Aarhus university

Computer-Science

Numerical Linear Algebra

Handin 3

Author
Søren M. Damsgaard

Student number
202309814

February 21, 2026

Polynomials for days!

(a) Plotting like a Caveman

(b) System of equations, but make it polynomial

(c) Vandermonde, Vandermonde, Vandermonde

(d) Too many polynumials, not enough time

(e) Sherlock Holmes and the One Solution

(f) Math Tariffs

(g) Judicial Bias

Appendix

```python
import numpy as np
import matplotlib.pyplot as plt


# a
time = np.array([2.0, 6.0, 9.0, 11.0, 12.0])
temp = np.array([25.0, 35.0, 45.0, 65.0, 70.0])

figA, drone = plt.subplots()
drone.plot(time, temp, 'o-')
drone.set_xlabel('Time(min)')
drone.set_ylabel('Temepratur( C )')
drone.set_title('Temeperatur of drone under operation #Ugly')



# b
# Do this part on paper

# c - polynomial interpolation

a = np.vander(time, len(time)) # The lenght is equal to 5 which
    corresponds to the exercise, which gives us 4 degrees (0,1,2,3,4)

# Augment and solve, This gives us the coefficients [a,b,c] see page 214
    in notes
coes = np.linalg.solve(a, temp)

t = np.linspace(2.0, 12.0, 100)

```

```
# This gives us p(x)
y_vals = np.vander(t, len(time)) @ coes


figC, vander = plt.subplots()
vander.plot(time, temp, 'o', label = 'Data pointies')
vander.plot(t, y_vals, label = '4 Degrees of Poly' )
vander.set_xlabel('Time(min)')
vander.set_ylabel('Temepratur( C )')



# d
# setup a a system of equaitons for both p_1 and p_2 and for checking the
    slope, take the derivative of them both,
#subtract the derivates and set them to 0


plt.show()
```

Listing 1: Python code for handin 3