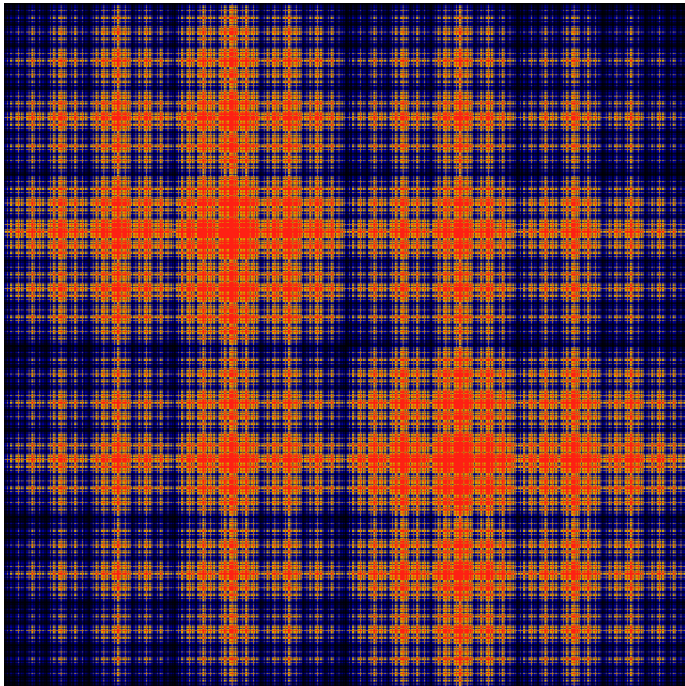


Applied Mathematics and Computing

Volume I



List of Contributors

J. Humpherys
Brigham Young University

J. Webb
Brigham Young University

R. Murray
Brigham Young University

J. West
University of Michigan

R. Grout
Brigham Young University

K. Finlinson
Brigham Young University

A. Zaitzeff
Brigham Young University

Preface

This lab manual is designed to accompany the textbook *Foundations of Applied Mathematics* by Dr. J. Humpherys.

©This work is licensed under the Creative Commons Attribution 3.0 United States License. You may copy, distribute, and display this copyrighted work only if you give credit to Dr. J. Humpherys. All derivative works must include an attribution to Dr. J. Humpherys as the owner of this work as well as the web address to

https://github.com/ayr0/numerical_computing

as the original source of this work.

To view a copy of the Creative Commons Attribution 3.0 License, visit

<http://creativecommons.org/licenses/by/3.0/us/>

or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



Lab 10

Applications: Least-squares fitting I (Stats2)

Lesson Objective: *This section will introduce a more advanced application of Least Squares: fitting data to an circle.*

Fitting data to a circle

Recall that the equation of a circle, with radius r centered at (c_1, c_2) , is given by

$$(x - c_1)^2 + (y - c_2)^2 = r^2. \quad (10.1)$$

Suppose we are given a set of data points closely forming a circle $\{(x_i, y_i)\}_{i=1}^n$. The “best” fit is found via least squares by expanding (10.1) to get

$$2c_1x + 2c_2y + c_3 = x^2 + y^2,$$

where $c_3 = r^2 - c_1^2 - c_2^2$. Then we can write the linear system $Ax = b$ as

$$\begin{pmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_n^2 + y_n^2 \end{pmatrix},$$

where the matrix A and the vector b are obtained by the given data and the unknown x contains the information about the center and radius of the circle and is obtained by finding the least squares solution.

Example

In this section, we fit the following points to a circle:

$$\begin{aligned} &(134, 76), (104, 146), (34, 176), (-36, 146), \\ &(-66, 76), (-36, 5), (34, -24), (104, 5), (134, 76) \end{aligned}$$

We enter them into Python as a 9×2 array:

```
: P = sp.array([[134,76],[ 104,146],[ 34,176],[ -36,146],[ -66,76],[
-36,5],[ 34,-24],[ 104,5],[ 134,76]])
```

Then we can separate the x and y coordinates by the commands $P[:,0]$ and $P[:,1]$, respectively. Hence, we compute A and b by entering the following:

```
: A = sp.column_stack((2*P, sp.ones((9,1), dtype=sp.int_)))
: b = P[:,0]**2 + P[:,1]**2
```

Hence, we get the least squares solution

```
: x = sp.dot(sp.dot(la.inv(sp.dot(A.T,A)),A.T),b)
```

Then we find c_1 , c_2 , and r by:

```
: c1 = x[0]
: c2 = x[1]
: c3 = x[2]
: r = sp.sqrt(c1**2 + c2**2 + c3)
```

We plot this by executing

```
: theta = sp.linspace(0,2*sp.pi,200)
: plt.plot(r*sp.cos(theta)+c1,r*sp.sin(theta)+c2,'-',P[:,0],P[:,1],'*'
)
: plt.show()
```

Problem 1 Download the file `lab10.txt` from the following link: <http://www.math.byu.edu/~jeffh/teaching/m343h/lab10.txt> You can load this datafile into Python by typing

```
: lab10 = sp.genfromtxt("lab10.txt")
```

Now the data is available in the matrix `lab10`. This consists of two columns corresponding to the x and y values of a given data set. Use least squares to find the center and radius of the circle that best fits the data. Then plot the data points and the circle on the same graph. Finish off the problem with a discussion of what you've learned.

Problem 2 The general equation for an ellipse is:

$$A(x - x_0)^2 + B(x - x_0)(y - y_0) + C(y - y_0)^2 = 1$$

Write a program that uses least squares to fit data to an ellipse. One option to finding the center point (x_0, y_0) is to use the mean function. Test the program

on `lab10`. Also test it against `sp.dot(lab10,sp.array([[2,0],[0,1]]))` . Plot the result. How well does your function work?

