PH 434 Autumn 2025 – Programming Lab.

Practical Class 8 (Dated: 10. 10. 2025)

Question 1

Write down a function that implements the fourth order Runge-Kutta discussed in the last class.

The $\frac{dy}{dt} = f(t, y)$, where y is some unknown function (scalar or vector), with some initial values, $t = t_0$ and $y(t_0) = y_0$.

Now, for some $h > 0$, the function y at different values of x can be iteratively estimated using,

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\, h, \\ t_{n+1} &= t_n + h, \end{aligned}$$

where $n = 0, 1, 2, 3, \ldots$, and

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}), \\ k_3 &= f(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}), \text{ and} \\ k_4 &= f(t_n + h, y_n + hk_3) \end{aligned}$$

Use the Runge-Kutta method to solve the simple Harmonic oscillator problem and compare it with solve_ivp by plotting t vs y (also plot t vs v_y).

The time span can be from 0 to 10π. Please vary h from 10^{-1} to 10^{-3} to check for accuracy. **You must show this to the TA.**

Question 2

In statistical physics, the **Ising model** describes spins on a lattice that can take values $+1$ (up) or -1 (down).
The energy of a 1D Ising chain with **nearest-neighbor interactions** is given by:

$$E = -J \sum_{i=1}^{N-1} s_i\, s_{i+1} \tag{1}$$

where $s_i \in \{+1, -1\}$ is the spin at site i, $J = 1$ is the interaction strength and N is the number of spins.

1. Write a Python function `ising_energy(spins, J)` that calculates the total energy of a given spin configuration (list of +1 and -1 values).

2. For $N = 6$ spins and $J = 1$:

- Compute the energy of the configuration $[1, 1, 1, 1, 1, 1]$ (all spins aligned).
- Compute the energy of the configuration $[1, -1, 1, -1, 1, -1]$ (alternating spins).

3. Generate all possible configurations of $N = 6$ spins (there are $2^6 = 64$ possibilities).

- Find the configuration(s) corresponding to the **minimum energy** for $J = 1$.

Hint:
You can use Python's `itertools.product` to generate all possible spin configurations. For example, for a chain of length L:

```python
import itertools

# Generate all possible configurations of length L
states = list(itertools.product([-1, 1], repeat=L))

# Each element in `states` is a tuple representing one spin configuration
for state in states:
    print(state)
```

Question 3

Write the matrix that represents the following Hamiltonian:

Consider the transverse Ising model for $N = 8$ spins, given by:

$$H = \sum_{i=1}^{N-1} J\, \sigma_i^x \sigma_{i+1}^x + \sum_{i=1}^{N} h\, \sigma_i^z,$$

where $\sigma_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\sigma_i^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $\mathbb{I}_i$ is the 2×2 identity matrix at site i.

Now,

$$\sigma_i^k \sigma_{i+1}^k = \mathbb{I}_1 \otimes \mathbb{I}_2 \cdots \sigma_i^k \otimes \sigma_{i+1}^k \otimes \cdots \mathbb{I}_N.$$

and

$$\sigma_i^k = \mathbb{I}_1 \otimes \mathbb{I}_2 \cdots \sigma_i^k \otimes \cdots \mathbb{I}_{N-1} \otimes \mathbb{I}_N.$$

1. Find the ground state and energy of the Hamiltonian for $J = 1$ and $h = 1/2$.
2. For $J = 1$ and $h = 0$ compare with the energy and state with the classical problem.

Challenge

Write a function that creates a square wave with peak at $x = 1$ and steps of $t = 1$. Plot the square wave.

Hint: You can repeatedly apply a step function.

Define a function to find the discrete Fourier transform, and plot the dominant frequencies or wave numbers that make up this square wave.

$$X_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp\left(\frac{-2\pi i}{N} kj\right) \times x_j$$

Compare this with the in-built function in **SciPy**:

from scipy.fft **import** fft
yf = **fft**(x)