Reservoir Simulation

AES1350 - Spring 2017

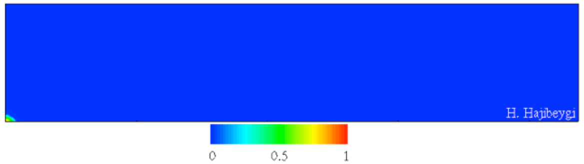
Lecture 1 - Introduction & Pressure Equation

Hadi Hajibeygi: h.hajibeygi@tudelft.nl

You took this class to:

- Explain why reservoir simulation is important.
- Develop single-phase flow (pressure) simulator.
- Develop 2-phase transport (saturation) simulator.
- Analyze stability & consistency of your simulator.
- Develop IMPES 2-phase simulator.

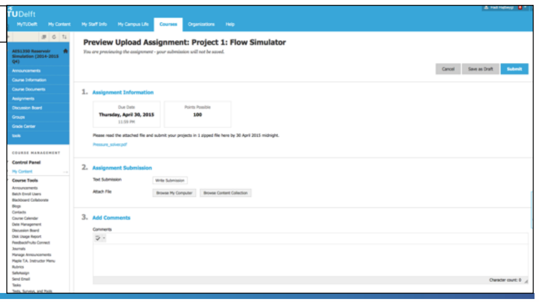
- *Develop: "discretize on paper", "implement on MATLAB/C++", and "analyze the results"!*



Assessments

- 3 **Projects** (Report + Simulator): 30+20+30 = **80%**
  - A scientific report contains:
    - Title, Author(s), Abstract, Introduction, Methodology, Results, Discussions References, Appendices.
- **Exam**: **If** (3 Proj > 50%), **20%; else** final grade = 0.
  - 2 hrs online/paper about the projects & the course, last lecture on **6 June**.

Submit to Blackboard



Assessments

- 3 **Projects** (Report + Simulator): 30+20+30 = **80%**
  - A scientific report contains:
    - Title, Author(s), Abstract, Introduction, Methodology, Results, Discussions References, Appendices.
- **Exam**: **If** (3 Proj > 50%), **20%; else** final grade = 0.
  - 2 hrs online/paper about the projects & the course, last lecture on **6 June**.

B ACTIVE - let us learn it now and forever!!!!

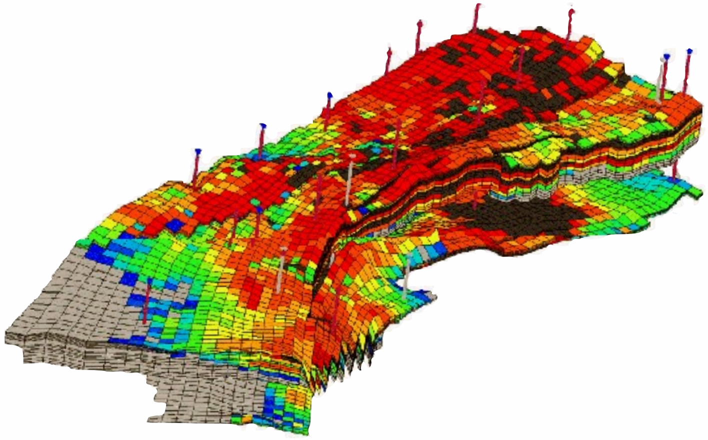
Part 1: Flow

- Explain why reservoir simulation is important.
- Develop single-phase flow (pressure) simulator.

3 Sessions

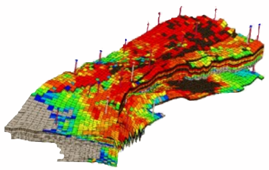
Why reservoir simulation?

What does a reservoir simulator do?



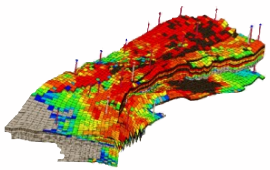
Why reservoir simulation?

- Input:
- Action:
- Analyze:



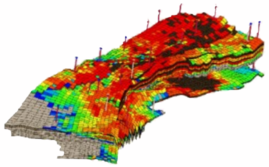
Why reservoir simulation?

- Input: geological data, experimental properties
- Action:
- Analyze:



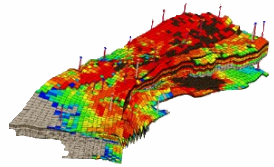
Why reservoir simulation?

- Input: geological data, experimental properties
- Action: discretize & solve
  - PDE's: flow, transport, energy, ...
  - Proper simplifications of physics & geometry
  - Apply *initial* and *boundary* conditions
  - accounting for *wells*
- Analyze:



Why reservoir simulation?

- Input: geological data, experimental properties
- Action: discretize & solve
  - PDE's: flow, transport, energy, ...
  - Proper simplifications of physics & geometry
  - Apply *initial* and *boundary* conditions
  - accounting for *wells*
- Analyze: the solution by determining
  - production (at each well and total) of the reservoir over time
  - zones of most and least extracted oil
  - production efficiency ...



Why reservoir simulation?

- Using a reservoir simulator, by changing the controllable parameters (injection fluids, injection/production well conditions, ...)

  we better manage the resource and finally

  optimize our "overall" profit!

Why reservoir simulation?

- Using a reservoir simulator, by changing the controllable parameters (injection fluids, injection/production well conditions, ...)

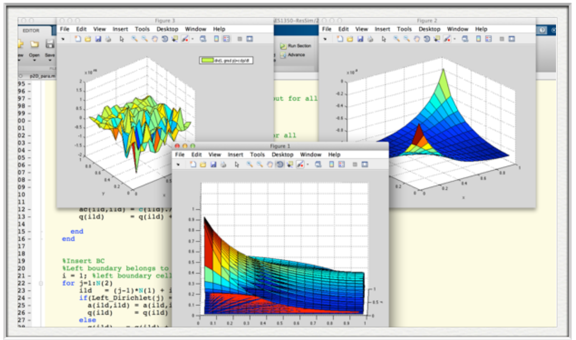
  we better manage the resource and finally

  optimize our "overall" profit!

Discuss in the class:

- Reservoir simulator is a powerful tool, yet not the goal!
- We have not found a better tool ...
  analytical, experimental, numerical, ...?

Flow simulator



Flow simulator, Learning Objectives:

- **Formulate** incompressible & compressible pressure Eq.
- **Explain its main computational challenges**
- **Discretize & analyze**
  stability, accuracy, ... of your discretization scheme
- **Develop** a 2D Flow Simulator, analyze results
- **Report** due **15 May 2017** **(30 %)**

Formulate Pressure (Flow) Equation

Q: Which Eq’s describe fluid flow in porous media?

- Pressure Eq. is derived as following:

Formulate Pressure (Flow) Equation

Q: Which Eq’s describe fluid flow in porous media?

A: Conservation laws, i.e.,

- Mass
- Instead of Momentum, we use Darcy’s law
- Energy (throughout this course: isothermal!),

and constitutive relations from physics, thermo., etc. (PVT, ...)

- Pressure Eq. is derived as following:

Formulate Pressure (Flow) Equation

Q: Which Eq's describe fluid flow in porous media?

A: Conservation laws, i.e.,

- Mass
- Instead of Momentum, we use Darcy's law
- Energy (throughout this course: isothermal!),

and constitutive relations from physics, thermo., etc. (PVT, ...)

- Pressure Eq. is derived as following:
  start with "Mass Conservation", substitute Darcy's law, do math ... and you will get it!

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $\frac{\partial}{\partial t}\left(\frac{m_\alpha}{V}\right) + \nabla \cdot (\rho_\alpha \quad u_\alpha \quad) = \rho_\alpha q_\alpha$

m_α

q_α

u_α

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) + \nabla \cdot (\rho_\alpha \quad u_\alpha \quad) = \rho_\alpha q_\alpha$

ρ_α

S_α

ϕ

q_α

u_α

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) - \nabla \cdot (\rho_\alpha \underline{\underline{\lambda}}_\alpha \cdot \nabla p) = \rho_\alpha q_\alpha$

ρ_α

S_α

ϕ

$$\underline{\underline{\lambda}}_\alpha = \frac{k_{r\alpha}}{\mu_\alpha} K$$

q_α

$$u_\alpha = -\underline{\underline{\lambda}}_\alpha \cdot \nabla p$$

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $$\frac{\partial}{\partial t}(\phi \cancel{\rho_\alpha} S_\alpha) - \nabla\cdot(\cancel{\rho_\alpha}\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = \cancel{\rho_\alpha} q_\alpha$$

$$\downarrow$$

Incompressible: $$\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha$$

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $$\frac{\partial}{\partial t}(\phi \cancel{\rho_\alpha} S_\alpha) - \nabla\cdot(\cancel{\rho_\alpha}\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = \cancel{\rho_\alpha} q_\alpha$$

$$\downarrow$$

Incompressible: $$\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha$$

$$\downarrow$$

Total balance: $$\sum_{\alpha=1}^{n_{ph}}\left\{\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha\right\}$$

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $$\frac{\partial}{\partial t}(\phi \cancel{\rho_\alpha} S_\alpha) - \nabla\cdot(\cancel{\rho_\alpha}\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = \cancel{\rho_\alpha} q_\alpha$$

$$\downarrow$$

Incompressible: $$\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha$$

$$\downarrow$$

Total balance: $$\sum_{\alpha=1}^{n_{ph}}\left\{\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha\right\}$$

$$\downarrow$$

$$\phi\frac{\partial}{\partial t}\sum_{\alpha=1}^{n_{ph}} S_\alpha - \nabla\cdot\left(\sum_{\alpha=1}^{n_{ph}}\underline{\underline{\lambda}}_\alpha\cdot\nabla p\right) = \sum_{\alpha=1}^{n_{ph}} q_\alpha$$

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $$\frac{\partial}{\partial t}(\phi \cancel{\rho_\alpha} S_\alpha) - \nabla\cdot(\cancel{\rho_\alpha}\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = \cancel{\rho_\alpha} q_\alpha$$

$$\downarrow$$

Incompressible: $$\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha$$

$$\downarrow$$

Total balance: $$\sum_{\alpha=1}^{n_{ph}}\left\{\phi\frac{\partial S_\alpha}{\partial t} - \nabla\cdot(\underline{\underline{\lambda}}_\alpha\cdot\nabla p) = q_\alpha\right\}$$

$$\downarrow$$

$$\boxed{\phi\frac{\partial}{\partial t}\underbrace{\sum_{\alpha=1}^{n_{ph}} S_\alpha}_{=1}} - \nabla\cdot\left(\underbrace{\sum_{\alpha=1}^{n_{ph}}\underline{\underline{\lambda}}_\alpha}_{\underline{\underline{\lambda}}_t}\cdot\nabla p\right) = \underbrace{\sum_{\alpha=1}^{n_{ph}} q_\alpha}_{q}$$

$$\boxed{\phi\frac{\partial 1}{\partial t} = 0}$$

Formulate Pressure (Flow) Equation

- "Mass conservation Eq. for component "i" in phase "α":

Mass conservation: $$\frac{\partial}{\partial t}(\phi \cancel{\rho_\alpha} S_\alpha) - \nabla \cdot (\cancel{\rho_\alpha} \underline{\underline{\lambda}}_\alpha \cdot \nabla p) = \cancel{\rho_\alpha} q_\alpha$$

$$\downarrow$$

Incompressible: $$\phi \frac{\partial S_\alpha}{\partial t} - \nabla \cdot (\underline{\underline{\lambda}}_\alpha \cdot \nabla p) = q_\alpha$$

$$\downarrow$$

Total balance: $$\sum_{\alpha=1}^{n_{ph}} \left\{ \phi \frac{\partial S_\alpha}{\partial t} - \nabla \cdot (\underline{\underline{\lambda}}_\alpha \cdot \nabla p) = q_\alpha \right\}$$

$$\downarrow$$

Incompressible pressure equation: $$\boxed{-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q}$$

Formulate Pressure (Flow) Equation

- Compressible pressure equation for single-phase flow:

Mass conservation: $$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) - \nabla \cdot (\rho_\alpha \underline{\underline{\lambda}}_\alpha \cdot \nabla p) = \rho_\alpha q_\alpha$$

Formulate Pressure (Flow) Equation

- Compressible pressure equation for single-phase flow:

Mass conservation: $$\frac{\partial}{\partial t}(\phi \rho_\alpha \underbrace{\cancel{S_\alpha}}_{1}) - \nabla \cdot (\rho_\alpha \underbrace{\cancel{\underline{\underline{\lambda}}_\alpha}}_{=\frac{K}{\mu}} \cdot \nabla p) = \rho_\alpha q_\alpha$$

$$\downarrow$$

Single-phase pressure equation: $$\boxed{\frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p) = \rho q}$$

In volume formation factor way (black-oil): $\rho = \rho^{STC}/B_f$ and $\rho^{STC} = \text{cte.}$

$$\Rightarrow \boxed{\frac{\partial}{\partial t}(\frac{\phi}{B_f}) - \nabla \cdot (\frac{K}{\mu B_f} \cdot \nabla p) = \frac{q}{B_f}}$$

or, after some math $\frac{\partial}{\partial t}(\phi\rho) = \frac{\partial(\phi\rho)}{\partial p}\frac{\partial p}{\partial t} = (\phi\rho)\frac{1}{\phi\rho}\frac{\partial(\phi\rho)}{\partial p}\frac{\partial p}{\partial t} = (\phi\rho)\underbrace{(c_f + c_\phi)}_{c_{\text{eff}}}\frac{\partial p}{\partial t}$

$$\Rightarrow \boxed{(\phi\rho) c_{\text{eff}} \frac{\partial p}{\partial t} - \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p) = \rho q}$$

Formulate Pressure (Flow) Equation

- Pressure equations:

Incompressible single/multiphase pressure equation:

$$\boxed{-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q}$$

Compressible single-phase pressure equations:

General form: $$\boxed{\frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p) = \rho q}$$

General form-BO: $$\boxed{\frac{\partial}{\partial t}(\frac{\phi}{B_f}) - \nabla \cdot (\frac{K}{\mu B_f} \cdot \nabla p) = \frac{q}{B_f}}$$

After some math: $$\boxed{(\phi\rho) c_{\text{eff}} \frac{\partial p}{\partial t} - \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p) = \rho q}$$

Compressible multi-phase pressure equation: to come later! :-)

Formulate Pressure (Flow) Equation

- Pressure equations:

Incompressible single/multiphase pressure equation:

$$\boxed{-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q}$$

Compressible single-phase pressure equation:

$$\boxed{(\phi\rho)c_{\text{eff}}\frac{\partial p}{\partial t} - \nabla \cdot (\rho\frac{K}{\mu} \cdot \nabla p) = \rho q}$$

What if density does not change much in spacial coordinate?

Formulate Pressure (Flow) Equation

- Pressure equations:

Incompressible single/multiphase pressure equation:

$$\boxed{-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q}$$

Compressible single-phase pressure equation:

$$\boxed{(\phi\rho)c_{\text{eff}}\frac{\partial p}{\partial t} - \nabla \cdot (\rho\frac{K}{\mu} \cdot \nabla p) = \rho q}$$

Slightly compressible single-phase pressure equation:

$$\boxed{\phi c_{\text{eff}}\frac{\partial p}{\partial t} - \nabla \cdot (\frac{K}{\mu} \cdot \nabla p) = q}$$

Formulate Pressure (Flow) Equation

- Pressure equations:

Incompressible single/multiphase pressure equation:

$$\boxed{-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q}$$

Compressible single-phase pressure equation:

$$\boxed{\frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot (\rho\frac{K}{\mu} \cdot \nabla p) = \rho q}$$

Slightly compressible single-phase pressure equation:

$$\boxed{\phi c_{\text{eff}}\frac{\partial p}{\partial t} - \nabla \cdot (\frac{K}{\mu} \cdot \nabla p) = q}$$

Explain the Challenges of Pressure (Flow) Equation

- Pressure equations:

Incompressible single/multiphase pressure equation:

$$\boxed{-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q}$$

Compressible single-phase pressure equation:

$$\boxed{\frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot (\rho\frac{K}{\mu} \cdot \nabla p) = \rho q}$$

Slightly compressible single-phase pressure equation:

$$\boxed{\phi c_{\text{eff}}\frac{\partial p}{\partial t} - \nabla \cdot (\frac{K}{\mu} \cdot \nabla p) = q}$$

Discuss:
Which PDE's are they?
What are the challenges in solving them?
How do we solve them?

Discretization of $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

Discretization of incompressible P Eq. $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

- Incompressible Pressure equation:

$$\nabla \cdot (u_t) = q \quad \Rightarrow \quad -\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$$

- We use FV method (why?)
  FD method with smart location of P & U works as well.

Discretization of incompressible P Eq. $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

- Incompressible Pressure equation:

$$\nabla \cdot (u_t) = q \quad \Rightarrow \quad -\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$$

- We use FV method (why?)
  FD method with smart location of P & U works as well.

- 1-dimensional Cartesian system: $-\frac{\partial}{\partial x}(\underline{\underline{\lambda}}_{t,x} \frac{\partial p}{\partial x}) = q$

$$\Downarrow$$

$$-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$$

Discretization of incompressible P Eq. $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

- Incompressible Pressure equation:

$$\nabla \cdot (u_t) = q \quad \Rightarrow \quad -\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$$

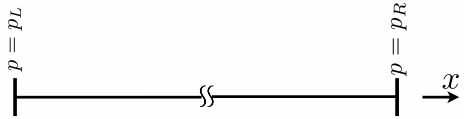
- We use FV method (why?)
  FD method with smart location of P & U works as well.

- 1-dimensional Cartesian system: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

- Let us solve it for a heterogeneous reservoir.

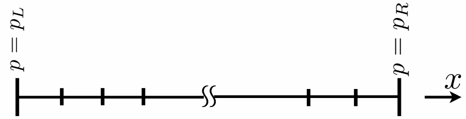
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



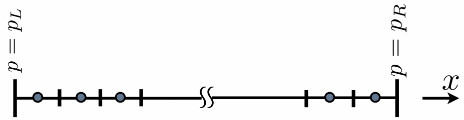
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



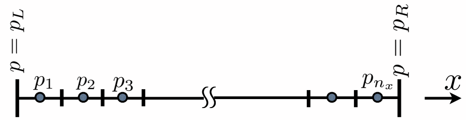
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



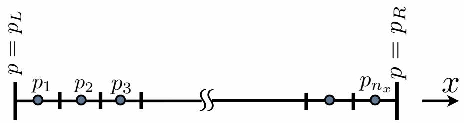
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

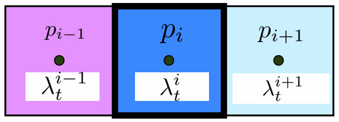
Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

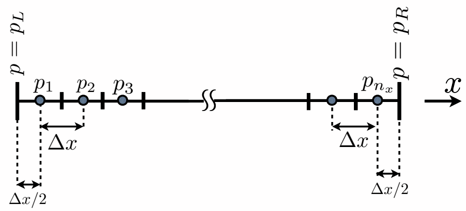
Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

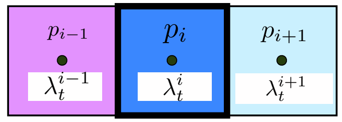




Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

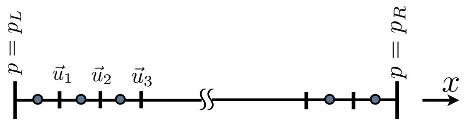
Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

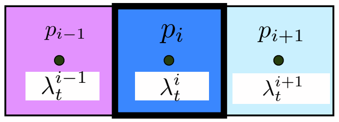




Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

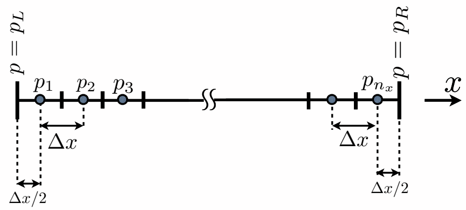
Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach





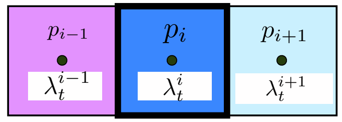
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



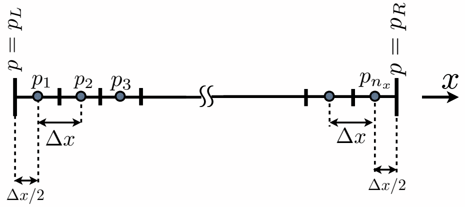
For cell i:

$$\Rightarrow \frac{(-\lambda \frac{dp}{dx})^{i+1/2} - (-\lambda \frac{dp}{dx})^{i-1/2}}{\Delta x} = q_i$$



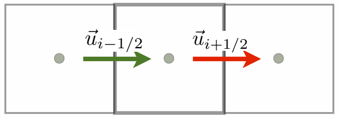
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



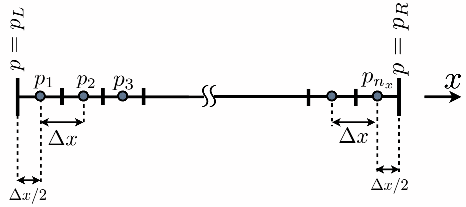
For cell i:

$$\Rightarrow \frac{(-\lambda \frac{dp}{dx})^{i+1/2} - (-\lambda \frac{dp}{dx})^{i-1/2}}{\Delta x} = q_i$$



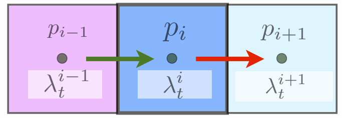
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



For cell i:

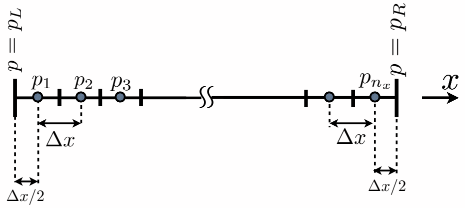
$$\Rightarrow \frac{(-\lambda \frac{dp}{dx})^{i+1/2} - (-\lambda \frac{dp}{dx})^{i-1/2}}{\Delta x} = q_i$$



$$\Rightarrow (-\lambda \frac{dp}{dx})^{i+1/2} \approx -\lambda_t^{i+1/2} \; \frac{p_{i+1} - p_i}{\Delta x}$$

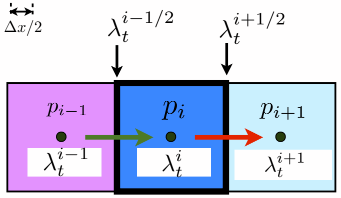
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



For cell i:

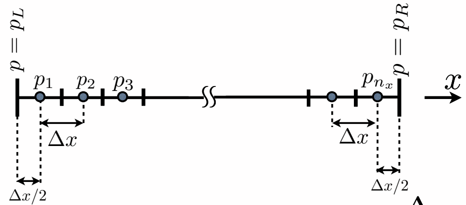
$$\Rightarrow \frac{(-\lambda \frac{dp}{dx})^{i+1/2} - (-\lambda \frac{dp}{dx})^{i-1/2}}{\Delta x} = q_i$$



$$\Rightarrow (-\lambda \frac{dp}{dx})^{i+1/2} \approx -\lambda_t^{i+1/2} \; \frac{p_{i+1} - p_i}{\Delta x}$$

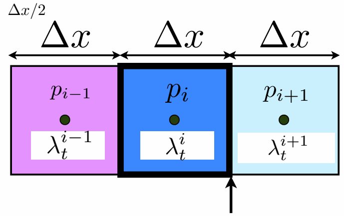
Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



For cell i:

$$\Rightarrow \frac{(-\lambda \frac{dp}{dx})^{i+1/2} - (-\lambda \frac{dp}{dx})^{i-1/2}}{\Delta x} = q_i$$

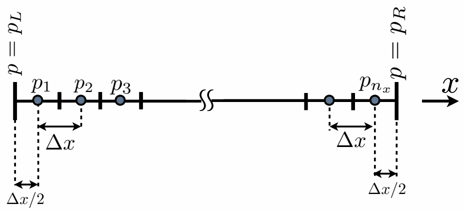


$$\Rightarrow (-\lambda \frac{dp}{dx})^{i+1/2} \approx -\lambda_t^{i+1/2} \; \frac{p_{i+1} - p_i}{\Delta x} \qquad \lambda_t^{i+1/2} = 2\frac{\lambda_t^i \lambda_t^{i+1}}{\lambda_t^i + \lambda_t^{i+1}}$$

consider single-phase flow, where $\lambda_t = k/\mu$ with $\mu = cte.$

Discretization of: $-\dfrac{\partial}{\partial x}(\lambda \dfrac{\partial p}{\partial x}) = q$

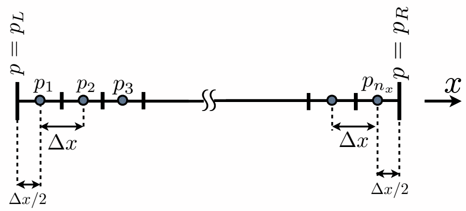
Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



For Cell i: $\dfrac{\lambda_t^{i-1/2}}{\Delta x^2}(p_i - p_{i-1}) + \dfrac{\lambda_t^{i+1/2}}{\Delta x^2}(p_i - p_{i+1}) = q_i$

Discretization of: $-\dfrac{\partial}{\partial x}(\lambda \dfrac{\partial p}{\partial x}) = q$

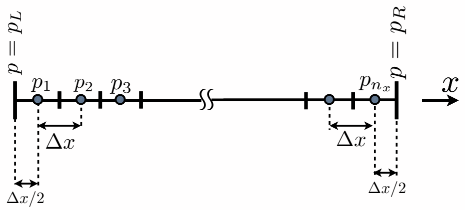
Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



For Cell i: $\overset{T_{i-1/2}}{\boxed{\dfrac{\lambda_t^{i-1/2}}{\Delta x^2}}}(p_i - p_{i-1}) + \overset{T_{i+1/2}}{\boxed{\dfrac{\lambda_t^{i+1/2}}{\Delta x^2}}}(p_i - p_{i+1}) = q_i$

Discretization of: $-\dfrac{\partial}{\partial x}(\lambda \dfrac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

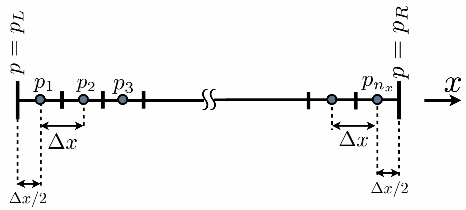


For Cell 1: $\boxed{?}(\boxed{?}) + \dfrac{\lambda_t^{1+1/2}}{\Delta x^2}(p_1 - p_2) = q_1$

For Cell i: $\dfrac{\lambda_t^{i-1/2}}{\Delta x^2}(p_i - p_{i-1}) + \dfrac{\lambda_t^{i+1/2}}{\Delta x^2}(p_i - p_{i+1}) = q_i$

Discretization of: $-\dfrac{\partial}{\partial x}(\lambda \dfrac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

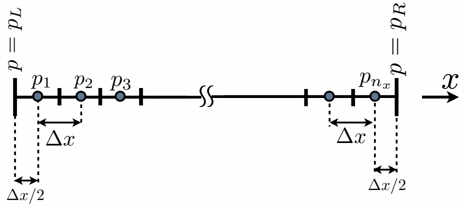


For Cell 1: $\boxed{?}(p_1 - p_L) + \dfrac{\lambda_t^{1+1/2}}{\Delta x^2}(p_1 - p_2) = q_1$

For Cell i: $\dfrac{\lambda_t^{i-1/2}}{\Delta x^2}(p_i - p_{i-1}) + \dfrac{\lambda_t^{i+1/2}}{\Delta x^2}(p_i - p_{i+1}) = q_i$

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

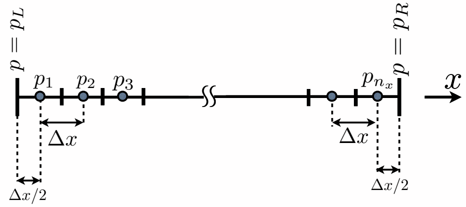


For Cell 1: $$\frac{\lambda_t^1}{\Delta x^2/2}(p_1 - p_L) + \frac{\lambda_t^{1+1/2}}{\Delta x^2}(p_1 - p_2) = q_1$$

For Cell i: $$\frac{\lambda_t^{i-1/2}}{\Delta x^2}(p_i - p_{i-1}) + \frac{\lambda_t^{i+1/2}}{\Delta x^2}(p_i - p_{i+1}) = q_i$$

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



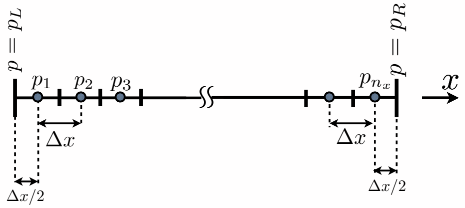
For Cell 1: $$\frac{\lambda_t^1}{\Delta x^2/2}(p_1 - p_L) + \frac{\lambda_t^{1+1/2}}{\Delta x^2}(p_1 - p_2) = q_1$$

For Cell i: $$\frac{\lambda_t^{i-1/2}}{\Delta x^2}(p_i - p_{i-1}) + \frac{\lambda_t^{i+1/2}}{\Delta x^2}(p_i - p_{i+1}) = q_i$$

For Cell n_x: $$\frac{\lambda_t^{n_x-1/2}}{\Delta x^2}(p_{n_x} - p_{n_x-1}) + \frac{\lambda_t^{n_x}}{\Delta x^2/2}(p_{n_x} - p_R) = q_{n_x}$$

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



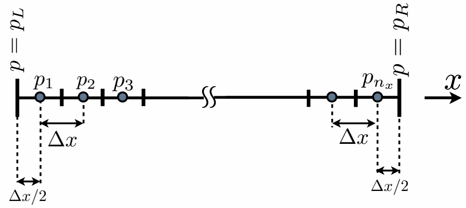
For Cell 1: $$\frac{\lambda_t^1}{\Delta x^2/2}(p_1 - p_L) + \frac{\lambda_t^{1+1/2}}{\Delta x^2}(p_1 - p_2) = q_1$$

For Cell i: $$\overset{T_{i-1/2}}{\boxed{\frac{\lambda_t^{i-1/2}}{\Delta x^2}}}(p_i - p_{i-1}) + \overset{T_{i+1/2}}{\boxed{\frac{\lambda_t^{i+1/2}}{\Delta x^2}}}(p_i - p_{i+1}) = q_i$$

For Cell n_x: $$\frac{\lambda_t^{n_x-1/2}}{\Delta x^2}(p_{n_x} - p_{n_x-1}) + \frac{\lambda_t^{n_x}}{\Delta x^2/2}(p_{n_x} - p_R) = q_{n_x}$$

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



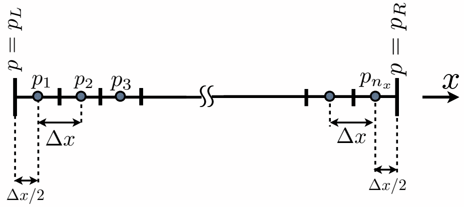
Mathematically:

$$T_{i-1/2}(p_i - p_{i-1}) + T_{i+1/2}(p_i - p_{i+1}) = q_i$$

Between (i) and (i-1):	$T_{i-1/2} = \frac{\lambda_t^{i-1/2}}{\Delta x^2}$
Between (1) and (L) Boundary:	$T_{1-1/2} = \frac{\lambda_t^{1}}{\Delta x^2/2}$
Between (n) and (R) Boundary:	$T_{n_x+1/2} = \frac{\lambda_t^{n_x}}{\Delta x^2/2}$

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

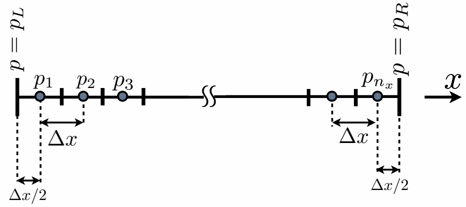


In MATLAB:

```
T(?) * (p(i) - P(i-1)) + T( ? ) * (p(i) - p(i+1)) = q(i)
```


Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach

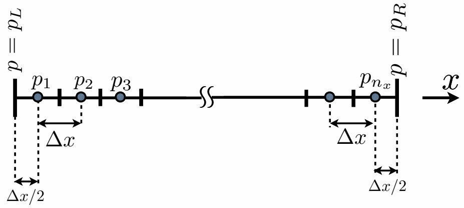


In MATLAB:

```
T(i) * (p(i) - P(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)
```


Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



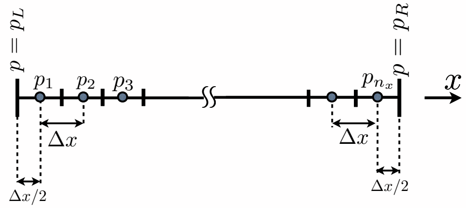
In MATLAB:

```
T(i) * (p(i) - P(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)
```

Between (i) and (i-1): `T(i) = LH(?) / DX^2;`

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



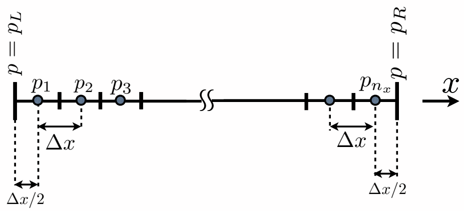
In MATLAB:

```
T(i) * (p(i) - P(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)
```

Between (i) and (i-1): `T(i) = LH(i) / DX^2;`

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FD approach



In MATLAB:

```
T(i) * (p(i) - P(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)
```

Between (i) and (i-1):	`T(i) = LH(i) / DX^2;`
Between (1) and (L) Boundary:	`T(1) = LH(1) / (DX^2/2);`
Between (n) and (R) Boundary:	`T(N+1) = LH(N+1) / (DX^2/2);`

Discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

- So far we used Finite-Difference (FD) scheme.

- Please study Finite-Volume (FV) scheme (next 3 pages).

- Results for our Structured Cartesian grids are the same.

Discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

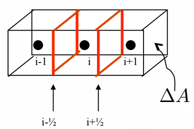
FV scheme: $\int_v \nabla \cdot (\bullet) dv \to \int_{\partial v} (\bullet) \cdot \vec{n} ds$ **theorem is used. It ensures conservation.**

$$-\int_V \nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p)\ dv = \int_V q\ dv \ \Rightarrow -\int_{\partial V} (\underline{\underline{\lambda}}_t \cdot \nabla p) \cdot \vec{n}\ ds = \int_V q\ dv$$

For our example, we start like this: (dv = dx dy dz)

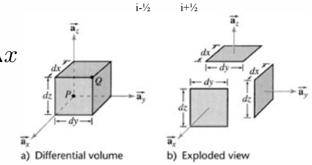
$$-\int_V \frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) dV = \int_V q dV \ \Rightarrow -\int_S (\lambda \frac{\partial p}{\partial x}) \cdot \vec{n}\ dS = q \Delta A\ \Delta x$$

$$\Rightarrow \Delta A\ (-\lambda \frac{dp}{dx})^{i+1/2} - \Delta A\ (-\lambda \frac{dp}{dx})^{i-1/2} = q_i \Delta A\ \Delta x$$



For Cell i:

$$\frac{\Delta A\ \lambda_t^{i-1/2}}{\Delta x}(p_i - p_{i-1}) + \frac{\Delta A\ \lambda_t^{i+1/2}}{\Delta x}(p_i - p_{i+1}) = q_i \Delta A \Delta x$$



Is not it the same as FD with our choice of U and P notes?

Discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

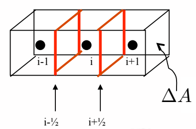
FV scheme: $\int_v \nabla \cdot (\bullet) dv \to \int_{\partial v} (\bullet) \cdot \vec{n} ds$ **theorem is used. It ensures conservation.**

$$-\int_V \nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p)\ dv = \int_V q\ dv \ \Rightarrow -\int_{\partial V} (\underline{\underline{\lambda}}_t \cdot \nabla p) \cdot \vec{n}\ ds = \int_V q\ dv$$

For our example, we start like this: (dv = dx dy dz)

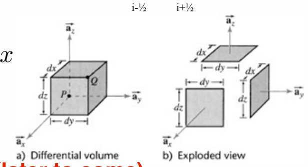
$$-\int_V \frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) dV = \int_V q dV \ \Rightarrow -\int_S (\lambda \frac{\partial p}{\partial x}) \cdot \vec{n}\ dS = q \Delta A\ \Delta x$$

$$\Rightarrow \Delta A\ (-\lambda \frac{dp}{dx})^{i+1/2} - \Delta A\ (-\lambda \frac{dp}{dx})^{i-1/2} = q_i \Delta A\ \Delta x$$



For Cell i:

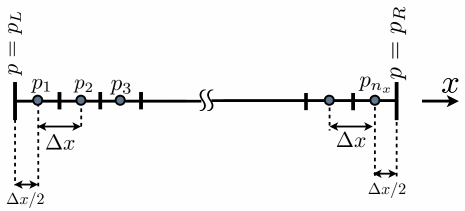
$$\underbrace{\boxed{\frac{\Delta A\ \lambda_t^{i-1/2}}{\Delta x}}}_{T_{i-1/2}}(p_i - p_{i-1}) + \underbrace{\boxed{\frac{\Delta A\ \lambda_t^{i+1/2}}{\Delta x}}}_{T_{i+1/2}}(p_i - p_{i+1}) = q_i \Delta A \Delta x$$



Transmissibility Coefficient (it has fluid part and rock part).
Rock part (k) will be harmonic averaged, fluid-part upwind (later to come)

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition: FV approach



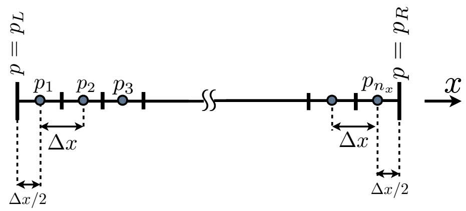
For Cell 1: $\frac{\lambda_t^1 \Delta A}{\Delta x/2}(p_1 - p_L) + \frac{\lambda_t^{1+1/2}\Delta A}{\Delta x}(p_1 - p_2) = q_1 \Delta V$

For Cell i: $\underbrace{\boxed{\frac{\lambda_t^{i-1/2}\Delta A}{\Delta x}}}_{}^{T_{i-1/2}}(p_i - p_{i-1}) + \boxed{\frac{\lambda_t^{i+1/2}\Delta A}{\Delta x}}^{T_{i+1/2}}(p_i - p_{i+1}) = q_i \Delta V$

For Cell n_x: $\frac{\lambda_t^{n_x-1/2}\Delta A}{\Delta x}(p_{n_x} - p_{n_x-1}) + \frac{\lambda_t^{n_x}\Delta A}{\Delta x/2}(p_{n_x} - p_R) = q_{n_x}\Delta V$

Discretization of: $-\frac{\partial}{\partial x}(\lambda \frac{\partial p}{\partial x}) = q$

Example: 1 Dimensional Case with Dirichlet Boundary Condition



Finite-Volume:

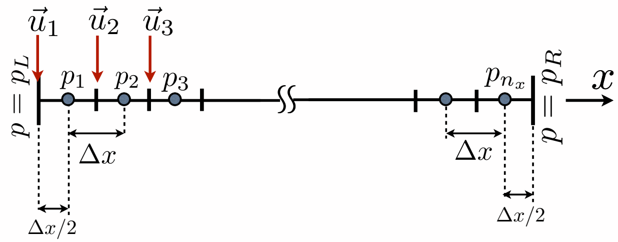
$T_{i-1/2} = \frac{\lambda_t^{i-1/2}\Delta A}{\Delta x}$ $\quad \boxed{T_{i-1/2}(p_i - p_{i-1}) + T_{i+1/2}(p_i - p_{i+1}) = q_i \Delta V}$

Finite-Difference:

$T_{i-1/2} = \frac{\lambda_t^{i-1/2}}{\Delta x^2}$ $\quad \boxed{T_{i-1/2}(p_i - p_{i-1}) + T_{i+1/2}(p_i - p_{i+1}) = q_i}$

MATLAB CODE: Some tips:

- Be careful with ordering of interfaces and grid cells.



MATLAB CODE: Some tips:

- Write discrete eq. for cell 'i'

$$T_{i-1/2}(p_i - p_{i-1}) + T_{i+1/2}(p_i - p_{i+1}) = q_i$$

in matlab ordering, i.e.:

```
T(i) * (p(i) - p(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)
```

MATLAB CODE: Some tips:

- Apply boundary condition carefully!

For cell i = 1:

T(i) * (p(i) - PL) + T(i+1) * (p(i) - p(i+1)) = q(i)

T(i) * p(i) + T(i+1) * (p(i) - p(i+1)) = q(i) + T(i) * PL

For cell i = N:

T(i) * (p(i) - P(i-1)) + T(i+1) * (p(i) - PR) = q(i)

T(i) * (p(i) - P(i-1)) + T(i+1) * p(i) = q(i) + T(i+1) * PR

1D Incompressible P Solver in MATLAB

- **INPUT** parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

- Construct & Fill linear system: Ap = q
- Apply boundary conditions (for first and last cells, e.g.)

- Solve for p: (in matlab: p = A\q)
- **OUTPUT** (print out) results

1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

For Example, our MATLAB code may start as:

1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

For Example, our MATLAB code may start as:

```
L  = 1.0;          % Length [m]
N  = 100;          % Number of grid cells
DX = L / N;        % Dx [m]
x  = linspace(Dx/2, L - Dx/2, N); %Grid centers
xi = linspace(0, L, N + 1);  %inteface locations
PL = ...;         % Left Pressure BC [Pa]
PR = ...;         % Right Pressure BC[Pa]
Lambda = zeros(N,1);   % Initialize Lambda, filled later
```

1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

For Example, our MATLAB code may start as:

```
L  = 1.0;          % Length [m]
N  = 100;          % Number of grid cells
DX = L / N;        % Dx [m]
x  = linspace(Dx/2, L - Dx/2, N); %Grid centers
xi = linspace(0, L, N + 1);  %inteface locations
PL = ...;          % Left Pressure BC [Pa]
PR = ...;          % Right Pressure BC[Pa]

Lambda = zeros(N,1);   % Initialize Lambda, filled later
LambdaH= zeros(N+1,1); % Initialize Lambda Avg., calc. later
T      = zeros (N+1,1);% Transmissibility at interfaces
```


1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

For Example, our MATLAB code may start as:

```
L  = 1.0;          % Length [m]
N  = 100;          % Number of grid cells
DX = L / N;        % Dx [m]
x  = linspace(Dx/2, L - Dx/2, N); %Grid centers
xi = linspace(0, L, N + 1);  %inteface locations
PL = ...;          % Left Pressure BC [Pa]
PR = ...;          % Right Pressure BC[Pa]

Lambda = zeros(N,1);   % Initialize Lambda, filled later
LambdaH= zeros(N+1,1); % Initialize Lambda Avg., calc. later
T      = zeros (N+1,1);% Transmissibility at interfaces
```

```
% fill in your lambda!
for i=1:N, lambda(i) = 1 ; end
```


1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

For Example, our MATLAB code may start as:

```
L  = 1.0;          % Length [m]
N  = 100;          % Number of grid cells
DX = L / N;        % Dx [m]
x  = linspace(Dx/2, L - Dx/2, N); %Grid centers
xi = linspace(0, L, N + 1);  %inteface locations
PL = ...;          % Left Pressure BC [Pa]
PR = ...;          % Right Pressure BC[Pa]

Lambda = zeros(N,1);   % Initialize Lambda, filled later
LambdaH= zeros(N+1,1); % Initialize Lambda Avg., calc. later
T      = zeros (N+1,1);% Transmissibility at interfaces
```

```
lambdaH(1)   = lambda(1);
lambdaH(N+1) = lambda(N);
lambdaH(2:N)=2*lambda(2:N).*lambda(1:N-1)./(lambda(2:N)+lambda(1:N-1));
```


1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

For Example, our MATLAB code may start as:

```
L  = 1.0;          % Length [m]
N  = 100;          % Number of grid cells
DX = L / N;        % Dx [m]
x  = linspace(Dx/2, L - Dx/2, N); %Grid centers
xi = linspace(0, L, N + 1);  %inteface locations
PL = ...;          % Left Pressure BC [Pa]
PR = ...;          % Right Pressure BC[Pa]

Lambda = zeros(N,1);   % Initialize Lambda, filled later
LambdaH= zeros(N+1,1); % Initialize Lambda Avg., calc. later
T      = zeros (N+1,1);% Transmissibility at interfaces
```

```
T(1)   = lambdaH(1)   ./ (Dx^2/2);
T(N+1) = lambdaH(N+1) ./ (Dx^2/2);
T(2:N) = lambdaH(2:N) ./ Dx^2;
```

1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

- Construct & Fill linear system: Ap = q

```
p = zeros(N,1);   % construct pressure vector of size N
q = zeros(N,1);   % construct source RHS vector of size N
A = zeros(N,N);   % construct A matrix of size NxN
```

```
for i=1:N,
    q(i) = 0; %if no sources exist (we can add manually e.g.)
end
```

```
for i=1:N,      % For All Cells
 if(i > 1),     % Except for cell 1, T(i)* (p(i) - p(i-1))
     A(?,?) = ?
     A(?,?) = ?
 end
 if(i < N),     % Except for cell N, T(i+1)*(p(i) - p(i+1))
     A(?,?) = ?
     A(?,?) = ?
 end
end
```


1D Incompressible P Solver in MATLAB

- INPUT parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

- Construct & Fill linear system: Ap = q
- Apply boundary conditions (for first and last cells, e.g.)

```
%Left boundary belongs to cell 1
i = 1; %left boundary cell
if(Left_Dirichlet == true)
 A(i,i) = A(i,i) + T(i);
 q(i)   = q(i) + T(i) * pL;
else
 q(i)   = q(i) + qL;
end
```

```
%Right boundary belongs to cell N
i = N; %left boundary cell
if(Right_Dirichlet == true)
  A(i,i) = A(i,i) + T(i+1);
  q(i)   = q(i) + T(i+1) * pR;
  else
    q(i)   = q(i) + qL;
end
```


1D Incompressible P Solver in MATLAB

- **INPUT** parameters (size, #grids, ...) / be able to access them
- Calculate quantities: T (transmissibility), ...

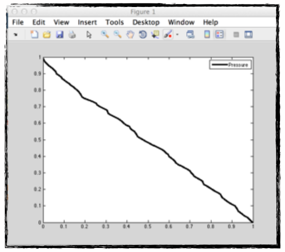
- Construct & Fill linear system: Ap = q
- Apply boundary conditions (for first and last cells, e.g.)

- Solve for p: (in matlab: p = A\q)
- **OUTPUT** (print out) results

```
p = A\q;
```

```
% Plot the solution :-)
plot (x,p, 'LineWidth',2, 'color',[0 0 0], ...
     'LineStyle','*','LineWidth',1.5,'DisplayName','FS'); %for line: '-'
legend show;
xlabel('x');
ylabel('Pressure');
```


1D Incompressible P Solver in MATLAB

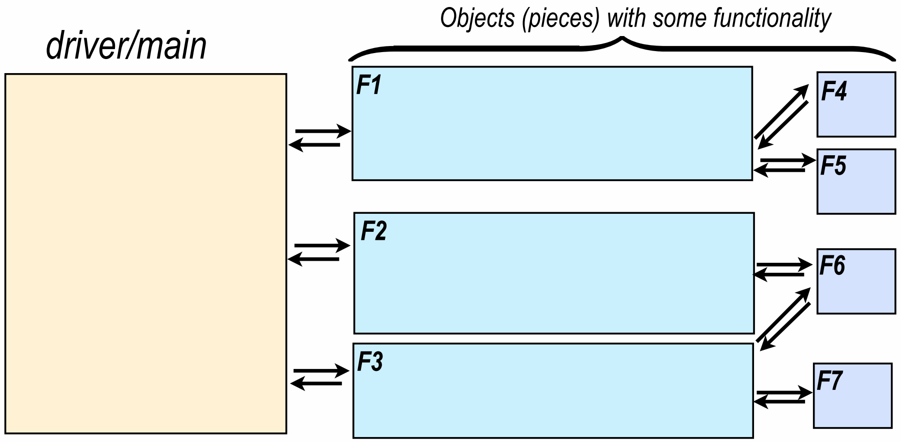


You will then find some solution for pressure!!! wow!

... très bien...!!!

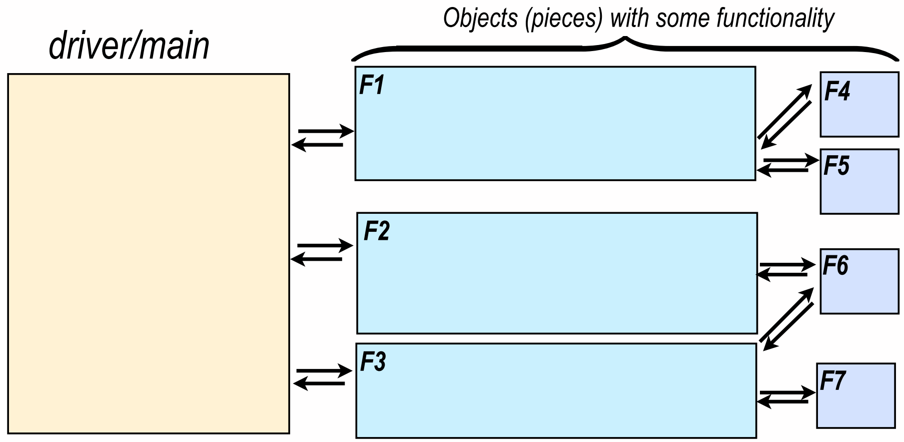
Object-Oriented Implementation: 1st step

"Divide (into manageable pieces) & Conquer"



Object-Oriented Implementation: 1st step

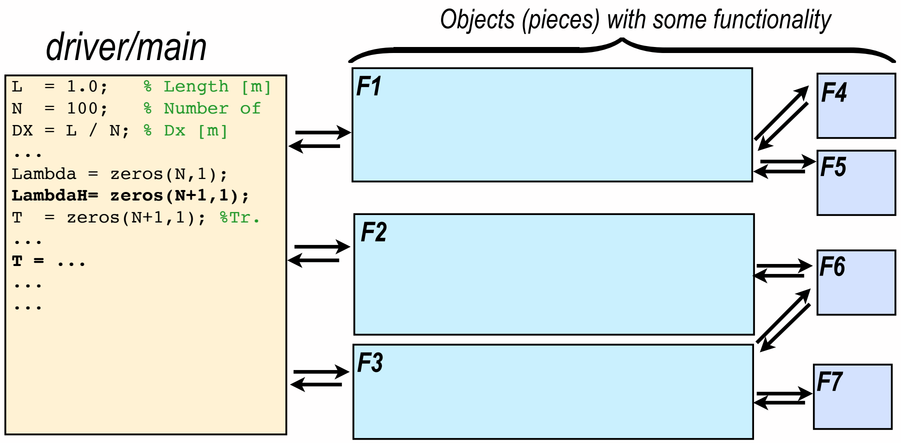
"Divide (into manageable pieces) & Conquer"



Let us Calculate T in our program with this approach!

Object-Oriented Implementation: 1st step

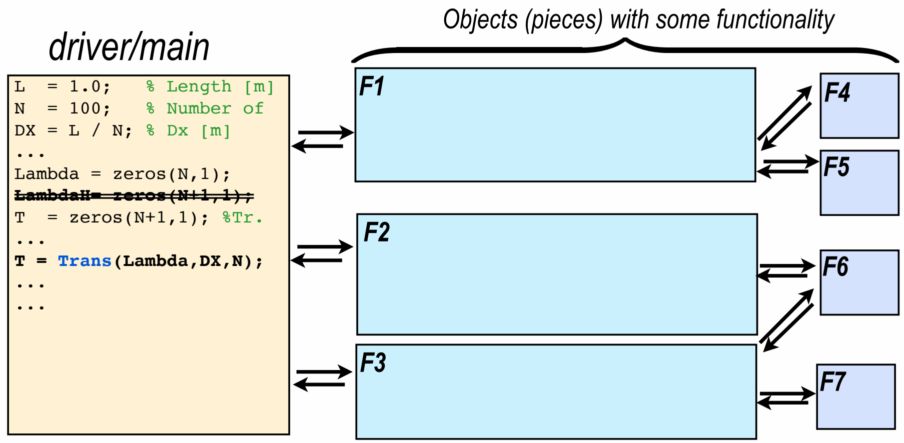
"Divide (into manageable pieces) & Conquer"



Let us Calculate T in our program with this approach!

Object-Oriented Implementation: 1st step

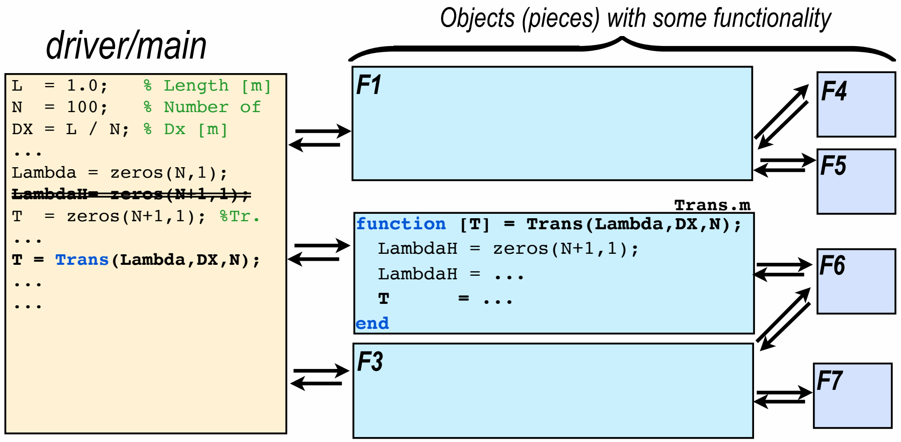
"Divide (into manageable pieces) & Conquer"



Let us Calculate T in our program with this approach!

Object-Oriented Implementation: 1st step

"Divide (into manageable pieces) & Conquer"



Let us Calculate T in our program with this approach!

Tasks until next lecture - 26th of April

- Solve 1D pressure for heterogeneous reservoirs.
- Calculate (& plot) velocity, using Darcy's law.

Tip:

```
U        = zeros (N+1, 1);
U(1)     = ?
U(N+1) = ?
for i = 2:N
    U(i) = - lambdaH(?) * (p(?) - p(?))/Dx;
end
```

- Study Finite-Volume slides (3 pages).
- Implement based on 'Trans' function.

End of Lecture 1 and PC 1

Review of Lecture 1

- Pressure Equation:

Formulate Pressure (Flow) Equation

- Pressure equations:

Incompressible single/multiphase pressure equation:

$$-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$$

Compressible single-phase pressure equation:

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p) = \rho q$$

Slightly compressible single-phase pressure equation:

$$\phi c_{\text{eff}} \frac{\partial p}{\partial t} - \nabla \cdot (\frac{K}{\mu} \cdot \nabla p) = q$$

Review of Lecture 1

- **Pressure Equation:**
- **Discretized 1D Elliptic -incomp.- Pressure Eq.**

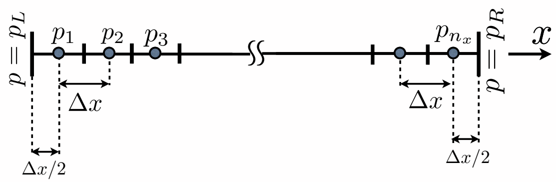
$$-\frac{\partial}{\partial x}\left(\lambda \frac{\partial p}{\partial x}\right) = q$$

Finite-Difference:

$$T_{i-1/2} = \frac{\lambda_t^{i-1/2}}{\Delta x^2}$$

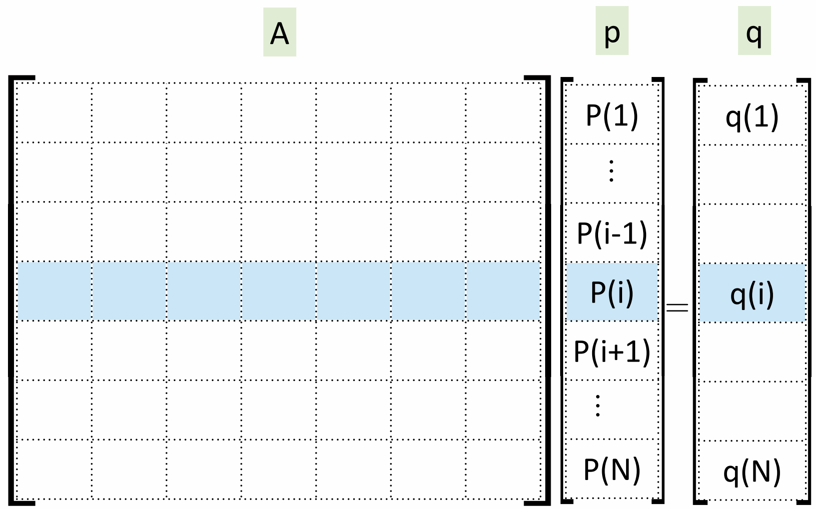
$$T_{i-1/2}(p_i - p_{i-1}) + T_{i+1/2}(p_i - p_{i+1}) = q_i$$

in matlab: T(i) * (p(i) - p(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)



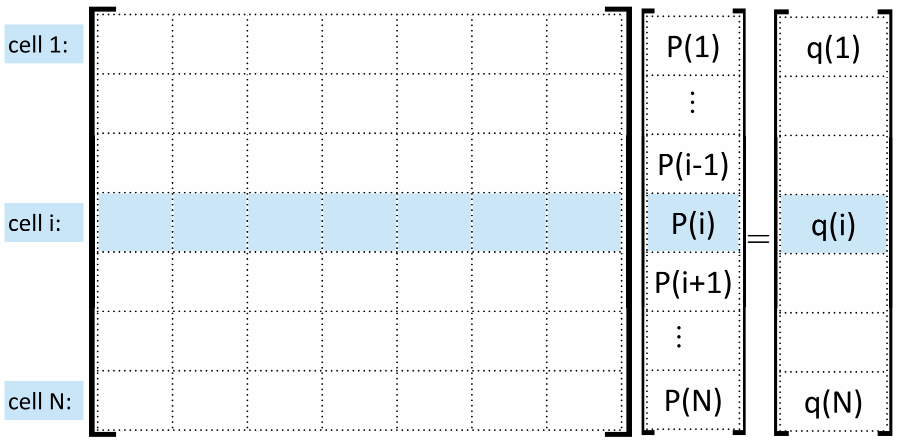
Review of Lecture 1

in matlab: T(i) * (p(i) - p(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)



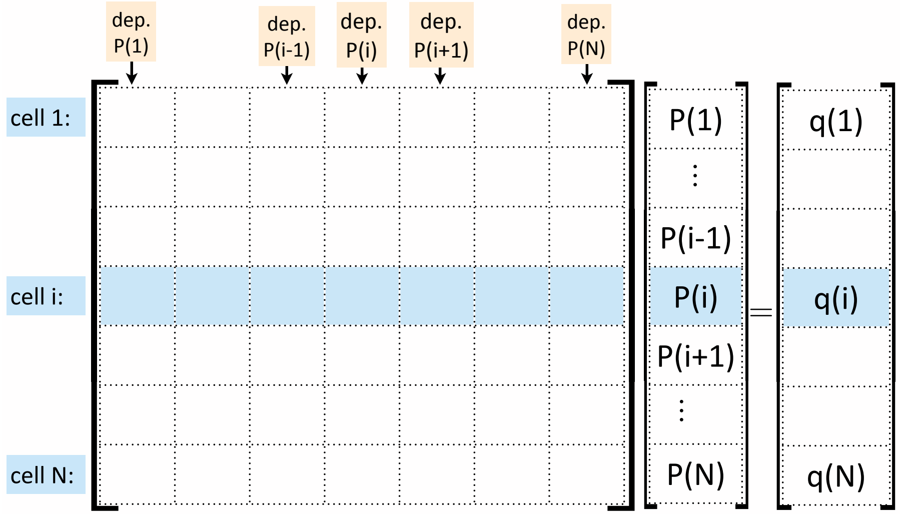
Review of Lecture 1

in matlab: T(i) * (p(i) - p(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)



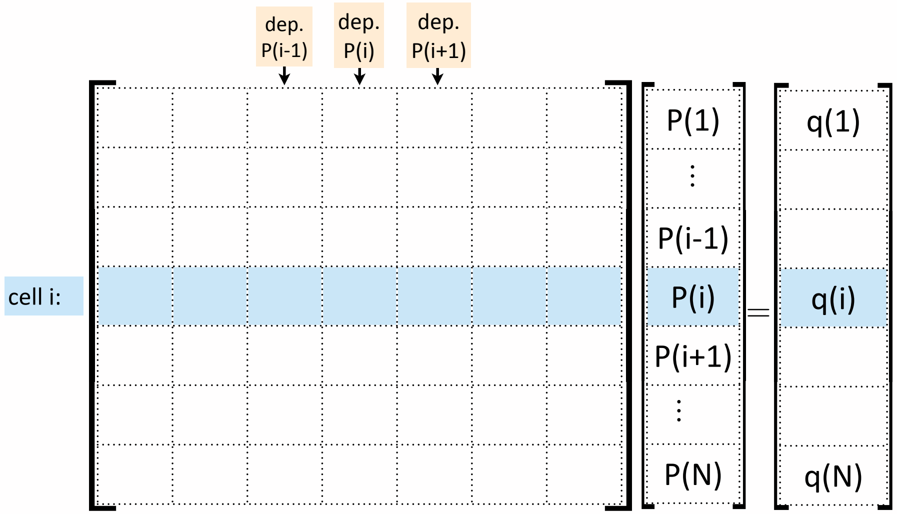
Review of Lecture 1

in matlab: T(i) * (p(i) - p(i-1)) + T(i+1) * (p(i) - p(i+1)) = q(i)



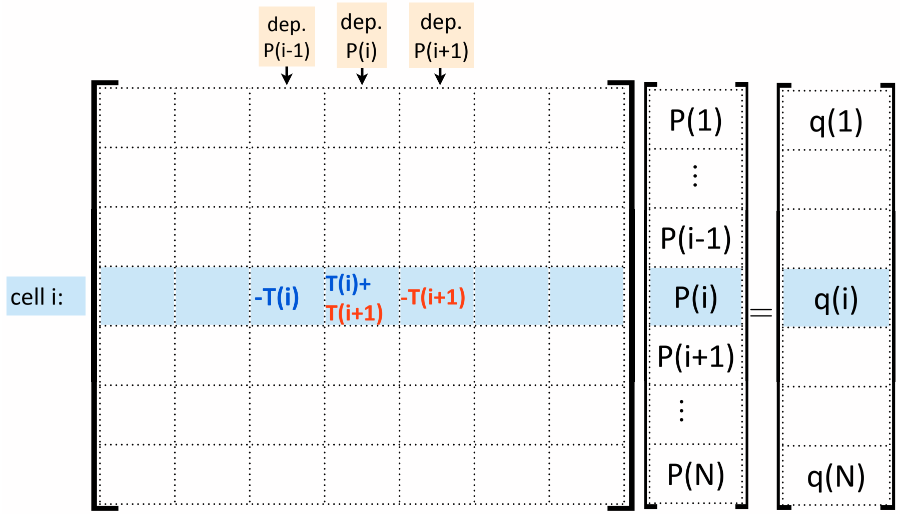
Review of Lecture 1

in matlab: **T(i)** * (p(i) - p(i-1)) + **T(i+1)** * (p(i) - p(i+1)) = q(i)



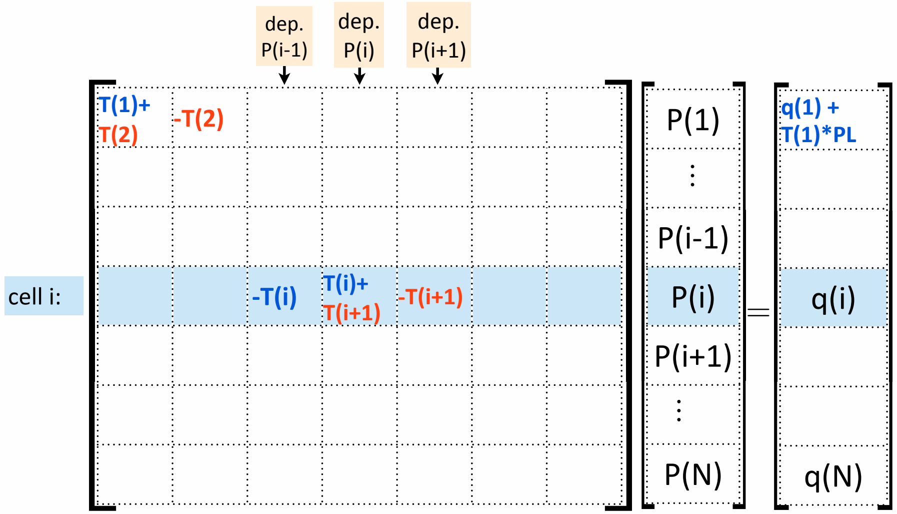
Review of Lecture 1

in matlab: **T(i)** * (p(i) - p(i-1)) + **T(i+1)** * (p(i) - p(i+1)) = q(i)



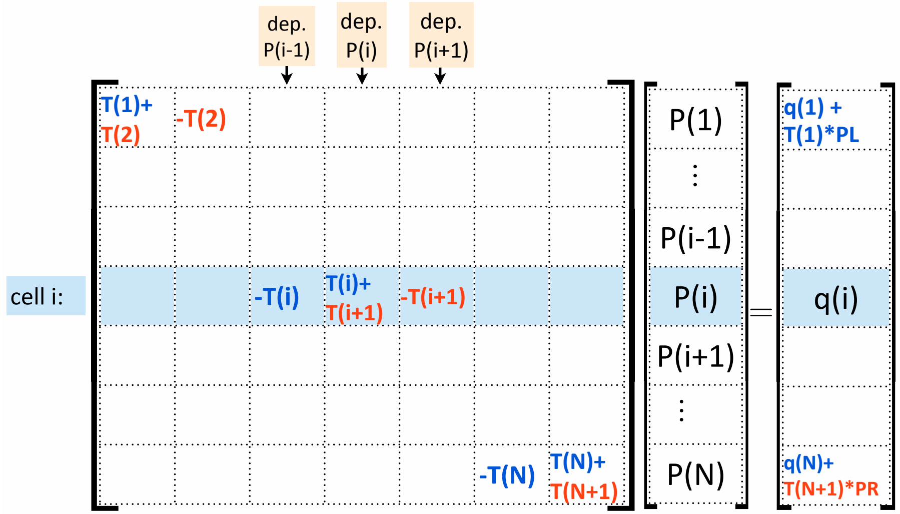
Review of Lecture 1

in matlab: **T(1)** * (p(1) - pL) + **T(1+1)** * (p(1) - p(1+1)) = q(1)



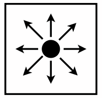
Review of Lecture 1

in matlab: **T(N)** * (p(N) - p(N-1)) + **T(N+1)** * (p(N) - pR) = q(1)

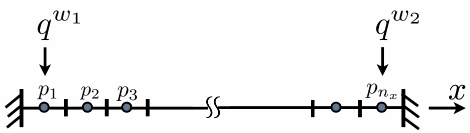


Well Model

Reservoir grid is much larger than well radius:



See refs. below: $q_t^w = \int_V q^w dV = PI^w \ \lambda_t \ (p^w - p)$

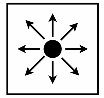


For Cell 1: $\frac{\lambda_t^{1+1/2} \Delta A}{\Delta x}(p_1 - p_2) =$?

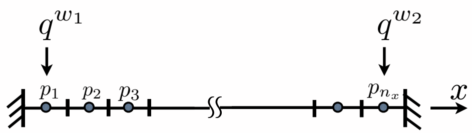
Aronofsky and Jenkins (1954), Matthews and Russel (1967) and ERCB (1975) Manual.
D.W. Peaceman (1978), Interpretation of Well-Block Pressures in Numerical Reservoir Simulation
D.W. Peaceman (1983), SPE-10528-PA: Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability

Well Model

Reservoir grid is much larger than well radius:



See refs. below: $q_t^w = \int_V q^w dV = PI^w \ \lambda_t \ (p^w - p)$



For Cell 1: $\frac{\lambda_t^{1+1/2} \Delta A}{\Delta x}(p_1 - p_2) = PI^{w_1} \lambda_t^1 (p_{w_1} - p_1)$

Aronofsky and Jenkins (1954), Matthews and Russel (1967) and ERCB (1975) Manual.
D.W. Peaceman (1978), Interpretation of Well-Block Pressures in Numerical Reservoir Simulation
D.W. Peaceman (1983), SPE-10528-PA: Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability

Slightly Compressible P

$$\phi c_{\text{eff}} \frac{\partial p}{\partial t} - \frac{\partial}{\partial x}\left(\frac{K}{\mu}\frac{\partial p}{\partial x}\right) = q$$

Slightly Compressible P

$$\underbrace{\phi c_{\text{eff}} \frac{\partial p}{\partial t}}\ \underbrace{- \frac{\partial}{\partial x}\left(\frac{K}{\mu}\frac{\partial p}{\partial x}\right) = q}_{\text{incompressible}}$$

$$\boxed{T_{i-1/2}(p_i^{n+1} - p_{i-1}^{n+1}) + T_{i+1/2}(p_i^{n+1} - p_{i+1}^{n+1}) = q_i}$$

$$(\quad \boxed{A}\ )\ p^{n+1} = (q \quad)$$

Slightly Compressible P

$$\underbrace{\phi c_{\text{eff}}\frac{\partial p}{\partial t}}\;\underbrace{-\frac{\partial}{\partial x}\left(\frac{K}{\mu}\frac{\partial p}{\partial x}\right)=q}_{\text{incompressible}}$$

$$\boxed{\frac{\phi c_{\text{eff}}}{\Delta t}(p_i^{n+1}-p_i^{n})} \qquad \boxed{T_{i-1/2}(p_i^{n+1}-p_{i-1}^{n+1})+T_{i+1/2}(p_i^{n+1}-p_{i+1}^{n+1})=q_i}$$

$$(\boxed{C}+\boxed{A})\;p^{n+1}=(q+\boxed{Cp^n})$$

Slightly Compressible P

$$\underbrace{\phi c_{\text{eff}}\frac{\partial p}{\partial t}}\;\underbrace{-\frac{\partial}{\partial x}\left(\frac{K}{\mu}\frac{\partial p}{\partial x}\right)=q}_{\text{incompressible}}$$

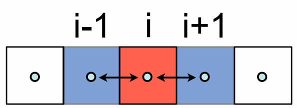
$$\boxed{\frac{\phi c_{\text{eff}}}{\Delta t}(p_i^{n+1}-p_i^{n})} \qquad \boxed{T_{i-1/2}(p_i^{n}-p_{i-1}^{n})+T_{i+1/2}(p_i^{n}-p_{i+1}^{n})=q_i}$$

$$\boxed{C}\;p^{n+1}=(q+\boxed{Cp^n})-\boxed{A}\;p^n$$

Slightly Compressible P

$$\boxed{\frac{\phi c_{\text{eff}}}{\Delta t}}(p^{n+1}-p^n)-\frac{\partial}{\partial x}\left(\frac{K}{\mu}\frac{\partial p}{\partial x}\right)=q$$

$$(\boxed{C}+A)\;p^{n+1}=(q+\boxed{C}p^n) \quad \text{or} \quad \boxed{C}\,p^{n+1}=(q+\boxed{C}p^n)-A\;p^n$$



$$\boxed{C}=\frac{1}{\Delta t}\begin{pmatrix}(\phi c_{\text{eff}})_1\\ \vdots \\ (\phi c_{\text{eff}})_N\end{pmatrix} I$$

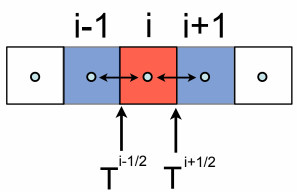
I is identity matrix:

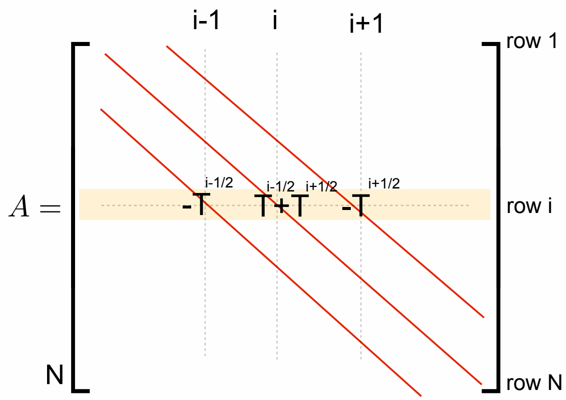
$$I=\begin{pmatrix}1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1\end{pmatrix}$$

Slightly Compressible P

$$\frac{\phi c_{\text{eff}}}{\Delta t}(p^{n+1}-p^n)-\boxed{\frac{\partial}{\partial x}\left(\frac{K}{\mu}\frac{\partial p}{\partial x}\right)}=q$$

$$(C+\boxed{A})\;p^{n+1}=(q+Cp^n) \quad \text{or} \quad C\;p^{n+1}=(q+Cp^n)-\boxed{A}\,p^n$$





Consistency, Order of Approximation, Convergence, Stability, ...

Discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

FD scheme: $T_{i-1/2}(\mathbf{p_i} - \mathbf{p_{i-1}}) + T_{i+1/2}(\mathbf{p_i} - \mathbf{p_{i+1}}) = Q_i$

Rearrange: $-T_{i-1/2}\ \mathbf{p_{i-1}} + (T_{i-1/2} + T_{i+1/2})\ \mathbf{p_i} - T_{i+1/2}\ \mathbf{p_{i+1}} = Q_i$

If T = cte.: $T(-\mathbf{p_{i-1}} + 2\ \mathbf{p_i} - \mathbf{p_{i+1}}) = Q_i$ **Corresponds to:** $-\underline{\underline{\lambda}}_t \dfrac{\partial^2 p}{\partial x^2} = q$

Q. How accurate is this?

Discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

FD scheme: $T_{i-1/2}(\mathbf{p_i} - \mathbf{p_{i-1}}) + T_{i+1/2}(\mathbf{p_i} - \mathbf{p_{i+1}}) = Q_i$

Rearrange: $-T_{i-1/2}\ \mathbf{p_{i-1}} + (T_{i-1/2} + T_{i+1/2})\ \mathbf{p_i} - T_{i+1/2}\ \mathbf{p_{i+1}} = Q_i$

If T = cte.: $T(-\mathbf{p_{i-1}} + 2\ \mathbf{p_i} - \mathbf{p_{i+1}}) = Q_i$ **Corresponds to:** $-\underline{\underline{\lambda}}_t \dfrac{\partial^2 p}{\partial x^2} = q$

Q. How accurate is this? **A. Taylor series expansion will tell us!**

$$p(x_i + \Delta x) = p(x_i) + \Delta x \left.\frac{\partial p}{\partial x}\right|_{x_i} + \frac{\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i} + \frac{\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i} + \frac{\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i} + O(\Delta x^5)$$

$$p(x_i - \Delta x) = p(x_i) - \Delta x \left.\frac{\partial p}{\partial x}\right|_{x_i} + \frac{\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i} - \frac{\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i} + \frac{\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i} + O(\Delta x^5)$$

Discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

FD scheme: $T_{i-1/2}(\mathbf{p_i} - \mathbf{p_{i-1}}) + T_{i+1/2}(\mathbf{p_i} - \mathbf{p_{i+1}}) = Q_i$

Rearrange: $-T_{i-1/2}\ \mathbf{p_{i-1}} + (T_{i-1/2} + T_{i+1/2})\ \mathbf{p_i} - T_{i+1/2}\ \mathbf{p_{i+1}} = Q_i$

If T = cte.: $T(-\mathbf{p_{i-1}} + 2\ \mathbf{p_i} - \mathbf{p_{i+1}}) = Q_i$ **Corresponds to:** $-\underline{\underline{\lambda}}_t \dfrac{\partial^2 p}{\partial x^2} = q$

Q. How accurate is this? **A. Taylor series expansion will tell us!**

$$p(x_i + \Delta x) = p(x_i) + \Delta x \left.\frac{\partial p}{\partial x}\right|_{x_i} + \frac{\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i} + \frac{\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i} + \frac{\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i} + O(\Delta x^5)$$

$$p(x_i - \Delta x) = p(x_i) - \Delta x \left.\frac{\partial p}{\partial x}\right|_{x_i} + \frac{\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i} - \frac{\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i} + \frac{\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i} + O(\Delta x^5)$$

$$\Rightarrow \frac{-p_{i+1} + 2p_i - p_{i-1}}{\Delta x^2} = -\frac{\partial^2 p}{\partial x^2} - \frac{\Delta x^2}{12}\frac{\partial^4 p}{\partial x^4} + O(\Delta x^3)$$

In other words, $\dfrac{-p_{i+1} + 2p_i - p_{i-1}}{\Delta x^2}$ **is a 2nd order accurate approximation of** $\dfrac{\partial^2 p}{\partial x^2}$

Discretization accuracy and consistency:

Let p_{approx} be the solution to the equation $T\left(-p_{i+1}+2p_i-p_{i-1}\right)=Q=0$

Let p_{exact} be the solution to the equation $-\lambda\frac{\partial^2 p}{\partial x^2}=0$

Now for the error holds (follows immediately from truncation error) that

$$|p_{exact}-p_{approx}|=O\left(\Delta x^2\right)$$

such that in the limit of Δx towards zero the error vanishes

Discretization accuracy and consistency:

Let p_{approx} be the solution to the equation $T\left(-p_{i+1}+2p_i-p_{i-1}\right)=Q=0$

Let p_{exact} be the solution to the equation $-\lambda\frac{\partial^2 p}{\partial x^2}=0$

Now for the error holds (follows immediately from truncation error) that

$$|p_{exact}-p_{approx}|=O\left(\Delta x^2\right)$$

such that in the limit of Δx towards zero the error vanishes

A system is said to be ***consistent*** and has ***consistency*** of order α, if the truncation error is order $O(\Delta x^{\alpha})$ with $\alpha\geq 1$.

Some extra info to obtain higher accurate discretization is in the appendix.
The whole idea is based on Taylor expansion (no magic!)

Compressible single-phase P:

Parabolic pressure equation: $(\phi\rho)c_{\text{eff}}\frac{\partial p}{\partial t}-\nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)=\rho q$

- Spacial (convective) term is the same as in incompressible flow.
- Different temporal discretization schemes:

  Euler Backward, Euler Forward, and Crank Nicolson

- In general, take: $\frac{\partial \mathbf{p}}{\partial t}=\mathbf{f}\left(t,\mathbf{p}\right)$
- Discretization of d/dt: $\frac{\partial p_i}{\partial t}\approx\frac{p_i(t+\Delta t)-p_i\left(t\right)}{\Delta t}=\frac{p_i^{n+1}-p_i^n}{\Delta t}$
- Now the question is: $\frac{p^{n+1}-p^n}{\Delta t}\approx\frac{\partial p}{\partial t}=\mathbf{f}(t^?,p^?)$

$$c_{eff}=c_\phi+c_f=\frac{1}{\phi}\frac{\partial\phi}{\partial p}+\frac{1}{\rho}\frac{\partial\rho}{\partial p}$$

Compressible single-phase P:

Parabolic pressure equation: $(\phi\rho)c_{\text{eff}}\frac{\partial p}{\partial t}-\nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)=\rho q$

Linear or slightly comp. form: $\phi c_{\text{eff}}\frac{\partial p}{\partial t}-\nabla\cdot\left(\frac{K}{\mu}\cdot\nabla p\right)=q$

- Options for calculation of **f** in $\frac{p^{n+1}-p^n}{\Delta t}\approx\frac{\partial p}{\partial t}=\mathbf{f}(t^?,p^?)$

Compressible single-phase P:

Parabolic pressure equation: $(\phi\rho)c_{\mathrm{eff}}\dfrac{\partial p}{\partial t} - \nabla\cdot(\rho\dfrac{K}{\mu}\cdot\nabla p) = \rho q$

Linear or slightly comp. form: $\phi c_{\mathrm{eff}}\dfrac{\partial p}{\partial t} - \nabla\cdot(\dfrac{K}{\mu}\cdot\nabla p) = q$

- Options for calculation of **f** in $\dfrac{p^{n+1}-p^n}{\Delta t} \approx \dfrac{\partial p}{\partial t} = \mathbf{f}(t^?, p^?)$

$$\mathbf{f}(t^?,\mathbf{p}^?) \approx \begin{cases} \mathbf{f}(t^n,\mathbf{p}^n) & \textit{Euler forward} \longleftarrow \text{explicit} \\ \mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{Euler backward} \longleftarrow \text{implicit} \\ \frac{1}{2}\left(\mathbf{f}(t^n,\mathbf{p}^n)+\mathbf{f}(t^{n+1},\mathbf{p}^{n+1})\right) & \textit{Crank Nicolson} \\ \alpha\mathbf{f}(t^n,\mathbf{p}^n)+[1-\alpha]\mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{General } \alpha\in[0,1] \end{cases}$$

Compressible single-phase P:

Euler Forward is consistent: $\dfrac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t} = \mathbf{f}(t^n,\mathbf{p}^n)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization:**

$$\mathbf{f}(t^?,\mathbf{p}^?) \approx \begin{cases} \mathbf{f}(t^n,\mathbf{p}^n) & \textit{Euler forward} \longleftarrow \text{explicit} \\ \mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{Euler backward} \longleftarrow \text{implicit} \\ \frac{1}{2}\left(\mathbf{f}(t^n,\mathbf{p}^n)+\mathbf{f}(t^{n+1},\mathbf{p}^{n+1})\right) & \textit{Crank Nicolson} \\ \alpha\mathbf{f}(t^n,\mathbf{p}^n)+[1-\alpha]\mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{General } \alpha\in[0,1] \end{cases}$$

Compressible single-phase P:

Euler Forward is consistent: $\dfrac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t} = \mathbf{f}(t^n,\mathbf{p}^n)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization:**

$$p^{n+1} = p^n + \Delta t\left.\frac{\partial p}{\partial t}\right|^n + \frac{\Delta t^2}{2}\left.\frac{\partial^2 p}{\partial t^2}\right|^n + O(\Delta t^3)$$
$$= p^n + \Delta t\,\mathbf{f}(t,p)\Big|^n + \frac{\Delta t^2}{2}\left.\frac{\partial \mathbf{f}(t,p)}{\partial t}\right|^n + O(\Delta t^3)$$

$$\mathbf{f}(t^?,\mathbf{p}^?) \approx \begin{cases} \mathbf{f}(t^n,\mathbf{p}^n) & \textit{Euler forward} \longleftarrow \text{explicit} \\ \mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{Euler backward} \longleftarrow \text{implicit} \\ \frac{1}{2}\left(\mathbf{f}(t^n,\mathbf{p}^n)+\mathbf{f}(t^{n+1},\mathbf{p}^{n+1})\right) & \textit{Crank Nicolson} \\ \alpha\mathbf{f}(t^n,\mathbf{p}^n)+[1-\alpha]\mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{General } \alpha\in[0,1] \end{cases}$$

Compressible single-phase P:

Euler Forward is consistent: $\dfrac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t} = \mathbf{f}(t^n,\mathbf{p}^n)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization:**

$$p^{n+1} = p^n + \Delta t\left.\frac{\partial p}{\partial t}\right|^n + \frac{\Delta t^2}{2}\left.\frac{\partial^2 p}{\partial t^2}\right|^n + O(\Delta t^3)$$
$$= p^n + \Delta t\,\mathbf{f}(t,p)\Big|^n + \frac{\Delta t^2}{2}\left.\frac{\partial \mathbf{f}(t,p)}{\partial t}\right|^n + O(\Delta t^3)$$

$$\Rightarrow \boxed{\tau = \frac{p^{n+1}-p^n}{\Delta t} - \mathbf{f}(t^n,p^n) = \frac{\Delta t}{2}\left.\frac{\partial \mathbf{f}}{\partial t}\right|^n + O(\Delta t^2) = O(\Delta t)}$$

$$\mathbf{f}(t^?,\mathbf{p}^?) \approx \begin{cases} \mathbf{f}(t^n,\mathbf{p}^n) & \textit{Euler forward} \longleftarrow \text{explicit} \\ \mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{Euler backward} \longleftarrow \text{implicit} \\ \frac{1}{2}\left(\mathbf{f}(t^n,\mathbf{p}^n)+\mathbf{f}(t^{n+1},\mathbf{p}^{n+1})\right) & \textit{Crank Nicolson} \\ \alpha\mathbf{f}(t^n,\mathbf{p}^n)+[1-\alpha]\mathbf{f}(t^{n+1},\mathbf{p}^{n+1}) & \textit{General } \alpha\in[0,1] \end{cases}$$

Compressible single-phase P:

Euler Forward is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\mathbf{f}\left(t^n,\mathbf{p}^n\right)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization**

Crank Nicolson is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\frac{\mathbf{f}\left(t^n,\mathbf{p}^n\right)+\mathbf{f}\left(t^{n+1},\mathbf{p}^{n+1}\right)}{2}$

- **2nd order in spacial discretization for T = cte.**
- **2nd order in temporal discretization** $p^{n+1}=p^n+\Delta t\frac{\partial p}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 p}{\partial t^2}\Big|^n+O(\Delta t^3)$

Compressible single-phase P:

Euler Forward is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\mathbf{f}\left(t^n,\mathbf{p}^n\right)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization**

Crank Nicolson is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\frac{\mathbf{f}\left(t^n,\mathbf{p}^n\right)+\mathbf{f}\left(t^{n+1},\mathbf{p}^{n+1}\right)}{2}$

- **2nd order in spacial discretization for T = cte.**
- **2nd order in temporal discretization** $p^{n+1}=p^n+\Delta t\frac{\partial p}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 p}{\partial t^2}\Big|^n+O(\Delta t^3)$

$$\boxed{p^{n+1}=p^n+\Delta t\;\mathbf{f}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial \mathbf{f}}{\partial t}\Big|^n+O(\Delta t^3)}\quad\boxed{\mathbf{f}^{n+1}=\mathbf{f}^n+\Delta t\frac{\partial \mathbf{f}}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{f}}{\partial t^2}\Big|^n+O(\Delta t^3)}$$

Compressible single-phase P:

Euler Forward is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\mathbf{f}\left(t^n,\mathbf{p}^n\right)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization**

Crank Nicolson is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\frac{\mathbf{f}\left(t^n,\mathbf{p}^n\right)+\mathbf{f}\left(t^{n+1},\mathbf{p}^{n+1}\right)}{2}$

- **2nd order in spacial discretization for T = cte.**
- **2nd order in temporal discretization** $p^{n+1}=p^n+\Delta t\frac{\partial p}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 p}{\partial t^2}\Big|^n+O(\Delta t^3)$

$$p^{n+1}=p^n+\Delta t\;\mathbf{f}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial \mathbf{f}}{\partial t}\Big|^n+O(\Delta t^3)\quad\mathbf{f}^{n+1}=\mathbf{f}^n+\Delta t\frac{\partial \mathbf{f}}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{f}}{\partial t^2}\Big|^n+O(\Delta t^3)$$

$$\tau=\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}-\frac{\mathbf{f}^{n+1}+\mathbf{f}^n}{2}$$

Compressible single-phase P:

Euler Forward is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\mathbf{f}\left(t^n,\mathbf{p}^n\right)$

- **2nd order in spacial discretization for T = cte. (like incompressible)**
- **1st order in temporal discretization**

Crank Nicolson is consistent: $\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}=\frac{\mathbf{f}\left(t^n,\mathbf{p}^n\right)+\mathbf{f}\left(t^{n+1},\mathbf{p}^{n+1}\right)}{2}$

- **2nd order in spacial discretization for T = cte.**
- **2nd order in temporal discretization** $p^{n+1}=p^n+\Delta t\frac{\partial p}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 p}{\partial t^2}\Big|^n+O(\Delta t^3)$

$$p^{n+1}=p^n+\Delta t\;\mathbf{f}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial \mathbf{f}}{\partial t}\Big|^n+O(\Delta t^3)\quad\mathbf{f}^{n+1}=\mathbf{f}^n+\Delta t\frac{\partial \mathbf{f}}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{f}}{\partial t^2}\Big|^n+O(\Delta t^3)$$

$$\tau=\frac{\mathbf{p}^{n+1}-\mathbf{p}^n}{\Delta t}-\frac{\mathbf{f}^{n+1}+\mathbf{f}^n}{2}$$

$$=\frac{\Delta t\;\mathbf{f}^n+\frac{\Delta t^2}{2}\frac{\partial \mathbf{f}}{\partial t}\Big|^n+O(\Delta t^3)}{\Delta t}-\frac{2\mathbf{f}^n+\Delta t\frac{\partial \mathbf{f}}{\partial t}\Big|^n+\frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{f}}{\partial t^2}\Big|^n+O(\Delta t^3)}{2}$$

$$\boxed{=-\frac{2\Delta t^2}{24}\frac{\partial^2 \mathbf{f}}{\partial t^2}\Big|^n+O(\Delta t^3)=O(\Delta t^2)}$$

Check consistency with your code:

Let $\tilde{p}\left(t;\Delta t=\frac{\Delta t_0}{2^m}\right)$ be the solution obtained by halving the time step m-times

For P-consistent scheme holds:: $\tilde{p}\left(t;\Delta t=\frac{\Delta t_0}{2^m}\right)=p_{exact}(t)+h(t)\left(\Delta t\right)^P$

Define the error between two sub-sequent time step refinements as:

After some math .... (read at home)

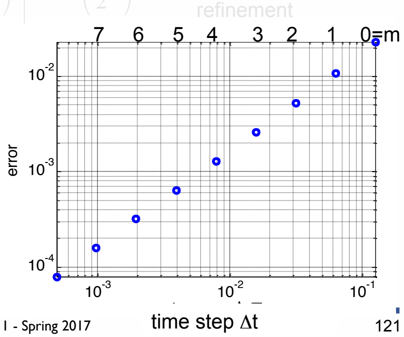
If the discretisation method has order P then

$$e(t;m)\approx\max_t\left|h(t)\left[1-2^P\right]\left(\frac{\Delta t_0}{2^m}\right)^P\right|\le M\bullet\left(\frac{\Delta t_0}{2^m}\right)^P$$

for some finite value M.

If we now plot the error *e* as function of time step on log-log paper, we can deduce the order of consistency, P, from the slope of the data.

(Remark: provided that the solution and source/sink behaviour is continuous in time.)



Check consistency with your code:

Let $\tilde{p}\left(t;\Delta t=\frac{\Delta t_0}{2^m}\right)$ be the solution obtained by halving the time step m-times

For P-consistent scheme holds:: $\tilde{p}\left(t;\Delta t=\frac{\Delta t_0}{2^m}\right)=p_{exact}(t)+h(t)\left(\Delta t\right)^P$

Define the error between two sub-sequent time step refinements as:

$$e(t;m)\equiv\max_t\left|\tilde{p}\left(t;\frac{\Delta t_0}{2^m}\right)-\tilde{p}\left(t;\frac{\Delta t_0}{2^{m-1}}\right)\right|$$

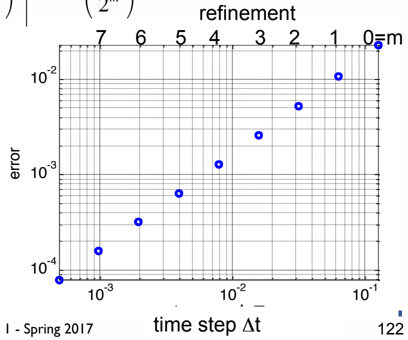
If the discretisation method has order P then

$$e(t;m)\approx\max_t\left|h(t)\left[1-2^P\right]\left(\frac{\Delta t_0}{2^m}\right)^P\right|\le M\bullet\left(\frac{\Delta t_0}{2^m}\right)^P$$

for some finite value M.

If we now plot the error *e* as function of time step on log-log paper, we can deduce the order of consistency, P, from the slope of the data.

(Remark: provided that the solution and source/sink behaviour is continuous in time.)



Check consistency with your code:

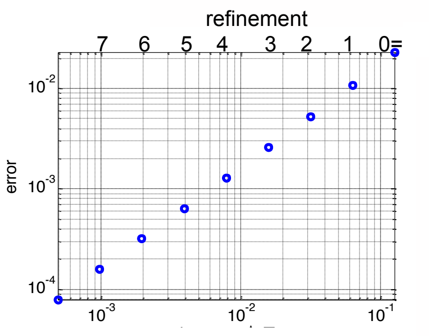
Easy way (check the math):

$$\epsilon_1=O(\Delta x_1^m)$$

$$\epsilon_2=O(\Delta x_2^m)$$

$$\frac{\epsilon_1}{\epsilon_2}\sim\left(\frac{\Delta x_1^m}{\Delta x_2^m}\right)=\left(\frac{\Delta x_1}{\Delta x_2}\right)^m$$

$$\frac{ln(\epsilon_1)-ln(\epsilon_2)}{ln(\Delta x_1)-ln(\Delta x_2)}=m$$



Convergence = Consistency + Stability

- Consistency of a time dependent method is a ***necessary*** but ***insufficient*** requirement that it converges to the “exact solution”.

- Perturbations in the approximation must damp over time (should not grow!).

- For a convergent numerical method it is therefore required that the method is also ***numerically stable***.

To check stability we will look at one of the following two different methods:

1. **Von Neumann stability analysis (the hard way)**
2. Matrix stability analysis (the easy way)

Stability analysis

What does stability mean?

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\frac{\partial p}{\partial t} - D\frac{\partial^2 p}{\partial x^2} = 0 \qquad \boxed{p_i^{n+1} = p_i^n + D\frac{\Delta t}{\Delta x^2}[p_{i+1}^n - 2p_i^n + p_{i-1}^n]}$$

$$(\phi c)\frac{\partial p}{\partial t} - \frac{K}{\mu}\frac{\partial^2 p}{\partial x^2} = 0 \qquad \boxed{p_i^{n+1} = p_i^n + \frac{K}{\mu\phi c_i}\frac{\Delta t}{\Delta x^2}[p_{i+1}^n - 2p_i^n + p_{i-1}^n]}$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

Error Finite Fourier series: $\epsilon(x) = \sum_{m=1}^{M} A_m e^{ik_m x} \quad k_m = \frac{\pi m}{L} \quad M = L/\Delta x$

A_m is time-dependent: $\epsilon(x,t) = \sum_{m=1}^{M} e^{at} e^{ik_m x}$

Take the m-th mode: $\epsilon_m(x,t) = e^{at} e^{ik_m x}$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$
$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$
$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

Error Finite Fourier series: $\epsilon(x) = \sum_{m=1}^{M} A_m e^{ik_m x} \quad k_m = \frac{\pi m}{L} \quad M = L/\Delta x$

A$_m$ is time-dependent: $\epsilon(x,t) = \sum_{m=1}^{M} e^{at} e^{ik_m x}$

Take the m-th mode: $\epsilon_m(x,t) = e^{at}e^{ik_m x}$

For cell i time n: $\epsilon_i^n = e^{at}e^{ik_m x}$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$
$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$
$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

Error Finite Fourier series: $\epsilon(x) = \sum_{m=1}^{M} A_m e^{ik_m x} \quad k_m = \frac{\pi m}{L} \quad M = L/\Delta x$

A$_m$ is time-dependent: $\epsilon(x,t) = \sum_{m=1}^{M} e^{at} e^{ik_m x}$

Take the m-th mode: $\epsilon_m(x,t) = e^{at}e^{ik_m x}$

$$\epsilon_i^{n+1} = e^{a(t+\Delta t)}e^{ik_m x}$$
$$\epsilon_{i-1}^n = e^{at}e^{ik_m(x-\Delta x)}$$

For cell i+1 time n: $\epsilon_{i+1}^n = e^{at}e^{ik_m(x+\Delta x)}$

For cell i time n: $\epsilon_i^n = e^{at}e^{ik_m x}$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$
$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$
$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

$$\epsilon_i^{n+1} = e^{a\Delta t}\epsilon_i^n = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x})\epsilon_i^n$$

$$\epsilon_i^{n+1} = e^{a(t+\Delta t)}e^{ik_m x}$$
$$\epsilon_{i-1}^n = e^{at}e^{ik_m(x-\Delta x)}$$
$$\epsilon_{i+1}^n = e^{at}e^{ik_m(x+\Delta x)}$$
$$\epsilon_i^n = e^{at}e^{ik_m x}$$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$
$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$
$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

$$\epsilon_i^{n+1} = e^{a\Delta t}\epsilon_i^n = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x})\epsilon_i^n$$

$$e^{a\Delta t} = 1 + \frac{\Delta t\ D}{\Delta x^2}(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x})$$

$$e^{a\Delta t} = 1 - \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m\Delta x/2)$$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

$$e^{a\Delta t} = 1 - \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m \Delta x/2)$$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

$$e^{a\Delta t} = 1 - \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m \Delta x/2)$$

Von Neumann stability analysis

Euler Forward (slightly comp. Homogeneous reservoir):

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^n - 2\mathbf{p}_i^n + \mathbf{p}_{i-1}^n)$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$$

$$e^{a\Delta t} = 1 - \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m \Delta x/2)$$

Stability: $$|G| \equiv \left|\frac{\epsilon_i^{n+1}}{\epsilon_i^n}\right| = |e^{a\Delta t}| = \left|1 - \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m \Delta x/2)\right| \le 1$$

$$\left|1 - \frac{2\Delta t\ D}{\Delta x^2}(1 - \cos(k_m \Delta x))\right| \le 1 \qquad \frac{\Delta t\ D}{\Delta x^2} \le \frac{1}{2} \qquad \boxed{\Delta t \le \frac{1}{2}\frac{\Delta x^2}{D}}$$

Note: |R|<1 => -1 < R < 1

So Euler Forward is conditionally stable!

Von Neumann stability analysis

Euler Backward:

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t\ D}{\Delta x^2}(\mathbf{p}_{i+1}^{n+1} - 2\mathbf{p}_i^{n+1} + \mathbf{p}_{i-1}^{n+1})$$

$$\epsilon_i^n = \bar{\mathbf{p}}_i^n - \mathbf{p}_i^n$$

$$\epsilon_i^{n+1} = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(\epsilon_{i+1}^{n+1} - 2\epsilon_i^{n+1} + \epsilon_{i-1}^{n+1})$$

$$\epsilon_i^{n+1} = e^{a\Delta t}\epsilon_i^n = \epsilon_i^n + \frac{\Delta t\ D}{\Delta x^2}(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x})e^{a\Delta t}\epsilon_i^n$$

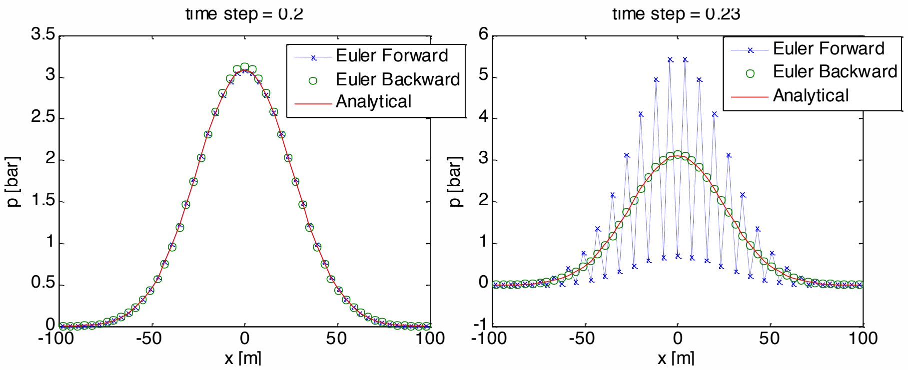
$$e^{a\Delta t}\left(1 - \frac{\Delta t\ D}{\Delta x^2}(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x})\right) = 1$$

$$e^{a\Delta t}\left(1 + \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m\Delta x/2)\right) = 1$$

Stability: $$|G| \equiv \left|\frac{\epsilon_i^{n+1}}{\epsilon_i^n}\right| = |e^{a\Delta t}| = \frac{1}{\left(1 + \frac{4\Delta t\ D}{\Delta x^2}\sin^2(k_m\Delta x/2)\right)} \le 1$$

So Euler Backward is unconditionally stable!

Von Neumann stability analysis



Other time-integration methods

1. Single-step methods, $dp/dt-f \approx g(p^{n+1},p^n)$
   i.e., the discretisation only depends on (n) and (n+1)

2. Multi-step methods, e.g., $dp/dt-f \approx g(p^{n+1}, p^n, p^{n-1}, p^{n-2})$
   i.e, the discretisation depends on more steps than (n) and (n+1)

   their analysis to determine stability regions is complicated.
   Although interesting, outside the scope of this course….

General Compressible Flow (Pressure) Eq.

General Compressible Pressure Equation

Pressure Eq:
(nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right) = \rho q$$

General Compressible Pressure Equation

Pressure Eq: (nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$$

$$\downarrow$$

Euler-backward:

$$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

General Compressible Pressure Equation

Pressure Eq: (nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$$

$$\downarrow$$

Euler-backward:

$$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

Newton Linearization Lemma: $(\bullet)^{n+1} \approx (\bullet)^\nu + \frac{\partial(\bullet)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$

General Compressible Pressure Equation

Pressure Eq: (nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$$

$$\downarrow$$

Euler-backward:

$$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

Newton Linearization Lemma: $(\bullet)^{n+1} \approx (\bullet)^\nu + \frac{\partial(\bullet)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$

$$(\phi\rho)^{n+1} \approx (\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

$$(\rho q)^{n+1} \approx (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

General Compressible Pressure Equation

Pressure Eq: (nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$$

$$\downarrow$$

Euler-backward:

$$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

Newton Linearization Lemma: $(\bullet)^{n+1} \approx (\bullet)^\nu + \frac{\partial(\bullet)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$

$$(\phi\rho)^{n+1} \approx (\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

$$(\rho q)^{n+1} \approx (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

$$-\nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} \approx -\nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla p^{\nu+1}\right)$$

General Compressible Pressure Equation

Pressure Eq:
(nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$$

Euler-backward:

$$\frac{(\phi\rho)^{n+1}-(\phi\rho)^{n}}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

$$\frac{(\phi\rho)^{\nu} + \frac{\partial(\phi\rho)}{\partial p}\big|^{\nu}(p^{\nu+1}-p^{\nu})}{\Delta t} - \nabla\cdot\left(\rho^{\nu}\frac{K}{\mu}\cdot\nabla p^{\nu+1}\right) = \frac{(\phi\rho)^{n}}{\Delta t} + (\rho q)^{\nu} + \cancel{\frac{\partial(\rho q)}{\partial p}\big|^{\nu}(p^{\nu+1}-p^{\nu})}$$

General Compressible Pressure Equation

Pressure Eq:
(nonlinear)

$$\frac{\partial}{\partial t}(\phi\rho) - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$$

Euler-backward:

$$\frac{(\phi\rho)^{n+1}-(\phi\rho)^{n}}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

$$\frac{(\phi\rho)^{\nu} + \frac{\partial(\phi\rho)}{\partial p}\big|^{\nu}(p^{\nu+1}-p^{\nu})}{\Delta t} - \nabla\cdot\left(\rho^{\nu}\frac{K}{\mu}\cdot\nabla p^{\nu+1}\right) = \frac{(\phi\rho)^{n}}{\Delta t} + (\rho q)^{\nu} + \cancel{\frac{\partial(\rho q)}{\partial p}\big|^{\nu}(p^{\nu+1}-p^{\nu})}$$

$$\underbrace{\frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}\big|^{\nu}p^{\nu+1} - \nabla\cdot\left(\rho^{\nu}\frac{K}{\mu}\cdot\nabla p^{\nu+1}\right) - \cancel{\frac{\partial(\rho q)}{\partial p}}\big|^{\nu}p^{\nu+1}}_{A^{\nu}p^{\nu+1}} = \underbrace{\frac{(\phi\rho)^{n}}{\Delta t} + (\rho q)^{\nu} - \cancel{\frac{\partial(\rho q)}{\partial p}}\big|^{\nu}p^{\nu} - \frac{(\phi\rho)^{\nu}}{\Delta t} + \frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}\big|^{\nu}p^{\nu}}_{r^{\nu}}$$

Reservoir Simulators are all developed in Residual Form!

General Compressible Pressure Equation

Find P which satisfies:

$$\boxed{\frac{(\phi\rho)^{n+1}-(\phi\rho)^{n}}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}}$$

General Compressible Pressure Equation

Find P which satisfies:

$$\boxed{\frac{(\phi\rho)^{n+1}-(\phi\rho)^{n}}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}}$$

$$\boxed{R(p^{\nu}) = (\rho q)^{\nu} - \frac{(\phi\rho)^{\nu}-(\phi\rho)^{n}}{\Delta t} + \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{\nu}}$$

General Compressible Pressure Equation

Find P which satisfies: $$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^{n+1} = (\rho q)^{n+1}$$

P is the answer if R(p) is zero: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^\nu$$

General Compressible Pressure Equation

Find P which satisfies: $$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^{n+1} = (\rho q)^{n+1}$$

P is the answer if R(p) is zero: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^\nu$$

Linearized:

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu (p^{\nu+1} - p^\nu)}{\Delta t} - \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla p^{\nu+1}\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

General Compressible Pressure Equation

Find P which satisfies: $$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^{n+1} = (\rho q)^{n+1}$$

P is the answer if R(p) is zero: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^\nu$$

Linearized:

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu (p^{\nu+1} - p^\nu)}{\Delta t} - \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla p^{\nu+1}\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

Rewrite it for: $\delta p^{\nu+1} = p^{\nu+1} - p^\nu$

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu \delta p^{\nu+1}}{\Delta t} - \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla(\delta p^{\nu+1} + p^\nu)\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu \delta p^{\nu+1}$$

General Compressible Pressure Equation

Find P which satisfies: $$\frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} - \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^{n+1} = (\rho q)^{n+1}$$

P is the answer if R(p) is zero: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot \left(\rho \frac{K}{\mu} \cdot \nabla p\right)^\nu$$

Linearized:

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu (p^{\nu+1} - p^\nu)}{\Delta t} - \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla p^{\nu+1}\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1} - p^\nu)$$

Rewrite it for: $\delta p^{\nu+1} = p^{\nu+1} - p^\nu$

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu \delta p^{\nu+1}}{\Delta t} - \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla(\delta p^{\nu+1} + p^\nu)\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu \delta p^{\nu+1}$$

$$\frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}\Big|^\nu \delta p^{\nu+1} - \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla \delta p^{\nu+1}\right) - \frac{\partial(\rho q)}{\partial p}\Big|^\nu \delta p^{\nu+1} = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu - \frac{1}{\Delta t}(\phi\rho)^\nu + \nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla p^\nu\right)$$

General Compressible Pressure Equation

Find P which satisfies: $$\frac{(\phi\rho)^{n+1}-(\phi\rho)^n}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

P is the answer if R(p) is zero: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu-(\phi\rho)^n}{\Delta t} + \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^\nu$$

Linearized:

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu (p^{\nu+1}-p^\nu)}{\Delta t} - \nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla p^{\nu+1}\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1}-p^\nu)$$

Rewrite it for: $\delta p^{\nu+1} = p^{\nu+1} - p^\nu$

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu \delta p^{\nu+1}}{\Delta t} - \nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla(\delta p^{\nu+1}+p^\nu)\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu \delta p^{\nu+1}$$

$$\frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}\Big|^\nu \delta p^{\nu+1} - \nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla\delta p^{\nu+1}\right) - \frac{\partial(\rho q)}{\partial p}\Big|^\nu \delta p^{\nu+1} = R^\nu$$

General Compressible Pressure Equation

Find P which satisfies: $$\frac{(\phi\rho)^{n+1}-(\phi\rho)^n}{\Delta t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^{n+1} = (\rho q)^{n+1}$$

P is the answer if R(p) is zero: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu-(\phi\rho)^n}{\Delta t} + \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^\nu$$

Linearized:

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu (p^{\nu+1}-p^\nu)}{\Delta t} - \nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla p^{\nu+1}\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu (p^{\nu+1}-p^\nu)$$

Rewrite it for: $\delta p^{\nu+1} = p^{\nu+1} - p^\nu$

$$\frac{(\phi\rho)^\nu + \frac{\partial(\phi\rho)}{\partial p}\big|^\nu \delta p^{\nu+1}}{\Delta t} - \nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla(\delta p^{\nu+1}+p^\nu)\right) = \frac{(\phi\rho)^n}{\Delta t} + (\rho q)^\nu + \frac{\partial(\rho q)}{\partial p}\Big|^\nu \delta p^{\nu+1}$$

$$\frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}\Big|^\nu \boxed{\delta p^{\nu+1}} - \nabla\cdot\left(\rho^\nu\frac{K}{\mu}\cdot\nabla\boxed{\delta p^{\nu+1}}\right) - \frac{\partial(\rho q)}{\partial p}\Big|^\nu \boxed{\delta p^{\nu+1}} = R^\nu$$

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1-

2-

3-

4-

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $$R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu-(\phi\rho)^n}{\Delta t} + \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right)^\nu$$

2-

3-

4-

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \dfrac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \dfrac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $\dfrac{1}{\Delta t}\dfrac{\partial(\phi\rho)}{\partial p}|^\nu \delta p^{\nu+1} - \nabla \cdot (\rho^\nu \dfrac{K}{\mu} \cdot \nabla \delta p^{\nu+1}) - \dfrac{\partial(\rho q)}{\partial p}|^\nu \delta p^{\nu+1} = R^\nu$

3- Update:

4-

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \dfrac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \dfrac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $\dfrac{1}{\Delta t}\dfrac{\partial(\phi\rho)}{\partial p}|^\nu \delta p^{\nu+1} - \nabla \cdot (\rho^\nu \dfrac{K}{\mu} \cdot \nabla \delta p^{\nu+1}) - \dfrac{\partial(\rho q)}{\partial p}|^\nu \delta p^{\nu+1} = R^\nu$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4-

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \dfrac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \dfrac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $\dfrac{1}{\Delta t}\dfrac{\partial(\phi\rho)}{\partial p}|^\nu \delta p^{\nu+1} - \nabla \cdot (\rho^\nu \dfrac{K}{\mu} \cdot \nabla \delta p^{\nu+1}) - \dfrac{\partial(\rho q)}{\partial p}|^\nu \delta p^{\nu+1} = R^\nu$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4- **If** (Converged)

else:

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \dfrac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \dfrac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $\dfrac{1}{\Delta t}\dfrac{\partial(\phi\rho)}{\partial p}|^\nu \delta p^{\nu+1} - \nabla \cdot (\rho^\nu \dfrac{K}{\mu} \cdot \nabla \delta p^{\nu+1}) - \dfrac{\partial(\rho q)}{\partial p}|^\nu \delta p^{\nu+1} = R^\nu$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4- **If** (Converged) $p^{n+1} \leftarrow p^{\nu+1}$ and leave;

else: $p^\nu \leftarrow p^{\nu+1}$, update properties, and go to step 1.

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $\frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}|^\nu \delta p^{\nu+1} - \nabla \cdot (\rho^\nu \frac{K}{\mu} \cdot \nabla \delta p^{\nu+1}) - \frac{\partial(\rho q)}{\partial p}|^\nu \delta p^{\nu+1} = R^\nu$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4- **If** (Converged) $p^{n+1} \leftarrow p^{\nu+1}$ and leave;

else: $p^\nu \leftarrow p^{\nu+1}$, update properties, and go to step 1.

Convergence check: || R || or $||\delta p^{\nu+1}|| < \epsilon$

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $(C + A)\ \delta p^{\nu+1} = R^\nu$

$$C = \frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}|^\nu \ \mathbf{I}$$

$$-\nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla \delta p^{\nu+1}\right) = A\delta p^{\nu+1}$$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4- **If** (Converged) $p^{n+1} \leftarrow p^{\nu+1}$ and leave;

else: $p^\nu \leftarrow p^{\nu+1}$, update properties, and go to step 1.

Convergence check: || R || or $||\delta p^{\nu+1}|| < \epsilon$

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $\underbrace{(C + A)}_{-J}\ \delta p^{\nu+1} = R^\nu$

$$C = \frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}|^\nu \ \mathbf{I}$$

$$-\nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla \delta p^{\nu+1}\right) = A\delta p^{\nu+1}$$

$$J = \frac{\partial R}{\partial p}$$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4- **If** (Converged) $p^{n+1} \leftarrow p^{\nu+1}$ and leave;

else: $p^\nu \leftarrow p^{\nu+1}$, update properties, and go to step 1.

Convergence check: || R || or $||\delta p^{\nu+1}|| < \epsilon$

General Compressible Pressure Equation

0- Initial guess: $p^\nu \leftarrow p^n$

1- Compute: $R(p^\nu) = (\rho q)^\nu - \frac{(\phi\rho)^\nu - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^\nu$

2- Solve: $J^\nu \delta p^{\nu+1} = -R^\nu$

$$C = \frac{1}{\Delta t}\frac{\partial(\phi\rho)}{\partial p}|^\nu \ \mathbf{I}$$

$$-\nabla \cdot \left(\rho^\nu \frac{K}{\mu} \cdot \nabla \delta p^{\nu+1}\right) = A\delta p^{\nu+1}$$

$$J = \frac{\partial R}{\partial p}$$

3- Update: $p^{\nu+1} = p^\nu + \delta p^{\nu+1}$

4- **If** (Converged) $p^{n+1} \leftarrow p^{\nu+1}$ and leave;

else: $p^\nu \leftarrow p^{\nu+1}$, update properties, and go to step 1.

Convergence check: || R || or $||\delta p^{\nu+1}|| < \epsilon$

General Compressible Pressure Equation

One can describe flow simulator as:

1- Describe residual as function of primary unknowns (here, P)

$$R(p^{n+1}) = (\rho q)^{n+1} - \frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^{n+1}$$

General Compressible Pressure Equation

One can describe flow simulator as:

1- Describe residual as function of primary unknowns (here, P)

$$R(p^{n+1}) = (\rho q)^{n+1} - \frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^{n+1}$$

2- Our objective is $R(p^{n+1}) = 0$

General Compressible Pressure Equation

One can describe flow simulator as:

1- Describe residual as function of primary unknowns (here, P)

$$R(p^{n+1}) = (\rho q)^{n+1} - \frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^{n+1}$$

2- Our objective is $R(p^{n+1}) = 0$

3- Residual is a nonlinear function of pressure, so linearize and solve:

$$0 = R(p^{n+1}) \approx R(p^\nu) + \frac{\partial R}{\partial p} \delta p^{\nu+1}$$

General Compressible Pressure Equation

One can describe flow simulator as:

1- Describe residual as function of primary unknowns (here, P)

$$R(p^{n+1}) = (\rho q)^{n+1} - \frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^{n+1}$$

2- Our objective is $R(p^{n+1}) = 0$

3- Residual is a nonlinear function of pressure, so linearize and solve:

$$0 = R(p^{n+1}) \approx R(p^\nu) + \frac{\partial R}{\partial p} \delta p^{\nu+1}$$

$$\Rightarrow \quad \frac{\partial R}{\partial p} \delta p^{\nu+1} = -R(p^\nu)$$

General Compressible Pressure Equation

One can describe flow simulator as:

1- Describe residual as function of primary unknowns (here, P)

$$\boxed{R(p^{n+1}) = (\rho q)^{n+1} - \frac{(\phi\rho)^{n+1} - (\phi\rho)^n}{\Delta t} + \nabla \cdot (\rho \frac{K}{\mu} \cdot \nabla p)^{n+1}}$$

2- Our objective is $\boxed{R(p^{n+1}) = 0}$

3- Residual is a nonlinear function of pressure, so linearize and solve:

$$\boxed{0 = R(p^{n+1}) \approx R(p^\nu) + \frac{\partial R}{\partial p}\delta p^{\nu+1}}$$

$$\Rightarrow \boxed{\frac{\partial R}{\partial p}\delta p^{\nu+1} = -R(p^\nu)}$$

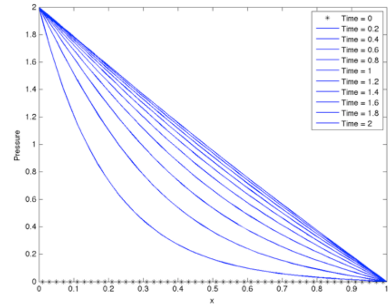
$$\Rightarrow \boxed{J\delta p^{\nu+1} = -R(p^\nu)}$$

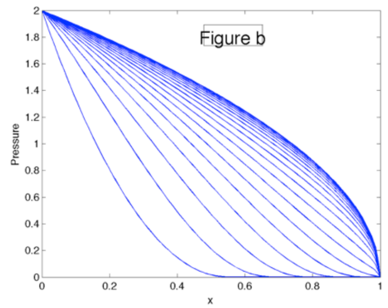
if no well

$J = -(A + C)$

MATLAB: Fully compressible flow sim

Determine slightly compressible and fully compressible.



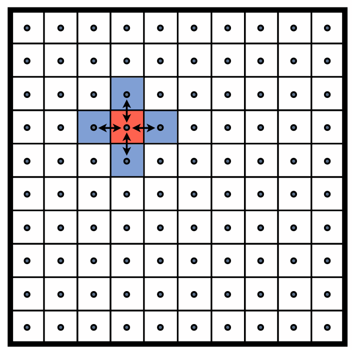


2 Dimensional …

"It is all the matter of
ordering, storing & accessing the data
correctly"

2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

FV scheme at fine-scale:

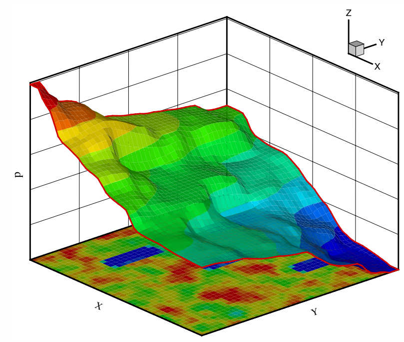


$$F_{ij} = T_{ij}(p_j - p_i)$$

$$-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q : Ap = r$$

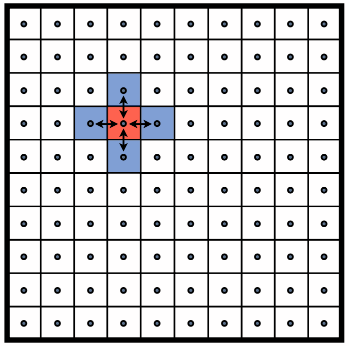
2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

FV scheme at fine-scale:



grid: 44x44

Pressure solution

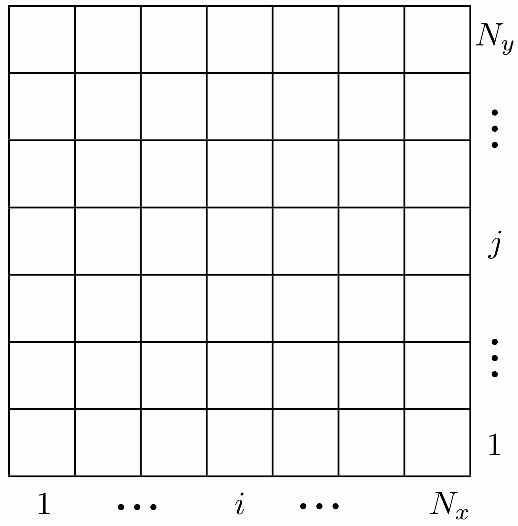


$$F_{ij} = T_{ij}(p_j - p_i)$$

$$-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q : Ap = r$$

2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

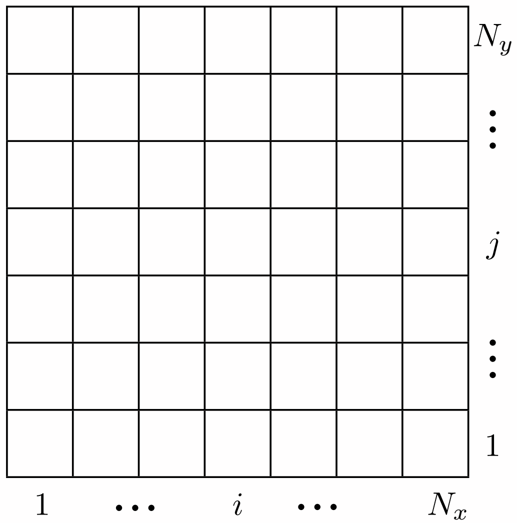
"storing & accessing the data" is based on "**Ordering**"



2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- ...

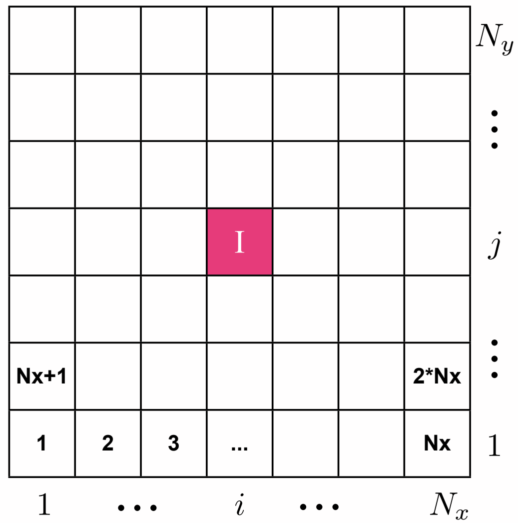


2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- ...

```
%Index of cell (i,j)
I    = ?
```

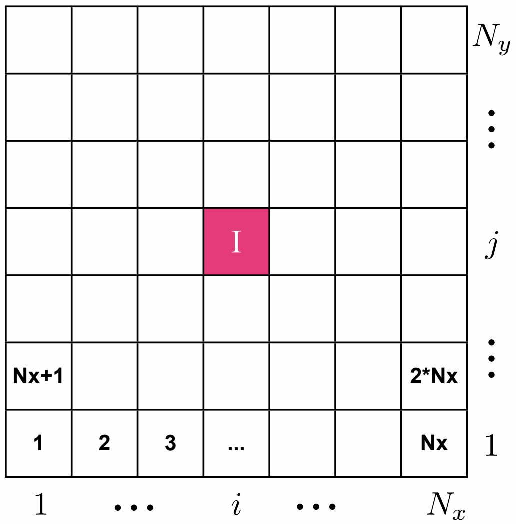


2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- …

```
%Index of cell (i,j)
I   = (j-1) * Nx + i;
```

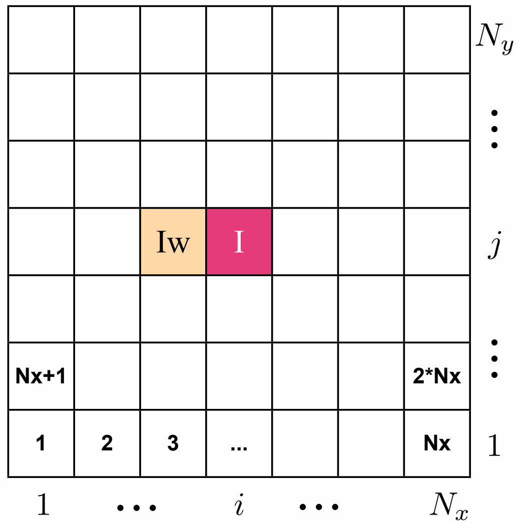


2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- …

```
%Index of cell (i,j)
I   = (j-1) * Nx + i;

%Index of cell (i-1,j)
Iw  = (j-1)*Nx + i-1;
```

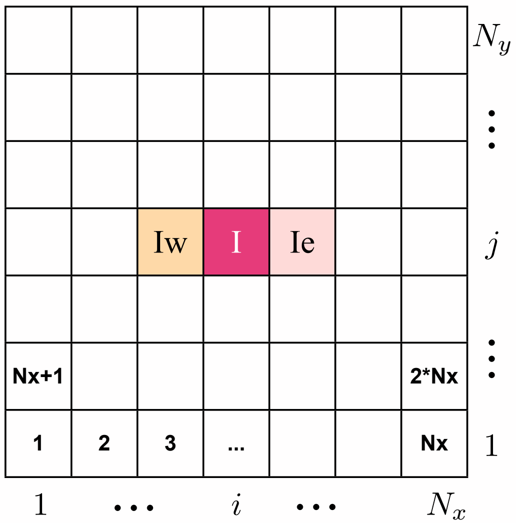


2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- …

```
%Index of cell (i,j)
I   = (j-1) * Nx + i;

%Index of cell (i-1,j)
Iw  = (j-1)*Nx + i-1;

%Index of cell (i+1,j)
Ie  = (j-1)*Nx + i+1;
```

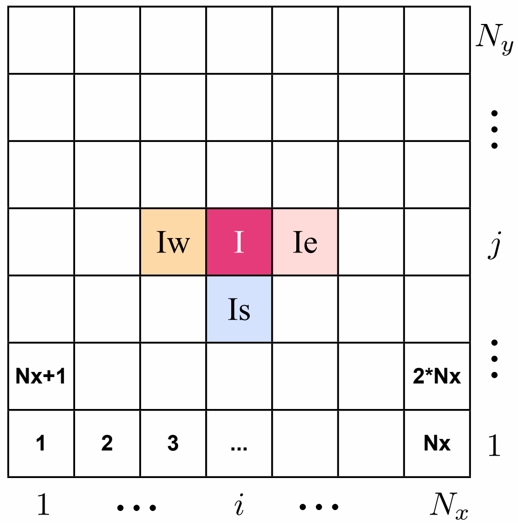


2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- …

```
%Index of cell (i,j)
I   = (j-1) * Nx + i;

%Index of cell (i-1,j)
Iw  = (j-1)*Nx + i-1;

%Index of cell (i+1,j)
Ie  = (j-1)*Nx + i+1;

%Index of cell (i,j-1)
Is  = (j-1-1)*Nx + i;
```

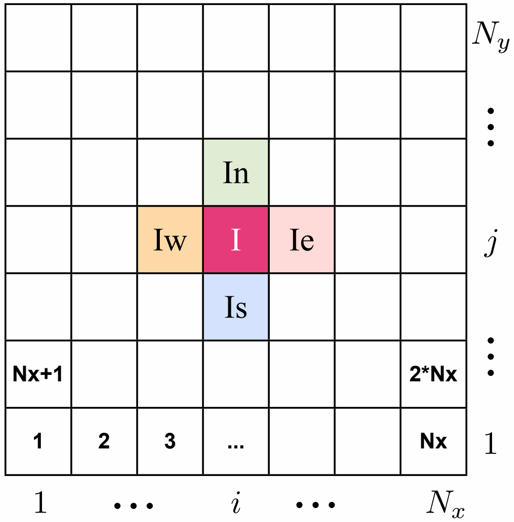


2D discretization of: $-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$

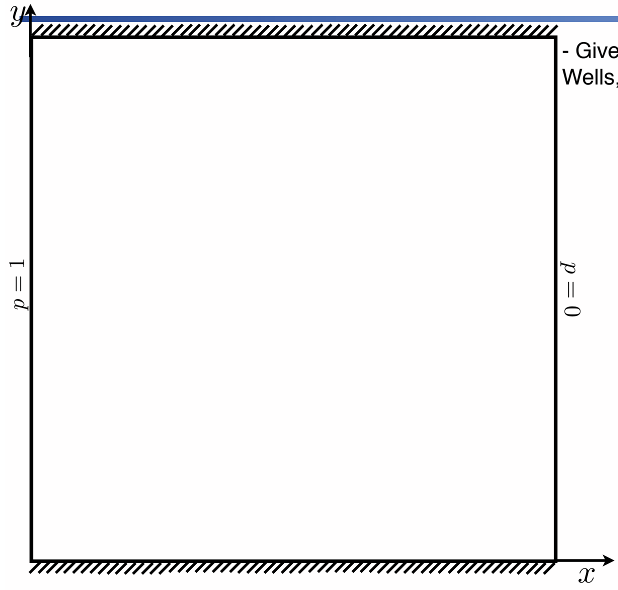
"storing & accessing the data" is based on "**Ordering**"

- **Row-based Ordering**
- Column-based Ordering
- ...

```
%Index of cell (i,j)
I   = (j-1) * Nx + i;

%Index of cell (i-1,j)
Iw  = (j-1)*Nx + i-1;

%Index of cell (i+1,j)
Ie  = (j-1)*Nx + i+1;

%Index of cell (i,j-1)
Is  = (j-1-1)*Nx + i;

%Index of cell (i,j+1)
In  = (j+1-1)*Nx + i;
```

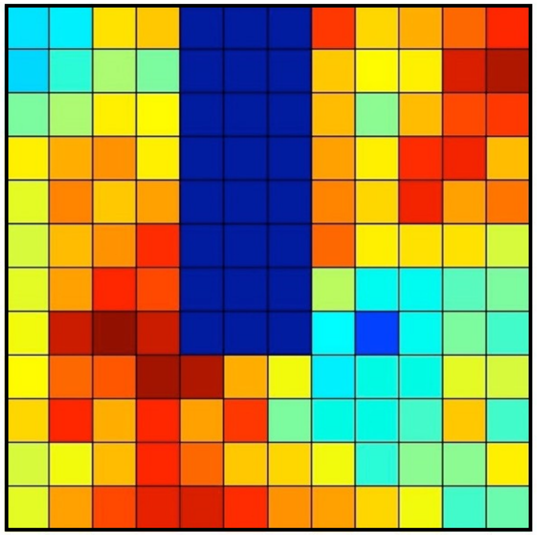


Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



- Given: Reservoir Geometry, BC, Wells, parameters, ...

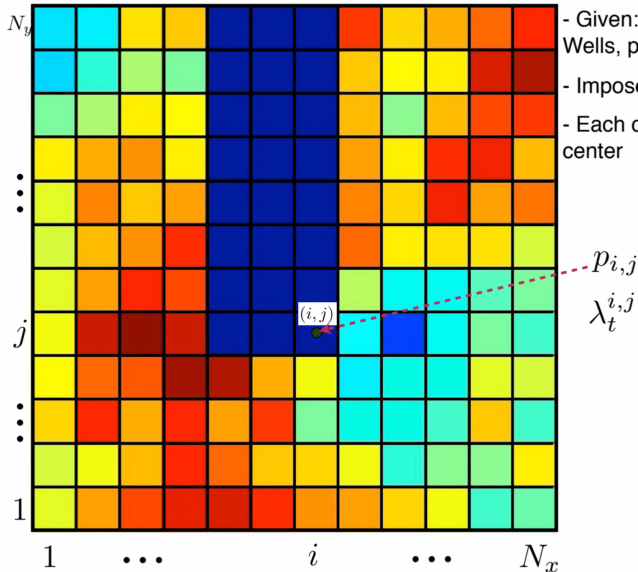
Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



λ_t

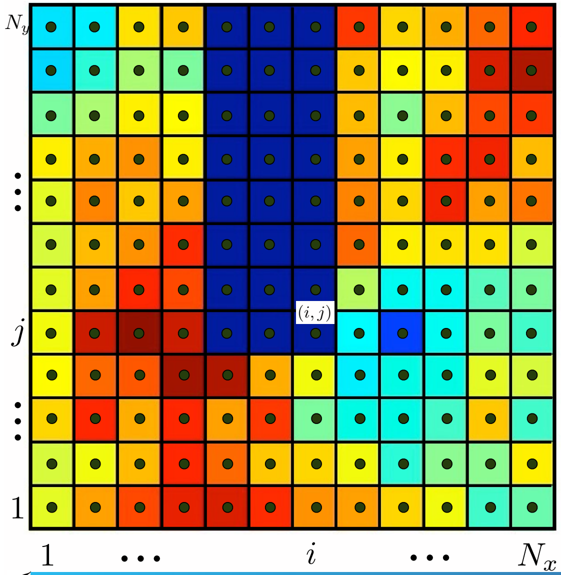
- Given: Reservoir Geometry, BC, Wells, parameters, ...

Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center

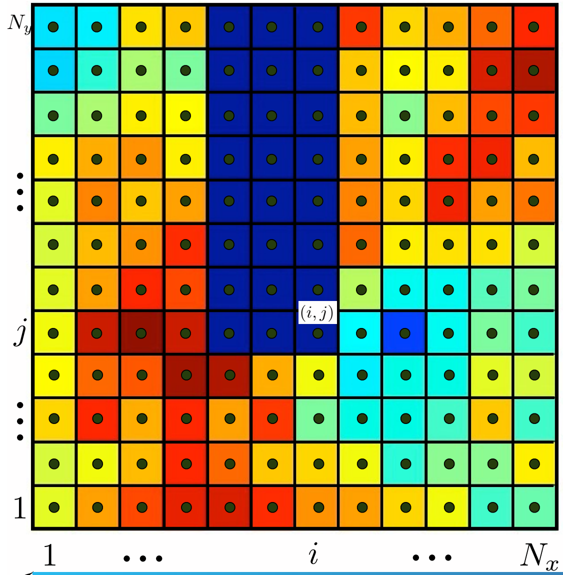
Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



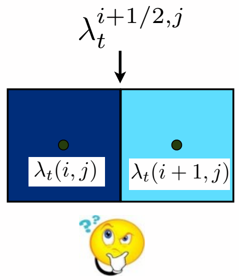
- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ...

$$\boxed{\frac{\lambda_t^{i-1/2,j}\Delta A_x}{\Delta x}}(p_{i,j} - p_{i-1,j})$$
$$\boxed{\frac{\lambda_t^{i+1/2,j}\Delta A_x}{\Delta x}}(p_{i,j} - p_{i+1,j})$$
$$\boxed{\frac{\lambda_t^{i,j-1/2}\Delta A_y}{\Delta y}}(p_{i,j} - p_{i,j-1})$$
$$\boxed{\frac{\lambda_t^{i,j+1/2}\Delta A_y}{\Delta y}}(p_{i,j} - p_{i,j+1})$$
$$= q_{i,j}\Delta V$$

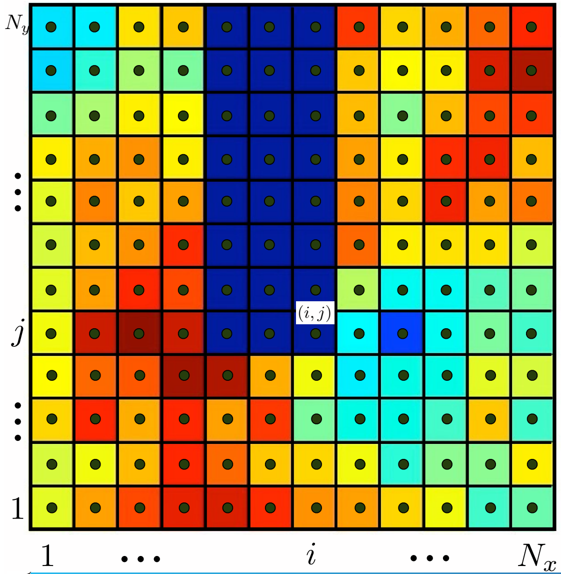
Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



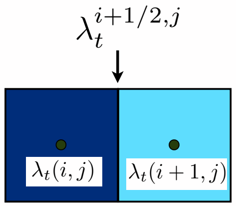
- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ...



Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



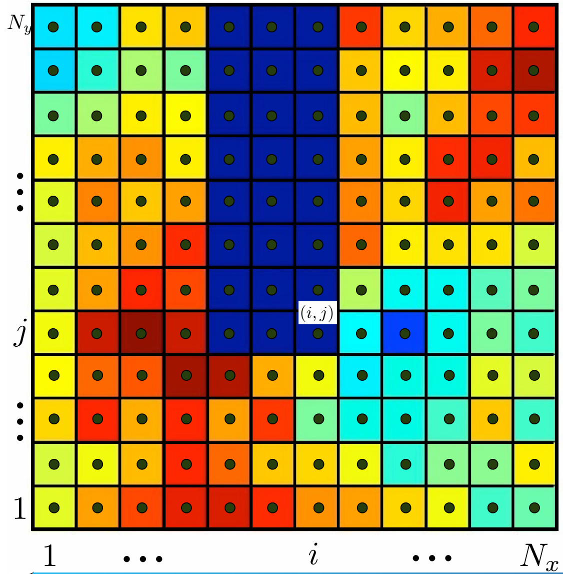
- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ...



$$\lambda_t^{i+1/2,j} = \frac{2\lambda_t^{i+1,j}\lambda_t^{i,j}}{\lambda_t^{i+1,j} + \lambda_t^{i,j}}$$

Harmonic averaging

Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$

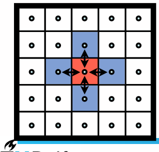


- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ...

$$\frac{\lambda_t^{i-1/2,j}}{\Delta x^2}(p_{i,j} - p_{i-1,j}) + \frac{\lambda_t^{i+1/2,j}}{\Delta x^2}(p_{i,j} - p_{i+1,j}) + \frac{\lambda_t^{i,j-1/2}}{\Delta y^2}(p_{i,j} - p_{i,j-1}) + \frac{\lambda_t^{i,j+1/2}}{\Delta y^2}(p_{i,j} - p_{i,j+1}) = q_{i,j}$$

Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$

$$\begin{array}{c|ccccccc} & (1,1) & (i-1,j) & (i,j-1) & (i,j) & (i+1,j) & (i,j+1) & (N_x,N_y) \\ \hline (1,1) & & & & & & & \\ (i,j) & & T^{i,j-1} & T^{i-1,j} & T^{i,j} & T^{i+1,j} & T^{i,j+1} & \\ (N_x,N_y) & & & & & & & \end{array}$$

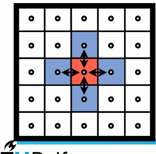


- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ...

$$\frac{\lambda_t^{i-1/2,j}}{\Delta x^2}(p_{i,j}-p_{i-1,j}) + \frac{\lambda_t^{i+1/2,j}}{\Delta x^2}(p_{i,j}-p_{i+1,j}) + \frac{\lambda_t^{i,j-1/2}}{\Delta y^2}(p_{i,j}-p_{i,j-1}) + \frac{\lambda_t^{i,j+1/2}}{\Delta y^2}(p_{i,j}-p_{i,j+1}) = q_{i,j}$$

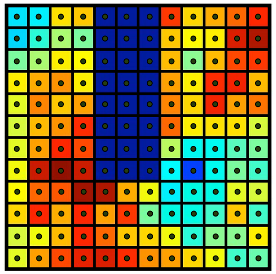
Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$

$$\begin{array}{c|ccccccc} & (1,1) & (i-1,j) & (i,j-1) & (i,j) & (i+1,j) & (i,j+1) & (N_x,N_y) \\ \hline (1,1) & & & & & & & \\ (i,j) & & T^{i,j-1} & T^{i-1,j} & T^{i,j} & T^{i+1,j} & T^{i,j+1} & \\ (N_x,N_y) & & & & & & & \end{array} \begin{bmatrix} \vdots \\ p_{i,j-1} \\ \vdots \\ p_{i-1,j} \\ p_{i,j} \\ p_{i+1,j} \\ \vdots \\ p_{i,j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ q_{i,j} \\ \\ \\ \\ \\ \end{bmatrix}$$



$$\frac{\lambda_t^{i-1/2,j}}{\Delta x^2}(p_{i,j}-p_{i-1,j}) + \frac{\lambda_t^{i+1/2,j}}{\Delta x^2}(p_{i,j}-p_{i+1,j}) + \ldots = q_{i,j}$$

Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$

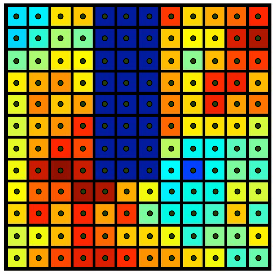


$\Downarrow$

$$\begin{array}{c|ccccccc} & (1,1) & (i-1,j) & (i,j-1) & (i,j) & (i+1,j) & (i,j+1) & (N_x,N_y) \\ \hline (1,1) & & & & & & & \\ (i,j) & & T^{i,j-1} & T^{i-1,j} & T^{i,j} & T^{i+1,j} & T^{i,j+1} & \\ (N_x,N_y) & & & & & & & \end{array} \begin{bmatrix} \vdots \\ p^{i,j-1} \\ \vdots \\ p^{i-1,j} \\ p^{i,j} \\ p^{i+1,j} \\ \vdots \\ p^{i,j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ q^{i,j} \\ \\ \\ \\ \\ \end{bmatrix}$$

- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ... we obtain a linear system.
- Impose BC in the system (Dirichlet, Neumann, etc.) and solve it to fine pressure at each point.

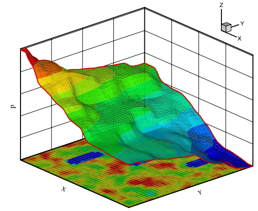
Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



$\Downarrow$

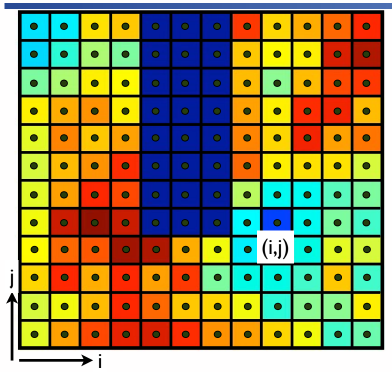
$$\begin{array}{c|ccccccc} & (1,1) & (i-1,j) & (i,j-1) & (i,j) & (i+1,j) & (i,j+1) & (N_x,N_y) \\ \hline (1,1) & & & & & & & \\ (i,j) & & T^{i,j-1} & T^{i-1,j} & T^{i,j} & T^{i+1,j} & T^{i,j+1} & \\ (N_x,N_y) & & & & & & & \end{array} \begin{bmatrix} \vdots \\ p^{i,j-1} \\ \vdots \\ p^{i-1,j} \\ p^{i,j} \\ p^{i+1,j} \\ \vdots \\ p^{i,j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ q^{i,j} \\ \\ \\ \\ \\ \end{bmatrix} \overset{\text{e.g.}}{\Longrightarrow}$$

- Given: Reservoir Geometry, BC, Wells, parameters, ...
- Impose grid (small control volumes)
- Each cell has 1 pressure value in its center
- After Mathematics ... we obtain a linear system.
- Impose BC in the system (Dirichlet, Neumann, etc.) and solve it to fine pressure at each point.



An example for pressure obtained numerically for a specific BC using 44x44 grid cells.

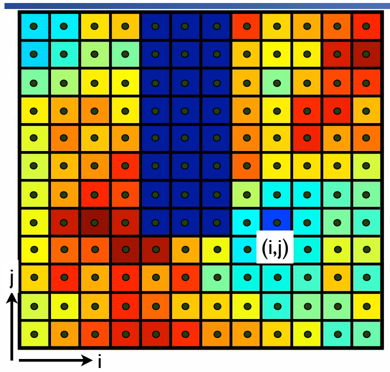
Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



We should map 2D (i,j) grid cells into 1D matrix location:

```
for i=1:Nx,        %for all cells
  for j=1:Ny
    I   = (j-1)*Nx + i  ;%row location of cell (i,j)




...
  end
end
```

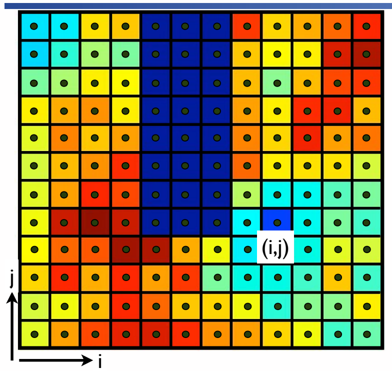

Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



We should map 2D (i,j) grid cells into 1D matrix location:

```
for i=1:Nx,        %for all cells
  for j=1:Ny
    I   = (j-1)*Nx + i  ;%row location of cell (i,j)
    Is  = (j-2)*Nx + i  ;%row location of cell (i,j-1)
    In  = (j  )*Nx + i  ;%row location of cell (i,j-1)
    Ie  = (j-1)*Nx + i+1;%row location of cell (i+1,j)
    Iw  = (j-1)*Nx + i-1;%row location of cell (i-1,j)
...
  end
end
```


Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$

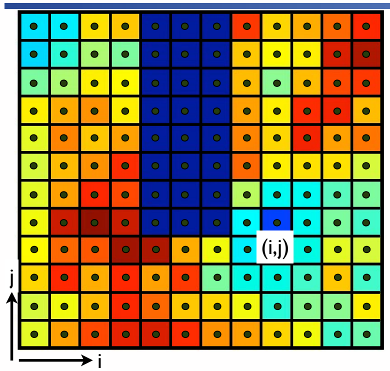


Transmissibility data structure:

- Tx (Nx+1, Ny) for E and W connections
- Ty (Nx , Ny+1) for N and S connections

```
for i=1:Nx,        %for all cells
  for j=1:Ny
    I   = (j-1)*Nx + i  ;%row location of cell (i,j)
    Is  = (j-2)*Nx + i  ;%row location of cell (i,j-1)
    In  = (j  )*Nx + i  ;%row location of cell (i,j-1)
    Ie  = (j-1)*Nx + i+1;%row location of cell (i+1,j)
    Iw  = (j-1)*Nx + i-1;%row location of cell (i-1,j)
...
  end
end
```


Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$



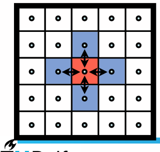
Transmissibility data structure:

- Tx (Nx+1, Ny) for E and W connections
- Ty (Nx , Ny+1) for N and S connections

```
for i=1:Nx,        %for all cells
  for j=1:Ny
    ... I, Is, In, Iw, Ie... Tx, Ty.
    if(i > 1)    add west  neighbor Tx(i,j)*[p(I) - p(Iw)]
    if(i < Nx)   add east  neighbor Tx(i+1,j)*[p(I) - p(Ie)]
    if(j > 1)    add south neighbor Ty(i,j)*[p(I) - p(Is)]
    if(j < Ny)   add north neighbor Ty(i,j+1)*[p(I) - p(In)]
  end
end
```

Introduction to discretization of $-\nabla \cdot (\lambda_t \cdot \nabla p) = q$

$$
\begin{array}{c|cccccc|}
 & (1,1) & \texttt{p(Is)} & \texttt{p(Iw)} & \texttt{p(I)} & \texttt{p(Ie)} & \texttt{p(In)} \; (N_x,N_y) \\
(1,1) & & & & & & \\
(i,j) & & \texttt{-Ty(i,j)} & \texttt{-Tx(i,j)} & \texttt{sum(T)} & \texttt{-Tx(i+1,j)} & \texttt{-Ty(i,j+1)} \\
(N_x,N_y) & & & & & &
\end{array}
\begin{bmatrix} \vdots \\ \texttt{p(Is)} \\ \vdots \\ \texttt{p(Iw)} \\ \texttt{p(I)} \\ \texttt{p(Ie)} \\ \vdots \\ \texttt{p(In)} \\ \vdots \end{bmatrix}
=
\begin{bmatrix} \\ \\ \\ \texttt{q(I)} \\ \\ \\ \\ \end{bmatrix}
$$



This was Simulation of Flow in porous media!

- Governing Equations
(incomp., slightly comp., fully comp.)

- 1D simulator development for all 3 types!
For time dependent ones: Implicit & Explicit schemes

- Well Modeling

- Stability analysis
Consistency, Order of Approx., Stability, Convergence

- 2D Simulators

Appendix

Constructing a scheme of a desired order (1/3)

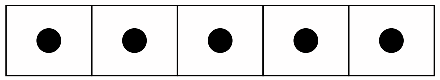
Suppose we want to approximate $-\lambda \dfrac{\partial^2 p}{\partial x^2} = q$ in gridblock i using (at least) a third order accurate approximate discretisation. How do we do this?

Taylor series expansion of the potential around gridblock i gives:

$$p(x_i+\alpha) = p(x_i) + \alpha \left.\frac{\partial p}{\partial x}\right|_{x_i} + \frac{\alpha^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i} + \frac{\alpha^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i} + \frac{\alpha^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i} + O(\alpha^5)$$

The key-idea is to combine the information of a sufficient amount of neighbouring grid blocks in a way that only the desired derivatives remain.

Construction of the solution using 5 support points: $x_{i-2}, x_{i-1}, x_i, x_{i+1}$ and x_{i+2}



i-2 i-1 i i+1 i+2

Combine the potentials within the five gridblocks around x_i such that this combination yields a 3[rd] order approximation of the second derivative:

$$a_{+2}p(x_i+2\Delta x) + a_{+1}p(x_i+\Delta x) + a_0 p(x_i) + a_{-1}p(x_i-\Delta x) + a_{-2}p(x_i-2\Delta x) = -\frac{\partial^2 p}{\partial x^2} + O(\Delta x^3)$$

Constructing a scheme of desired order (2/3)

Our target: $a_{+2}p(x_i+2\Delta x)+a_{+1}p(x_i+\Delta x)+a_0p(x_i)+a_{-1}p(x_i-\Delta x)+a_{-2}p(x_i-2\Delta x)=\frac{\partial^2 p_i}{\partial x^2}+O(\Delta x^3)$

Taylor expansion of the potential in each gridblock w.r.t. x_i:

$$a_{+2}p(x_i+2\Delta x)=a_{+2}\left(p(x_i)+2\Delta x\left.\frac{\partial p}{\partial x}\right|_{x_i}+\frac{4\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i}+\frac{8\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i}+\frac{16\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i}+O(\Delta x^5)\right)$$

$$a_{+1}p(x_i+\Delta x)=a_{+1}\left(p(x_i)+\Delta x\left.\frac{\partial p}{\partial x}\right|_{x_i}+\frac{\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i}+\frac{\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i}+\frac{\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i}+O(\Delta x^5)\right)$$

$$a_0p(x_i)=a_0\quad p(x_i)$$

$$a_{-1}p(x_i-\Delta x)=a_{-1}\left(p(x_i)-\Delta x\left.\frac{\partial p}{\partial x}\right|_{x_i}+\frac{\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i}-\frac{\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i}+\frac{\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i}+O(\Delta x^5)\right)$$

$$a_{-2}p(x_i-2\Delta x)=a_{+2}\left(p(x_i)-2\Delta x\left.\frac{\partial p}{\partial x}\right|_{x_i}+\frac{4\Delta x^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i}-\frac{8\Delta x^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i}+\frac{16\Delta x^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i}+O(\Delta x^5)\right)$$

Summation gives:

$$a_{-2}p(x_i-2\Delta x)+a_{-1}p(x_i-\Delta x)+a_0p(x_i)+a_{+1}p(x_i+\Delta x)+a_{+2}p(x_i+2\Delta x)=$$

$$[a_{-2}+a_{-1}+a_0+a_{+1}+a_{+2}]p(x_i)+[-2a_{-2}-a_{-1}+a_0+a_{+1}+2a_{+2}]\Delta x\frac{\partial p}{\partial x}(x_i)$$

$$+[4a_{-2}+a_{-1}+a_0+a_{+1}+4a_{+2}]\frac{\Delta x^2}{2}\frac{\partial^2 p}{\partial x^2}(x_i)+[-8a_{-2}-a_{-1}+a_0+a_{+1}+8a_{+2}]\frac{\Delta x^3}{6}\frac{\partial^3 p}{\partial x^3}(x_i)$$

$$+[16a_{-2}+a_{-1}+a_0+a_{+1}+16a_{+2}]\frac{\Delta x^4}{24}\frac{\partial^4 p}{\partial x^4}(x_i)+O\left(\Delta x^5\frac{\partial^5 p}{\partial x^5}(x_i)\right)$$

Constructing a scheme of desired order (3/3)

We want all derivatives to disappear except for the 2nd derivative:

$$p(x_i)[a_{+2}+a_{+1}+a_0+a_{-1}+a_{-2}]=0$$

$$\frac{\partial p(x_i)}{\partial x}\Delta x[2a_{+2}+a_{+1}-a_{-1}-2a_{-2}]=0$$

$$\frac{\partial^2 p(x_i)}{\partial x^2}\frac{\Delta x^2}{2}[4a_{+2}+a_{+1}+a_{-1}+4a_{-2}]=-\frac{\partial^2 p(x_i)}{\partial x^2}$$

$$\frac{\partial^3 p(x_i)}{\partial x^3}\frac{\Delta x^3}{6}[8a_{+2}+a_{+1}-a_{-1}-8a_{-2}]=0$$

$$\frac{\partial^4 p(x_i)}{\partial x^4}\frac{\Delta x^4}{24}[16a_{+2}+a_{+1}-a_{-1}-16a_{-2}]=0$$

This result in the coefficient matrix

$$\begin{bmatrix}1&1&1&1&1\\2&1&0&-1&-2\\4&1&0&1&4\\8&1&0&-1&8\\16&1&0&1&16\end{bmatrix}\begin{bmatrix}a_{+2}\\a_{+1}\\a_0\\a_{-1}\\a_{-2}\end{bmatrix}=\begin{bmatrix}0\\0\\-2/\Delta x^2\\0\\0\end{bmatrix}$$

from which we find the following: $a_{+2}=a_{-2}=\frac{1}{12}\frac{1}{\Delta x^2},\ a_{+1}=a_{-1}=-\frac{16}{12}\frac{1}{\Delta x^2},\ a_0=\frac{30}{12}\frac{1}{\Delta x^2}$

that by construction satisfies

$$\frac{a_{+2}p(x_i+2\Delta x)+a_{+1}p(x_i+\Delta x)+a_0p(x_i)+a_{-1}p(x_i-\Delta x)+a_{-2}p(x_i-2\Delta x)}{\Delta x^2}=-\frac{\partial^2 p(x_i)}{\partial x^2}+O(\Delta x^3)$$

Constructing a scheme of a desired order (1/3)

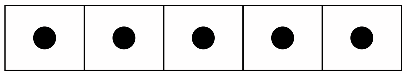
Suppose we want to approximate $-\lambda\frac{\partial^2 p}{\partial x^2}=q$ in gridblock i using (at least) a third order accurate approximate discretisation. How do we do this?

Taylor series expansion of the potential around gridblock i gives:

$$p(x_i+\alpha)=p(x_i)+\alpha\left.\frac{\partial p}{\partial x}\right|_{x_i}+\frac{\alpha^2}{2}\left.\frac{\partial^2 p}{\partial x^2}\right|_{x_i}+\frac{\alpha^3}{6}\left.\frac{\partial^3 p}{\partial x^3}\right|_{x_i}+\frac{\alpha^4}{24}\left.\frac{\partial^4 p}{\partial x^4}\right|_{x_i}+O(\alpha^5)$$

The key-idea is to combine the information of a sufficient amount of neighbouring grid blocks in a way that only the desired derivatives remain.

Construction of the solution using 5 support points: $x_{i-2}, x_{i-1}, x_i, x_{i+1}$ and x_{i+2}



i-2 i-1 i i+1 i+2

Combine the potentials within the five gridblocks around x_i such that this combination yields a 3rd order approximation of the second derivative:

$$a_{+2}p(x_i+2\Delta x)+a_{+1}p(x_i+\Delta x)+a_0p(x_i)+a_{-1}p(x_i-\Delta x)+a_{-2}p(x_i-2\Delta x)=-\frac{\partial^2 p}{\partial x^2}+O(\Delta x^3)$$

Formulate Pressure (Flow) Equation

- Study yourself ...

Flow type	PDE	Linear?	PDE type
General form	$\frac{\partial}{\partial t}\left(\frac{\phi}{B^f}\right)-\nabla\bullet\left(\frac{K}{B^f\mu^f}[\nabla p^f+\rho^f g\nabla z]\right)=q$ with additional relations: $B^f(p^f),\phi(p^f)$	Non-linear	(Generally) parabolic
Compressible	$(c^f+c^r)\frac{\phi}{B^f}\frac{\partial p}{\partial t}-\nabla\bullet\left(\frac{K}{B^f\mu^f}[\nabla p^f+\rho^f g\nabla z]\right)=q$	Non-linear	Parabolic
Compressible (Bousinesq)	$(c_f+c_r)\phi\frac{\partial p_f}{\partial t}-\nabla\bullet\left(\frac{K}{\mu_f}[\nabla p_f+\rho_f g\nabla z]\right)=B_f q$	(Weakly) Non-linear	Parabolic
Slightly compressible (2)	$(c_f+c_r)\boxed{\phi_0}\frac{\partial p}{\partial t}-\nabla\bullet\left(\frac{K}{\mu_f}[\nabla p_f+\rho_f g\nabla z]\right)=\boxed{B_{f,0}}q$	Linear	Parabolic
Incompressible	$-\nabla\bullet\left(\frac{K}{\mu_f}[\nabla p_f+\rho_f g\nabla z]\right)=\boxed{B_{f,0}}q$	Linear	Elliptic
Compressible gas	$\phi c\mu\frac{\partial\Psi}{\partial t}-\nabla\bullet(K\nabla\Psi)=\frac{2qRT}{M}$	Non-linear	Parabolic

$$c_\phi=+\frac{1}{\phi}\frac{\partial\phi}{\partial p_f}$$

$$\phi=\phi_0\exp\left(+c_\phi[p_f-p_{f,0}]\right)$$

$$c_f=-\frac{1}{V_f}\frac{\partial V_f}{\partial p_f}=\frac{1}{\rho_f}\frac{\partial\rho_f}{\partial p_f}$$

$$\rho_f=\rho_{f,0}\exp\left(+c_f[p_f-p_{f,0}]\right)$$

$$c_f=-\frac{1}{B_f}\frac{\partial B_f}{\partial p_f}=+B_f\frac{\partial}{\partial p_f}\left(\frac{1}{B_f}\right)$$

$$B_f=B_{f,0}\exp\left(-c_f[p_f-p_{f,0}]\right)$$

Formulate Pressure (Flow) Equation

- Study yourself ...

start from this (explain):
$$\phi c \frac{p}{Z}\frac{\partial p}{\partial t} - \nabla \bullet \left(\frac{K}{\mu}\frac{p}{Z} \nabla p \right) = \frac{qRT}{M}$$

Define pseudo potential (with p' arbitrarily chosen):
$$\frac{d\Psi}{dp} = \frac{2p}{\mu Z}, \qquad \Psi(p) = \int_{p'}^{p} \frac{2\tilde{p}}{\mu Z} d\tilde{p}$$

For ideal gas (Z=1) we have:
$$\Psi = \frac{1}{2}\left(p^2\right) - \left(p'\right)^2$$

Pseudo potential gradient vs. pressure gradient:
$$\nabla\Psi = \frac{d\Psi}{dp}\nabla p = \frac{2p}{\mu Z}\nabla p$$

Time derivative pressure vs. time derivative pseudo potential
$$\frac{\partial \Psi}{\partial t} = \frac{d\Psi}{dp}\frac{\partial p}{\partial t} = \frac{2p}{\mu Z}\frac{\partial p}{\partial t}$$

Substitution of [1] and [2] in mass balance [0] gives final expression:
$$\phi c \mu \frac{\partial \Psi}{\partial t} - \nabla \bullet \left(K \nabla \Psi \right) = \frac{2qRT}{M}$$

Compressible gas	$\phi c \mu \frac{\partial \Psi}{\partial t} - \nabla \bullet \left(K \nabla \Psi \right) = \frac{2qRT}{M}$	Non-linear	Parabolic

Some text about PDEs follow ...

you must know them ... study at home!

Explain the Challenges of Pressure (Flow) Equation

- Study yourself ...

General 2nd order PDE:
$$A\frac{\partial^2 c}{\partial \xi^2} + B\frac{\partial^2 c}{\partial \xi \partial \eta} + C\frac{\partial^2 c}{\partial \eta^2} + \left[D\frac{\partial c}{\partial \xi} + E\frac{\partial c}{\partial \eta} - F \right] = 0$$

Discriminant:
$$b = B^2 - 4AC$$

PDE type:
$$b = B^2 - 4AC = \begin{cases} > 0, & hyperbolic \\ = 0, & parabolic \\ < 0, & elliptic \end{cases}$$

Note that if the coefficients A, B and C are not constants and depend on c the (non-linear) 2nd order PDE may be of mixed type. A general parabolic equations may then have elliptic regions, etc. Mixed type PDE's may be mathematically challenging.

Explain the Challenges of Pressure (Flow) Equation

- Study yourself ...

Elliptic equations

Example: incompressible pressure equation
$$-\nabla \cdot (\underline{\underline{\lambda}}_t \cdot \nabla p) = q$$

Characteristics:

- Information (e.g. pressure) travels with infinite speed of propagation through medium. (Changing pressure in Holland is immediately felt in Australia)

- Information (e.g. pressure) adapts instantaneously to a new equilibrium situation (e.g. change in boundary conditions, or change in permeability somewhere in medium)

- Elliptic equation do heavily depend on boundary conditions that need to be described at all of the domain boundaries but do not depend on the initial conditions

Explain the Challenges of Pressure (Flow) Equation

– Study yourself ...

Parabolic equations

Example: compressible pressure equation $(\phi\rho)c_{\text{eff}}\frac{\partial p}{\partial t} - \nabla\cdot\left(\rho\frac{K}{\mu}\cdot\nabla p\right) = \rho q$

Characteristics:

- Information (e.g. pressure) travels with infinite speed of propagation through medium. E.g. changing concentration of salt in Dutch coastal waters is immediately sensed in Australia (albeit the value of the info changes during the time)

- Information (e.g. pressure) does NOT instantaneously adapt to new equilibrium situation. I.e. although the salt added in Holland is immediately sensed in Australia, it will take (a long) time before it is at the same concentration

- Parabolic equation do weakly depend on boundary conditions that need to be described at all of the domain boundaries. Since the behavior of parabolic equations change in time, they also depend on the initial condition.

Explain the Challenges of Pressure (Flow) Equation

– Study yourself ...

Hyperbolic equations - to be discussed later!

Example: saturation equations

1st order: $\frac{\partial S_w}{\partial t} + \left(\frac{u_t}{\phi}\frac{df}{dS_w}\right)\frac{\partial S_w}{\partial x} = 0$

2st order: $\frac{\partial^2 c}{\partial t^2} - v^2\frac{\partial^2 c}{\partial x^2} = 0$

Characteristics:

- Information travels with a finite speed of propagation through the medium. For our case the wave speeds are:

- Information travels from a "source" or upstream boundary only in down-stream direction. So no information downstrream of wave is needed

- Hyperbolic equations require initial conditions (one per wave) and upstream boundary conditions (one per wave).

Note: For the 2nd order equation above, two waves are moving with speed +/- v

Explain the Challenges of Pressure (Flow) Equation

– Study yourself ...

General 2nd order PDE: $A\frac{\partial^2 c}{\partial \xi^2} + B\frac{\partial^2 c}{\partial \xi\partial\eta} + C\frac{\partial^2 c}{\partial \eta^2} + \left[D\frac{\partial c}{\partial \xi} + E\frac{\partial c}{\partial \eta} - F\right] = 0$

Discriminant: $b = B^2 - 4AC$

Class (discriminant)	Example	Speed of propagation	Equilibrium	Initial (IC) and boundary (BC) conditions
Elliptic (b<0)	Laplacian $\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = 0$	Infinite	Instantaneous	2 domain BC's
Parabolic (b=0)	Transient pressure / Diffusion equation $\frac{\partial p}{\partial t} + D\frac{\partial^2 p}{\partial x^2} = 0$	Infinite	Not-instantaneous	1 IC 2 domain BC's
Hyperbolic (b>0)	Wave equation $\frac{\partial^2 c}{\partial t^2} - v^2\frac{\partial^2 c}{\partial x^2} = 0$	Finite	N/A	2 IC + 2 upstream BC's