Fall 2021 MATH 5720 Homework 6

(*) When submitting on GradeScope, please indicates pages for each question.

1. Suppose data points in $\mathbb{R}^2$ are given by

x_k	-4	-3.5	-3	-2.5	1	1.2	1.5	1.7	2
y_k	-3	-2	-3	-4	-3	1	0	-3	-5

Use least squares approach to fit the data by

(a) A quadratic function.

(b) A polynomial of degree 8 (which will pass all the data points).

Provide a graph for each case. Which model is a better choice for this particular data set? Why?

Note: You may use the operations `A\b` in Matlab/Julia. You are NOT allowed to use the function `polyfit()`. Provide sufficient details of your work.

2. Find the quadratic curve in $\mathbb{R}^2$ of the form

$$(E): \quad ax^2 + bxy + cy^2 + dx + ey + f = 1$$
$$(a, b, c, d, e, f \in \mathbb{R} \ \text{ are parameters to be determined}).$$

that best fits (in least squares sense) the points

x	-5.67	-3.12	-2.05	-2.31	-3.38	3.90	2.69	2.20	3.28	5.83
y	1.11	0.37	-0.42	-0.89	-1.88	2.21	1.13	0.52	-0.27	-1.01

Sketch the quadratic curve (E). You may find the implicit plot example (enclosed) to be useful.

3. (A denoising problem)

Create random vectors $t = (t_1, \ldots, t_n)$ and $b = (b_1, \ldots, b_n)$ by using the following codes

Julia:

```
n=100
t=sort(rand(n))
b=(t-0.3).*sin(pi*t.^2)-0.2*rand(n)
```

Matlab:

```
n=100
t=sort(rand(n,1))
b=(t-0.3).*sin.(pi*t.^2)-0.2*rand(n,1)
```

Use the regularized least squares model

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x} - \mathbf{b}\|^2 + \lambda R(\mathbf{x})$$

$$\text{where} \quad R(\mathbf{x}) = \sum_{i=1}^{n-1} (x_i - x_{i+1})^2, \quad \mathbf{b} = (b_1, \ldots, b_n), \ \mathbf{x} = (x_1, \ldots, x_n).$$

to find an approximate smooth signal $\mathbf{x}$ with different values $\lambda \in \{5, 100, 500\}$. Provide your answers with some graphs.

4. (Circle fitting) Use Julia/Matlab to generate $m = 50$ points of the form

$$(u_i, v_i) = (\alpha_i + \eta_i \cos\theta_i, \beta_i + \eta_i \sin\theta_i) \in \mathbb{R}^2, \quad i = 1, 2, \ldots, m.$$

where $\alpha_1, \ldots, \alpha_m$ are uniformly distributed on $[-1, 0]$,
$\beta_1, \ldots, \beta_m$ are uniformly distributed on $[1, 2]$,
$\eta_1, \ldots, \eta_m$ are uniformly distributed on $[3, 5]$,
$\theta_1, \ldots, \theta_m$ are uniformly distributed on $[0, 2\pi]$.

Find the circle that best fits these points using least squares approach, then create a figure that contains the circle and all the points.

Hint: Some useful commands

Julia	Matlab	vector of n random numbers that are:
`rand(m)`	`rand(m,1)`	uniformly distributed on $[0, 1]$.
`rand(m)*0.5`	`rand(m,1)*0.5`	uniformly distributed on $[0, 0.5]$ (scaling effect).
`rand(m) .+ 1`	`rand(m,1)+1`	uniformly distributed on $[1, 2]$ (translating effect).

For example, the parameters can be generated by

Julia:

```
m = 50;
alpha = rand(m) .- 1;
beta = rand(m) .+ 1;
eta = 2*rand(m) .+ 3;
theta = 2pi*rand(m);
```

Matlab:

```
m = 50;
alpha = rand(m,1) - 1;
beta = rand(m,1) + 1;
eta = 2*rand(m,1) + 3;
theta = 2*pi*rand(m,1);
```

Implicit Plot

Hung Phan, UMass Lowell

September 17, 2020

1 An Example

Plot a curve in $\mathbb{R}^2$ given by an equation

$$C := \{(x, y) \in \mathbb{R}^2 \ , \ f(x, y) = c\}$$

For example, in $\mathbb{R}^2$, sketch the curve

$$x^2 + y^2 - xy - 2x + 4y = 5$$

within the region $[-5, 5] \times [-6, 2] \subset \mathbb{R}^2$.

2 Matlab code

Matlab already has the function `fimplicit` that plots the curve $g(x, y) = 0$:

```
g = @(x,y) x.^2 + y.^2 - x.*y - 2x + 4y - 5;
fimplicit(g, [-5 5 -6 2])
```

For more details: https://www.mathworks.com/help/matlab/ref/fimplicit.html#d122e395345

3 Define `fimplicit` in Julia

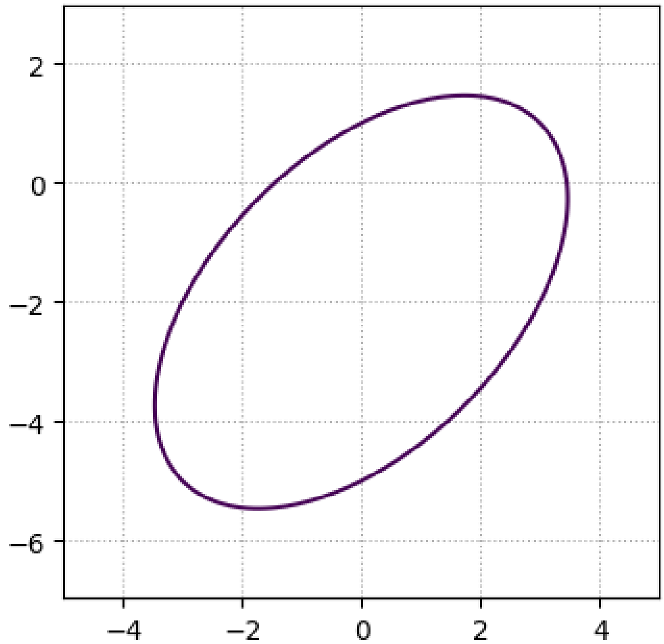
Since Julia does not have the function `fimplicit`, we will define one.

[1]:
```
using PyPlot;

function fimplicit(f,c,xrge,yrge)
    n = 101;
    xs = range(xrge[1], stop=xrge[2], length=n);
    ys = range(yrge[1], stop=yrge[2], length=n);
    xgrid = repeat(xs,1,n);
    ygrid = repeat(ys',n,1);
    z = f(xgrid,ygrid);
    contour(xgrid, ygrid, z, levels=c);
end
```

[3]:
```
f = (x,y) -> x.^2 + y.^2 - x.*y - 2x + 4y;
```

[4]:
```
figure(figsize=(4,4));
axis("equal");
grid(linestyle="dotted");
fimplicit(f,[5], [-5;5], [-6;2]);
```



[ ]:
```
```