April 20, 2021 (Due: 08:00 May 11, 2021)

1. Use a linear combination of $f(t)$, $f(t+h)$, $f(t+2h)$ to approximate $f'(t)$ (as accurate as you can). Estimate the truncation error.
2. Use a linear combination of nine function values $f(x+ih, y+jh)$ (for i, $j \in \{-1, 0, 1\}$) to approximate

$$\frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$$

(as accurate as you can). Estimate the truncation error.
3. Use Richardson extrapolation to estimate the derivative of $f(x) = x^3 \mathrm{e}^x$ at $x = 1$. Keep a record for intermediate results. What happens if you iterate for many steps?
4. Use a cubic spline function $s(x)$ to approximate $f(x) = \mathrm{e}^x + \ln x$ over $[1, 4]$. Plot the first and second derivatives, as well as the errors. You are encouraged to try different step sizes and observe the behavior of the error with respect to the step size.
5. In the homework in March, you have been asked to interpolate/fit temperature in human body using cubic spline. Plot the first and second derivatives for interpolating and fitting splines. What can you say about the difference?
6. (optional) An ancient way of computing π can be interpreted in modern terms as $\pi \approx n \sin(\pi/n)$, where n is typically chosen as $n = 3 \cdot 2^k$ for $k \in \mathbb{N}$. Unfortunately, the convergence of such an approximation is very slow, and the calculation is expensive—it involves a lot of square roots (because ancient mathematicians had to use geometry instead of Taylor series to compute $\sin(\pi/n)$). It is believed that Chinese mathematicians, Liu Hui and Zu Chongzhi, discovered a way to largely accelerate the calculation using some sort of extrapolation.

Use Richardson extrapolation to calculate π to seven correct digits after the decimal point, based on the asymptotic expansion

$$n \sin \frac{\pi}{n} = \pi - \frac{\pi^3}{3!\, n^2} + \frac{\pi^5}{5!\, n^4} - \cdots .$$

What is the largest value of n in your calculation?

For simplicity, you may perform calculations such as `sin(pi/n)` directly in your program.