Some key numerical techniques

In this class we look at some interesting numerical techniques ranging from Runge-Kutta method to solve ODEs to Fast Fourier transformation

1. Runge-Kutta (RK) methods

RK methods are a family of iterative method for solving ordinary differential equations. It's simplest form, the first order RK, is nothing but the Euler method, which we have encountered earlier. In general, it is the fourth-order Runge Kutta method (aka $\mathbf{RK4}$), which is widely used and is often eponymous while discussing these methods.

The RK4 can be described as: $\frac{dy}{dx} = f(x, y)$, where y is some unknown function (scalar or vector) that we want to find, with some initial values, $x = x_0$ and $y(x_0) = y_0$.

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

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

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

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

Note: If $k_i = k_1 = f(x_n, y_n), \forall i$, we get back the first-order RK or the Euler method. Use the RK4 method above to solve the following problem:

- $\frac{dy}{dt} = y\sin(t)^2$, where $y(0) = 1$. Plot the function y for different values of h and also the exact solution for $t \in [0, 2\pi]$. Compare with "solve_ivp" using the RK45 solver.
- $\frac{dx}{dy} = x^2 + y^2$, where $x(1) = 1.2$. Find, x for $y \in [1.0, 10.0]$ with $h = 0.01$. Compare with "solve_ivp" using the LSODA solver.

2. Discrete Fourier transform (DFT)

The discrete Fourier transform (DFT) takes a sequence of complex numbers $\{\mathbf{x}\} := \{x_0, x_1, x_2, \ldots\ldots, x_{N-1}\}$, and transforms it to another set of complex numbers $\{\mathbf{X}\}$, where

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

i) Define a function to calculate the n^{th} root of unity, i.e., $w_n = \exp(\frac{-2\pi i}{n})$, and another to find a matrix $\mathcal{F}$, which performs the above DFT.

ii) Define a function that takes $\{\mathbf{x}\}$, as argument and uses $\mathcal{F}$ to return $\{\mathbf{X}\}$. Transform the cosine function, $\{\mathbf{x}\} := \{x_j = \cos(\frac{2\pi}{n} jl)\}$, where $j = 0, 1, 2, \ldots, n-1$, and $n = 100$ and $l = 5$. Plot both $\{\mathbf{x}\}$ and the transformed function $\{\mathbf{X}\}$ in different plots.

Find the peak value of $\{\mathbf{X}\}$ and note the indices where it occurs.

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

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

Please read about DFT and FFT while solving these.

3. Krylov basis

An example of the **Krylov** basis or subspace is a linear subspace generated by a square matrix H and a vector $|\psi\rangle$, such that the vectors

$$\mathcal{K} = \{|\psi\rangle, H|\psi\rangle, H^2|\psi\rangle, \ldots, H^k|\psi\rangle\},$$

span a k-dimensional subspace.

Without being mathematical rigorous, we note that if n is the dimensiom of H and $|\psi\rangle$, then $k << n$. For larger k, the linear independence of the vectors $\mathcal{K}$ is not ensured.

The Krylov basis is often used in computational quantum many-body physics to find the ground state or low excited state of Hamiltonians of large dimension.

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

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

1. What is the dimension of this Hamiltonian? You can use $J = 1$ and $h = 1/2$.
2. If n is the dimension of H, take $|\psi\rangle$ as a random n-dimensional vector with norm 1.
3. Find the Krylov basis for $k = 20$ and check if they are linearly independent. If not, then start with a different $|\psi\rangle$ or reduce k.
4. Use Gram-Schmidt orthogonalization to create an orthogonal basis from the above Krylov basis.