Stability analysis of the FTCS scheme in 2D

2D diffusion equation:

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

2D ansatz:

$$\epsilon_{j_x,j_y}^{n} = \sum_{m=-M}^{M} A^n \cdot e^{ikj_xh} e^{ikj_yh}$$

T: Temperature
t: Time
x, y: Spatial coordinates
κ: Diffusion coefficient
ϵ: Error
n: Time index
j_x, j_y: Space indices
i: Imaginary unit
k: Wave number ($k = \pi m/L$)
L: Length of the model domain
M: Maximum wave number ($M = L/\Delta x$)
h: Grid spacing

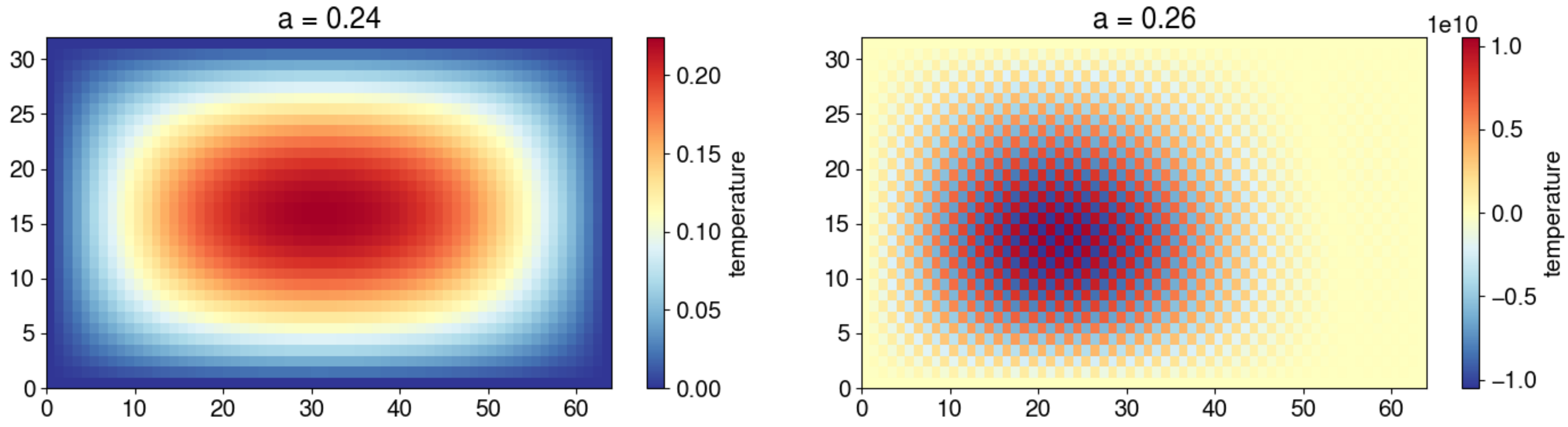
Finite differences:

$$\epsilon_{j_x,j_y}^{n+1} = \epsilon_{j_x,j_y}^{n} + \frac{\Delta t \cdot \kappa}{h^2} \cdot \left(\epsilon_{j_x-1,j_y}^{n} + \epsilon_{j_x+1,j_y}^{n} + \epsilon_{j_x,j_y-1}^{n} + \epsilon_{j_x,j_y+1}^{n} - 4\epsilon_{j_x,j_y}^{n}\right)$$

Stability analysis of the FTCS scheme in 2D

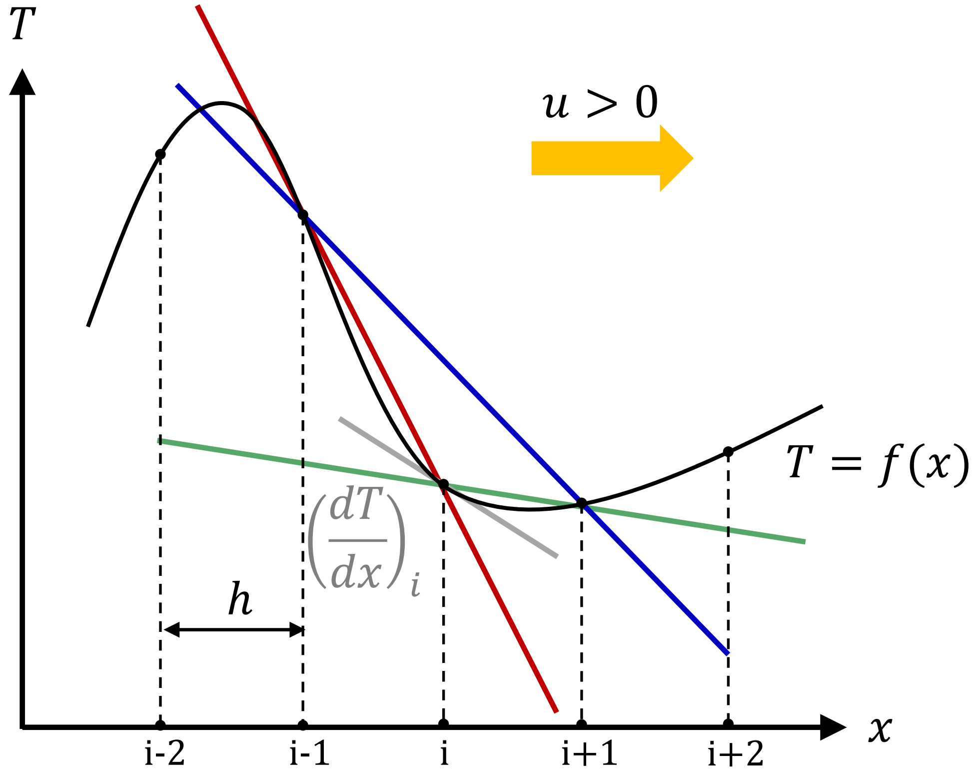
Same procedure as last week results in:

$$A = 1 + \frac{2 \cdot \Delta t \cdot \kappa}{h^2} \cdot (\cos(kh) + \cos(kh) - 2) \leq 1 \quad \rightarrow \quad \frac{\Delta t \cdot \kappa}{h^2} \leq \underbrace{\frac{1}{4}}_{a}$$



Advection

Is difficult to solve numerically



Pure advection

1D $$\frac{\partial T}{\partial t} + u \cdot \frac{\partial T}{\partial x} = 0$$

2D $$\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T = 0$$

upstream: $\left(\frac{dT}{dx}\right)_i \approx \frac{T_i - T_{i-1}}{h}$

downstream: $\left(\frac{dT}{dx}\right)_i \approx \frac{T_{i+1} - T_i}{h}$

centered: $\left(\frac{dT}{dx}\right)_i \approx \frac{T_{i+1} - T_{i-1}}{2h}$

Stability analysis of the centered advection scheme (in 1D)

Finite differences:

$$\frac{\epsilon_j^{n+1} - \epsilon_j^n}{\Delta t} + u\frac{\epsilon_{j+1}^n - \epsilon_{j-1}^n}{2h} = 0 \quad \rightarrow \quad \epsilon_j^{n+1} = \epsilon_j^n - \frac{u\Delta t}{2h}\left(\epsilon_{j+1}^n - \epsilon_{j-1}^n\right)$$

Insert ansatz $\epsilon_j^n = \sum_{m=-M}^{M} A^n \cdot e^{ikjh}$:

$$A = 1 - \frac{u\Delta t}{2h} \cdot \left(e^{ikh} - e^{-ikh}\right) = 1 - \frac{u\Delta t}{2h} \cdot 2i\sin(kh)$$

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

Stability analysis of the centered advection scheme (in 1D)

For stability, we need $|A| \leq 1 \;\rightarrow\; AA^* \leq 1$

$$AA^* = \left(1 - \frac{u\Delta t}{h} \cdot i\sin(kh)\right) \cdot \left(1 + \frac{u\Delta t}{h} \cdot i\sin(kh)\right)$$

$$\underset{\uparrow}{=} 1 + \underbrace{\left(\frac{u\Delta t}{h}\right)^2 \cdot \sin^2(kh)}$$

$\boxed{i^2 = -1}$

always positive → unstable

Stability analysis of the upstream advection scheme (in 1D)

Finite differences:

$$\frac{\epsilon_j^{n+1} - \epsilon_j^n}{\Delta t} + u\frac{\epsilon_j^n - \epsilon_{j-1}^n}{h} = 0 \quad \rightarrow \quad \epsilon_j^{n+1} = \epsilon_j^n - \frac{u\Delta t}{h}\left(\epsilon_j^n - \epsilon_{j-1}^n\right)$$

Insert ansatz $\epsilon_j^n = \sum_{m=-M}^{M} A^n \cdot e^{ikjh}$:

$$A = 1 - \underbrace{\frac{u\Delta t}{h}}_{a} \cdot \left(1 - e^{-ikh}\right)$$

Stability analysis of the upstream advection scheme (in 1D)

For stability, we need $|A| \leq 1$ → $AA^* \leq 1$

$$AA^* = \left(1 - a\left(1 - e^{-ikh}\right)\right) \cdot \left(1 - a\left(1 - e^{ikh}\right)\right)$$

$$= \left(1 - a + ae^{-ikh}\right) \cdot \left(1 - a + ae^{ikh}\right)$$

$$= (1-a)^2 + (1-a)\underbrace{\left(ae^{-ikh} + ae^{ikh}\right)}_{2a\cos(kh)} + a^2 \cancel{e^{ikh-ikh}}$$

$$= 1 - 2a + a^2 + 2a(1-a)\cos(kh) + a^2$$

$$= 1 - 2a(1-a)(1 - \cos(kh))$$

Most dangerous when $\cos(kh) = -1$

Stability analysis of the upstream advection scheme (in 1D)

Condition for stability:

$$1 - 4a(1-a) \leq 1 \quad \rightarrow \quad a(1-a) \geq 0$$

Since a is positive, we can divide by a:

$$1 - a \geq 0 \quad \rightarrow \quad a \leq 1$$

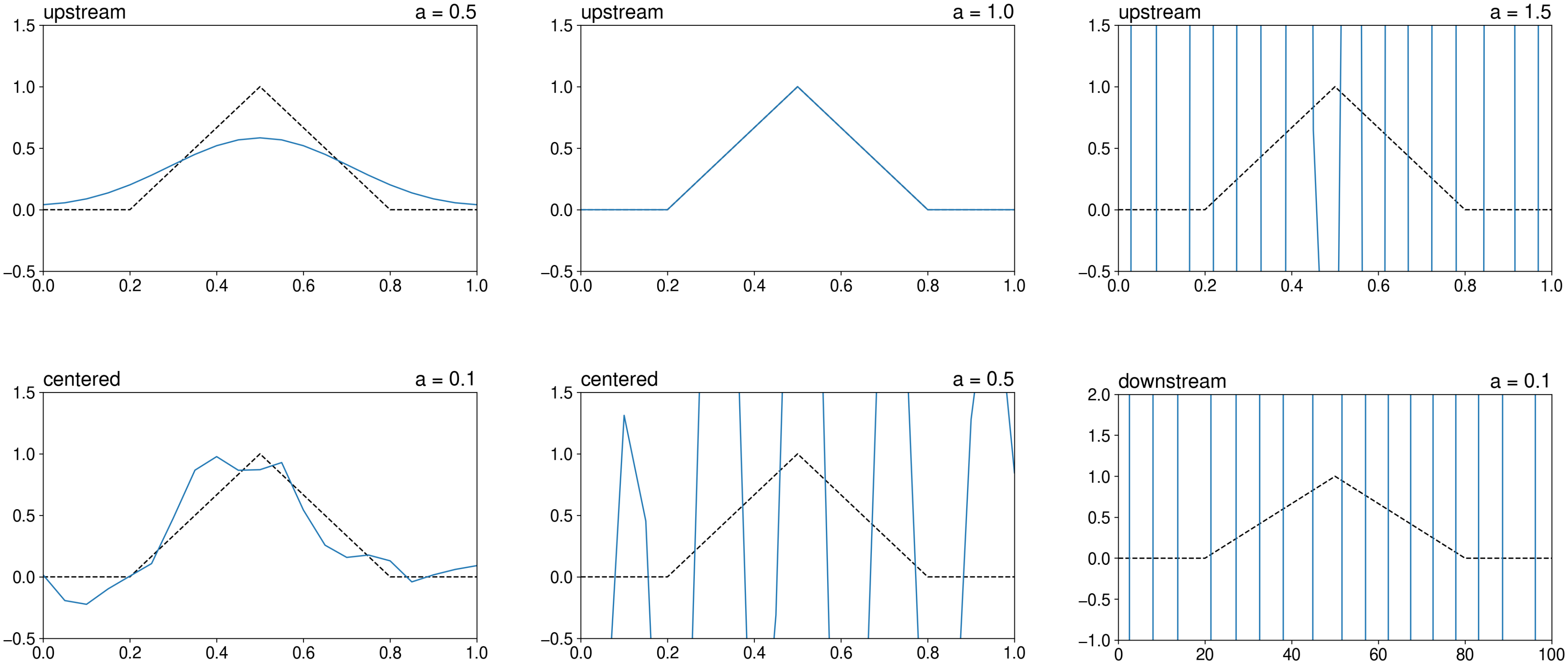
The upstream scheme is stable if: $\boxed{\frac{u\Delta t}{h} \leq 1}$

Courant-Friedrichs-Lewy condition

Triangle anomaly after one round in the model domain

$$a = \frac{u\Delta t}{h}$$

u=1, L=1, h=0.05



Exercise 5

Adapt your Fortran program from Exercise 4 such that it solves the two-dimensional advection-diffusion equation:

$$\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T = \kappa \nabla^2 T \qquad (1)$$

The flow is given by the stream function ψ:

$$\psi = B \cdot \sin\left(\frac{\pi x}{x_{max}}\right) \cdot \sin\left(\frac{\pi y}{y_{max}}\right) \qquad (2)$$

Where B is a constant.

Exercise 5

1. Read in the control parameters from a namelist:
   - Number of grid points in x and y direction `nx`, `ny`
   - Diffusion coefficient `kappa`
   - Integration time `total_time`
   - Time step constants `a_adv` (advection) and `a_diff` (diffusion)
   - Flow constant `B`

2. Initialize the variables
   - Temperature: 2D array of size `(nx,ny)` with random numbers or delta function
   - Stream function: 2D array of size `(nx,ny)` with equation 2
   - Grid spacing `h=1./(ny-1.)`

Exercise 5

3. Compute the velocities in x and y direction from the stream function with centered differences:

$$u_{i,j} = \left(\frac{\partial \psi}{\partial y}\right)_{i,j} \approx \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2h} \qquad v_{i,j} = -\left(\frac{\partial \psi}{\partial x}\right)_{i,j} \approx -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}$$

4. Compute the time step from the maximum velocity in the model domain:
   - Time step: `dt=MIN(a_diff*h**2/kappa, a_adv*h/vmax)`
   - Number of time steps: `nsteps=INT(total_time/dt)`

Exercise 5

5. Perform `nsteps` time steps with the following calculations for all `i=2,nx-1` and `j=2,ny-1`:
   - Calculate $\nabla^2 T_{i,j}^n$ using the subroutine from Exercise 4.
   - Write a function or subroutine that computes $\vec{v}_{i,j} \cdot \nabla T_{i,j}^n$ using the upstream scheme, e.g.:

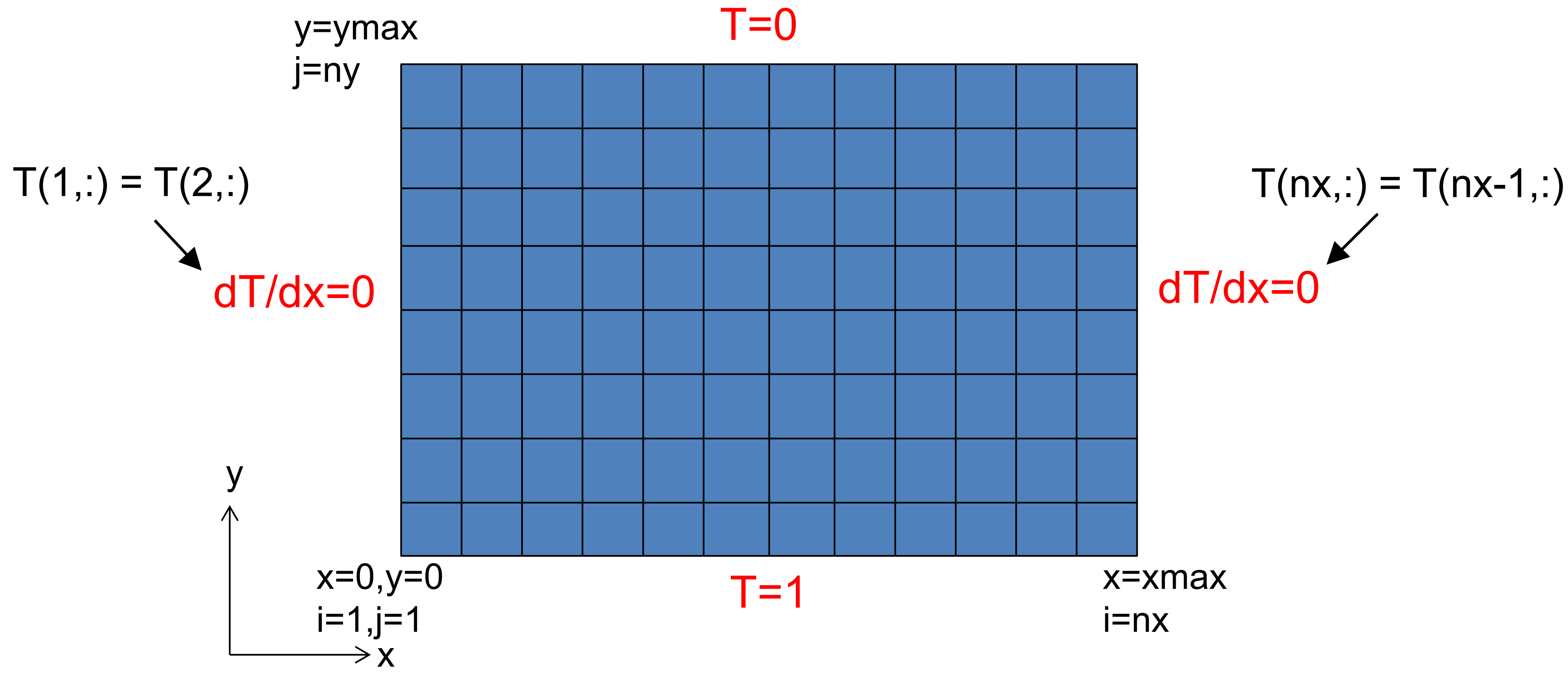
     $$\vec{v}_{i,j} \cdot \nabla T_{i,j}^n \approx u_{i,j} \cdot \frac{T_{i,j}^n - T_{i-1,j}^n}{h} + v_{i,j} \cdot \frac{T_{i,j}^n - T_{i,j-1}^n}{h} \quad \text{if } u_{i,j} > 0 \text{ and } v_{i,j} > 0$$

     (This needs to be adapted based on the wind direction.)
   - Integrate forward in time:

     $$T_{i,j}^{n+1} = T_{i,j}^n + \Delta t \cdot \left(\kappa \cdot \nabla^2 T_{i,j}^n - \vec{v}_{i,j} \cdot \nabla T_{i,j}^n\right)$$

Exercise 5

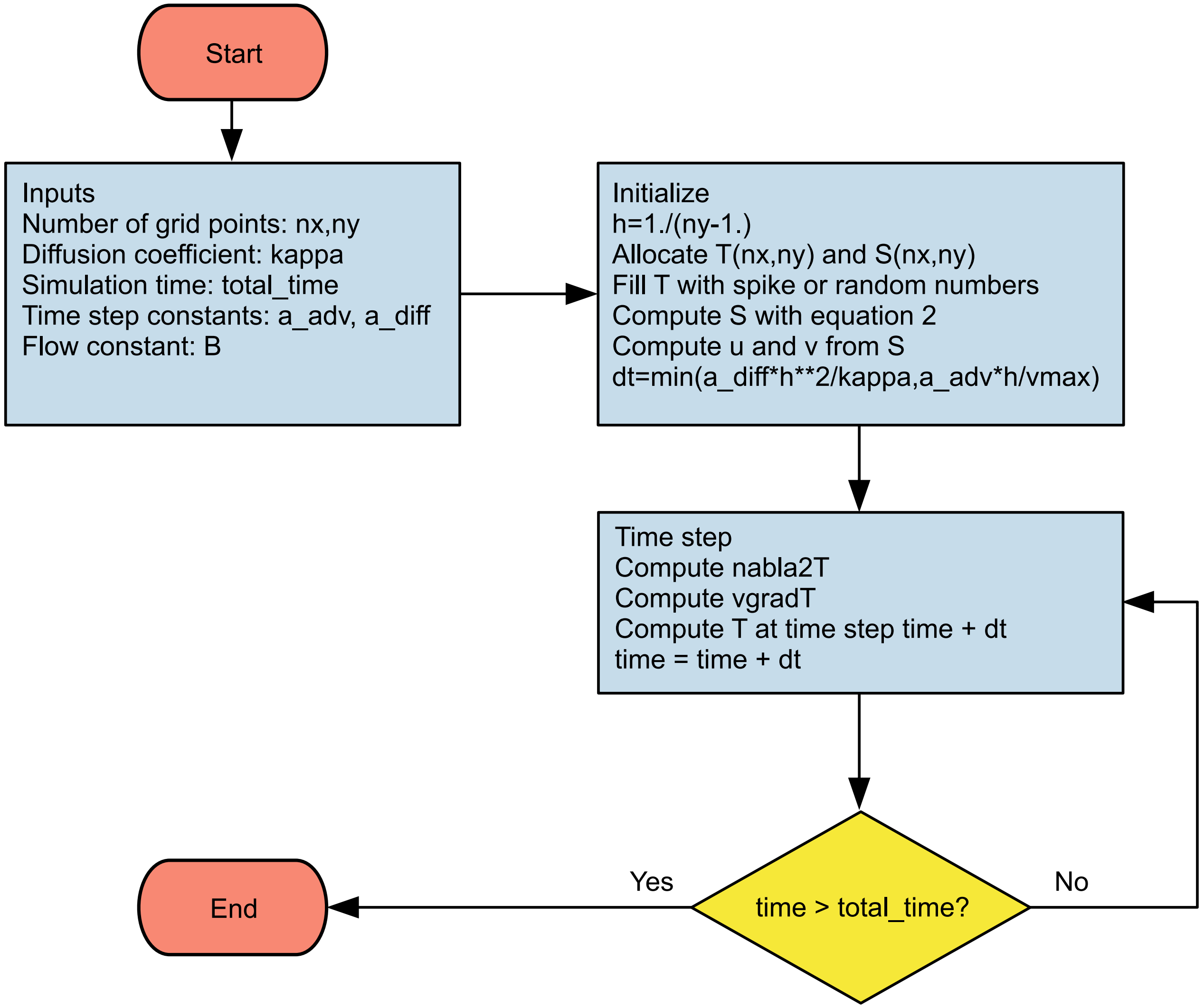
– Boundary conditions:



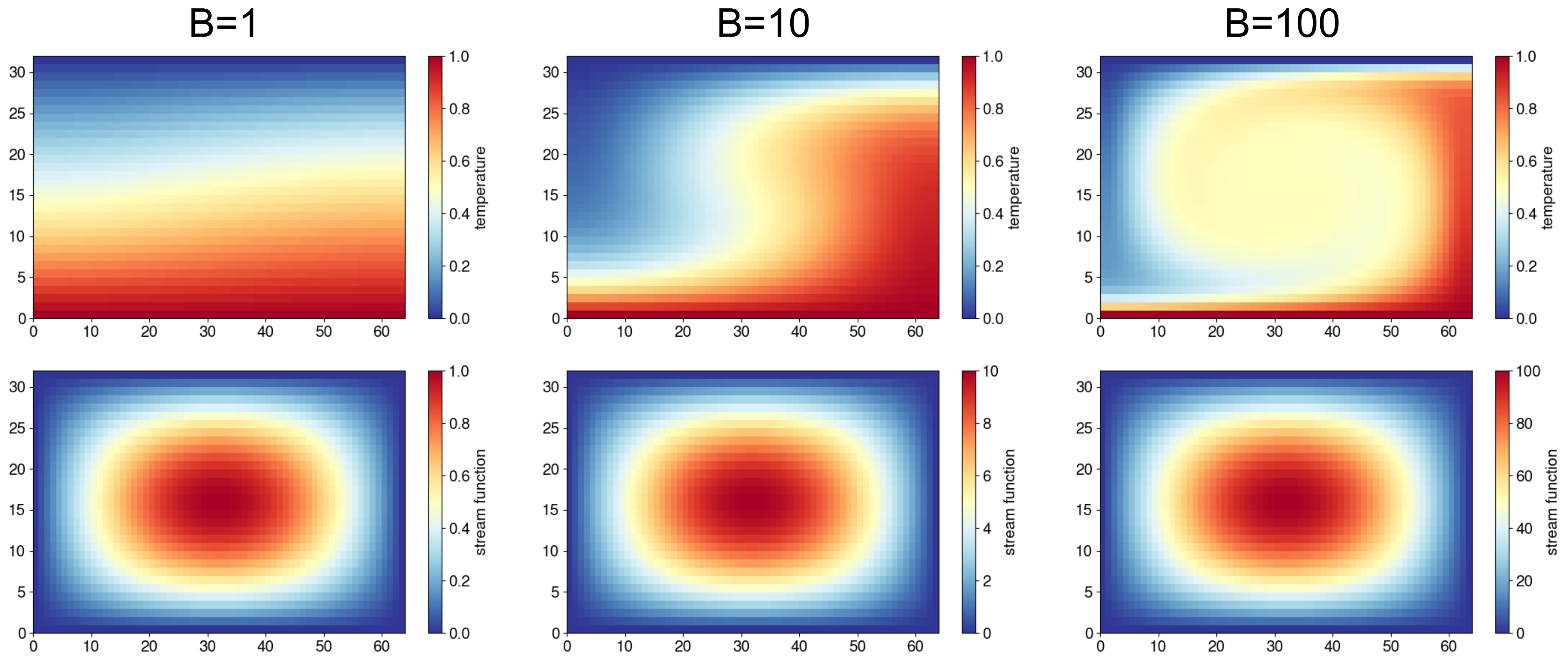
Exercise 5

6. Run your program and test it.
   - Use different values for B (e.g. 1, 10 and 100).
   - After a sufficient integration time, the system should reach a steady state, i.e. the temperature should not change anymore. The steady state is independent of the initial conditions.
   - Save the resulting temperature fields in a text file and plot them using your favorite plotting tool.

Flowchart
for Exercise 5



Resulting temperature and stream function



Exercise 5

Deadline:

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

Questions?

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