Quiz 4 - Computational Physics II

NAME: Alan Palma **SCORE:** 18.3/20

Date: Thursday 29 May 2025 (17h00) **Duration:** 45 minutes

Credits: 20 points (4 questions) **Type of evaluation:** LAB

Provide concise answers to the following items:

1. **(4 points) Partial differential equations (PDEs) in Fourier space**
   (a) Write down the 1D heat equation and the 1D one-way wave equation in Fourier space.
   (b) Explain the difference between diffusion and advection processes.

-0.25

a)

• 1D heat equation

$\mathbb{R}$: $\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial t^2}$

$\downarrow$

$\mathbb{F}$: $\boxed{\frac{d\hat{u}(k,t)}{dt} = -c^2k^2\hat{u}(k,t)}$

• 1D one-way wave equation

$\mathbb{R}$: $\frac{\partial u}{\partial t} \pm c\frac{\partial u}{\partial x} = 0$

$\mathbb{F}$: $\boxed{\frac{d\hat{u}(k,t)}{dt} = \mp ikc\,\hat{u}(k,t)}$

− : moving to the rigth

+ : moving to the left.

b) Diffusion is modeled with parabolic equation and means the process in which a quatity (e.g. temperature, concentration, etc) moves from a higher "concentration" to a lower one, the physical quantity spreads out with time. In the other hand, advection is the process in which a quantity is transported by conserving some of its propetics (e.g. shape (amplitude)). This quantity is transposted as a single component (bulk).

Ok, but refer to (a).

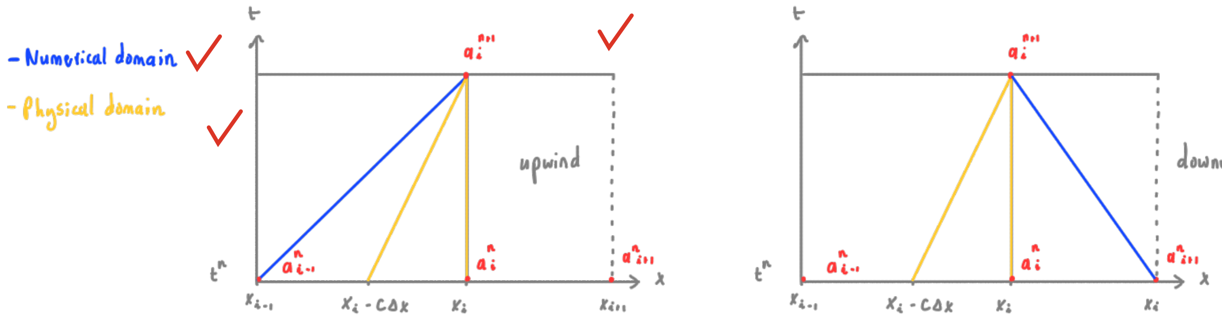
2. **(6 points) Numerical Stability**
   Explain 3 different methods by which we can determine the stability of a numerical scheme.

1) Von Neuman analysis: Analysis of the growth of the solution in our discretitation scheme. Usually it is used a single Fourier mode ($a_i^n = A^n e^{\mp i\theta}$). However, it is important to remark this method works only for linear schemes.

$$\left|\frac{A^{n+1}}{A^n}\right| \leq 1$$

2) Truncation analysis: This consists in kepping the higher order terms of a Taylor serie. In this way, we are able to analize the error leading from them and understand how they modify our target equation.

3) Domain of dependence: This method provides us an analysis of how the solution "propagates" through the initial grid. The numerical domain of dependence shows how the points influence the solution a^{n+1} using finite difference method. The physical domain must be within the numerical domain of dependence to be considered stable.



3. **(5 points) Finite-difference methods for PDEs**
   Write down the 3D Poisson equation and its central-difference approximation including errors.

Poisson equation:

$$\nabla^2 \phi = f$$

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = f$$

central-difference approximation:

$$[\nabla\phi \cdot \hat{x}]_{i+1/2,j,k} = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x}, \quad [\nabla\phi \cdot \hat{y}]_{i,j+1/2,k} = \frac{\phi_{i,j+1,k} - \phi_{i,j,k}}{\Delta y}, \quad [\nabla\phi \cdot \hat{z}]_{i,j,k+1/2} = \frac{\phi_{i,j,k+1} - \phi_{i,j,k}}{\Delta z}$$

$$\Rightarrow [\nabla^2\phi]_{i,j,k} = \frac{[\nabla\phi\cdot\hat{x}]_{i+1/2,j,k} - [\nabla\phi\cdot\hat{x}]_{i-1/2,j,k}}{\Delta x} + \frac{[\nabla\phi\cdot\hat{y}]_{i,j+1/2,k} - [\nabla\phi\cdot\hat{y}]_{i,j-1/2,k}}{\Delta y} + \frac{[\nabla\phi\cdot\hat{z}]_{i,j,k+1/2} - [\nabla\phi\cdot\hat{z}]_{i,j,k-1/2}}{\Delta z}$$

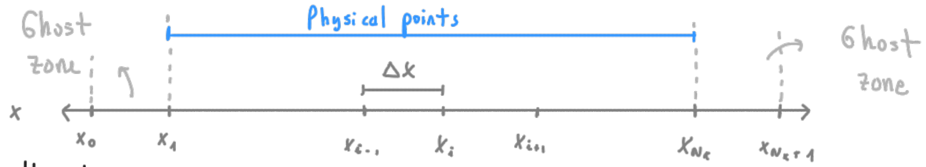
$$+ \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2) + \mathcal{O}(\Delta z)^2 = f_{i,j,k}$$

$$\Rightarrow \boxed{[\nabla^2\phi]_{i,j,k} = \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2} + \frac{\phi_{i,j+1,k} - 2\phi_{i,j,k} + \phi_{i,j-1,k}}{\Delta y^2} + \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta z^2} + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2) + \mathcal{O}(\Delta z^2) = f_{i,j,k}}$$

4. **(5 points) Boundary Conditions for advection problems**
   Consider the advection equation, $u_t + c\,u_x = 0$ (with $c > 0$) on the domain $x \in [0, L]$ with N_x physical grid points plus 2 ghost zones (one at each end). The array `u` has size $N_x + 2$, and the physical points have indices 1 to N_x. The CFL number is defined as $\mathrm{CFL} = c\,\Delta t/\Delta x$. The upwind scheme for interior points is: `u_new[1:Nx+1] = u[1:Nx+1] - CFL * (u[1:Nx+1] - u[0:Nx])`. Provide the Python code lines to set the ghost zone values of `u_new` for each boundary condition type below:

   1. **Periodic boundaries:** The domain wraps around, so $u(0,t) = u(L,t)$.
   2. **Dirichlet boundaries:** The boundaries have fixed values, so $u(0,t) = 0.5$ and $u(L,t) = 0.0$.
   3. **Neumann boundaries:** The boundary gradients are set, so $u_x(0,t) = 0.0$ and $u_x(L,t) = 1.0$.

-1.5



1) At each time iteration the ghost zones should be updated as follow:

```
>> u_new [0] = u [Nx]
>> u_new [Nx+1] = u [1]
```

2) This boundary conditions are defined as follows:

$$u\_new[1] = \frac{1}{2}\left(u[0] + u[1]\right) = \frac{1}{2} \Rightarrow u[0] = 1 - u[1]$$?

$$u\_new[N_x] = \frac{1}{2}\left(u[N_x] + u[N_x+1]\right) = 0.0 \Rightarrow u[N_x+1] = -u[N_x]$$?

```
>> u_new [0] = 1 - u[1]
>> u_new [Nx+1] = - u[Nx]
```

3) The boundary gradients should follow, and it will be updated at each time iteration:

- $u_x(0,t) = 0.0 \Rightarrow \frac{u[1] - u[0]}{\Delta x} = 0.0 \Rightarrow u[0] = u[1]$
- $u_x(L,t) = 1.0 \Rightarrow \frac{u[N_x+1] - u[N_x]}{\Delta x} = 1.0 \Rightarrow u[N_x+1] = \Delta x + u[N_x]$

```
>> u_new [0] = u[1]
>> u_new[Nx+1] = dx + u[Nx]
```

Note: dx = Δx is the spatial step size which must be defined previously.