DORiE – A Versatile Discontinuous Galerkin Richards Solver

Lukas Riedel,[1,2] Dion Häfner,[1] Ole Klein,[2] and Kurt Roth[1,2]

[1]*Institute of Environmental Physics, Heidelberg University*
[2]*Interdisciplinary Center for Scientific Computing, Heidelberg University*

Interdisciplinary Center for Scientific Computing

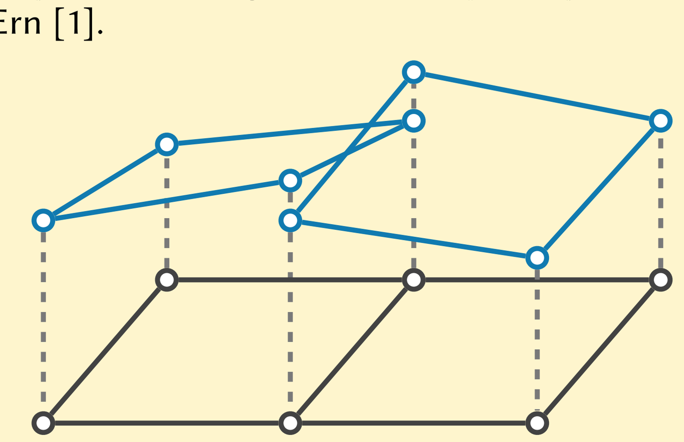
The Setting

- Richards' equation describes the highly non-linear fluid dynamics in unsaturated porous media. The phenomena exhibited are very sensitive to the forcing, the soil architecture, and the parameterization applied. DORiE solves the equation for the continuous matric head h_m.

$$\frac{\partial}{\partial t}\theta_w + \nabla \cdot \boldsymbol{j}_w = 0$$

$$\boldsymbol{j}_w = -K(\theta_w)\left[\nabla h_m - \hat{\boldsymbol{g}}\right]$$

- Discontinuous Galerkin (DG) schemes approximate solutions of partial differential equations (PDEs) with discontinuous polynomials. They are highly accurate and flexible, yet computationally expensive. DORiE implements the symmetric weighted interior penalty (SWIP) method derived by Di Petro & Ern [1].



- The Distributed and Unified Numerics Environment (DUNE) is a C++ template library for solving PDEs [2]. It implements various grid managers, and supplies a common interface for setting up polynomial function spaces and solving equation systems. DORiE uses classes and operators of the DUNE-PDELab discretization module for solving Richards Equation.

Dune
Distributed and Unified Numerics Environment

Availability

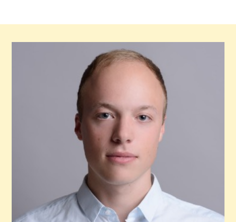
- Visit our website for the most recent information on DORiE:

`http://ts.iup.uni-heidelberg.de/research/terrestrial-systems/dorie/`

- A pre-compiled image for executing DORiE has been uploaded to the Docker Hub. After installing Docker, you can download and directly run the image:

```
$ docker run -it -v $PWD:/sim
        dorie/dorie:latest <cmd> <args>
```

Contact



Lukas Riedel
lriedel@iup.uni-heidelberg.de
http://ts.iup.uni-heidelberg.de/people/lukas-riedel/

The Program

DORiE is an actively developed software suite for computing transient soil water flow with a DG discretization scheme. The solver features

- structured & unstructured rectangular or simplex grids (2D & 3D)
- adaptive local grid refinement based on a posteriori flux error estimators [1]
- an adaptive, implicit two-stage time step scheme
- grid-independent soil architectures supporting Miller-scaling
- transient boundary conditions on segmented boundaries
- Python wrappers for program setup, execution control, & output analysis

Benchmarks: Precision

- We benchmark DORiE and muPhi ('μφ') which implements a finite volume scheme [3]. Similar to DORiE, it uses DUNE implementations to solve (non-)linear equation systems.
- For vertically symmetric setups disregarding transient dynamics, Richards equation can be reduced to an ordinary differential equation (ODE).
- We compute symmetric 2D steady state solutions with muPhi and DORiE for fixed boundary conditions and a homogeneous medium, and compare these to more accurate, numeric ODE solutions.

Results:

- DORiE achieves a higher precision and better convergence rates in the solution L2 error norm than muPhi.
- Finite elements of 2nd and 3rd order achieve convergence rates 2 … 4, depending on the domain parameterization.

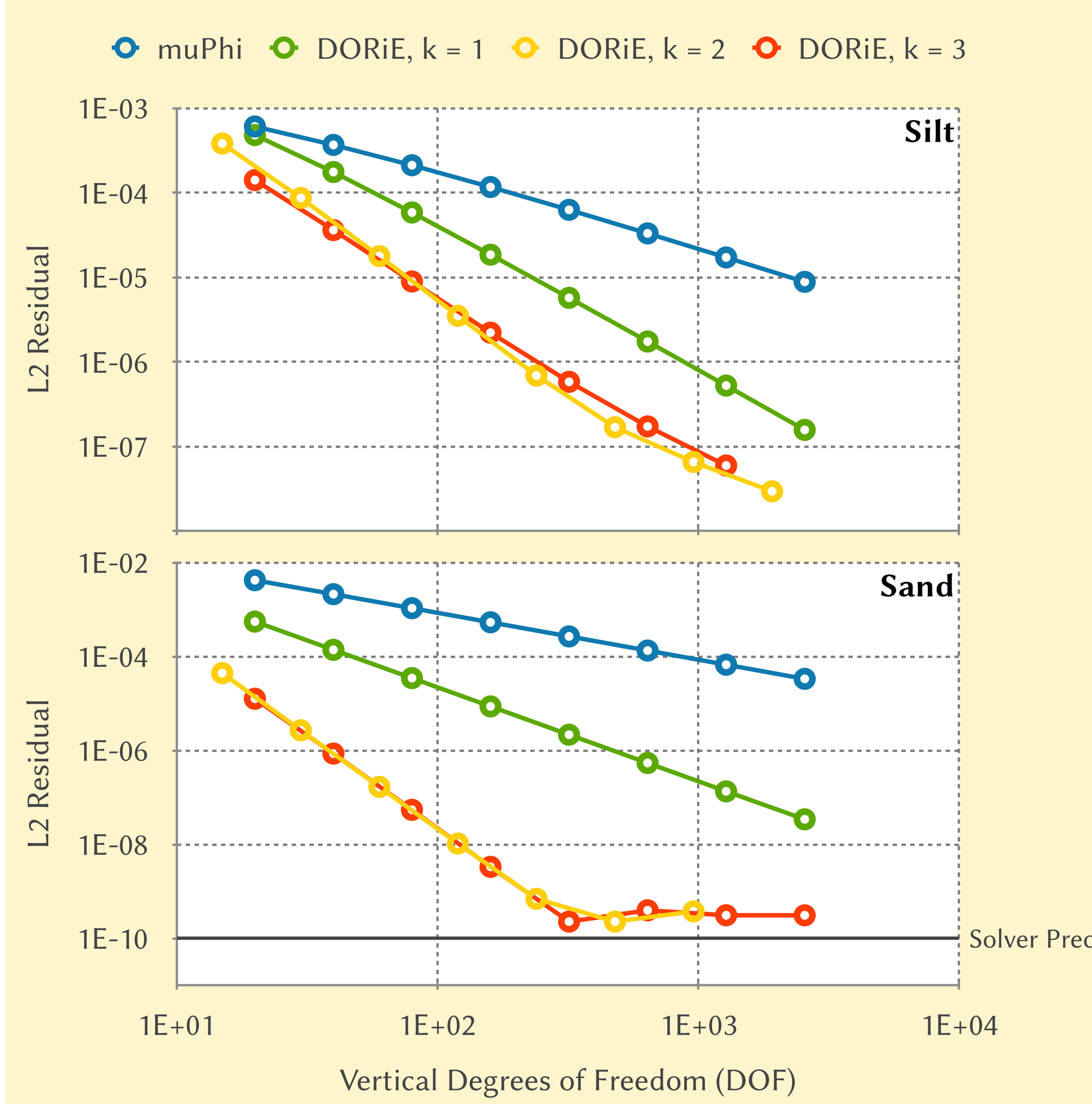


Figure 2: Precision benchmarks of steady-state solutions for two different soil parameterizations. The polynomial order of the solution function spaces is given by k.

Showcase

Figure 1 (right): 2D-Simulation of uniform rainfall infiltrating a Miller-similar sandy soil. Colors encode saturated hydraulic conductivity which ranges over almost three orders of magnitude (see color bar). Arrows indicate flux direction and magnitude. The grid is locally refined or coarsened based on the error estimate for fluxes across grid intersections.

Benchmarks: Runtime (2D)

- We benchmark DORiE against muPhi [3]. With DORiE, we employ both structured and adaptive grids with different resolutions and flux error estimate tolerances, respectively.
- muPhi is restricted to structured rectangular grids, but optimized for high performance and scalability.
- We compute transient solutions of infiltration into a fixed heterogeneous soil architecture on different grids, and compare these to a high-precision reference solution of DORiE.

Results:

- muPhi is faster than DORiE in the low resolution regime.
- Localizing grid resolution/refinement to the dynamics of interest is crucial for a precise solution.
- Adaptive grid coarsening mostly outweighs the efficiency benefits of structured grids for DORiE.

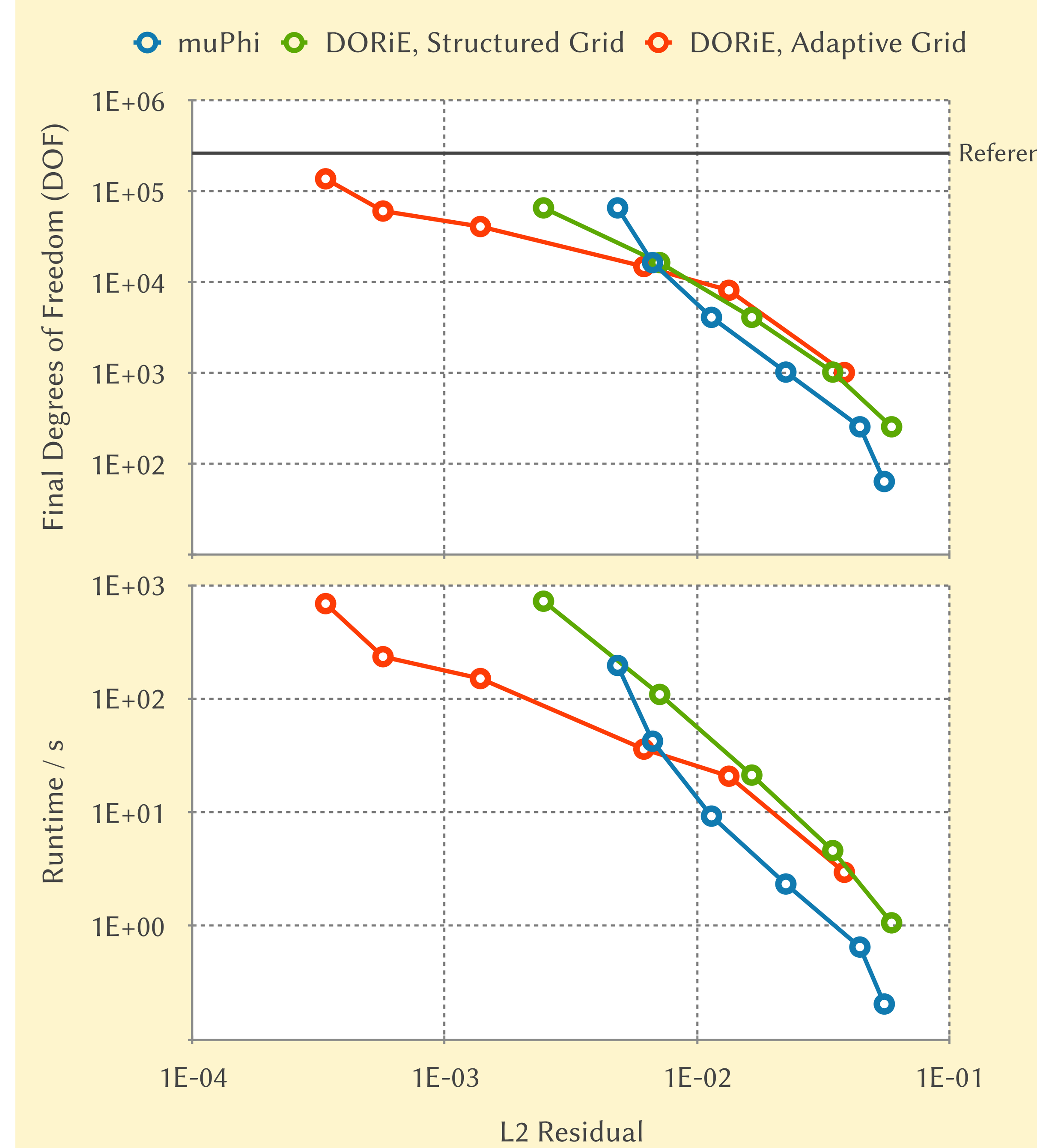
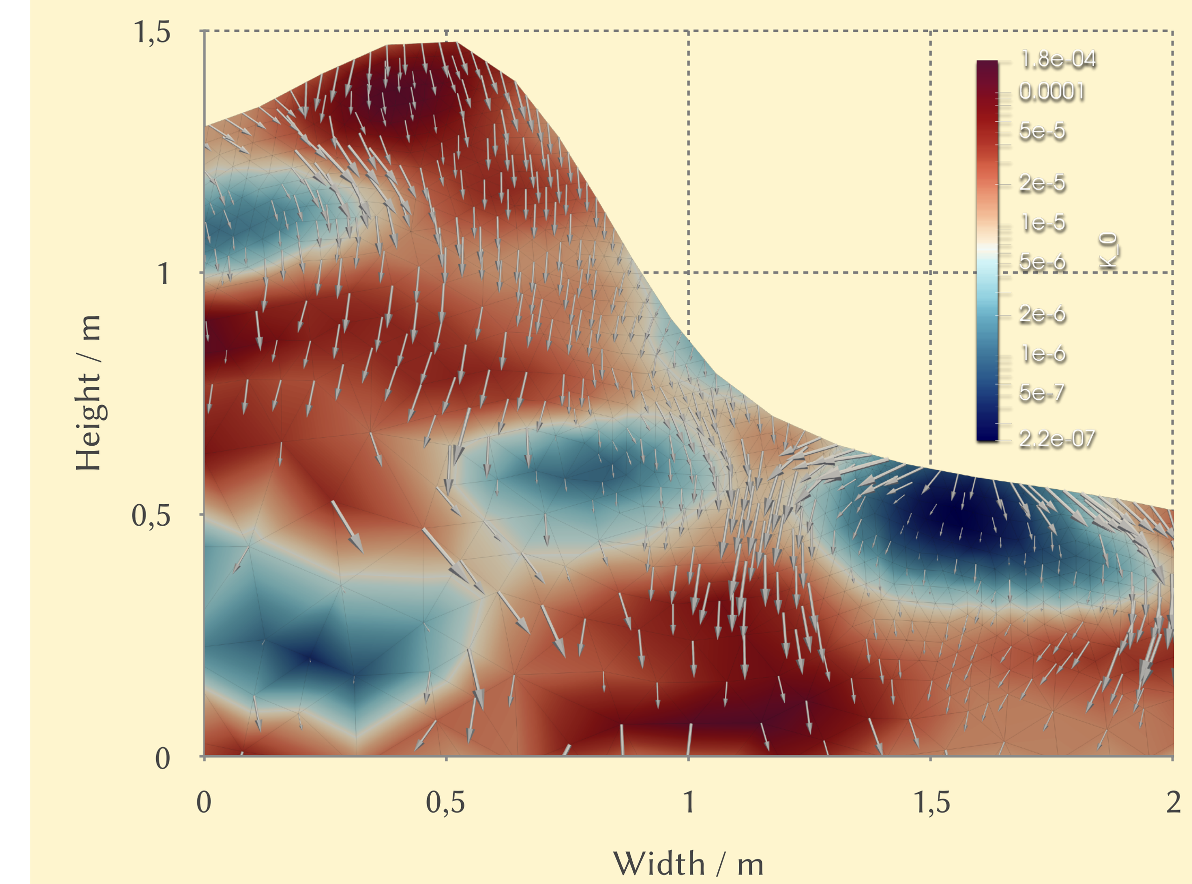


Figure 3: Runtime benchmarks for heterogeneous soil. Residuals are computed with respect to a high-precision DORiE solution. DORiE is only tested in polynomial order k = 1.

Showcase



Conclusion

- Using the DUNE backend and the SWIP DG stencil, DORiE computes solutions on different grid configurations using the same methods and operators.
- The DG method ensures a high accuracy of the solution and outperforms finite element and finite volume schemes.
- The algorithms for adaptive local grid refinement and time step size adaptation efficiently balance runtime against precision.
- With Docker, DORiE can be built and executed without the hassle of managing software dependencies.
- DORiE is actively maintained and developed. The source code is managed with a GitLab repository.

GitLab docker

References

[1] Di Petro, D. A. & Ern, A. 2012. *Mathematical Aspects of Discontinuous Galerkin Methods.* Mathématiques at Applications 69. Berlin & Heidelberg: Springer.

[2] Blatt, M. et al. 2016. "The Distributed and Unified Numerics Environment, Version 2.4." *Archive of Numerical Software* 4 (100): 13–29. `https://dune-project.org`

[3] Ippisch, O., Vogel, H.-J., & Bastian, P. 2006. "Validity limits for the van Genuchten–Mualem model and implications for parameter estimation and numerical simulation." *Advances in Water Resources* 29: 1780–1789. `http://fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/muPhi/_node.html`