Problem 1. This problem derives and applies the following \textit{barycentric representation} of the Hermite interpolating polynomial

\[ H_n(x) = (\Psi_n(x))^2 \sum_{i=1}^{n} w_i^2 \left[ \frac{y_i}{(x-x_i)^2} + \frac{y'_i - 2y_i w'_i}{x-x_i} \right] \tag{*} \]

where \( H_n(x) \) is the unique polynomial of degree \( \leq 2n - 1 \) which satisfies \( H_n(x_i) = y_i \) and \( H'_n(x_i) = y'_i \) for distinct points \( x_i, i = 1, \ldots, n \), where \( w_i = 1/\prod_{j \neq i} (x_i - x_j), i = 1, \ldots, n \) are the weights in the barycentric representation of the Lagrange interpolating polynomial, and where \( w'_i = \ell'_i(x_i) \) for the Lagrange polynomial \( \ell_i(x) \), \( i = 1, \ldots, n \).

(a) Use the results of Homework #3, Problem 6 to derive the formula (*).

(b) The coefficient \( w'_i = D_{ii} \) for the Lagrange differentiation matrix \( D_{ij} = \ell'_j(x_i) \). Prove that

\[ \sum_{j=1}^{n} D_{ij} = 0 \]

and use this result to derive a formula for weights \( w'_i \) in terms of the off-diagonal matrix elements \( D_{ij} = \frac{w'_j/w'_i}{x_i - x_j}, j \neq i \). Modify the course code \texttt{barycrtwt.m} to write a new function code \texttt{hermbrarycrtwt.m} which outputs both the weights \( w_i, w'_i, i = 1, \ldots, n \) with distinct points \( x_i, i = 1, \ldots, n \) as inputs, by implementing your formula.

(c) Write a code \texttt{hermbarycrtint.m} which evaluates the Hermite interpolating polynomial using the representation (*), by modifying the course code \texttt{barycrtint.m}. Verify that your code gives the same result for the data \( y_i = \sin(x_i), y'_i = \cos(x_i), \) with \( x_i = 2\pi i/10, i = 0, 1, \ldots, 10 \) when compared with the output of the course code \texttt{hermiteint.m}, which evaluates the Newton form of the Hermite polynomial.

(d) Apply your code from part (c) to evaluate the Hermite interpolating polynomial for the Runge function

\[ f(x) = \frac{1}{1 + 25x^2} \]

and for the points \( x_i = -1 + (i - 1)/n, i = 1, \ldots, 2n + 1 \) with \( n = 20, 30, 40, 50 \). Plot the function \( f(x) \) together with the interpolating polynomials for these values of \( n \). Based on these results, do you expect that the \textit{Runge phenomenon} occurs for Hermite interpolating polynomials, despite matching both function values and derivatives?
Problem 2. This problem considers the triple recurrence relations for orthogonal polynomials and their important applications for Gaussian quadrature.

(a) If \( p_n(x), n = 0, 1, 2, \ldots \) are the monic orthogonal polynomials for the weight function \( w(x) \) on the interval \([a, b]\), then prove that these polynomials satisfy the recurrence relation

\[
p_{n+1}(x) = (x - a_n)p_n(x) - b_n p_{n-1}(x), \quad n = 1, 2, \ldots
\]

where we use the inner product \( \langle f, g \rangle = \int_a^b f(x)g(x)w(x)\,dx \).

Hint: Expand \( q_n(x) = p_{n+1}(x) - xp_n(x) \) in the basis of orthogonal polynomials.

(b) As an example, consider the generalized Laguerre polynomials \( L_{\alpha}^{(a)}(x) \) for the weight function \( w(x) = x^\alpha e^{-x} \) on the interval \((0, \infty)\), which are given by the recurrence relation

\[
(n + 1)L_{\alpha}^{(a)}(x) = (2n + \alpha + 1 - x)L_{\alpha}^{(a)}(x) - (n + \alpha)L_{\alpha}^{(a-1)}(x),
\]

starting with \( L_{\alpha}^{(a)}(x) = 1 \) and \( L_{\alpha}^{(a)}(x) = 1 + \alpha - x \). Show by induction that these polynomials are not monic but instead have the form

\[
L_{\alpha}^{(a)}(x) = (-1)^n x^n + \text{terms of degree } < n.
\]

Derive the triple recurrence relation for the monic polynomials \( \tilde{L}_{\alpha}^{(a)}(x) := (-1)^n n! L_{\alpha}^{(a)}(x) \) and find the coefficients \( a_n, b_n \) for all \( n \).

(c) The Golub-Welsch algorithm for Gaussian quadrature calculates the nodes \( x_i, i = 1, \ldots, n \) as the eigenvalues of the symmetric, tridiagonal "Jacobi matrix"

\[
J = \begin{pmatrix}
    a_0 & \sqrt{b_1} & 0 & \cdots & \cdots & 0 & 0 \\
    \sqrt{b_1} & a_1 & \sqrt{b_2} & 0 & \cdots & \cdots & 0 \\
    0 & \sqrt{b_2} & a_2 & \ddots & \ddots & \ddots & \vdots \\
    \vdots & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\
    \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\
    0 & 0 & \cdots & 0 & \sqrt{b_{n-2}} & a_{n-2} & \sqrt{b_{n-1}} \\
    0 & 0 & \cdots & 0 & \sqrt{b_{n-1}} & a_{n-1}
\end{pmatrix}
\]

and the weights by the formulas \( w_i = \mu_0 |e_1^{(i)}|^2 \), where \( e_1^{(i)}, i = 1, \ldots, n \) are the orthonormal eigenvectors and where \( \mu_0 = \int_a^b w(x)\,dx \).
Using the results of part (b) and Gollub-Welsch algorithm, write a code `glaugquad.m` to implement Gauss-Laguerre quadrature for integrals of form $I = \int_{0}^{\infty} f(x)x^{\alpha}e^{-x}dx$. You can modify the course code `gausseg.m`, which implements Gauss-Legendre quadrature by using Matlab’s `eig` function (based on the QR algorithm) to solve for the eigenvalues and eigenfunctions of the Jacobi matrix.

(d) Use the code in part (c) to approximate the integrals given by Euler’s Gamma function

$$I = \int_{0}^{\infty} x^{m+\alpha}e^{-x}dx = \Gamma(m + \alpha + 1)$$

for the values $\alpha = 1/3$ and $m = 1, 2, \ldots, 10$, by applying Gauss-Laguerre quadrature for $n = 1, 2, \ldots, 6$. Calculate the relative error for each value of $m, n$ and explain your observations.

(e) In addition, approximate the integral

$$I = \int_{0}^{\infty} x^{\alpha}e^{-x}\text{sech}(x)dx$$

for $\alpha = 1/3$ using Gauss-Laguerre quadrature for $n = 10, 20, \ldots, 60$. Compare with the result of composite Simpson from the course code `simc.m` using the interval $[a, b] = [0, 20]$ and also $n = 10, 20, \ldots, 60$. To calculate the relative errors in both methods, assume that the result from the integral function in Matlab is accurate to double precision. How much more accurate is Gauss-Laguerre quadrature here? Explain why composite Simpson converges so slowly in this example.
Problem 3. For each of the following 4-step methods

\[
(i) \quad y_{n+1} = -\frac{1}{6}y_n - \frac{5}{14}y_{n-1} + \frac{3}{2}y_{n-2} + \frac{1}{42}y_{n-3} + h\left(\frac{9}{7}y_{n+1} + \frac{82}{21}y_n + \frac{38}{21}y_{n-1} + \frac{2}{7}y_{n-2} - \frac{2}{21}y_{n-3}\right).
\]

\[
(ii) \quad y_{n+1} = \frac{32}{21}y_{n-1} - \frac{1}{31}y_{n-3} + h\left(\frac{10}{31}y_{n+1} + \frac{128}{93}y_n + \frac{8}{31}y_{n-1} - \frac{2}{93}y_{n-3}\right).
\]

\[
(iii) \quad y_{n+1} = \frac{41}{54}y_n - \frac{1}{4}y_{n-1} + \frac{1}{2}y_{n-2} - \frac{1}{108}y_{n-3} + h\left(\frac{16}{9}y_{n+1} - \frac{1}{4}y_n + \frac{7}{36}y_{n-1}\right).
\]

answer the following questions:

(a) Is the method explicit or implicit?

(b) Is the method consistent?

(c) Find the characteristic polynomial \(\rho(r) = r^{p+1} - \sum_{j=0}^{p} a_j r^{p-j}\) of the method and its roots \(r_0, r_1, \ldots, r_p\). You may use the Matlab function \texttt{roots} if you cannot obtain the roots algebraically.

(d) Is the method weakly stable? Convergent?

(e) If the method is convergent, what is the order of convergence and the leading-order truncation error \(T_n(y)\)?

(f) If the method is weakly stable, is it also relatively stable?
Problem 4. This problem investigates the concept of absolute stability of numerical integration schemes and its importance in solving stiff ODE's.

(a) Consider the application of the Heun method

\[ y_{n+1} = y_n + \frac{h}{2} \left[ f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n)) \right] \]

to solve the model problem

\[ \dot{y} = \lambda y, \quad y(0) = 1, \quad \lambda \in \mathbb{C}. \]

Note that when \( \text{Re}(\lambda) < 0 \), then the exact solution \( Y(t) = e^{\lambda t} \to 0 \) as \( t \to \infty \).

Show that the Heun approximation for this problem takes the form

\[ y_n = [r_0(z)]^n, \]

for \( r_0(z) = 1 + z + \frac{1}{2}z^2 \) and \( z = h\lambda \in \mathbb{C} \). Show, therefore, that there are some \( z \) with \( \text{Re}(z) < 0 \) for which \( y_n \to \infty \) as \( n \to \infty \)!

(b) If a \( p \)-step method has \( |r_i(z)| < 1, \ i = 1, \ldots, p \), then \( y_n \to 0 \) as \( n \to \infty \) and the method is called absolutely stable for that \( z \in \mathbb{C} \). If the method is absolutely stable for all \( \text{Re}(z) < 0 \), then it is called \( A \)-stable. Show that the 1-step trapezoidal method

\[ y_{n+1} = y_n + \frac{h}{2} \left[ f(t_n, y_n) + f(t_{n+1}, y_{n+1}) \right] \]

has

\[ r_0(z) = \frac{1 + z/2}{1 - z/2} \]

and is \( A \)-stable.

Hint: Use the fact that \( |1 + w|^2 = 1 + 2\text{Re}(w) + |w|^2 \).

(c) ODE's are described as stiff when they possess an invariant "slow manifold" on which solutions evolve smoothly and slowly, but nearby solutions either decay extremely rapidly to the slow manifold or else oscillate extremely rapidly around it.

A very simple example of a stiff ODE is

\[ \dot{y} = \frac{1}{\epsilon}(y - y^2), \quad 0 < \epsilon \ll 1 \]

in which the slow manifold is just the constant solution \( y_*(t) = 1 \). Explain why any solution obtained by a small perturbation \( y(t) = 1 + \delta(t) \) from \( y_*(t) = 1 \) will evolve approximately as

\[ \delta(t) = -\frac{2}{\epsilon} \delta(t) \]

and thus \( \delta(t) = \delta(0) \exp(-2t/\epsilon) \) should decay rapidly to 0, so that \( y(t) \to y_*(t) = 1 \).
(d) Unfortunately, numerical integration schemes which are not A-stable can convert the rapid decay of perturbations in (*) instead to rapid growth, unless the time-step \( h \) is chosen so small that \( |r_0(z)| < 1 \) for \( z = h\lambda \). Apply the Heun method with the course code heun.m to the nonlinear ODE in (c) over the time interval \( 0 < t < 1 \) for initial condition \( y(0) = 1/\sqrt{1+\epsilon} \), with exact solution

\[
y(t) = \frac{1}{\sqrt{1+ce^{-2t/\epsilon}}},
\]

for \( \epsilon = 10^{-4} \), using the number of steps \( N = 9900, 9990, \) and \( 10000 \). Compare each of these three Heun approximations with the approximation from the trapezoidal method using only \( N = 10 \), plotting the two solutions together and plotting also their errors together. Use the course code trapezoid.m which implements the trapezoidal method with Newton iteration.

(e) Using the results of part (a) and (b), explain why \( N \geq 10000 \) is required in the Heun method to ensure absolute stability for this problem, whereas trapezoidal method is absolutely stable for any \( N \).