Homework 2 - BIOMATH 205

SIMON LEE

October 2023

1 Chapter 5

Q5: Find by hand the Cholesky decomposition of the matrix

$$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$$

Answer:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

To solve L_{11}

$$L_{11} = \sqrt{a_{11}} = \sqrt{2}$$

To solve L_{21}

$$L_{21} = \frac{a_{21}}{L_{11}} = \frac{-2}{\sqrt{2}}$$

To solve L_{22}

$$L_{22} = \sqrt{a_{22} - L_{21} l_{21}}$$

$$L_{22} = \sqrt{5 - \frac{-2}{\sqrt{2}} \frac{-2}{\sqrt{2}}}$$

$$L_{22} = \sqrt{5 - -\frac{\sqrt{2}\sqrt{2}}{\sqrt{2}} - \frac{\sqrt{2}\sqrt{2}}{\sqrt{2}}}$$

$$L_{22} = \sqrt{5 - \sqrt{2}\sqrt{2}}$$

$$L_{22} = \sqrt{5 - 2}$$

$$L_{22} = \sqrt{3}$$

You can find L and L^T by plugging it into the following equation

$$L = \begin{bmatrix} \sqrt{2} & 0 \\ \frac{-2}{\sqrt{2}} & \sqrt{3} \end{bmatrix}$$

$$L^T = \begin{bmatrix} \sqrt{2} & \frac{-2}{\sqrt{2}} \\ 0 & \sqrt{3} \end{bmatrix}$$

azsx **Q13:** Find the QR Decomposition of the matrix

$$X = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 3 & 1 \\ 1 & 1 & 5 \\ 1 & 1 & 3 \end{bmatrix}$$

Answer:
To solve μ_1

$$\mu_1 = a_1\textbf{(1,1,1,1)}$$

To solve e_1

$$e_1 = \frac{\mu_1}{||\mu_1||} = \frac{1}{\sqrt{4}}(1,1,1,1)$$

$$e_1 = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$$

To solve μ_2

$$\mu_2 = a_2 - (a_2 \cdot e_1)e_1, a_2 = (3,3,1,1)$$

$$\mu_2 = (3,3,1,1) - 4(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$$

$$\mu_2 = (3,3,1,1) - (2,2,2,2)$$

$$\mu_2 = \textbf{(1,1,-1,-1)}$$

To solve e_2

$$e_2 = \frac{\mu_2}{||\mu_2||} = \frac{1}{\sqrt{4}} = \frac{1}{2}(1,1,-1,-1)$$

$$e_2 = (\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})$$

To solve μ_3

$$\mu_3 = a_3 - (a_3 \cdot e_1)e_1 - (a_3 \cdot e_2)e_2$$

$$\mu_3 = (3,1,5,3) - 6(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) - (-2)(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})$$

$$\mu_3 = (3,1,5,3) - (3,3,3,3) + (1,1,-1,-1)$$

$$\mu_3 = (1-1,1,-1)$$

To solve e_3

$$e_3 = \frac{\mu_3}{||\mu_3||} = \frac{1}{\sqrt{4}} = \frac{1}{2}(1, -1, 1, -1)$$

$$e_3 = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2})$$

We can find Q using the following formula

$$Q = [e_1^T | e_2^T | e_3^T]$$

$$Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

Using the formula

$$R = \begin{bmatrix} a_1 \cdot e_1 & a_2 \cdot e_1 & a_3 \cdot e_1 \\ 0 & a_2 \cdot e_2 & a_3 \cdot e_2 \\ 0 & 0 & a_3 \cdot e_3 \end{bmatrix}$$

And what we know

$$a_1 = (1, 1, 1, 1)$$

$$a_2 = (3, 3, 1, 1)$$

$$a_3 = (3, 1, 5, 3)$$

We can find R

$$R = \begin{bmatrix} 2 & 4 & 6 \\ 0 & 2 & -2 \\ 0 & 0 & 2 \end{bmatrix}$$

2 Chapter 6

Q1: On a one-dimensional problem of your choice, implement Newton's method. Check your result using the fzero function of the Julia package Roots.jl.

```
import Pkg; Pkg.add("Roots")
using Roots

# Define the polynomial function
function func(coefficients, x)
    result = zero(x)
    for (i, coeff) in enumerate(coefficients)
        result += coeff * x^(i-1)
    end
    return result
end
```

```
# Define the derivative of the polynomial function
function func_derivative(coefficients, x)
    result = zero(x)
    for (i, coeff) in enumerate(coefficients[2:end])
        result += i * coeff * x^(i-1)
    end
    return result
end

# Implement Newton's method for any polynomial function
function newton_method(coefficients, initial_guess; tolerance=1e-5, max_iterations=100)
    x = initial_guess
    for _ in 1:max_iterations
        x_next = x - func(coefficients, x) / func_derivative(coefficients, x)
        if abs(x_next - x) < tolerance
            break
        end
        x = x_next
    end
    return x
end

# Test the method with an initial guess and coefficients of the polynomial
coefficients = [5, 0, -2, 1]  # Coefficients of the polynomial 5x^3 - 2x + 1
initial_guess = 2.0
root = newton_method(coefficients, initial_guess)
println("Approximate root using Newton's method: ", root)

# Verify the result using fzero from the Roots package
fzero_root = fzero(x -> func(coefficients, x), initial_guess)
println("Root obtained using fzero: ", fzero_root)

>>>Approximate root using Newton's method: -1.241896591850117
   Root obtained using fzero: -1.2418965630344798
```

Q2: Write a Julia program to solve Lambert's equation $we^w = x$ by Newtons method for $x > 0$. Prove that the iterates defined by

$$w_{n+1} = w_n * \frac{(w_n + (\frac{x}{w_n e^{w_n}}))}{w_n + 1}$$

Make the argument that $w_{n+1} > w_n$ when $w_n e^{w_n} < x$ and that $w_{n+1} < w_n$ when $w_n e^{w_n} > x$

```
function lamberts_equation(x, initial_guess; tolerance=1e-10, max_iterations=100)
    w_n = initial_guess
```

```
    for i in 1:max_iterations
        numerator = w_n + (x / (w_n * exp(w_n)))
        denominator = w_n + 1
        w_n_plus_1 = w_n * (numerator / denominator)

        if abs(w_n_plus_1 - w_n) < tolerance
            break
        end

        w_n = w_n_plus_1
    end

    return w_n
end

# Example usage
x = 5.0  # Example value for x
initial_guess = 2  # Example initial guess for w
solution = lamberts_equation(x, initial_guess)
println("Solution to x = $x: w=$solution.")

>>>Solution to x = 5: w=1.3267246652423588.
```

PROOF:
To prove the two conditions:
1. $w_{n+1} > w_n$ when $w_n e^{w_n} < x$
2. $w_{n+1} < w_n$ when $w_n e^{w_n} > x$

we can utilize the derivative of the Lamberts function $we^w - x$ with respect to w and apply the first derivative test.

Lets define $f(w) = we^w - x$. The derivative of this function is $f'(w) = (w+1)e^w$. Now lets consider the value of $f'(w)$ for different values of w.

1. When $we^w < x$, the value of $f'(w)$ is positive, implying that $f(w)$ is increasing. Consequently, at any point $w = w_n$, we have $w_{n+1} > w$
2. When $we^w > x$, the value of $f'(w)$ is negative, implying that $f(w)$ is decreasing. Consequently, at any point $w = w_n$, we have $w_{n+1} < w$

Thus the conditions $w_{n+1} > w_n$ when $w_n e^{w_n} < x$ and $w_{n+1} < w_n$ when $w_n e^{w_n} > x$ are satisfied based on the behavior the derivative of the function $f(w)$