Extra problem number 5: The speed of some object was measured each six hours. The data are collected in the following table.

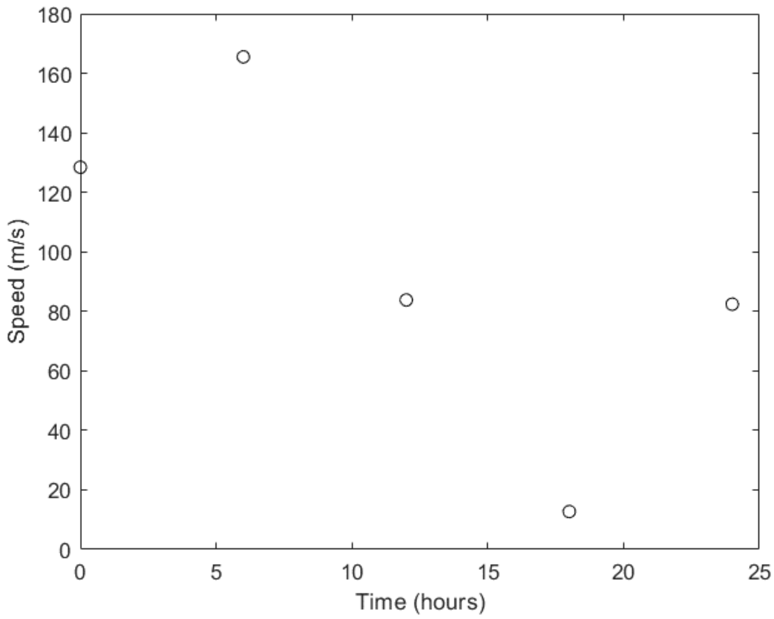
Time (hours)	0	6	12	18	24
Speed (m/s)	128.4312	165.5232	83.8152	12.7092	82.3992

When (what time) the speed was at its maximum and minimum? What are its maximum and minimum values?

Hints: Lagrange or Newton interpolation, some technique for finding the extrema (e.g. the golden-section search). Matlab Prepare a report, describe the methods, the code, and the results.

Solution:

Firstly, let see how our data look like on a plot.



We have only a few points, so we have to make an interpolation. Interpolation is an estimation of the values of a quantity between the individual data points. After interpolation we can minimize a function, find its root, its maximum and minimum values, etc. Let's make Lagrange interpolation.

Lagrange Interpolating Polynomial formula:

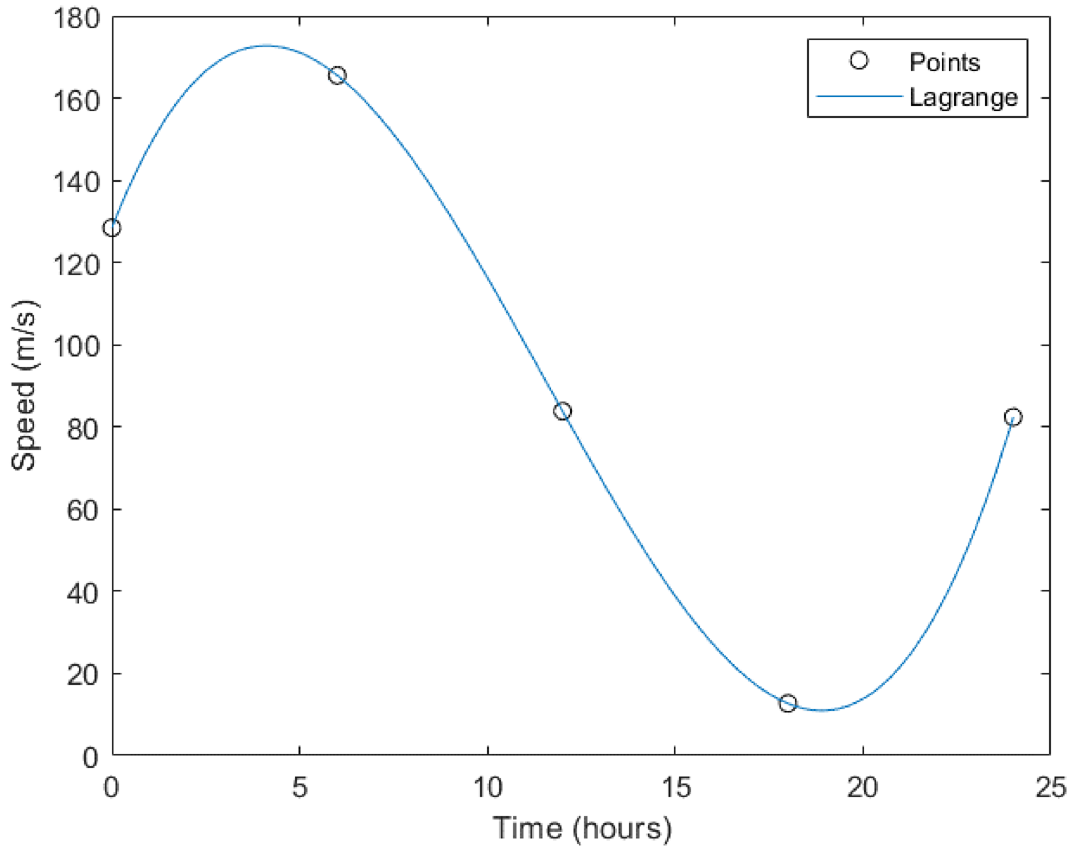
$$\phi(x) = \prod_{j=1, i \neq j}^{n} \frac{(x - x_j)}{(x_i - x_j)}$$

for i = 1, 2 . . . n + 1.

$$F(x) = \sum_{i=1}^{n+1} y_i \phi_i(x)$$

We implemented it in Matlab. You can see code at the end of this report.

Now, we can show results of our interpolation on the plot.



It's look pretty good.

We made our own algorithm to obtain the maximum and minimum of the function. It is based on the three parts method. On the plot we can see that a maximum is between 0 and 12 and the minimum between 12 and 24. So, we try to find these points in the intervals.

The way in which our algorithm works:

1. We have points:
    a) at the beginning (denoted by a)
    b) at the end (denoted by b)
    c) at the $\frac{2}{3}a + \frac{1}{3}b$ (denoted by t1)
    d) at the $\frac{1}{3}a + \frac{2}{3}b$ (denoted by t2)
2. We calculate t1, t2 and values of interpolation function at these points (f1 and f2).
3. We check which value of function is bigger (finding the max)/smaller (finding the min).
    a) If the the first value is bigger (finding the max)/smaller (finding the min), now our b = t2.
    b) If the the second value is bigger (finding the max)/smaller (finding the min), now our a = t1.
4. If the difference between points a and b is smaller than 10^{-5}, we return to point 2. (We have chosen the value 10^{-5}, because it is enough to get accurate result.)

In this way we obtained that the maximum is at 4.1111 hour ($\approx 4\ hours\ and\ 7\ minutes$) and the velocity is equal to 172.7541 m/s. The minimum is at 18.8960 hour ($\approx 18\ hours\ and\ 54\ minutes$) and the velocity at this point is 10.9949 m/s.

	Time (hour)	Velocity (m/s)
Minimum	18.8960	10.9949
Maximum	4.1111	172.7541

We calculated the result again using the Newton's interpolation formula to check if we have done any errors in the Lagrange interpolation. The newton's interpolation can be calculated by calculating the Newton's divided differences method which will be the coefficients for the Newton's polynomial.

First we calculate the divided difference for the data from the:

The divided differences for a function $f[x]$ are defined as follows:

$$f[x_{i-1}, x_i] = \frac{f[x_i] - f[x_{i-1}]}{x_i - x_{i-1}}$$

$$f[x_{i-2}, x_{i-1}, x_i] = \frac{f[x_{i-1}, x_i] - f[x_{i-2}, x_{i-1}]}{x_i - x_{i-2}}$$

$$f[x_{i-3}, x_{i-2}, x_{i-1}, x_i] = \frac{f[x_{i-2}, x_{i-1}, x_i] - f[x_{i-3}, x_{i-2}, x_{i-1}]}{x_i - x_{i-3}}$$

$$f[x_{i-j}, x_{i-j+1}, \ldots, x_i] = \frac{f[x_{i-j+1}, \ldots, x_i] - f[x_{i-j}, \ldots, x_{i-1}]}{x_i - x_{i-j}}$$

The coefficient of the Newton polynomial is the top element in the column of the i-th divided differences.

```
d =

    6.1820   -1.6500    0.0998    0.0000
  -13.6180    0.1472    0.1005         0
  -11.8510    1.9555         0         0
   11.6150         0         0         0
```

After obtaining the table we can now calculate the Newton's polynomial to check the result with the results obtained by the Lagrange interpolation.

The formula for the Newton's polynomial is as follows

$$N(x) = [y_0] + [y_0, y_1](x - x_0) + \cdots + [y_0, \ldots, y_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}).$$

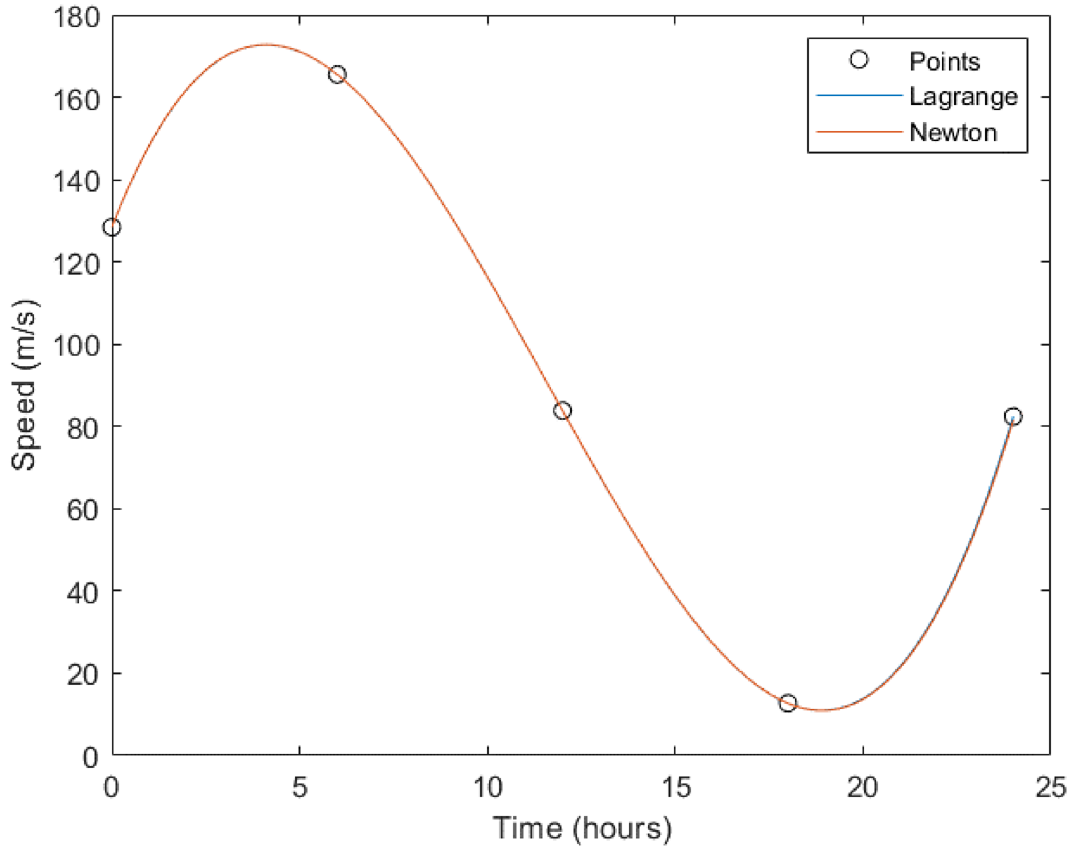
Where the expressions in the [] are the coefficients calculated from the divided difference.

The polynomial based on this data is:

W(x) = $0.0998x^3 - 3.4320x^2 + 23.2132x + 128.4312$

We implemented it in Matlab. You can see code at the end of this report.

Now we can make a plot based on the polynomial:



The last plot contains the plot of data, the Lagrange and Newton's interpolations plots. As we can see, the two methods have given us almost the same result. In fact the average difference between values is equal to 0.0630.

We used our algorithm once again and we obtained that the maximum is at 4.1088 hour ($\approx 4\ hours\ and\ 7\ minutes$) and the velocity is equal to 172.7758 m/s. The minimum is at 18.9081 hour ($\approx 18\ hours\ and\ 55\ minutes$) and the velocity at this point is 10.9562 m/s.

	Time (hour)	Velocity (m/s)
Minimum	18.9081	10.9562
Maximum	4.1088	172.7758

Conclusions:

The goal of this task was to obtain the time, when the speed was at its maximum and minimum. To find the time when the speed was in its min/max we had to use some interpolation methods which enabled us to predict values of the function between points.

For that we had to implement the Lagrange interpolation, Newton's interpolation and our own algorithm to find the max and min of the functions.

Results:

	Lagrange interpolation		Newton's interpolation	
	Time (hour)	Velocity (m/s)	Time (hour)	Velocity (m/s)
Minimum	18.8960	10.9949	18.9081	10.9562
Maximum	4.1111	172.7541	4.1088	172.7758

As we can see, the results obtained by the two interpolations are minimally different, the average difference between velocity values at the same time is equal to 0.0630. So, we assume that the difference is negligibly small. It confirms that we did not make a mistake and our code works well.

Matlab code:

```
close all;
clc;
clear all;
%our data (vectors)
x = [0 6 12 18 24];
y = [128.4312 165.5232 83.8152 12.7092 82.3992];
plot(x,y,'ko')
xlabel('Time (hours)')
ylabel('Speed (m/s)')

%degree of polynomial
n = length(x)-1;
%vectors i_x i and i_y wiil contain results of the Lagrange interploation
i_x = 0:0.1:24;
q = length(i_x);
for i=1: length(i_x)
    i_y(i) = 0;
end

%Lagrange interpolation
for i=1: length(i_x)
    i_y(i) = Lagrange(n,x,y,i_x(i));
end
plot(x,y,'ko')
hold on
plot(i_x,i_y)
xlabel('Time (hours)')
ylabel('Speed (m/s)')
legend('Points','Lagrange');
hold off

%determining the minimum
a = 12;
b = 24;
while 1
    t1 = (2/3)*a + (1/3)*b;
    t2 = (1/3)*a + (2/3)*b;
    f1 = Lagrange(n,x,y,t1);
```

```
    f2 = Lagrange(n,x,y,t2);
    if f1 <= f2
        b = t2;
    else
        a = t1;
    end
    err = b - a;
    if err < 0.00001
        break
    end
end
a
f1

%determining the maximum
a = 0;
b = 12;
while 1
    t1 = (2/3)*a + (1/3)*b;
    t2 = (1/3)*a + (2/3)*b;
    f1 = Lagrange(n,x,y,t1);
    f2 = Lagrange(n,x,y,t2);
    if f1 >= f2
        b = t2;
    else
        a = t1;
    end
    err = b - a;
    if err < 0.00001
        break
    end
end
a
f1
%---------------------------------------------------------------------------
function [sm] = Lagrange(n,x,y,z)
    sm = 0;
    for i=1: (n+1)
        p = 1;
        for j=1: (n+1)
            if j ~= i   %we must not divide by 0
                p = p * ( z - x(j) )/( x(i) - x(j) );
            end
        end
        sm = sm + y(i) *p;
    end
end
%---------------------------------------------------------------------------
% the interval at which we will plot our data
xx = 0:0.1:24;
%degree of the polynomial
n = length(x);

% d will be the table of the divided difference
% calculating the first iteration for the divided difference
for k = 1 : n - 1
   d(k, 1) = (y(k+1) - y(k))/(x(k+1) - x(k));
end
% Calculating the rest of the coefficients for the difference dividend
for j =  2 : n - 1
   for k = 1 : n - j
      d(k, j) = (d(k+1, j - 1) - d(k, j - 1))/(x(k+j) - x(k));
   end
end

d
% The formula for the Newtons polynomial
```

```
fun = y(1) + d(1,1)*(xx-x(1)) + d(1,2)*(xx.^2- x(2)*xx) + d(1,3)*(xx.^3-
18*xx.^2+72*xx);

plot(x,y,'ko') %  data at the beginning
hold on
plot(i_x,i_y) % data after the Lagrange interpolation
plot(xx,fun) % data after the Newton interpolation
xlabel('Time (hours)')
ylabel('Speed (m/s)')
legend('Points','Lagrange','Newton');
hold off

%determining the minimum
a = 12;
b = 24;
while 1
    t1 = (2/3)*a + (1/3)*b;
    t2 = (1/3)*a + (2/3)*b;
    f1 = y(1) + d(1,1)*(t1-x(1)) + d(1,2)*(t1.^2- x(2)*t1) + d(1,3)*(t1.^3-
18*t1.^2+72*t1);
    f2 = y(1) + d(1,1)*(t2-x(1)) + d(1,2)*(t2.^2- x(2)*t2) + d(1,3)*(t2.^3-
18*t2.^2+72*t2);
    if f1 <= f2
        b = t2;
    else
        a = t1;
    end
    err = b - a;
    if err < 0.00001
        break
    end
end
a
f1

%determining the maximum
a = 0;
b = 12;
while 1
    t1 = (2/3)*a + (1/3)*b;
    t2 = (1/3)*a + (2/3)*b;
    f1 = y(1) + d(1,1)*(t1-x(1)) + d(1,2)*(t1.^2- x(2)*t1) + d(1,3)*(t1.^3-
18*t1.^2+72*t1);
    f2 = y(1) + d(1,1)*(t2-x(1)) + d(1,2)*(t2.^2- x(2)*t2) + d(1,3)*(t2.^3-
18*t2.^2+72*t2);
    if f1 >= f2
        b = t2;
    else
        a = t1;
    end
    err = b - a;
    if err < 0.00001
        break
    end
end
a
f1

%the average difference between i_y and fun
sum(i_y-fun)/length(i_y)
```
