1. (a) In this problem we will establish some numerical differentiation formulas using Taylor series with remainders, such as

\[ f(x \pm h) = f(x) \pm f'(x)h + \frac{1}{2} f''(x)h^2 \pm \frac{1}{6} f'''(\xi_{\pm})h^3, \]

where \( \xi_{\pm} \) are real numbers between \( x \) and \( x \pm h \), respectively. Use this series to show that for \( f \in C^3 \) the three-point midpoint formula for the first derivative satisfies

\[ \frac{f(x + h) - f(x - h)}{2h} = f'(x) + \frac{1}{6} f'''(\xi)h^2, \]

for some \( \xi \) between \( x - h \) and \( x + h \). Hint: Use the Intermediate Value Theorem.

(b) Use a similar method to derive the centered-difference formula for the second-derivative of a function \( f \in C^4 \),

\[ \frac{f(x + h) - 2f(x) + f(x - h)}{h^2} = f''(x) + \frac{1}{12} f^{(4)}(\xi)h^2, \]

with \( \xi \) between \( x - h \) and \( x + h \).

2. (a) Convert the following second-order pendulum equation

\[ y'' = -k \sin(y) \]

to a first-order vector system.

(b) Write an M-file \texttt{pendulum.m} to evaluate the function \( f(t, y, k) \) in part (a) (with \( k \) as a parameter). Use this function and the MATLAB script \texttt{euler.m} to solve the pendulum equation with \( k = 2 \) over time interval \( 0 < t < 7 \) for the initial conditions \( y(0) = 1 \) and \( y'(0) = 0 \), with a number of time-steps \( N = 100, 1000, 10000 \).

(c) Plot the solution orbits in the \((y_1, y_2)\)-plane for \( N = 100, 1000, 10000 \). Does the Euler method appear to converge? Describe the motion of the solution.
3. Using the code `euler.m`, numerically solve the initial-value problem

\[ y' = 2t(1 + y^2), \quad y(t_0) = 1 \]

for \( t_0 = \frac{1}{2}\sqrt{\pi} \) with exact solution \( Y(t) = \tan(t^2) \). Obtain the approximate solution for \( t_0 \leq t \leq t_1 \) with \( t_1 = \pi/3 \), using step sizes \( h = 8^{-(i+1)} \) for \( i = 1, 2, 3 \) in succession. Make two plots at each value of \( h \): (i) the true solution and approximate solution for all of the times, and (ii) the error and relative error for all of the times. Analyze your output and supply written comments on it. Finally, in a third figure (iii) plot the relative errors \( y(t_1)/Y(t_1) - 1 \) at the final time \( t_1 \) versus the index \( i = 1, 2, 3 \), using `semilogy` in Matlab. Does the decrease in error with increasing \( i \) agree with the convergence theory for Euler method? If `relerr` is the 3-vector of relative errors, you can make an estimate of the order of convergence using

\[
\text{PP} = \text{polyfit}(1:3, \log(\text{relerr})/\log(8), 1)
\]

in Matlab, which makes a least-square fit of a straight line to the data for \( \log_8(\text{relerr}(i)) \) versus the index \( i = 1, 2, 3 \).

4. (a) For the problem \( \dot{x} = x, \quad x(0) = 1 \) with exact solution \( X(t) = e^t \), give the explicit solution \( x_n \) for Euler’s approximation to the equation. Use this explicit formula to show that for \( t_n = 1 \), \( X(1) - x_n \approx (h/2)e \) as \( h \to 0 \).

(b) Show that the same error estimate is obtained from the general asymptotic error formula for Euler’s method:

\[ X(t_n) - x_n \approx \delta(t_n)h + O(h^2), \]

where \( \delta(t) \) is the solution of the linear initial-value problem

\[ \dot{\delta}(t) = f_x(t, X(t))\delta(t) + \frac{1}{2} X(t), \quad \delta(t_0) = \delta_0. \]