PROJECT REPORT

TIME SERIES MODELING USING FOURIER BASIS FUNCTIONS

Semester **I**, Year **2022-2023**

Project made by: *Bejenariu Adina*
Ferenț Sarah
Marin Călin

Group Number: *30331/1*

Data set: *"product_20.mat"*

Contents

Chapter 1. Introduction

For the first part of the project, we are given a time series of the products sold by a store in each month. For this event we have to find an approximation of the quantities based on each month.

The data is given as a MATLAB data file which contains the *time* vector (the *k* index of the month) and the *y* vector (the actual quantities *y(k)* that were sold in the kth month) . We will use 80% of the data for identifying the other 20% for validation. Firstly, we want to see how the sales are changing over time, so we plot them with respect to the time. We can observe that the data is changing drastically from an index to another, so we are expecting some difficulties with the approximation.

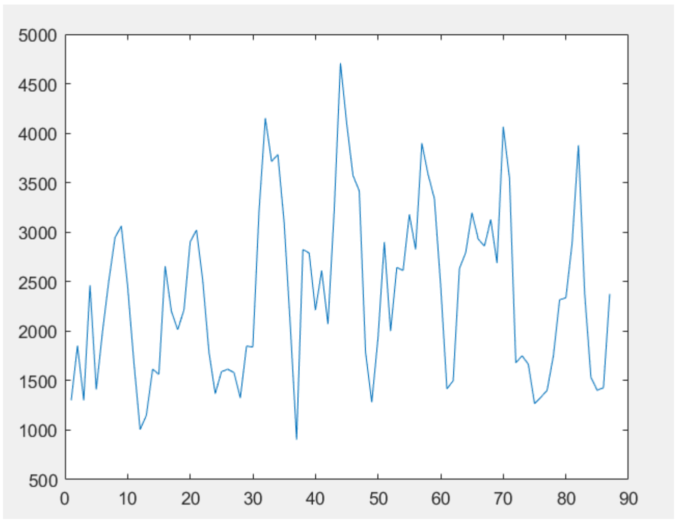


Figure 1*. The initial data set*

In order to estimate this time series, we are going to use linear regression with a linear trend and a configurable Fourier basis (1).

$$\hat{y}(k) = t_0 + t_1 k + \sum_{i=1}^{m} [a_i cos(\frac{2\pi ik}{P}) + b_i sin(\frac{2\pi ik}{P})] \quad (1)$$

For each m and every $y(k)$ from the identification data set we will have multiple equations with the unknowns t_0, t_1, a_i and b_i. With these equations we will create an system of equations for each m . In the matrix ϕ we will have the regressors 1, k and the sines and cosines of $\frac{2\pi ik}{P}$ (these values are known and computable for each index *i*) and in the matrix θ we will have the unknown parameters mentioned above. By doing these procedures, our system of equations will be simplified to one equation (2).

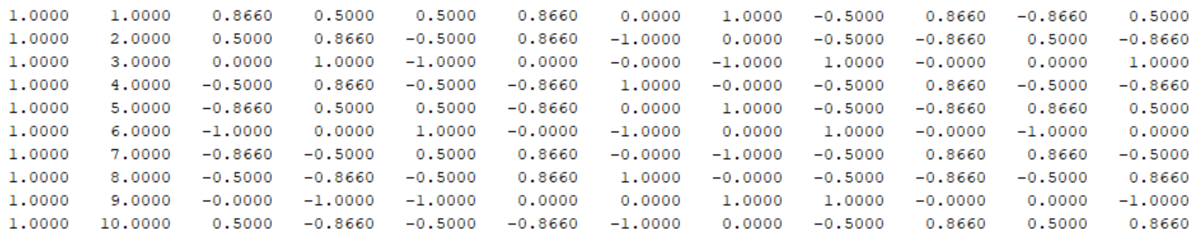
$$Y = \phi\theta \quad (2)$$

We are required to create a MATLAB script that can compute the two matrices for each m and to compare it to the validation data. Speaking straight to the point, we need to compute $\widehat{Y}$for each m and find its Mean Squared Error. Our purpose is to find what is the value of m that gives the best approximation of the data set.

Chapter 2. Linear Regression in MATLAB

Before we proceed to the mathematical part of the problem, it is needed to split the data for a better understanding. The data was split into identification data which consisted of 80% of the data given (87 elements out of the 109 given) and the rest of 20% being the validation data (22 elements out of the 109 given). Parameter P is imposed to be 12 because the periodicity of the behavior of the data set is roughly yearly (see *Figure 1.*).

The MATLAB procedure begins with a loop that for each m will find ϕ, θ and $\widehat{y}$ in order to get the most accurate approximation for the data set given by computing the MSE. We chose this method so that the code will automatically give us all the values of MSE. We begin by composing ϕ on the identification time. For example, for $m = 5$ the first 10 rows are the following: (*Figure 2.*)



```
 1.0000    1.0000    0.8660    0.5000    0.5000    0.8660    0.0000    1.0000   -0.5000    0.8660   -0.8660    0.5000
 1.0000    2.0000    0.5000    0.8660   -0.5000    0.8660   -1.0000    0.0000   -0.5000   -0.8660    0.5000   -0.8660
 1.0000    3.0000    0.0000    1.0000   -1.0000    0.0000   -0.0000   -1.0000    1.0000   -0.0000    0.0000    1.0000
 1.0000    4.0000   -0.5000    0.8660   -0.5000   -0.8660    1.0000   -0.0000   -0.5000    0.8660   -0.5000   -0.8660
 1.0000    5.0000   -0.8660    0.5000    0.5000   -0.8660    0.0000    1.0000   -0.5000   -0.8660    0.8660    0.5000
 1.0000    6.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000    0.0000    1.0000   -0.0000   -1.0000    0.0000
 1.0000    7.0000   -0.8660   -0.5000    0.5000    0.8660   -0.0000   -1.0000   -0.5000    0.8660    0.8660   -0.5000
 1.0000    8.0000   -0.5000   -0.8660   -0.5000    0.8660    1.0000   -0.0000   -0.5000   -0.8660   -0.5000    0.8660
 1.0000    9.0000   -0.0000   -1.0000   -1.0000    0.0000    0.0000    1.0000    1.0000   -0.0000    0.0000   -1.0000
 1.0000   10.0000    0.5000   -0.8660   -0.5000   -0.8660   -1.0000    0.0000   -0.5000    0.8660    0.5000    0.8660
```

***Figure 2**. The first 10 rows of* ϕ *when* $m = 5$.

After finding ϕ, we can proceed to find θ from the equation (2) from above. In order to find the matrix we would have to multiply to the left with the transpose and the inverse of ϕ so that the number of lines and columns are favorable for the operations between matrices. However, MATLAB has a simpler solution - by using left matrix division '\' it makes all these computations for us.

$$\theta = \phi \backslash Y \quad (3)$$

In the matrix below (*Figure3.*) we can see the parameters of our regression.

Moving forward it is necessary to find $\widehat{Y}$ for each m and find its Mean Squared Error. $\widehat{Y}$ will be computed multiplying ϕ on the whole time domain and θ.

Finally, for each *m* we will plot with green the initial data set and with red its approximation. (See *Figures 3, 4* and *5*)

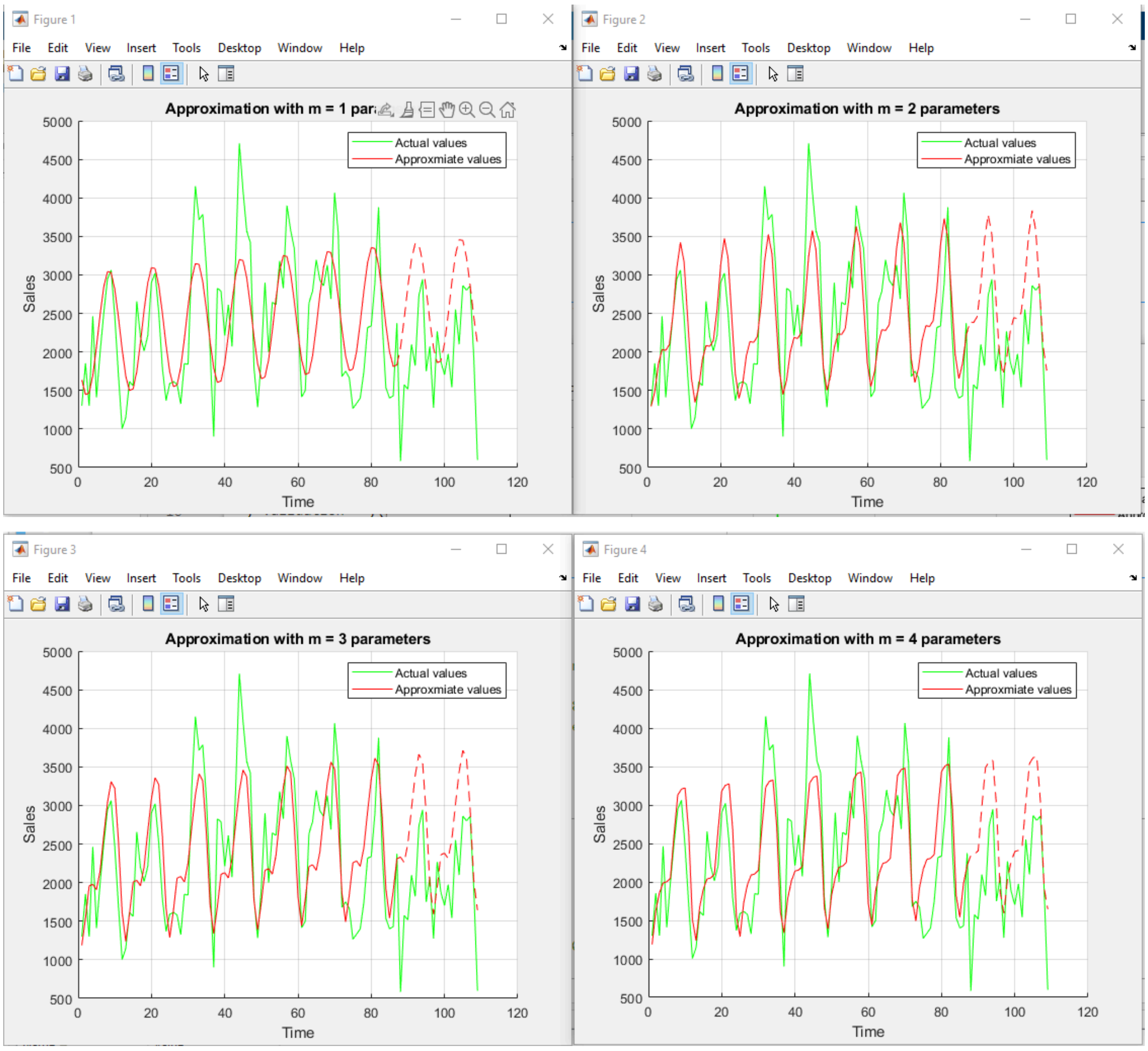


Figure 3. *The approximations for m = 1:4*

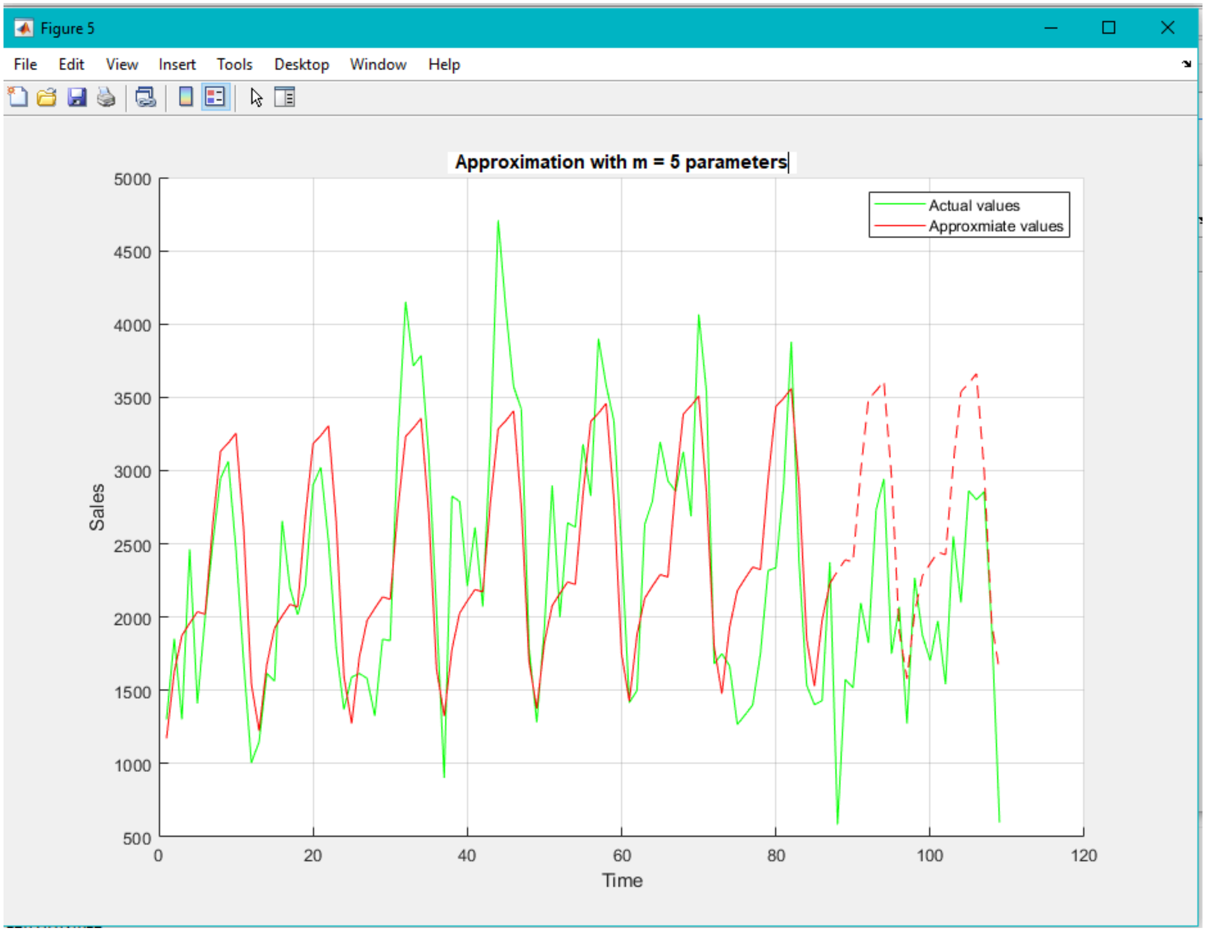


Figure 4. *The approximations for m = 5*

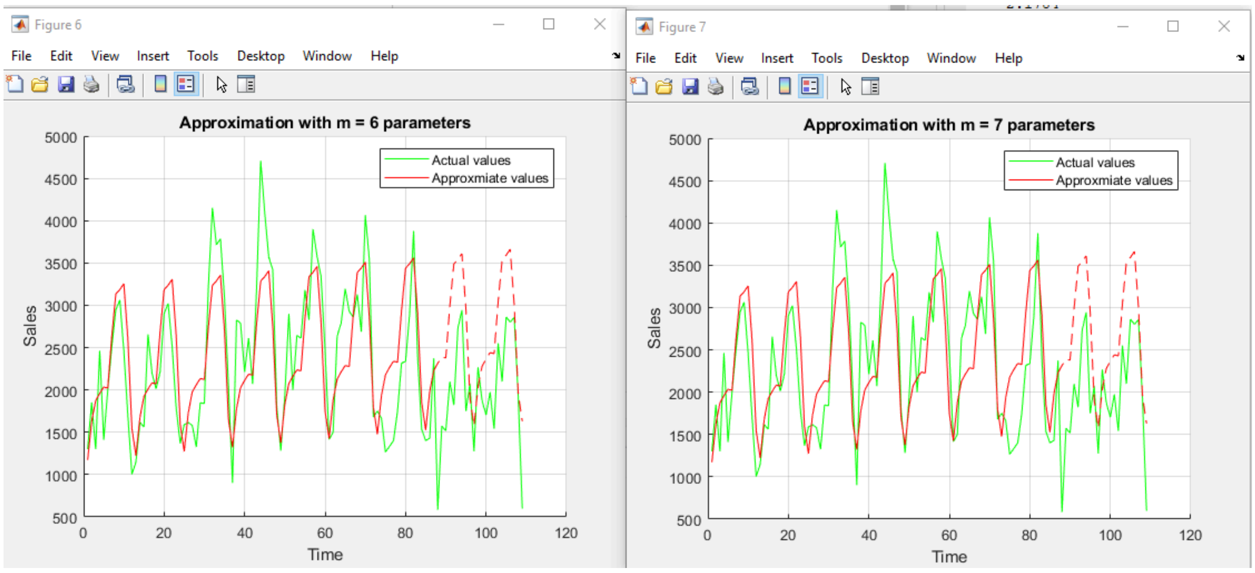


Figure 5. *The approximations for m = 6:7*

As you can see the approximation gets more and more accurate from $m = 1$ to $m = 5$ (*Figure 3*), reaching its best value at $m = 5$ (*Figure 4*). After that we observe that the approximation does not change anymore (*Figure 5*), that is because overfitting takes place. It happens when data gets more and more accurate until a point where it stagnates despite the fact that the precision is increasing. This is exactly what happens for m = 6 and m = 7, it cannot get more accurate than this. So it is clear that for our data set the most ideal m to get the most accurate approximation si m = 5.

Lastly, we had to find the Mean Squared Error of the data set for each m. This will calculate the difference between the average squares of the estimated value and the actual value.

$$MSE = \frac{1}{N}\sum_{k=1}^{N} (\widehat{Y}(k) - Y(k)) \quad (3)$$

For each m the Mean Squared Error is calculated for the whole dataset, for the identification dataset and for the validation dataset.

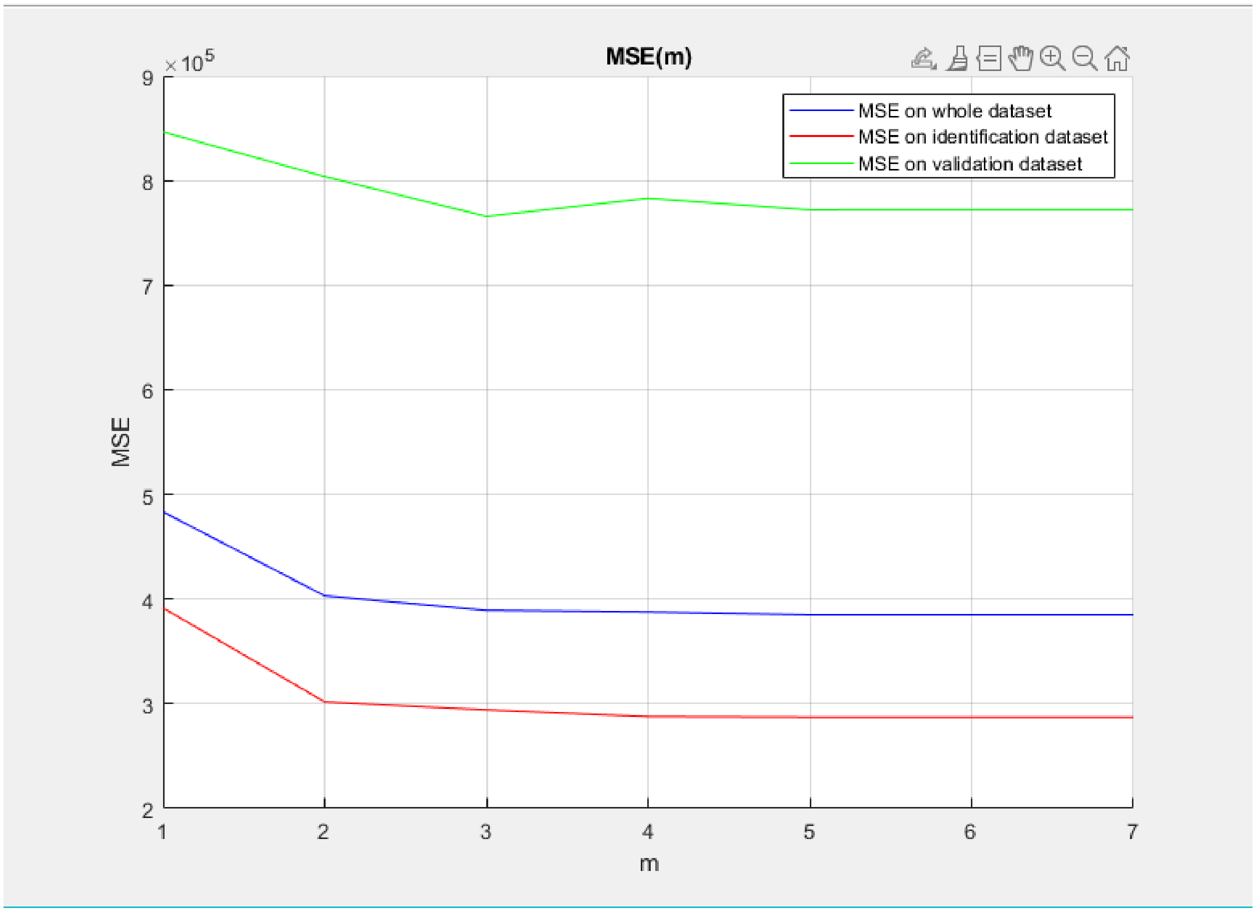
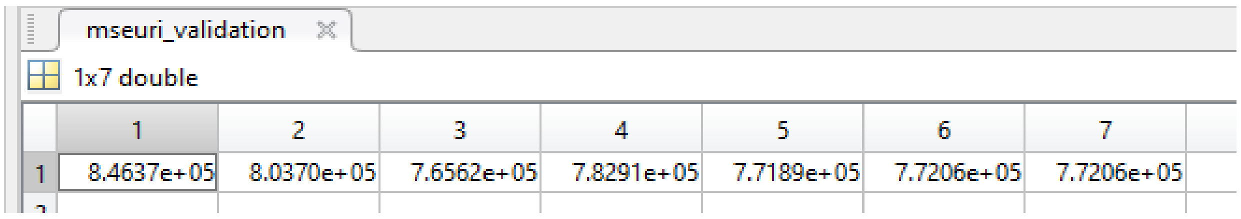


Figure 6*. MSE*

Furthermore, we can verify that 5 is the best value for m also from the Mean Squared Error, because when m = 5 the MSE is the lowest.



	1	2	3	4	5	6	7
1	8.4637e+05	8.0370e+05	7.6562e+05	7.8291e+05	7.7189e+05	7.7206e+05	7.7206e+05

***Figure** 7. MSE on the validation dataset*

Despite being this big, this Mean Squared Error is still correct because of two major facts: the data set given contained bigger values and the fluctuations on the data set are really abrupt which means they cannot be approximated perfectly in any periodicity.

Chapter 3. Conclusions

Given the results above, it can be concluded that the model resulted by using Fourier basis functions to estimate a time series does indeed produce a system that somewhat predicts general trends, managing to take into account seasonalities (generated in the store example by the passing of the seasons and fluctuations in demand). It falls short, however, of being able to handle anomalies that exceed simple seasonality, such as sudden spikes or crashes in sales. Depending on the real-life process that is being modeled, this shortcoming could prove fatal, rendering the whole approximation insufficient for the task at hand.

It is, thus, our conclusion that, while this way of modeling time series may prove to be just good enough for simple time series, whose complexity does not surpass simple seasonality, a more sophisticated approximation may be required for cases when accuracy (even when faced with the unpredictable) is of greater importance.

REFERENCES

https://busoniu.net/teaching/sysid2022/sysid22en_transient_handout.pdf
https://busoniu.net/teaching/sysid2022/linregcrashcourse_handwritten.pdf
https://en.wikipedia.org/wiki/Mean_squared_error

APPENDIX

```
%% Project
%% 1. Load data
clear variables;
load("product_20.mat");

%% 2. Split data for identification and validation

n = length(y);
n_identification = floor(n * 80 / 100);
t_identification = time(1 : n_identification);
y_identification = y(1 : n_identification);
t_validation = time(n_identification + 1 : n);
y_validation = y(n_identification + 1 : n);
plot(t_identification, y_identification);

%% 3. The procedure

P = 12;

muri = 1 : 7;
mseuri = zeros(1, 7);
mseuri_identification = zeros(1, 7);
mseuri_validation = zeros(1, 7);

for i = 1 : length(muri)
   m = muri(i);

   phi = computePhi(t_identification, m, P);

   theta = phi \ y_identification;
   Y_hat = computePhi(time, m, P) * theta;

   mseuri(i) = computeMse(y, Y_hat);
   mseuri_identification(i) = computeMse(y(1 : n_identification), Y_hat(1 :
n_identification));
   mseuri_validation(i) = computeMse(y(n_identification + 1 : n),
Y_hat(n_identification + 1 : n));

   figure
   hold on,
   plot(time, y, 'g'),
   plot(time(1 : n_identification), Y_hat(1 : n_identification), 'r'),
   plot(time(n_identification : n), Y_hat(n_identification : n), 'r--'),
   title(sprintf("Approximation with m = %d parameters", m)),
   legend('Actual values', 'Approxmiate values'),
   xlabel('Time'),
   ylabel('Sales'),
   hold off
   grid
end

figure,
hold on
plot(muri, mseuri, 'b'),
plot(muri, mseuri_identification, 'r'),
plot(muri, mseuri_validation, 'g'),
hold off
xlabel('m'),
ylabel('MSE'),
title('MSE(m)'),
legend('MSE on whole dataset', 'MSE on identification dataset', 'MSE on
validation dataset'),
grid

%% Functions

function mse = computeMse(Y_actual, Y_approximate)
   N = length(Y_actual);
   msd = 1 : N;
   for i = 1 : N
```

```
        msd(i) = (Y_actual(i) - Y_approximate(i)) ^ 2;
    end
    mse = (1 / N) * sum(msd);
end

function phi = computePhi(input, m, P)
    n = length(input);
    phi = zeros(n, 2 + 2 * m);
    for i = 1 : n
        k = input(i);

        phi(i, 1) = 1;
        phi(i, 2) = k;
        for j = 1 : m
            phi(i, 2 + (2 * j - 1)) = cos(2 * pi * k * j / P);
            phi(i, 2 + (2 * j - 1) + 1) = sin(2 * pi * k * j / P);
        end
    end
end
```