

Numerical Methods in Subsurface Geoscience Simulations

Denis Voskov: d.v.voskov@tudelft.nl

Delft University of Technology, The Netherlands

May 16, 2023

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

1 Introduction to reservoir simulation

2 Basic numerical methods

3 Single-phase incompressible flow

4 Slightly compressible flow

5 Wells in simulation

6 Connection-list for simulation

7 Basic two-phase transport

8 Two-phase flow and transport

9 Multiphase flash

10 Compositional simulation

What is reservoir simulation?

- Reservoir simulation is the numerical solution of the equations that describe the physical processes of interest on a mesh representing the porous formation
- The predictions (numerical solutions) simulate (mimic) the behavior of the real system
- Predictive reliability:
 - Quality and quantity of available information (static and dynamic)
 - Model assumptions (throughout the workflow)

Representation on simulation grid

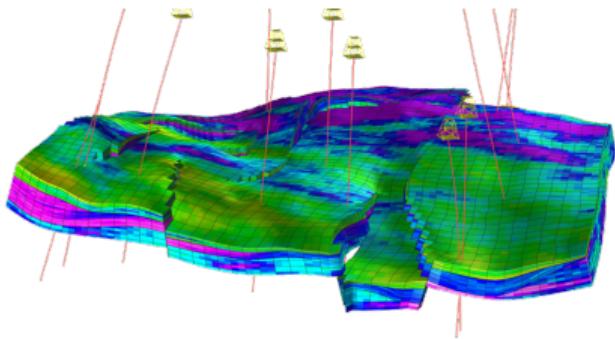


Figure: Real life and model representation¹

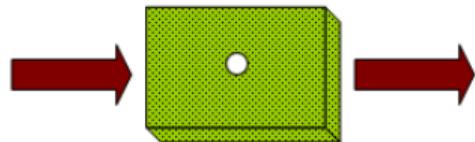
¹ Gringarten; 2002.

Reservoir simulation in 4 steps

- ① Construct the reservoir characterization model - gridding (geometry, properties)
- ② Decide on the equations that govern the physics (modelling assumptions)
- ③ Discretize the equations
 - Time discretization (Backward Euler)
 - Space discretization (Finite Volume)
- ④ Solve for the dynamic properties (pressure, saturation, masses) as a function of space and time

Solving of mass conservation

Reservoir Divided into Blocks



Inflow - Outflow = Accumulation

Flow Rate=Transmissibility (Driving Force)

Pore Volume = Fluid Volume

Forward simulation response

- Main objective – get a similar response to the model with an accurate geological characterization
- Option 1: solve at fine scale
 - Direct solution
 - Multiscale reconstruction
- Option 2: solve at coarser scale:
 - Direct upscaling
 - Data-driven proxies

Two sources of errors: gridding and discretization

- Grid representation (upscaleing):

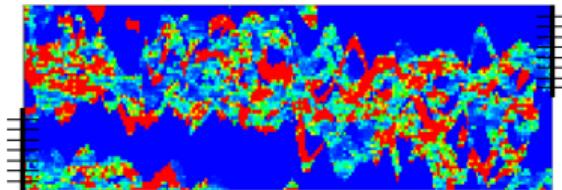
Discrete representation of the reservoir model - geometry and properties on the mesh

- PDE discretization:

Discrete representation of the governing equations on the grid in space and time

Accuracy in upscaling²

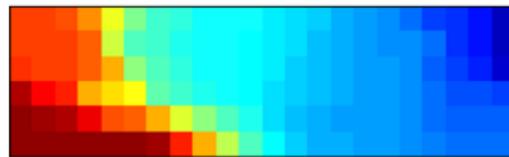
Figure: 73rd layer from SPE 10



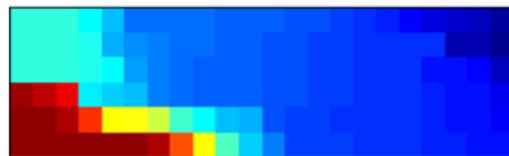
Flow rate for a specified pressure

- Fine scale: $Q = 11.02$
- Standard upscaling: $Q = 4.60$
- Advance upscaling: $Q = 10.77$

Pressure Distribution



Averaged fine



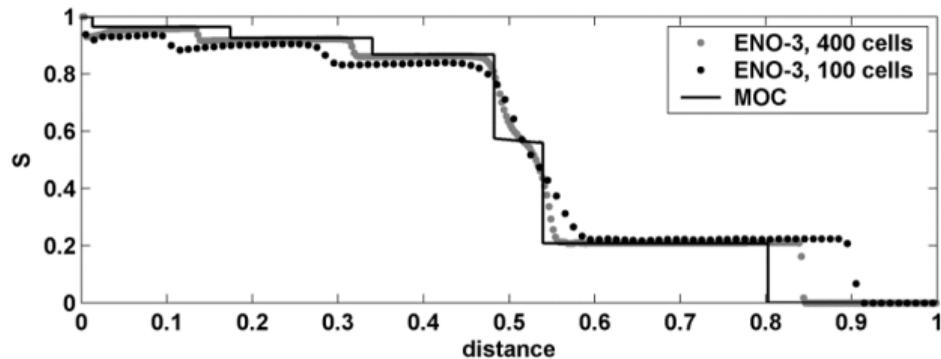
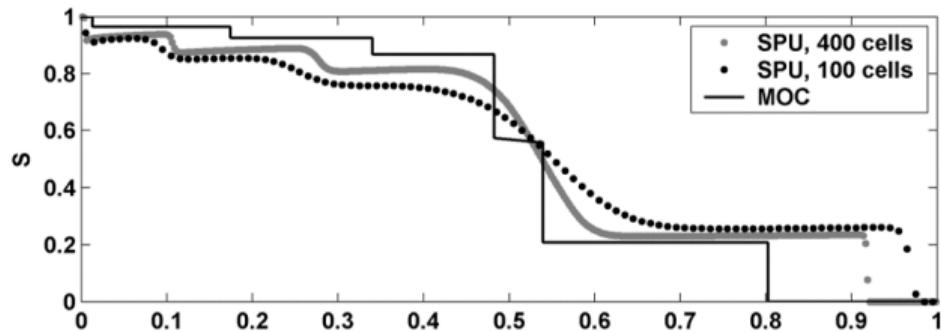
Coarse: standard upscaling



Coarse: advance upscaling

²Durlofsky et al.; 2008

Accuracy in discretization ³



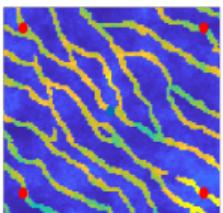
³ Jessen et al.; 2007

Stochastic simulation

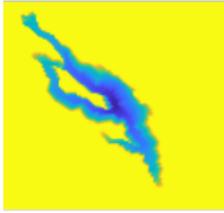
- Main objective – get a representative stochastic response
- Requirements – efficient and robust forward simulation capabilities
- Option 1: high fidelity response
 - Markov-Chain Monte-Carlo
 - Various filters (ensemble Kalman, particle etc.)
- Option 2: solve at a coarser scale:
 - Multi-Level MC
 - Multi-scale clustering
 - Information Theory

Stochastic response in fluvial systems ⁴

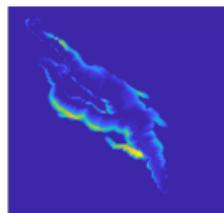
permeability



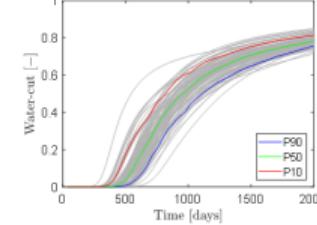
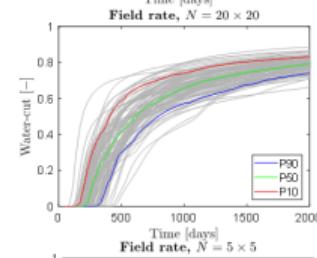
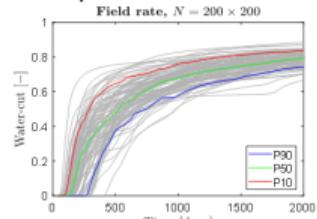
saturation



error

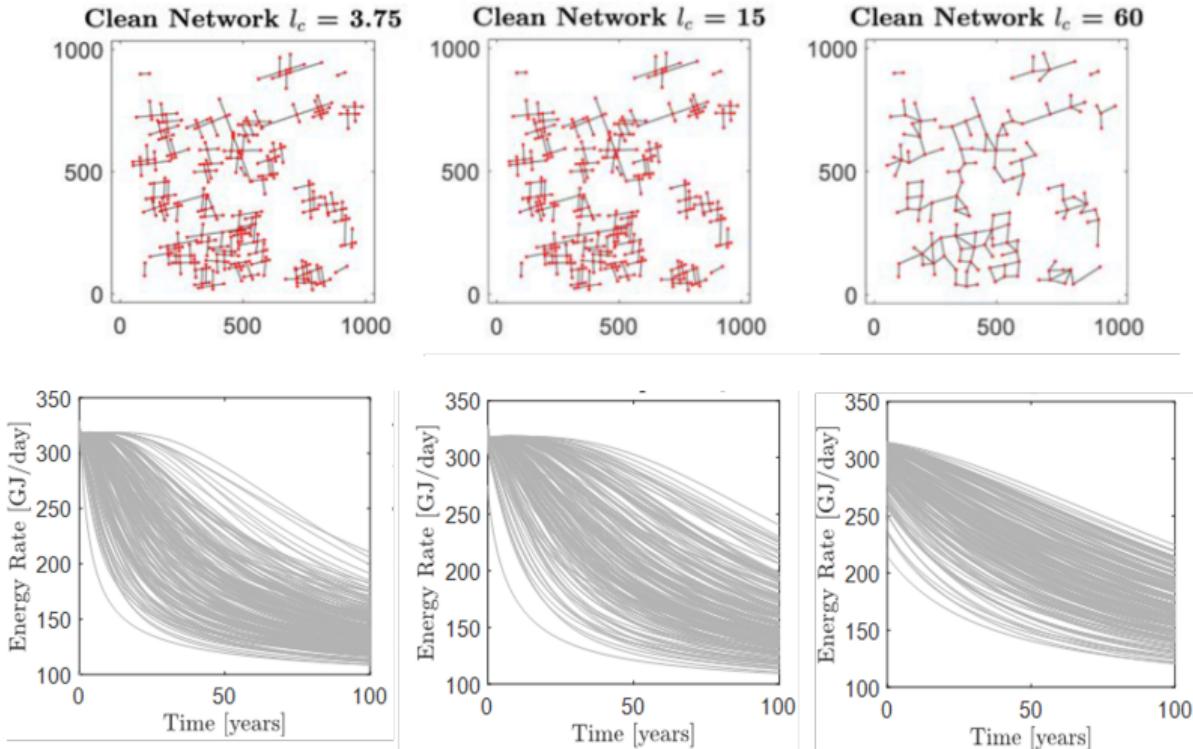


production



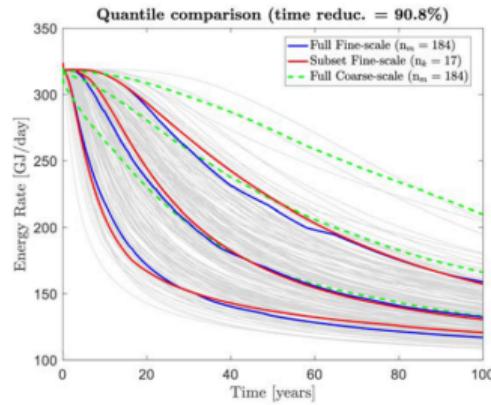
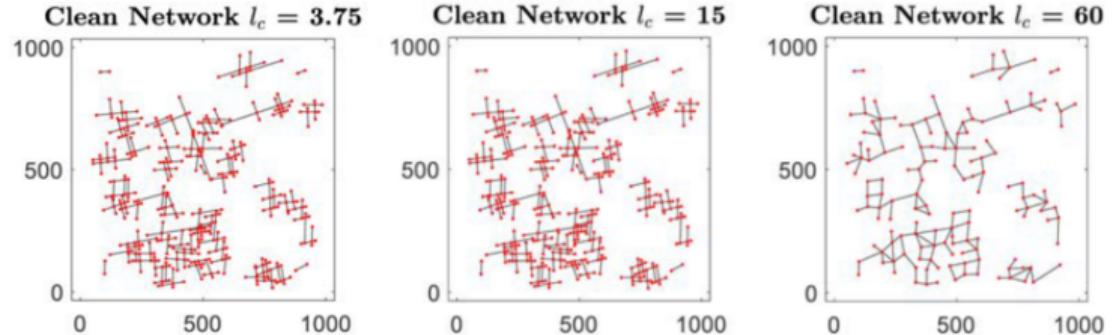
⁴de Hoop et al.; 2017

Stochastic response in fractured system⁵



⁵de Hoop et al.; 2021

Stochastic response in fractured system



- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Norm of vector

Vector norm – real scalar function $\|x\|$

1. $\|x\| \geq 0$; $\|x\| = 0$ only for $x = 0 = (0, 0, \dots, 0)^T$
2. $\|x + y\| \leq \|x\| + \|y\|$ - triangle rule.
3. $\|\alpha x\| = |\alpha| \|x\|$ - α scalar multiplier.

Different norms:

$$\|x\|_2 = \sqrt{|x_1|^2 + \dots + |x_p|^2}; \quad \|x\|_\infty = \max_k |x_k|.$$

Linear matrix

*Linear matrix (or operator) is represented
by square matrix [p x p]*

$$L = \{l_{ik}\} \quad i,k = \overline{1,p}$$

Matrix L operates on vector x as

$$\mathbf{y} = \mathbf{Lx} ; \quad \mathbf{y} = (y_1, \dots, y_p)^T ; \quad \mathbf{x} = (x_1, \dots, x_p)^T$$

$$y_i = \sum_{k=1}^p l_{ik} x_k , \quad i = \overline{1, p};$$

Determinant of matrix

$$\det(\mathbf{A}) = \sum_{k=1}^n a_{ik} \alpha_{ik} = \sum_{k=1}^n a_{kj} \alpha_{kj}$$

expansion on i-th row *expansion on j-th column*

$$\alpha_{ik} = (-1)^{i+k} \det(\mathbf{M}_{ik})$$

\mathbf{M}_{ik} – matrix $(n-1) \times (n-1)$ obtained by exclusion of *i-th* row and *k-th* column

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21}$$

Eigenvalues and eigenvectors

If for λ exists $\mathbf{y} \neq \mathbf{0}$ | $(\mathbf{A} - \lambda \mathbf{I})\mathbf{y} = \mathbf{0}$ then λ is *eigenvalue* and \mathbf{y} – *eigenvector* of matrix \mathbf{A} .

Any λ is the root of characteristic equation:

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0.$$

Any square matrix \mathbf{A} with dimensions $N \times N$ has exactly N eigenvalues.

$S(\mathbf{A}) = \max |\lambda_i|$ - *spectral radius*.

Matrix norm

Matrix norm is defined as:

$$\|L\| = \sup_{x \neq 0} \|Lx\|/\|x\|; \quad \|Lx\| \leq \|L\| \|x\|.$$

Properties of *matrix norms*:

$$\|AB\| \leq \|A\| \|B\|; \quad \|\alpha A\| = |\alpha| \|A\|; \quad \|A+B\| \leq \|A\| + \|B\|.$$

Respective *matrix norms*:

$$\|L\|_2 = \sqrt{\max_{L^T L} \lambda}; \quad \|L\|_\infty = \max_i \left(\sum_{j=1}^p |l_{ij}| \right).$$

Solution of linear systems

System of linear equations

$$\begin{cases} l_{11}x_1 + l_{12}x_2 + \dots + l_{1p}x_p = f_1 \\ l_{21}x_1 + l_{22}x_2 + \dots + l_{2p}x_p = f_2 \\ \dots \dots \dots \dots \\ l_{p1}x_1 + l_{p2}x_2 + \dots + l_{pp}x_p = f_p \end{cases} \Leftrightarrow \begin{bmatrix} l_{11} & l_{12} & \vdots & l_{1p} \\ l_{21} & l_{22} & \vdots & l_{2p} \\ \dots & \dots & \dots & \dots \\ l_{p1} & l_{p2} & \vdots & l_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \end{bmatrix}$$

$$\mathbf{L}\mathbf{x} = \mathbf{f}, \quad \text{if exists} \quad \mathbf{L}^{-1} \Rightarrow \mathbf{x} = \mathbf{L}^{-1}\mathbf{f}$$

Existence requires $D \neq 0$

Cramer's rule

If matrix determinant $D \neq 0 \Rightarrow x_i = D_i / D \quad i = \overline{1, p}.$

D_i is order i -th determinant (replace i -th column by \mathbf{f}).

Number of **arithmetic multiplications** is $N! N.$

If $N = 20$ than the number of operation is $4.8 \cdot 10^{19};$

For processor with speed 10^9 multiplications per second
computation time is about **1500 years!**

Gaussian elimination

Successive exclusion of unknowns:

$$l_{11} \neq 0 \Rightarrow l_{11}x_1 + l_{12}x_2 + \dots + l_{1p}x_p = f_1 \quad / \quad l_{11} \times l_{k1}$$

Subtract this equation from the rest of
equations, we get:

$$\left\{ \begin{array}{l} x_1 + m_{12}x_2 + \dots + m_{1p}x_p = g_1 \\ m_{22}x_2 + \dots + m_{2p}x_p = g_2 \\ \cdots \cdots \cdots \\ m_{p2}x_2 + \dots + m_{pp}x_p = g_p \end{array} \right.$$

Gaussian elimination: triangular form

Repeat
procedure for
the rest of rows

$$\left\{ \begin{array}{l} x_1 + n_{12}x_2 + n_{13}x_3 + \dots + n_{1p}x_p = h_1 \\ x_2 + n_{23}x_3 + \dots + n_{2p}x_p = h_2 \\ \dots \dots \dots \dots \dots \dots \dots \dots \\ x_{p-1} + n_{p-1,p}x_p = h_{p-1} \\ x_p = h_p \end{array} \right.$$

Finally, we've got:

$$\mathbf{Nx = h} : \mathbf{h} = (h_1, \dots, h_p)^T ; \quad \mathbf{N} = \begin{bmatrix} 1 & n_{12} & \dots & n_{1p} \\ 0 & 1 & \dots & n_{2p} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{x} = (x_1, \dots, x_p)^T ;$$

Gaussian elimination: back substitution

$$x_p = h_p, \quad x_{p-1} = h_{p-1} - n_{p-1,p} x_p, \quad \dots .$$

The most expensive in terms of operations is the first part - triangulation. However, compared to Cramer's rule the computational cost is only around $2/3 N^3$ operations.

It is important for Gaussian elimination that diagonal element is not equal to 0 on every step of elimination. In numerical implementation of Gaussian elimination, it is better to rearrange the matrix to get a large number for division (pivoting).

Excercise 1: Gaussian elimination

Solve system of linear equations using Gaussian elimination:

$$\begin{cases} 4x - y + z = 3 \\ 2x + 5y - z = 2 \\ x - 2y + 6z = 5 \end{cases}$$

Truncation errors

$\mathbf{L}\mathbf{x} = \mathbf{f}$; \mathbf{f} is exact RHS, \mathbf{x} is exact solution.

We find $\tilde{\mathbf{f}} = \mathbf{f} + \Delta\mathbf{f}$ and find approximate solution

$$\tilde{\mathbf{x}} = \mathbf{L}^{-1}\tilde{\mathbf{f}}, \text{ where } \tilde{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{x} = \mathbf{L}^{-1}(\mathbf{f} + \Delta\mathbf{f})$$

$$\|\Delta\mathbf{x}\| \leq \|\mathbf{L}^{-1}\| \|\Delta\mathbf{f}\| - \text{absolute error}$$

$$\delta(\mathbf{x}) = \frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{L}^{-1}\| \|\mathbf{L}\| \delta(\mathbf{f}) \text{ relative error}$$

$$r = \|\mathbf{L}^{-1}\| \|\mathbf{L}\| - \text{conditioning number of system}$$

Exercise 2: truncation error

Example: find minimal root of equation $u^2 - 140u + 1 = 0$.

$$u = 70 - \sqrt{4899} ; \sqrt{4899} = 69.9929 \dots \approx 69.99$$

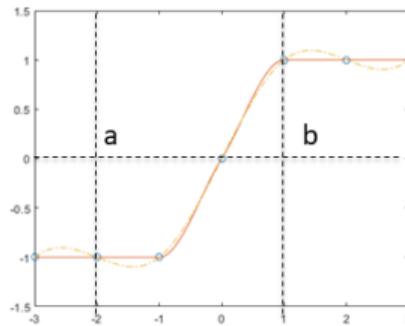
$$u \approx 70 - 69.99 = \boxed{0.01} ;$$

better solution:

$$\begin{aligned} u &= \frac{70^2 - (\sqrt{4899})^2}{70 + (\sqrt{4899})} \approx \frac{4900 - 4899}{70 + 69.99} = \\ &= \frac{1}{139.99} \approx \boxed{0.007143}. \end{aligned}$$

Nonlinear problem

- Numerical function is defined based on the parameter x . Need to find $f(x) = 0$ in some interval $x \in [a, b]$.
- The problem has a unique solution only when $f(x)$ is continuous and monotone.



Bisection method

We need to solve equation: $f(x) = 0, x \in [a, b]$.

Assume $f(a) < 0, f(b) > 0$ and build iteration process as:

$$a_0 = a, b_0 = b; f(a_0) < 0, f(b_0) > 0 \Rightarrow x_0 = \frac{1}{2}(a_0 + b_0)$$

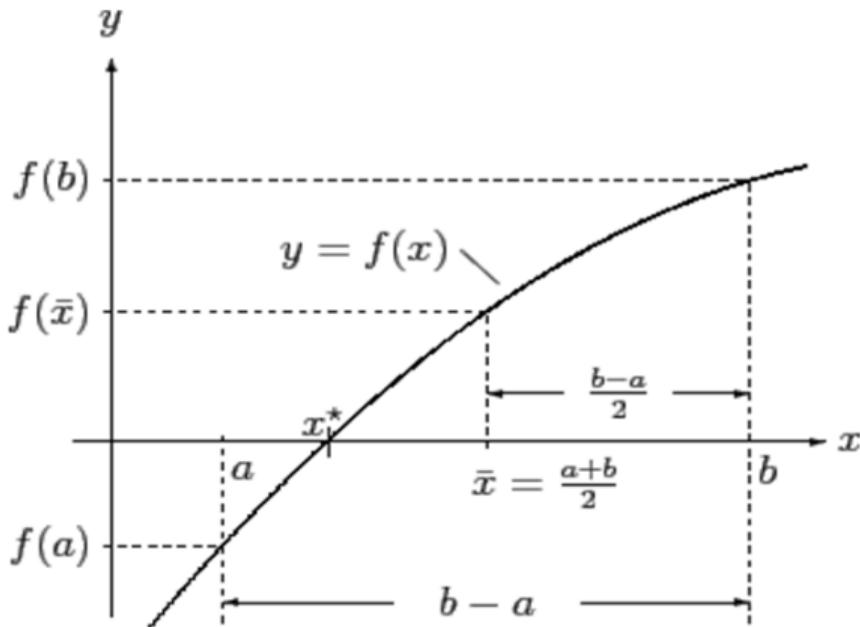
$$f(x_0) < 0 \Rightarrow a_1 = x_0, b_1 = b_0;$$

$$f(x_0) > 0 \Rightarrow a_1 = a_0, b_1 = x_0.$$

Sequence $x_k = \frac{1}{2}(a_k + b_k)$ converge to the root x^* ;

The error of approximation $< \varepsilon = 2^{-(k+1)}(b_0 - a_0)$.

Bisection method (cont.)



Exercise 3: bisection method

Solve equation $f(x) = \sin x - 1/2 x = 0$.

$$a = \pi/2, b = \pi; f(a) > 0, f(b) < 0.$$

N	a	b	c	$f(c)$	
1	1.57075	3.1415	2.356125	-0.47091	<0
2	1.57075	2.356125	1.963438	-0.05782	<0
3	1.57075	1.963438	1.767094	0.097249	>0
4	1.767094	1.963438	1.865266	0.024323	>0
5	1.865266	1.963438	1.914352	-0.01561	<0
6	1.865266	1.914352	1.889809	0.004641	>0
7	1.889809	1.914352	1.90208	-0.00541	<0
8	1.889809	1.90208	1.895944	-0.00037	<0
9	1.889809	1.895944	1.892876	0.002141	>0
10	1.892876	1.895944	1.89441	0.000887	>0
11	1.89441	1.895944	1.895177	0.000259	>0
12	1.895177	1.895944	1.895561	-5.5E-05	<0
13	1.895177	1.895561	1.895369	0.000102	>0
14	1.895369	1.895561	1.895465	2.4E-05	>0
15	1.895465	1.895561	1.895513	-1.5E-05	<0

Successive Substitution iterations

Successive substitution method is based on the following:

We need to solve $f(x)=0$, $x \in [a,b]$

Replace this equation by another equation in form $x=g(x)$,
 $x \in [a,b]$, such so they equivalent $x=g(x) \Leftrightarrow f(x)=0$.

$$f(x)=0 \quad \mapsto \quad x = g(x)$$

$$x_{n+1} = g(x_n)$$

Trivial example: $g(x)=f(x) - x$ and solve $x_{n+1}=g(x_n)$

Convergence of SS iterations

Sequence $x_{k+1} = \varphi(x_k)$ guaranteed to converge if:

Condition 1 For any point x , belongs to $[a,b]$, $g(x)$ also belongs to $[a,b]$.

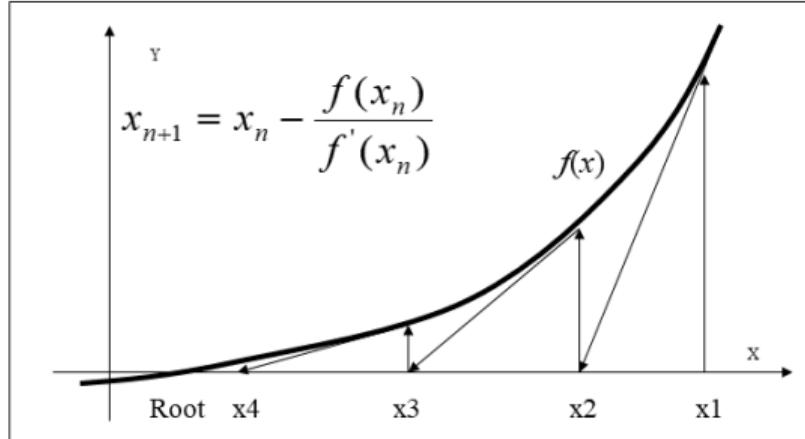
Condition 2 For any two points x_1 and x_2 from $[a,b]$ we have following inequality $|g(x_1) - g(x_2)| \leq q|x_1 - x_2|$ with $q = \text{const}$ and $q < 1$.

$$|g'(x)| \leq C < 1 - \text{sufficient condition}$$

Newton's method

The most efficient method is based on derivative:

$$f(x) = 0 \quad \mapsto \quad x = x - \frac{f(x)}{f'(x)}$$



Exercise 4: Newton's vs SS methods

$$f(x) = \sin x - 1/2 \quad x=0 ; \quad x_0 = 3\pi/4 .$$

Successive substitution method: $x_{n+1} = 2 \sin x_n ;$

Newton's method: $x_{n+1} = x_n - \frac{\sin x_n - x_n / 2}{\cos x_n - 1/2} .$

N	ITERATION	NEWTON	
1	1.414312	0.280625	1.965997 -0.06008
2	1.975563	-0.06859	1.89811 -0.00215
3	1.838389	0.045216	1.895498 -3.2E-06
4	1.92882	-0.02782	1.895494 -7.4E-12
.....			
18	1.895558	-5.2E-05	
19	1.895454	3.31E-05	
20	1.89552	-2.1E-05	
21	1.895478	1.35E-05	

Secant and relaxation methods

Often derivative computation $f'(x_k)$ is hard and/or expensive; we can replace derivative in Newton method

$$x_{k+1} = x_k - [f'(x_k)]^{-1} f(x_k)$$

by approximate derivative (based on definition)

$$x_{k+1} = x_k - \left[\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \right]^{-1} f(x_k)$$

Another possibility is to replace function $y=f(x)$ by straight line that intersects $(x_k, f(x_k))$

$$x_{k+1} = x_k + \alpha f(x_k), \alpha = \text{const}$$

Jacobi method

$$x_1^{v+1} = c_{11}x_1^v + c_{12}x_2^v + \dots + c_{1n}x_n^v + b_1$$

$$x_2^{v+1} = c_{21}x_1^v + c_{22}x_2^v + \dots + c_{2n}x_n^v + b_2$$

.....

$$x_n^{v+1} = c_{n1}x_1^v + c_{n2}x_2^v + \dots + c_{nn}x_n^v + b_n$$

For good convergence, initial guess
should be close to the solution.

Exercise 5: Jacobi solution

$$\begin{cases} 4x - y + z = 3 \\ 2x + 5y - z = 2 \\ x - 2y + 6z = 5 \end{cases}$$

original system

$$\Rightarrow \begin{cases} x = 0.25y - 0.25z + 0.75 \\ y = -0.4x + 0.2z + 0.4 \\ z = -1/6x + 1/3y + 5/6 \end{cases}$$

iterative system

$$\|R\| = \sqrt{(x^v - x^{v-1})^2 + (y^v - y^{v-1})^2 + (z^v - z^{v-1})^2}$$

Convergence
of system:

$$\|R\| \rightarrow 0$$

N	X	Y	Z	R
0	0.000000	0.000000	0.000000	
1	0.750000	0.400000	0.833333	1.190355
2	0.641667	0.266667	0.841667	0.171998
3	0.606250	0.311667	0.815278	0.063053
4	0.624097	0.320556	0.836181	0.028887
16	0.620690	0.318966	0.836207	9.88E-09
17	0.620690	0.318966	0.836207	1.54E-09
18	0.620690	0.318966	0.836207	5.7E-10

Exercise 6: Gauss-Seidel method

We can use the latest information on the update

$$x_1^{v+1} = c_{11}x_1^v + c_{12}x_2^v + \dots + c_{1n}x_n^v + b_1$$

$$x_2^{v+1} = c_{21}\underline{x_1^{v+1}} + c_{22}x_2^v + \dots + c_{2n}x_n^v + b_2$$

.....

$$x_n^{v+1} = c_{n1}\underline{x_1^{v+1}} + c_{n2}\underline{x_2^{v+1}} + \dots + c_{nn}\underline{x_n^{v+1}} + b_n$$

$$\begin{cases} x = 0.25y - 0.25z + 0.75 \\ y = -0.4x + 0.2z + 0.4 \\ z = -1/6x + 1/3y + 5/6 \end{cases}$$

N	X	Y	Z	R
0	0.000000	0.000000	0.000000	
1	0.750000	0.100000	0.741667	1.059514
2	0.589583	0.312500	0.839236	0.283566
3	0.618316	0.320521	0.837121	0.029906
4	0.620850	0.319084	0.836220	0.003049
9	0.620690	0.318966	0.836207	1.54E-08
10	0.620690	0.318966	0.836207	1.54E-09
11	0.620690	0.318966	0.836207	1.64E-10

Convergence criteria

$$\mathbf{x} = \mathbf{Cx} + \mathbf{b}, \quad c_{ii} \neq 0, \quad i = 1, \dots, n.$$

Iterative method converges if \mathbf{C} satisfies condition:

$$\sum_{j=1}^n |c_{ij}| < 1, \quad i = 1, 2, \dots, n.$$

In general, if norm of matrix \mathbf{C} satisfies $\|\mathbf{C}\| < 1$, then solution \mathbf{x} exists, and it is unique; the speed of convergence can be estimated using

$$\|\mathbf{x}^n - \mathbf{x}\| \leq \|\mathbf{C}\|^n \|\mathbf{x}^0 - \mathbf{x}\|; \quad \|\mathbf{x}^n - \mathbf{x}\| \leq \frac{\|\mathbf{C}\|}{1 - \|\mathbf{C}\|} \|\mathbf{x}^n - \mathbf{x}^{n-1}\|.$$

Exercise 7: Nonlinear system of equations

$$\begin{cases} 5x - 6y + 20 \lg x + 16 = 0 \\ 2x + y - 10 \lg y - 4 = 0 \end{cases}$$

$$\begin{cases} x = \varphi(x, y) = 10^{-\frac{1}{4}x + \frac{3}{10}y - \frac{4}{5}} \\ y = \psi(x, y) = 10^{\frac{1}{5}x + \frac{1}{10}y - \frac{2}{5}} \end{cases}$$

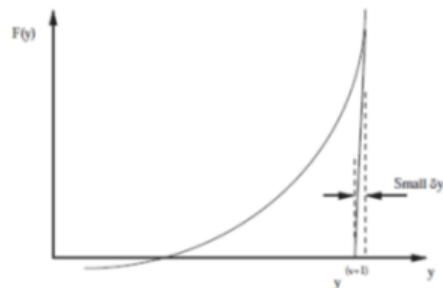
Exercise 8: Newton's method

$$\begin{cases} 3x^2y + y^3 - 1 = 0 \\ x^4 + xy^3 - 1 = 0 \end{cases} \quad x_0 = 1 \quad y_0 = 0.3$$

$$\mathbf{J}(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = -\mathbf{R}(\mathbf{x}^k)$$

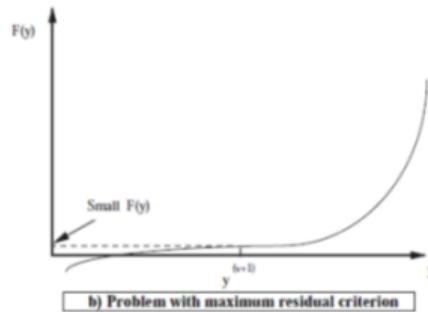
$$J(\mathbf{x}) = \begin{pmatrix} 6xy & 3x^2 + 3y^2 \\ 4x^3 + y^3 & 3xy \end{pmatrix}$$

Convergence criteria



$$\|\Delta x^{k+1}\| < \varepsilon$$

a) Problem with maximum change criterion



$$\|R(x^{k+1})\| < \varepsilon$$

b) Problem with maximum residual criterion

Taylor expansion

Any function $f(x)$ with existed $n+1$ derivatives, can be extended into Taylor series around point x

$$\begin{aligned}f(x + h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \\&+ \frac{h^3}{6}f'''(x) + \dots + \frac{h^n}{n!}f^{(n)}(x) + O(h^n)\end{aligned}$$

$$\begin{aligned}e^{0.2} &= e^0 + 0.2e^0 + \frac{(0.2)^2}{2}e^0 + \frac{(0.2)^3}{6}e^0 + O((0.2)^3) \\&\approx 1 + 0.2 + 0.02 + 0.0013 \approx 1.2213\end{aligned}$$

Numerical differentiation

$$f(x + h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + O(h^3) \quad (1)$$

$$f(x - h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + O(h^3) \quad (2)$$

from equation (2):

$$f'(x) = \frac{f(x) - f(x - h)}{h} + O(h)$$

from equation (1):

$$f'(x) = \frac{f(x + h) - f(x)}{h} + O(h)$$

Numerical differentiation

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + O(h^3) \quad (1)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + O(h^3) \quad (2)$$

from equation (2):

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$$

from equation (1):

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

if subtract

(1)-(2) \Rightarrow

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

if add

(1)+(2) \Rightarrow

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Discrete flow equation for model system

First, we write the mass balance equations for a single phase single component system directly on arbitrary volume in space under Darcy assumption.

Here we introduced:



- ϕ - porosity,
- n - outward normal for ∂D ,
- A - total surface area,
- ρ - density of the fluid.

$$\int_D \frac{\partial}{\partial t} \phi \rho dV + \int_{\partial D} (\rho \mathbf{u}) \mathbf{n} dA + \int_D \tilde{m} dV = 0 \quad (1)$$

\mathbf{u} - Darcy velocity, \tilde{m} - source term [mass per volume per time]

Continuous form of mass balance

Applying the divergence theorem, we will get

$$\int_{\partial D} (\rho \mathbf{u}) \mathbf{n} dA = \int_D \nabla(\rho \mathbf{u}) dV. \quad (2)$$

Substituting Eq. 2 into Eq. 1, we will get

$$\int_D \left(\frac{\partial}{\partial t} \phi \rho + \nabla(\rho \mathbf{u}) + \tilde{m} \right) dV = 0. \quad (3)$$

Since D is arbitrary, we can skip the integral and finally get

$$\frac{\partial}{\partial t} (\phi \rho) + \nabla(\rho \mathbf{u}) + \tilde{m} = 0. \quad (4)$$

Problem statement: single-phase flow

$$\frac{\partial}{\partial t}(\phi\rho) + \operatorname{div}(\rho\mathbf{u}) = -\tilde{m}. \quad (5)$$

Simplifying assumptions:

- single-phase flow,
- horizontal 1D reservoir,
- homogeneous rock,
- fixed position of wells,
- incompressible rock.

Darcy assumptions

In Darcy's assumptions, velocity of fluid follows

$$\mathbf{u} = -\frac{1}{\mu} \mathbf{K} \nabla p, \quad (6)$$

where p is pressure, \mathbf{K} is permeability tensor and μ is fluid viscosity.
Substituting Eq. 6 into Eq. 5 we've got

$$\frac{\partial}{\partial t}(\phi\rho) - \operatorname{div}\left(\rho\frac{1}{\mu}\mathbf{K}\nabla p\right) = -\tilde{m}. \quad (7)$$

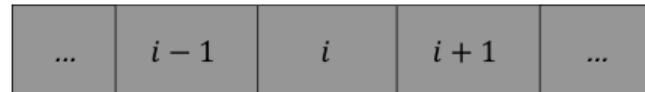
If the system is completely incompressible (density and porosity are constants), we've got

$$\operatorname{div}\left(\rho\frac{1}{\mu}\mathbf{K}\nabla p\right) = \tilde{m}. \quad (8)$$

Single phase flow equation in discretized form

In addition, let's assume K and μ are constants. Then we've got

$$\frac{k_x}{\mu} \frac{\partial^2 p}{\partial x^2} = \frac{\tilde{m}}{\rho} = \tilde{q}. \quad (9)$$



Now assuming discretized reservoir with constant blocks

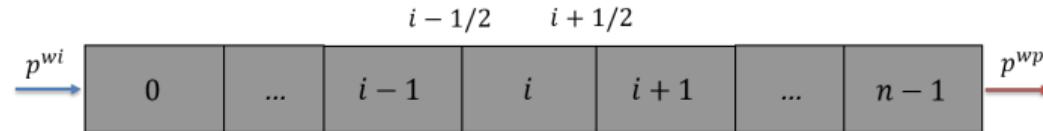
$$\frac{\partial^2 p}{\partial x^2} = \frac{p_{i+1} - 2p_i + p_{i-1}}{(\Delta x)^2}, \quad (10)$$

we finally obtain

$$\frac{k_x}{\mu} \frac{p_{i+1} - 2p_i + p_{i-1}}{(\Delta x)^2} = \tilde{q}. \quad (11)$$

Transmissibility and sources

Let's define $\Psi = k_x / (\mu \Delta x^2)$ and well configuration below:



We assume that wells are controlled by pressure p^w and the rate is

$$\tilde{q} = \Psi(p_i - p^w). \quad (12)$$

The equation for single-phase incompressible flow in internal blocks is

$$\Psi p_{i+1} - 2\Psi p_i + \Psi p_{i-1} = 0. \quad (13)$$

For the first (0) and last ($n - 1$) blocks this equation is written as

$$\Psi p_1 - \Psi p_0 = \Psi(p_0 - p^{wi}), \quad -\Psi p_{n-1} + \Psi p_{n-2} = \Psi(p_{n-1} - p^{wp}). \quad (14)$$

First single phase simulator

The combination of Eqs. 13 and 14 describes a linear system of algebraic equations. Below is an example of a 1D reservoir with 5 blocks.

$$\begin{array}{cc|c|c} 2\Psi & -\Psi & p_0 & \Psi p^{wi} \\ -\Psi & 2\Psi & -\Psi & p_1 \\ -\Psi & 2\Psi & -\Psi & p_2 \\ -\Psi & 2\Psi & -\Psi & p_3 \\ -\Psi & 2\Psi & & p_4 \\ \end{array} = \begin{array}{c} \Psi p^{wp} \end{array}$$

Practical 1: incompressible single-phase 1D simulator

Create the first reservoir simulator using the following assumptions

- both rock and fluid are incompressible,
- assume permeability is constant $k = 160$
- pressures: injection $p = 200$, production $p = 100$ and initial $p = 150$.
- water viscosity $\mu_w = 1$
- reservoir length $L = 1200$

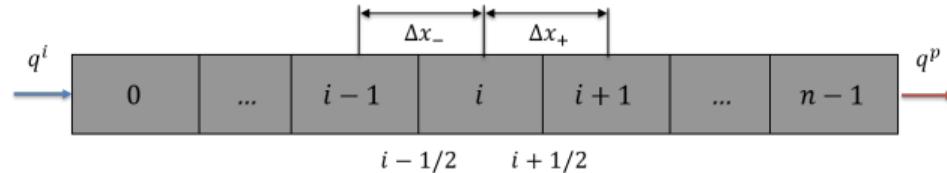
Main task: build an incompressible simulator for a single-phase 1D problem with $n_x = 10$.

Additional task: change the number of blocks $n_x = 100$ and compare two solutions.

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Starting from discrete equation

Let's start from more general 1D discrete model shown below.



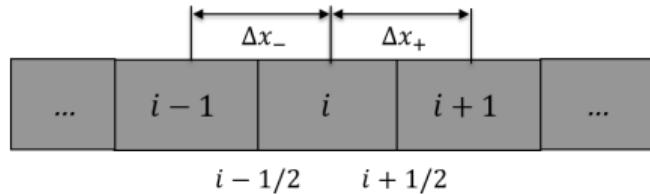
The mass conservation equation states the following

$$\left\{ \begin{array}{l} \text{Net rate} \\ \text{of mass} \\ \text{into } i \end{array} \right\} - \left\{ \begin{array}{l} \text{Net rate} \\ \text{of mass} \\ \text{out of } i \end{array} \right\} + \left\{ \begin{array}{l} \text{Net rate} \\ \text{of mass} \\ \text{from well} \end{array} \right\} = \left\{ \begin{array}{l} \text{Rate of} \\ \text{accumulation} \\ \text{of mass in } i \end{array} \right\}$$

Assuming flow direction from left to right, it can be written as

$$m_{i-1/2} - m_{i+1/2} + m_i^w = \frac{\partial}{\partial t} M_i \quad (15)$$

Convective flux discretization



Now $m_{i-1/2} = (u_x A\rho)_{i-1/2}$ and we can substitute Darcy velocity

$$u_x = -\frac{\kappa}{\mu} \nabla p = -\frac{k_x}{\mu} \frac{\partial p}{\partial x}. \quad (16)$$

Rewrite all terms in discrete form: $(A\rho)_{i-1/2} = (\Delta y \Delta z \rho)_{i-1/2}$,

$$(u_x)_{i-1/2} = -\frac{(k_x)_{i-1/2}}{\mu_{i-1/2}} \left. \frac{\partial p}{\partial x} \right|_{i-1/2} \quad \& \quad \left. \frac{\partial p}{\partial x} \right|_{i-1/2} \approx \frac{p_i - p_{i-1}}{\Delta x_-}. \quad (17)$$

Transmissibility for flow

Assume $\rho = \rho(p)$ and $\Delta x, \Delta y, \Delta z, k_x, \mu$ are constants. Then

$$m_{i-1/2} = -\frac{k_x}{\mu} \frac{p_i - p_{i-1}}{\Delta x} \Delta y \Delta z \rho_{i-1/2}. \quad (18)$$

Next, we can introduce the transmissibility of the interface defined as

$$T = \frac{k_x}{\mu} \frac{\Delta y \Delta z}{\Delta x} \quad \& \quad q_{i-1/2} = -T(p_i - p_{i-1}). \quad (19)$$

Finally, mass passing through the interfaces $i \pm 1/2$ can be defined as

$$m_{i-1/2} = q_{i-1/2} \rho_{i-1/2} = -T(p_i - p_{i-1}) \rho(p_{i-1/2}). \quad (20)$$

$$m_{i+1/2} = q_{i+1/2} \rho_{i+1/2} = -T(p_{i+1} - p_i) \rho(p_{i+1/2}). \quad (21)$$

Accumulation term

Using finite difference Backward Euler approximation we've got

$$t^n \rightarrow t^{n+1} \Rightarrow \frac{\partial M_i}{\partial t} = \frac{M_i^{n+1} - M_i^n}{\Delta t}, \quad \Delta t = t^{n+1} - t^n. \quad (22)$$

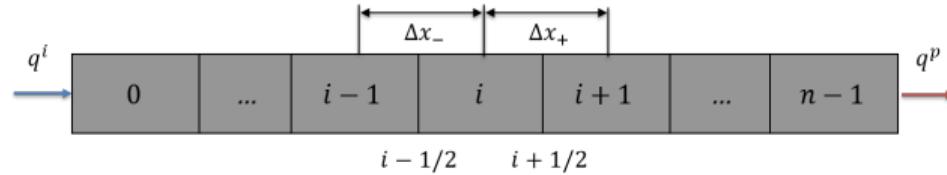
The mass $M_i = \phi_i V_i \rho_i$ and substituting it into Eq. 22 gives

$$\frac{M_i^{n+1} - M_i^n}{\Delta t} = V_i \frac{(\phi_i \rho_i)^{n+1} - (\phi_i \rho_i)^n}{\Delta t}, \quad V_i = \Delta x \Delta y \Delta z \quad (23)$$

Further simplifying assume ρ is constant and $\phi = \phi_0 (1 + c_r(p - p^0))$ where c_r is rock compressibility and p^0 is the reference pressure. Then

$$\frac{\partial M_i}{\partial t} \approx V_i \rho \phi_0 c_r \frac{p_i^{n+1} - p_i^n}{\Delta t}. \quad (24)$$

Finite-volume simulator



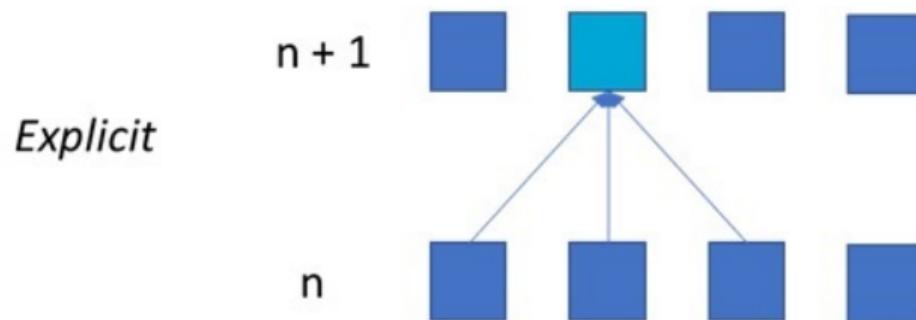
Substituting Eqs. 20, 21 and 24 into Eq. 15, we will get

$$\begin{aligned} \rho_{i+} T(p_{i+1} - p_i) & - \rho_{i-} T(p_i - p_{i-1}) \\ & - \rho_i \frac{V_i \phi_0 c_r}{\Delta t} (p_i^{n+1} - p_i^n) + \rho_i q_i = 0. \end{aligned} \tag{25}$$

Since we assume incompressible fluid, all densities are equal and can be reduced. Notice that except accumulation term, we haven't specified time where p is defined. There are two obvious options for it: explicit (at t^n) or implicit (at t^{n+1}) approximation.

Explicit schema

In this method, the solution at each time step is computed directly from the previous time step, without having to solve a system of equations. The advantage of the explicit scheme method is that it is computationally efficient and easy to implement. However, the method has a stability constraint, which limits the size of the time step that can be used. If the time step is too large, the solution may become unstable and the simulation may diverge.



Explicit schema

In all terms except accumulation $p = p^n$ which translates Eq. 25 to

$$p_i^{n+1} = p_i^n + \frac{\Delta t}{V_i \phi_0 c_r} [T(p_{i+1}^n - 2p_i^n + p_{i-1}^n)]. \quad (26)$$

For boundary gridblocks, Eq. 25 is reduce to

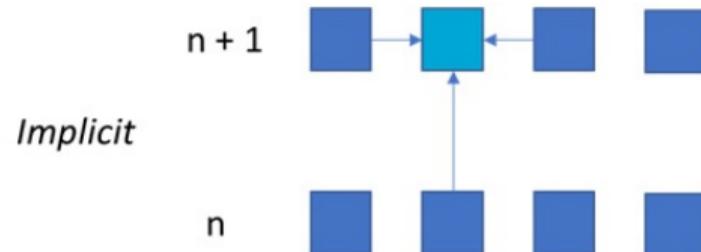
$$\begin{aligned} p_0^{n+1} &= p_0^n + \frac{\Delta t}{V_0 \phi_0 c_r} [T(p_1^n - p_0^n) - q^w], \\ p_{nx-1}^{n+1} &= p_{nx-1}^n + \frac{\Delta t}{V_{nx-1} \phi_0 c_r} [T(p_{nx-2}^n - p_{nx-1}^n) + q^w]. \end{aligned}$$

This scheme is conditionally stable when

$$\Delta t < \theta_c \Delta x^2, \quad \theta_c \approx \frac{V_i}{k_x}. \quad (27)$$

Implicit schema

The implicit scheme method computes the solution at each time step by solving a system of equations. This system is obtained by discretizing the equations in time using backward finite differences, which leads to a system of linear equations that must be solved at each time step. The advantage of the implicit scheme method is that it is unconditionally stable, meaning that it can handle large time steps without becoming unstable. However, it requires solving a system of equations at each time step, which can be computationally expensive.



Implicit schema

In all terms except accumulation $p = p^{n+1}$ which translates Eq. 25 to

$$T(p_{i+1}^{n+1} - 2p_i^{n+1} + p_{i-1}^{n+1}) - \frac{V_i \phi_0 c_r}{\Delta t} p_i^{n+1} = -\frac{V_i \phi_0 c_r}{\Delta t} p_i^n. \quad (28)$$

Rearranging Eq. 28, we've got the following system of equations

$$\begin{aligned} Tp_1^{n+1} - \left(T + \frac{V_1 \phi_0 c_r}{\Delta t}\right) p_0^{n+1} &= q^i - \frac{V_0 \phi_0 c_r}{\Delta t} p_0^n, \\ Tp_{i+1}^{n+1} - \left(2T + \frac{V_i \phi_0 c_r}{\Delta t}\right) p_i^{n+1} + Tp_{i-1}^{n+1} &= -\frac{V_i \phi_0 c_r}{\Delta t} p_i^n, \\ -\left(T + \frac{V_{nx-1} \phi_0 c_r}{\Delta t}\right) p_{nx-1}^{n+1} + Tp_{nx-2}^{n+1} &= q^p - \frac{V_{nx-1} \phi_0 c_r}{\Delta t} p_{nx-1}^n. \end{aligned}$$

This scheme is unconditionally stable (does not depend on Δt).

Structure of linear matrix in incompressible case

Let's assume again incompressible case (both fluids and rock):

$$\begin{vmatrix} T & -T & & \\ -T & 2T & -T & \\ & -T & 2T & -T \\ & & -T & 2T & -T \\ & & & -T & T \end{vmatrix} \begin{vmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{vmatrix} = \begin{vmatrix} q^i \\ \\ \\ \\ q^P \end{vmatrix}$$

The difference with Eq. 28 is the compressibility term $V_i\phi c_r/\Delta t$ in the diagonal and RHS.

First single phase simulator

If you assume $V = \Delta x \Delta y \Delta z$ and divide the Eq. 28 by the volume

$$T/V = \frac{k_x}{\mu} \frac{\Delta y \Delta z}{\Delta x} / (\Delta x \Delta y \Delta z) = \frac{k_x}{\mu \Delta x^2} = \Psi. \quad (29)$$

$$\begin{vmatrix} \Psi & -\Psi & & \\ -\Psi & 2\Psi & -\Psi & \\ -\Psi & 2\Psi & -\Psi & \\ -\Psi & 2\Psi & -\Psi & \\ -\Psi & \Psi & & \end{vmatrix} \begin{vmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{vmatrix} = \begin{vmatrix} \tilde{q}^i \\ \tilde{q}^P \end{vmatrix}$$

Practical 2: compressible single-phase simulator

Create 1D reservoir simulator using the following assumptions:

- rock is compressible with $\phi_0 = 0.2$ and $c_r = 10^{-5}$,
- assume viscosity is constant $\mu = 1$,
- take permeability $k = 1000$,
- initial pressure is $p = 150$,
- left and right boundary conditions are rate controlled at 0.4 pore-volume.

Main task: build a simulator with compressible rock for a single-phase 1D problem with dimensions $1200 \times 10 \times 10$ m and use $\Delta t = 10^{-4}$; increase Δt and check solution.

Additional task: find the solution for a model with two equal regions of permeability $k_1 = 1000$ and $k_2 = 10$; check sensitivity to the ratio between k_1 and k_2 .

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

What is inside well model?

This well model must account for the geometric characteristics of the well (radius, type of completion zone, location within the block, well inclination, etc.) and the reservoir properties in the vicinity of the well (skin, anisotropy, heterogeneities, saturations, etc.). Also if the block contains more than one well, interaction among these wells may have to be considered. Furthermore, while the conservation equations contain flow of individual “components” it is not possible to specify the production of each component independently. The relative flow of different phases depends on the relative mobility of the phase and pressure drawdown. The flow rate of a component depends on its concentration in all of the phases being produced.

Single phase well inflow

The pressure of a well in a block is different from the pressure of the block containing the well. These two pressures are related by assuming single-phase flow in the vicinity of the well.

Peaceman (1978) has shown that the computed the pressure of the block containing the well is not the average pressure of that block. A well model combines the analytical and numerical solutions of a single phase flow problem.

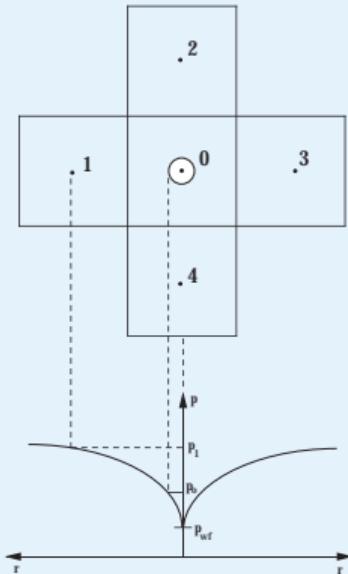


Figure: Well inflow.

The basic well model: pressure

The calculated pressure in block with and without well is different. Let's assume single-phase flow and write pressure distribution around radial source

$$p(r) = p^w + \frac{q^w \mu}{2\pi kh} \left(\ln \frac{r}{r_w} \right) \quad (30)$$

Then for block 0 pressure should be defined

$$p_0 = p^w + \frac{q^w \mu}{2\pi kh} \left(\ln \frac{r_0}{r_w} \right), \quad (31)$$

where r_0 defines radius where pressure from radial flow equate with block pressure p_0 , see Fig. 3.

The basic well model: rates

At the same time, well inflow for the model in Fig. 3 should be equal to inflow from each block

$$q^w = \sum_{i=1}^4 q_i, \quad (32)$$

Here q_i should follow Darcy's assumptions

$$q_i = \frac{kA}{\mu} \frac{p_i - p_0}{a}, \quad (33)$$

where $a = \Delta x = \Delta y$ and $A = ha$. When substitute Eq. 33 into Eq. 32 we've got

$$q^w = \frac{kh}{\mu} \left(\sum_{i=1}^4 p_i - 4p_0 \right). \quad (34)$$

The basic well model: Well Index

Now assuming radial flow from block 0

$$p_i = p_0 + \frac{q^w \mu}{2\pi kh} \left(\ln \frac{a}{r_0} \right). \quad (35)$$

Substituting Eq. 35 into Eq. 34 and simplifying we finally can find

$$\frac{r_0}{a} = \exp \left(-\frac{\pi}{2} \right) \approx 0.208. \quad (36)$$

That defines

$$q_i = T^w (p_0 - p^w), \quad (37)$$

where

$$T^w = \frac{2\pi kh}{\mu \ln \frac{r_0}{r_w}}, \quad r_0 = 0.208a \quad (38)$$

What is the Skin Effect?

The skin effect (s) refers to the near-wellbore damage that reduces the productivity of a well. This damage can be caused by a variety of factors such as drilling fluids, completion practices, natural formation damage, or perforation damage. The skin effect can be quantified as a dimensionless parameter called the skin factor, which is defined as the ratio of the pressure drop across the damaged zone around the wellbore to the pressure drop in the undamaged reservoir.

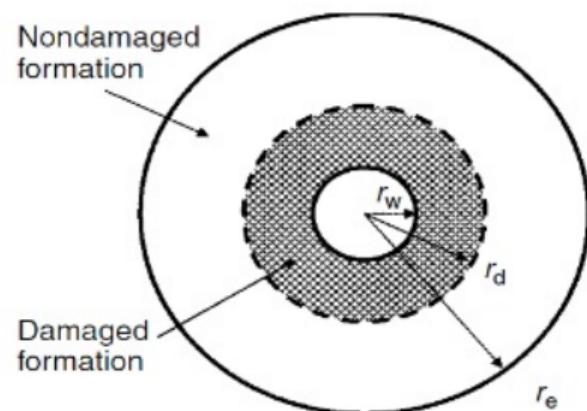


Figure: Skin effect

Overall, including the skin factor in the well model is important to accurately simulate the near-wellbore behavior and help the history match the well productivity.

Computing Skin Effect

The skin effect can be included in the well model by modifying the transmissibility term T^w to account for the near-wellbore damage that reduces the permeability.

The modified transmissibility, T_s , is defined as:

$$T_s = \frac{2\pi kh}{\mu \ln \frac{r_0}{r_w} + s}, \quad r_0 = 0.208a, \quad (39)$$

where s is the skin factor.

By adding the skin factor to the denominator of the transmissibility term, we can account for the near-wellbore damage and calculate the well productivity more accurately.

- A positive skin factor means that the well productivity is reduced due to the near-wellbore damage
- a negative skin factor implies that the well productivity is enhanced due to some stimulation or recovery technique applied near the wellbore.

Slightly compressible fluid

Let's assume now that ϕ is constant and $\rho = \rho_0 [1 + c_f(p - p^0)]$ where c_f is fluid compressibility and p^0 is the reference pressure. Then

$$\frac{\partial M_i}{\partial t} \approx V_i \rho_0 \phi c_f \frac{p_i^{n+1} - p_i^n}{\Delta t}, \quad (40)$$

and conservation equations can be written as

$$T(p_i - p_{i+1}) - T(p_{i-1} - p_i) + \rho_i \frac{V_i \phi c_f}{\Delta t} (p_i^{n+1} - p_i^n) + \rho_i q_i = 0. \quad (41)$$

In compressible liquid case, transmissibility T start depending on ρ as

$$T = \rho(p) \frac{k_x}{\mu} \frac{\Delta y \Delta z}{\Delta x}. \quad (42)$$

Practical 3: compressible single-phase simulator with wells

Create 1D reservoir simulator using the following assumptions:

- fluid is compressible with $\rho_0 = 1$, $c_f = 10^{-3}$,
- assume viscosity is constant $\mu = 1$,
- take permeability $k = 1000$,
- initial pressure is $p = 150$,
- left boundary condition is an injection of 0.4 pore-volume,
- right boundary condition is the fixed $p = 100$.

Main task: build simulator with compressible liquid for single-phase 1D problem with dimensions $1200 \times 10 \times 10$ m and well added. Take the radius of well $r_w = 0.15$.

Additional task: introduce an injection well in the middle of the reservoir and two production wells in the first and last blocks; all wells operate at the same controls.

Multiphase flow with well

All derivations were done under the assumption of single phase flow in the reservoir. The well flow equation is :

$$q_j^w = T_j^w (p_j - p^w) \quad (43)$$

This equation can be written for each block in which the well is completed. The well transmissibility is defined by :

$$T_j^w = \frac{2\pi k \Delta z}{\left[\ln \left(\frac{r_o}{r_w} \right) + s \right]} \left(\frac{k_{rj}}{\mu_j} \right) = WI \lambda_j \quad (44)$$

Accurate mobility evaluation

The well mobility contains both pressure and saturation dependent terms.

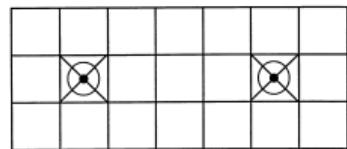
- Size of the block is relatively small - the mobility can be computed using the block pressure and saturation.
- Size of the block is big - the use of block average saturations can lead to large errors in the prediction of WOR and GOR when sharp saturation gradients exist (water or gas coning).

There are three approaches to handling such problems associated with gas and water coning:

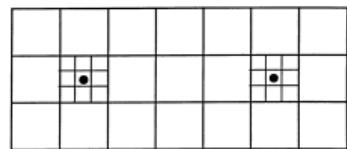
- grid refinement,
- pseudo relative permeabilities,
- coning correlations.

Grid Refinement

The most accurate approach to handling coning problems is through the use of small grid blocks near the wells, which can be "blended" into larger blocks in regions of low activity.



(a) Hybrid LGR



(b) Cartesian LGR

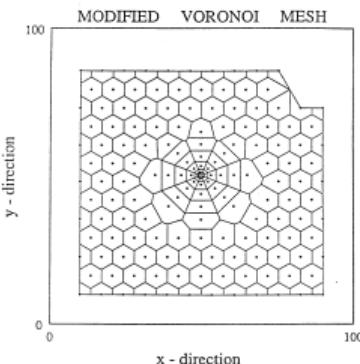


Figure: Different types of local grid refinement (Nacul *et al.*, 1990) and modified Voronoi mesh (Palagi, 1992).

Pseudo Relative Permeability

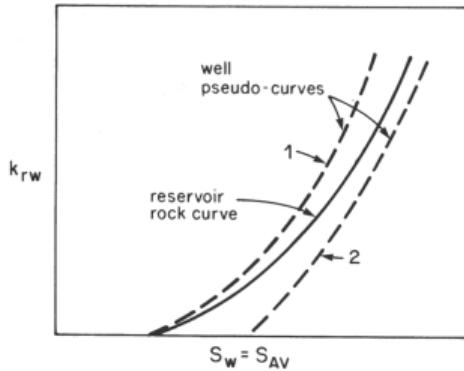


Figure: (1) Bottom water drive, (2) Edge water drive

Potential issues:

- ① Large amount of work is required in a full field study,
- ② Pseudo functions depend on field development and history.

Coning Correlations

Chappelear and Hirasaki (1976) discuss the use of oil–water coning correlations in areal models and Addington (1981) discusses the use of gas coning correlations. These correlations predict the W/O and G/O contacts and the well WOR and GOR are computed from the relative exposure of the perforated interval to water and gas. Nolen (1990) discusses some problems associated with the implementation of this approaches.

Well Equation as a Constraint

Let us consider the case when the total rate is specified

$$q_T^w - \sum_j (T_j^w) (p_i - p^w) = 0 \quad (45)$$

This equation along with the additional unknown p_i^w can be added to our set of nonlinear equations, or we can solve for p_i^w and eliminate this unknown from our flow equations:

$$p^w = \frac{\sum_j (T_j^w p_i) - q_T^w}{\sum_j (T_j^w)} \quad (46)$$

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Connection-based simulators

Modern simulators not anymore rely on particular grid. Instead they use graph-based connection list to represent the grid inside the simulator.

List of existing connection-based simulators:

- Intesect (IX, Schlumberger-Chevron)
- Nexus (Landmark-Halliburton)
- Automatic Differentiation General Purpose Research Simulator (ADGPRS, Stanford University)
- Delft Advanced Research Terra Simulator (DARTS, TU Delft)
- and more are coming

Flux in neighboring blocks

Let rearrange Eq. 41 and write it for two neighboring blocks (ignore source term) :

$$T(p_i^{n+1} - p_{i+1}^{n+1}) + T(p_i^{n+1} - p_{i-1}^{n+1}) + \rho_i \frac{V_i \phi c_f}{\Delta t} p_i^{n+1} = \rho_i \frac{V_i \phi c_f}{\Delta t} p_i^n, \quad (47)$$

$$T(p_{i+1}^{n+1} - p_{i+2}^{n+1}) + T(p_{i+1}^{n+1} - p_i^{n+1}) + \rho_i \frac{V_i \phi c_f}{\Delta t} p_i^{n+1} = \rho_i \frac{V_i \phi c_f}{\Delta t} p_i^n. \quad (48)$$

Here flux between blocks i and $i + 1$ adds T to block i in the first equation written for block i and subtracts it in the second equation written for block $i + 1$. Similarly, it adds T for block $i + 1$ in the second equation written for block $i + 1$ and subtracts from block $i + 1$ in the first equation written for block i . In general, the matrix M is modified by one flux as:

$$M[i, i] += T_k, \quad M[i, j] -= T_k, \quad M[j, j] += T_k, \quad M[j, i] -= T_k, \quad (49)$$

where i and j are neighboring blocks and k is the number of interface.

Connection list: example and main concept

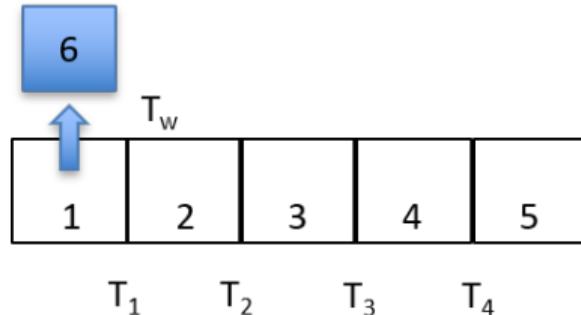


Figure: Small 1D reservoir

This model can be described by a set of cell properties including volumes and porosity. In addition, you need to know connection list with connectivity graph and corresponding transmissibilities:

$$\{1, 2 : T_1\}, \{2, 3 : T_2\}, \{3, 4 : T_3\}, \{4, 5 : T_4\}, \{1, 6 : T_w\}$$

Assemble of flow matrix: diagonal

The RHS and flow matrix can be assembled in 3 simple steps (loops):

1. Add accumulation term and diagonal entries for matrix in the loop over grid blocks :

- Account for accumulation entries as:

$$i : f_i = \rho_i \frac{V_i \phi c_f}{\Delta t} p_i^n \quad (50)$$

- Fill diagonal part of the Matrix \mathbf{M} :

$$i : M_{i,i} = \rho_i \frac{V_i \phi c_f}{\Delta t} \quad (51)$$

Assemble of flow matrix: off-diagonal

2. Loop over $k \rightarrow \{m, p\}$ and account for $Q_k = T_k(p_m - p_p)$ between connected blocks.
Fill off-diagonal part of the matrix:

$$\begin{aligned} M_{m,m} &= M_{m,m} + T_k, & M_{m,p} &= M_{m,p} - T_k, \\ M_{p,p} &= M_{p,p} + T_k, & M_{p,m} &= M_{p,m} - T_k. \end{aligned} \tag{52}$$

3. Loop over wells and account for $Q_w = T_w(p_w - p_i)$ between block and well. Fill the diagonal part of the matrix and RHS:

$$f_i = f_i + T_w p_w, \quad M_{i,i} = M_{i,i} + T_w. \tag{53}$$

Practical 4: construct connection-based simulator

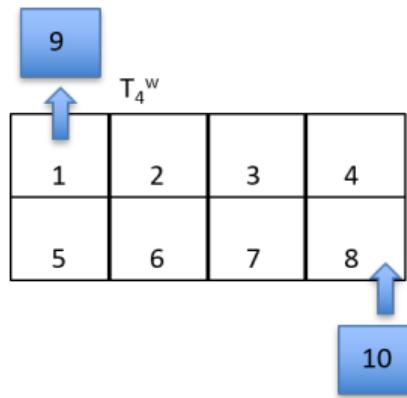
Main task: construct 1D reservoir simulator based on the connection list.

Few practical suggestions on how to change your existing simulator:

- For connection list, introduce two arrays: $\text{block_m} = [0, 1, \dots, n_b - 1]$ and $\text{block_p} = [1, 2, \dots, n_b]$;
- use the same array (Ψ or T) for transmissibility;
- fill compressibility term in diagonal using a loop over grid blocks;
- fill off-diagonal part of flow matrix using a loop over connections;
- Debug your results against the existing 1D simulator by comparing RHS and flow matrix.

How to switch to 2D reservoir now?

Change the connection list - that is all, no need to change the code!



$T_1: 1,2$
 $T_2: 2,3$
 $T_3: 3,4$
 $T_4: 5,6$
 $T_5: 6,7$
 $T_6: 7,8$
 $T_7: 1,5$
 $T_8: 2,6$
 $T_9: 3,7$
 $T_{10}: 4,8$
 $T_{11}: 9,1$
 $T_{12}: 10,8$

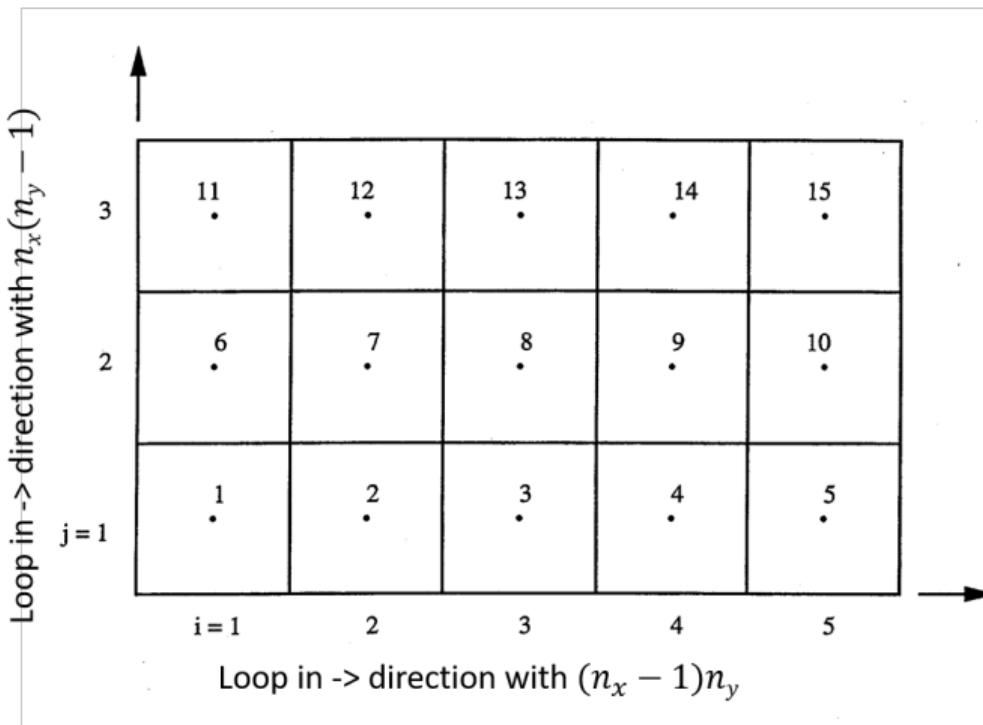
Figure: Small 2D reservoir

	1	2	3	4	5	6	7	8
1	x	x			x			
2	x	x	x			x		
3		x	x	x			x	
4			x	x				x
5	x				x	x		
6		x			x	x	x	
7			x		x	x	x	
8				x		x	x	x

x non-zero elements

Figure: Flow matrix for 2D reservoir

Connection list for 2D reservoir: generic numeration



Transmissibility evaluation

Transmissibility in the general case can be split into two parts - part dependent on the properties of the grid and part dependent on the fluid properties:

$$T_{i+1/2} = \left(\frac{k_x A}{\Delta x} \right)_{i+1/2} \left(\frac{\rho}{\mu_j} \right)_{i+1/2} = \Gamma_{i+1/2} \beta_{j,i+1/2}. \quad (54)$$

The grid part of transmissibility can be evaluated as

$$\Gamma_{i+1/2} = \frac{\gamma_i \gamma_{i+1}}{\gamma_i + \gamma_{i+1}}, \quad \gamma_i = A_i \frac{K_i}{\Delta x_i}. \quad (55)$$

The fluid part depends on pressure and can be evaluated as

$$\beta_{i+1/2} = \frac{\beta(p_{i+1}) + \beta(p_i)}{2}, \quad \beta = \frac{\rho(p)}{\mu_j(p)}. \quad (56)$$

Practical 5: 2D connection-based simulator

Main task: translate your 1D code to 2D.

Few practical suggestions on how to change your existing simulator:

- Introduce horizontal and vertical permeability vectors (k_x and k_y)
- Fill your `block_m` and `block_p` arrays in two loops for horizontal and vertical directions.
- Make sure to adjust the position of wells (enumeration is sequential);
- No need to change well treatment or any other pieces of code;
- Adjust plotting procedure: use `np.reshape` and `plt.imshow`;
- Add two injectors to corners and one producer in the middle;
- Introduce heterogeneity to your model (e.g. four regions with different permeability) and make the reservoir asymmetric ($n_x \neq n_y$).

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Two-phase formulation with simplifying assumptions

$$\frac{\partial}{\partial t}(\phi \rho_j s_j) + \operatorname{div}(\rho_j \mathbf{u}_j) + \rho_j \tilde{q}_j = 0, \quad j = o, w. \quad (57)$$

Simplifying assumptions:

- incompressible fluid and rock,
- horizontal 1D reservoir,
- homogeneous rock,
- no wells inside the reservoir,
- immiscible fluids (oil and water).

Fractional flow formulation

$$\phi \frac{\partial s_j}{\partial t} + \operatorname{div}(\mathbf{u}_j) = 0, \quad j = o, w. \quad (58)$$

$$\mathbf{u}_j = -\lambda_j \nabla p, \quad \lambda_j = \frac{k_{rj}}{\mu_j}, \quad j = o, w. \quad (59)$$

$$\mathbf{u}_t = \mathbf{u}_o + \mathbf{u}_w = -\Lambda \nabla p, \quad \Lambda = \lambda_o + \lambda_w \Rightarrow \mathbf{u}_w = \frac{\lambda_w}{\Lambda} \mathbf{u}_t = f \mathbf{u}_t. \quad (60)$$

$$s_o + s_w = 1 \Rightarrow \phi \frac{\partial 1}{\partial t} + \operatorname{div}(\mathbf{u}_o + \mathbf{u}_w) = 0 \Rightarrow \operatorname{div}(\mathbf{u}_t) = 0. \quad (61)$$

Transport equation

We need to solve the following transport equation:

$$\frac{\partial s}{\partial t} + \frac{u_t}{\phi} \frac{\partial f}{\partial x} = 0 \quad (62)$$

where

- s - water saturation
- u_t - total velocity
- ϕ - porosity
- f - fractional flow curve

Explicit discretization

For accumulation term we use backward Euler approximation:

$$\frac{\partial s}{\partial t} = \frac{s_m^{n+1} - s_m^n}{\Delta t} + o(\Delta t), \quad (63)$$

where n and $n + 1$ corresponds to the current and next timestep respectively, and m corresponds to the current gridblock. For flux term we use explicit forward Euler approximation:

$$\frac{\partial f}{\partial x} = \frac{f_m^n - f_{m-1}^n}{\Delta x} + o(\Delta x) \quad (64)$$

Practical 6: explicit transport solver

Properties and simplifications:

- initial and boundary conditions: $s_1^n = 1$ and $s_m^0 = 0$;
- assume viscosity is constant and fractional flow curve is

$$f(s) = \frac{s^2}{s^2 + 0.1(1-s)^2} \quad (65)$$

- both porosity and total velocity are constants.

Main task: build an explicit transport solver for a two-phase 1D problem.

Advanced task: implement advanced nonlinear solvers and test them for larger timesteps; count for nonlinear iterations with different nonlinear solvers and relative permeabilities exponents

Implicit discretization

For accumulation term we use backward Euler approximation:

$$\frac{\partial s}{\partial t} = \frac{s_m^{n+1} - s_m^n}{\Delta t} + o(\Delta t), \quad (66)$$

where n and $n + 1$ corresponds to the current and next timestep respectively, and m corresponds to the current gridblock. For flux term we use implicit forward Euler approximation:

$$\frac{\partial f}{\partial x} = \frac{f_m^{n+1} - f_{m-1}^{n+1}}{\Delta x} + o(\Delta x) \quad (67)$$

Source of nonlinearity

The flux function f is an explicit function of s which makes the Eq. 64 equal to:

$$\frac{\partial f}{\partial x} = \frac{f(s_m^{n+1}) - f(s_{m-1}^{n+1})}{\Delta x} + o(\Delta x) \quad (68)$$

For the purpose of this practical, we assume

$$\lambda_w = \frac{s^2}{1}, \quad \lambda_o = \frac{(1-s)^2}{10}, \quad (69)$$

which leads to

$$f(s) = \frac{s^2}{s^2 + 0.1(1-s)^2}. \quad (70)$$

Discretized transport equation

The original transport equation to the following nonlinear equation in the residual form:

$$r_m(s_m^{n+1}, s_{m-1}^{n+1}) = s_m^{n+1} - s_m^n + \frac{u_t}{\phi} \frac{\Delta t}{\Delta x} [f(s_m^{n+1}) - f(s_{m-1}^{n+1})] = 0 \quad (71)$$

with corresponding boundary and initial conditions respectively

$$s_1^n = s_{inj}, \quad s_m^0 = s_{ini}. \quad (72)$$

For the purpose of this practical, we assume that $s_1^n = 1$ and $s_m^0 = 0$.

Solution of equations

We apply the Newton-Raphson method for the solution of this equation, using the following formula:

$$\mathbf{J}(\mathbf{s}^{n+1,k})\Delta\mathbf{s} = -\mathbf{r}(\mathbf{s}^{n+1,k}). \quad (73)$$

Here \mathbf{s} is a vector of saturations, $\Delta\mathbf{s} = \mathbf{s}^{n+1,k+1} - \mathbf{s}^{n+1,k}$, k is the Newton iteration, J is Jacobian:

$$\mathbf{J} = [J_{ij}] = \left[\frac{\partial r_i}{\partial s_j} \right]. \quad (74)$$

Note: don't mix \mathbf{s}^n , $\mathbf{s}^{n+1,k}$ and $\mathbf{s}^{n+1,k+1}$.

Structure of Jacobian

Our Jacobian matrix has a sparse-linear structure with two diagonals defined by the following relations:

$$\mathbf{J}(m, m) = \frac{u_t \Delta t}{\phi \Delta x} \left(\frac{\partial f}{\partial s} \right)_m, \quad (75)$$

$$\mathbf{J}(m, m-1) = -\frac{u_t \Delta t}{\phi \Delta x} \left(\frac{\partial f}{\partial s} \right)_{m-1}. \quad (76)$$

This structure is due to the upstream weighting of the transport (fractional flow) function

Structure of Jacobian

	1	2	3	4	5	6	7	8	9	10
1	x									
2	x	x								
3		x	x							
4			x	x						
5				x	x					
6				x	x					
7					x	x				
8						x	x			
9							x	x		
10								x	x	

x - non-zero elements

Figure: Structure of Jacobian for 1D implicit transport problem

Nonlinear solution

- For every iteration, we are starting from the initial guess $s^{n+1,0} = s^n$. The solution can be found from:

$$\Delta s = -J^{-1}r. \quad (77)$$

Note: be careful with Matlab syntax for a linear solution

- The solution is updated using the following:

$$s^{n+1,k+1} = s^{n+1,k} + \Delta s, \quad (78)$$

where k is the number of Newton's iteration.

- Repeat Newton's iterations until the convergence when both

$$||r|| < \varepsilon_r, \quad ||\Delta s|| < \varepsilon_s. \quad (79)$$

Practical 7: description

Properties and simplifications:

- initial and boundary conditions: $s_1^n = 1$ and $s_m^0 = 0$;
- assume viscosity is constant and fractional flow curve is

$$f(s) = \frac{s^2}{s^2 + 0.1(1-s)^2} \quad (80)$$

- both porosity and total velocity are constants.

Main task: build implicit transport solver for two-phase 1D problem.

Advanced task: implement advanced nonlinear solvers and test them for larger timesteps; count for nonlinear iterations with different nonlinear solvers and relative permeabilities exponents

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Preferred method for forward simulation

The fully implicit method (FIM) is state-of-the-art in reservoir simulation. Most successful commercial and academic simulators use FIM or some version of it. Why?

The main advantage of FIM – it is **unconditionally stable** method:

- Timestep is not restricted and chosen from practical reasons,
- No unexpected oscillations in the solution due to violation of CFL,
- Very efficient linear and nonlinear solver options,
- Requires nonlinear solution with full Jacobian construction,
- Nonlinear solution is **not** unconditionally convergent,
- Can be expensive if timesteps are not chosen wisely.

Two-phase equations

We need to solve the following system of equations:

$$\frac{\partial M_o}{\partial t} + \frac{\partial m_o}{\partial x} = m_o^w \quad (81)$$

$$\frac{\partial M_w}{\partial t} + \frac{\partial m_w}{\partial x} = m_w^w \quad (82)$$

where

- $M_o = \phi \rho_o s_o$, $M_w = \phi \rho_w s_w$ – accumulated mass of oil and water,
- $m_o = \rho_o u_o$, $m_w = \rho_w u_w$ – flowing mass of oil and water,
- m_o^w, m_w^w – sources (sinks) of oil and water

Here mass is taken per unit volume.

Finite-volume discretization

For the accumulation term we use backward Euler approximation for block i :

$$\int_{V_i} \frac{\partial M_i}{\partial t} dV = V_i \frac{(\phi\rho s)_i^{n+1} - (\phi\rho s)_i^n}{\Delta t} + O(\Delta t), \quad (83)$$

where n and $n + 1$ correspond to the current and next timestep respectively.

For flux term we use finite volume approximation:

$$\int_{V_i} \frac{\partial \tilde{m}}{\partial x} dV = \int_{\partial V_i} \frac{\partial \tilde{m}}{\partial x} \mathbf{n} dS = \frac{m_{i+1/2}^{n+1} - m_{i-1/2}^{n+1}}{\Delta x} + O(\Delta x). \quad (84)$$

Two-point flux approximation

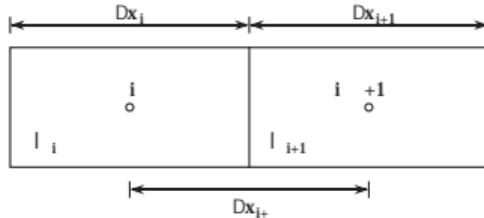


Figure: Discretization grid

Mass flux across the interface $i + 1/2$ is defined as

$$m_{i+1/2} = A_{i+1/2} \rho_{i+1/2} u_{i+1/2}. \quad (85)$$

Following Darcy's assumption, velocity for the interface $i + 1/2$ is defined

$$u_{i+1/2} = - \left(k_x \frac{k_r}{\mu} \frac{\partial p}{\partial x} \right)_{i+1/2} \simeq - \left(k_x \frac{k_r}{\mu} \right)_{i+1/2} \frac{p_{i+1} - p_i}{\Delta x_{i+}} \quad (86)$$

Discretized conservation equation

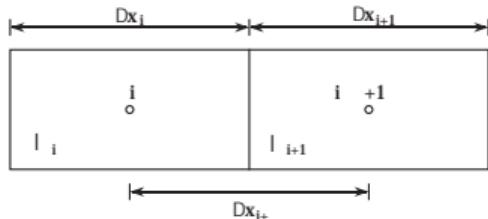


Figure: Discretization grid

Following assumptions $\Delta x, \Delta y, \Delta z, k_x = \text{const}$ we've got

$$m_{i+1/2} = -k_x \lambda_{i+1/2} \frac{A_{i+1/2}}{\Delta x} \rho_{i+1/2} (p_{i+1} - p_i), \quad (87)$$

$$m_{i+1/2} = -T_{i+1/2} (p_{i+1} - p_i), \quad (88)$$

$$T_{i+1/2} = k_x \lambda_{i+1/2} \frac{A_{i+1/2}}{\Delta x} \rho_{i+1/2} = \Gamma_{i+1/2} \lambda_{i+1/2} \rho_{i+1/2}. \quad (89)$$

Well term

The source term is defined as

$$\int_{V_i} \tilde{m}_i^w dV = m_i^w = \rho_i q_i. \quad (90)$$

where

$$m_i^w = -\frac{2\pi k_x \Delta z}{\log \frac{r_o}{r_w}} \lambda_i \rho_i (p_i - p^w) = -T_i^w (p_i - p^w) \quad (91)$$

Here we assume that the well is vertical. Well transmissibility (or well index) is defined as

$$T_i^w = \frac{2\pi k_x \Delta z}{\log \frac{r_o}{r_w}} \lambda_i \rho_i = \Gamma^w \lambda_i \rho_i \quad (92)$$

Closing relations

Saturation constraint:

$$s_w + s_o = 1 \quad \Rightarrow \quad s = s_w, \quad s_o = 1 - s. \quad (93)$$

Capillary pressure assumptions:

$$p_o - p_w = P_{cow}(s) \quad \Rightarrow \quad p = p_o, \quad p_w = p - P_{cow}(s) \quad (94)$$

Note: possible to use $\{p_o, p_w\}$ or even $\{M_j\}$ as nonlinear unknowns.

In the following derivations, we ignore capillary pressure; the rest of the properties are direct functions of nonlinear unknowns:

$$k_{rj}(s), \quad \mu_j(p), \quad \rho_j(p), \quad \phi(p). \quad (95)$$

Note: transmissibilities become $T_j(p, s)$ and $T_j^w(p, s)$ in Eq. 89 and 92.

Residual equations

Residual form of discretized Eq. 81 and Eq. 82 can be defined as

$$\begin{aligned} r_{o,i} &= \frac{V_i}{\Delta t} [(\phi\rho(1-s))_i^{n+1} - (\phi\rho(1-s))_i^n] \\ &\quad - T_{o,i+1/2}(p_{i+1} - p_i) + T_{o,i-1/2}(p_i - p_{i-1}) + T_o^w(p_i - p^w) \end{aligned} \tag{96}$$

$$\begin{aligned} r_{w,i} &= \frac{V_i}{\Delta t} [(\phi\rho s)_i^{n+1} - (\phi\rho s)_i^n] \\ &\quad - T_{w,i+1/2}(p_{i+1} - p_i) + T_{w,i-1/2}(p_i - p_{i-1}) + T_w^w(p_i - p^w) \end{aligned} \tag{97}$$

Transmissibility in Eq. 96 and Eq. 97 are defined as

$$T_{j,i+1/2} = \left(\frac{k_x A}{\Delta x} \right)_{i+1/2} \left(\rho_j \frac{k_{rj}}{\mu_j} \right)_{i+1/2} = \Gamma_{i+1/2} \beta_{j,i+1/2} \tag{98}$$

Transmissibility computation in CPG grid

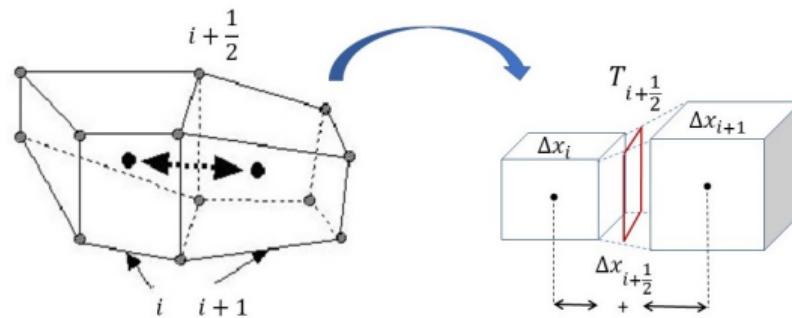


Figure: Corner-Point Geometry (CPG) grid

The geometric part of transmissibility is defined at the interface $i + 1/2$ as

$$\Gamma_{i+1/2} = \frac{\gamma_i \gamma_{i+1}}{\gamma_i + \gamma_{i+1}}, \quad \gamma_i = A_i \frac{K_i}{\Delta x_i}. \quad (99)$$

Units in simulation code

In reservoir simulation, we often used specific units called Metric (similar to SI with few exceptions); here, we are going to utilize Metric units which uses

- Length in meters;
- Time in days;
- Permeability in mDarcy;
- Pressure in bars;
- Mass in moles;
- Viscosity in cPoise.

That only requires correction to the transmissibility

$$\Gamma_{i+1/2} = c \left(\frac{k_x A}{\Delta x} \right)_{i+1/2}, \quad \Gamma_i^w = c \frac{2\pi k_x \Delta z}{\log \frac{r_o}{r_w}}, \quad (100)$$

where c is the Darcy constant $c = 0.008526$ and $r_0 = 0.2\Delta x$.

Upstream weighting

The full transmissibility in the discretized equation can be shown as

$$T_{j,i+1/2} = \left(\frac{k_x A}{\Delta x} \right)_{i+1/2} (\rho_j \lambda_j)_{i+1/2} = \Gamma_{i+1/2} \beta_{j,i+1/2} \quad (101)$$

The phase-based part of transmissibility is defined as

$$\beta_{j,i+1/2} = \begin{cases} \beta_{j,i}, & u_{j,i+1/2} > 0, \\ \beta_{j,i+1}, & u_{j,i+1/2} < 0. \end{cases} \quad (102)$$

Note: velocity direction is equivalent (in our assumptions) to

$$u_{j,i+1/2} > 0 \iff p_{i+1} < p_i. \quad (103)$$

Impact of gravity and capillarity

In the general case, the velocity is defined through the potential of phase j

$$u_{j,i+1/2} = T_{j,i+1/2} \Phi_{j,i+1/2}. \quad (104)$$

Even in 1D, if depths are not constant, then

$$\Phi_{j,i+1/2} = p_{j,i+1} - p_{j,i} - \gamma_{j,i+1/2}(z_{i+1} - z_i), \quad (105)$$

where $\gamma_{j,i+1/2} = g\rho_{j,i+1/2}$. Then upstream weighting becomes

$$\beta_{j,i+1/2} = \begin{cases} \beta_{j,i}, & \Phi_{j,i+1/2} > 0, \\ \beta_{j,i+1}, & \Phi_{j,i+1/2} < 0. \end{cases} \quad (106)$$

Solution of fully-implicit equations

System of conservation equations:

$$\mathbf{r}(\mathbf{p}^{n+1,k+1}, \mathbf{s}^{n+1,k+1}) = \mathbf{0}. \quad (107)$$

We apply the Newton-Raphson method for the solution of this equation:

$$\mathbf{J}(\mathbf{p}^{n+1,k}, \mathbf{s}^{n+1,k}) \begin{bmatrix} \Delta \mathbf{p} \\ \Delta \mathbf{s} \end{bmatrix} = -\mathbf{r}(\mathbf{p}^{n+1,k}, \mathbf{s}^{n+1,k}). \quad (108)$$

Here $\Delta \mathbf{p} = \mathbf{p}^{n+1,k+1} - \mathbf{p}^{n+1,k}$ and $\Delta \mathbf{s} = \mathbf{s}^{n+1,k+1} - \mathbf{s}^{n+1,k}$, k is the Newton's iteration, \mathbf{J} is Jacobian:

$$\mathbf{J} = [J_{ij}] = \begin{bmatrix} \frac{\partial r_{o,i}}{\partial p_j} & \frac{\partial r_{o,i}}{\partial s_j} \\ \frac{\partial r_{w,i}}{\partial p_j} & \frac{\partial r_{w,i}}{\partial s_j} \end{bmatrix}. \quad (109)$$

Calculation of derivatives

Different approaches to the calculation of Jacobian:

- *Analytical* – derive and implement derivatives manually
 - + *the best option in terms of performance*
 - *difficult and not easy for complex physics*
- *Numerical* – use numerical perturbation of residual
 - + *simple to implement and flexible*
 - *can be slow and not always robust*
- *Automatic Differentiation* – capability in advanced libraries
 - + *simple to implement and flexible*
 - *introduce (sometimes severe) overhead*

Nonlinear solution

Need to solve the nonlinear equation :

$$\mathbf{r}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} = \{\mathbf{p}^{n+1}, \mathbf{s}^{n+1}\} \quad (110)$$

Newton-Raphson iterative method:

$$\begin{aligned} \mathbf{x}^0 &= \{\mathbf{p}^n, \mathbf{s}^n\} & : & \quad \mathbf{J}(\mathbf{x}^0)\Delta\mathbf{x} = -\mathbf{r}(\mathbf{x}^0) \Rightarrow \Delta\mathbf{x}^1 \\ \mathbf{x}^1 &= \mathbf{x}^0 + \Delta\mathbf{x}^1 & : & \quad \mathbf{J}(\mathbf{x}^1)\Delta\mathbf{x} = -\mathbf{r}(\mathbf{x}^1) \Rightarrow \Delta\mathbf{x}^2 \\ &&&\dots\\ &||\mathbf{r}(\mathbf{x}^k)|| < \varepsilon_r, \quad ||\Delta\mathbf{x}^k|| < \varepsilon_x \end{aligned}$$

When converged:

$$\{\mathbf{p}^{n+1}, \mathbf{s}^{n+1}\} = \mathbf{x}^k \Rightarrow t = t + \Delta t, \quad \{\mathbf{p}^n, \mathbf{s}^n\} = \{\mathbf{p}^{n+1}, \mathbf{s}^{n+1}\}$$

Problem definition

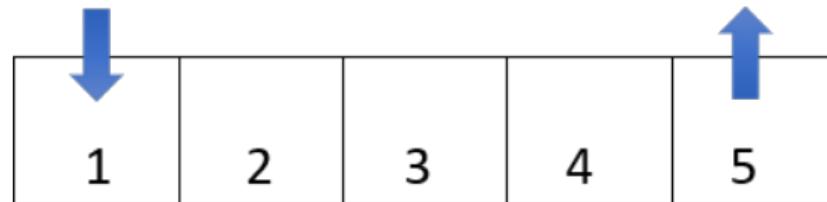


Figure: Simple 1D model

Next, we are going to solve the following problem:

- Reservoir represented by homogeneous 1D domain;
- Injection well in first and production well in last blocks;
- Both wells are controlled by Bottom Hole Pressure (BHP);
- Two-phase flow with water injected into oil + water.

Numeration of equations: sparse-block Jacobian

	1	2	3	4	5	6	7	8	9	10
1	x x					x				
2	x x x					x x				
3		x x x					x x			
4			x x x					x x		
5			x x						x x	
6	x x					x				
7	x x x					x x				
8		x x x					x x			
9			x x x					x x		
10			x x						x x	

x is non-zero elements

Figure: Structure of Jacobian for 1D fully implicit two-phase problem with numeration

Numeration of equations: block-sparse Jacobian

	1	2	3	4	5	6	7	8	9	10
1	x	x	x							
2	x	x	x							
3	x	x	x	x	x					
4	x	x	x	x	x					
5			x	x	x	x	x			
6			x	x	x	x	x			
7				x	x	x	x	x		
8				x	x	x	x	x		
9					x	x	x	x		
10					x	x	x	x		

x is non-zero elements

Figure: Structure of Jacobian for 1D fully implicit two-phase problem with numeration

Residual equations

Homogeneous 1D reservoir, equation for water

$$\begin{aligned} r_{w,i} &= \frac{V\phi}{\Delta t} [(\rho_w s_w)_i^{n+1} - (\rho_w s_w)_i^n] \\ &\quad - T_{w,i+1/2}(p_{i+1} - p_i) + T_{w,i-1/2}(p_i - p_{i-1}) + T_w^w(p_i - p^w) \end{aligned} \tag{111}$$

Within assumptions, transmissibilities in Eq. 111 are defined as

$$T_{j,i+1/2} = \left(\frac{k_x A}{\Delta x} \right)_{i+1/2} \left(\rho_j \frac{k_{rj}}{\mu_j} \right)_{i+1/2} = \Gamma \beta_{j,i} \tag{112}$$

$$T_{j,i-1/2} = \left(\frac{k_x A}{\Delta x} \right)_{i-1/2} \left(\rho_j \frac{k_{rj}}{\mu_j} \right)_{i-1/2} = \Gamma \beta_{j,i-1} \tag{113}$$

In addition, introduce the accumulation term:

$$\alpha_{j,i} = (\rho_j s_j)_i \tag{114}$$

Operators of conservation equations

State-dependent operators in governing equations are defined as

$$\alpha_{j,i} = (\rho_j s_j)_i, \quad \beta_{j,i} = \left(\rho_j \frac{k_{rj}}{\mu_j} \right)_i. \quad (115)$$

For simplicity, let's assume that $\mu_j = \text{const}$ and

$$\rho_j = \rho_j^0 [1 + c_j(p - p^0)], \quad k_{rj} = s_j^{n_j}. \quad (116)$$

Here ρ_j^0 , c_j and p^0 are density at surface conditions, fluid compressibility and pressure at surface conditions respectively. Then

$$\alpha_{1,i} = (\rho_w(p)s)_i, \quad \beta_{1,i} = (\rho_w(p)s^{n_w}/\mu_w)_i, \quad (117)$$

$$\alpha_{2,i} = (\rho_o(p)(1-s))_i, \quad \beta_{2,i} = (\rho_o(p)(1-s)^{n_o}/\mu_o)_i.$$

Nonlinear equation in operator form

Here we rewrite Eqs. 96 and 97 in the following form:

$$r_{j,i} = \Theta_i(\alpha_{j,i}^{n+1} - \alpha_{j,i}^n) - Q_{j,i+} + Q_{j,i-} + Q_{j,w}. \quad (118)$$

Here $\Theta_i = \frac{V_i \phi_i^0}{\Delta t}$ and fluxes defined as

$$Q_{j,i+} = \Gamma_{i+} \hat{\beta}_{j,i+}(p_{i+1} - p_i), \quad Q_{j,i-} = \Gamma_{i-} \hat{\beta}_{j,i-}(p_i - p_{i-1}), \quad (119)$$

$$Q_{j,w} = \Gamma_w \beta_{j,w}(p_i - p_w), \quad \beta_{j,w} = \begin{cases} \beta_{j,inj}, & \text{for injector} \\ \beta_{j,i}, & \text{for producer} \end{cases} \quad (120)$$

All operators are defined as

$$\alpha_{j,i} = [\rho_j s_j (1 + c_r(p - p^0))]_i, \quad (121)$$

$$\hat{\beta}_{j,i+} = \begin{cases} \beta_{j,i+1}, & p_{i+1} > p_i \\ \beta_{j,i}, & p_{i+1} < p_i \end{cases}, \quad \beta_{j,i} = \left(\rho_j \frac{k_{rj}}{\mu_j} \right)_i. \quad (122)$$

Few practical suggestions

- Introduce all parameters at the beginning
- Use lambda functions for properties:

```
rho_w = lambda p: rho_w0 * (1 + cw * (p - p0))
```

- Use a dictionary to define an array of functions (with j index):

```
a = dict([(0, lambda p, s: rho_w(p) * s),  
           (1, lambda p, s: rho_o(p) * (1-s))])
```

- Create function for RHS assemble and use a loop over phases (j):

```
rhs[2*i+j] = Th * (a[j](p[i], s[i]) ...
```

- Use PyTorch for automatic differentiation of residual with respect to nonlinear unknowns.

Numerical derivatives of residual

As an alternative, you can use numerical derivatives to find Jacobian. When residual is implemented as a direct function of your nonlinear unknowns, it is very easy to make:

$$\frac{\partial r_i(p, s)}{\partial p_j} = \frac{r_i(p + \delta_{ij}\varepsilon, s) - r_i(p, s)}{\varepsilon}, \quad (123)$$

$$\frac{\partial r_i(p, s)}{\partial s_j} = \frac{r_i(p, s + \delta_{ij}\varepsilon) - r_i(p, s)}{\varepsilon}. \quad (124)$$

Note: make sure that the perturbation parameter ε stays small, between $10^{-4} - 10^{-8}$; make sure that it is adjusted to the characteristic value of your unknowns, e.g. $p \approx 10^5 - 10^7$ Pa or $10^1 - 10^3$ bars, while $s \in [0, 1]$.

Practical 8: two-phase fully-implicit simulator (numerical)

- rock is incompressible, oil and water are slightly compressible:

$$\rho_o = \rho_j^0 (1 + c_j(p - p^0)), \quad \rho_w^0 = 10, \quad \rho_o^0 = 20, \quad c_w = c_o = 10^{-5};$$

- assume viscosities are constant $\mu_w = 1$, $\mu_o = 2$ and

$$k_{rj} = s_j^{n_j}, \quad n_w = 2, \quad n_o = 2;$$

- you can define explicitly $\Theta = 1$;
- take initial and boundary conditions from transport problem;
- pressures: injection 200 bars, production 100 bars, initial linearly changing between injection and production;
- saturation: initial $S_{ini} = 0.1$ and injection $S_{inj} = 0.9$.

Main task: build a fully implicit simulator using numerical derivatives.

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

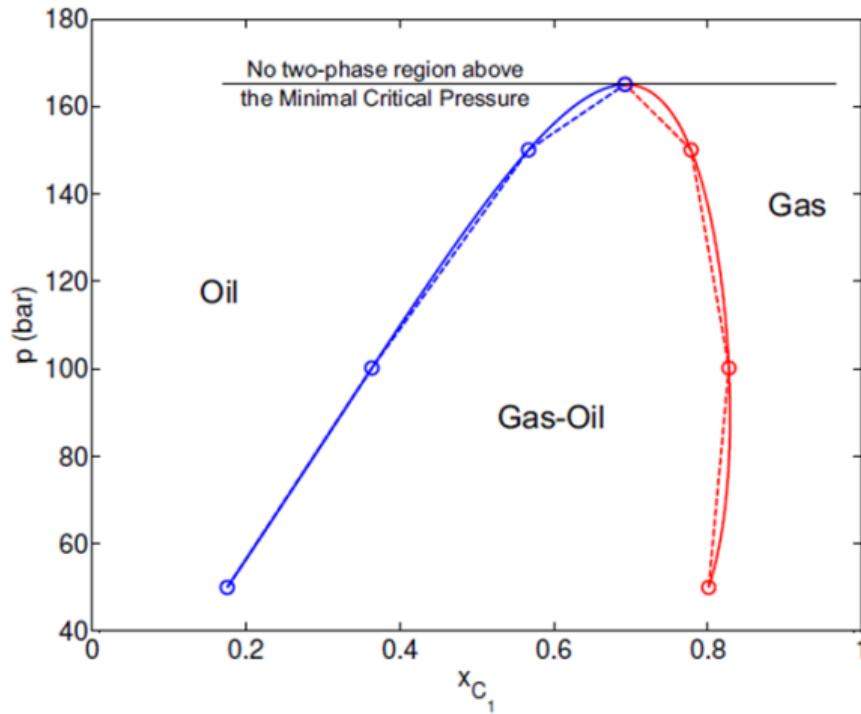
Compositional models

- Compositional models include oil and gas description as a mix of different hydrocarbon and other chemical components
- That increase number of equation from 3 (black-oil) to n_c (usually 6-12) - number of separate components or pseudo components
- The complexity of phase behavior calculations is also increased
- As the result, time needed for simulation can increase by several (5-100) folds

When to use?

- Usually, compositional models and their extensions are used for Enhanced Oil Recovery (EOR) operations
- Applications include (but not limited to)
 - Hydrocarbon or other gas injection processes
 - Gas-condensate systems
 - Complex thermal processes
 - Chemical flooding

Binary diagram for C₁ - C₁₀



Few important definitions

- The mole is the unit of measurement in the SI units for amount of substance – N_i or n_i
- Overall molar fraction: $z_i = n_i/n_{tot}$, $n_{tot} = \sum n_j$
- Phase split: $n_i = n_i^L + n_i^V$

$$n_{tot} = n^L + n^V \rightarrow 1 = \frac{n^L}{n_{tot}} + \frac{n^V}{n_{tot}} = L + V$$

$$z_i = \frac{n_i^L}{n_{tot}} + \frac{n_i^V}{n_{tot}} = \frac{n_i^L}{n^L} \frac{n^L}{n_{tot}} + \frac{n_i^V}{n^V} \frac{n^V}{n_{tot}} = x_i L + y_i V$$

$$\sum x_i = \sum \frac{n_i^L}{n^L} = 1, \sum y_i = \sum \frac{n_i^V}{n^V} = 1 \rightarrow \sum x_i - \sum y_i = 0$$

Phase description: general VLE

$$z_i = V y_i + (1 - V) x_i$$

$$K_i = y_i / x_i$$

Partitioning coefficients

Phase description: general VLE

$$z_i = V y_i + (1 - V) x_i$$

$$K_i = y_i / x_i$$

Partitioning coefficients

$$z_i = V K_i x_i + (1 - V) x_i$$

$$y_i = K_i x_i$$

Phase description: general VLE

$$z_i = V y_i + (1 - V) x_i$$

$$K_i = y_i / x_i$$

Partitioning coefficients

$$z_i = V K_i x_i + (1 - V) x_i$$

$$y_i = K_i x_i$$

$$z_i = x_i (V K_i + (1 - V)) = x_i (V(K_i - 1) + 1)$$

$$x_i = \frac{z_i}{V(K_i - 1) + 1} \quad y_i = \frac{K_i z_i}{V(K_i - 1) + 1}$$

Rachford-Rice equation

$$x_i = \frac{z_i}{V(K_i - 1) + 1} \quad y_i = \frac{K_i z_i}{V(K_i - 1) + 1} \quad \rightarrow \quad \sum (x_i - y_i) = 0$$

Rachford-Rice equation

$$x_i = \frac{z_i}{V(K_i - 1) + 1} \quad y_i = \frac{K_i z_i}{V(K_i - 1) + 1} \quad \rightarrow \quad \sum (x_i - y_i) = 0$$

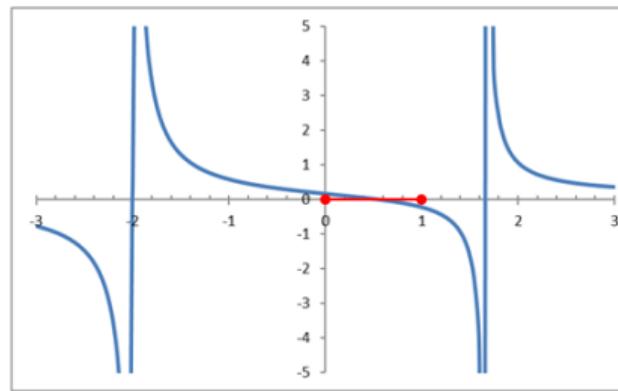
$$g(V) = \frac{z_1(K_1 - 1)}{V(K_1 - 1) + 1} + \frac{z_2(K_2 - 1)}{V(K_2 - 1) + 1} = 0$$

Rachford-Rice equation

$$x_i = \frac{z_i}{V(K_i - 1) + 1} \quad y_i = \frac{K_i z_i}{V(K_i - 1) + 1} \quad \rightarrow \quad \sum (x_i - y_i) = 0$$

$$g(V) = \frac{z_1(K_1 - 1)}{V(K_1 - 1) + 1} + \frac{z_2(K_2 - 1)}{V(K_2 - 1) + 1} = 0$$

$$K_1 = 1.5, K_2 = 0.4 \\ z_1 = 0.7, z_2 = 0.3$$



Nonlinear solution - bisection

We need to solve equation: $f(x)=0, x \in [a,b]$.

Assume $f(a) < 0, f(b) > 0$ and build iteration process as:

$$a_0 = a, b_0 = b; f(a_0) < 0, f(b_0) > 0 \Rightarrow x_0 = \frac{1}{2}(a_0 + b_0)$$

$$f(x_0) < 0 \Rightarrow a_1 = x_0, b_1 = b_0;$$

$$f(x_0) > 0 \Rightarrow a_1 = a_0, b_1 = x_0.$$

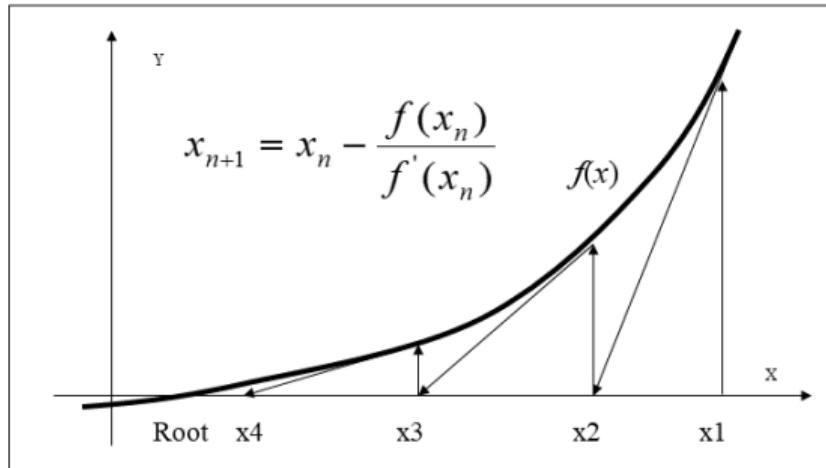
Sequence $x_k = \frac{1}{2}(a_k + b_k)$ converge to the root x^* ;

The error of approximation is always $< \varepsilon = 2^{-(k+1)}(b_0 - a_0)$.

Newton's method

The most efficient method is based on derivative:

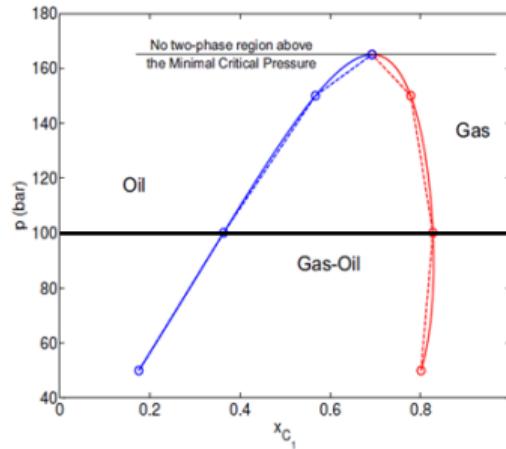
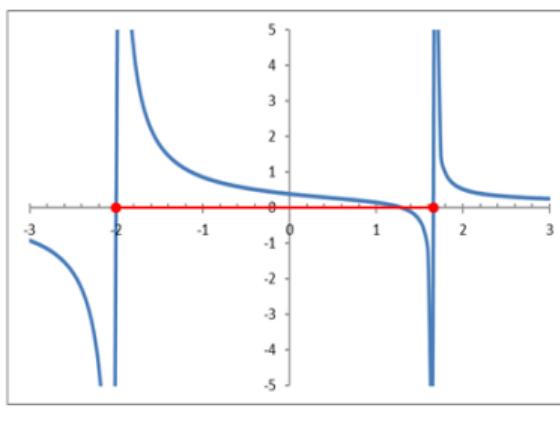
$$f(x) = 0 \Rightarrow x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$



Negative flash

$$K_1 = 1.5, K_2 = 0.4 \\ z_1 = 0.9, z_2 = 0.1$$

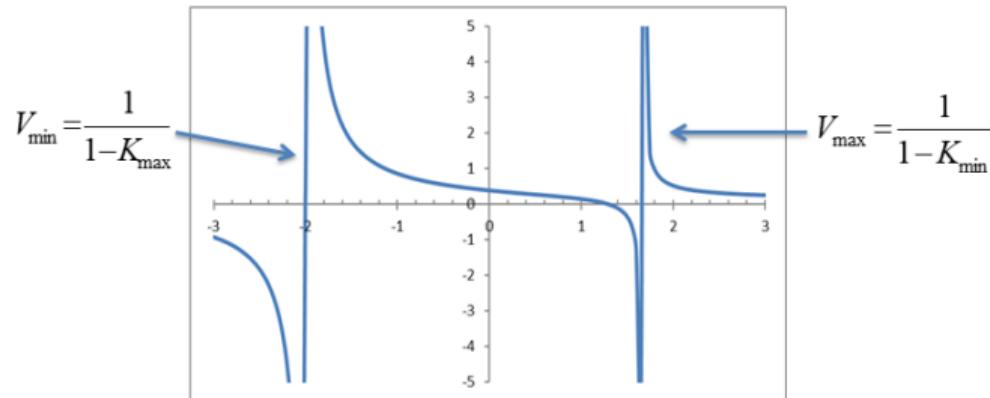
$$\frac{z_1(K_1 - 1)}{V(K_1 - 1) + 1} + \frac{z_2(K_2 - 1)}{V(K_2 - 1) + 1} = 0$$



$$V(K_1 - 1) + 1 = 0 \rightarrow V = \frac{1}{1 - K_1}$$

Practical 9: RR for binary system

$$K_1 = 1.5, K_2 = 0.4 \quad \frac{z_1(K_1 - 1)}{V(K_1 - 1) + 1} + \frac{z_2(K_2 - 1)}{V(K_2 - 1) + 1} = 0$$
$$z_1 = 0.9, z_2 = 0.1$$



$$V_{\min} = \frac{1}{1-K_{\max}}$$

$$V_{\max} = \frac{1}{1-K_{\min}}$$

$$f(x_0) < 0 \Rightarrow a_I = x_0, b_I = b_0;$$

$$f(x_0) > 0 \Rightarrow a_I = a_0, b_I = x_0.$$

- 1 Introduction to reservoir simulation
- 2 Basic numerical methods
- 3 Single-phase incompressible flow
- 4 Slightly compressible flow
- 5 Wells in simulation
- 6 Connection-list for simulation
- 7 Basic two-phase transport
- 8 Two-phase flow and transport
- 9 Multiphase flash
- 10 Compositional simulation

Problem statement

$$\frac{\partial}{\partial t} \left(\phi \sum_j x_{cj} \rho_j s_j \right) + \operatorname{div} \left(\sum_j x_{cj} \rho_j \mathbf{v}_j \right) = \sum_j x_{cj} \rho_j \tilde{q}_j, \quad c = 1, \dots, C. \quad (125)$$

Here we introduced

- ϕ - rock porosity,
- ρ_j - phase density
- s_j - phase saturation
- x_{cj} - molar fraction of component, ‘
- \mathbf{v}_j - phase velocity,
- q_j - volumetric phase rate.

Molar formulation

Introduce overall composition z_c as

$$z_c = \frac{M_c}{M_{tot}} = \frac{n_c}{n_T} = \frac{n_c}{\rho_T} = \frac{\sum_j x_{cj} \rho_j s_j}{\sum_j \rho_j s_j}. \quad (126)$$

Assuming two phases, phase fractions are defined as

$$V = \frac{\rho_g s_g}{\sum_j \rho_j s_j}, \quad L = \frac{\rho_o s_o}{\sum_j \rho_j s_j}, \quad (127)$$

which gives the following relation for overall composition

$$z_c = Lx_{co} + (1 - L)x_{cg} = Lx_c + (1 - L)y_c. \quad (128)$$

since by definition, $L + V = 1$.

Conservation equation in molar form

Conservation equation for each component has the following form:

$$\frac{\partial}{\partial t}(\phi\rho_T z_c) + \operatorname{div}\left(\sum_j x_{cj}\rho_j \mathbf{v}_j\right) = \sum_j x_{cj}\rho_j \tilde{q}_j, \quad c = 1, \dots, C. \quad (129)$$

For simplicity, the following lecture will be explained for 1D reservoir:

$$\frac{\partial M_c}{\partial t} + \frac{\partial m_c}{\partial x} = \tilde{m}_c^q, \quad c = 1, \dots, C, \quad (130)$$

where

- $M_c = \phi\rho_T z_c$,
- $m_c = x_c\rho_o u_o + y_c\rho_g u_g$,
- $\tilde{m}_c^q = x_c\rho_o q_o + y_c\rho_g q_g$.

Residual equations and total velocity

Residual form of discretized Eq. 129 is defined as

$$\begin{aligned} r_{c,i} &= \frac{V}{\Delta t} \left[(\phi \rho_T z_c)_i^{n+1} - (\phi \rho_T z_c)_i^n \right] + T_c^w (p_i - p^w) \\ &\quad - T_{c,i+1/2} (p_{i+1} - p_i) + T_{c,i-1/2} (p_i - p_{i-1}), \quad c = 1, \dots, C \end{aligned} \tag{131}$$

For incompressible fluid and with equal densities ($\rho_g = \rho_o = \rho$):

$$\rho_t = \sum_j \rho_j s_j = \rho \sum_j s_j = \rho. \tag{132}$$

Total velocity is defined as

$$U_t = - \left[\frac{k_x A}{\Delta x} (\lambda_g + \lambda_o) \right]_{i+1/2} (p_{i+1} - p_i) = const. \tag{133}$$

Similarly U_t is defined at $i - 1/2$ interface.

Simple 1D compositional transport equations

Now the flux at $i + 1/2$ interface can be translated to

$$Q_{c,i+1/2} = T_{c,i+1/2}(p_{i+1} - p_i) = -U_t \rho \left[\frac{x_c \lambda_o + y_c \lambda_g}{\lambda_o + \lambda_g} \right]_{i+1/2}, \quad (134)$$

with similar relation at $i - 1/2$ interface. Finally Eq. 131 becomes

$$r_{c,i} = (z_c)_i^{n+1} - (z_c)_i^n + \Theta [(F_c)_{i+1/2} - (F_c)_{i-1/2}] = 0. \quad (135)$$

Here

$$\Theta = \frac{U_t}{\phi} \frac{\Delta t}{\Delta x}, \quad F_c = \frac{x_c \lambda_o + y_c \lambda_g}{\lambda_o + \lambda_g} = x_c f + y_c (1 - f), \quad (136)$$

where f is a fractional flow function of liquid phase. Also notice that in equal density assumptions, $S_o = L$.

Fractional flow curves

$$f = \frac{\lambda_o}{\lambda_o + \lambda_g} = \frac{k_{ro}}{k_{ro} + Mk_{rg}}, \quad M = \frac{\mu_o}{\mu_g}; \quad F_c = x_c f + y_c (1 - f) \quad (137)$$

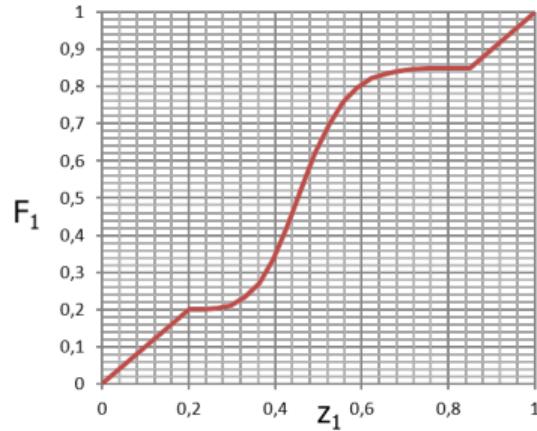
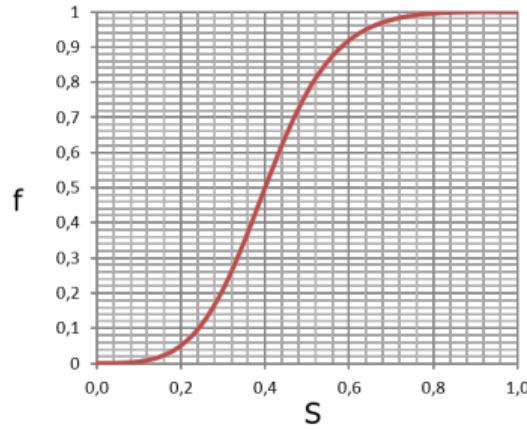


Figure: Fractional flow: (a) for two-phase, and (b) for particular composition

Stability of explicit approximation

The derived scheme is conditionally stable which means that Δt is constrained by the size of gridblock and mobility of the fluid. The criterion, similar to 1D immiscible two-phase transport, is applied:

$$\Delta t < \frac{\Delta x}{\Lambda}. \quad (138)$$

In compositional transport, there are several conservation equations depending on the number of components; the criteria becomes

$$\Delta t < \min_c \frac{\Delta x}{\Lambda_c}, \quad (139)$$

where

$$\Lambda_c = x_c \lambda_o + y_c \lambda_g. \quad (140)$$

That means that the timestep in compositional simulation is constrained by the fastest moving component (usually the lightest gas).

Practical 10: Explicit 1D compositional transport

Main task: construct an explicit 1D reservoir simulator for incompressible compositional transport.

Few practical suggestions:

- Use two-phase flash with constant K-values for evaluation of phase behavior (calculation of L and then x_c and y_c if two-phase).
- In this exercise you should use explicit time approximation.
- Use fixed K-values, e.g. $K_1 = 3$ and $K_2 = 0.1$ for binary system
- As initial conditions use 1% of gas component and 99% of liquid.
- For boundary (injection) conditions use composition with 99% of gas component and 1% of liquid.
- For relative permeability take exponents = 2 and the viscosity of gas 10 times higher than liquid.

Fully-implicit compositional formulation

The conservation equation for each component has the following form:

$$\begin{aligned} r_{c,i} &= \frac{V}{\Delta t} \left[(\phi \rho_T z_c)_i^{n+1} - (\phi \rho_T z_c)_i^n \right] + T_c^w (p_i - p^w) \\ &\quad - T_{c,i+1/2} (p_{i+1} - p_i) + T_{c,i-1/2} (p_i - p_{i-1}), \quad c = 1, \dots, C \end{aligned} \tag{141}$$

In molar formulation for isothermal compositional problems, the unknowns are p and z_c , $c = 1, \dots, C - 1$.

To evaluate properties, for a given $\{p, T, z_c\}$ we need to get $\{x_c, y_c, L\}$ using flash. The fully-implicit solution requires getting partial derivatives of flash results:

$$\frac{\partial x_{cj}}{\partial p}, \quad \frac{\partial x_{cj}}{\partial z_c}, \quad \frac{\partial L}{\partial p}, \quad \frac{\partial L}{\partial z_c}. \tag{142}$$

Operator form of discretized equations

Let's re-write transmissibility in operator form:

$$T_{c,i+1/2} = \left(\frac{k_x A}{\Delta x} \right)_{i+1/2} \left(\sum_j x_{cj} \rho_j \lambda_j \right)_{i+1/2} = \Gamma_{i+1/2} \beta_{c,i+1/2} \quad (143)$$

The component part of transmissibility can be as

$$\beta_{c,i+1/2} = \sum_j (x_{cj} \rho_j \lambda_j)_{i+1/2} = \sum_j \beta_{cj,i+1/2}, \quad (144)$$

$$\beta_{cj,i+1/2} = \begin{cases} \beta_{cj,i}, & p_i > p_{i+1}, \\ \beta_{cj,i+1}, & p_i < p_{i+1}. \end{cases} \quad (145)$$

Note: due to gravity or capillarity, it is possible that the same component is moving in different directions within different phases.

Well term in operator form

By analogy, the well term in operator form is

$$T_c^w = \frac{2\pi k_x \Delta z}{\log \frac{r_o}{r_w}} \sum_j x_{cj} \lambda_j \rho_j = \Gamma^w \beta_c. \quad (146)$$

The control equation is usually defined with inflow from different perforations at separator conditions

$$q_j^s = q(p, T, z, p^s, T^s, z^s), \quad (147)$$

which requires flash at p^s , T^s and

$$z_c^s = \frac{m_c^q}{\sum_i m_i^q} = \frac{m_c^q}{m_T^q}. \quad (148)$$

Solution of implicit compositional problem

Now back to the system of conservation equations:

$$\mathbf{r}(p^{n+1,k+1}, \mathbf{z}^{n+1,k+1}) = \mathbf{0}. \quad (149)$$

We apply the Newton-Raphson method for the solution of this equation:

$$\mathbf{J}(p^{n+1,k}, \mathbf{z}^{n+1,k})\Delta\mathbf{x} = -\mathbf{r}(p^{n+1,k}, \mathbf{z}^{n+1,k}). \quad (150)$$

Here $\Delta\mathbf{x} = \{p^{n+1,k+1} - p^{n+1,k}, \mathbf{z}^{n+1,k+1} - \mathbf{z}^{n+1,k}\}$, k is the Newton's iteration and \mathbf{J} is Jacobian entry:

$$\mathbf{J} = [\mathbf{J}_{ij}] = \left[\frac{\partial \mathbf{r}_i}{\partial \mathbf{x}_j} \right]. \quad (151)$$

Here each entry \mathbf{J}_{ij} in Jacobian is a matrix of $[C \times C]$ size.

Compositional problem: flash and derivatives

To get $\{x_c, y_c, L\}$ for a given \mathbf{z} , we solve $\mathbf{r} = 0$ where:

$$f_{1c} = Lx_c + (1 - L)y_c - z_c, \quad c = 1, \dots, C, \quad (152)$$

$$f_{2c} = K(p)x_c - y_c, \quad c = 1, \dots, C, \quad (153)$$

$$f_3 = \sum (x_c - y_c). \quad (154)$$

The fully-implicit solution requires getting partial derivatives of flash results:

$$\frac{\partial x_c}{\partial p}, \frac{\partial y_c}{\partial p}, \frac{\partial L}{\partial p}, \frac{\partial x_c}{\partial z_c}, \frac{\partial y_c}{\partial z_c}, \frac{\partial L}{\partial z_c}?$$

For $\mathbf{x} = \{x, y, L\}$ and $\mathbf{z} = \{p, z\}$ we can apply the inverse theorem for $\mathbf{f}(\mathbf{x}, \mathbf{z}) = 0$:

$$\frac{\partial \mathbf{x}}{\partial \mathbf{z}} = \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1}. \quad (155)$$

Derivatives for binary system

Luckily, for binary system at fixed K-values ($K(p) = \text{const}$), x and y are independent from p and z , and

$$L = \frac{z_c - y_c}{x_c - y_c}, \quad c = 1, \dots, C. \quad (156)$$

The fully-implicit solution requires getting partial derivatives of flash results:

$$\frac{\partial x_c}{\partial z_c} = 0, \quad \frac{\partial y_c}{\partial z_c} = 0, \quad \frac{\partial L}{\partial z_c} = \frac{1}{x_c - y_c}. \quad (157)$$

Generic form of compositional problem

For conventional compositional problems, operators are defined as

$$\alpha_{c,i} = ([1 + c_r(p - p^0)]\rho_t z_c)_i, \quad \beta_{c,i} = \left(\rho_o x_c \frac{k_{ro}}{\mu_o} + \rho_g y_c \frac{k_{rg}}{\mu_g} \right)_i. \quad (158)$$

Here we assume that $\phi = \phi^0(1 + c_r(p - p^0))$, where ϕ^0 is porosity at p^0 and c_r is rock compressibility.

Notice that in this case we use the general compositional formulation

$$\begin{aligned} r_{c,i} &= \frac{V\phi^0}{\Delta t} (\alpha_i^{n+1} - \alpha_i^n) + \Gamma^w \beta_c^w (p_i - p^w) \\ &\quad - (\Gamma \beta_c)_{i+1/2} (p_{i+1} - p_i) + (\Gamma \beta_c)_{i-1/2} (p_i - p_{i-1}), \quad c = 1, \dots, C. \end{aligned} \quad (159)$$

Calculation of derivatives

Different approaches to calculation of Jacobian:

- *Analytical* – derive and implement derivatives manually
 - + *the best option in terms of performance*
 - *difficult and not easy for complex physics*
- *Numerical* – use numerical perturbation of residual
 - + *simple to implement and flexible*
 - *can be slow and not always robust*
- *Automatic Differentiation* – capability in advanced libraries
 - + *simple to implement and flexible*
 - *introduce (sometimes severe) overhead*

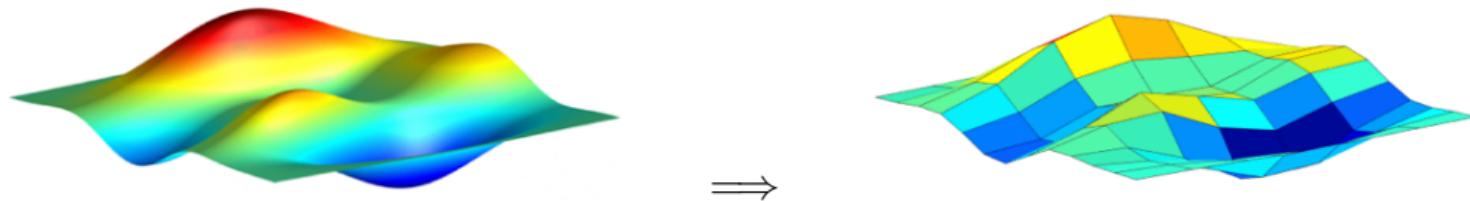
Operator-Based Linearization (OBL)

Due to the cost of flash procedure, derivatives calculations are very complex and expensive in Eqs. 158-159.

Alternative approach is an Operator-Based Linearization:

- ① Introduce uniform grid in parameter space $\{p, z\}$;
- ② compute nonlinear operators α_c and β_c in vertices of the mesh;
- ③ apply multi-linear interpolation to evaluate an approximate (continuous) version of operators and use them in solution;
- ④ use interpolation coefficients as derivatives for Jacobian assembly.

Advantages of the OBL approach



Pros:

- direct evaluation of derivatives (interpolation coefficients);
- simplified assemble of Jacobian (chain rule combination);
- skip redundant flux calculations.

Cons:

- introduce error in physical terms of conservation equation;
- interpolation in high-dimensional parameter space is expensive.