CSU33081: Computational Mathematics

Assignment 1 - Senán d'Art - 17329580

Question 1

(a)
(ii) 13

(b)
(i) 0

The following MatLab code was used to get the results. It is a recursive algorithm that gets the determinant of a matrix by breaking it down into the sum of determinants of smaller matrices.

```
function D = Determinant (A)
    matrixSize = size(A);
    isInRange = ismember(matrixSize(1,1), [2 3 4]);
    if(matrixSize(1,1) ~= matrixSize(1,2) || ~isInRange(1,1))
        D = "The matrix must be square.";
        return
    end
    D = detRecurse(A);
end

function detSum = detRecurse(mat)
    detSum = 0;
    matSize = size(mat);
    if matSize(1,1) ~= 2
        pos = true;
        for i = 1:matSize(1,1)
            tmpSum = mat(1,i)*detRecurse(getSubMat(mat, 1, i));
            if pos
                detSum = detSum + tmpSum;
            else
                detSum = detSum - tmpSum;
            end
            pos = ~pos;
        end
    else
        detSum = (mat(1,1)*mat(2,2))-(mat(2,1)*mat(1,2));
    end
end

function subMat = getSubMat(mat, row, col)
    matSize = size(mat);
    rowVec = 1:matSize(1,1);
    colVec = 1:matSize(1,2);
    rowVec(row) = [];
    colVec(col) = [];
    subMat = mat(rowVec, colVec);
end
```

Question 2

For the following answers:

$$f(x) = x - 2e^{-x}$$

(a) Bisection Method
(ii) *0.8125*

$a = 0, \quad b = 1$

$$f(0.5) = -0.713$$

$a = 0.5, \quad b = 1$

$$f(0.75) = -0.195$$

$a = 0.75, \quad b = 1$

$$f(0.875) = 0.041$$

$a = 0.75, \quad b = 0.875$

$$f(0.8125) = 0.075$$

(b) Secant Method

(ii) *0.85261*

$$x_3 = x_2 - \frac{f(x_2)(x_1 - x_2)}{f(x_1) - f(x_2)}$$

$x_1 = 0, \quad x_2 = 1$

$$x_3 = 1 - \frac{f(1)(0-1)}{f(0) - f(1)} = 0.883298$$

$x_1 = 1, \quad x_2 = 0.883298$

$$x_3 = 0.851584$$

$x_1 = 0.883298, \quad x_2 = 0.851584$

$$x_3 = 0.850555$$

$x_1 = 0.851584, \quad x_2 = 0.85055$

$$x_3 = 0.852605 \approx 0.85261$$

(c) Newton's Method

(ii) *0.85261*

$$f(x) = x - 2e^{-x} \qquad f'(x) = 1 + 2e-x$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$x_1 = 1$

$$x_2 = 1 - \frac{1 - 2e^{-1}}{1 + 2e^{-1}} = 0.847766$$

$x_1 = 0.847766$

$$x_2 = 0.847766 - \frac{0.847766 - 2e^{-0.847766}}{1 + 2e^{-0.847766}} = 0.852600$$

$x_1 = 0.852600$

$$x_2 = 0.852606$$

$x_1 = 0.852606$

$$x_2 = 0.852606 \approx 0.85261$$

Question 3

Answer: (i)

Matrix (a):

$$\begin{matrix} -1 & 2 & 1 \\ 2 & 2 & -4 \\ 0.2 & 1 & 0.5 \end{matrix}$$

Inverse (a):

$$\begin{matrix} -0.7143 & 0 & 1.4286 \\ 0.2571 & 0.1000 & 0.2857 \\ -0.2286 & -0.2000 & 0.8571 \end{matrix}$$

Matrix (b):

$$\begin{matrix} -1 & -2 & 1 & 2 \\ 1 & 1 & -4 & -2 \\ 1 & -2 & -4 & -2 \\ 2 & -4 & 1 & -2 \end{matrix}$$

Inverse (b):

$$\begin{matrix} 1.6667 & 2.8889 & -2.2222 & 1.0000 \\ 0 & 0.3333 & -0.3333 & 0 \\ -0.3333 & -0.4444 & 0.1111 & 0 \\ 1.5000 & 2.0000 & -1.5000 & 0.5000 \end{matrix}$$

The following code was used to solve the inverse of both matrices:

```
function Ainv =Inverse (A)
    matSize = size(A);
    if length(matSize) > 2 || matSize(1) ~= matSize(2)
        Ainv = "Matrix is not square";
       return
    end

    Ainv = eye(matSize(1));
    for i = 1:matSize(1)
        for j = i:matSize(1) % get the next non-zero
            if A(i,j) ~= 0 % we got one
                %swap rows
                tmp = A(i,:);
                A(i,:) = A(j,:);
                A(j,:) = tmp;

                tmpInv = Ainv(i,:);
                Ainv(i,:) = Ainv(j,:);
                Ainv(j,:) = tmpInv;
                break
            end
        end
        tmp = A(i,i);
        A(i,:) = A(i,:) / tmp;
        Ainv(i,:) = Ainv(i,:) / tmp;
        for j = 1:matSize(1)
           if j ~= i
              tmp = A(j,i);
              A(j,:) = A(j,:) - (A(i,:) * tmp);
              Ainv(j,:) = Ainv(j,:) - (Ainv(i,:) * tmp);
           end
        end
    end
end
```