

Numerical Simulation Methods in Geophysics, Part 2: Finite Differences

1. MGPY+MGIN, 3. MDRS+MGEX-CMG

thomas.guenther@geophysik.tu-freiberg.de

Finite differences

Approximate derivative operators by differences

$$\frac{\partial u}{\partial x} \approx \frac{\Delta u}{\Delta x}$$

and solution u by finite values u_i at points x_i , e.g.

$$du/dx_{2.5} := (u_3 - u_2)/(x_3 - x_2)$$

$$\frac{\partial^2 u_3}{\partial x^2} \approx \frac{du/dx_{3.5} - du/dx_{2.5}}{(x_4 - x_2)/2} = \frac{(u_4 - u_3)/(x_4 - x_3) - (u_3 - u_2)/(x_3 - x_2)}{(x_4 - x_2)/2}$$

Summarize in a matrix

Assumption: equidistant discretization $\Delta x=1$, conductivity $a=1$

1st derivative: $[-1, +1]/dx$, 2nd derivative $[+1, -2, +1]/dx^2$

Matrix-Vector product $\mathbf{A} \cdot \mathbf{u} = \mathbf{f}$ with matrix $A \in R^{N-2 \times N}$

$$\mathbf{A} = \begin{bmatrix} -1 & +2 & -1 & 0 & \dots & \\ 0 & -1 & +2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \dots & \dots & 0 & -1 & +2 & -1 \end{bmatrix}$$

Dirichlet BC implementation

$$u_0 = u_L \text{ \& } u_N = u_R$$

$$\begin{bmatrix} +1 & 0 & 0 & \dots & & \\ -1 & +2 & -1 & 0 & \dots & \\ 0 & -1 & +2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \dots & \dots & 0 & -1 & +2 & -1 \\ \dots & \dots & \dots & 0 & 0 & +1 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} u_B \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}$$

Neumann BC implementation

$$u_1 - u_0 = g_L$$

$$\begin{bmatrix} +1 & -1 & 0 & \dots & & \\ -1 & +2 & -1 & 0 & \dots & \\ 0 & -1 & +2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \dots & \dots & 0 & -1 & +2 & -1 \\ \dots & \dots & \dots & 0 & -1 & +1 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} g_L \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ g_R \end{bmatrix}$$

Analytical solution for $f=1$

$$u(x) = -\frac{1}{2}x^2 + C_1x + C_0 \quad \Rightarrow \quad u'(x) = -x + C_1$$

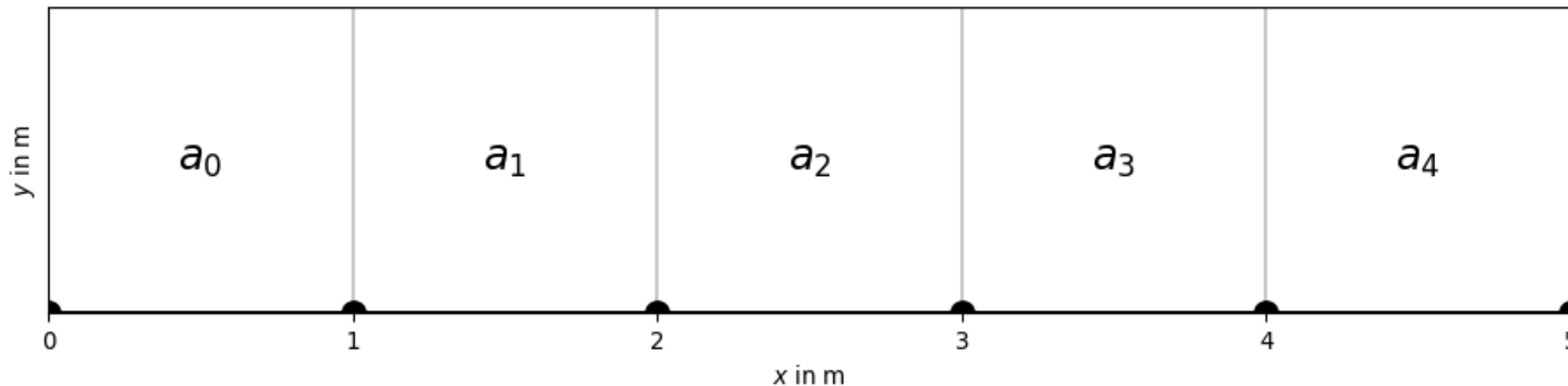
BC $x=0$	BC $x = X$	C_0	C_1
Dirichlet	Dirichlet	u_0	$X/2 + (u_X - u_0)/X$
Dirichlet	Neumann	u_0	$u'_X + X$
Neumann	Dirichlet	u'_0	$u_X - u'_0X + X^2/2$

Tasks (last exercise)

1. Create a system matrix for unit quantities
2. Implement Dirichlet BC on one and Neumann on other side
3. Solve system for different right-hand sides:
 - no source at all
 - single source in the middle or at the boundary
 - several sources with different strengths (& signs)
 - source on part of the domain
4. Always plot the solution and its Laplacian

The general case

$\Delta x \neq 1$ & $a \neq 1$ with $\mathbf{x} \in \mathbf{R}^N$ and $a \in R^{N-1}$ (in-between)



$\frac{\partial u}{\partial x}$ live on .5 space like $a \Rightarrow a/\text{np.diff}(x)$

second derivative: `np.diff(a*np.diff(x))/...`

Coupling coefficients

$$A_{i,i-1} = C_i^{left} = -a_{i-1}/(x_i - x_{i-1})/(x_{i+1} - x_{i-1}) \cdot 2$$

$$A_{i,i+1} = C_i^{right} = -a_i/(x_{i+1} - x_i)/(x_{i+1} - x_{i-1}) \cdot 2$$

$$\begin{bmatrix} +1 & 0 & 0 & \dots \\ C_1^L & -(C_1^L + C_1^R) & C_1^R & 0 & \dots \\ \vdots & \vdots & \ddots & \vdots & \\ \dots & \dots & 0 & C_N^L & -(C_N^L + C_N^R) & C_N^R \\ \dots & \dots & 0 & & -1 & +1 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} u_E \\ f_1 \\ \vdots \\ f_N \\ g_B \Delta \end{bmatrix}$$

Tasks for today

1. Write a function solving the general Poisson equation in 1D with FD
 - Boundary conditions: Dirichlet left, Neumann right
 - Input: \mathbf{x} vector, \mathbf{a} vector, source vector \mathbf{f} , boundary terms
 - Output: system matrix \mathbf{A} , right-hand-side vector \mathbf{b}
2. Test the function with $\Delta x = 1$ and $a = 1$ and compare with above
3. Play around with different boundary terms (values, gradients)
4. Divide the “subsurface” in regions with different a and play
5. Compute the solution for different source fields
6. Test whether a non-equidistant discretisation works likewise

Symmetry

$$A_{i+1,i} = C_{i+1}^{left} = a_i / (x_{i+1} - x_i) / (x_{i+2} - x_i) \cdot 2$$

$$A_{i,i+1} = C_i^{right} = a_i / (x_{i+1} - x_i) / (x_{i+1} - x_{i-1}) \cdot 2$$

only symmetric if Δx is constant around x_i , better take $a_i / (\Delta x_i)^2$

$$A_{i,i-1} = C_i^{left} = a_{i-1} / (x_i - x_{i-1})^2$$

$$A_{i,i+1} = C_i^{right} = a_i / (x_{i+1} - x_i)^2$$

\Rightarrow inaccuracies expected for non-equidistant discretization

A closer look at the Dirichlet boundary

$$\begin{bmatrix} +1 & 0 & 0 & \dots & \dots \\ C_1^L & -(C_1^L + C_1^R) & C_1^R & 0 & \dots \end{bmatrix} \begin{bmatrix} u_B \\ f_1 \end{bmatrix}$$

1. C can be differently scaled from 1 \Rightarrow multiply with C_i^L
2. Matrix is non-symmetric

$$\begin{bmatrix} C_1^L & 0 & 0 & \dots & \dots \\ C_1^L & -(C_1^L + C_1^R) & C_1^R & 0 & \dots \end{bmatrix} \begin{bmatrix} u_B C_i^L \\ f_1 \end{bmatrix}$$

A closer look at the Neumann boundary

$$\begin{bmatrix} \dots & \dots & 0 & C_N^L & -(C_N^L + C_N^R) & C_N^R \\ \dots & \dots & & 0 & -1 & +1 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} f_N \\ g_B \Delta x_N \end{bmatrix}$$

1. C can be differently scaled from 1 \Rightarrow multiply with C_N^R
2. Matrix is non-symmetric \Rightarrow multiply with -1

$$\begin{bmatrix} \dots & \dots & 0 & C_N^L & -(C_N^L + C_N^R) & C_N^R \\ \dots & \dots & & 0 & C_N^R & -C_N^R \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} f_N \\ -g_B \Delta x_N C_N^R \end{bmatrix}$$

Dirichlet BC implementation way 2

$$-u_L + 2u_1 - u_2 = f_1 \Rightarrow 2u_1 - u_2 = f_1 + u_L$$

$$\begin{bmatrix} +2 & -1 & 0 & \dots & & \\ -1 & +2 & -1 & 0 & \dots & \\ 0 & -1 & +2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \dots & \dots & 0 & -1 & +2 & -1 \\ \dots & \dots & \dots & 0 & -1 & +2 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} f_1 + u_L \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + u_R \end{bmatrix}$$

Neumann BC implementation way 2

$$-u_L + 2u_1 - u_2 = f_1 \quad u_1 - u_0 = g_L \Rightarrow u_2 - u_1 = f_1 + g_L$$

$$\begin{bmatrix} +1 & -1 & 0 & \dots & & \\ -1 & +2 & -1 & 0 & \dots & \\ 0 & -1 & +2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ \dots & \dots & 0 & -1 & +2 & -1 \\ \dots & \dots & \dots & 0 & -1 & +1 \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} f_1 + g_L \\ f_2 \\ f_3 \\ \vdots \\ f_{N-1} \\ f_{N-1} + g_R \end{bmatrix}$$