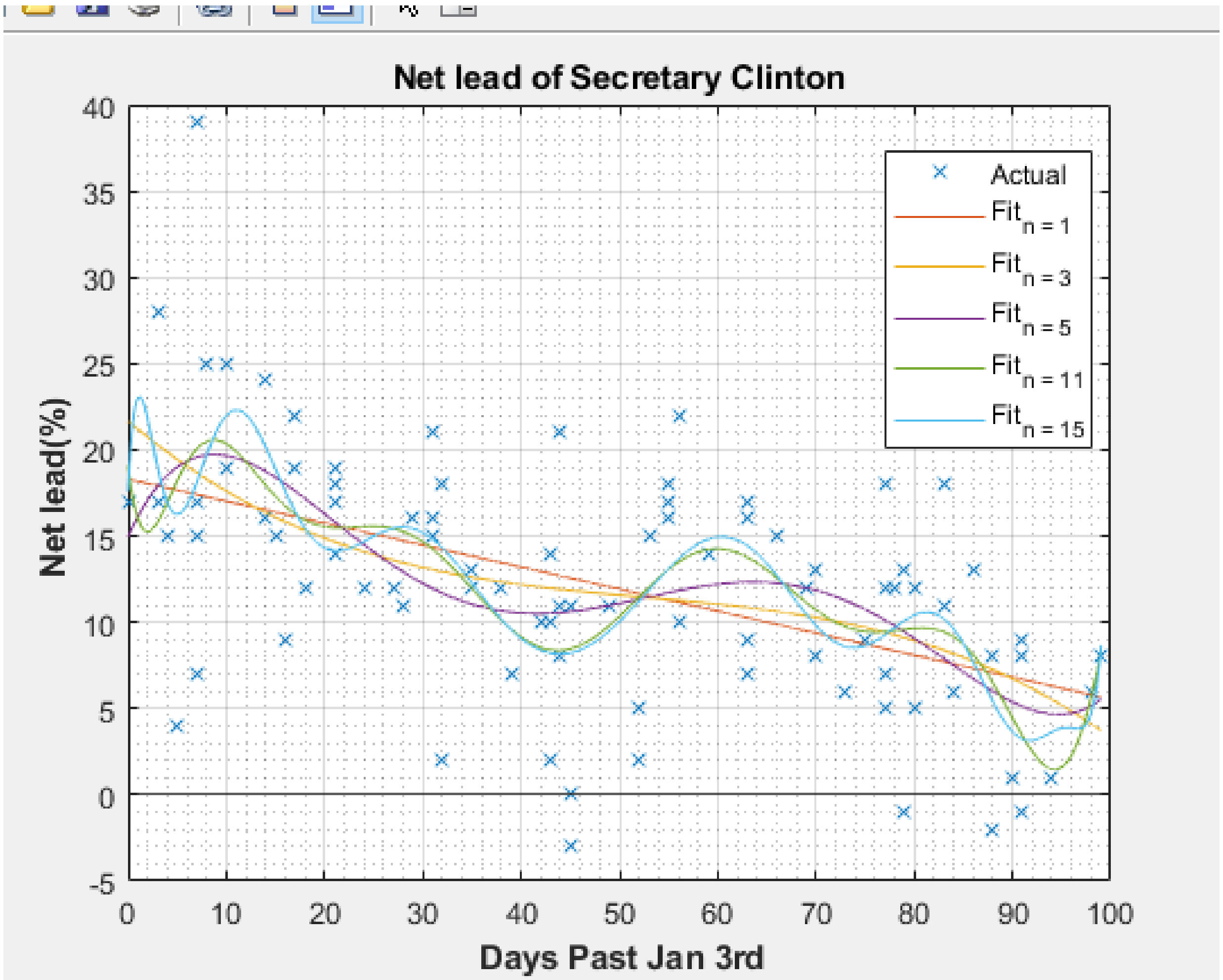
HW 9

MATH 375

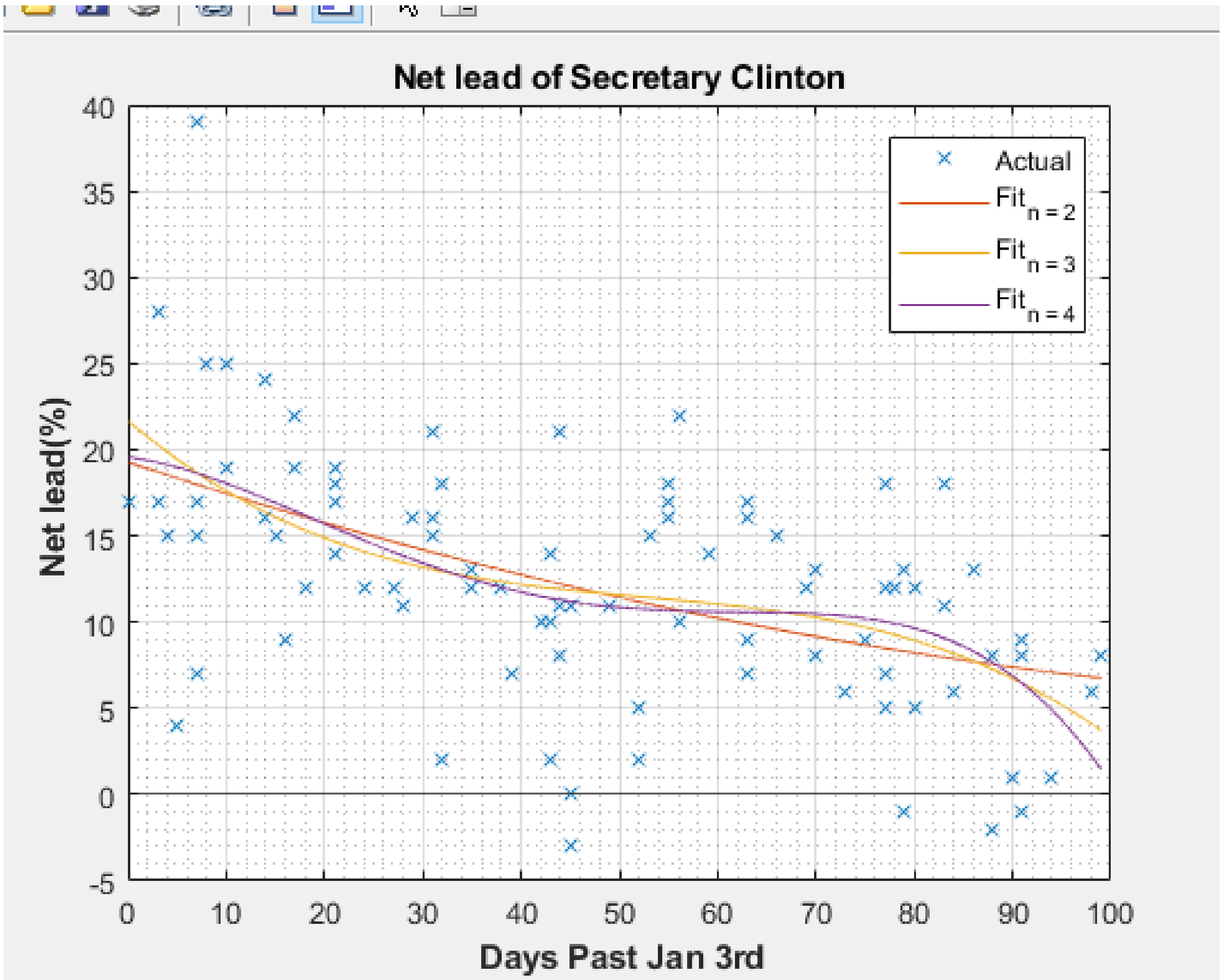
Michael Tanguay

TYPED

2.)



Net lead of Secretary Clinton
Net lead(%)
Days Past Jan 3rd
Actual
Fit n = 1
Fit n = 3
Fit n = 5
Fit n = 11
Fit n = 15
40
35
30
25
20
15
10
5
0
-5
0
10
20
30
40
50
60
70
80
90
100


Net lead of Secretary Clinton
Actual
Fit$_{n = 2}$
Fit$_{n = 3}$
Fit$_{n = 4}$
Net lead(%)
Days Past Jan 3rd
40
35
30
25
20
15
10
5
0
-5
0
10
20
30
40
50
60
70
80
90
100

WRITTEN

1a) $A - \lambda I = \begin{bmatrix} 1-\lambda & -3 & 3 \\ 2 & -2-\lambda & 2 \\ 2 & 0 & -\lambda \end{bmatrix}$

$\det(A - \lambda I) = (1-\lambda)\left(-\lambda(-2-\lambda)\right) + 3(-2\lambda - 4) + 3\left(2(2+\lambda)\right)$

$= 1-\lambda(2\lambda + \lambda^2) - 6\lambda - 12 + 12 + 6\lambda$

$= 2\lambda + \lambda^2 - 2\lambda^2 - \lambda^3$

$= \boxed{-\lambda^3 - \lambda^2 + 2\lambda}$

$= -\lambda(\lambda - 1)(\lambda + 2)$, $\lambda = \boxed{-2, 0, 1}$

1b.) See matlab code. Yes, they agree.

1c.) $x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

iteration #1:

$x_1 = Ax_0 = \begin{bmatrix} 1 & -3 & 3 \\ 2 & -2 & 2 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$

$x_1 = x_1 / \|x_1\|_\infty = [1\ 2\ 2] / 2 = \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix}$

$x_2 = Ax_1 = \begin{bmatrix} 1 & -3 & 3 \\ 2 & -2 & 2 \\ 2 & 6 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix}$

$x_2 = x_2 / \|x_2\|_\infty = \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix} \Rightarrow -3/2 \begin{bmatrix} -1/3 \\ -2/3 \\ -2/3 \end{bmatrix}$

$\Rightarrow$ The eigenvector is $\boxed{\propto \begin{bmatrix} 1/3 \\ 2/3 \\ 2/3 \end{bmatrix}}$ with an eigenvalue of:

$$\lambda = \frac{Ax \cdot x}{x \cdot x} = \frac{[0.5, 1, 1] \cdot [0.5, 1, 1]}{[0.5, 1, 1] \cdot [0.5, 1, 1]} = 1$$

but we already know $\boxed{\lambda = 1}$ from the Matlab calc. I know it is supposed to narrow down to the dominant vector and eigenvalue but that is with a guess of $x = \langle x_1, 0, x_3 \rangle$. It really depends on your guess.

2a.)

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \Rightarrow \begin{bmatrix} x_1^n & x_1^{n-1} & \cdots & 1 \\ x_2^n & x_2^{n-1} & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ x_n^n & x_n^{n-1} & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$n \times 2 \;\times\; 2 \times 1 = 2 \times 1$

← This can be reversed, see Matlab code.

2e.) The fit should not exceed $n = 5$ primarily due to Kcond and being ill-conditioned. Moreover, I believe $n \geq 2$ primarily because other values of n seem to represent more points. If I had to guess the best fit (I could review the average error associated with each points and choose the right answer mathematically) that fit would be $n = 4$. (close tie to $n = 3$) Simply because it represents the most amount of points with the least error

See Plot of $n = [2, 3, 4]$

MATLAB CODE

Problem 1

```
clear, clc, close all, format short
A = [1 -3 3; 2 -2 2; 2 0 0]; norm(A,inf)
[V,D] = eig(A)
%Check by hand and with Matlab
syms lam
B = A - eye(length(A))*lam;
C = det(B);
solve(C,lam)
syms x
y2 = ((1-x)*(-x*(-2-x))+ 3*(-2*x-4) + 3*(2*(2+x)));
y2 = simplify(y2)
eigen_vals = solve(y2 == 0, x)
```

Problem 2

```
clear, clc, close all
data = xlsread('POLLS.csv');
figure(1); plot(data(:,1), data(:,4), 'x'); grid on; grid minor
title('Net lead of Secretary Clinton'); ylabel('\bf Net lead(%)')
xlabel('\bf Days Past Jan 3rd'); hold on; Legend{1} = 'Actual';
n = [1,3,5,11,15]; x= [min(data(:,1)):0.01:max(data(:,1))];
for z = 1:length(n)
a = flip(poll_projection(((data(:,1))),((data(:,4))),n(z)));
plot(x,polyval(a,x)); %Plotted with more points for smoothness
Legend{z+1} = sprintf('Fit_{n = %d}',n(z));
end
yline(0); legend(Legend,'Location', 'best')
```

Functions

```
function [coeffs] = poll_projection(xvals,yvals,n)
for i = 1:n+1
    M(:,i) = (xvals).^(i-1);
end
[row,column] = size(M);
[Q,R,P] = qr(M);
c = Q'*yvals;
yvals = R(1:column,1:column)\c(1:column);
coeffs = P*yvals;
end
```