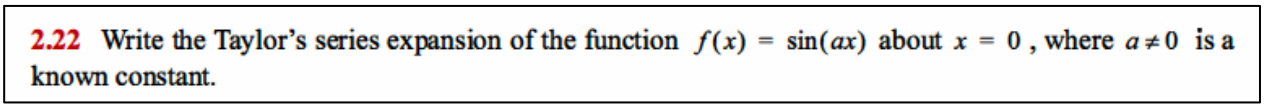
CS3081 – Computational Mathematics
Recommended Questions Solutions



2.22 Write the Taylor's series expansion of the function $f(x) = \sin(ax)$ about $x = 0$, where $a \neq 0$ is a known constant.

Consider a function $f(x)$ that is differentiable $n+1$ times in an interval containing the point $x = x_0$.

The Taylor theorem states that for each x in the interval, there exists a value $x = \xi$ between x and x_0 such that:

$$f(x) = \left[f(x_0) + (x - x_0)\frac{df}{dx}\bigg|_{x=x_0} + \frac{(x-x_0)^2}{2!}\frac{d^2 f}{dx^2}\bigg|_{x=x_0} + \frac{(x-x_0)^3}{3!}\frac{d^3 f}{dx^3}\bigg|_{x=x_0} + \cdots + \frac{(x-x_0)^n}{(n+1)!}\frac{d^n f}{dx^n}\bigg|_{x=x_0} + R_n(x) \right]$$

Where $R_n(x)$ called the remainder and is given by:

$$R_n(x) = \frac{(x-x_0)^{n+1}}{(n+1)!}\frac{d^{n+1} f}{dx^{n+1}}\bigg|_{x=\xi}$$

The Taylor series expansion of the function $f(x) = \sin\,(ax)$ about $x = 0$ (i.e $x_0 = 0$) is therefore as follows:

$$f(x) = \left[\sin\,(ax_0) + (x - x_0)\frac{d\sin\,(ax_0)}{dx}\bigg|_{x=x_0} + \frac{(x-x_0)^2}{2!}\frac{d^2 \sin\,(ax_0)}{dx^2}\bigg|_{x=x_0} + \frac{(x-x_0)^3}{3!}\frac{d^3 \sin\,(ax_0)}{dx^3}\bigg|_{x=x_0} + \cdots \right]$$

$$f(x) = \left[\sin\,(ax_0) + (x - x_0)\left[\mathrm{a}\cos\,(ax_0)|_{x=x_0}\right] + \frac{(x-x_0)^2}{2!}\left[-\mathrm{a}^2\sin\,(ax_0)|_{x=x_0}\right] + \frac{(x-x_0)^3}{3!}\left[-\mathrm{a}^3\cos\,(ax_0)|_{x=x_0}\right] + \cdots \right]$$

Substitute $x_0 = 0$ into this function:

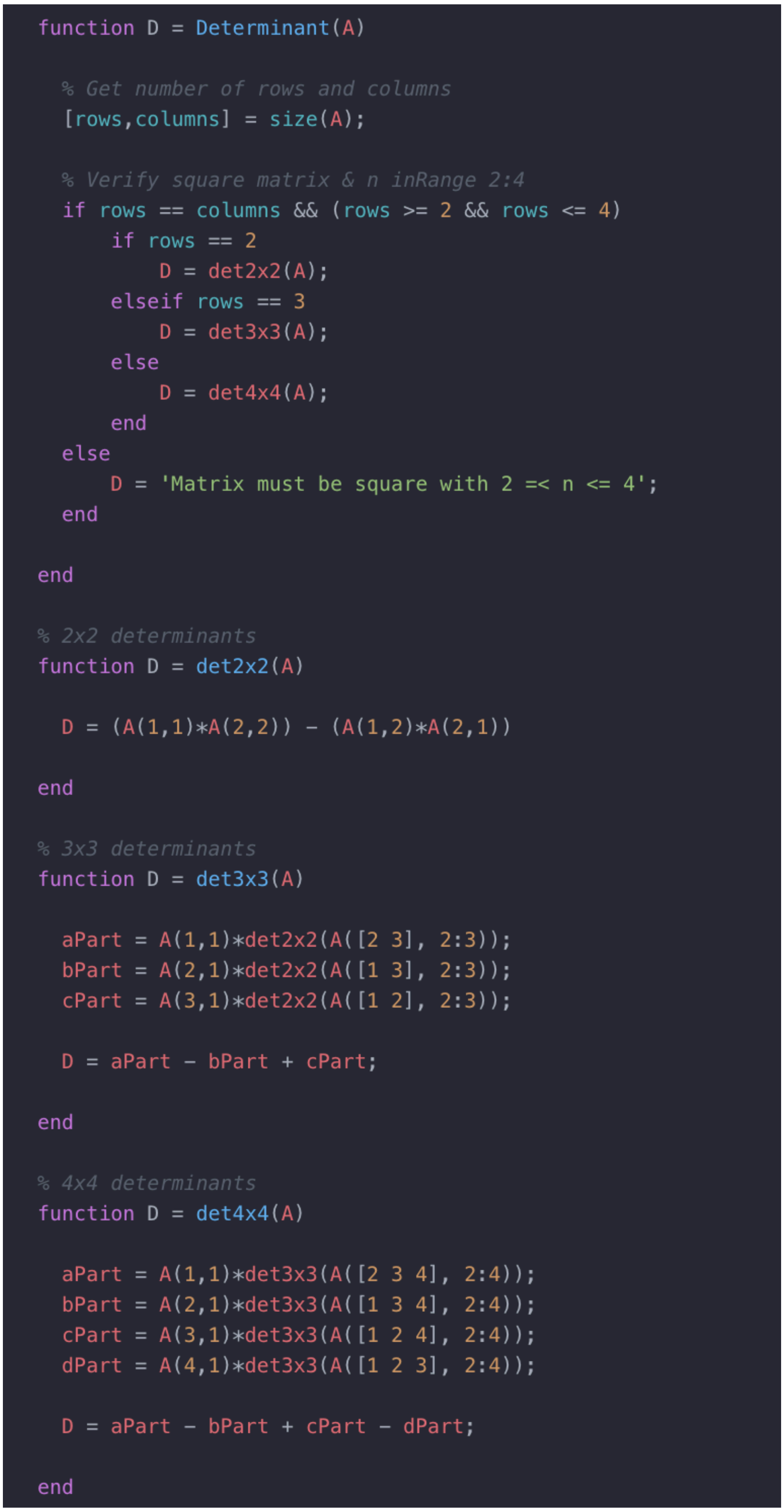
$$f(x) = \left[\sin\,(0) + x[\mathrm{a}\cos\,(0)] + \frac{(x)^2}{2!}[-\mathrm{a}^2\sin\,(0)] + \frac{(x)^3}{3!}[-\mathrm{a}^3\cos\,(0)] + \cdots \right]$$

$$f(x) = 0 + x[\mathrm{a}] + \frac{(x)^3}{3!}[-\mathrm{a}^3] + \cdots$$

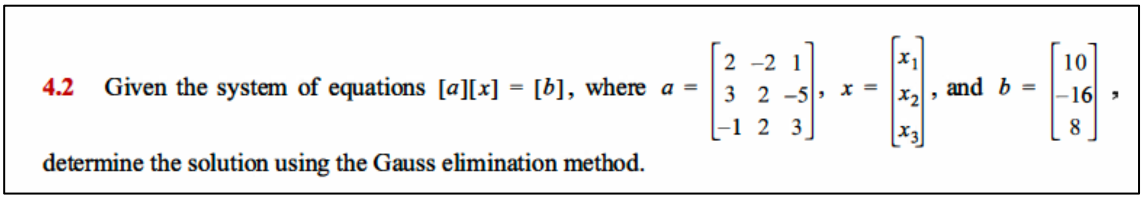
Therefore, the Taylor's series expansion of the function $f(x) = \sin\,(ax)$ is:

$$\boldsymbol{f(x) = \frac{(x)^1}{1!}[\mathrm{a}] - \frac{(x)^3}{3!}[\mathrm{a}^3] + \frac{(x)^5}{5!}[\mathrm{a}^5] + \cdots}$$

2.31 Write a user-defined MATLAB function that calculates the determinant of a square $(n \times n)$ matrix, where n can be 2, 3, or 4. For function name and arguments, use `D = Determinant(A)`. The input argument `A` is the matrix whose determinant is calculated. The function `Determinant` should first check if



```matlab
function D = Determinant(A)

  % Get number of rows and columns
  [rows,columns] = size(A);

  % Verify square matrix & n inRange 2:4
  if rows == columns && (rows >= 2 && rows <= 4)
      if rows == 2
          D = det2x2(A);
      elseif rows == 3
          D = det3x3(A);
      else
          D = det4x4(A);
      end
  else
      D = 'Matrix must be square with 2 =< n <= 4';
  end

end

% 2x2 determinants
function D = det2x2(A)

  D = (A(1,1)*A(2,2)) - (A(1,2)*A(2,1))

end

% 3x3 determinants
function D = det3x3(A)

  aPart = A(1,1)*det2x2(A([2 3], 2:3));
  bPart = A(2,1)*det2x2(A([1 3], 2:3));
  cPart = A(3,1)*det2x2(A([1 2], 2:3));

  D = aPart - bPart + cPart;

end

% 4x4 determinants
function D = det4x4(A)

  aPart = A(1,1)*det3x3(A([2 3 4], 2:4));
  bPart = A(2,1)*det3x3(A([1 3 4], 2:4));
  cPart = A(3,1)*det3x3(A([1 2 4], 2:4));
  dPart = A(4,1)*det3x3(A([1 2 3], 2:4));

  D = aPart - bPart + cPart - dPart;

end
```



4.2 Given the system of equations $[a][x] = [b]$, where $a = \begin{bmatrix} 2 & -2 & 1 \\ 3 & 2 & -5 \\ -1 & 2 & 3 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, and $b = \begin{bmatrix} 10 \\ -16 \\ 8 \end{bmatrix}$, determine the solution using the Gauss elimination method.

First, substitute the given matrices in $[a] * [x] = [b]$ form:

$$\begin{bmatrix} 2 & -2 & 1 \\ 3 & 2 & -5 \\ -1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ -16 \\ 8 \end{bmatrix} \begin{array}{l} \rightarrow R_1 \\ \rightarrow R_2 \\ \rightarrow R_3 \end{array}$$

We then want to eliminate the all of the bottom row ($R_3's$) elements besides the 3 so that we can get a value for x3. Therefore perform the following row operations on matrices $[a]$ and $[b]$:

- $R_2 = 2(R_2)$ - $3(R_1)$
- $R_3 = 2(R_3) + R_1$

$$\begin{array}{r} \rightarrow R_1 \\ \rightarrow \mathbf{2(R_2) - 3(R_1)} \\ \rightarrow \mathbf{2(R_3) - R_1} \end{array} \begin{bmatrix} 2 & -2 & 1 \\ 0 & 10 & -13 \\ 0 & 2 & 7 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ -62 \\ 26 \end{bmatrix}$$

- $R_3 = 5(R_3)$ - R_2

$$\begin{array}{r} \rightarrow R_1 \\ \rightarrow R_2 \\ \rightarrow \mathbf{5(R_3) - R_2} \end{array} \begin{bmatrix} 2 & -2 & 1 \\ 0 & 10 & -13 \\ 0 & 0 & 48 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ -62 \\ 196 \end{bmatrix}$$

From this we can now solve for x_3:

$$48x_3 = 196$$

$$\boldsymbol{x_3 = 4}$$

We can then sub this into row 2 and solve for x_2:

$$10x_2 - 13x_3 = -63$$

$$10x_2 - 13(4) = -63$$

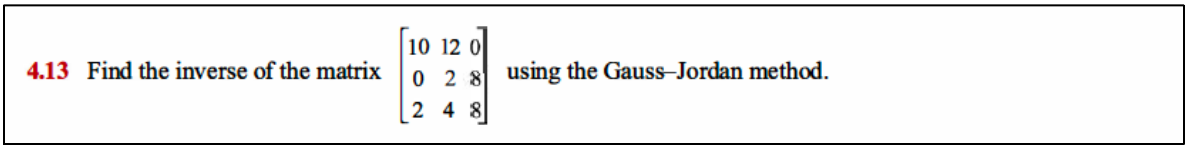
$$\boldsymbol{x_2 = -1}$$

And then solve for x_1 using row 1:

$$2x_1 - 2x_2 + 1x_3 = 10$$

$$2x_1 - 2(-1) + 1(4) = 10$$

$$\boldsymbol{x_1} = \mathbf{2}$$



4.13 Find the inverse of the matrix $\begin{bmatrix} 10 & 12 & 0 \\ 0 & 2 & 8 \\ 2 & 4 & 8 \end{bmatrix}$ using the Gauss–Jordan method.

First write out the matrix and the augmented matrix:

$$\begin{bmatrix} 10 & 12 & 0 \\ 0 & 2 & 8 \\ 2 & 4 & 8 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The aim of the game here is to reduce the LHS matrix to the RHS matrix by performing row operations:

- $R_1 = R_1/10$

$$\begin{array}{r} \to R_1/10 \\ \to R_2 \\ \to R_3 \end{array} \begin{bmatrix} 1 & 1.2 & 0 \\ 0 & 2 & 8 \\ 2 & 4 & 8 \end{bmatrix} \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- $R_3 = R_3 - 2(R_1)$

$$\begin{array}{r} \to R_1 \\ \to R_2 \\ \to R_3 - 2(R_1) \end{array} \begin{bmatrix} 1 & 1.2 & 0 \\ 0 & 2 & 8 \\ 0 & 1.6 & 8 \end{bmatrix} \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.2 & 0 & 1 \end{bmatrix}$$

- $R_2 = R_2/2$

$$\begin{array}{r} \to R_1 \\ \to R_2/2 \\ \to R_3 \end{array} \begin{bmatrix} 1 & 1.2 & 0 \\ 0 & 1 & 4 \\ 0 & 1.6 & 8 \end{bmatrix} \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.5 & 0 \\ -0.2 & 0 & 1 \end{bmatrix}$$

- $R_1 = R_1 - 1.2(R_2)$

$$\begin{array}{r} \to R_1 - 1.2(R_2) \\ \to R_2 \\ \to R_3 \end{array} \begin{bmatrix} 1 & 0 & -4.8 \\ 0 & 1 & 4 \\ 0 & 1.6 & 8 \end{bmatrix} \begin{bmatrix} 0.1 & -0.6 & 0 \\ 0 & 0.5 & 0 \\ -0.2 & 0 & 1 \end{bmatrix}$$

- $R_3 = R_3 - 1.6(R_2)$

$$\begin{array}{r} \to R_1 \\ \to R_2 \\ \to R_3 - 1.6(R_2) \end{array} \begin{bmatrix} 1 & 0 & -4.8 \\ 0 & 1 & 4 \\ 0 & 0 & 1.6 \end{bmatrix} \begin{bmatrix} 0.1 & -0.6 & 0 \\ 0 & 0.5 & 0 \\ -0.2 & -0.8 & 1 \end{bmatrix}$$

- $R_3 = R_3/1.6$

$$\begin{array}{r} \rightarrow R_1 \\ \rightarrow R_2 \\ \rightarrow R_3/1.6 \end{array} \begin{bmatrix} 1 & 0 & -4.8 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 & -0.6 & 0 \\ 0 & 0.5 & 0 \\ -0.125 & -0.5 & 0.625 \end{bmatrix}$$

- $R_1 = R_1 + 4.8(R_3)$
- $R_2 = R_2 + 4(R_3)$

$$\begin{array}{r} \rightarrow R_1 + 4.8(R_3) \\ \rightarrow R_2 + 4(R_3) \\ \rightarrow R_3 \end{array} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.5 & -3 & 3 \\ 0.5 & 2.5 & -2.5 \\ -0.125 & -0.5 & 0.625 \end{bmatrix}$$

4.19 **Find the condition number of the matrix in Problem 4.13 using the 1-norm.**

$$a = \begin{bmatrix} 10 & 12 & 0 \\ 0 & 2 & 8 \\ 2 & 4 & 8 \end{bmatrix}$$

We know from 4.13 that the inverse of a is:

$$a^{-1} = \begin{bmatrix} -0.5 & -3 & 3 \\ 0.5 & 2.5 & -2.5 \\ -0.125 & -0.5 & 0.625 \end{bmatrix}$$

Consider the formula for the condition number of a matrix:

$$cond[a] = \|[a]\|\|[a]^{-1}\| \ldots$$

Where:

- $\|[a]\|$ denotes the norm of matrix a
- $\|[a]^{-1}\|$ denotes the norm of inverse of matrix a

Write the formula to find the 1-norm of a matrix:

$$\|[a]\|_1 = \max_{1 \le j \le n} \sum_{j=1}^{n} |a_{ij}|$$

Consider the matrix a as stated above and 3 for n:

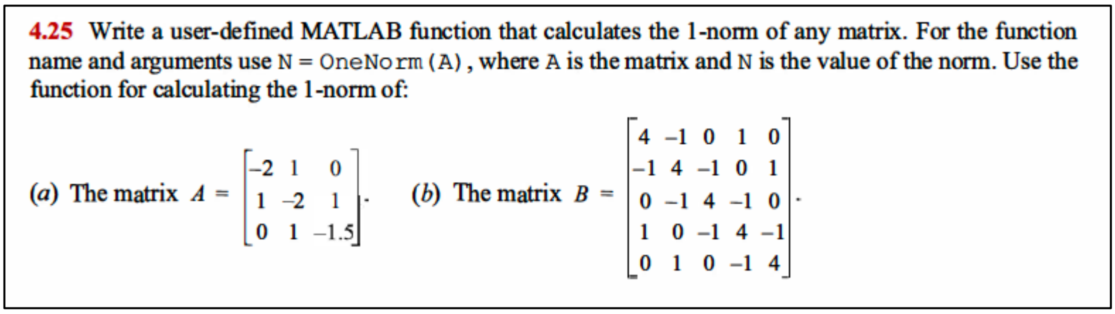
$$\|[a]\|_1 = \max_{1 \le j \le 3} \sum_{j=1}^{3} |a_{ij}|$$

$$\|[a]\|_1 = \max_{1 \le j \le 3} [(|10| + |0| + |2|), (|12| + |2| + |4|), (|0| + |8| + |8|)]$$

$$\|[a]\|_1 = \max_{1 \le j \le 3} [12, 18, 16]$$

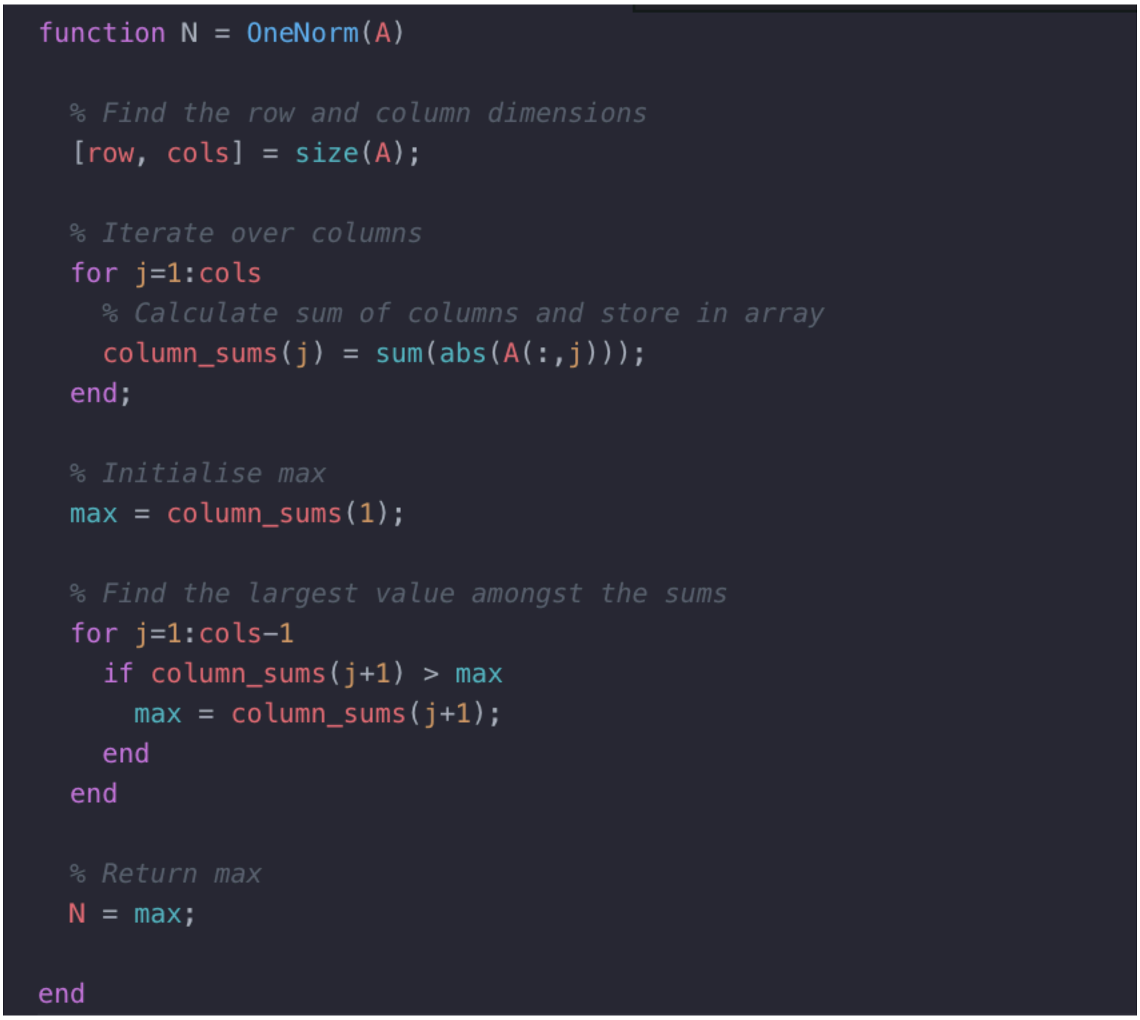
$$\boldsymbol{\|[a]\|_1 = 18}$$

Therefore, the 1-norm of matrix a is 18.

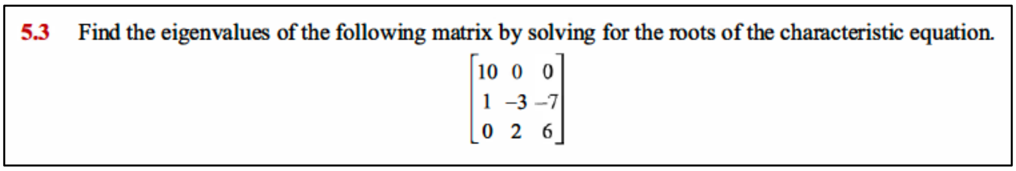


4.25 Write a user-defined MATLAB function that calculates the 1-norm of any matrix. For the function name and arguments use `N = OneNorm(A)`, where `A` is the matrix and `N` is the value of the norm. Use the function for calculating the 1-norm of:

(*a*) The matrix $A = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1.5 \end{bmatrix}$. (*b*) The matrix $B = \begin{bmatrix} 4 & -1 & 0 & 1 & 0 \\ -1 & 4 & -1 & 0 & 1 \\ 0 & -1 & 4 & -1 & 0 \\ 1 & 0 & -1 & 4 & -1 \\ 0 & 1 & 0 & -1 & 4 \end{bmatrix}$.



```
function N = OneNorm(A)

  % Find the row and column dimensions
  [row, cols] = size(A);

  % Iterate over columns
  for j=1:cols
    % Calculate sum of columns and store in array
    column_sums(j) = sum(abs(A(:,j)));
  end;

  % Initialise max
  max = column_sums(1);

  % Find the largest value amongst the sums
  for j=1:cols-1
    if column_sums(j+1) > max
      max = column_sums(j+1);
    end
  end

  % Return max
  N = max;

end
```



5.3 Find the eigenvalues of the following matrix by solving for the roots of the characteristic equation.

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix}$$

The expression for the characteristic equation to find the Eigen values is:

$$det[a - \lambda I] = 0$$

$$|a - \lambda I| = 0$$

As the matrix a is of order 3x3, the order of the identity matrix is also 3x3:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Calculate the matrix $a - \lambda I$:

$$a - \lambda I = \begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

$$a - \lambda I = \begin{bmatrix} 10 - \lambda & 0 & 0 \\ 1 & -3 - \lambda & -7 \\ 0 & 2 & 6 - \lambda \end{bmatrix}$$

Using this calculate the characteristic equation $|a - \lambda I| = 0$:

$$|a - \lambda I| = 0$$

$$\begin{vmatrix} 10 - \lambda & 0 & 0 \\ 1 & -3 - \lambda & -7 \\ 0 & 2 & 6 - \lambda \end{vmatrix} = 0$$

$$10 - \lambda[(-3 - \lambda)(6 - \lambda) - (-7)(2)] - 0 + 0 = 0$$

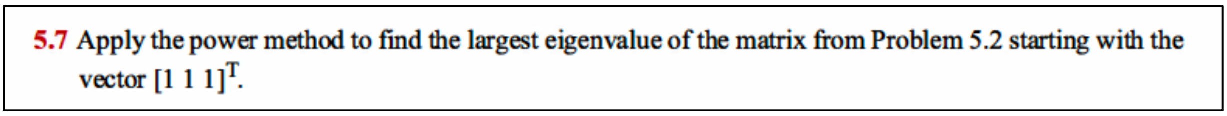
$$10 - \lambda[3\lambda - 6\lambda + \lambda^2 - 18 + 14] = 0$$

$$(10 - \lambda)(\lambda^2 - 3\lambda - 4) = 0$$

$$(10 - \lambda)(\lambda + 1)(\lambda - 4) = 0$$

$$\boldsymbol{\lambda = 10, \lambda = 4, \lambda = -1}$$

Therefore, the Eigen values of the matrix are 10, 4 and -1



5.7 Apply the power method to find the largest eigenvalue of the matrix from Problem 5.2 starting with the vector $[1\ 1\ 1]^T$.

$$a = \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

For $i = 1$, the Eigen vector $[x]_1 = [1 \quad 1 \quad 1]^T$ that is:

$$[x]_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Calculate the next Eigen vector $[x]_2$ for $i = 2$

$$[x]_2 = a[x]_1$$

$$[x]_2 = \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$[x]_2 = \begin{bmatrix} 1+1+2 \\ 4+1+5 \\ 0+0+1 \end{bmatrix}$$

$$[x]_2 = \begin{bmatrix} 4 \\ 10 \\ 1 \end{bmatrix}$$

We must then extract the highest element from the vector and normalise the vector:

$$[x]_2 = \begin{bmatrix} 4 \\ \mathbf{10} \\ 1 \end{bmatrix}$$

$$[x]_2 = 10 \begin{bmatrix} 4/10 \\ \mathbf{10/10} \\ 1/10 \end{bmatrix}$$

$$[x]_2 = 10 \begin{bmatrix} 0.4 \\ 1 \\ 0.1 \end{bmatrix}$$

Therefore, the normalized unit vector of $[x]_2$ is $\begin{bmatrix} 0.4 \\ 1 \\ 0.1 \end{bmatrix}$

We then repeat these steps, for the next Eigen vector $[x]_3$ for $i = 3$

$$[x]_3 = a[x]_2$$

$$[x]_3 = \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4 \\ 1 \\ 0.1 \end{bmatrix}$$

$$[x]_3 = \begin{bmatrix} 0.4 + 1 + 0.2 \\ 1.6 + 1 + 0.5 \\ 0 + 0 + 0.1 \end{bmatrix}$$

$$[x]_3 = \begin{bmatrix} 1.6 \\ 3.1 \\ 0.1 \end{bmatrix}$$

We must then extract the highest element from the vector and normalise the vector:

$$[x]_3 = \begin{bmatrix} 1.6 \\ \mathbf{3.1} \\ 0.1 \end{bmatrix}$$

$$[x]_3 = 3.1 \begin{bmatrix} 1.6/\mathbf{3.1} \\ 3.1/\mathbf{3.1} \\ 0.1/\mathbf{3.1} \end{bmatrix}$$

$$[x]_3 = 3.1 \begin{bmatrix} 0.516 \\ 1 \\ 0.0323 \end{bmatrix}$$

Therefore, the normalized unit vector of $[x]_3$ is $\begin{bmatrix} 0.516 \\ 1 \\ 0.0323 \end{bmatrix}$

We then repeat these steps, for the next Eigen vector $[x]_4$ for $i = 4$

$$[x]_4 = a[x]_3$$

$$[x]_4 = \begin{bmatrix} 1 & 1 & 2 \\ 4 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.516 \\ 1 \\ 0.0323 \end{bmatrix}$$

$$[x]_4 = \begin{bmatrix} 0.516 + 1 + 0.0646 \\ 2.064 + 1 + 0.1615 \\ 0 + 0 + 0.0323 \end{bmatrix}$$

$$[x]_4 = \begin{bmatrix} 1.58 \\ 3.225 \\ 0.0323 \end{bmatrix}$$

We must then extract the highest element from the vector and normalise the vector:

$$[x]_4 = \begin{bmatrix} 1.58 \\ 3.225 \\ 0.0323 \end{bmatrix}$$

$$[x]_4 = 3.225 \begin{bmatrix} 1.58/3.225 \\ 3.225/3.225 \\ 0.0323/3.225 \end{bmatrix}$$

$$[x]_4 = 3.225 \begin{bmatrix} 0.49 \\ 1 \\ 0.01 \end{bmatrix}$$

Therefore, the normalized unit vector of $[x]_4$ is $\begin{bmatrix} 0.49 \\ 1 \\ 0.01 \end{bmatrix}$

This is repeated and on our 6th iteration we get:

$$[x]_6 = \begin{bmatrix} 1.5066 \\ 3.0165 \\ 0.0033 \end{bmatrix}$$

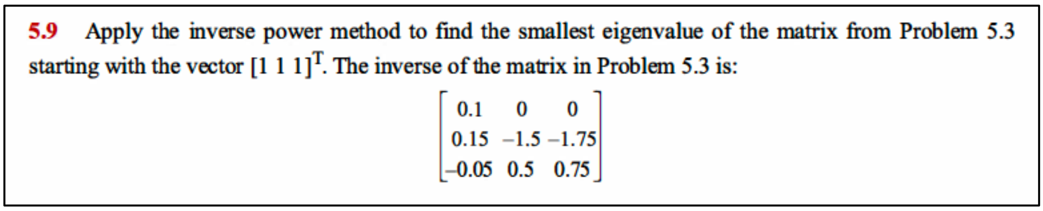
$$[x]_6 = 3.0165 \begin{bmatrix} 1.5066/3.0165 \\ 3.0165/3.0165 \\ 0.0033/3.0165 \end{bmatrix}$$

$$[x]_6 = 3.0165 \begin{bmatrix} 0.5 \\ 1 \\ 0.001 \end{bmatrix}$$

Since the difference between the multiplicative factors (was 3.01 for $[x]_5$) is relatively small we can terminate our iterations here at $[x]_6$.

We can also say that the largest Eigenvalue for the given matrix a is:

3.0165



5.9 Apply the inverse power method to find the smallest eigenvalue of the matrix from Problem 5.3 starting with the vector $[1\ 1\ 1]^T$. The inverse of the matrix in Problem 5.3 is:

$$\begin{bmatrix} 0.1 & 0 & 0 \\ 0.15 & -1.5 & -1.75 \\ -0.05 & 0.5 & 0.75 \end{bmatrix}$$

$$a = \begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 2 & 6 \end{bmatrix}$$

$$a^{-1} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0.15 & -1.5 & -1.75 \\ -0.05 & 0.5 & 0.75 \end{bmatrix}$$

For $i = 1$, the Eigen vector $[x]_1 = [1 \quad 1 \quad 1]^T$ that is:

$$[x]_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Calculate the next Eigen vector $[x]_2$ for $i = 2$

$$[x]_2 = a^{-1}[x]_1$$

$$[x]_2 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0.15 & -1.5 & -1.75 \\ -0.05 & 0.5 & 0.75 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$[x]_2 = \begin{bmatrix} 0.1 \\ -3.1 \\ 1.2 \end{bmatrix}$$

We must then extract the highest element from the vector and normalise the vector:

$$[x]_2 = \begin{bmatrix} 0.1 \\ -3.1 \\ 1.2 \end{bmatrix}$$

$$[x]_2 = -3.1 \begin{bmatrix} 0.1/-3.1 \\ -3.1 \\ 1.2/-3.1 \end{bmatrix}$$

$$[x]_2 = -3.1 \begin{bmatrix} -0.0323 \\ 1 \\ -0.3871 \end{bmatrix}$$

Therefore, the normalized unit vector of $[x]_2$ is $\begin{bmatrix} -0.0323 \\ 1 \\ -0.3871 \end{bmatrix}$

We then repeat these steps, for the next Eigen vector $[x]_3$ for $i = 3$

$$[x]_3 = a^{-1}[x]_2$$

$$[x]_3 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0.15 & -1.5 & -1.75 \\ -0.05 & 0.5 & 0.75 \end{bmatrix} \begin{bmatrix} -0.0323 \\ 1 \\ -0.3871 \end{bmatrix}$$

$$[x]_3 = \begin{bmatrix} -0.00323 \\ -0.8274 \\ 0.2113 \end{bmatrix}$$

We must then extract the highest element from the vector and normalise the vector:

$$[x]_3 = \begin{bmatrix} -0.00323 \\ -0.8274 \\ 0.2113 \end{bmatrix}$$

$$[x]_2 = -0.8274 \begin{bmatrix} -0.00323/-0.8274 \\ -0.8274/-0.8274 \\ 0.2113/-0.8274 \end{bmatrix}$$

$$[x]_2 = -0.8274 \begin{bmatrix} 0.0039 \\ 1 \\ -0.2554 \end{bmatrix}$$

Therefore, the normalized unit vector of $[x]_2$ is $\begin{bmatrix} 0.0039 \\ 1 \\ -0.2554 \end{bmatrix}$

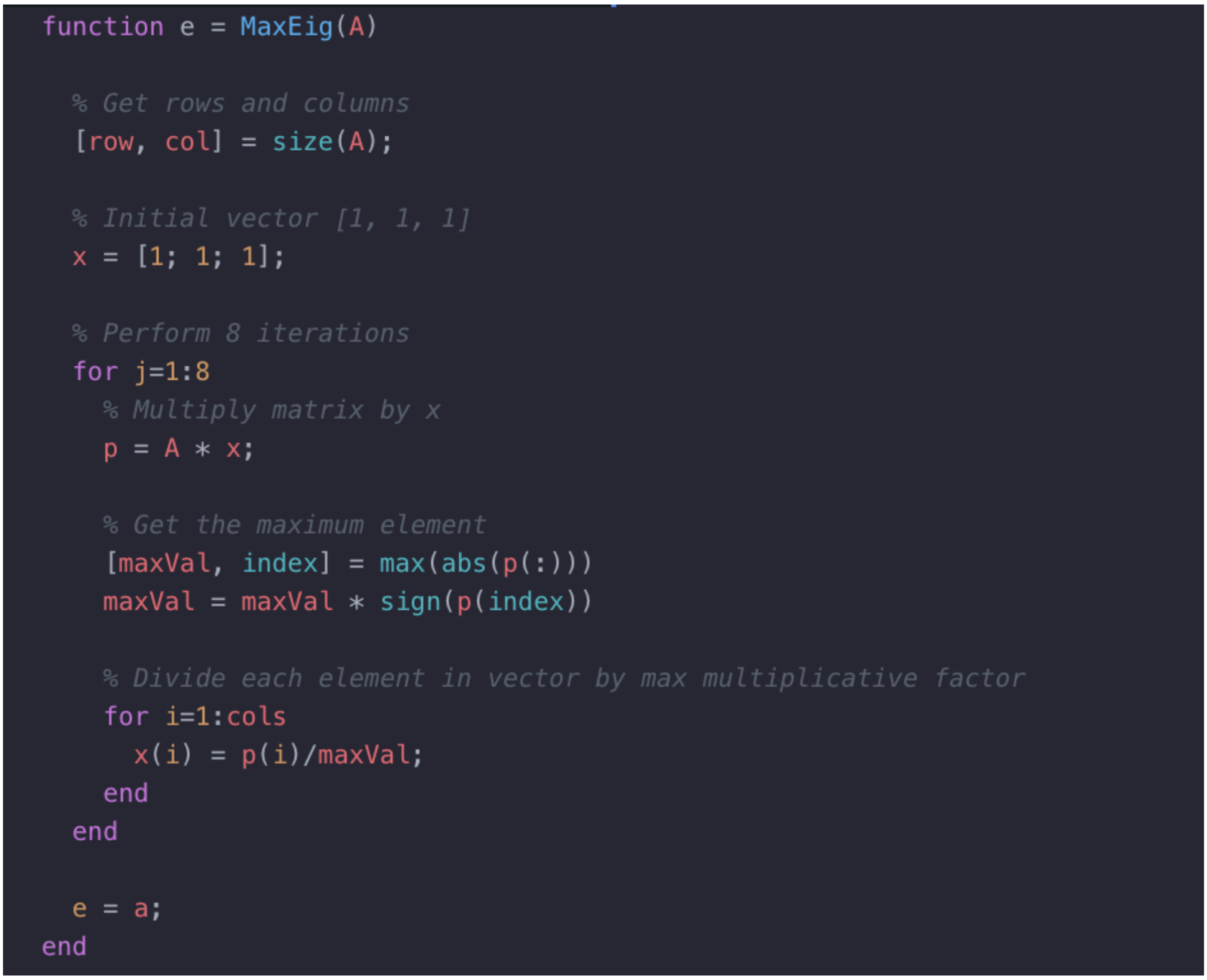
These steps are repeated until the difference between the multiplicative factors becomes significantly small. In our case this happens to be when $[x]_9$'s multiplicative factor is -1 and $[x]_8$'s multiplicative factor is -1.0002. We can terminate the iterations here and confidently say this is approximately the largest Eigenvalue and Eigenvector.

$$[x]_9 = -1 \begin{bmatrix} 0 \\ 1 \\ -0.2857 \end{bmatrix}$$

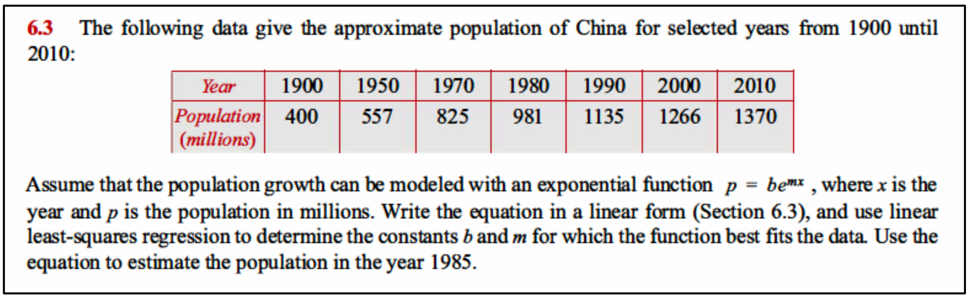
We can also say that the largest Eigenvalue for the given matrix a is:

-1

5.10 Write a user-defined MATLAB function that determines the largest eigenvalue of an $(n \times n)$ matrix by using the power method. For the function name and argument use `e = MaxEig(A)`, where `A` is the matrix and `e` is the value of the largest eigenvalue. Use the function `MaxEig` for calculating the largest eigenvalue of the matrix of Problem 5.8. Check the answer by using MATLAB's built-in function for finding the eigenvalues of a matrix.



```
function e = MaxEig(A)

  % Get rows and columns
  [row, col] = size(A);

  % Initial vector [1, 1, 1]
  x = [1; 1; 1];

  % Perform 8 iterations
  for j=1:8
    % Multiply matrix by x
    p = A * x;

    % Get the maximum element
    [maxVal, index] = max(abs(p(:)))
    maxVal = maxVal * sign(p(index))

    % Divide each element in vector by max multiplicative factor
    for i=1:cols
      x(i) = p(i)/maxVal;
    end
  end

  e = a;
end
```



6.3 The following data give the approximate population of China for selected years from 1900 until 2010:

Year	1900	1950	1970	1980	1990	2000	2010
Population (millions)	400	557	825	981	1135	1266	1370

Assume that the population growth can be modeled with an exponential function $p = be^{mx}$, where x is the year and p is the population in millions. Write the equation in a linear form (Section 6.3), and use linear least-squares regression to determine the constants b and m for which the function best fits the data. Use the equation to estimate the population in the year 1985.

Consider the function:

$$p = be^{mx}$$

Apply the natural logarithm on both sides to bring down the exponent:

$$\ln(p) = \ln\,(be^{mx})$$

$$\ln(p) = \ln(b) + \ln\,(e^{mx})$$

$$\ln(p) = \ln(b) + mx\ln\,(e)$$

$$\ln(p) = \ln(b) + mx$$

Now, define the terms as follows:

$$\ln(p) = Y$$

$$\ln(b) = B$$

We can now write the equation in linear form as:

$$Y = mx + B$$

Re-write the data in the table taking the natural logarithm of each entry:

Year (x)	1900	1950	1970	1980	1990	2000	2010
Population ln(p) = Y	5.9915	6.3226	6.7154	6.8886	7.0344	7.1436	7.2226

Now, apply the least square regression method. First, calculate the value of S_x

$$S_x = \sum_{i=1}^{7} x_i$$

$$S_x = 1900 + 1950 + 1970 + 1980 + 1990 + 2000 + 2010$$

$$\boldsymbol{S_x = 13,800}$$

Then, calculate the value of S_y

$$S_y = \sum_{i=1}^{7} y_i$$

$$S_y = 5.9915 + 6.3226 + 6.7154 + 6.8886 + 7.0344 + 7.1436 + 7.2226$$

$$\boldsymbol{S_y = 47.3816}$$

Then, calculate the value of S_{xx}

$$S_{xx} = \sum_{i=1}^{7} {x_i}^2$$

$$S_{xx} = (1900)^2 + \cdots + (2010)^2$$

$$\boldsymbol{S_{xx} = 27214000}$$

Then, calculate the value of S_{xy}

$$S_{xy} = \sum_{i=1}^{7} x_i y_i$$

$$S_{xy} = (1900)(5.9915) + \cdots + (2010)(7.2226)$$

$$\boldsymbol{S_{xy} = 9.3384 \times 10^4}$$

Then, calculate the value of m

$$m = \frac{nS_{xy} - S_x S_y}{nS_{xx} - (S_x)^2}$$

$$m = \frac{7(9.3384 \times 10^4) - (13800)(47.3186)}{7(27214000) - (13800)^2}$$

$$\boldsymbol{m = 0.0119}$$

Then, calculate the value of B

$$B = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - (S_x)^2}$$

$$B = \frac{(27214000)(47.3186) - (9.3384 \times 10^4)(13800)}{7(27214000) - (13800)^2}$$

$$\boldsymbol{B = -16.7383}$$

Now, using our original equations solve the value of b:

$$B = \ln(b)$$

$$b = e^{B}$$

$$b = e^{-16.7383}$$

$$\boldsymbol{b = 5.3785 \times 10^{-8}}$$

Therefore, using our newfound values for b and m, the new relation is:

$$p = 5.3785 \times 10^{-8} e^{0.0119x}$$

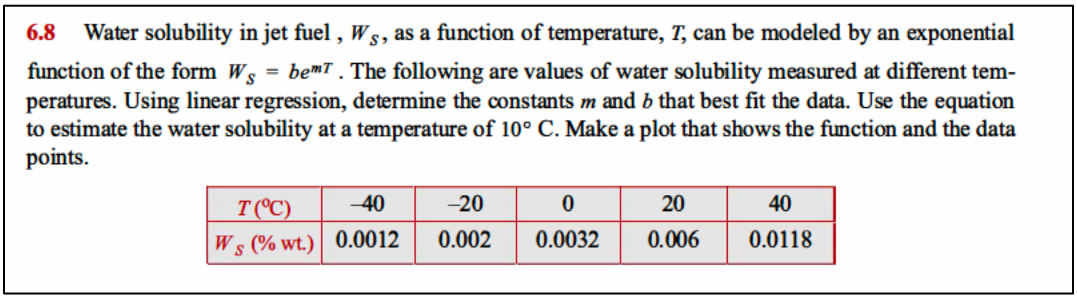
We can then, using this determine an estimation of the population in the year 1985 by letting x equal to 1985 in the above relation:

$$p = 5.3785 \times 10^{-8} e^{0.0119x}$$

$$p = 5.3785 \times 10^{-8} e^{0.0119(1985)}$$

$$p = 975.7718$$

Therefore, we can estimate that the population in 1985 will be approximately 975.7718 million



6.8 Water solubility in jet fuel, W_S, as a function of temperature, T, can be modeled by an exponential function of the form $W_S = be^{mT}$. The following are values of water solubility measured at different temperatures. Using linear regression, determine the constants m and b that best fit the data. Use the equation to estimate the water solubility at a temperature of 10° C. Make a plot that shows the function and the data points.

T (°C)	–40	–20	0	20	40
W_S (% wt.)	0.0012	0.002	0.0032	0.006	0.0118

Consider the function:

$$W_S = be^{mT}$$

Apply the natural logarithm to both sides:

$$\ln(W_S) = \ln(be^{mT})$$

$$\ln(W_S) = \ln(b) + \ln\left(e^{mT}\right)$$

$$\ln(W_S) = \ln(b) + mT\ln(e)$$

$$\ln(W_S) = \ln(b) + mT$$

Let:

$$\ln(W_S) = Y$$

$$\ln(b) = B$$

We can now express our function as a linear expression (let T =X):

$$Y = mX + B$$

Re-write the data in the table taking the natural logarithm of each entry:

Temp (X)	-40	-20	0	20	40
ln(Ws) (Y)	6.7254	6.2146	5.7446	5.1160	4.4397

Now, apply the least square regression method. First, calculate the value of S_x

$$S_x = \sum_{i=1}^{7} x_i$$

$$S_x = -40 - 20 + 20 + 40$$

$$\boldsymbol{S_x = 0}$$

Then, calculate the value of S_y

$$S_y = \sum_{i=1}^{7} y_i$$

$$S_y = -6.725 - 6.2146 - 5.7446 - 5.1160 - 4.4397$$

$$\boldsymbol{S_y = -28.2403}$$

Then, calculate the value of S_{xx}

$$S_{xx} = \sum_{i=1}^{7} {x_i}^2$$

$$S_{xx} = (-40)^2 + \cdots + (40)^2$$

$$\boldsymbol{S_{xx} = 4000}$$

Then, calculate the value of S_{xy}

$$S_{xy} = \sum_{i=1}^{7} x_i y_i$$

$$S_{xy} = (-40)(-6.7254) + \cdots + (2010)(7.2226)$$

$$\boldsymbol{S_{xy} = 113.4034}$$

Then, calculate the value of m

$$m = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x S_x}$$

$$m = \frac{5(113.4034) - (0)(28.2403)}{5(4000) - (0)^2}$$

$$\boldsymbol{m = 0.0284}$$

Then, calculate the value of B

$$B = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - S_x S_x}$$

$$B = \frac{(4000)(47.3186) - (113.4034)(0)}{5(4000) - (0)^2}$$

$$\boldsymbol{B = -5.6481}$$

Now, using our original equations solve the value of b:

$$B = \ln(b)$$

$$b = e^{B}$$

$$b = e^{-5.6481}$$

$$\boldsymbol{b = 0.0035}$$

Therefore, using our newfound values for b and m, the new relation is:

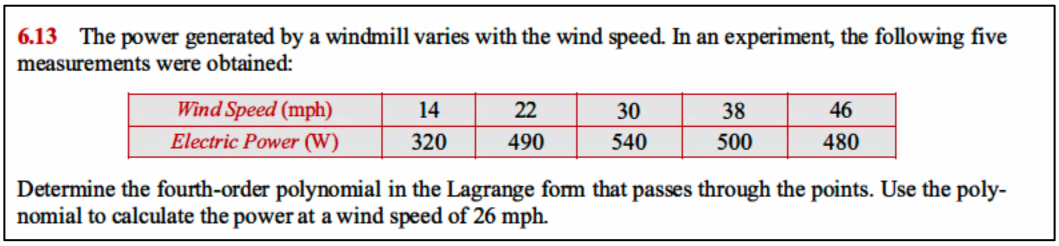
$$W_S = 0.0035e^{0.0284T}$$

We can then, using this determine an estimation of the water solubility in jet fuel at a temperature of 10° C:

$$W_S = 0.0035e^{0.0284T}$$

$$W_S = 0.0035e^{0.0284(10)}$$

$$W_S = 0.0046$$



6.13 The power generated by a windmill varies with the wind speed. In an experiment, the following five measurements were obtained:

Wind Speed (mph)	14	22	30	38	46
Electric Power (W)	320	490	540	500	480

Determine the fourth-order polynomial in the Lagrange form that passes through the points. Use the polynomial to calculate the power at a wind speed of 26 mph.

The n-1th order Lagrange polynomial passing through n points can be defined as:

$$f(x) = \left[\frac{(x-x_2)(x-x_3)\ldots(x-x_n)}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_n)}y_1 + \frac{(x-x_1)(x-x_3)\ldots(x-x_n)}{(x_2-x_1)(x_2-x_3)\ldots(x_2-x_n)}y_2 + \cdots + \frac{(x-x_1)(x-x_2)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x-x_1)(x-x_2)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}y_i + \frac{(x-x_1)(x-x_2)\ldots(x-x_{n-1})}{(x-x_1)(x-x_2)\ldots(x-x_{n-1})}y_n\right]$$

This can also be written as:

$$f(x) = \sum_{i=1}^{n} y_i L_i(x) = \sum_{i=1}^{n} y_i \prod_{j=1, j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

Therefore, the 4th order Lagrange polynomial passing through the five points $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ and (x_5, y_5) is:

$$f(x) = \left[\begin{array}{l} \frac{(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)(x_1-x_5)}y_1 + \\ \frac{(x-x_1)(x-x_3)(x-x_4)(x-x_5)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)(x_2-x_5)}y_2 + \\ \frac{(x-x_1)(x-x_2)(x-x_4)(x-x_5)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)(x_3-x_5)}y_3 + \\ \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_5)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)(x_4-x_5)}y_4 + \\ \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_5-x_1)(x_5-x_2)(x_5-x_3)(x_5-x_4)}y_5 \end{array}\right]$$

Using the data from the supplied table we then sub in the values for the following:

- $(x_1, y_1) = (14, 320)$
- $(x_2, y_2) = (22, 490)$
- $(x_3, y_3) = (30, 540)$
- $(x_4, y_4) = (38, 500)$
- $(x_5, y_5) = (46, 480)$

When we multiply this out, we are then left with the fourth order Lagrange polynomial as follows:

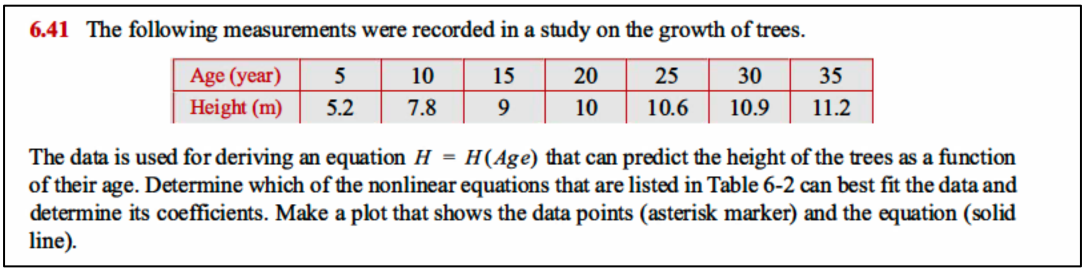
$$f(x) = 8.98 \times 10^{-4}x^4 - 0.085x^3 + 2.016x^2 + 10.468x - 23.212$$

We can then use this polynomial to estimate the power at a wind speed of 26mph:

$$f(26) = 8.98 \times 10^{-4}(26)^4 - 0.085(26)^3 + 2.016(26)^2 + 10.468(26) - 23.212$$

$$f(26) = 528.18\, W$$

Thus, using the Lagrange fourth order polynomial we can estimate that the power at a wind speed of 26mph is approximately 528.18W.



6.41 The following measurements were recorded in a study on the growth of trees.

Age (year)	5	10	15	20	25	30	35
Height (m)	5.2	7.8	9	10	10.6	10.9	11.2

The data is used for deriving an equation $H = H(Age)$ that can predict the height of the trees as a function of their age. Determine which of the nonlinear equations that are listed in Table 6-2 can best fit the data and determine its coefficients. Make a plot that shows the data points (asterisk marker) and the equation (solid line).

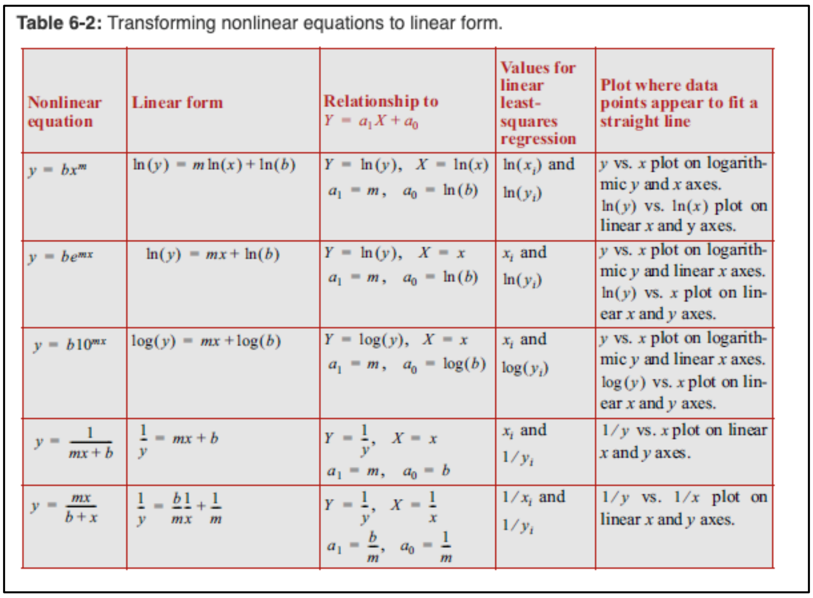
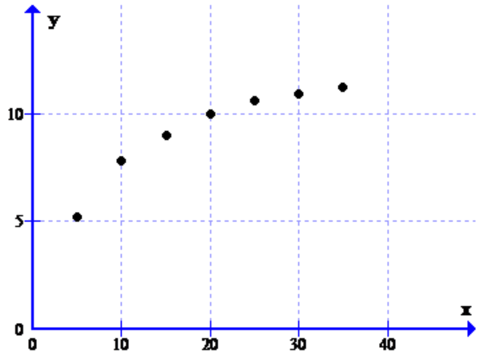


Table 6-2: Transforming nonlinear equations to linear form.

Nonlinear equation	Linear form	Relationship to $Y = a_1X + a_0$	Values for linear least-squares regression	Plot where data points appear to fit a straight line
$y = bx^m$	$\ln(y) = m\ln(x) + \ln(b)$	$Y = \ln(y),\ X = \ln(x)$ $a_1 = m,\ a_0 = \ln(b)$	$\ln(x_i)$ and $\ln(y_i)$	y vs. x plot on logarithmic y and x axes. $\ln(y)$ vs. $\ln(x)$ plot on linear x and y axes.
$y = be^{mx}$	$\ln(y) = mx + \ln(b)$	$Y = \ln(y),\ X = x$ $a_1 = m,\ a_0 = \ln(b)$	x_i and $\ln(y_i)$	y vs. x plot on logarithmic y and linear x axes. $\ln(y)$ vs. x plot on linear x and y axes.
$y = b10^{mx}$	$\log(y) = mx + \log(b)$	$Y = \log(y),\ X = x$ $a_1 = m,\ a_0 = \log(b)$	x_i and $\log(y_i)$	y vs. x plot on logarithmic y and linear x axes. $\log(y)$ vs. x plot on linear x and y axes.
$y = \frac{1}{mx+b}$	$\frac{1}{y} = mx + b$	$Y = \frac{1}{y},\ X = x$ $a_1 = m,\ a_0 = b$	x_i and $1/y_i$	$1/y$ vs. x plot on linear x and y axes.
$y = \frac{mx}{b+x}$	$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$	$Y = \frac{1}{y},\ X = \frac{1}{x}$ $a_1 = \frac{b}{m},\ a_0 = \frac{1}{m}$	$1/x_i$ and $1/y_i$	$1/y$ vs. $1/x$ plot on linear x and y axes.

In order to choose which function would be best suited to model the above data points, first plot x vs y:



The scatter plot shows that the graph is similar to the nonlinear equation from the table:

$$y = \frac{mx}{b + x}$$

The linear form of this equation is:

$$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$$

We can then consider this:

- $Y = \frac{1}{y}$
- $X = \frac{1}{x}$
- $M = \frac{b}{m}$
- $C = \frac{1}{m}$

And from that, the equation can be considered linear as:

$$Y = MX + C$$

Now, apply the least square regression method. First, calculate the value of S_x

$$S_x = \sum_{i=1}^{7} x_i$$

Since X = 1/x, we then have:

$$S_x = \frac{1}{5} + \cdots + \frac{1}{35}$$

Then, calculate the value of S_y

$$S_y = \sum_{i=1}^{7} y_i$$

Since Y = 1/y, we then have:

$$S_y = \frac{1}{5.2} + \cdots + \frac{1}{11.2}$$

Then, calculate the value of S_{xx}

$$S_{xx} = \sum_{i=1}^{7} {x_i}^2$$

$$S_{xx} = \left(\frac{1}{5}\right)^2 + \cdots + \left(\frac{1}{35}\right)^2$$

Then, calculate the value of S_{xy}

$$S_{xy} = \sum_{i=1}^{7} x_i y_i$$

$$S_{xy} = \left(\frac{1}{5}\right)\left(\frac{1}{5.2}\right) + \cdots + \left(\frac{1}{35}\right)\left(\frac{1}{11.2}\right)$$

Then, calculate the value of M

$$M = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x S_x}$$

$$\boldsymbol{M = 0.6025}$$

Then, calculate the value of C

$$C = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - S_x S_x}$$

$$\boldsymbol{C = 0.0706}$$

Now, using our original equations solve the value of m:

$$C = \frac{1}{m}$$

$$0.0706 = \frac{1}{m}$$

$$m = \frac{1}{0.0706}$$

$$\boldsymbol{m = 14.164}$$

We can also solve the value of b:

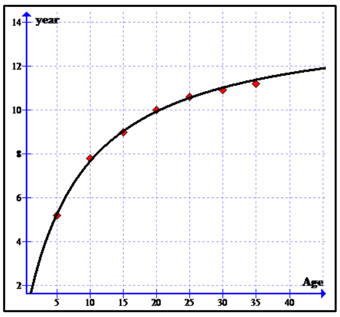
$$0.6025 = \frac{b}{14.164}$$

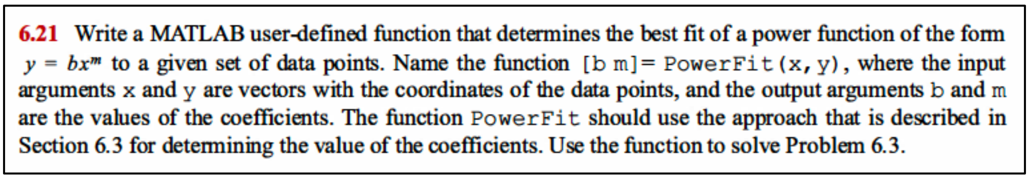
$$\boldsymbol{b = 8.534}$$

Thus, we can now write our equation as:

$$\boldsymbol{y = \frac{14.164x}{8.534 + x}}$$

By plotting the data points, and this given equation we can show that we have defined an accurate function to represent H=H(age):





6.21 Write a MATLAB user-defined function that determines the best fit of a power function of the form $y = bx^m$ to a given set of data points. Name the function `[b m]= PowerFit(x,y)`, where the input arguments `x` and `y` are vectors with the coordinates of the data points, and the output arguments `b` and `m` are the values of the coefficients. The function `PowerFit` should use the approach that is described in Section 6.3 for determining the value of the coefficients. Use the function to solve Problem 6.3.

The exponential function is as follows:

$$y = bx^m$$

This equation can be written in linear form as follows:

$$\ln(y) = \ln(bx^m)$$

$$\ln(y) = \ln(b) + \ln\,(x^m)$$

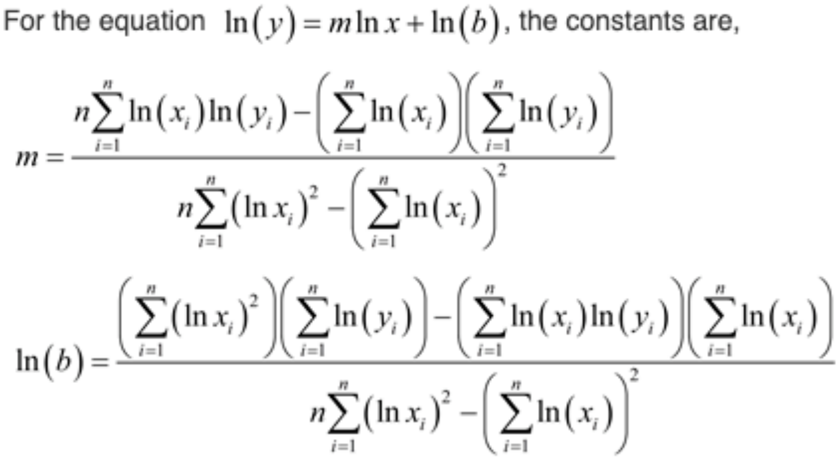
$$\ln(y) = \ln(b) + \text{m}\ln(x)$$

$$\ln(y) = \text{m}\ln(x) + \ln(b)$$

Now what we need to do is to determine the constants m and b. For a linear equation y = mx + b the coeffecients are:

$$m = \frac{n\sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n\sum_{i=1}^{n} {x_i}^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$b = \frac{(\sum_{i=1}^{n} {x_i}^2)(\sum_{i=1}^{n} y_i) - (\sum_{i=1}^{n} x_i y_i)(\sum_{i=1}^{n} x_i)}{n\sum_{i=1}^{n} {x_i}^2 - (\sum_{i=1}^{n} x_i)^2}$$

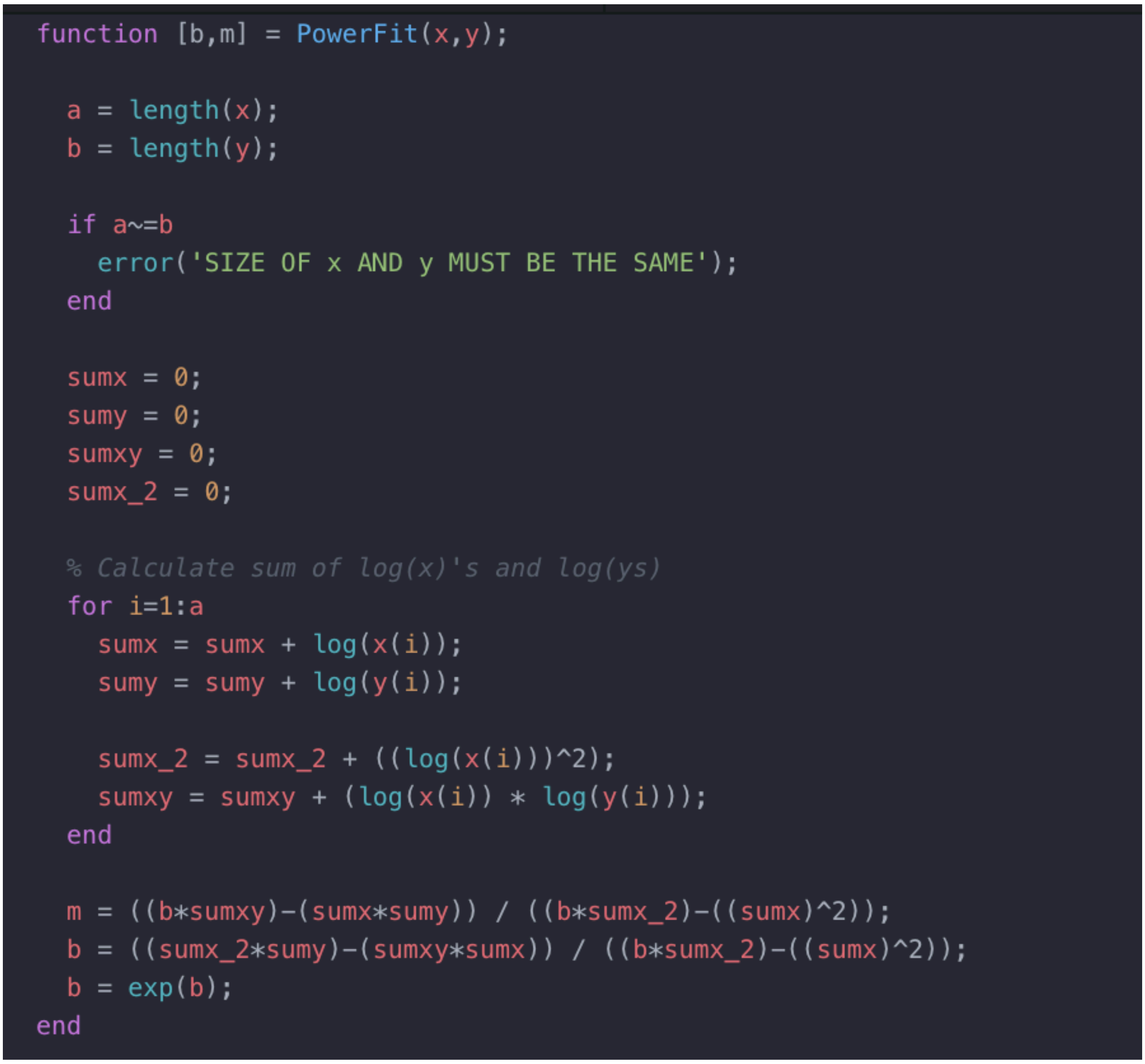


For the equation $\ln(y) = m\ln x + \ln(b)$, the constants are,

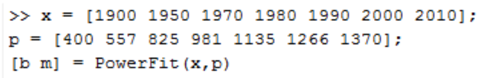
$$m = \frac{n\sum_{i=1}^{n}\ln(x_i)\ln(y_i) - \left(\sum_{i=1}^{n}\ln(x_i)\right)\left(\sum_{i=1}^{n}\ln(y_i)\right)}{n\sum_{i=1}^{n}(\ln x_i)^2 - \left(\sum_{i=1}^{n}\ln(x_i)\right)^2}$$

$$\ln(b) = \frac{\left(\sum_{i=1}^{n}(\ln x_i)^2\right)\left(\sum_{i=1}^{n}\ln(y_i)\right) - \left(\sum_{i=1}^{n}\ln(x_i)\ln(y_i)\right)\left(\sum_{i=1}^{n}\ln(x_i)\right)}{n\sum_{i=1}^{n}(\ln x_i)^2 - \left(\sum_{i=1}^{n}\ln(x_i)\right)^2}$$

The MATLAB function PowerFit can be defined as follows:

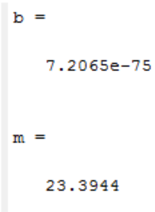


```
function [b,m] = PowerFit(x,y);

  a = length(x);
  b = length(y);

  if a~=b
    error('SIZE OF x AND y MUST BE THE SAME');
  end

  sumx = 0;
  sumy = 0;
  sumxy = 0;
  sumx_2 = 0;

  % Calculate sum of log(x)'s and log(ys)
  for i=1:a
    sumx = sumx + log(x(i));
    sumy = sumy + log(y(i));

    sumx_2 = sumx_2 + ((log(x(i)))^2);
    sumxy = sumxy + (log(x(i)) * log(y(i)));
  end

  m = ((b*sumxy)-(sumx*sumy)) / ((b*sumx_2)-((sumx)^2));
  b = ((sumx_2*sumy)-(sumxy*sumx)) / ((b*sumx_2)-((sumx)^2));
  b = exp(b);
end
```



```
>> x = [1900 1950 1970 1980 1990 2000 2010];
p = [400 557 825 981 1135 1266 1370];
[b m] = PowerFit(x,p)
```

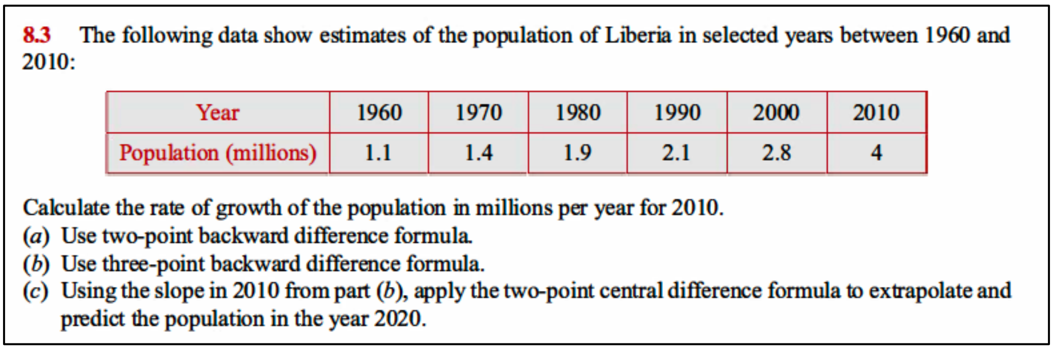
The following is the MATLAB output:



```
b =

    7.2065e-75


m =

    23.3944
```

Therefore, the values of the coefficients are,

$$\boxed{\begin{array}{l} m = 23.3944 \\ b = 7.2065 \times 10^{-75} \end{array}}$$



8.3 The following data show estimates of the population of Liberia in selected years between 1960 and 2010:

Year	1960	1970	1980	1990	2000	2010
Population (millions)	1.1	1.4	1.9	2.1	2.8	4

Calculate the rate of growth of the population in millions per year for 2010.
(*a*) Use two-point backward difference formula.
(*b*) Use three-point backward difference formula.
(*c*) Using the slope in 2010 from part (*b*), apply the two-point central difference formula to extrapolate and predict the population in the year 2020.

The formula for the two point backward difference formula for the first derivative is as follows:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h)$$

We can neglect O(h) here. Then, substitute 2010 for x_i and 2000 for x_{i-1} and 10 for h (distance between points):

$$f'(2010) = \frac{f(2010) - f(2000)}{10}$$

$$f'(2010) = \frac{4 - 2.8}{10}$$

$$\boldsymbol{f'(2010) = 0.12}$$

The formula for the three point backward difference formula for the first derivative is as follows:

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} + O(h^2)$$

We can neglect O(h) here. The, substitute 2010 for x_i, 2000 for x_{i-1}, 1990 for x_{i-2} and 10 for h (distance between points):

$$f'(2010) = \frac{f(1990) - 4f(2000) + 3f(2010)}{2(10)}$$

$$f'(2010) = \frac{2.1 - 4(2.8) + 3(4)}{2(10)}$$

$$\boldsymbol{f'(2010) = 0.145}$$

The formula for the two point central difference formula for the first derivative is as follows:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2)$$

We can neglect O(h) here. The, substitute 2010 for x_i, for x_{i-1}, 1990, 10 for h (distance between points) and we will solve for x_{i+1} :

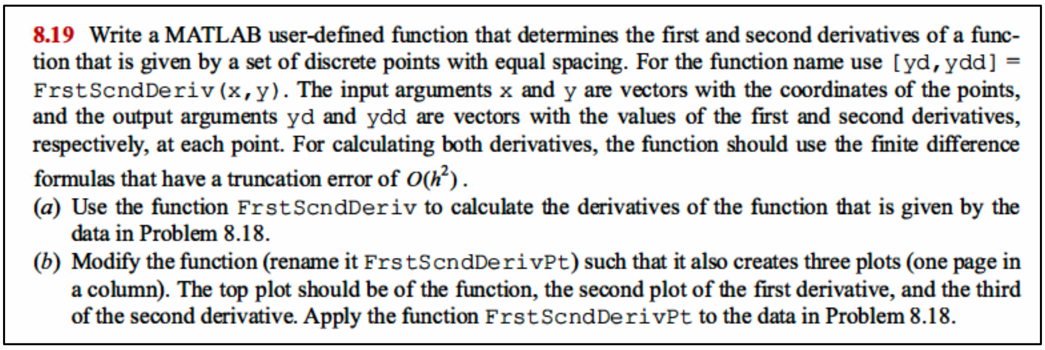
$$f'(2010) = \frac{f(2020) - f(1990)}{2(10)}$$

$$0.145 = \frac{f(2020) - 2.1}{20}$$

$$f(2020) = 20(0.145) + 2.8$$

$$f(2020) = 5.7$$

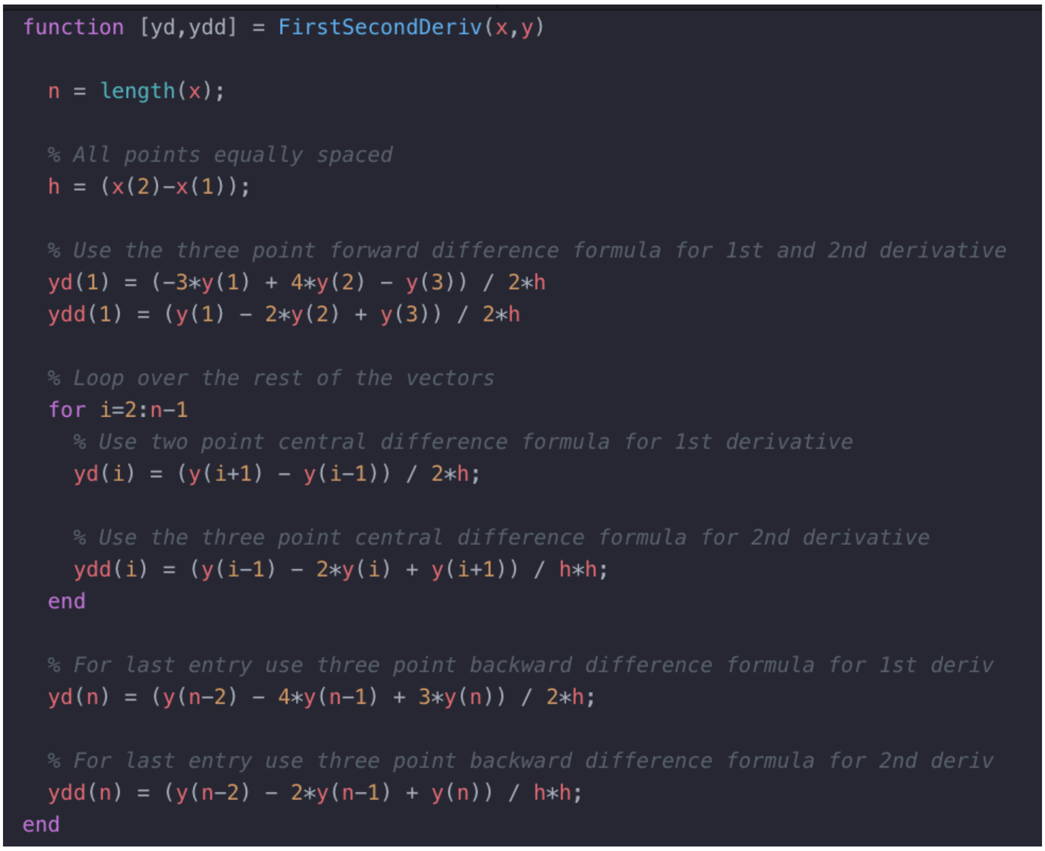
Therefore, our prediction for the population of Liberia in 2020 is approximately 5.7 million.



8.19 Write a MATLAB user-defined function that determines the first and second derivatives of a function that is given by a set of discrete points with equal spacing. For the function name use `[yd,ydd] = FrstScndDeriv(x,y)`. The input arguments `x` and `y` are vectors with the coordinates of the points, and the output arguments `yd` and `ydd` are vectors with the values of the first and second derivatives, respectively, at each point. For calculating both derivatives, the function should use the finite difference formulas that have a truncation error of $O(h^2)$.

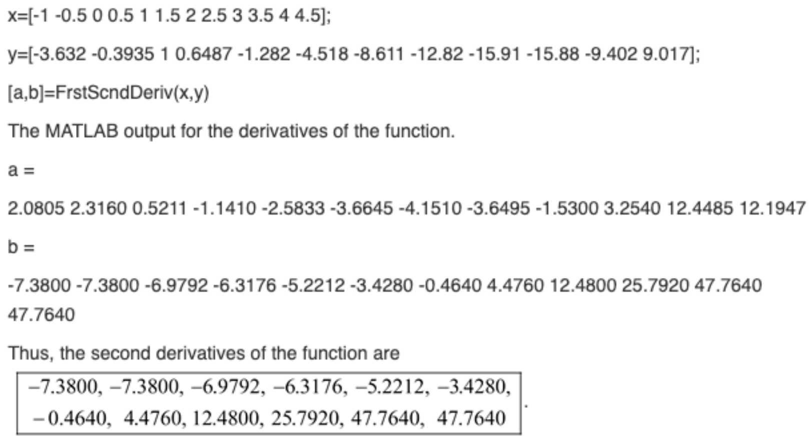
(*a*) Use the function `FrstScndDeriv` to calculate the derivatives of the function that is given by the data in Problem 8.18.

(*b*) Modify the function (rename it `FrstScndDerivPt`) such that it also creates three plots (one page in a column). The top plot should be of the function, the second plot of the first derivative, and the third of the second derivative. Apply the function `FrstScndDerivPt` to the data in Problem 8.18.



```
function [yd,ydd] = FirstSecondDeriv(x,y)

  n = length(x);

  % All points equally spaced
  h = (x(2)-x(1));

  % Use the three point forward difference formula for 1st and 2nd derivative
  yd(1) = (-3*y(1) + 4*y(2) - y(3)) / 2*h
  ydd(1) = (y(1) - 2*y(2) + y(3)) / 2*h

  % Loop over the rest of the vectors
  for i=2:n-1
    % Use two point central difference formula for 1st derivative
    yd(i) = (y(i+1) - y(i-1)) / 2*h;

    % Use the three point central difference formula for 2nd derivative
    ydd(i) = (y(i-1) - 2*y(i) + y(i+1)) / h*h;
  end

  % For last entry use three point backward difference formula for 1st deriv
  yd(n) = (y(n-2) - 4*y(n-1) + 3*y(n)) / 2*h;

  % For last entry use three point backward difference formula for 2nd deriv
  ydd(n) = (y(n-2) - 2*y(n-1) + y(n)) / h*h;
end
```

To then test this code to get the first and second derivatives do the following:



```
x=[-1 -0.5 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5];

y=[-3.632 -0.3935 1 0.6487 -1.282 -4.518 -8.611 -12.82 -15.91 -15.88 -9.402 9.017];

[a,b]=FrstScndDeriv(x,y)
```

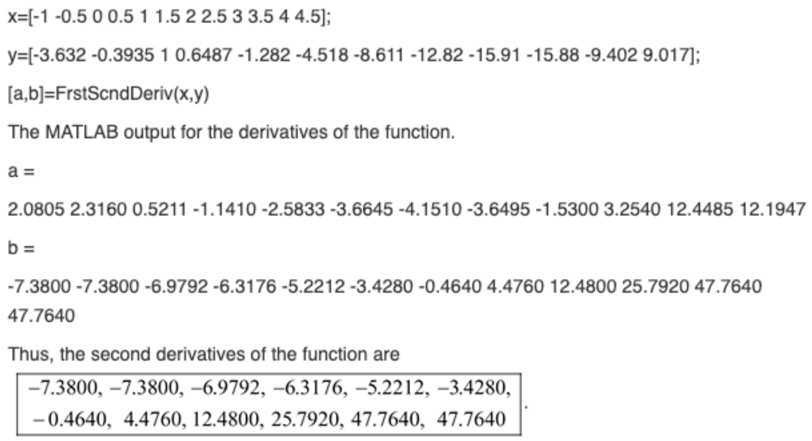
The MATLAB output for the derivatives of the function.

```
a =

2.0805 2.3160 0.5211 -1.1410 -2.5833 -3.6645 -4.1510 -3.6495 -1.5300 3.2540 12.4485 12.1947

b =

-7.3800 -7.3800 -6.9792 -6.3176 -5.2212 -3.4280 -0.4640 4.4760 12.4800 25.7920 47.7640
47.7640
```

Thus, the second derivatives of the function are

$$\boxed{\begin{array}{l}-7.3800,\ -7.3800,\ -6.9792,\ -6.3176,\ -5.2212,\ -3.4280,\\ -0.4640,\ \ 4.4760,\ 12.4800,\ 25.7920,\ 47.7640,\ \ 47.7640\end{array}}.$$

To then graph the results we would do the following:



```
x=[-1 -0.5 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5];
y=[-3.632 -0.3935 1 0.6487 -1.282 -4.518 -8.611 -12.82 -15.91 -15.88 -9.402 9.017];
[a,b]=FrstScndDeriv(x,y)
```

The MATLAB output for the derivatives of the function.

```
a =
2.0805 2.3160 0.5211 -1.1410 -2.5833 -3.6645 -4.1510 -3.6495 -1.5300 3.2540 12.4485 12.1947
b =
-7.3800 -7.3800 -6.9792 -6.3176 -5.2212 -3.4280 -0.4640 4.4760 12.4800 25.7920 47.7640
47.7640
```

Thus, the second derivatives of the function are

$$\boxed{\begin{array}{l}-7.3800, -7.3800, -6.9792, -6.3176, -5.2212, -3.4280, \\ -0.4640,\ 4.4760, 12.4800, 25.7920, 47.7640,\ 47.7640\end{array}}.$$

This then produces the plots:

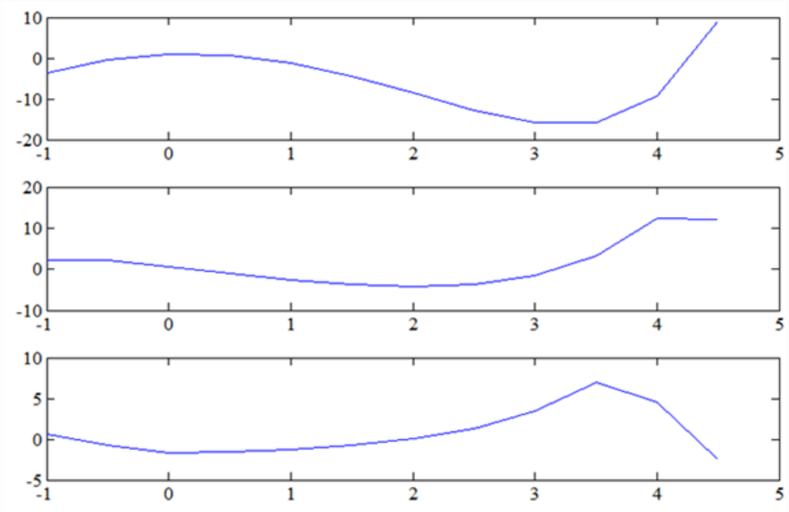
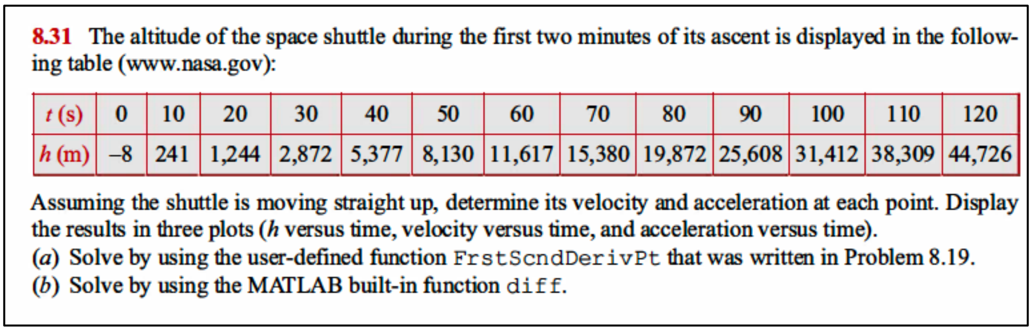


Figure 1



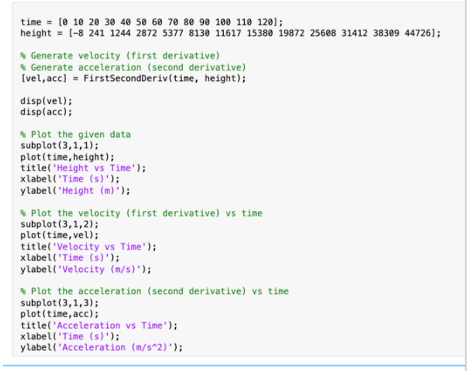
8.31 The altitude of the space shuttle during the first two minutes of its ascent is displayed in the following table (www.nasa.gov):

t (s)	0	10	20	30	40	50	60	70	80	90	100	110	120
h (m)	–8	241	1,244	2,872	5,377	8,130	11,617	15,380	19,872	25,608	31,412	38,309	44,726

Assuming the shuttle is moving straight up, determine its velocity and acceleration at each point. Display the results in three plots (*h* versus time, velocity versus time, and acceleration versus time).

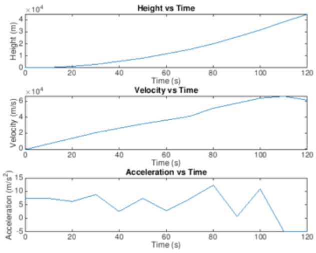
(*a*) Solve by using the user-defined function `FrstScndDerivPt` that was written in Problem 8.19.

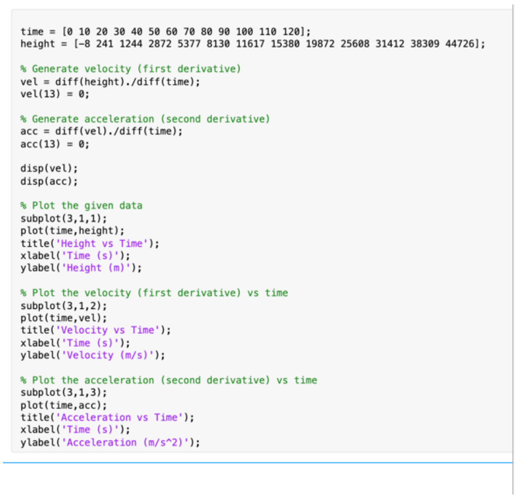
(*b*) Solve by using the MATLAB built-in function `diff`.



```
time = [0 10 20 30 40 50 60 70 80 90 100 110 120];
height = [-8 241 1244 2872 5377 8130 11617 15380 19872 25608 31412 38309 44726];

% Generate velocity (first derivative)
% Generate acceleration (second derivative)
[vel,acc] = FirstSecondDeriv(time, height);

disp(vel);
disp(acc);

% Plot the given data
subplot(3,1,1);
plot(time,height);
title('Height vs Time');
xlabel('Time (s)');
ylabel('Height (m)');

% Plot the velocity (first derivative) vs time
subplot(3,1,2);
plot(time,vel);
title('Velocity vs Time');
xlabel('Time (s)');
ylabel('Velocity (m/s)');

% Plot the acceleration (second derivative) vs time
subplot(3,1,3);
plot(time,acc);
title('Acceleration vs Time');
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');
```

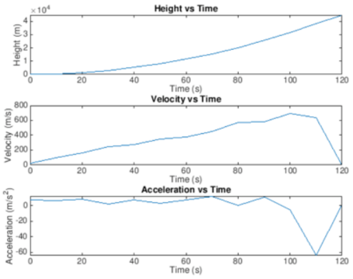
```
     -1280        6260       13155       20665       26290       31200       36250       41275       51140       57700       63505       66570       61770
  7.5400    7.5400    6.2500    8.7700    2.4800    7.3400    2.7600    7.2900   12.4400    0.6800   10.9300   -4.8000   -4.8000
```





```
time = [0 10 20 30 40 50 60 70 80 90 100 110 120];
height = [-8 241 1244 2872 5377 8130 11617 15380 19872 25608 31412 38309 44726];

% Generate velocity (first derivative)
vel = diff(height)./diff(time);
vel(13) = 0;

% Generate acceleration (second derivative)
acc = diff(vel)./diff(time);
acc(13) = 0;

disp(vel);
disp(acc);

% Plot the given data
subplot(3,1,1);
plot(time,height);
title('Height vs Time');
xlabel('Time (s)');
ylabel('Height (m)');

% Plot the velocity (first derivative) vs time
subplot(3,1,2);
plot(time,vel);
title('Velocity vs Time');
xlabel('Time (s)');
ylabel('Velocity (m/s)');

% Plot the acceleration (second derivative) vs time
subplot(3,1,3);
plot(time,acc);
title('Acceleration vs Time');
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');
```

```
  24.9000  100.3000  162.8000  250.5000  275.3000  348.7000  376.3000  449.2000  573.6000  580.4000  689.7000  641.7000         0
   7.5400    6.2500    8.7700    2.4800    7.3400    2.7600    7.2900   12.4400    0.6800   10.9300   -4.8000  -64.1700         0
```


Height vs Time
×10⁴
Height (m)
Time (s)
Velocity vs Time
Velocity (m/s)
Time (s)
Acceleration vs Time
Acceleration (m/s²)
Time (s)