Assignment 5

1. 2. 3. Problem 8.1, 8.3, 8.4; due April 5. The next problem due April 12

4. Denote by \( J_0(z) \) the Bessel \( J \) function of order 0; in Matlab it is specified by \( \text{nu}=0; \)
\( \text{b} = \text{besselj(\text{nu},z)} \). Let
\[
    u_j(z) = c_j \cos(a_j z + b_j),
\]
and let \( a = [a_1, a_2, \ldots, a_m]' \) be a column vector. Define the column vectors \( b \) and \( c \) the same way. The purpose of this exercise is to approximate \( J_0(z) \) in \( [\alpha, \beta] = [0, 2\pi] \) by \( u_j(z) \) for \( j = 1 : m \) by solving the so-called (nonlinear) least-squares problem formulated as
\[
    \min F(a, b, c) = \int_{\alpha}^{\beta} \left( J_0(z) - \sum_{j=1}^{m} u_j(z) \right)^2 dz,
\]
which is an unconstrained nonlinear optimization problem. Here we’ll consider only the cases of \( m = 1 \) and \( m = 2 \). We’ll also replace the integral by a Gaussian quadrature of \( n = 10 \) nodes \( \{z_i\} \) and weights \( \{w_i\} \), scaled and translated to \([\alpha, \beta]\), so that the optimization problem (2) becomes
\[
    \min f(a, b, c) = \sum_{i=1}^{n} w_i \left( J_0(z_i) - \sum_{j} u_j(z_i) \right)^2,
\]

**Remark.** See Problem 4 of HW4 for the calculation of the roots \( \{x_i\} \) of Legendre polynomial, then \( z_i = x_i \cdot (\beta - \alpha)/2 + (\beta + \alpha)/2 \) are the Gaussian nodes. The weights are given by the formula \( w_i = 2/(1 - x_i^2) [p_n'(x_i)]^2 \); see Abramowitz and Stegun, page 887, section 25.4.29, where \( p \) is the Legendre polynomial of degree \( n \) and with \( p_n(1) = 1 \) — the one we used in Problem 4 of HW4.

a. Plot the function \( J_0(z) \) in \([\alpha, \beta]\).

b. For \( m = 1 \), only one function \( u_1(z) = c_1 \cos(a_1 z + b_1) \) is used to approximate \( J_0(z) \) in \([\alpha, \beta]\). Find \( a_1, b_1, c_1 \) to solve the optimization problem (3) by finding the zeros of the gradient of \( f \). Namely, solve the three equations \( \nabla f(a_1, b_1, c_1) = 0 \) by Newton’s iteration (denote a zero by \( (\tilde{a}_1, \tilde{b}_1, \tilde{c}_1) \), a point in 3-D)

c. Provide the residual \( f \) and the gradient \( \nabla f \) at \( (\tilde{a}_1, \tilde{b}_1, \tilde{c}_1) \). Plot the error function
\[
    e_1(z) = J_0(z) - u_1(z).
\]

d. How do you make sure that the zero you find corresponds the global minimum of the error function.

e. Extra. Fix \( (a_1, b_1, c_1) = (\tilde{a}_1, \tilde{b}_1, \tilde{c}_1) \) and repeat steps (b), (c) and (d) with \( J_0(z) \) replaced by \( e(z) \), \( u_1(z) \) replaced by \( u_2(z) \), and \( e_1 \) replaced by
\[
    e_2(z) = J_0(z) - u_1(z) - u_2(z).
\]

**Hint.** For each optimization problem, it is important to find a good initial guess to start the Newton’s iteration, particularly for \( a_1 \), and \( a_2 \) if you do the last part (e). Experiment by looking at the plots of \( J_0(z) \) v.s. that of \( u_1(z) \), and the plots of the error function (4).