Part 1. Objectives

- Be able to derive and apply finite difference stencils
- Understand how to use interpolants to numerically compute integrals of functions
- Understand how to derive quadrature rules
- Understand the bisection method

Part 2. Scaling a finite difference stencil

A finite difference stencil was computed that approximates a derivative near a position $x$ with a scaling parameter $h$:

$$f'(x) \approx \alpha f(x - h) + \beta f(x) + \gamma f(x + 2h)$$

For another position $z$, which of the following is a finite difference stencil (with the same order of accuracy) near $z$?

(A) $f'(z) \approx \alpha f(z - h/2) + \beta f(z) + \gamma f(z + h)$

(B) $f'(z) \approx \frac{1}{2}(\alpha f(z - h/2) + \beta f(z) + \gamma f(z + h))$

(C) $f'(z) \approx 2(\alpha f(z - h/2) + \beta f(z) + \gamma f(z + h))$

(D) $f'(z) \approx (\alpha + (z - x))f(z - h/2) + (\beta + (z - x))f(z) + (\gamma + (z - x))f(z + h)$

Part 3. Integration: Error prediction

Suppose we numerically integrate a function $f(x)$ using monomials at five equally-spaced points on an interval of length $h$. The obtained interpolation error is found to be 0.3. What integration error do you predict for the same function on a subinterval of the original one that has length $h/10$?

Part 4. Numerical Integration

You are given nodes and data for an interpolant $\tilde{f}$. Compute its integral over the interval $[-1, 1]$. 

INPUT:
• $x$, a 4-vector of $x$ coordinates
• $f_x$, a 4-vector of function values $f(x)$

OUTPUT:

• integral, the value of $\int_{-1}^{1} \tilde{f}(x)dx$

import numpy as np
import numpy.linalg as la