Institut für Strömungsmechanik
und Umweltphysik im Bauwesen

Assignment 1

1D stationary mass transport equation - analytical and numerical solutions (by using finite difference method)

Date of issue: 11.04.2024

Date of submission: 08.05.2024 (13:00)

Requirements

- working groups of 2-3 people
- all codes and plots have to be submitted via e-mail to krishna@hydromech.uni-hannover.de
- the present document has to be signed digitally
- codes have to have headers stating the date and names and matriculation numbers of all participants
- tasks marked with a * are optional
- add comments to your codes (at least one comment for each individual information)
- all plots have to have labeled axes and legends
- always use the given variables and functions' names

Declaration of Independence

I hereby declare that the present assignment was worked on independently without help from a third party. No codes or other results were taken from other homework. I agree to a plagiarism assessment.

Surname	First name	Matriculation number	Signature
Duvvuru	Lokesh	10063226	
Tersteegen	Marie	10061302	
Toumi	Yasmes Abdeldjalil	10064473	

1 Analytical solution of the 1D mass transport equation

The stationary 1D advection-diffusion equation is

$$v\frac{\mathrm{d}c}{\mathrm{d}x} - D\frac{\mathrm{d}^2c}{\mathrm{d}x^2} = 0, \quad x \text{ in } [0,1], \tag{1}$$

where $c = c(x)$ is the concentration of a dissolved substance (primary variable), v and D denote the advection velocity and the diffusion coefficient correspondingly. The analytical solution of equation (1) with the boundary conditions $c(x=0) = 0$ and $c(x=1) = 1$ reads:

$$c(x) = \frac{\exp(Pe \cdot x) - 1}{\exp(Pe) - 1}. \tag{2}$$

1. Write the script `analytic.m` that plots the analytical solution (2) for a predefined list of Péclet numbers $Pe = \frac{v \cdot 1}{D} = (-10, -2, 0, 2, 10)$.

   Advection vs. Diffusion

   - define vector **x** which dependents on the number of discretization points n
   - write the anonymous function $f_{ana} = @(\mathbf{x}, Pe)$ (see matlab tutorial section 5.3).
   - use a loop to plot the solution for each Péclet number in one plot (useful commands: `plot`, `hold on`, `hold off`)
   - save the figure as `anaSolution.png`
   - useful matlab-functions: `length`, `size`, `legend`, and `num2str` (Hints: You can convert numbers to strings by using the function `num2str`, e.g., title(['Pe='num2str(Pe)]).)

2. How does the solution corresponding to $Pe = 0$ look like, and why is it not plotted?

   For Pe = 0 the denominator exp(0) − 1 = 1 − 1 = 0 and c(x) is therefore not defined. Accordingly it can not be plotted.

2 Numerical Solution of 1D stationary mass transport equation

The dimensionless 1D stationary advection-diffusion equation

$$Pe\frac{\mathrm{d}c}{\mathrm{d}x} - \frac{\mathrm{d}^2c}{\mathrm{d}x^2} = 0, \quad x \text{ in } [0,1], \tag{3}$$

with the boundary conditions $c(x=0)=0$ and $c(x=1)=1$ is considered. The finite difference method will be applied for solving the equation numerically.

The grid Péclet number is defined as

$$Pe_G = \frac{v\Delta x}{D}. \tag{4}$$

The grid Péclet number measures the comparison of the diffusion time scale ($t_d = \Delta x^2/D$) to the time scale of the advection ($t_a = \Delta x/v$) over the length of a single grid (Δx).

2.1 Matlab functions for generating special matrices

In the following, functions for generating special matrices shall be implemented.

Useful Matlab commands: `zeros`, `ones`, `eye`, `diag`, `disp`, `if`.

In Matlab you can access the elements of a matrix by addressing their row and column. An example:

```
A=magic(5)        % arbitrary matrix as example
A(2,3)            % returns the value in the second row and third column
n = 4
A(n, 3)           % returns the value in the nth row and third column
A(2,3) = 5        % value in the second row and third column is set to five
A(end, 3) = 7     % value in the last row and third column is set to seven
A(:, end) = 1     % overwrite values in the entire first column with one
```

1. Create the function `A = tridiag(n, u1, h, o1)`. This function creates a matrix A of size $n \times n$. The value `h` is on the main diagonal, `u1` on the lower first diagonal and `o1` on the upper first diagonal. All remaining entries are zero. If n is lower than three, an error message shall be displayed and the return value is $A = 0$.

   An example:

```
>> A = tridiag(5, 1, 2, 3)
A =
     2     3     0     0     0
     1     2     3     0     0
     0     1     2     3     0
     0     0     1     2     3
     0     0     0     1     2
```

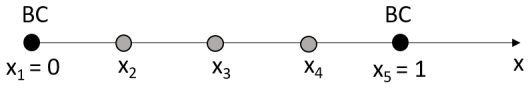
2. Create the function `A = tridiagcyc(n, u1, h, o1)`. This function creates a matrix A of size $n \times n$. The value `h` is on the main diagonal, `u1` on the lower first diagonal and `o1` on the upper first diagonal. In the upper right corner is the value of the lower first diagonal. In the lower left corner is the value of the upper first diagonal. All remaining entries are zero. If n is lower than three, an error message shall be displayed and the return value is $A = 0$.

   An example:

```
>> A = tridiagcyc(5, 1, 2, 3)
A =
     2     3     0     0     1
     1     2     3     0     0
     0     1     2     3     0
     0     0     1     2     3
     3     0     0     1     2
```

2.2 Central finite difference discretization

The domain is represented by five equally spaced nodes. The two nodes located at $x_1 = 0$ and $x_5 = 1$ are boundary nodes.



1. At which nodes are the values known? At which nodes are the values unknown? How many unknown nodes are there?

known: x_1 and x_5 because of Dirichlet boundary conditions $c(x_1)=0$, $c(x_2)=1$

unknown: x_2, x_3, x_4 → 3 unknown nodes

2. Write down the discretized equation for every node. Discretize the first and second derivative terms in equation (3) with central finite difference scheme.

for $i=2,3,4$:

$$Pe\left(\frac{c_{i+1}-c_{i-1}}{2\Delta x}\right)-\left(\frac{c_{i+1}-2c_i+c_{i-1}}{\Delta x^2}\right)=0 \quad |\cdot\Delta x^2$$

$$Pe_G\left(\frac{c_{i+1}-c_{i-1}}{2}\right)-\left(c_{i+1}-2c_i+c_{i-1}\right)=0$$

$$\Leftrightarrow c_{i-1}\left(\frac{Pe_G}{2}-1\right)+c_i(2)+c_{i+1}\left(-\frac{Pe_G}{2}-1\right)=0$$

$i=2$: $c_3\left(\frac{Pe_G}{2}-1\right)+2c_2+c_1\left(-\frac{Pe_G}{2}-1\right)=0$

$i=3$: $c_4\left(\frac{Pe_G}{2}-1\right)+2c_3+c_2\left(-\frac{Pe_G}{2}-1\right)=0$

$i=4$: $c_5\left(\frac{Pe_G}{2}-1\right)+2c_4+c_2\left(-\frac{Pe_G}{2}-1\right)=0$

for $i=1$ $\quad c_1=c_L=0$

for $i=2$ $\quad c_5=c_R=1$

3. Rewrite the system of equations to the following matrix-vector form,

$$\underbrace{\{Pe_G \quad - \quad\}}_{\underline{\underline{A}}}\vec{c}=\vec{rhs}, \tag{5}$$

where $\mathbf{P}_c \in \mathbb{R}^{n\times n}$ represents the discretized advection term with central scheme, and $\mathbf{K} \in \mathbb{R}^{n\times n}$ is the discretized diffusion term. On the right-hand-side of equation (5), $\overrightarrow{rhs} \in \mathbb{R}^{n\times 1}$ represents a nodal-wise vector (with n nodes) which contains information of the boundary conditions. The vector $\vec{c}$ is defined as

$$\vec{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \tag{6}$$

and the right-hand-side vector $\overrightarrow{rhs}$, in this case, only contains the values at the Dirichlet boundaries $\vec{c_d}$

$$\overrightarrow{rhs} = \vec{c_d} = \begin{pmatrix} c_L \\ 0 \\ 0 \\ 0 \\ c_R \end{pmatrix} \tag{7}$$

Define the matrices $\mathbf{P}_c$ and $\mathbf{K}$.

$$\mathbf{P_c} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

$$\mathbf{K} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{9}$$

4. The system of equations (5) can now be written as

$$\mathbf{A}\overrightarrow{c} = \overrightarrow{rhs}. \tag{10}$$

Define the matrix $\mathbf{A}$

$a_1 := -\frac{Pe_G}{2} - 1$

$a_2 := \frac{Pe_G}{2} - 1$

$$\mathbf{A^{n\times n}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ a_1 & 2 & a_2 & 0 & 0 \\ 0 & a_1 & 2 & a_2 & 0 \\ 0 & 0 & a_1 & 2 & a_2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5. What do the entries in the first and last row of $\mathbf{A}$ have to look like to correctly include the boundary conditions? Respect the rules of matrix-vector multiplication, as e.g.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_{11}\cdot b_1 + a_{12}\cdot b_2 + a_{13}\cdot b_3 \\ a_{21}\cdot b_1 + a_{22}\cdot b_2 + a_{23}\cdot b_3 \end{pmatrix}.$$

first row: 1 0 0 0 0

last row: 0 0 0 0 1

We have Dirichlet boundary conditions $c_1 = c_R = 0$ and $c_5 = c_L = 1$

Therefore (first row) $\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \overset{!}{=} c_R = (1\,0\,0\,0\,0)\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = c_1$ ✓

(last row) $\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \overset{!}{=} c_L = (0\,0\,0\,0\,1)\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = c_5$ ✓

2.3 Numerical implementation of the scheme with Matlab

To solve the unknown nodal vector $\overrightarrow{c}$ in equation (10), you need to generate the $n \times n$ matrix $\mathbf{A}$ and the right-hand-side vector $\overrightarrow{rhs}$ first.

1. Create the function `[c x PeG] = ADEstationary(n, Pe, cL, cR)`, which calculates the numerical solution $\vec{c}$ for the problem given in section 2.2 with the finite difference method.

With input values...	n	number of nodes
	Pe	Péclet number Pe
	cL	boundary condition on the left side
	cR	boundary condition on the right side

With output values...	c	approximate solution $\vec{c}$, also contains c_L and c_R
	x	vector containing the coordinates of $\vec{c}$
	PeG	vector containing the grid Péclet number

   Follow this schedule, please:

   1. generate the grid and calculate necessary parameters;
   2. set up the system of equations: define $\mathbf{A}$ and $\overrightarrow{rhs}$;
   3. solve system of equations (10).

   Tip: Use your `tridiag` function to generate the matrix.

2. Copy your script `analytic.m` into `stationary.m`. Then extend the script so that it plots the numerical solution together with the analytical solution for the given Péclet numbers $Pe = (-10, -2, 0, 2, 10)$. Please solve the problem with 20 nodes ($n = 20$).
   - use a new subplot for each Péclet number (useful command: `subplot`)
   - save the figure as `numSolution.png`

2.4 A more general finite difference scheme

A general finite difference scheme for the discretization of the stationary advection-diffusion equation (3) reads

$$Pe_G \left[\alpha \left(c_i - c_{i-1}\right) + \left(1 - \alpha\right)\left(c_{i+1} - c_i\right)\right] - \left(c_{i-1} - 2c_i + c_{i+1}\right) = 0. \tag{11}$$

The choice of α determines the difference scheme for the discretization of the advection term as

$\alpha = 1$: backward;
$\alpha = 0$: forward;
$\alpha = 0.5$: central.

Using the matrix-vector form introduced in the previous section, the matrix $\mathbf{A}$ in equation (10) can be written as:

$$\mathbf{A} = Pe_G \left[\alpha \mathbf{P}_b + \left(1 - \alpha\right)\mathbf{P}_f\right] - \mathbf{K}, \tag{12}$$

and

$$\overrightarrow{rhs} = \vec{c}_d, \tag{13}$$

where $\mathbf{P}_b$ represents the discretization of the advection term with the backward difference scheme, and $\mathbf{P}_f$ represents the corresponding discretization with the forward difference scheme.

1. Define the matrices $\mathbf{P}_b$ and $\mathbf{P}_f$.

$$\mathbf{P}_b = \begin{bmatrix} 0 & 0 & & & \\ -1 & 1 & 0 & 0 & \\ & -1 & 1 & 0 & \\ & & -1 & 1 & 0 \\ 0 & & & 0 & 0 \end{bmatrix}$$

$$\mathbf{P}_f = \begin{bmatrix} 0 & 0 & & & \\ & -1 & 1 & 0 & \\ & & -1 & 1 & \\ & & & -1 & 1 \\ & 0 & & 0 & 0 \end{bmatrix}$$

2. Copy the function `ADEstationary` into `ADEstationary2` and modify the code (`ADEstationary2`) in such a way that the matrix $\mathbf{A}$ is generated by using the general discretization scheme (12) instead of central differences.

3. Write the script `stationary2.m` that plots the analytical and the numerical solutions for $\alpha = (0,\ 0.5,\ 1)$ and $Pe = (-10, -2, 0, 2, 10)$. Use your script `stationary.m` as a base.

   - loop over α
   - add a legend ('analytical','backward','forward','central')
   - use a new subplot for every Péclet number (useful commands: `subplot`, `hold`)
   - save the plot as `generalScheme.png`

   Tip: `num2str` converts numbers into strings. Strings can be concatenated like vectors using square brackets `[]`. An example: `title(['Pe_G = ' num2str(PeG) ' :) '])`.

2.5 Stability analysis

1. Set the number of nodes in your script `stationary2.m` to 11. Investigate the numerical solutions corresponding to different grid Péclet numbers as listed in the table below. Which α value fits best for which grid Péclet number (use the table below)? Is the forward difference useful at all?

P_{eG}	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
1.0	worst	best	ok
1.5	worst	best	ok
2.0	det(A)=0	ok	ok
3.0	worst	ok	ok

for central and backwards

for P_{eG} 2.0 and 3.0 central and backwards follow the analytical solution equally well (visually).

Forward difference is not useful for these parameters at all.

For $P_{eG}=2$ there is no solution at all for forward as A is not invertable.

2. Is there a scheme that matches the analytical solution? Which scheme over- and which scheme underestimates advective transport process (use the table below)?

P_{eG}	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
1.0	u	m	o
1.5	both	m (u)	o
2.0	/	u	o
3.0	o	u	o

m: match
o: overestimation
u: underestimation

m(u): Almost matches, but slightly underestimates

*3. Please give reasons for the presented over-/underestimation of the numerical solutions.

The sign of the lowest order error (Δx for for/backward) looking at derivation of the schemes is different.
For backward difference, combined with a positive slope, this yields a positive error, therefore an overestimation.
For forward the error is negative, which can lead to underestimation.

2.6 Convergence analysis

1. Compare the analytical solution to the numerical solution from the central difference scheme. Copy your script `stationary.m` into `convergence.m` and modify it to do so. Use a grid Péclet number of 1. Calculate the deviation e_1 as geometrical distance between the two solutions at locations x_i:

$$e_1 = \sqrt{\sum_{i=1}^{3} \left(c_{ana}(x_i) - c_{num}|_{x_i}\right)^2}, \quad x_i = (0.25, 0.5, 0.75). \tag{14}$$

   - analyze the deviation e_1 depending on the grid size Δx
   - use the function `norm` to calculate e_1
   - test for grid sizes $\Delta x = [0.25, 0.125, 0.0625, 0.03125]$
   - plot e_1 over the grid sizes Δx using the functions `plot`, `semilogx`, `semilogy` or `loglog`
   - save the plot as `convergence.png`

2. Which grid do you choose and why?

For $\Delta X = 0.03125$ the error is the smallest. As this is a rather simple problem, we do not have to worry too much about computation time. Therfor we choose $\Delta X = 0.03125$ as it yields the smallest error

*3. Most problems that one may face in reality are nonlinear, and the analytical solutions usually do not exist. To carry out a convergence analysis, in this case, requires to compare the measured errors between two adjacent refinements. Please copy your script `convergence.m` into `convergence2.m` and modify it. Test the same grid sizes Δx as you tested in 1. and plot e_1 over the grid sizes Δx and save the plot as `convergence2.png`. You can take the Δx value from either the finer or coarser mesh when calculating the difference between two meshes. However, please keep using the same rule all time.

2.7 Taylor expansion and finite differences

1. Deduce the central finite differences scheme for the first derivative in space $\mathrm{d}f/\mathrm{d}x$ and an arbitrary finite differences scheme for the second derivative in space $\mathrm{d}^2f/\mathrm{d}x^2$ from the Taylor expansion.

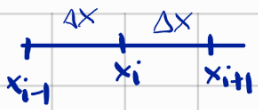
see attached sheet below

2. What is the truncation error of the schemes? Underline the corresponding term in your Taylor expansion.

When the Taylorexpansion is truncated, the function f is approximated by it. The more terms are taken into account, the further the Taylor expansion can describe f. As it is only an approximation, we have a truncation error depending on where we truncate. The lowest order truncation error for (2.7.1) are marked in red.

In consistent schemes the truncation error vanishes for $\Delta x \to 0$

2.7 Taylor expansion and finite difference



Taylor expansion of a function f at point x is defined as $f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d^n f}{dx^n}\Big|_{x_0} (x-x_0)^n$

To deduce central finite difference scheme, we expand around x_i and evaluate at x_{i-1} and x_{i+1}

$$f(x_{i-1}) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d^n f}{dx^n}\Big|_{x=x_i} \underbrace{(x_{i-1}-x_i)}_{-\Delta x}{}^n = f_i - \frac{df}{dx}\Big|_{x_i}\Delta x + \frac{1}{2}\frac{d^2f}{dx}\Big|_{x_i}\Delta x^2 - \frac{1}{6}\frac{d^3f}{dx}\Big|_{x_i}\Delta x^3 + \dots \quad (I)$$

$$f(x_{i+1}) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d^n f}{dx^n}\Big|_{x=x_i} \underbrace{(x_{i+1}-x_i)}_{\Delta x}{}^n = f_i + \frac{df}{dx}\Big|_{x_i}\Delta x + \frac{1}{2}\frac{d^2f}{dx}\Big|_{x_i}\Delta x^2 + \frac{1}{6}\frac{d^3f}{dx}\Big|_{x_i}\Delta x^3 + \dots \quad (II)$$

substract (I) from (II):

$$(II)-(I) \Longrightarrow f(x_{i+1}) - f(x_{i-1}) = 2\frac{df}{dx}\Big|_{x_i}\Delta x + \frac{2}{6}\frac{d^3f}{dx^3}\Big|_{x_i}\Delta x^3 + \dots$$

$$\Longleftrightarrow \frac{df}{dx} = \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x} - \frac{1}{6}\frac{d^3f}{dx^3}\Big|_{x_i}\Delta x^2 + \dots$$

We can truncate the Taylor series and approximate with truncation error

$$\Rightarrow \frac{df}{dx}\Big|_{x_i} = \frac{f_{i+1} - f_i}{2\Delta x} \underbrace{- \frac{1}{6}\frac{d^3f}{dx^3}\Big|_{x_i}\boxed{\Delta x^2} + \dots}_{\text{truncate}}$$

lowest truncation error in Δx of second order $\mathcal{O}(\Delta x^2)$

We receive

$$\frac{df}{dx}\Big|_{x_i} = \frac{f_{i+1} - f_i}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

For second order derivative, we add (I)+(II)

(I)+(II) $\Rightarrow f(x_{i-1}) + f(x_{i+1}) = 2f_i + 0\left.\frac{df}{dx}\right|_{x_i}\Delta x + \frac{2}{2}\left.\frac{d^2f}{dx^2}\right|_{x_i}\Delta x^2 + 0\left.\frac{d^3f}{dx^3}\right|_{x_i}\Delta x^3 + \frac{2}{24}\left.\frac{d^4f}{dx^4}\right|_{x_i}\Delta x^4 + \dots$

$$\Longleftrightarrow \left.\frac{d^2f}{dx^2}\right|_{x_i} = \frac{f(x_{i+1}) + f(x_{i-1}) - 2f_i}{\Delta x^2} + \boxed{\frac{1}{12}\left.\frac{d^4f}{dx^4}\right|_{x_i}\Delta x^2} + \dots$$

truncate

lowest truncation error is $\mathcal{O}(\Delta x^2)$

$$\Rightarrow \left.\frac{d^2f}{dx^2}\right|_{x_i} = \frac{f(x_{i+1}) + f(x_{i-1}) - 2f_i}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$