Numerical Mathematics Assignment 1 Peter-Jan Derks

Exercise 1

Discription:

The function leastsquares looks for the exponents and constants of a linear combination of exponential functions such that the function the best fit of the data in the sense of least squares. The function accepts to inputs, a matrix consisting of x and y values of points as an input and INITIALGUESSES accepts a vector that guesses the optimal exponents. Least squares returns two vectors, one with the calculated optimal exponents and one with the calculated optimal constants. Additionally it returns the residual and a picture of the data and the fit of the function calculated.

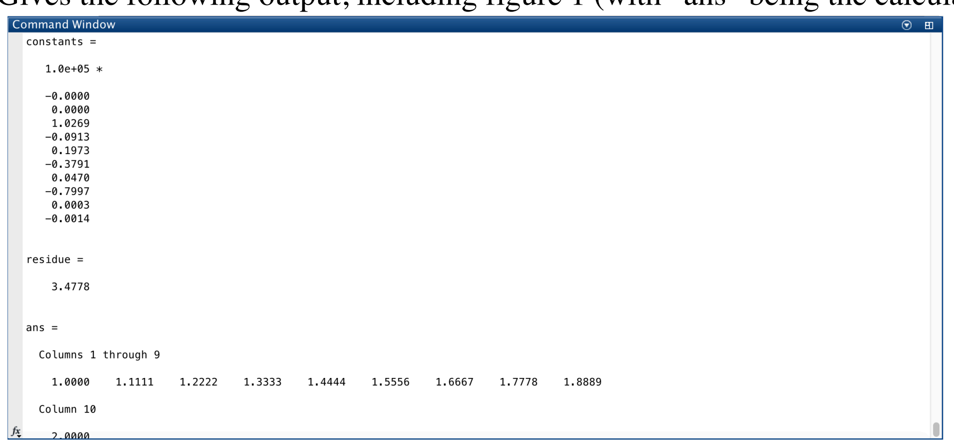
The first computation done by the function uses the method of least squares to find the values of the constants that fit the data best with the initially given lambdas by the user. The “best” solution to Ax=b is found by x = pinv(A)* b.(pinv is a built in matlab function). Than the corresponding residue (depending on the exponents calculated) is minimalized using the built in matlab function fminsearch. Lastly the constants are calculated again using pinv[1]. All calculations are done with matrices and no while loops where used (something which I am proud of).

Picture:

The following input, which uses the first set of data (data1) from the expo-examples.mat file on blackboard:

>> lambda = linspace(1,2,10);[2]
>> leastsquares(data1,lambda)

Gives the following output, including figure 1 (with “ans” being the calculated optimal lambda):

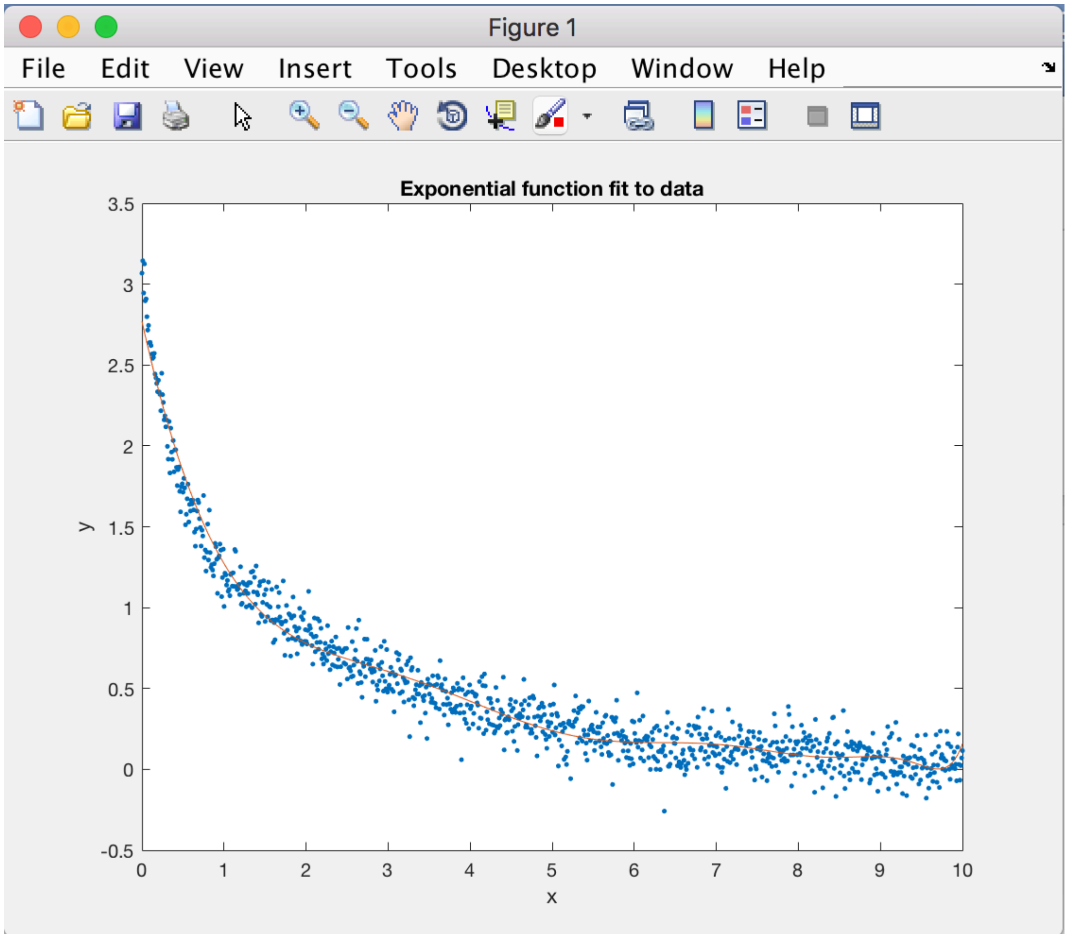


```
Command Window
constants =

   1.0e+05 *

   -0.0000
    0.0000
    1.0269
   -0.0913
    0.1973
   -0.3791
    0.0470
   -0.7997
    0.0003
   -0.0014


residue =

    3.4778


ans =

  Columns 1 through 9

    1.0000    1.1111    1.2222    1.3333    1.4444    1.5556    1.6667    1.7778    1.8889

  Column 10

    2.0000
```

---

[1] (Pictures showed it was needed to calculate the constants again but I don’t fully understand why. When I did not do it the function was way off.)

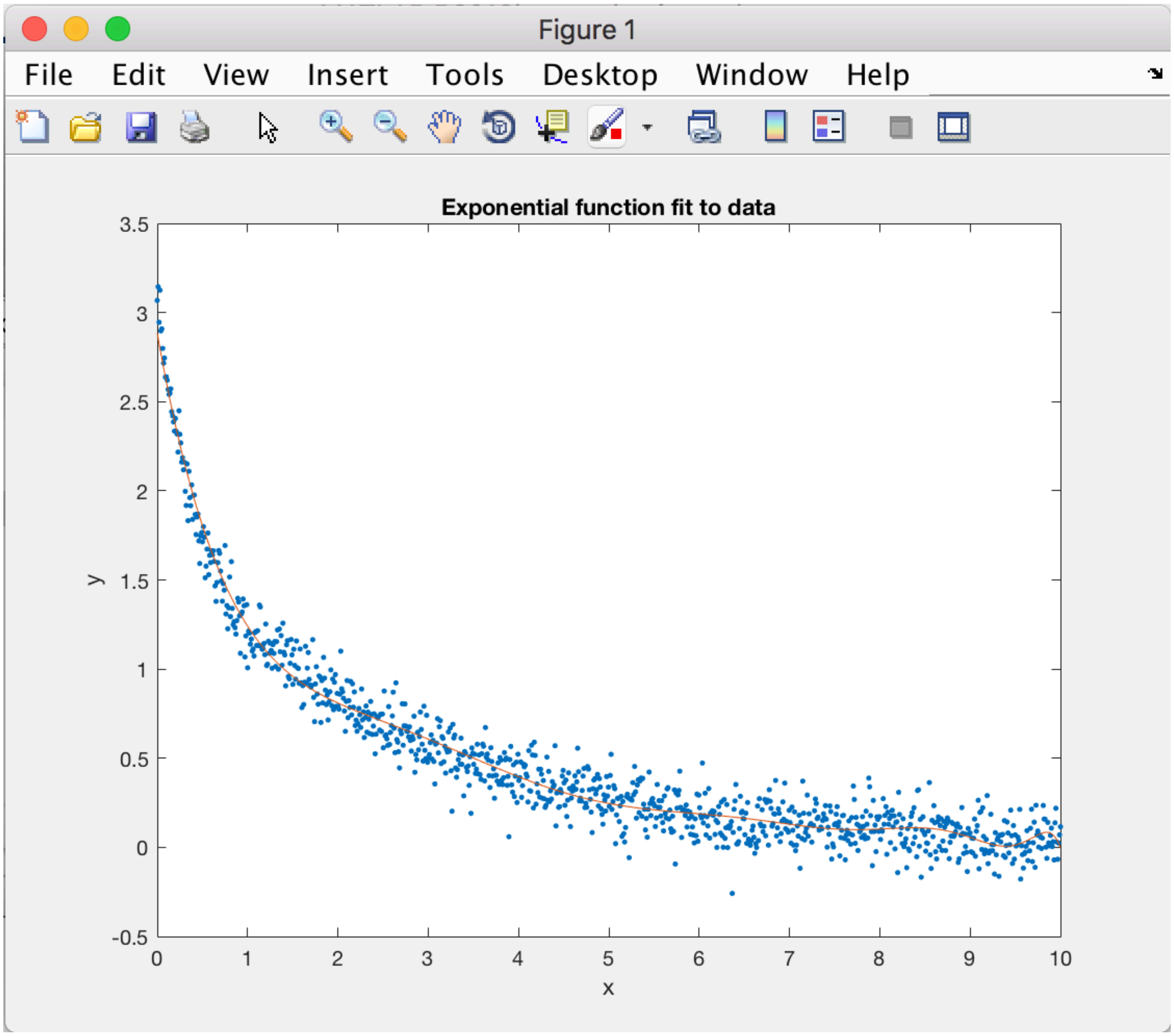
[2] this gives lambda = 1.0000 1.1111 1.2222 1.3333 1.4444 1.5556 1.6667 1.7778 1.8889 2.0000



Questions

The residual gives an indication of the error of the function. The residual I find when running my function in the example above is 3.4778. The amount of exponents present is equal to the length of the vector of the guessed exponential values (the lambdas). At x → 0 the function does not represent the average of the data points. The reason for this is that there are “hidden” exponents which have a value that becomes important at very low x-values. When the amount of lambdas is increased from 10 to 30 the following picture is produced.

```
>> lambda = linspace(1,2,30)
>> leastsquares(data1,lambda)
```



The residual is slightly lower, for 30 lambda values it is 3.3103, meaning the function is a better fit.

Exercise 2

Description

[]=Center() determines the center of mass of a domain enclosed by an interpolated curve. This function has no input or output variables. It opens a figure with a big rectangle on the screen. In the command window the function asks the user to put his hand on the rectangle and outline the circumference of his hand with 40 mouse clicks. When the user accidently clicks on the same point twice in a row a message is displayed in the command window that tells the user not to click on the same position twice. When the 40 points are given the function uses pseudo arc-length parametrization to interpolate the points. Two splines are created one of the x-values with respect to t and one of the y-values with respect to t. “t” is a made-up parameter with t1= 0 and

$t_{i+1} = t_i + \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$. The area of the enclosed domain is calculated using Greene's theorem: $\oint_{\partial\Omega}[f_1\,dx + f_2\,dy] = \int_{\Omega}[\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}]\,dxdy.$

When f2 = x/2 and f1 = -y/2, $\frac{\partial f2}{\partial x} - \frac{\partial f1}{\partial y} = 1$ and the equation calculates the area of the hand. By putting f2 = x/2 and f1 = -y/2 into the left-hand side of the Greene's theorem equation, the area can be calculated. I chose to use the trapz function in MATLAB to calculate –y/2 dx and x/2dy, (y being yy (the y spline in my function) and x being xx (the x spline in my function)).

The error in the trapezoidal rule can be found with: $I(f) - I_t(f) = -\frac{b-a}{12}H^3 f''(\xi)$. (Page 40 course notes). As I don't know the second derivative of the splines I cannot calculate the error. My center of mass is not precise but it is not possible for me to easily calculate the error as I don't know what it should be.

Finally, the function produces a picture of the input points, the interpolated curve, with a clear mark at the center of mass: