuber, A. A. (2012). Rupture of thin liquid films induced by impinging air-jets. *Langmuir*, *28*(26), 9977–9985. <https://doi.org/10.1021/la301353f>
- Besard, T., Churavy, V., Edelman, A., & De Sutter, B. (2019). Rapid software prototyping for heterogeneous and distributed platforms. *Advances in Engineering Software*, *132*, 29–46. <https://doi.org/10.1016/j.advengsoft.2019.02.002>
- Besard, T., Foket, C., & De Sutter, B. (2018). Effective extensible programming: Unleashing Julia on GPUs. *IEEE Transactions on Parallel and Distributed Systems*. <https://doi.org/10.1109/TPDS.2018.2872064>
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. <https://doi.org/10.1137/141000671>
- Bonn, D., Eggers, J., Indekeu, J., Meunier, J., & Rolley, E. (2009). Wetting and spreading. *Rev. Mod. Phys.*, *81*, 739–805. <https://doi.org/10.1103/RevModPhys.81.739>
- Brelhoff, T. (2021). *Plots.jl*. <https://doi.org/10.5281/zenodo.4907285>
- Chapman, S., & Cowling, T. G. (1939). *The mathematical theory of non-uniform gases*. Cambridge University Press.
- Diez, J. A., Kondic, L., & Bertozzi, A. (2000). Global models for moving contact lines. *Phys. Rev. E*, *63*, 011208. <https://doi.org/10.1103/PhysRevE.63.011208>
- Eddi, A., Winkels, K. G., & Snoeijer, J. H. (2013). Influence of droplet geometry on the coalescence of low viscosity drops. *Phys. Rev. Lett.*, *111*, 144502. <https://doi.org/10.1103/PhysRevLett.111.144502>
- Eggers, J., Lister, J. R., & Stone, H. A. (1999). Coalescence of liquid drops. *Journal of Fluid Mechanics*, *401*, 293–310. <https://doi.org/10.1017/S002211209900662X>
- Engelnkemper, S., Wilczek, M., Gurevich, S. V., & Thiele, U. (2016). Morphological transitions of sliding drops: Dynamics and bifurcations. *Phys. Rev. Fluids*, *1*, 073901. <https://doi.org/10.1103/PhysRevFluids.1.073901>
- Enskog, D. (1917). *Kinetische theorie der vorgänge in mässig verdünnten gasen*. PhD Thesis.
- Heil, M., Hazel, A. L., Duprat, C., & Stone, H. (2015). Flow in flexible/collapsible tubes. *Fluid-Structure Interactions in Low-Reynolds-Number Flows* (Ed. C. Duprat & H. Stone), 280–312.

- Hernández-Sánchez, J. F., Lubbers, L. A., Eddi, A., & Snoeijer, J. H. (2012). Symmetric and asymmetric coalescence of drops on a substrate. *Phys. Rev. Lett.*, *109*, 184502. <https://doi.org/10.1103/PhysRevLett.109.184502>
- Kornblith, S., Thaney, K., & Hahnel, M. (2017). JLD2. In *GitHub repository*. GitHub. <https://github.com/JuliaIO/JLD2.jl>
- Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & Viggen, E. M. (2016). *The lattice Boltzmann method: Principles and practice*. Springer. ISBN: 978-3-319-44647-9
- Lin, T. S., Murisic, N., Mema, E., Mahady, K., Seric, I., Lam, M., Barra, V., & Allaire, R. (2019). ThinViscoelasticFilms. In *GitHub repository*. GitHub. <https://github.com/valeriabarra/ThinViscoelasticFilms>
- Nesic, S. (2017). Stochastic thin films. In *GitHub repository*. GitHub. https://github.com/snesic/stochastic_thin_films
- Pawar, N. D., Bahga, S. S., Kale, S. R., & Kondaraju, S. (2019). Symmetric and asymmetric coalescence of droplets on a solid surface in the inertia-dominated regime. *Physics of Fluids*, *31*(9), 092106. <https://doi.org/10.1063/1.5119014>
- Peschka, D., Haefner, S., Marquant, L., Jacobs, K., Münch, A., & Wagner, B. (2019). Signatures of slip in dewetting polymer films. *Proceedings of the National Academy of Sciences*, *116*(19), 9275–9284. <https://doi.org/10.1073/pnas.1820487116>
- Pihler-Puzović, D., Juel, A., Peng, G. G., Lister, J. R., & Heil, M. (2015). Displacement flows under elastic membranes. Part 1. Experiments and direct numerical simulations. *Journal of Fluid Mechanics*, *784*, 487–511. <https://doi.org/10.1017/jfm.2015.590>
- Pihler-Puzović, D., Périllat, R., Russell, M., Juel, A., & Heil, M. (2013). Modelling the suppression of viscous fingering in elastic-walled hele-shaw cells. *Journal of Fluid Mechanics*, *731*, 162–183. <https://doi.org/10.1017/jfm.2013.375>
- Salmon, R. (1999). The lattice Boltzmann method as a basis for ocean circulation modeling. *J. Mar. Res.*, *57*(3), 503–535. <https://doi.org/10.1357/002224099764805174>
- Sander, O. (2020). *DUNE—the distributed and unified numerics environment* (Vol. 140). Springer Nature.
- Shah, M. S., Steijn, V. van, Kleijn, C. R., & Kreutzer, M. T. (2019). Thermal fluctuations in capillary thinning of thin liquid films. *Journal of Fluid Mechanics*, *876*, 1090–1107. <https://doi.org/10.1017/jfm.2019.595>
- Sui, Y., Maglio, M., Spelt, P. D. M., Legendre, D., & Ding, H. (2013). Inertial coalescence of droplets on a partially wetting substrate. *Physics of Fluids*, *25*(10), 101701. <https://doi.org/10.1063/1.4824108>
- Wedershoven, H. M. J. M., Berendsen, C. W. J., Zeegers, J. C. H., & Darhuber, A. A. (2014). Infrared laser induced rupture of thin liquid films on stationary substrates. *Applied Physics Letters*, *104*(5), 054101. <https://doi.org/10.1063/1.4863318>
- Wilczek, M., Tewes, W., Engelkemper, S., Gurevich, S. V., & Thiele, U. (2017). Sliding drops: Ensemble statistics from single drop bifurcations. *Phys. Rev. Lett.*, *119*, 204501. <https://doi.org/10.1103/PhysRevLett.119.204501>
- Zitz, S., Scagliarini, A., & Harting, J. (2021). Lattice Boltzmann simulations of stochastic thin film dewetting. *Phys. Rev. E*, *104*, 034801. <https://doi.org/10.1103/PhysRevE.104.034801>
- Zitz, S., Scagliarini, A., Maddu, S., Darhuber, A. A., & Harting, J. (2019). Lattice Boltzmann method for thin-liquid-film hydrodynamics. *Phys. Rev. E*, *100*, 033313. <https://doi.org/10.1103/PhysRevE.100.033313>