AMS 326: Exam 1 Report

AMS 326-1: Numerical Analysis, Spring 2025
Joe Martinez

February 25, 2025

1 Problem Description

The monthly temperature averages, along with min and max, in Stony Brook are tabulated and graphed here. To simplify the project, we assume each month has precisely 31 days and the average for a given month can be considered as the temperature of the 16th day.

2 Fitted Polynomial

Given the 12 data points of average temperaure with day, we must fit the points into the following cubic

$$\mathrm{P}_3(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$

Solving for the equation: $A^TAx = A^Ty$
We found the polynomial to be:

$$\mathrm{P}_3(t) = -3.99050448 * 10^{-6}t^3 + 9.57687541 * 10^{-4}t^2 + 0.187139469t + 25.9417808$$

3 Temperature on June 4th and December 25th

Given our fitted polynomial, we calculate the temperature on June 4tha dn December 25th, which are day t=159 and day t=366 respectively

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
High	40°F	41°F	48°F	60°F	69°F	77°F	83°F	82°F	75°F	64°F	54°F	45°F
Average	33°F	34°F	40°F	51°F	60°F	69°F	75°F	74°F	67°F	56°F	47°F	38°F
Low	26°F	26°F	32°F	42°F	51°F	61°F	67°F	66°F	60°F	49°F	39°F	31°F

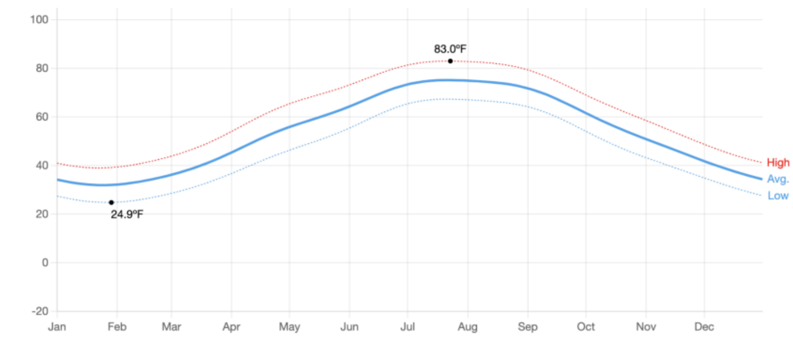
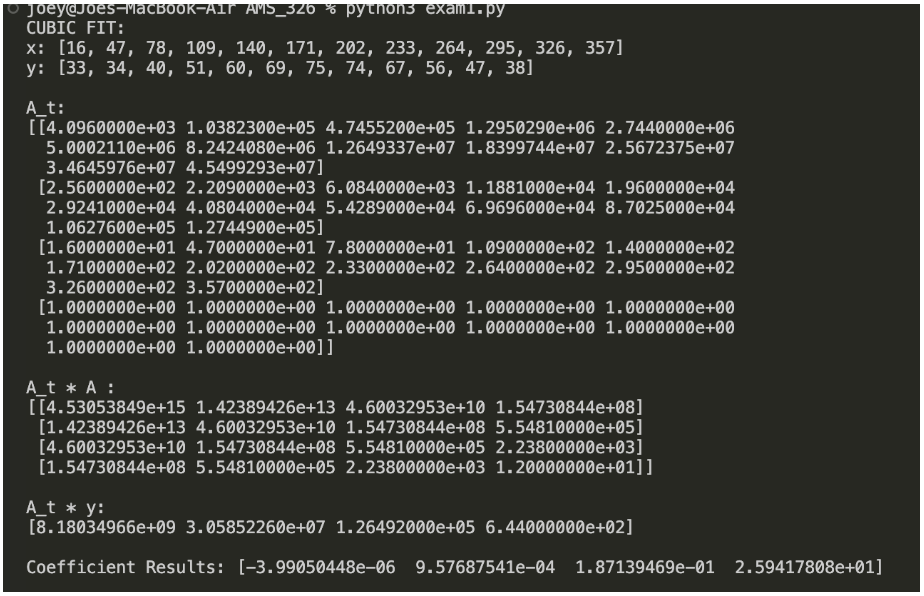


Figure 1: Enter Caption

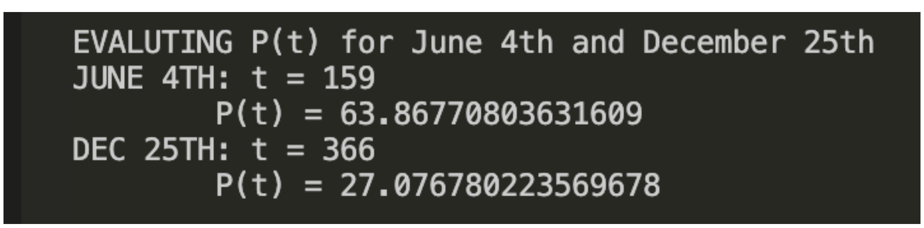
```
Coefficient Matrix:
[[4.0960000e+03 2.5600000e+02 1.6000000e+01 1.0000000e+00]
 [1.0382300e+05 2.2090000e+03 4.7000000e+01 1.0000000e+00]
 [4.7455200e+05 6.0840000e+03 7.8000000e+01 1.0000000e+00]
 [1.2950290e+06 1.1881000e+04 1.0900000e+02 1.0000000e+00]
 [2.7440000e+06 1.9600000e+04 1.4000000e+02 1.0000000e+00]
 [5.0002110e+06 2.9241000e+04 1.7100000e+02 1.0000000e+00]
 [8.2424080e+06 4.0804000e+04 2.0200000e+02 1.0000000e+00]
 [1.2649337e+07 5.4289000e+04 2.3300000e+02 1.0000000e+00]
 [1.8399744e+07 6.9696000e+04 2.6400000e+02 1.0000000e+00]
 [2.5672375e+07 8.7025000e+04 2.9500000e+02 1.0000000e+00]
 [3.4645976e+07 1.0627600e+05 3.2600000e+02 1.0000000e+00]
 [4.5499293e+07 1.2744900e+05 3.5700000e+02 1.0000000e+00]]
```

Figure 2: Enter Caption



```
joey@Joes-MacBook-Air AMS_326 % python3 exam1.py
CUBIC FIT:
x: [16, 47, 78, 109, 140, 171, 202, 233, 264, 295, 326, 357]
y: [33, 34, 40, 51, 60, 69, 75, 74, 67, 56, 47, 38]

A_t:
[[4.0960000e+03 1.0382300e+05 4.7455200e+05 1.2950290e+06 2.7440000e+06
  5.0002110e+06 8.2424080e+06 1.2649337e+07 1.8399744e+07 2.5672375e+07
  3.4645976e+07 4.5499293e+07]
 [2.5600000e+02 2.2090000e+03 6.0840000e+03 1.1881000e+04 1.9600000e+04
  2.9241000e+04 4.0804000e+04 5.4289000e+04 6.9696000e+04 8.7025000e+04
  1.0627600e+05 1.2744900e+05]
 [1.6000000e+01 4.7000000e+01 7.8000000e+01 1.0900000e+02 1.4000000e+02
  1.7100000e+02 2.0200000e+02 2.3300000e+02 2.6400000e+02 2.9500000e+02
  3.2600000e+02 3.5700000e+02]
 [1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00
  1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00
  1.0000000e+00 1.0000000e+00]]

A_t * A :
[[4.53053849e+15 1.42389426e+13 4.60032953e+10 1.54730844e+08]
 [1.42389426e+13 4.60032953e+10 1.54730844e+08 5.54810000e+05]
 [4.60032953e+10 1.54730844e+08 5.54810000e+05 2.23800000e+03]
 [1.54730844e+08 5.54810000e+05 2.23800000e+03 1.20000000e+01]]

A_t * y:
[8.18034966e+09 3.05852260e+07 1.26492000e+05 6.44000000e+02]

Coefficient Results: [-3.99050448e-06  9.57687541e-04  1.87139469e-01  2.59417808e+01]
```

Figure 3: Enter Caption



```
EVALUTING P(t) for June 4th and December 25th
JUNE 4TH: t = 159
        P(t) = 63.86770803631609
DEC 25TH: t = 366
        P(t) = 27.076780223569678
```

Figure 4: Enter Caption

4 When Will Temperatures be 64.89 degrees

In order to solve this problem, we took our polynomicla fittted equation and set it equal to 64.89. We can tend move the 64.89 to the left hand side and solve for the root of the equation using secant's metho

$$P_3(t) = -3.99050448*10^{-6}t^3+9.57687541*10^(-4)t^2+0.187139469t+25.9417808-6$$

The Secant method finds the roots of a function by approximating the derivative of $f(x)$ finding the slope of the secant that goes through both two root guesses. In order to find 4-decimal accuracy in our solution, we will use the same inequality as in Newton's method for measuring the difference between consecutive iterations:

$$|x_n - x_{n-1}| < 10^{-4}$$

Algorithm 1: Secant Method 4-decimal Approximation

Input: Two initial guesses x_0 and x_1
Output: A root to the given function $f(x)$ with 4 decimal places accuracy and the number of iterations i

do
  $y1 \leftarrow f(x1)$;
  $y2 \leftarrow f(x2)$;
  $denom \leftarrow \frac{y2-y1}{x2-x1}$;
  $x3 = x2 - \frac{y2}{denom}$;
  $x1 = x2$;
  $x2 = x3$;
  $error \leftarrow |x2 - x1|$;
  $i \leftarrow i + 1$;
while $error \geq 10^-4$;
OUTPUT i;
return x2

Running this algorithm using Python we with two initial guesses of day $x_0 = 295$ and $x_1 = 284$, we got a solution of day 284.14873445879124.

With two more initial guess of day $x_0 = 164$ and $x_1 = 166$, we got a second solution of day 164.56671363868992.

Both of these solutions acn be need in Figure 1.

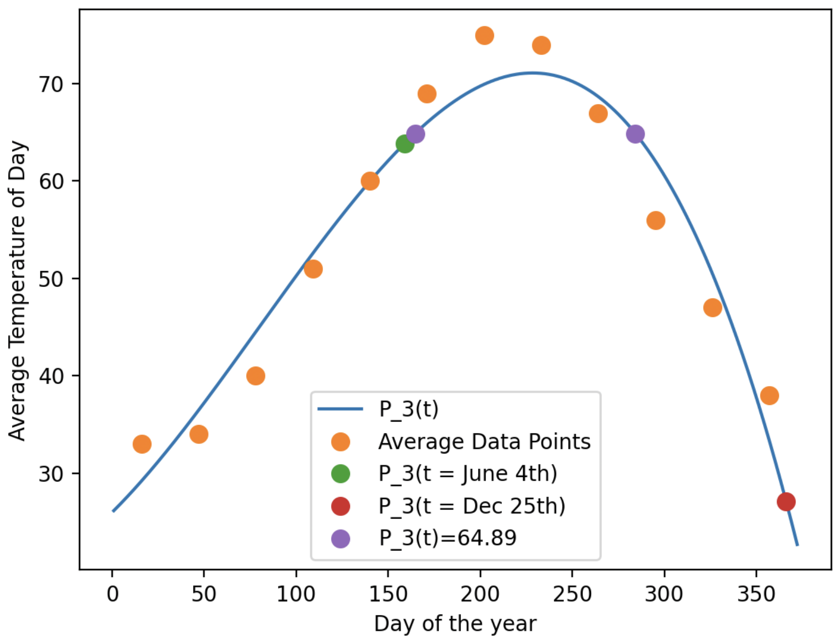


Figure 5: Enter Caption