Flow problems in porous media: Modeling, approximation and implementation

Monika Eisenmann

Presentation for eSSENCE

March 31, 2023

LUND UNIVERSITY

Outline

(A) Development of a model for the assessment of the performance of groundwater remediation.

(B) Construction and analysis of modern time-stepping methods.

(C) Efficient implementation of the time-stepping methods for relevant problems.

Collaboration

My background

- Associate senior lecturer at Centre for Mathematical Sciences in Lund.
- Research in numerical analysis of (stochastic) differential equations

Collaborator: Léa Lévy

- Associate senior lecturer at the division of Engineering Geology at Lund University.
- Research in the assessment of the performance of groundwater remediation.

Collaborator: Robert Klöfkorn

- Senior lecturer at Centre for Mathematical Sciences in Lund.
- Research in scientific computing and co-founder of the code base DUNE.

Outline

(A) Development of a model for the assessment of the performance of groundwater remediation.

(B) Construction and analysis of modern time-stepping methods.

(C) Efficient implementation of the time-stepping methods for relevant problems.

(A) Underlying application

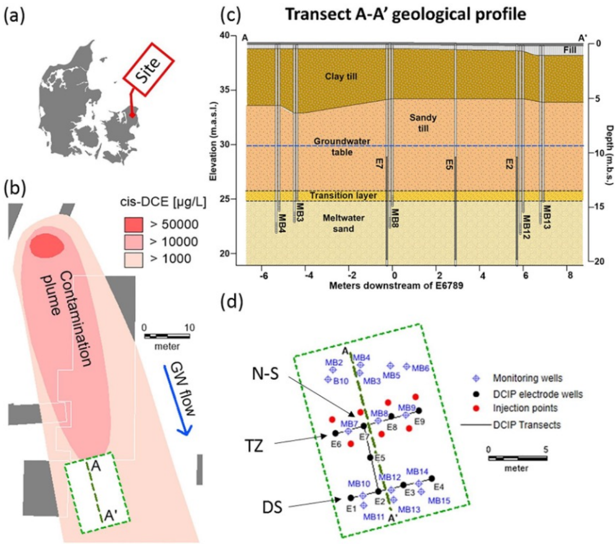
Ground water contamination

- Growth in consumption can decreasing groundwater quality.
- To remediate the contamination, a reactive agent is injected into groundwater to degrade or stop the contamination.

Then:

- Adequate spreading of the reagent is a challenge.
- Use geo-electrical methods to quantify reagent spreading.

(A) Underlying application



From [Lévy Et al., 2022].

(A) Modeling of the ground water flow

Porous Media Equation

$$\begin{cases} \dot{u} + \Delta\Phi(u) = f(t,u), \quad t \in (0,T), \\ u(0) = u_0, \end{cases}$$

The function f can include lower-order derivatives of u such as

- advection $\gamma(t, \nabla u)$,
- reaction $\rho(t, u)$,
- source term of the type $\beta(t)$.

(A) Connection to application

Problem

- Precise structure of the ground and its different layers is not known.
- The structure is important to describe the coefficients and nonlinear functions that appear in the system.

What to do

- To find suitable coefficients for the differential equations, we can use a Markov-chain-Monte-Carlo (MCMC).
- This has been used in [Irving & Singha, 2010] for an artificial data set for the resistivity equation.

(A) Goals of the project

- Use the MCMC approach for the resistivity equations and a data set obtained by Léa Lévy.
- The resulting coefficients are now random points with respect to a suitable distribution.
  $\Rightarrow$ We have a stochastic differential equation instead of a deterministic problem.
- The stochastic equation gives a real world application for parts (B) and (C) of the problem.
- Add a time dependency to the electrical model.

Outline

(A) Development of a model for the assessment of the performance of groundwater remediation.

(B) Construction and analysis of modern time-stepping methods.

(C) Efficient implementation of the time-stepping methods for relevant problems.

(B) Temporal discretization

Approximate the solution of

$$\begin{cases} \frac{\partial u(t)}{\partial t} - \Delta u(t) = f(t) \quad \text{for } t \in (0, T], \\ u(0) = u_0. \end{cases}$$

(B) Temporal discretization

Approximate the solution of

$$\begin{cases} \frac{\partial u(t,x)}{\partial t} - \frac{\partial^2 u(t,x)}{\partial x_1^2} - \frac{\partial^2 u(t,x)}{\partial x_2^2} = f(t,x) \quad \text{for } t \in (0, T], \\ u(0) = u_0. \end{cases}$$

(B) Temporal discretization

Approximate the solution of

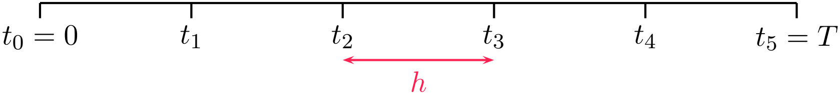
$$\begin{cases} \frac{\partial u(t)}{\partial t} - \Delta u(t) = f(t) & \text{for } t \in (0, T], \\ u(0) = u_0. \end{cases}$$

(B) Temporal discretization

Approximate the solution of

$$\begin{cases} \frac{\partial u(t)}{\partial t} - \Delta u(t) = f(t) & \text{for } t \in (0, T], \\ u(0) = u_0. \end{cases}$$

We subdivide the temporal interval

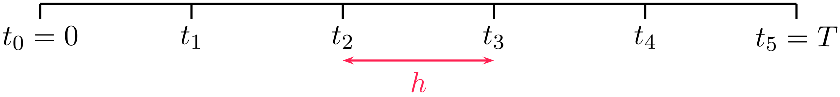


(B) Temporal discretization

Approximate the solution of

$$\begin{cases} \frac{\partial u(t)}{\partial t} - \Delta u(t) = f(t) & \text{for } t \in (0, T], \\ u(0) = u_0. \end{cases}$$

We subdivide the temporal interval



Then we approximate $u(t_n) \approx U^n$ defined by

$$\frac{U^n - U^{n-1}}{h} - \Delta U^n = f(t_n).$$

(B) Error Analysis

We have

$$\{U^0, U^1, \dots, U^N\}.$$

We want to find

$$\{u(0), u(t_1), \dots, u(T)\}.$$

(B) Error Analysis

We have

$$\{U^0, U^1, \dots, U^N\}.$$

We want to find

$$\{u(0), u(t_1), \dots, u(T)\}.$$

Questions

- Is U^n an approximation of $u(t_n)$?
- Does U^n an approximate $u(t_n)$ better if h becomes smaller?
- Can we make statements about the size of the error before hand?

(B) Error Analysis

We have

$$\{U^0, U^1, \ldots, U^N\}.$$

We want to find

$$\{u(0), u(t_1), \ldots, u(T)\}.$$

Tasks

- For a given problem, find a suitable scheme.
- Analyze the error.
- Find error bounds.

(B) Back to test equation

We want to approximate the solution of

$$\begin{cases} \frac{\partial u(t)}{\partial t} - \Delta u(t) = f(t) & \text{for } t \in (0, T] \\ u(0) = u_0. \end{cases}$$

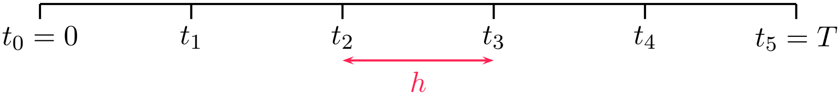
.

(B) Back to test equation

We want to approximate the solution of

$$\begin{cases} \frac{\partial u(t)}{\partial t} - \Delta u(t) = f(t) & \text{for } t \in (0, T] \\ u(0) = u_0. \end{cases}$$

We subdivide the temporal interval



Then we can approximate $u(t_n) \approx U^n$ defined by

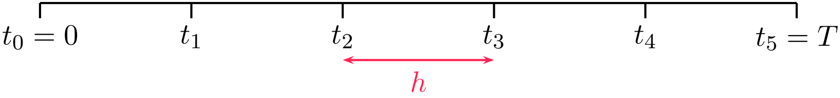
$$(I - h\Delta)U^n = U^{n-1} + hf(t_n).$$

(B) Back to test equation

We want to approximate the solution of

$$\begin{cases} \frac{\partial u(t,x)}{\partial t} - \Delta u(t,x) = f(t,x) & \text{for } t \in (0, T], x \in \mathcal{D} \\ u(0) = u_0. \end{cases}$$

We subdivide the temporal interval



Then we can approximate $u(t_n, x) \approx U^n(x)$ defined by

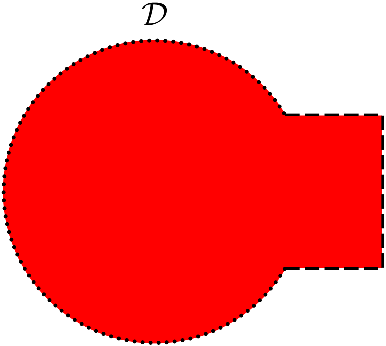
$$(I - h\Delta)U^n(x) = U^{n-1}(x) + hf(t_n, x), \quad x \in \mathcal{D}.$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$

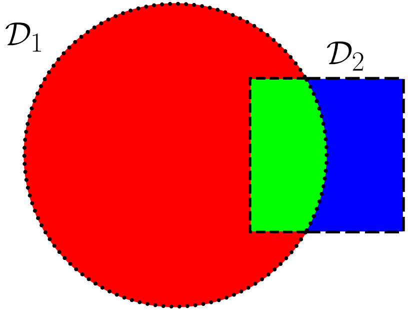


(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$

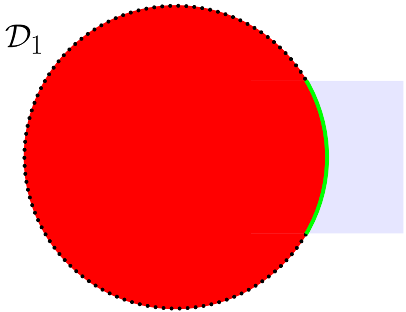


(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

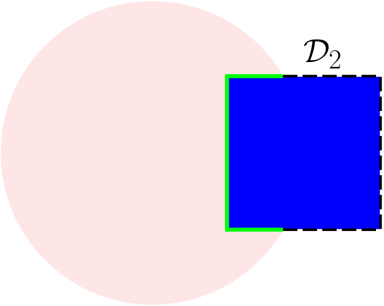
$$\begin{cases} (I - h\Delta)u_1 = \tilde{f} & \text{in } \mathcal{D}_1, \\ u_1 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_1, \\ u_1 = ? & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2., \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

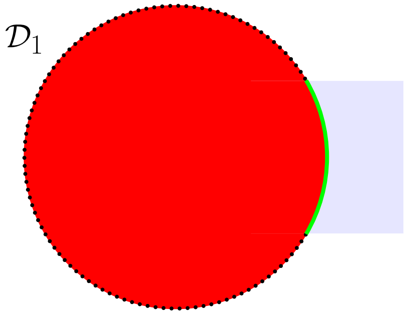
$$\begin{cases} (I - h\Delta)u_2 = \tilde{f} & \text{in } \mathcal{D}_2, \\ u_2 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_2, \\ u_2 = ? & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

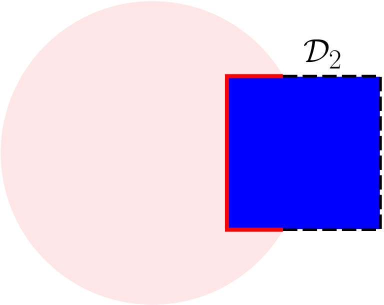
$$\begin{cases} (I - h\Delta)u_1^1 = \tilde{f} & \text{in } \mathcal{D}_2, \\ u_1^1 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_2, \\ u_1^1 = 0 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

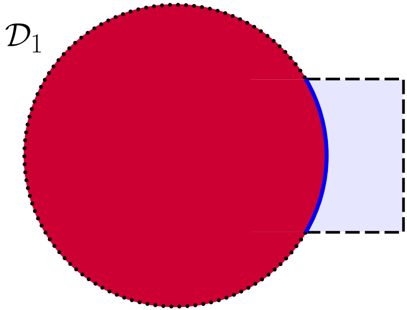
$$\begin{cases} (I - h\Delta)u_2^1 = \tilde{f} & \text{in } \mathcal{D}_2, \\ u_2^1 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_2, \\ u_2^1 = u_1^1 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

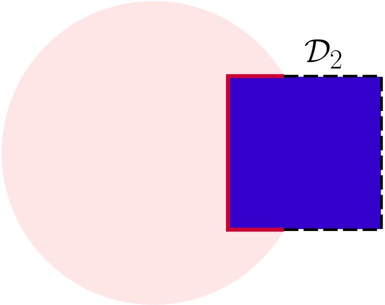
$$\begin{cases} (I - h\Delta)u_1^2 = \tilde{f} & \text{in } \mathcal{D}_1, \\ u_1^2 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_1, \\ u_1^2 = u_2^1 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

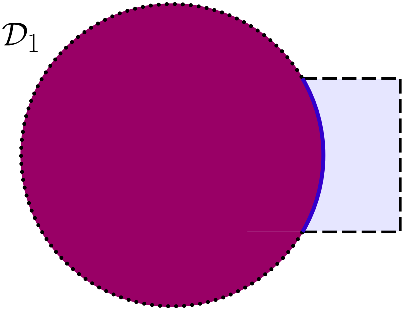
$$\begin{cases} (I - h\Delta)u_2^2 = \tilde{f} & \text{in } \mathcal{D}_2, \\ u_2^2 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_2, \\ u_2^2 = u_1^2 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

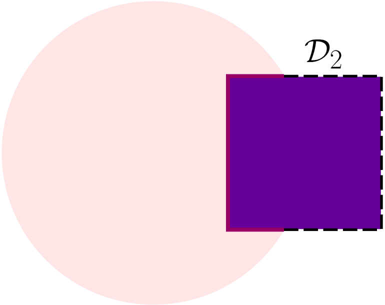
$$\begin{cases} (I - h\Delta)u_1^3 = \tilde{f} & \text{in } \mathcal{D}_1, \\ u_1^3 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_1, \\ u_1^3 = u_2^2 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

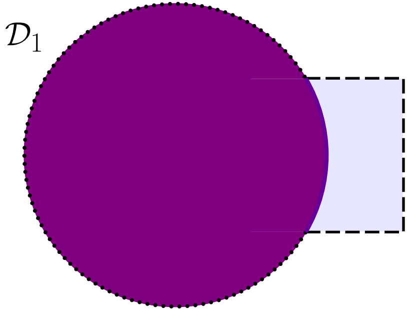
$$\begin{cases} (I - h\Delta)u_2^3 = \tilde{f} & \text{in } \mathcal{D}_2, \\ u_2^3 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_2, \\ u_2^3 = u_1^3 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

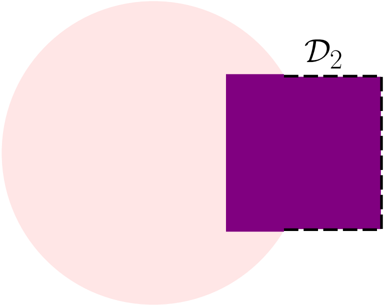
$$\begin{cases} (I - h\Delta)u_1^4 = \tilde{f} & \text{in } \mathcal{D}_1, \\ u_1^4 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_1, \\ u_1^4 = u_2^3 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



- It can be easier to solve

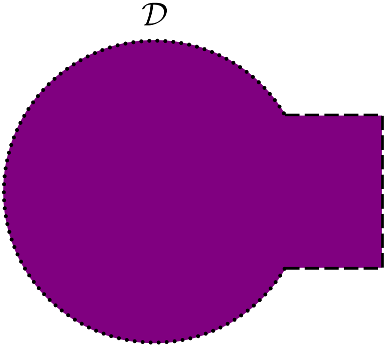
$$\begin{cases} (I - h\Delta)u_2^4 = \tilde{f} & \text{in } \mathcal{D}_2, \\ u_2^4 = 0 & \text{on } \partial\mathcal{D} \cap \partial\mathcal{D}_2, \\ u_2^4 = u_1^4 & \text{on } \partial\mathcal{D}_1 \cap \partial\mathcal{D}_2. \end{cases}$$

(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$



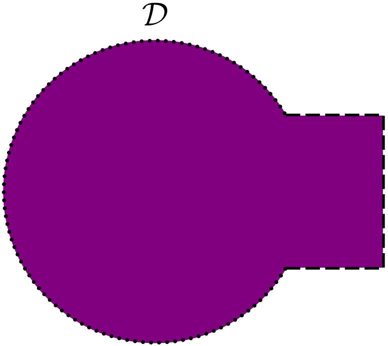
(B) Domain decomposition: Schwartz method

- First: We look at a classical method for an iterative domain decomposition scheme.

- Example problem:

$$\begin{cases} (I - h\Delta)u = \tilde{f} & \text{in } \mathcal{D}, \\ u = 0 & \text{on } \partial\mathcal{D}. \end{cases}$$

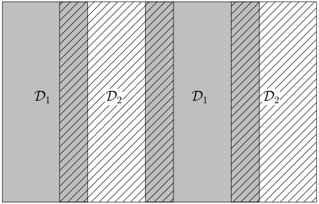
- This method has many further variations.



(B) Domain decomposition

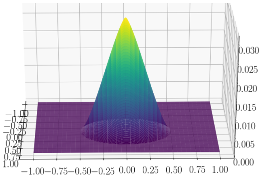
Then:

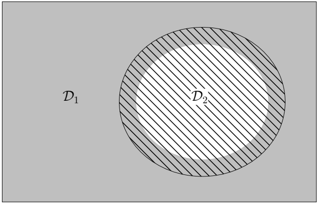
- Incorporate the domain decomposition in the time-stepping method through an operator splitting.
- Offers possibility for parallel implementation.



A decomposition into two sub-domains $\mathcal{D}_1, \mathcal{D}_2$ for rectangular domain $\mathcal{D}$.

(B) Nonlinear equations





- Left: Plot of a nonlinear pulse with a sharp edge (free boundary).
- Right: The support (part of the domain where the solution is non-zero) of the pulse.

(B) What has been done

Deterministic nonlinear differential equation

$$\begin{cases} \dot{u} + A(t,u) = f(t,u), & t \in (0,T), \\ u(0) = u_0, \end{cases}$$

as for example the porous media equation

$$\begin{cases} \dot{u} = \Delta\Phi(u), & (t,x) \in (0,T) \times \mathcal{D}, \\ u(0) = u_0. \end{cases}$$

(B) What has been done

Deterministic nonlinear differential equation

$$\begin{cases} \dot{u} + A(t,u) = f(t,u), \quad t \in (0,T), \\ u(0) = u_0, \end{cases}$$

as for example the porous media equation

$$\begin{cases} \dot{u} = \Delta\Phi(u), \quad (t,x) \in (0,T) \times \mathcal{D}, \\ u(0) = u_0. \end{cases}$$

Known results for operator splitting schemes:

- Rigorous convergence analysis [Hansen & Henningsson, 2016], [E. & Hansen, 2018, 2022].

(B) What has been done

Deterministic nonlinear differential equation

$$\begin{cases} \dot{u} + A(t,u) = f(t,u), \quad t \in (0,T), \\ u(0) = u_0, \end{cases}$$

as for example the porous media equation

$$\begin{cases} \dot{u} = \Delta\Phi(u), \quad (t,x) \in (0,T) \times \mathcal{D}, \\ u(0) = u_0. \end{cases}$$

Goals:

- Rigorous analysis for moving domains.
- Randomized splittings.

(B) What has been done

Stochastic differential equation

$$\begin{cases} \mathrm{d}u + A(t,\omega,u)\,\mathrm{d}t = f(t,\omega,u)\,\mathrm{d}t + g(t,\omega,u)\,\mathrm{d}W(t), \ \ t \in (0,T), \\ u(0) = u_0, \end{cases}$$

as for example the stochastic porous media equation

$$\begin{cases} \mathrm{d}u + \Delta\Phi(u)\,\mathrm{d}t = g(t,\omega,u)\,\mathrm{d}W(t), \quad t \in (0,T), \\ u(0) = u_0. \end{cases}$$

(B) What has been done

Stochastic differential equation

$$\begin{cases} \mathrm{d}u + A(t,\omega,u)\,\mathrm{d}t = f(t,\omega,u)\,\mathrm{d}t + g(t,\omega,u)\,\mathrm{d}W(t), \; t \in (0,T), \\ u(0) = u_0, \end{cases}$$

as for example the stochastic porous media equation

$$\begin{cases} \mathrm{d}u + \Delta\Phi(u)\,\mathrm{d}t = g(t,\omega,u)\,\mathrm{d}W(t), \quad t \in (0,T), \\ u(0) = u_0. \end{cases}$$

Known results:

- Existence theory, for linear, semi-linear, quasi-linear equations. [Rozovskii, 1990], [Barbu Et al., 2017], ...
- Simple numerical schemes for linear/semi-linear equation [Gyoengy & Millet, 2009], [Brehier & Wang, 2020], ...

(B) What has been done

Stochastic differential equation

$$\begin{cases} \mathrm{d}u + A(t,\omega,u)\,\mathrm{d}t = f(t,\omega,u)\,\mathrm{d}t + g(t,\omega,u)\,\mathrm{d}W(t), & t \in (0,T), \\ u(0) = u_0, \end{cases}$$

as for example the stochastic porous media equation

$$\begin{cases} \mathrm{d}u + \Delta\Phi(u)\,\mathrm{d}t = g(t,\omega,u)\,\mathrm{d}W(t), \quad t \in (0,T), \\ u(0) = u_0. \end{cases}$$

Goals:

- More advanced numerical methods, as for example, domain decomposition schemes.
- Efficient implementation and relevant numerical examples.
- Work towards models from part (A).

Outline

(A) Development of a model for the assessment of the performance of groundwater remediation.

(B) Construction and analysis of modern time-stepping methods.

(C) Efficient implementation of the time-stepping methods for relevant problems.

(C) So far out numerical examples

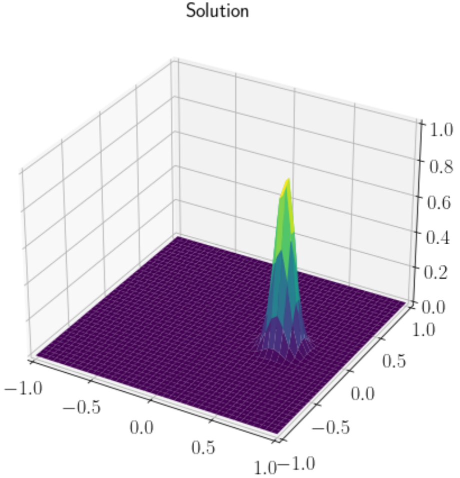
For $\mathcal{D} = (-1, 1) \times (-1, 1)$ and $\mathcal{D}_T = D \times (0, 1)$,

$$\begin{cases} \partial_t u(t,x) - \frac{1}{10}\Delta u(t,x) = f(t,x), & (t,x) \in \mathcal{D}_T \\ u(t,x) = 0, & (t,x) \in \mathcal{D}_T \\ u(0,x) = u_0(x) & x \in \mathcal{D}, \end{cases}$$

we choose the source term f such that the exact solution is given by

$$u(t,x,y) = e^{-100(x - r\cos(2\pi t))^2 - 100(y - r\sin(2\pi t))^2}.$$

(C) Visualization of the test problem



(C) Improvement

Good

- We can see (on small scale test equations) that our theory is working.
- The code is simple and it is easy to make changes.

Improvement needed:

- Test examples are not close enough to applications.
- Our implementation is not build to be efficient.

(C) DUNE

- DUNE, the **D**istributed and **U**nified **N**umerics **E**nvironment is a modular toolbox for solving partial differential equations.
- It a free software that supports ready to use methods like
  - Finite Elements,
  - Finite Volumes,
  - Finite Differences.
- Idea of DUNE: create slim interfaces allowing to use efficient methods that are based on modern C++ codes.
- DUNE ensures efficiency in scientific computations.

(C) Why DUNE?

We receive

- Existing code base that we can add our methods to.
- More relevant examples become accessible with DUNE.
- More efficient implementations than our naive implementation.
- Close (or decrease) gap to real world applications like (A).

Moreover

- Robert Klöfkorn (Lund University) is one of the developers of DUNE.
- Collaboration and support to fulfill goals.

(C) More complex nonlinear example

Euler equation in divergence form:

$$\partial_t U + \nabla \cdot \big(\mathcal{F}(U) - \mathcal{A}(U)\nabla U\big) = S(U) \quad \text{in } \Omega$$

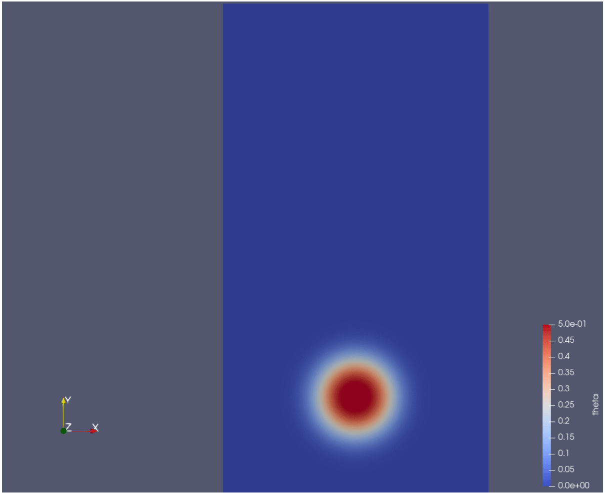
where the source term and the fluxes are

$$\mathcal{F}(U) = \begin{bmatrix} \rho u & \rho w \\ \rho u^2 + p & \rho u w \\ \rho u w & \rho w^2 + p \\ u\rho\theta & w\rho\theta \end{bmatrix},\ \mathcal{A}(U)\nabla U = \begin{bmatrix} 0 & 0 \\ \partial_x u & \partial_z u \\ \partial_x w & \partial_z w \\ \partial_x \theta & \partial_z \theta \end{bmatrix},\ \mathcal{S}(U) = \begin{bmatrix} 0 \\ 0 \\ -\rho g \\ 0 \end{bmatrix},$$

for $U = (\rho, \rho v^T, \rho\theta)$ and

- ρ being the density,
- θ the potential temperature,
- $v = (u, w)$ the velocity field.

(C) Visualization Euler's equation with DUNE



Special thanks to Johannes Kasimir for providing me with this animation and the relevant information!

Outline

(A) Development of a model for the assessment of the performance of groundwater remediation.

(B) Construction and analysis of modern time-stepping methods.

(C) Efficient implementation of the time-stepping methods for relevant problems.

Thank you for your attention!