Stability analysis of the FTCS scheme

FTCS = forward in time, centered in space

Von Neumann stability analysis: for linear differential equations with constant coefficients

Diffusion equation

$$\frac{\partial T}{\partial t} = \kappa \cdot \frac{\partial^2 T}{\partial x^2}$$

Ansatz:

$$\epsilon_j^n = N_j^n - T_j^{\overset{\text{Index}}{\overset{\downarrow}{n}}} = \sum_{m=-M}^{M} A^{\overset{\text{Power}}{\overset{\downarrow}{n}}} \cdot e^{ikj\Delta x}$$

If |A| > 1 the solution is unstable

T: Temperature (analytical solution)
t: Time
x: Spatial coordinate
κ: Diffusion coefficient

ϵ: Error
N: Temperature (numerical solution)
n: Time index
j: Space index
i: Imaginary unit
k: Wave number ($k = \pi m/L$)
L: Length of the model domain
M: Maximum wave number ($M = L/\Delta x$)
Δx: Grid spacing

Stability analysis of the FTCS scheme

Finite differences:

$$\epsilon_j^{n+1} = \epsilon_j^n + \frac{\Delta t \cdot \kappa}{\Delta x^2} \cdot \left(\epsilon_{j+1}^n - 2\epsilon_j^n + \epsilon_{j-1}^n\right)$$

Insert ansatz:

$$A^{n+1} e^{ikj\Delta x} = A^n e^{ikj\Delta x} + \frac{\Delta t \cdot \kappa}{\Delta x^2} \cdot \left(A^n e^{ik(j+1)\Delta x} - 2A^n e^{ikj\Delta x} + A^n e^{ik(j-1)\Delta x}\right)$$

Divide by $A^n e^{ikj\Delta x}$:

$$\boxed{2\cos(\gamma) = e^{i\gamma} + e^{-i\gamma}}$$

Most dangerous when $\cos(k\Delta x) = -1$

$$A = 1 + \frac{\Delta t \cdot \kappa}{\Delta x^2} \cdot \left(e^{ik\Delta x} - 2 + e^{-ik\Delta x}\right) \overset{\downarrow}{=} 1 + \frac{2 \cdot \Delta t \cdot \kappa}{\Delta x^2} \cdot (\cos(k\Delta x) - 1)$$

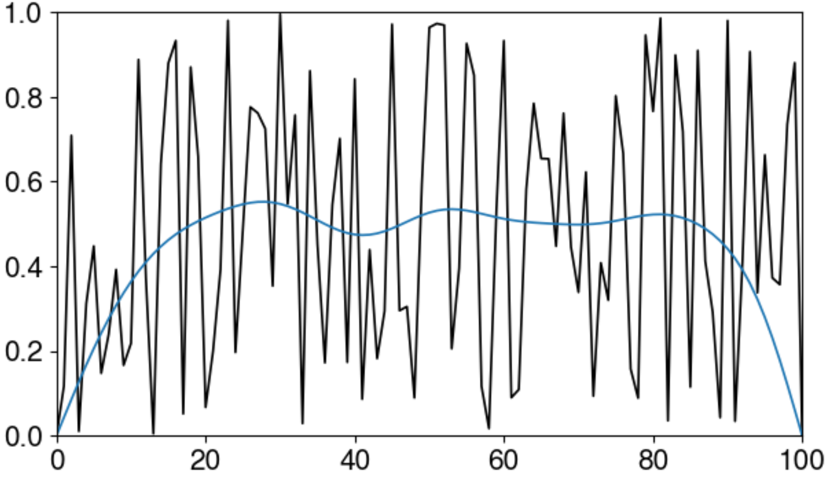
Stability analysis of the FTCS scheme

Condition for stability:

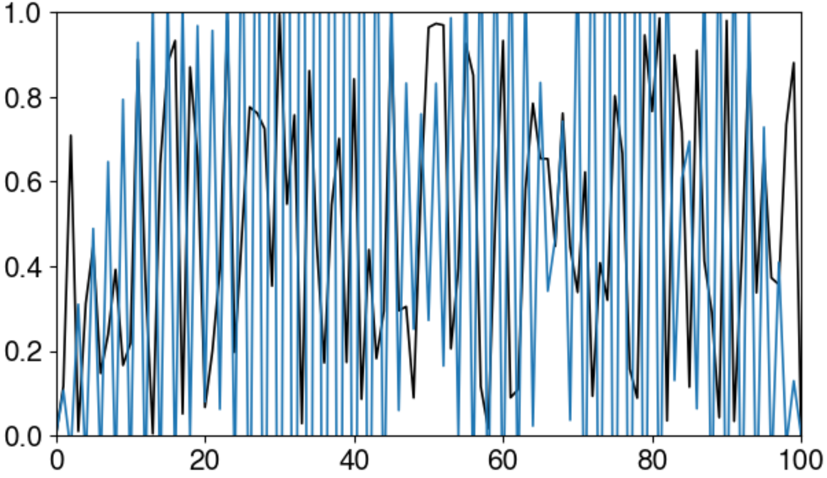
$$\left|1 + \frac{2 \cdot \Delta t \cdot \kappa}{\Delta x^2} \cdot (-2)\right| \leq 1$$

$$\rightarrow \quad \frac{\Delta t \cdot \kappa}{\Delta x^2} \leq \frac{1}{2}$$

$\Delta t = 0.40 \cdot \Delta x^2$



$\Delta t = 0.52 \cdot \Delta x^2$



Two-dimensional heat diffusion equation

$$\frac{\partial T}{\partial t} = \kappa \cdot \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) = \kappa \cdot \nabla^2 T$$

T: Temperature
t: Time
x,y: Spatial coordinates
κ: Diffusion coefficient

Finite differences

$$\nabla^2 T_{i,j}^n \approx \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{\Delta x^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{\Delta y^2}$$

$n = 1 \ldots n_t$, $i = 1 \ldots n_x$, $j = 1 \ldots n_y$

$$= \frac{T_{i-1,j}^n + T_{i+1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n}{h^2}$$

if $\Delta x = \Delta y = h$

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = \kappa \cdot \nabla^2 T_{i,j}^n \quad \rightarrow \quad T_{i,j}^{n+1} = T_{i,j}^n + \Delta t \cdot \kappa \cdot \nabla^2 T_{i,j}^n$$

Exercise 4

Rewrite your Fortran program from Exercise 3, Task 3 such that it solves the two-dimensional heat diffusion equation:

$$\frac{\partial T}{\partial t} = \kappa \cdot \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) = \kappa \cdot \nabla^2 T$$

1. Read in the control parameters from a namelist:
    - Number of grid points in x and y direction `nx`, `ny`
    - Diffusion coefficient `kappa` (start with `kappa=1`)
    - Simulation time `total_time` (start with `total_time=0.1`)
    - Time step constant `a`

The program should also run without a complete namelist → default values!

Exercise 4

2. Initialize the variables.
    - Temperature field: 2D array of size (`nx,ny`) with random numbers (`random`) or delta function (`spike`)
    - Grid spacing `h=1./(ny-1.)`
    - Time step `dt=a*h**2/kappa`
    - Number of time steps `nsteps=total_time/dt`

Exercise 4

3. Write a loop that executes `nsteps` time steps.
   - Calculate the second derivative with a 2D version of your subroutine from Exercise 2:

$$\nabla^2 T_{i,j}^n = \frac{T_{i-1,j}^n + T_{i+1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n}{h^2}$$

   - Integrate forward in time:

$$T_{i,j}^{n+1} = T_{i,j}^n + \Delta t \cdot \kappa \cdot \nabla^2 T_{i,j}^n$$

   - Use the boundary conditions `T=0` (at the beginning as well as after each time step).
   - Write the temperature field at the first and last time step into a text file using the `OPEN()` and `WRITE()` statements.

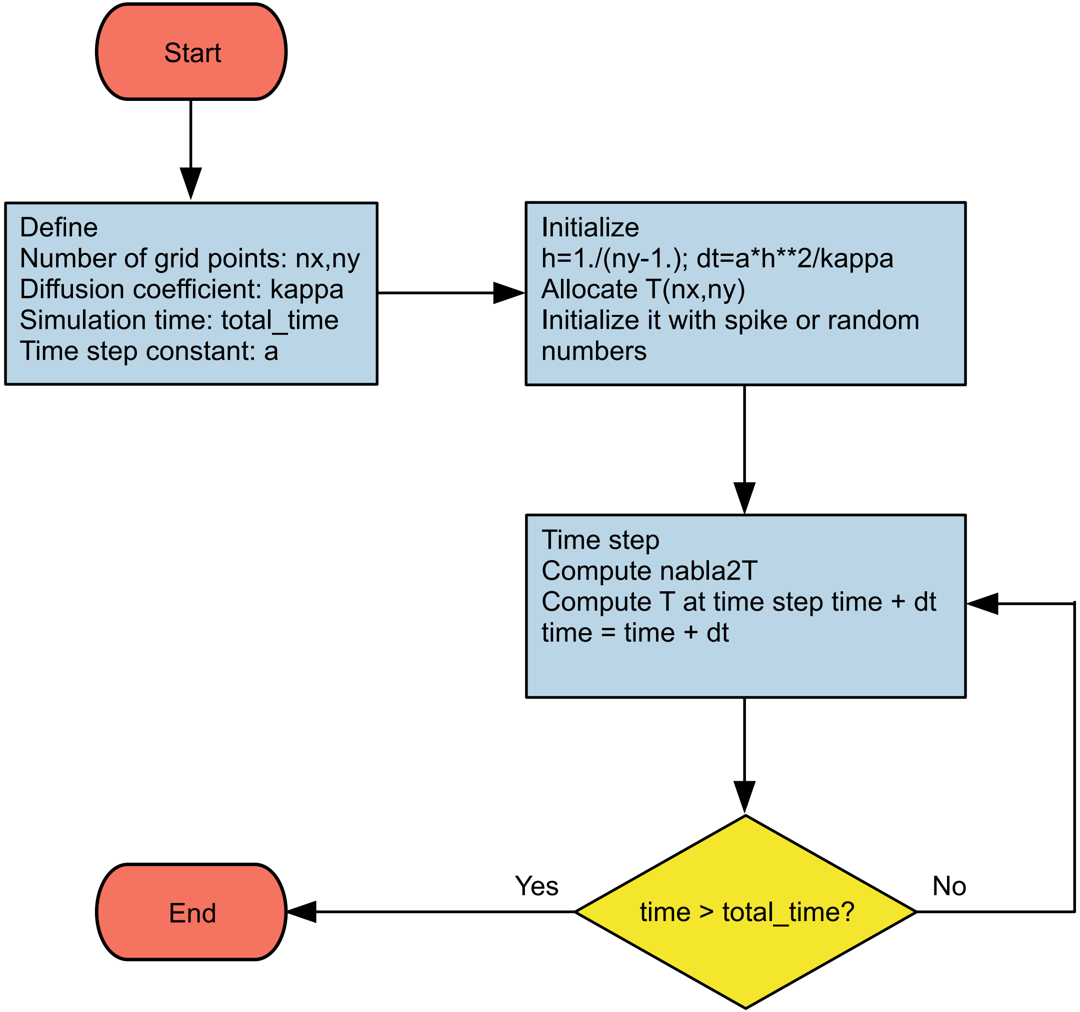
Exercise 4

4. Run your program and test it.

- Use two different initial conditions for the temperature (`random` and `spike`) and vary the number of grid points.
- Determine the critical value for `a` above which the solution becomes unstable.
- Plot the resulting temperature fields with your favorite plotting tool.
- Optional: If you write out multiple time steps you can create an animated gif as follows:

```
micromamba activate /headless/envs/magic
convert -delay 20 *.png animation.gif
```

Flowchart for Exercise 4



Start

Define
Number of grid points: nx,ny
Diffusion coefficient: kappa
Simulation time: total_time
Time step constant: a

Initialize
h=1./(ny-1.); dt=a*h**2/kappa
Allocate T(nx,ny)
Initialize it with spike or random numbers

Time step
Compute nabla2T
Compute T at time step time + dt
time = time + dt

time > total_time?

Yes

No

End

Exercise 4

Deadline:

- Please hand in your solutions (.f90 files and plots of T) no later than Tuesday, **16 April 2024, 23:59**.

Questions?

- Email me (marina.duetsch@univie.ac.at)
  Or pass by my office (UZA II, 2G551).