Università della Svizzera italiana
Institute of Computing CI

Numerical Computing **2023**

Student: Birindelli Leonardo **Discussed with:** Morales Jeferson

Bonus assignment **Due date:** Wednesday, 22 November 2023, 11:59 PM

Exercise 1: Inconsistent systems of equations [10 points]

Consider the following inconsistent systems of equations:

a)

$$Ax = b, \quad \text{where} \quad A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix}$$

b)

$$Ax = b, \quad \text{where} \quad A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Find the least squares solution x^* and compute the Euclidean norm of the residual, SE and RMSE.

a) For the first system I obtained the following results:

$$A^T b = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 11 \\ 0 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}$$

$$x^* = \frac{A^T b}{A^T A} = (A^T A)^{-1}(A^T b) = \begin{bmatrix} 1/3 & 0 \\ 0 & \infty \end{bmatrix} \begin{bmatrix} 11 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{11}{3} + 0 * 0 \\ 0 * 11 + \infty * 0 \end{bmatrix} = \begin{bmatrix} \frac{11}{3} \\ Und \end{bmatrix}$$

Looking at the computation, it is possible to notice that is not possible to determine the value of the metrices due to the undefined value I achieved. Therefore, Euclidean norm , SE and RMSE are all equals to undefined

$$r = b - Ax^* = \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \overline{11} \\ 3 \\ Und \end{bmatrix} = Undefined$$

Euclidean norm $= Undefined$
SE $= Undefined$
RMSE $= Undefined$

b) For the second system I achieved the next metrices values :

$$A^T b = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 01 & 1 & 1 & \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 9 \\ 10 \\ 9 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 2 \\ 3 & 6 & 3 \\ 2 & 3 & 3 \end{bmatrix}$$

$$x^* = \frac{A^T b}{A^T A} = (A^T A)^{-1}(A^T b) = \begin{bmatrix} 0.75 & -0.25 & -0.25 \\ -0.25 & 0.4167 & -0.25 \\ -0.25 & -0.25 & 0.75 \end{bmatrix} \begin{bmatrix} 9 \\ 10 \\ 9 \end{bmatrix} = \begin{bmatrix} 2 \\ -0.33 \\ 2 \end{bmatrix}$$

$$r = b - Ax^* = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -0.33 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.33 \\ 0.33 \\ -0.33 \\ 0 \end{bmatrix}$$

Euclidean norm $= \|r\|_2 = \sqrt{\sum_{i=1}^{n} r_i^2} = \sqrt{0.33^2 + 0.33^2 + (-0.33)^2 + 0} = 0.577$

SE $= \|r\|_2^2 = \sum_{i=1}^{n} r_i^2 = 0.33^2 + 0.33^2 + (-0.33)^2 + 0 = 0.33$

RMSE $= \sqrt{\frac{SE}{m}} = \sqrt{\frac{0.33^2 + 0.33^2 + (-0.33)^2 + 0}{4}} = 0.288$

Exercise 2: Polynomials models for least squares [20 points]

a) Write a Matlab function leastSquares.m which takes, as inputs, matrix A and vector b of a generic system $Ax = b$ and returns, as outputs, the least squares solution x^*, the Euclidean norm of the residual, the SE and the RMSE. In order to test your function, write a script `ex2a.m`, in which you use `leastSquares()` to find the solutions of the two inconsistent systems of Exercise 1 and compare the results obtained with the ones you computed by hand

First of all, I managed to create the function `"leastSquares"` as follow:

```
function [LQ_solution,Euclid_norm,SE,RMSE] = leastSquares(A,b)
% Calculate the least squares solution "LQ_solution" of a inconsistent
    system

% Generic system : Ax = b
% A : matrix
% b : vector

LQ_solution= (A'*A)\(A'*b); % calculate the least square solution

%% Computing the residual vector r
r= b-A*LQ_solution;

r_norm=norm(r); % Calculate the norm of the residual vector

%% Computing Euclidean norm of the residual
Euclid_norm = r_norm;

%% Computing the SE
SE=r_norm ^2;

%% Computing the RMSE

m= size(A,1); % memorize the number of equations contained in the system

RMSE = sqrt(SE/m);


end
```

a)

$$Ax = b, \quad \text{where} \quad A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix}$$

My hand-computed results (a) are in line with the values I obtained from the computation of the `"leastSquares"` function applied to the system $Ax = b$.

```
LQ_solution =

    3.6667
       NaN

Euclid_norm =

    NaN

SE =

    NaN

RMSE =

    NaN
```

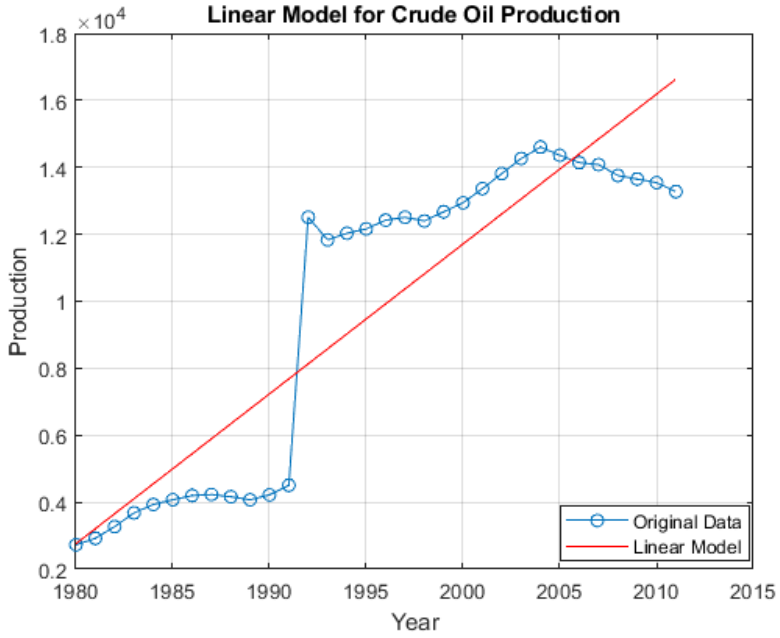
b)

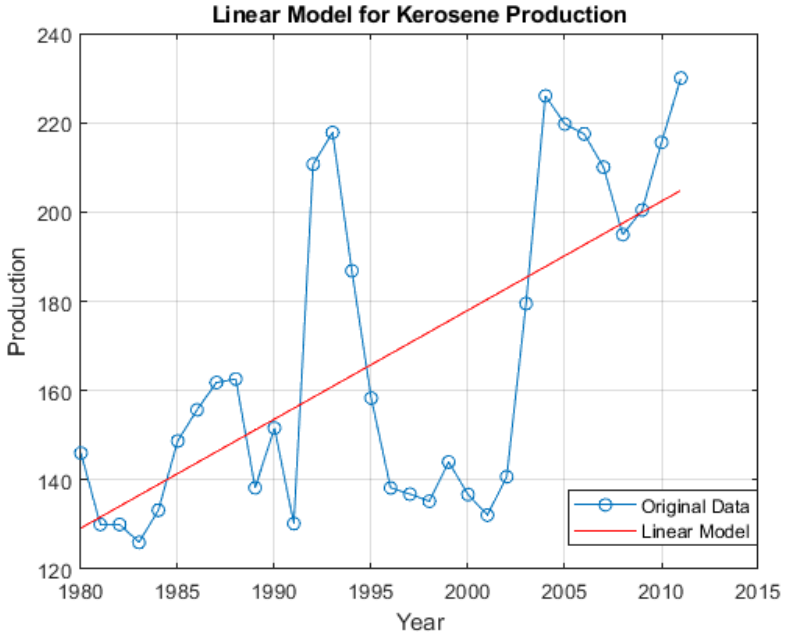
$$Ax = b, \quad \text{where} \quad A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

The following calculations are the values that I obtained computing by hand (b), even in this case the results I achieve was in line with the values I computed with the matlab function

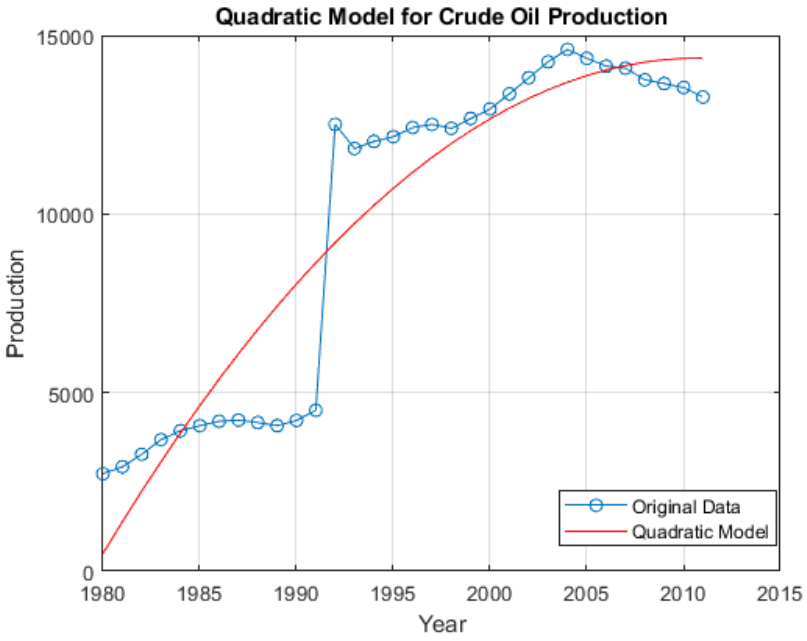
```
LQ_solution =

    2.0000
   -0.3333
    2.0000

Euclid_norm =

    0.5774

SE =

    0.3333


RMSE =

    0.2887
```

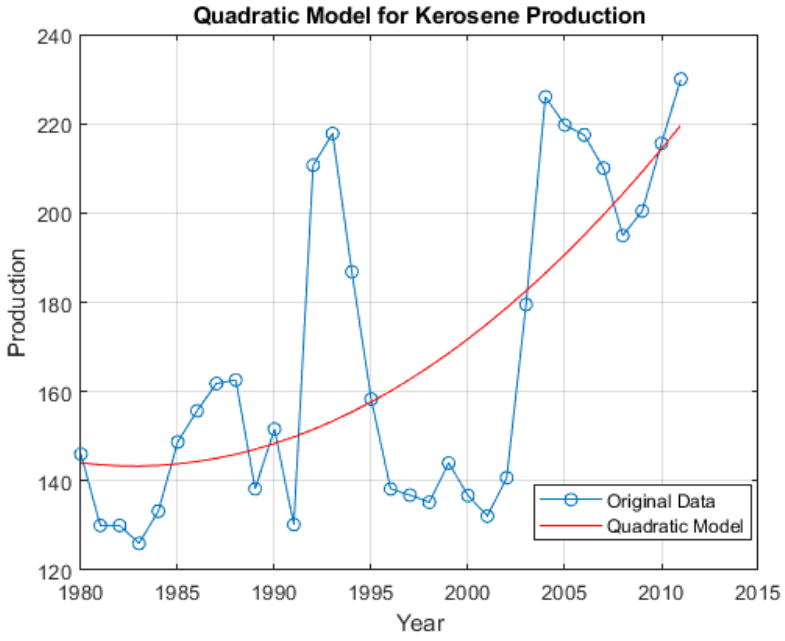
b) Consider the linear model $y_i = \alpha_1 + \alpha_2 x_i$ and apply it to the crude oil and kerosene production data in the period 1980-2011. Write a script `linearModel.m` in which you use `leastSquares()` to compute the least squares solution x^* and the metrics of the residual. For each dataset, create a figure in which you plot the original data points and the linear model



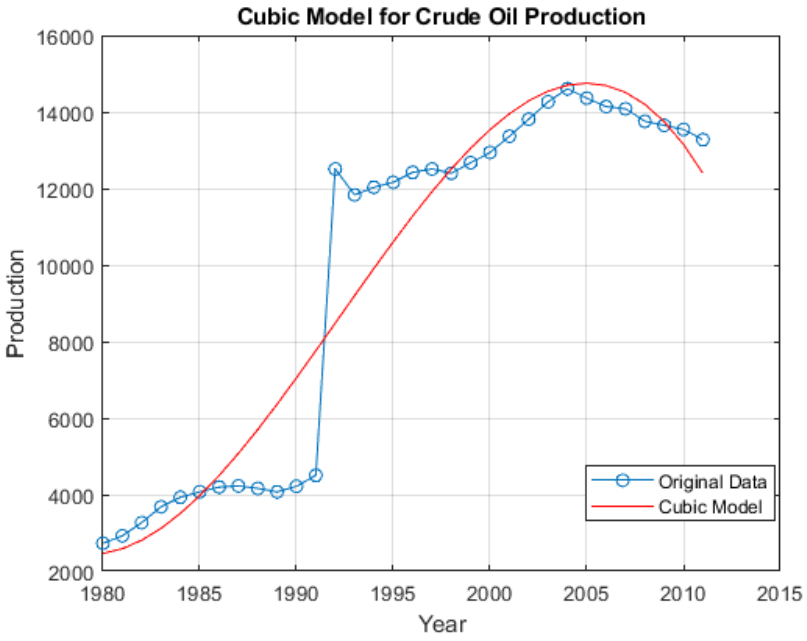


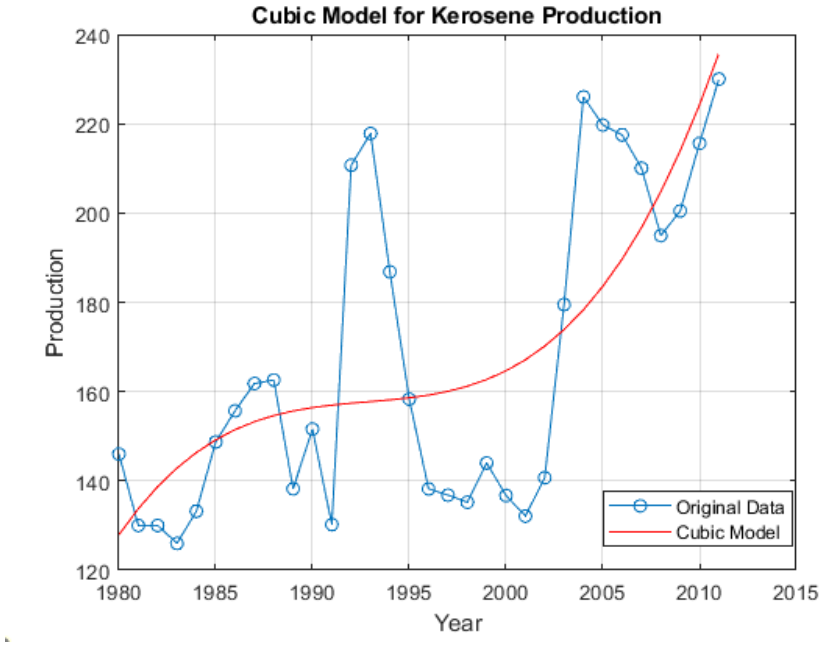
c) Consider the quadratic model $y_i = \alpha_1 + \alpha_2 * x_i + \alpha_3 x_i^2$ and apply it to the crude oil and kerosene production data in the period 1980-2011. Write a script `quadraticModel.m` in which you use `leastSquares()` to compute the least squares solution x* and the metrics of the residual. For each dataset, create a figure in which you plot the original data points and the quadratic model.





d) Consider the cubic model $y_i = \alpha_1 + \alpha_2 * x_i + \alpha_3 x_i^2 + \alpha_4 x_i^3$ and apply it to the crude oil and kerosene production data in the period 1980-2011. Write a script `cubicModel.m` in which you use `leastSquares()` to compute the least squares solution x* and the metrics of the residual. For each dataset, create a figure in which you plot the original data points and the cubic model.





e) Compare the linear, quadratic and cubic models on the basis of the quality metrics computed above, by creating a table containing the results for the two models. Which one of the three models would you pick for the crude oil data? And for the kerosene? Provide an estimate of the crude oil and kerosene production in 2012 by using the three models and compare the values obtained with the real values reported in the data source. Comment on your results.

Table 1: Crude oil data

Model	Euclidean Norm	SE	RMSE
Linear	$1,1471 * 10^4$	$1,3159 * 10^8$	$2,027 * 10^3$
Quadratic	$9,5826 * 10^3$	$9,1827 * 10^7$	$1,6940 * 10^3$
Cubic	$7,9179 * 10^3$	$6,2692 * 10^7$	$1,3997 * 10^3$

Looking the data in the table , it is obvious to say that the best model for the crude oil data is the cubic one due to the fact that has the lower metrics values between all models.

Table 2: Kerosene data

Model	Euclidean Norm	SE	RMSE
Linear	$151,0222$	$2,2808 * 10^4$	$26,6972$
Quadratic	$145,3013$	$2,1112 * 10^4$	$25,6859$
Cubic	$138,4756$	$1,9175 * 10^4$	$24,4793$

In this case ,the best model for the kerosene data is the cubic one given that it produced lower metrics values between all models.

The following tables show the difference between the productions data of each materials contained in the initial file and the values obtained from the estimation of the three different models

Table 3: Productions data for the year 2012

Crude oil	Kerosene
13111,91	267,89

It can be seen that each of the models returns a different estimate for the value for the year 2012, in fact, these values come closer in different ways to the actual values contained within the data file. As it is possible to notice, some models are more accurate than others depending directly on the type of data processed. For example, the estimated value that comes closest to the crude oil production for the year 2012 is obtained from the Quadratic model, while for kerosene production, the value that comes closest to the data contained in the file is obtained from the Cubic model.

Table 4: Productions estimation for the year 2012

Model	Crude oil production estimation	Kerosene production estimation
Linear	$1,7082 * 10^4$	207,2856
Quadratic	$1,4344 * 10^4$	225,1621
Cubic	$1,1473 * 10^4$	248,5750

Exercise 3: Analysis of periodic data [20 points]

The file `temperature.txt` contains the area mean-temperatures of Switzerland between January 1864 and March 2021 included. Temperature data exhibit a periodic behaviour and we will try to capture it by using periodic models. You will need the function `leastSquares()` implemented in Exercise 2.

a) Consider the periodic model $y_i = \alpha_1 + \alpha_2 \cos(2\pi x_i) + \alpha_3 \sin(2\pi x_i)$ and apply it to the temperature data: (I) between January 1960 and January 1963; (II) between January 1960 and January 1970. Write a script `periodicA.m` in which you compute the least squares solutions and the metrics of the residual, and plot the outputs of the model against the original data in both cases.

Figure 1: Period : January 1960 - January 1963

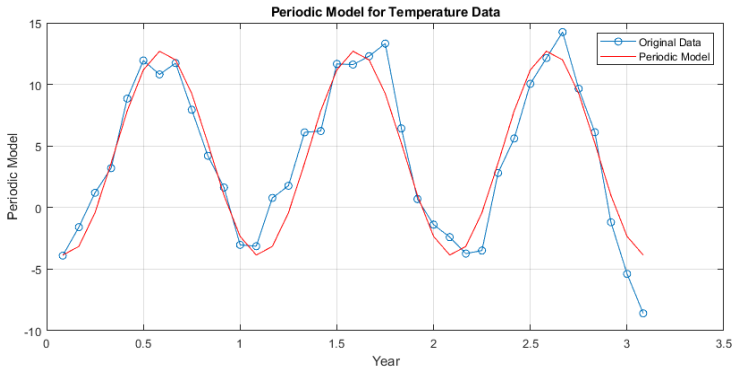
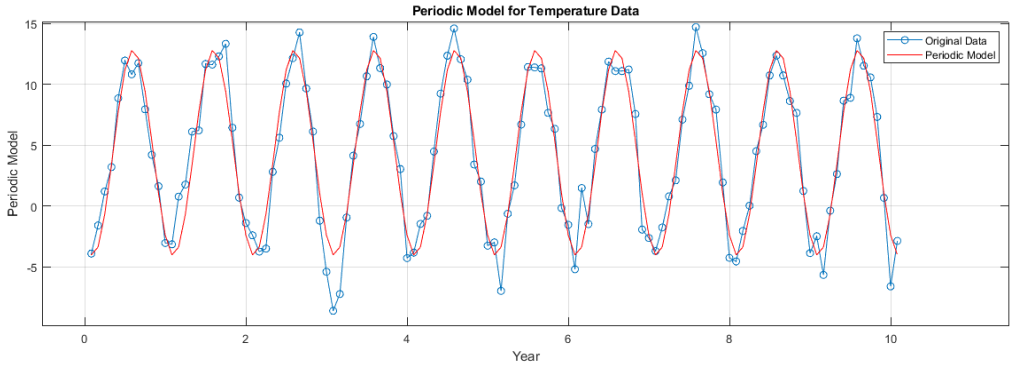


Figure 2: Period : January 1960 - January 1970



b) Repeat the same analysis and plots of the previous point for both time series, by using the periodic model $y_i = \alpha_1 + \alpha_2 \cos(2\pi x_i) + \alpha_3 \sin(2\pi x_i) + \alpha_4 \cos(4\pi x_i)$in the script `periodicB.m`

Figure 3: Period : January 1960 - January 1963

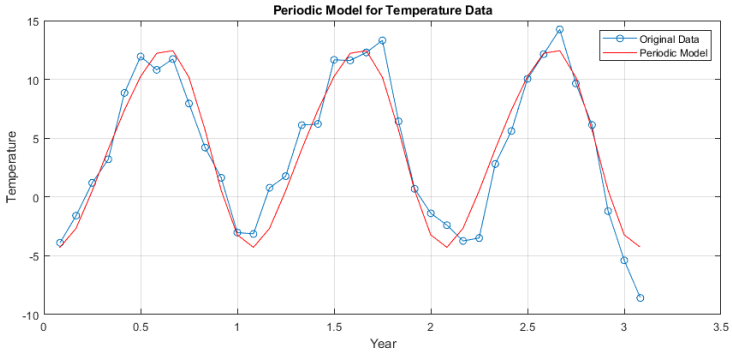
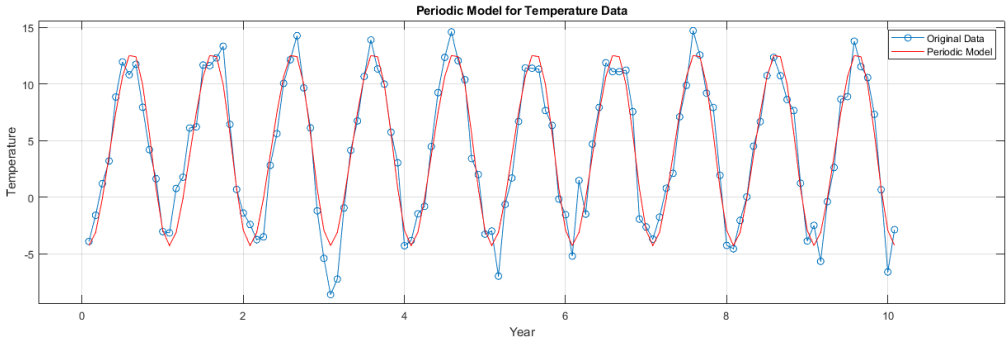


Figure 4: Period : January 1960 - January 1970



c) Compare the models of point (a) and (b). Was it beneficial to include more data? Which model would you prefer? Are you satisfied with the results obtained? If necessary, what would you suggest to improve your models? Motivate your answers.

Analyzing the plots it is possible to notice that including more data lets each model be more precise and accurate instead of having few information as in the case of the data between January 1960 and January 1963. However, I believe that the best model is the model $y_i = \alpha_1 + \alpha_2 \cos(2\pi x_i) + \alpha_3 \sin(2\pi x_i) + \alpha_4 \cos(4\pi x_i)$ given that ,looking at the plots, it seems this model gives a more accurate and in line structure instead of the former which is ,even there are apparently few differences, less precise especially if it models few data

Exercise 4: Data linearization and Levenberg-Marquardt method for the exponential model [20 points]

The file `nuclear.txt` contains the data on the nuclear electric power consumption by year in China in the period 1999-2006. We consider the power law model, expressed as:

$$y_i = \alpha_1 x_i^{\alpha_2}$$

a) Find the least squares best fit by using data linearization and compute the RMSE both of the log-linearized model and of the original exponential model. Include in your report all the computations and the necessary steps, as explained in the slides of the tutorial

First of all, I managed to create the log-linearized version of the power low model in order to be able to calculate the least squares computation. The model I obtained is the following:

$$\ln(y_i) = k_1 + \alpha_2 \ln(x_i)$$

Then,I fitted the data in order to create the system in the form $Ax = b$

$$\begin{bmatrix} 1 & \ln(1) \\ 1 & \ln(2) \\ 1 & \ln(3) \\ 1 & \ln(4) \\ 1 & \ln(5) \\ 1 & \ln(6) \\ 1 & \ln(7) \\ 1 & \ln(8) \end{bmatrix} \begin{bmatrix} k_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 14.09 \\ 15.90 \\ 16.60 \\ 25.17 \\ 41.66 \\ 47.95 \\ 50.33 \\ 54.85 \end{bmatrix} \text{ where } \quad k_1 = \ln(\alpha_1)$$

Once I calculated the least squares solution, I mapped back my results to the exponential model:

$$x^* = \begin{bmatrix} k_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 2.3705 \\ 0.7549 \end{bmatrix}$$

Now the log-linearized model is given by the following equation:

$$\ln(y_i) = 2.3705 + 0.7549^{\ln(x_i)}$$

Hence, the original exponential model is defined by :

$$y_i = 10.7026 x_i^{0.7549}$$

At the end of the models, I obtain the subsequent values for RMSE metric:

Table 5: RMSE metrics results

Log-linearized model	Original Power Law model
0.2042	4.8141

The code that I used to compute the data contained in the `"nuclear.txt"` file is contained in the file named `"ex4a.m"`

b) Write a function levenbergMarquardt() in which you implement the Levenberg-Marquardt algorithm for solving nonlinear least squares problems. Following again the slides of the tutorial, show how you can formulate the problem in order to solve it with Levenberg-Marquardt method and compute analytically all the necessary quantities. Finally, write a script ex4b.m in which you use the function levenbergMarquardt() to fit the data points and compute the RMSE.

The idea behind the resolution of nonlinear least squares solution with the Levenberg-Marquardt algorithm is characterized by the searching for a solution through root resolution of the gradient of the squared residual. Indeed , given a random initial x and a regularization parameter λ, the function computes the vector of the residual $r(\mathbf{x}_i)$, subsequently calculates the associated Jacobian matrix $J_r(\mathbf{x}_i)$ and in conclusion it reckons the residual vector $v^{(i)} = -(A^\top A + \lambda \text{diag}(A^\top A))^{-1} A^\top r(\mathbf{x}_i)$ that it will be added up to the solution $x_{i+1} = x_i + v_i$ for each iteration until the convergence is not reached $\|x - x_k\|_2 \geq 1e-6$ (or the maximum number of iterations is achieved).
Notice that if the value of $\lambda = 0$, the obtained algorithm is equals to the Gauss-Newton method,otherwise if λ value increments , it will increase the influence degree of the matrix $A^t A$ on the condition number of that matrix which one it is used to define how well-conditioned or ill-conditioned a matrix is. Note : the value $1e-6$ represents the threshold to check if the convergence has been reached in a certain iteration of the cycle.

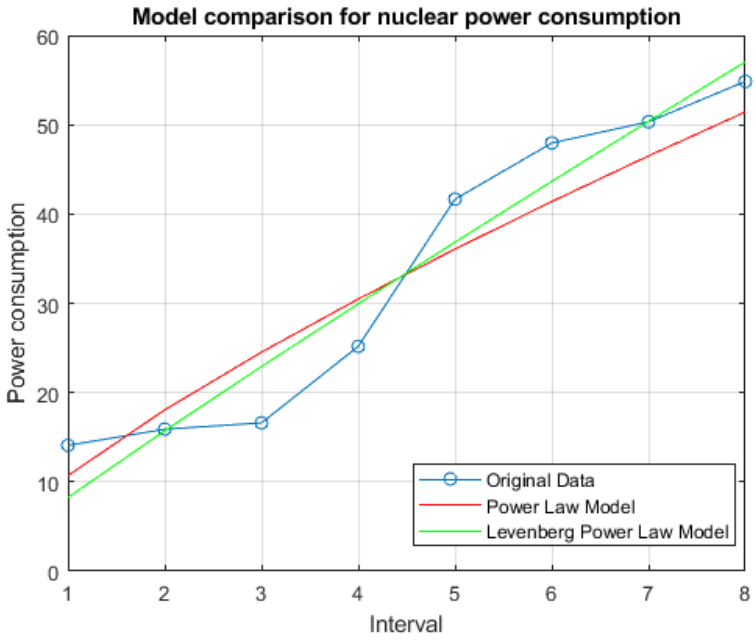
Next, I show the implementation I created to model the Levenberg-Marquard algorithm in the file named "levenbergMarquardt.m":

```
function x = levenbergMargquardt(initial_alpha, lambda, x_data,y_data)
    % Levenberg-Marquardt function
    % Given a random initial guess x and a regularization parameter
       lambda
    % Initialize variables
    x = initial_alpha;
    tolerance= 1e-6; % convergence threshold
    i=1;
    x_k=0;
    %iterative method
     while norm(x-x_k,2)>= tolerance && i<100
        x_k=x;
        alpha1=x(1);
        alpha2=x(2);

        % compute the residual vector 'r'
        r=alpha1*x_data.^(alpha2)-y_data;

        %  Jacobian matrix
        A= [x_data.^alpha2 , x_data.^(alpha2).*alpha1.*log(x_data)];
        % Compute vector v

        B = A'*A;
        v = -((B + lambda * diag(B)) \ (A.' * r));

        if(isnan(v)) %control of the obtained value
            break;
        end

        % Update the parameter vector x(i+1)
        x = x + v;
        i=i+1;
     end
end
```

The RMSE metric I obtained for the application of Levenberg-Marquardt function is the following

```
RMSE of Levenberg-Marquardt application: 2.935365
```

c) Compare the results obtained in points (a) and (b), by extending the script ex4b.m to produce a plot of the original data points together with the two models. Which model would you choose?



Although the plots are really similar , I would choose the Levenberg-Marquardt result because it seems to be more precise than the result obtained by the other model which least squares solution are computed from the application of `leastSquares.m` function. This better accuracy is confirmed by the fact that the RMSE metric is lower than the one obtained in the power law model (Table5).

Note : For some seeds value passed to the `rng`, the random generated numbers used in the initial vector α passed to the `levenbergMarque` function, I notice that it has some problems to calculate the vector solution and the RMSE metric.For example when I set `rng('default')`, the RMSE was $2,3284 * 10^{50}$ and naturally the plot of Levenberg-Marquardt solution has not been generated.
Inside my `ex4b.m` file I set `rng(10)` in order to make the previous plot repeatable

Exercise 5: Tikhonov regularization [15 points]

Consider the ill-posed problem $Ax = b$, where $A \in R^{n \times m}$ and $b \in R^m$. The Tikhonov regularized linear least squares problem is given by:

$$\min_x \|Ax - b\|_2^2 + \alpha \|x\|_2^2 \quad (1)$$

a) Derive analytically the optimal solution x^* of Eq. (1). What is the purpose of the parameter α and how should we proceed to choose its value?

To obtain the optimal solution x^* I need to :

Starting from the initial expression

$$\min_x \|Ax - b\|_2^2 + \alpha \|x\|_2^2$$

Set the gradient to zero

$$2A^T(Ax - b) + 2\alpha x = 0$$

Solve for x and find the optimal solution

$$(A^T A + \alpha I)x = A^T b$$

$$x^* = (A^T A + \alpha I)^{-1} A^T b$$

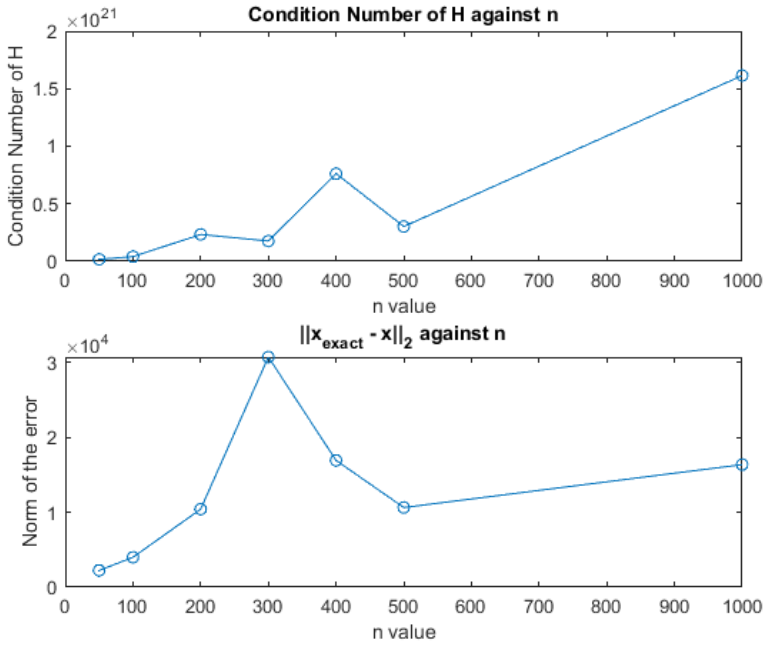
The parameter α is a non-negative number ($\alpha \geq 0$) used in the Tikhonov regularization introduce a value for shifting the diagonals of the moment matrix (which is the matrix $A^T A$) by multiplying itself with the identity matrix I , in practice it becomes a regularization value that check a correct adjustment of the data .

The choice of α is stricly dipendent from the application but generally it is chosen according to a heuristic criterion

b) We consider the Hilbert matrix $H \in R^{n \times n}$ with entries defined as follows:

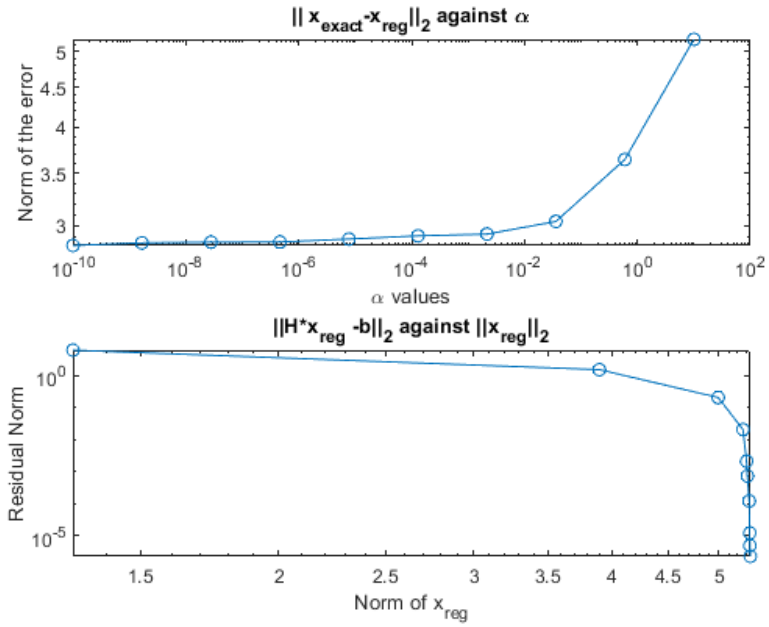
$$H_{ij} = \frac{1}{i + j - 1}$$

for every $i, j = 1, \ldots, n$.. Write a Matlab script illposedHilbert.m in which you generate H for $n = 50, 100, 200, 300, 400, 500, 1000$ and solve the problem $Hx = b$, for $b = Hx_{exact}$ and x_{exact} generated through the function `rand(n,1)`. To make your results reproducible, reinitialize the random number generator to its startup configuration by adding `rng('default')` at the beginning of your script. Produce also two figures in which you plot: (I) the condition number of H (use `cond()` in Matlab) against n; (II) the norm of the error $\|x_{exact} - x\|_2$ against n.



The code to create the previous plot it is contained in the file `illposedHilbert.m`

c) We now focus our attention on the case n = 100. Write a Matlab script regularizedHilbert.m in which you estimate the regularized solution x_{reg} according to Eq. (1) by using at least 10 different values of the parameter α (explain your choice of the values). To visualize the results, produce two figures in which you plot: (I) the norm of the error $\|x_{exact} - x_{reg}\|_2$ against the values of α; (II) $\|Hx - b\|_2$ against $\|x\|_2$ for the different values of α. Comment your results.



In my matlab script I decided to use the function `logspace(-10, 1, 10)` which creates an array of 10 values logarithmically spaced between decades 10^{-10} and 10^{1} because I wanted to show how the increasing value α impacts on the final solutions , beginning from a really small value to (which means the regularization term as a tiny influence so the returned solutions are not regularized) reaching a big and strong regularization value that provide a smoother and shaped solutions.

I noticed that as the value of α increases, we obtain a value of x_{reg} such that the expression $\|Hx_{reg} - b\|$ becomes very close to 0, while the norm of the error $\|x_{exact} - x_{reg}\|$ tends to increase.

1. References

1. "Introduction to Least Squares" slides of the self-study lecture avaiable on iCorsi platform
2. "Advanced Topics on Least Squares" slides of the self-study lecture avaiable on iCorsi platform
3. "NonLinear Least Squares" book chapter 4.5 avaiable on iCorsi platform
4. A. N. Tikhonov and V. Y. Arsenin, Solutions of Ill-posed Problems