Weierstrass Institute for Applied Analysis and Stochastics

WIAS-PDELib: A Julia PDE solver ecosystem in a GitHub organization

Jürgen Fuhrmann, Patrick Jaap, Christian Merdon, Jan Philipp Thiele and contributors

Intro

The Weierstrass Institute for Applied Analysis and Stochastics (WIAS) develops PDE models, numerical methods and software for academic and industrial applications. WIAS-PDELib is a collection of Julia packages for the solution of PDEs. For efficient development, management and maintenance of the code, we created a GitHub organization.

Benefits

Visibility

The individual repositories of the **packages** are placed in **github.com/WIAS-PDELib** and thus share a common namespace. Users can easily **discover** related packages together with **documentation** and **changelogs**. Cross-referencing **Issues** and **Pull Requests** along different repositories is possible.

Integration

All packages are automatically **tested** for functionality and **interaction** with each other after each change. Periodically, automated **dependency** bumps are performed. **Documentation** pages are built automatically and accessible directly from the individual repositories.

Consistent Formatting

Across WIAS-PDELib, we enforce the same **coding style**. This improves readability and recognizability of our code a lot. For each Pull Request an automated **format check** is performed and will prohibit **merging** if format issues are found.

Expert Teams

A GitHub organization allows **fine grained** control over the **maintenance roles** of the individual developers. Each **individual** is only granted the necessary rights to develop and maintain parts of the repositories. Gathering developers in **teams** reduces the management effort by defining **team roles**.

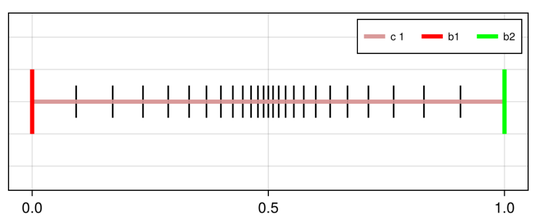
Internal Discussions

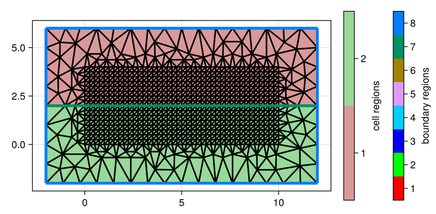
Not all details should be discussed in **public threads**. We have **internal** issue trackers only visible for **members** of the GitHub organization. New members are able to read internal documentations and guidelines.

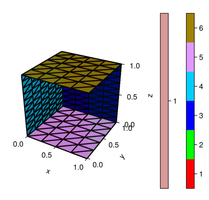
WIAS-PDELib

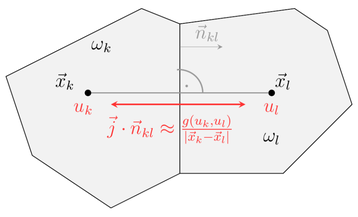
Grid Generation and **Grid Handling**

- **SimplexGridFactory.jl**: methods to construct grids on the fly
- meshing backends: **Tetgen.jl**, **Triangulate.jl**, **Gmsh.jl**
- **ExtendableGrids.jl**: grid representation with geometric properties
- efficient index mappings for cells, faces, edges, nodes









Finite Volume Discretization: VoronoiFVM.jl

- Solver for coupled convection-reaction-diffusion systems

$$\partial_t s(u) + \nabla \cdot \overrightarrow{j} + r(u) - f = 0$$

- description by (nonlinear) **storage**, **flux** and **reaction** operators
- in 1D, 2D or 3D, with various **boundary operators**

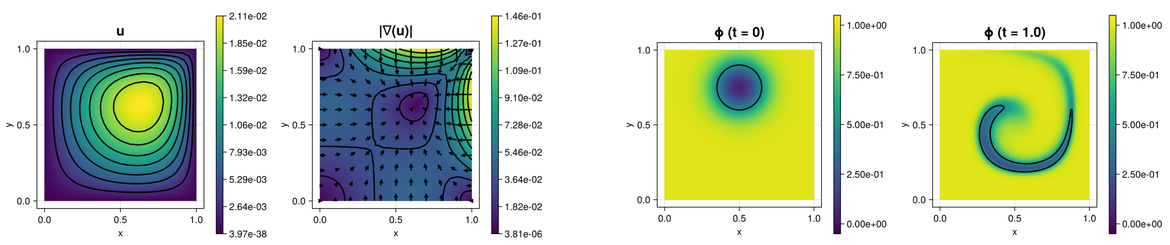
Finite Element Discretization: ExtendableFEM.jl

- High level FE toolbox for nonlinear multiphysics systems
- including FE spaces for $H(\mathrm{curl})$ and $H(\mathrm{div})$ in different dimensions
- **gradient-robust** flow discretizations
- operator form: (bi-)linear, nonlinear, boundary, **periodic coupling**, ...

Assembly and **Solvers**

- Newton's method with Jacobians via Julia's **automatic differentiation**
- efficient assembly of sparse linear system matrices
- compatible to **SciML**'s linear (**LinearSolve.jl**) and ODE (**DifferentialEquations.jl**) solver libraries
- you can use your **own** solver, preconditioners and error estimators

```
#IT   ------- RESIDUALS -------        ---- DURATION (s) ----          ---- ALLOCATIONS (MiB) ----
      NONLINEAR       LINEAR          ASSEMB  SOLVE   TOTAL           ASSEMB  SOLVE   TOTAL
INI                                                   0.18                            4.47
  1   7.450e-02       5.290e-16       4.38    0.01    4.40            220.28  0.37    220.65
  2   2.965e-03       4.450e-16       0.47    0.00    0.47            31.17   0.15    31.32
  3   1.222e-05       4.818e-16       0.00    0.00    0.00            0.01    0.15    0.16
  4   2.120e-10       4.597e-16       0.00    0.00    0.00            0.01    0.15    0.16
END   6.232e-16                       0.00            0.00            0.01            0.01
      converged                               SUM --> 5.04                    SUM --> 256.79
```



Visualization

- **GridVisualize.jl**: plot grids and results
- support for different backends: **Makie.jl**, **PyPlot.jl**, ...
- support for **slices**, **animations**
- interactive view: **zoom** and **rotate** in real time

Julia

A fresh Approach to Scientific Computing

- goal: flexible as Python, fast as C++
- **just-ahead-of-time** compilation of source code with caching
- strongly typed, **multiple dispatch**
- comprehensive **linear algebra** included
- **parallel**: built-in SIMD, MPI, OpenMP support
- directly call **Python**, **R** and **C** code
- automatic differentiation
- MIT license

Packages and Reproducibility

- every project is a Julia **package**
- common **structure** for sources, tests, scripts, documentation
- powerful **package manager**: inherited from Python, R
- public packages available in the **General Registry**
- registration of **new packages** and **new versions** is simple
- every project tracks its **internal state** in a manifest file
- with the manifest, computations are **reproducible** on different machines

Contact
github.com/WIAS-PDELib
<last name>@wias-berlin.de
Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany

Image Sources
https://github.com/WIAS-PDELib
https://wias-pdelib.github.io/SimplexGridFactory.jl
https://wias-pdelib.github.io/ExtendableGrids.jl
https://wias-pdelib.github.io/ExtendableFEM.jl
https://wias-pdelib.github.io/VoronoiFVM.jl
https://wias-pdelib.github.io/GridVisualize.jl
https://github.com/JuliaLang/julia-logo-graphics

Leibniz Association