Exercise 4

High Performance Computing for Science and Engineering

October 21, 2016

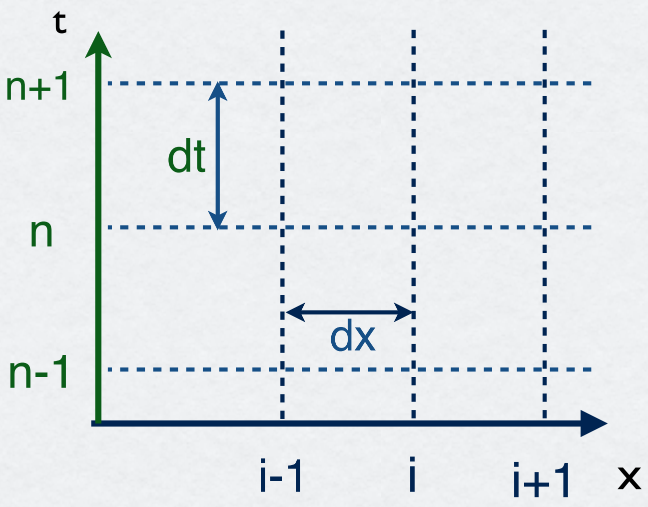
CSElab
Computational Science & Engineering Laboratory
http://www.cse-lab.ethz.ch

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Exercise 4

- Tasks:
    - Analyse stability of the discretized diffusion equation
    - Parallelize code from ex.1 with OpenMP
        - try out different compilers and OpenMP flags
        - present their effect on weak and strong scaling plots
    - Test the effect of viscosity on two moments of the field

1D Diffusion: finite differences



$$\frac{\partial \rho(x,t)}{\partial t} = D\frac{\partial^2 \rho(x,t)}{\partial x^2}$$

- Discretize equation on uniform grid: $\rho_i^{(n)} = \rho(x_i, t_n)$
- *n* is the index of time step: $t_n = n \cdot \Delta t$
- i is the index of spatial discretization: $x_i = i \cdot \Delta x$
- track evolution of ρ at discrete space/time points

- Approximate derivatives by looking for y(x) around xi:

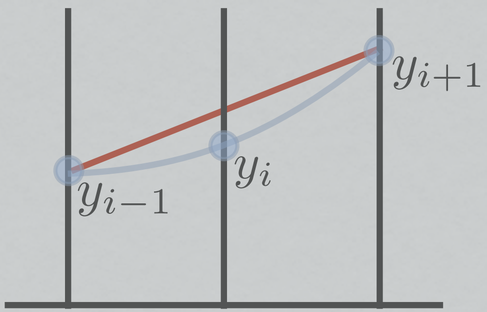
$$\rho_{i+1}^{(n)} = \rho_i^{(n)} + \delta x \cdot \left.\frac{\partial \rho}{\partial x}\right|_{x_i,t_n} + \frac{\delta x^2}{2} \cdot \left.\frac{\partial^2 \rho}{\partial x^2}\right|_{x_i,t_n} + \frac{\delta x^3}{6} \cdot \left.\frac{\partial^3 \rho}{\partial x^3}\right|_{x_i,t_n} + \text{h.o.t.}$$

$$\rho_{i-1}^{(n)} = \rho_i^{(n)} - \delta x \cdot \left.\frac{\partial \rho}{\partial x}\right|_{x_i,t_n} + \frac{\delta x^2}{2} \cdot \left.\frac{\partial^2 \rho}{\partial x^2}\right|_{x_i,t_n} - \frac{\delta x^3}{6} \cdot \left.\frac{\partial^3 \rho}{\partial x^3}\right|_{x_i,t_n} + \text{h.o.t.}$$

- Obtain a finite difference approximation for the second derivative:

$$\left.\frac{\partial^2 \rho}{\partial x^2}\right|_{x_i,t_n} = \frac{\rho_{i+1}^{(n)} - 2\rho_i^{(n)} + \rho_{i-1}^{(n)}}{\delta x^2} + \underbrace{\boxed{\text{h.o.t.}}}_{\text{error}}$$

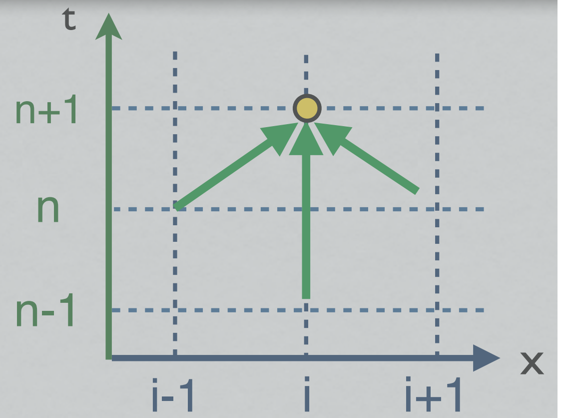
1D Diffusion: time stepping

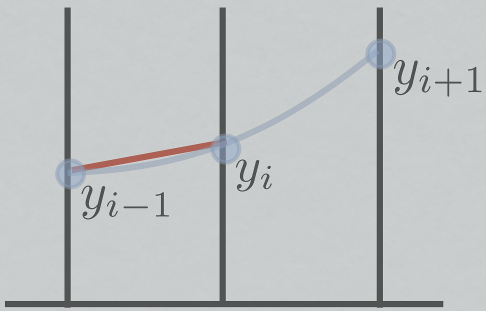


Central difference

$$\left.\frac{\partial \rho}{\partial t}\right|_{x_i,t_n} = \frac{\rho_i^{(n+1)} - \rho_i^{(n-1)}}{2\delta t} + \text{h.o.t.}$$

$$\frac{\rho_i^{(n+1)} - \rho_i^{(n-1)}}{2\delta t} = D\frac{\rho_{i+1}^{(n)} - 2\rho_i^{(n)} + \rho_{i-1}^{(n)}}{\delta x^2}$$

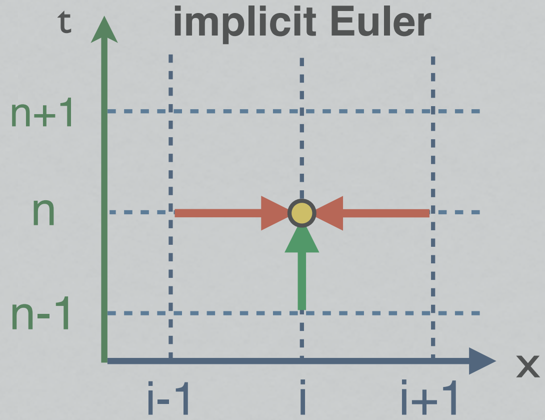


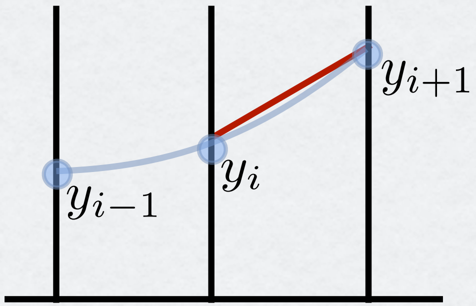


Backward difference

$$\left.\frac{\partial \rho}{\partial t}\right|_{x_i,t_n} = \frac{\rho_i^{(n)} - \rho_i^{(n-1)}}{\delta t} + \text{h.o.t.}$$

$$\frac{\rho_i^{(n)} - \rho_i^{(n-1)}}{\delta t} = D\frac{\rho_{i+1}^{(n)} - 2\rho_i^{(n)} + \rho_{i-1}^{(n)}}{\delta x^2}$$

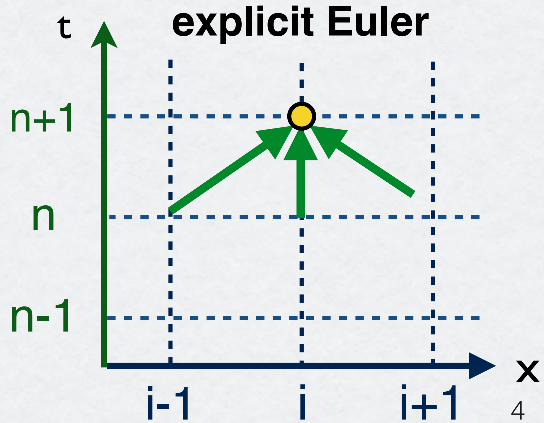




Forward difference

$$\left.\frac{\partial \rho}{\partial t}\right|_{x_i,t_n} = \frac{\rho_i^{(n+1)} - \rho_i^{(n)}}{\delta t} + \text{h.o.t.}$$

$$\frac{\rho_i^{(n+1)} - \rho_i^{(n)}}{\delta t} = D\frac{\rho_{i+1}^{(n)} - 2\rho_i^{(n)} + \rho_{i-1}^{(n)}}{\delta x^2}$$



von Neumann stability analysis

Is this going to work? $$\frac{\rho_i^{(n+1)} - \rho_i^{(n)}}{\delta t} = D\frac{\rho_{i+1}^{(n)} - 2\rho_i^{(n)} + \rho_{i-1}^{(n)}}{\delta x^2}$$

von Neumann stability analysis

- used to determine stability of finite difference schemes
- assumes that you are making a **round-off error** $\rho_i^{(n)} = \hat{\rho}_i^{(n)} + \epsilon_i^{(n)}$
- true solution $\hat{\rho}_i^{(n)}$ satisfies equation identically
- therefore, error also satisfies equation: $$\frac{\epsilon_i^{(n+1)} - \epsilon_i^{(n)}}{\delta t} = D\frac{\epsilon_{i+1}^{(n)} - 2\epsilon_i^{(n)} + \epsilon_{i-1}^{(n)}}{\delta x^2}$$
- assume that the error is composed of terms of form:

1D $\epsilon_l^{(n)} = \lambda^n e^{ik_x x_l}$

$\epsilon_{l,k}^{(n)} = \lambda^n e^{ik_x x_l} e^{ik_y y_k}$ 2D

oscillations in space

amplified or damped in time

- why?

(Fourier expansion of the solution of a linear differential equation with periodic BC)

Stability of explicit Euler

$$\frac{\epsilon_i^{(n+1)} - \epsilon_i^{(n)}}{\delta t} = D\frac{\epsilon_{i+1}^{(n)} - 2\epsilon_i^{(n)} + \epsilon_{i-1}^{(n)}}{\delta x^2} \qquad\qquad \epsilon_l^{(n)} = \lambda^n e^{ikx_l}$$

- evolve error with time stepping scheme and check whether it blows up!

$$\lambda^{n+1}e^{ikx_l} = \lambda^n e^{ikx_l} + \frac{D\delta t}{\delta x^2}\lambda^n\left(e^{ikx_{l+1}} - 2e^{ikx_l} + e^{ikx_{l-1}}\right)$$

- where: $x_{l-1} = x_l - \delta x \qquad x_{l+1} = x_l + \delta x$

$$\lambda = 1 + \frac{2D\delta t}{\delta x^2}\left[\cos\left(k\delta x\right) - 1\right]$$

- error is bounded (does not explode) if **|λ| < 1**

$$-1 \leq 1 + \frac{2D\delta t}{\delta x^2}\left[\cos\left(k\delta x\right) - 1\right] \leq 1$$

$$\delta t \leq \frac{\delta x^2}{D\left[1 - \cos\left(k\delta x\right)\right]}$$

Always satisfied

- worst case for stability condition $\cos\left(k\delta x\right) = -1 \longrightarrow \boxed{\delta t \leq \frac{\delta x^2}{2D}}$

2D Diffusion, explicit Euler

Diffusion of **ρ** (e.g. heat flow) can be described by diffusion equation:

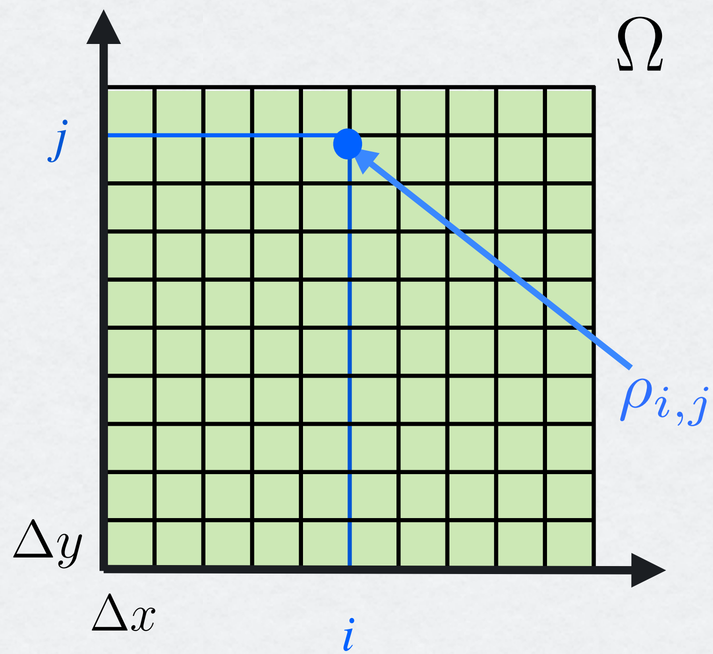
$$\frac{\partial \rho(x,y,t)}{\partial t} = D\nabla^2 \rho(x,y,t)$$

ρ(x,y,t) - measure for the amount of heat at position (x,y) and time t

D - diffusion coefficient (constant here)

- Discretizing eq. (1) using forward Euler in time and central differences in space yields:

$$\frac{\rho_{i,j}^{(n+1)} - \rho_{i,j}^{(n)}}{\Delta t} = D\left(\frac{\rho_{i+1,j}^{(n)} - 2\rho_{i,j}^{(n)} + \rho_{i-1,j}^{(n)}}{\Delta x^2} + \frac{\rho_{i,j+1}^{(n)} - 2\rho_{i,j}^{(n)} + \rho_{i,j-1}^{(n)}}{\Delta y^2}\right)$$



- where *n* is the index of time step: $t_n = n \cdot \Delta t$
- i, j are indices of spatial discretization: $x_i = i \cdot \Delta x$, $y_j = j \cdot \Delta y$
- and $\rho_{i,j}^{n} = \rho(x_i, y_j, t_n)$

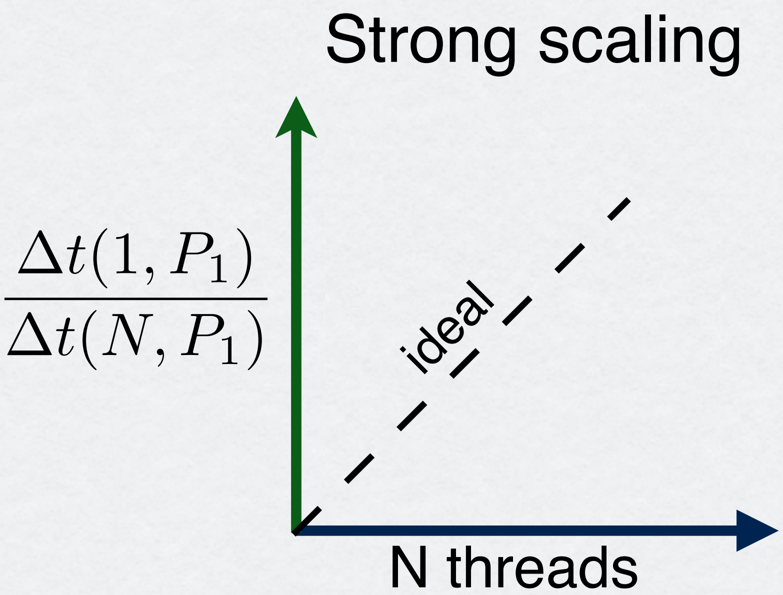
Scaling

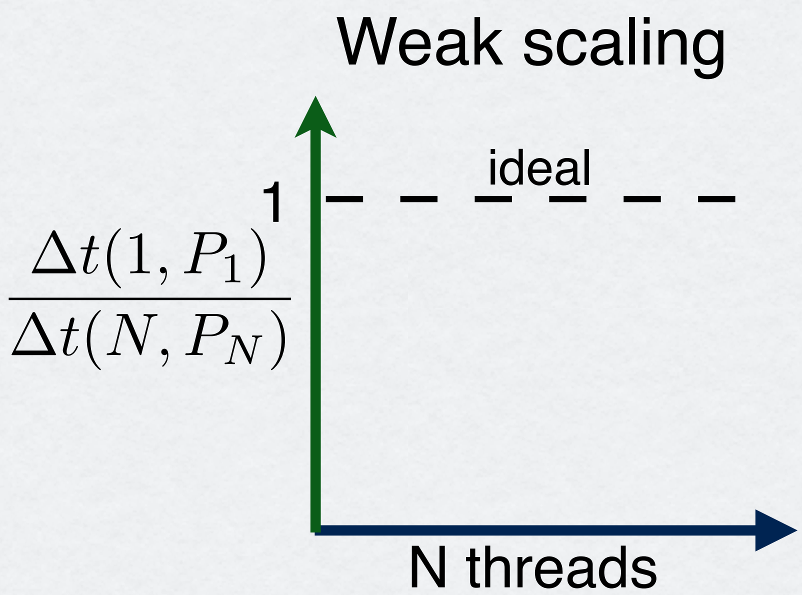
Strong scaling:

- speed-up on multiprocessors **without increase on problem size**

Weak scaling:

- speed-up on multiprocessors, while increasing the **problem size** ***proportionally*** to the increase in the **number of processors**





Report

- Hand-in
  - Code
  - PDF with results and comments
- Rule: feedback is proportional to the effort in reporting
  - Even if something does not work, tell us about it!