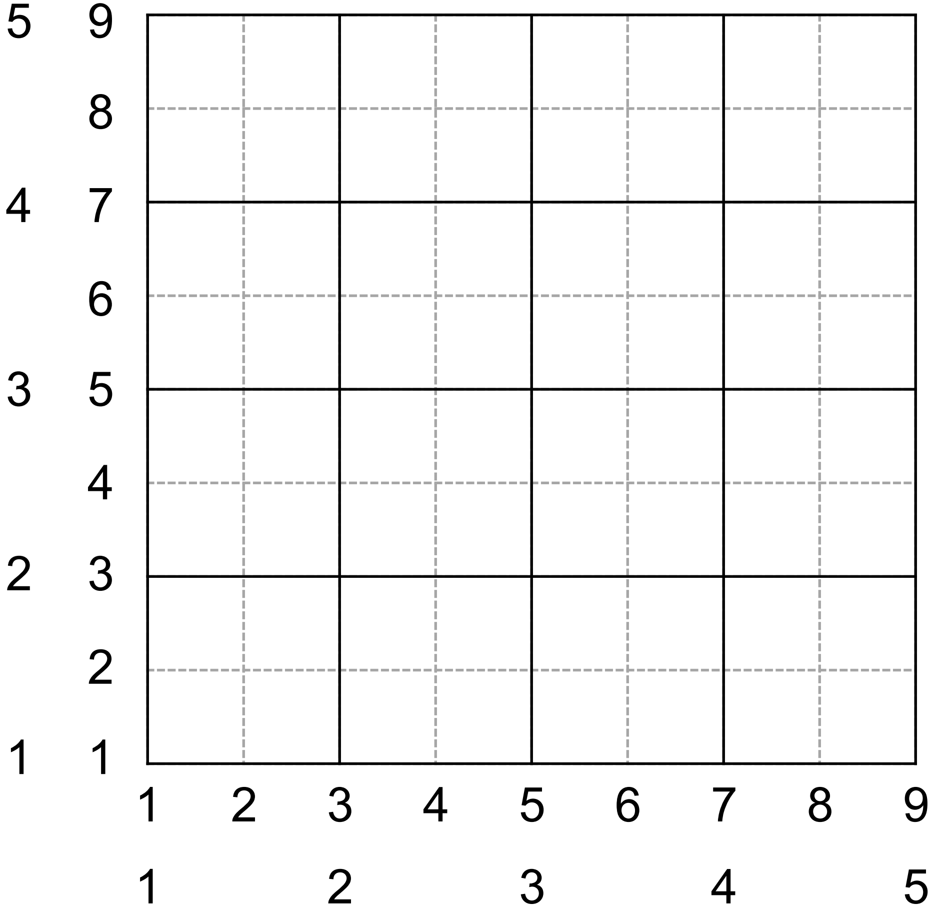


restrict:

```
coarse(i,j) = fine(2*i-1,2*j-1)
```

prolongate:

```
fine(i,j) = (coarse(floor ((i+1)/2.), floor ((j+1)/2.) &
          + coarse(floor ((i+1)/2.), ceiling((j+1)/2.) &
          + coarse(ceiling((i+1)/2.), floor ((j+1)/2.) &
          + coarse(ceiling((i+1)/2.), ceiling((j+1)/2.)) / 4.
```

Conservation equations

- No rotation
- No heat sources

Conservation of momentum (Navier-Stokes equation)

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{v} + \cancel{2\vec{\Omega} \times \vec{v}} + g\beta T \vec{e}_y$$

Conservation of energy (thermodynamic equation)

$$\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T = \kappa \cdot \nabla^2 T + \cancel{Q}$$

Conservation of mass (continuity equation)

$$\nabla \cdot \vec{v} = 0$$

$\vec{v}$:	Velocity vector (u,v)
ρ:	Density
p:	Pressure
ν:	Kinematic viscosity
Ω:	Angular velocity
g:	Gravitational acceleration
β:	Thermal expansion coefficient
T:	Temperature
$\vec{e}_y$:	Unit vector in y direction
κ:	Diffusion coefficient
Q:	Heating rate

Nondimensionalization

Advantages:

- Less parameters
- Easier to recognize the dynamic regime
- Easier to compare systems of different spatial / temporal scales and similar dynamics

$$\tilde{x} = \frac{x}{L} \qquad \tilde{T} = \frac{T}{\Delta T} \qquad \tilde{t} = \frac{t}{L^2/\kappa} \qquad \tilde{v} = \frac{v}{\kappa/L} \qquad \tilde{p} = \frac{p}{\rho\nu\kappa/L^2}$$

ρ: density ν: kinematic viscosity κ: diffusion coefficient L: characteristic length

Navier-Stokes equation

$$\frac{\kappa^2}{L^3}\cdot\left(\frac{\partial\vec{v}}{\partial t}+\vec{v}\cdot\nabla\vec{v}\right)=-\frac{\nu\kappa}{L^3}\nabla p+\ \nu\frac{\kappa/L}{L^2}\nabla^2\vec{v}+\Delta T g\beta T\vec{e}_y \quad \Bigg| \cdot\frac{L^3}{\nu\kappa}$$

$$\underbrace{\frac{\kappa}{\nu}}_{\frac{1}{\mathrm{Pr}}}\cdot\left(\frac{\partial\vec{v}}{\partial t}+\vec{v}\cdot\nabla\vec{v}\right)=-\nabla p+\ \nabla^2\vec{v}+\underbrace{\frac{\Delta T g\beta L^3}{\nu\kappa}}_{\mathrm{Ra}}T\vec{e}_y \quad \Bigg| \ \mathrm{Pr}\to\infty$$

$$-\nabla p+\ \nabla^2\vec{v}=-\mathrm{Ra}\cdot T\cdot\vec{e}_y \xrightarrow{\text{Exercise 6}}$$

Poisson equations

$$\nabla^2\psi=-\omega$$

$$-\nabla^2\omega=\mathrm{Ra}\cdot\frac{\partial T}{\partial x}$$

Pr: Prandtl number Ra: Rayleigh number

Thermodynamic equation

$$\frac{\Delta T}{L^2/\kappa} \cdot \left(\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T\right) = \frac{\Delta T}{L^2} \kappa \cdot \nabla^2 T \qquad \Bigg| \cdot \frac{L^2}{\kappa \Delta T}$$

$$\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T \; = \nabla^2 T$$

Exercise 7

The following dimensionless conservation equations describe convection in a fluid with high viscosity (e.g. Earth's mantle):

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

$$-\nabla^2 \omega = \mathrm{Ra} \cdot \frac{\partial T}{\partial x} \qquad (2)$$

$$\nabla^2 \psi = -\omega \qquad (3)$$

Your task in this exercise is to write a Fortran program that solves these equations. For this purpose, combine the Poisson solver from Exercise 6 with the advection-diffusion program from Exercise 5.

Exercise 7

1. Read in the input variables from a namelist:
   - Number of grid points `nx` and `ny`
   - Integration time `total_time`
   - Time step constants `a_adv` and `a_diff`
   - NEW: Convergence criterion for Poisson solver `max_err`
   - NEW: Rayleigh number `Ra`
   - NEW: Temperature initialization `T_ini_type`: `random` or `cosine`

2. Initialize the variables:
   - Temperature `T` with random numbers or a cosine function.
     Cosine e.g. $T(x) = 0.5 \cdot \left(1 + \cos\left(3\pi \cdot \frac{x}{x_{max}}\right)\right)$
   - Stream function and vorticity `S=W=0` (only at the beginning)
   - Grid spacing `h=1./(ny-1.)`

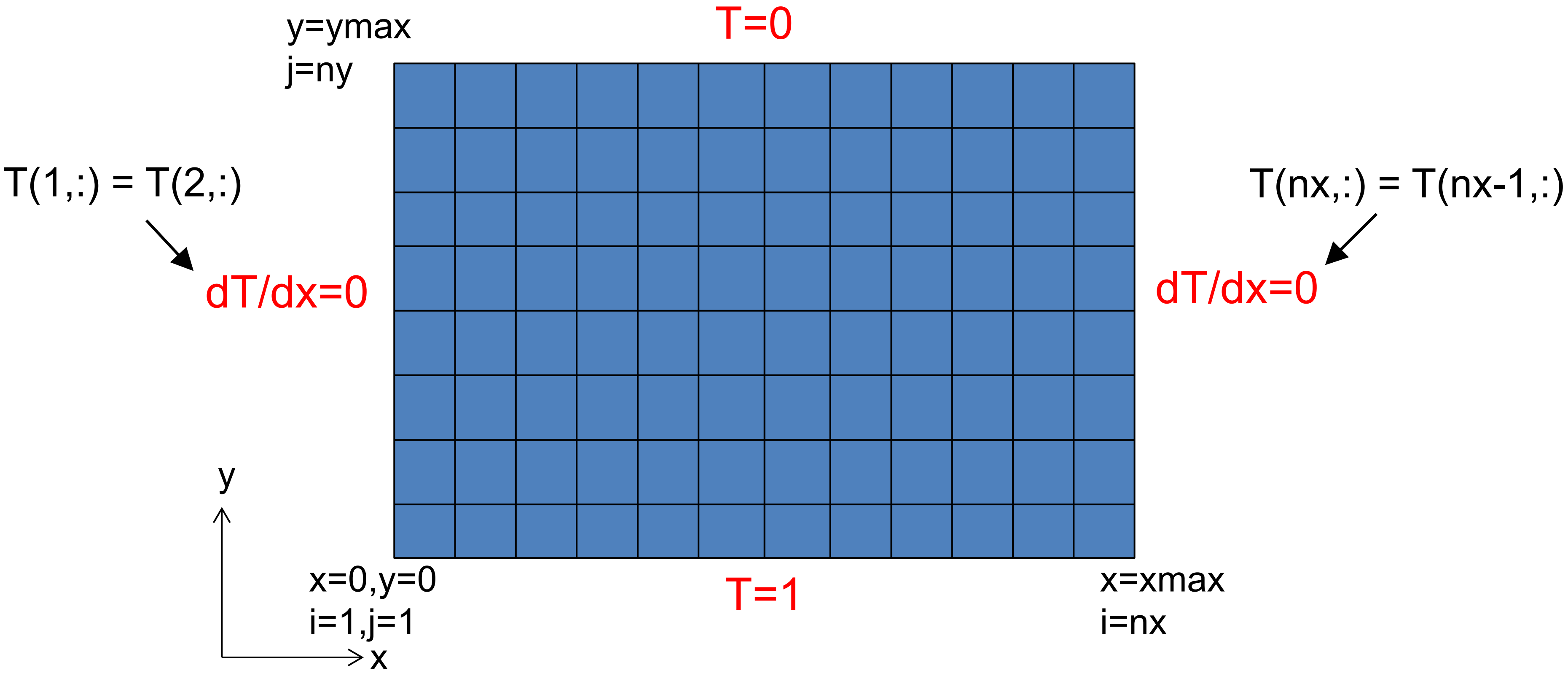
Exercise 7

3. Perform several time steps until the integration time is reached:

- Compute $\mathrm{Ra} \cdot \frac{\partial T}{\partial x}$
- Determine ω from $\mathrm{Ra} \cdot \frac{\partial T}{\partial x}$ using the Poisson solver (Equation 2)
- Determine ψ from ω using the Poisson solver (Equation 3)
- Compute the wind speeds u and v from ψ
- Compute the time step from a_adv, a_diff and the maximum wind speed in the model domain
- Compute $\nabla^2 T$ and $\vec{v} \cdot \nabla T$ using the subroutines from Exercise 5
- Integrate forward in time:
  - $T_{i,j}^{n+1} = T_{i,j}^{n} +$ ... (Equation 1)
  - $t = t + \Delta t$

Exercise 7

Boundary conditions:



Exercise 7

4. Test the program with the following two namelists, and save the output in binary files.

```
&INPUTS
  nx=257
  ny=65
  a_diff=0.23
  a_adv=0.4
  total_time=0.1
  max_err=1.E-3
  Ra=1.E5
  T_ini_type='cosine'
/
```

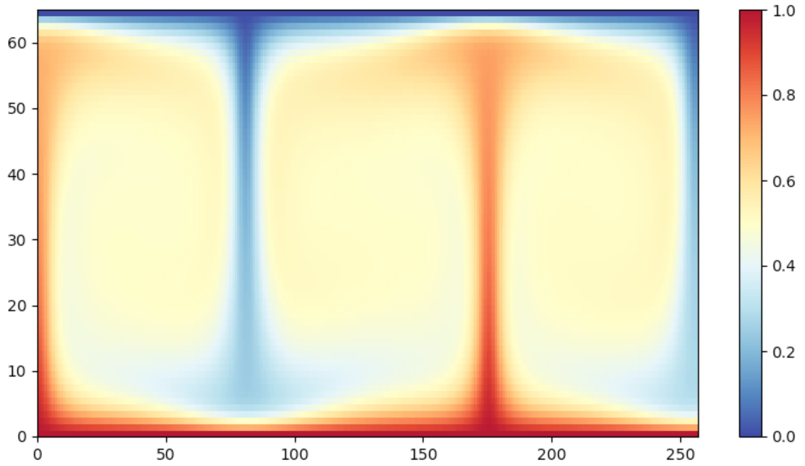
```
&INPUTS
  nx=1025
  ny=65
  a_diff=0.23
  a_adv=0.4
  total_time=0.1
  max_err=1.E-3
  Ra=1.E5
  T_ini_type='random'
/
```

5. Plot the temperature fields with your favorite plotting tool.

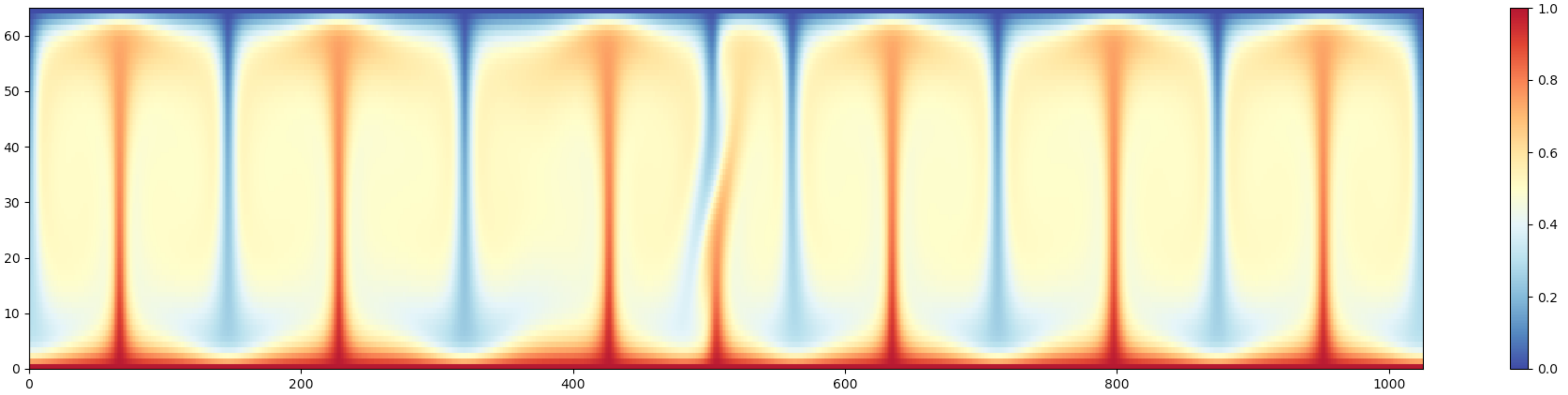
Resulting temperature fields

Should look like this (qualitatively)

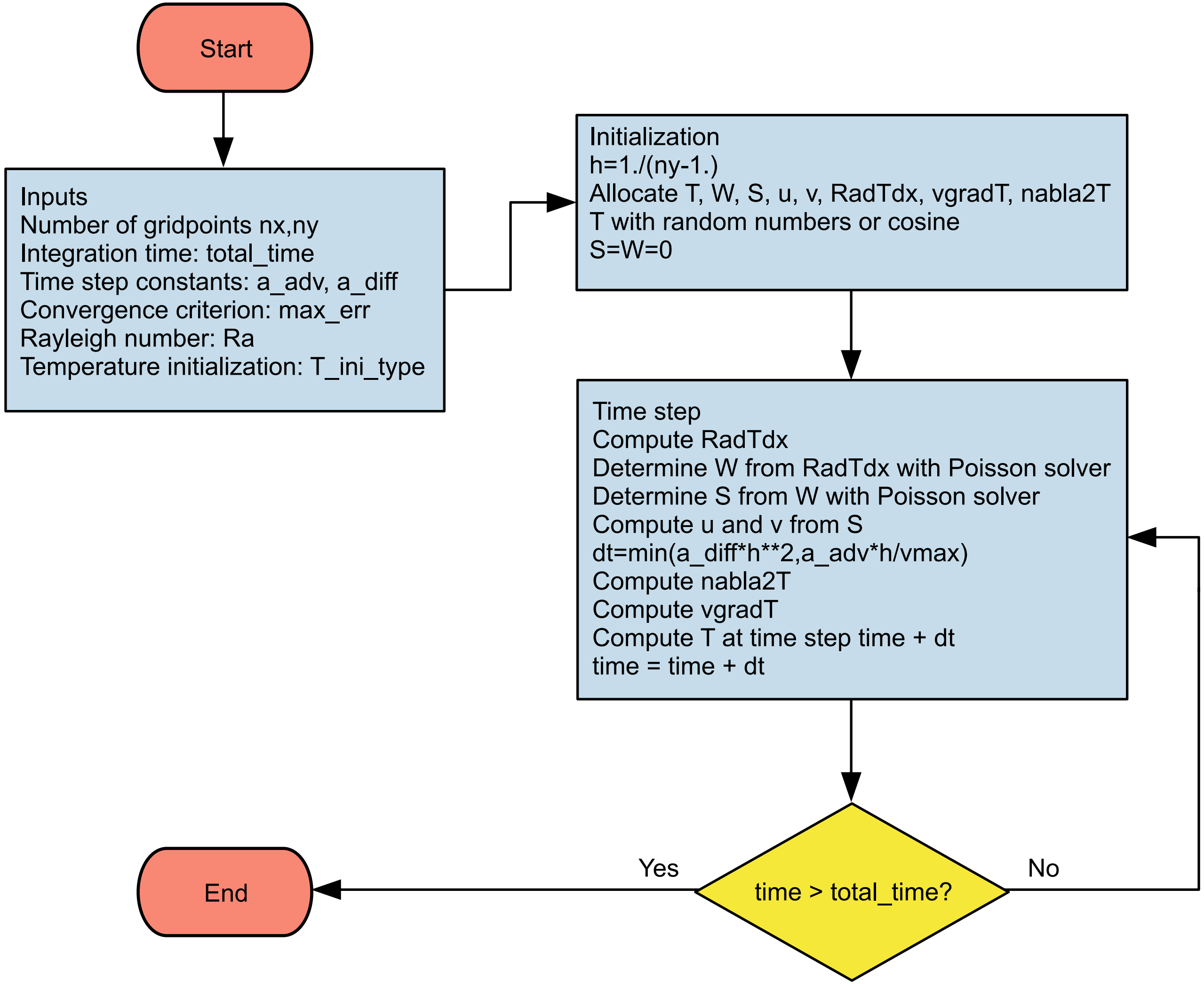
Namelist 1



Namelist 2



Flowchart for Exercise 7



Exercise 7

Deadline:

- Please hand in your solutions (.f90 files and plots T) no later than Tuesday, **14 May 2024, 23:59**.

Questions?

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