Quadratic Curve Fitter Application in Fall of and Object

Cagatay Kavas
cagataykavas2001@gmail.com
19290438

***Abstract*—In real life scenarios sometimes the data set we are going to make calculations on can have missing parts or values may have been measured incorrectly. In this type of situations interpolating the missing points is crucial. We estimated the mathematical model of free fall of an object which we know its place in 14 time inputs which are scattered and its movement function is in quadratic in this simulation. First we interpolated a function between given input and output signals with least square method. We calculated some statistic values to analyze how well our model fits to measured values such as deflection and norm of residuals. After interpolating a function we used Newton's principles and made a simultaneous model which compares the falling of the object according to measured and interpolated values under constant gravity while code is plotting height/time and velocity/time graphs of the interpolated function simultaneously.**

Keywords*—*Height, velocity, acceleration, gravity, free fall, vector, interpolation, estimation, linear algebra, coefficient, sequences, transpose, row, column, inverse, curve fitting, least square method, matrix, dot multiplication, residual, norm.

I. INTRODUCTION

In this project I estimated the quadratic mathematical model of an object according to the given time inputs and height outputs while Newton principles are satisfied on Matlab in a m-file while not calculating the change in gravity at height. Code compares falling of object according to measured and calculated values and calculates and plots velocity of the object and distance traveled by the object according to calculated function over time simultaneously.

Velocity is the displacement over time and acceleration is the change in velocity over time so using Newton's principles we can derive the mathematical relation of displacement, velocity and acceleration to build a mathematical model to our problem and find its analytical solution.

In real life measurement devices may have some delay times or there may be some missing parts in measured data set. The process of interpolation fills the missing parts according to known relation with inputs and outputs.

Calculations change according to our expectation of the function's degree. Estimating the degree of function requires experience in the field however in this problem the missing function is known to be quadratic.

Interpolation method is required because it may contain data over time which may take long time to collect such as temperature of air hourly over a year. But some values may be missed because of technical issues such as power outage etc. In order to make calculations on data sets such as these we need to fill missing parts of our data set first. Missing parts must be filled according to relationship between known input and output datas.

II. THEORITICAL BACKGROUND

The Earth has tendency to pull objects towards its center because of its mass. This effect is called gravity. Since Earth is constantly pulling objects towards its center objects are under force. If an object does not initiate force in opposite direction force upon the object will be unbalanced (there will be a net force different than 0 N) and acceleration occurs. Free falling is one of the occurances of an object under unbalanced forces.

Acceleration is the change in velocity over time. Gravity changes as the object's distance from Earth's center changes but we will ignore this phenomena as well as the resistance of air. Constant gravity (g) is widely accepted as 9.81 m/s^2 and it is denoted as g.

Fall of an unmoving object from the air is our system to find a mathematical solution to. So starting speed of the object is considered as V_s. Starting point x_s is the height of fall and x_f final height (when an object hits the ground) so its height is 0 m. So Newton's principles are:

Equation for final speed: (1)

$$V_f = V_s + g\ x\ t$$

Equation for total displacement: (2)

$$x_f = x_s - V_s\ x\ t - \frac{1}{2} g\ x\ t^2$$

$$x_f = -\frac{1}{2} g\ x\ t^2 + V_s\ x\ t + x_f$$

Since x_s is the height of the fall and g is considered as a constant we can find the fall time from this equation. (3)

$$x_f = -\frac{1}{2} g\ x\ t^2 + V_s\ x\ t + x_f$$

$$-\frac{1}{2} g\ x\ t^2 + V_s\ x\ t + x_s - x_f = 0$$

We must apply the quadratic formula to solve this equation

Quadratic formula: If $ax^2 + bx + c = 0$ (4)

Solutions are:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

So the falling time is: (5)

$$\frac{-V_s + \sqrt{\frac{-4}{-\frac{1}{2}\ x\ 2}}}{\frac{-V_s \pm \sqrt{V_s^2 - \left(4\ x - \frac{1}{2}\ x\ (x_s - x_f)\right)}}{-\frac{1}{2} g\ x\ 2}}$$

$$\frac{-V_s \pm \sqrt{V_s^2 - \left(-2\ x\ (x_s - x_f)\right)}}{-g}$$

$$\frac{-V_s \pm \sqrt{V_s^2 + 2\ x_s - 2\ x_f}}{-g}$$

There are two solutions but since the time is flowing falling time can't be negative we must keep that in mind.

Since we derived mathematical equations for falling of an object without air resistance now we need to derive the equations for the interpolation.

The interpolation method I used on my code to fill the missing parts of our measurements is Curve Fitting with Least Squares Method.

As we know from linear algebra we can find the values of coefficients if we know some input and output relations. For example we have following equations where x and y are coefficients.

$$a_1 x + b_1 y + c = N_1$$
$$a_2 x + b_2 y + c = N_2$$
$$\dots$$
$$a_n x + b_n y + c = N_n$$

Let's put the coefficients of equations in matrices like that. (6)

Inputs:

$$\begin{matrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ & \dots & \\ a_n & b_n & c_n \end{matrix}$$

Coefficients:

$$\begin{matrix} x \\ y \\ z \end{matrix}$$

Outputs:

$$\begin{matrix} N_1 \\ N_2 \\ \dots \\ N_n \end{matrix}$$

If we do matrix multiplication with inputs matrix and coefficients matrix we will get the outputs. So our process is: (7)

$$\begin{matrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ & \dots & \\ a_n & b_n & c_n \end{matrix} \quad . \quad \begin{matrix} x \\ y \\ z \end{matrix} \quad = \quad \begin{matrix} N_1 \\ N_2 \\ \dots \\ N_3 \end{matrix}$$

If we consider inputs as *A*, coefficients as *x* and outputs as *B* we get: (8)

$$A\ .\ b \quad = \quad B$$

Normally when we get equation (7) we apply gauss operations to find coefficients however when there we can't solve the equation we apply Curve Fitting with Least Squares Method.

Least Squares Method says that multiplying each side of the unsolvable equation with transpose A (A^T) we will get and equation which is solvable: (9)

$$A^T\ .\ A\ .\ b\ = A^T\ .\ B$$

Our problem is to interpolate a function so our equation is: (10)

$$a\ {t_1}^2 + b\ t_1 + c = N_1$$
$$a\ {t_2}^2 + b\ t_2 + c = N_2$$
$$\dots$$
$$a\ {t_n}^2 + b\ t_n + c = N_n$$

If we divide the equation into inputs, coefficients and outputs we get: (11)

$$\begin{matrix} (t_1)^2 & t_1 & 1 \\ (t_2)^2 & t_2 & 1 \\ \dots & & \\ (t_n)^2 & t_n & 1 \end{matrix} \cdot \begin{matrix} a \\ b \\ c \end{matrix} = \begin{matrix} N_1 \\ N_2 \\ \dots \\ N_N \end{matrix}$$

Now we apply (9) to (11) : (12)

$$\begin{bmatrix} (t_1)^2 & (t_2)^2 & \dots & (t_n)^2 \\ t_1 & t_2 & \dots & t_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} (t_1)^2 & t_1 & 1 \\ (t_2)^2 & t_2 & 1 \\ \dots & \dots & \dots \\ (t_2)^2 & t_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ \dots \\ c \end{bmatrix} = \begin{bmatrix} (t_1)^2 & (t_2)^2 & \dots & (t_n)^2 \\ t_1 & t_2 & \dots & t_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} N_1 \\ N_2 \\ N_n \end{bmatrix}$$

Series sum is a process of adding values of function's output of all values between the integer under the sum symbol to the integer on the sum symbol one by one. By simplifying previous equation we get: (13)

Where $\sum = \sum_{i=1}^{n}$

$$\begin{matrix} \sum_1^n (t_i)^4 & \sum_1^n (t_i)^3 & \sum_1^n (t_i)^2 \\ \sum_1^n (t_i)^3 & \sum_1^n (t_i)^2 & \sum_1^n (t_i)^1 \\ \sum_1^n (t_i)^2 & \sum_1^n (t_i)^1 & \sum_1^n (t_i)^0 \end{matrix} \cdot \begin{matrix} a \\ b \\ c \end{matrix} = \begin{matrix} \sum_1^n (t_i)^2 N_i \\ \sum_1^n (t_i)^1 N_i \\ \sum_1^n (t_i)^0 N_i \end{matrix}$$

Notice that power of each t_i can be calculated by the number of:
Max row + max column – current column – current row

If we say: $A^T \cdot A = X$, $A^T \cdot B = Y$ and coefficients x we can calculate coefficients (x) by: (14)

$$X \cdot x = Y \rightarrow x = \frac{Y}{X} \; or \; x = Y^{-1} \cdot X$$

III. Description of the Simulation

We have been given two vectors named as xi and Yi. xi vector consists of independent variables and Yi vector consists of dependent variables. We know that these to vectors define a quadratic equation also independent variables are correct but dependent variables deviated from their true values.

In order to make calculations on that system we need to interpolate known input and outputs values to estimate the closest function which gives approximately close outputs to same given inputs.

So I made matrices from inputs, coefficients and outputs. But we can't find coefficients from gauss elimination method by applying matrix operations because system doesn't have a solution. So I applied Curve Fitting with Least Squares Method from (9) of the theoretical background. Starting of the Question 1 is indicated by "Question 1" printed on the screen.

First we entered Yi vector the dependent values and xi vector the independent values. We know the sizes of the matrices in (13) from theoretical background so we create zero matrices in required sizes for our code. Degree of our function is entered here as well (n).

We saw from theoretical background part that after we multiply each sides of the equation with transpose of the input matrix we get input and output matrices consisting of series sums and a coefficients matrix in multiplication with input series sum matrix.

We know the relation between powers of the t_is and current row, column number so we made a for loop which has increasing row and column values which fills X and Y matrices as we mentioned in (14) from theoretical background. Column number stays still until each row number is used in the function. Then column number increases and row number starts to increase over again until both values exceed their limits so.

After matrices are created we applied one of the two methods for finding coefficients we mentioned in (14) from theoretical background.

Since we know the coefficients now we can print the interpolated function. The first part ended so starting of the Question 2 is indicated by "Question 2" printed on the screen.

After that we move to the second question. We know that from now on these input and output values are actually an object's measured location for every seconds of falling after 13 seconds is given to us in the question. But it is also stated that measured values are not accurate. In order for us to find a mathematical model for the falling of an object we used the solutions we got from previous part. However only difference is that in the formula of height/time of a falling object gravity value is divided by 2 as seen from (2) of the theoretical background. So we multiply that value with 2 to get its true value.

I defined 2 subplots one for comparing measured and calculated height values over time and other one is for analyzing how well calculated model fits to measured data. For second subplot I calculated difference between calculated and measured values by taking the absolute value of subtracting measured values from calculated values. I used the norm function which "a statistic you can use to analyze how well a model fits your data" [3]. I found that function from tools in plot window from basic fitting section and it was explained in the link at references [3].

Users can view the graphs until they are satisfied then they need to enter “1” value from input part to close existing graphs so that the users can see the comparison of measured free falling and calculated free falling of the object.

Then I defined a value hmax which is used in the visualization of graphs in upcoming parts.

I got first gravity, first speed, starting height (these three values are in x), maximum height, given independent time (xi) and dependent measured height (Yi) to define the newtonFalling function which simulates the falling of an object according to Newton’s principals. Visualization of the object free falling according to measurements and calculations is simulated as smooth as possible also velocity/time and displacement/time graphs of calculated free falling are displayed simultaneously.

Falling time of the calculated model of falling ball is defined as fallTime. Since a quadratic equation has two solutions the positive one is the time because time value can’t be negative so we take the maximum root of that equation.

I calculated the maximum velocity and minimum velocity of the calculated object according to (1) at the theoretical background. First I defined a time value in increments of 0.1 then I added all the velocity values into velocityFunction. Then I used min max functions to get up and down limits for my velocity graph which is used later.

Then I defined inputs in the function. Falling time of the calculated object is calculated according to (3) at theoretical background. My code calculates the fallTime which is greatest fall time so I added an if statement to calculate that value. fallTime value is used in limits of graphs and breaking out of the newtonFalling function.

t is the instant time of the object, a is an integer which is used to determine the height of the measured object over time and b value increases a value after a second passes.

Since my code from first assignment worked and I tested the accuracy of that code I modified my homework to display three screens by using subplot function. The moment object starts to fall is defined as t=0, t increases in the while loop after each iteration.

While loop breaks when t (current time) > fallTime condition is reached.

I simulated the fall of object on a graph and I needed 2 graphs as well so I divided the figure to three parts by using subplot function. ax3 is the comparison of falling of object’s measured and calculated free fall, ax1 is for height/time graph of the calculated object and ax2 is for velocity/time graph of the calculated object.

Location of the falling object is defined by the (2) at the theoretical background and I used dots on graphs to visualize the falling of balls from [1] at the references.

“b” value increases over iteration as well. Increment of the “a” value is 10 times greater than t value and t value’s increment is 0.1 second per iteration. Each time mod of a is equal to 0 means passing of 1 second also an increase in b value. So the height of measured object is updated.

Title of the graph, x and y labels, name of the objects (calculated/measured) are entered according to matlab user’s guide from [2] at the references. In order to visualize the falling of the object I set limits to x and y axises so while ground and starting point stay still and object keeps moving towards the ground.

Instead of t I used a value “c” to plot displacement/time and velocity/time. It is because c is a vector starting from 0 and each element increases with 0.1 seconds until it reaches the current time (t) so that when iterations keep happening both graphs will continue to plot graphs simultaneously.

I displayed velocity/time graph without air resistance according to (1) from the theoretical background.

I displayed displacement/time graphs without air resistance according to (2) from the theoretical background.

I followed same steps (title of graph, x and y labels) for displacement/time and velocity/time graphs.

I added a pause so that we can observe the code’s outputs. If we didn’t put a pause we wouldn’t be able to observe the code’s outputs since computer runs the code very fast.

Then I cleared everything in ax3 so that objects don’t overlap into each other.

After while loop ends the fall times of the objects are printed and all objects are at the ground.

IV. Simulational Results

I compared the values in matrices one by one by looking at their outputs without using ; in the code. I saw that values are at their intended places.

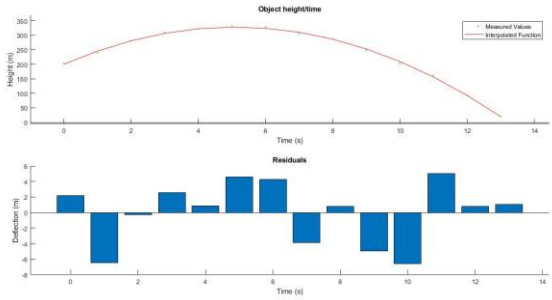
```
A =

       89271           0           0
           0           0           0
           0           0           0

A =

       89271        8281           0
           0           0           0
           0           0           0

A =

       89271        8281         819
           0           0           0
           0           0           0

B =

   1.0e+05 *

    1.3973
         0
         0
-----------------------------------
A =

       89271        8281         819
        8281           0           0
           0           0           0

A =

       89271        8281         819
        8281         819           0
           0           0           0

A =

       89271        8281         819
        8281         819          91
           0           0           0

B =

   1.0e+05 *

    1.3973
    0.1853
         0
-----------------------------------
A =

       89271        8281         819
        8281         819          91
         819           0           0

A =

       89271        8281         819
        8281         819          91
         819          91           0

A =

       89271        8281         819
        8281         819          91
         819          91          14

B =

   1.0e+05 *

    1.3973
    0.1853
```

Outputs of the code

After verifying that I compared the values in matrices with a calculator at the references [4]. And I saw in my code that interpolated function's outputs are also close to the measured values.

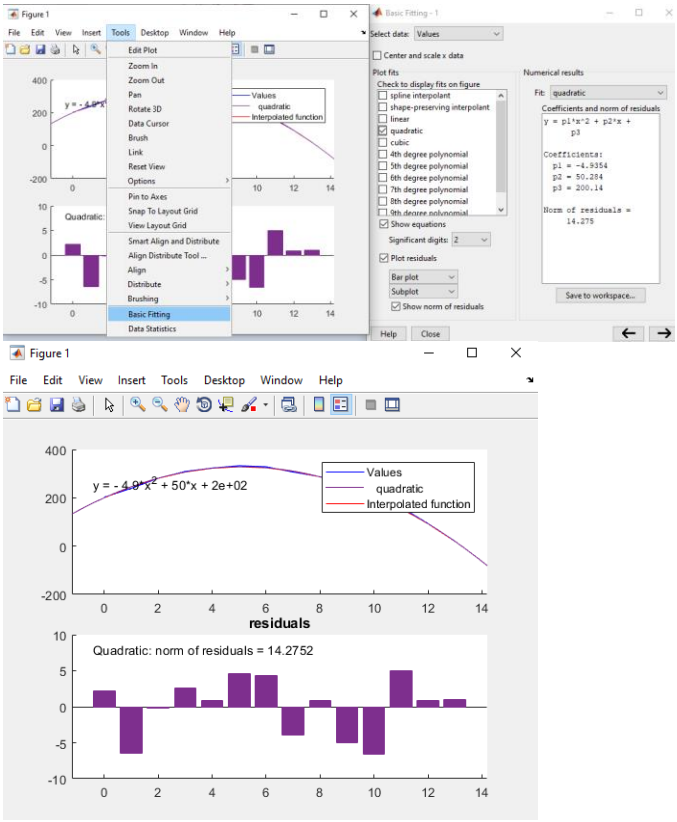
Then I used tools option on the figure and basic fitting.

My code results



Gravity is 9.870838 m/s^2
Starting speed of the object is 50.284380 m/s
Staring height of the object is 200.144250 m
The mathematical model of movement is: x(t) = -4.935419 t^2 + 50.284380 t^1 + 200.144250
Deflection from function is 3.406897
Norm of residuals is 14.275239

Basic fitting results



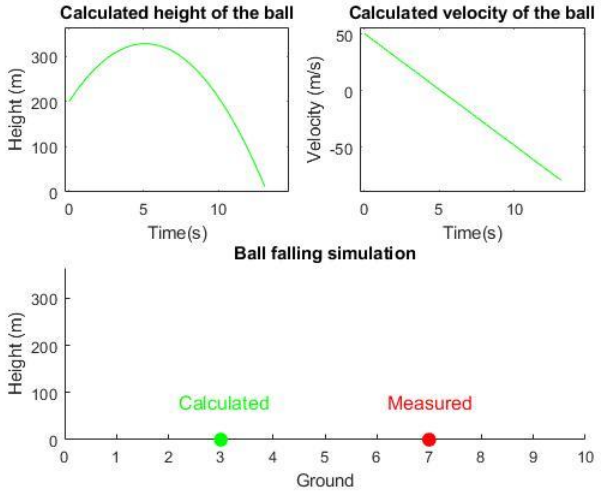
Basic fitting results proved that my code works. Also the limits of the graph are good for visualizing the fall of the object.

I modified my working code from first homework. I verified my last week's code's results by changing gravity value to 10 and placed pausing my simulation in every second. For example if our object were to fall from 80 meters in explained conditions: 80-$g*t^2/2$ = 80-10*$1^2/2$ = 75 would be the position of our object after a second.

After I typed 1 as asked from the code, existing figures got deleted and a new graph which I included here is shown.

"Press 1 to plot the measured falling vs function falling 1"

Second part of my code



Calculated object hit the ground after 13.25 seconds

Measured object hit the ground after 13.00-14.00 seconds

Limits of the graphs are good for visualizing the fall of the object.

I compared the falling time of the object at the references [5] and saw that my answer is correct.

Both object started around 200 meters and then they reached their maximum height about 5 seconds.

Height of the calculated object increased in a curve then decreased in a curve until it hit the ground which is expected because object started moving with a velocity towards sky.

Velocity of the calculated object kept decreasing linearly which is expected because gravitational force is through to core of the earth so it will reduce velocity of the object over time.

Graphs are simulated correctly however since our pause time couldn't match the iteration speed of our while segment there may be differences in real life model and our simulation.

CONCLUSIONS

There may be some situations where missing or incorrect datas in a data set exist. Process of data collection may take long time or it can require resources, man power and can be expensive or we couldn't have a chance to observe that situation again. In this type of situations we interpolate the missing values according to the known datas. We used Curve Fitting with Least Squares Method to interpolate the missing values in this simulation. An engineer must know the mathematics behind an event and built its mathematical model and bring an analytical solution to this problem. If the solution to this problem is as its expected than we can safely use interpolated values in our data set.

We used Newton's principles to bring an analytical solution to comparison of measured and calculated height of an object over time.

Because of the iteration speed of our code there may be some differences in the falling of the objects simultaneously.

REFERENCES

[1] Mathworks.com, https://www.mathworks.com/matlabcentral/answers/502320-how-to-put-a-dot-at-a-certain-place-in-the-graph , KSSV (name of the uploader), 28 Jan 2020

[2] Mathworks.com, The MathWorks. (1993). MATLAB User's Guide. The MathWorks, Inc., Natick, MA." ,https://www.mathworks.com/help/matlab/ref/plot.html

[3] Mathworks.com, The MathWorks. (1993). MATLAB User's Guide. The MathWorks, Inc., Natick, MA." https://www.mathworks.com/help/matlab/data_analysis/interactive-fitting.html

[4] https://www.emathhelp.net/calculators/calculus-2/series-calculator/?f=x%5E2&var=x&a=0&b=13

[5] https://www.omnicalculator.com/physics/free-fall