HACETTEPE UNIVERSITY
ENGINEERING FACULTY
ELECTRICAL AND ELECTRONICS ENGINEERING PROGRAM

2023-2024
SPRING SEMESTER

ELE708
NUMERICAL METHODS IN ELECTRICAL ENGINEERING

HW7

N23239410 – Ali Bölücü

1) Computer Problems

1.a) 7.5

An experiment has produced the following data:

t 0.0 0.5 1.0 6.0 7.0 9.0

y 0.0 1.6 2.0 2.0 1.5 0.0

We wish to interpolate the data with a smooth curve in the hope of obtaining reasonable values of y for values of t between the points at which measurements were taken.

(a) Using any method you like, determine the polynomial of degree five that interpolates the given data, and make a smooth plot of it over the range $0 \leq t \leq 9$.

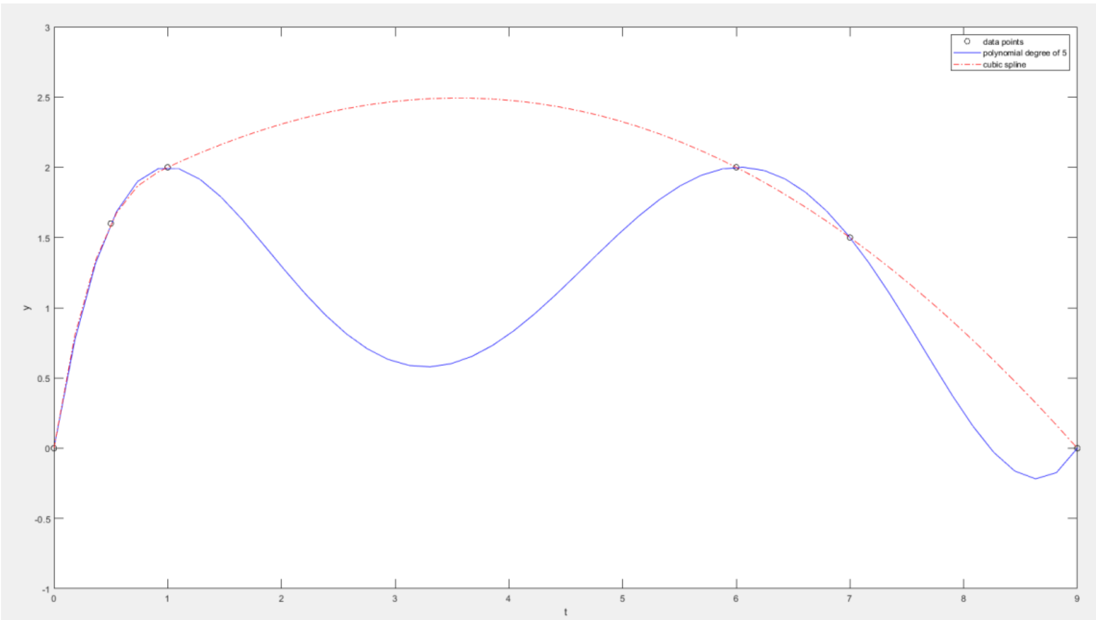
(b) Similarly, determine a cubic spline that interpolates the given data, and make a smooth plot of it over the same range.

(c) Which interpolant seems to give more reasonable values between the given data points? Can you explain why each curve behaves the way it does?

(d) Might piecewise linear interpolation be a better choice for these particular data? Why?

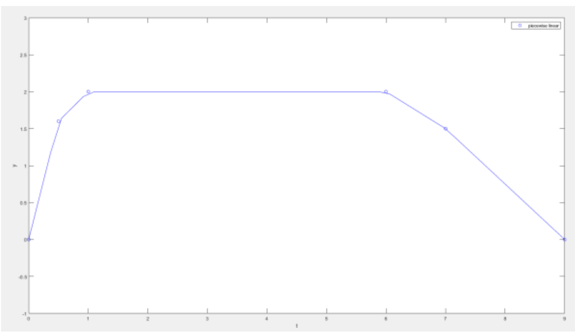
---

a,b)



c) Cubic spline is better fit and smoother than monomial polynomial. Since we picked the degree of the polynomial as 5, it oscillates according to it, making the degree lower than 5 increase the overshoot and making the degree higher than 5 increase the ossification.

d) It may a better choice because it creates reasonable values between the data points and the experiment may produce linear values.



1.b) 7.6

Interpolating the data points

t	0	1	4	9	16	25	36	49	64
y	0	1	2	3	4	5	6	7	8

should give an approximation to the square root function.

(a) Compute the polynomial of degree eight that interpolates these nine data points. Plot the resulting polynomial as well as the corresponding values given by the built-in sqrt function over the domain [0, 64].

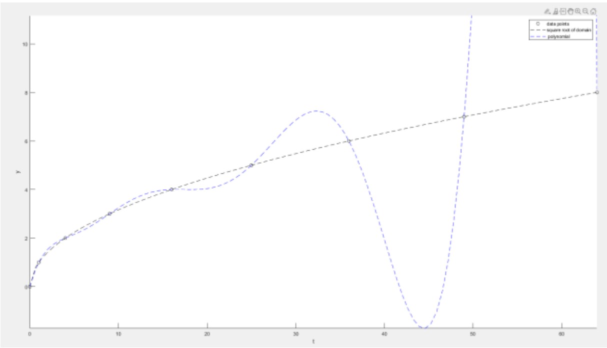
(b) Use a cubic spline routine to interpolate the same data and again plot the resulting curve along with the built-in sqrt function

(c) Which of the two interpolants is more accurate over most of the domain?

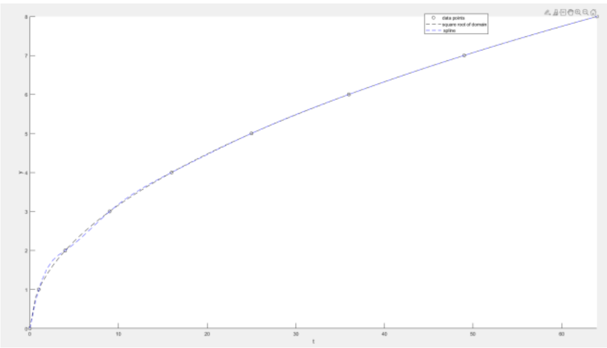
(d) Which of the two interpolants is more accurate between 0 and 1?

---

a)

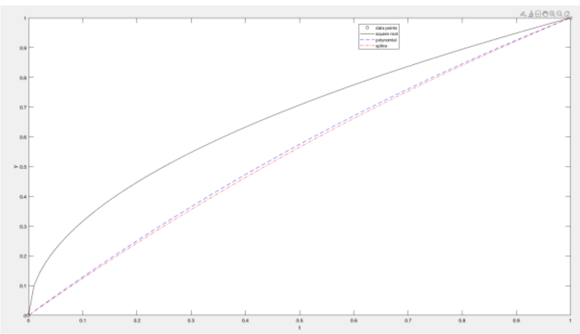


b)



c) Spline is definitely more accurate, polynomial is overshoots.

d) Both techniques is works fine



1.c) 7.8

Consider the following population data for the United States

Year	Population
1900	76,212,168
1910	92,228,496
1920	106,021,537
1930	123,202,624
1940	132,164,569
1950	151,325,798
1960	179,323,175
1970	203,302,031
1980	226,542,199

There is a unique polynomial of degree eight that interpolates these nine data points, but of course that polynomial can be represented in many different ways. Consider the following possible sets of basis functions ϕj(t), j = 1, . . . , 9: 1. ϕj(t) = tj−1

2. ϕj(t) = (t − 1900)j−1

3. ϕj(t) = (t − 1940)j−1

4. ϕj(t) = ((t − 1940)/40)j−1

(a) For each of these four sets of basis functions, generate the corresponding Vandermonde matrix and compute its condition number using a library routine for condition estimation. How do the condition numbers compare? Explain your results.

(b) Using the best-conditioned basis found in part a, compute the polynomial interpolant to the population data. Plot the resulting polynomial, using Horner's nested evaluation scheme to evaluate the polynomial at one-year intervals to obtain a smooth curve. Also plot the original data points on the same graph.

(c) Use a routine for Hermite cubic interpolation, such as pchip from MATLAB or Netlib, to compute a monotone Hermite cubic interpolant to the population data and again plot the resulting curve on the same graph.

(d) Use a cubic spline routine to interpolate the population data and again plot the resulting curve on the same graph.

(e) Extrapolate the population to 1990 using each of the polynomial, Hermite cubic, and cubic spline interpolants and compare the values obtained. How close are these to the true value of 248,709,873 according to the 1990 census?

(f) Determine the Lagrange interpolant to the same nine data points and evaluate it at the same yearly intervals as in parts b and c. Compare the total execution time with those for Horner's nested evaluation scheme and for evaluating the cubic spline.

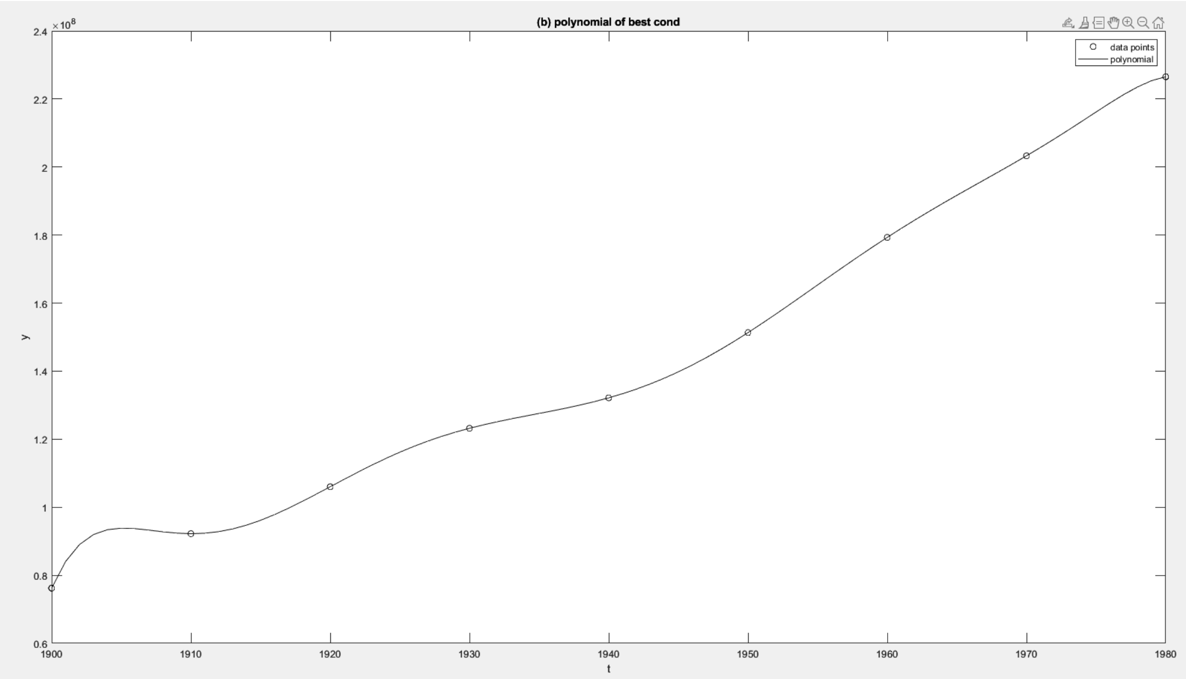
(g) Determine the Newton form of the polynomial interpolating the same nine data points. Now determine the Newton polynomial of one degree higher that also interpolates the additional data point for 1990 given in part d, without starting over from scratch (i.e., use the Newton polynomial of degree eight already computed to determine the new Newton polynomial). Plot both of the resulting polynomials (of degree eight and nine) over the interval from 1900 to 1990.

(h) Round the population data for each year to the nearest million and compute the corresponding polynomial interpolant of degree eight using the same basis as in part b. Compare the resulting coefficients with those determined in part b. Explain your results.

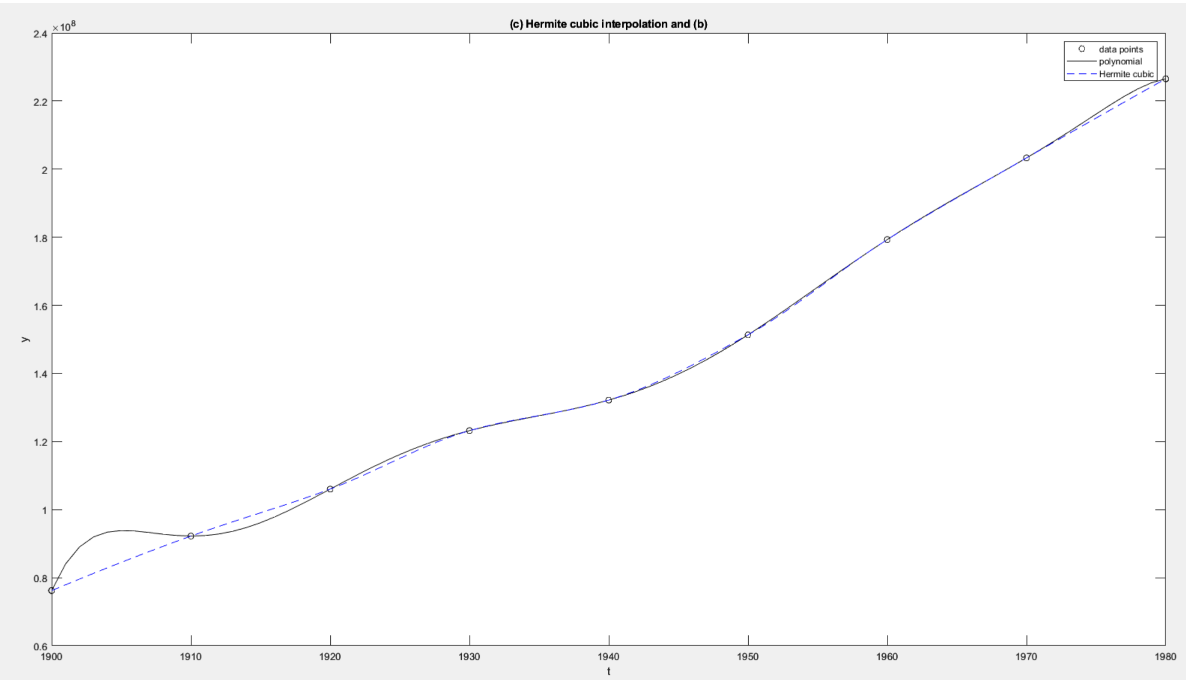
a) According to the answer, the last equation is more stable and less sensitive to small changes in the matrix.

```
(a) Condition number of basis matrices:
1: 1.068528e+42
2: 5994335190596687
3: 9.315536e+12
4: 1.605444e+03
(a) --------------------------------
```

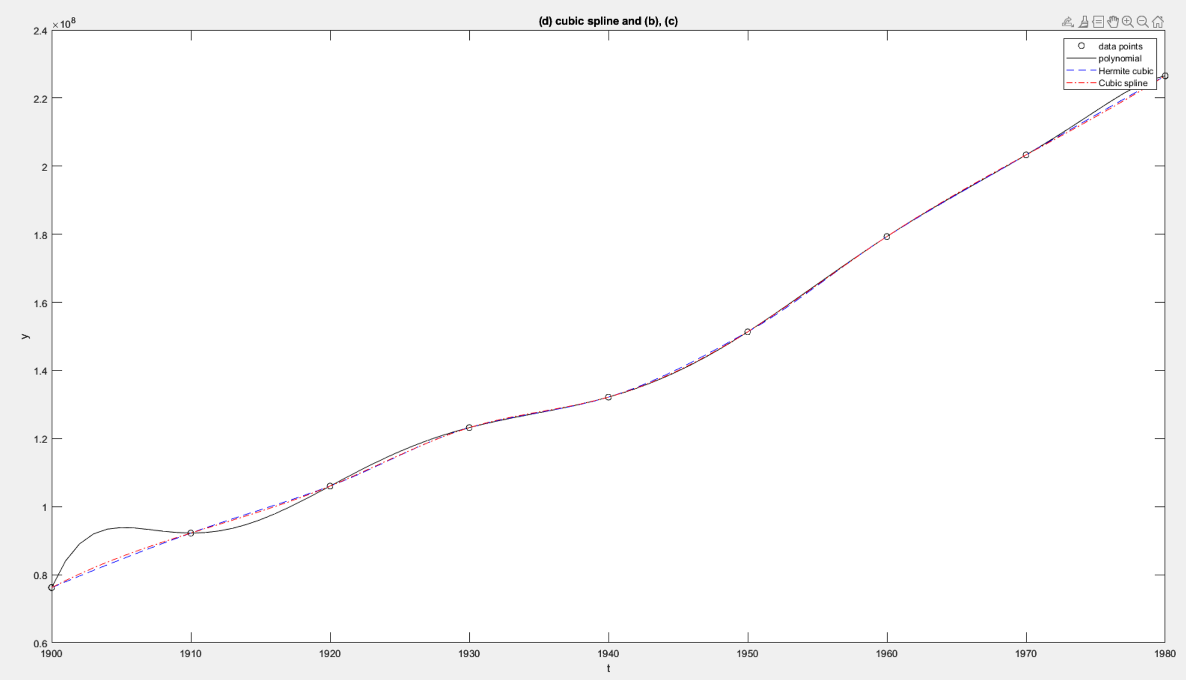
b)



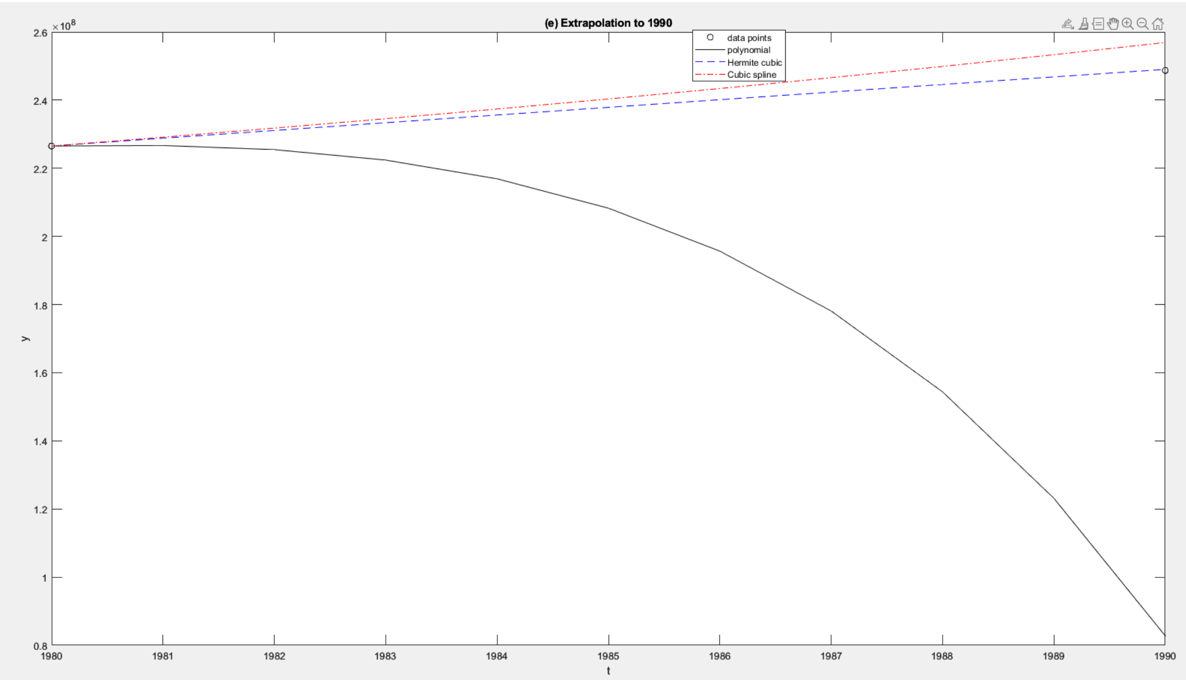
c)



d)

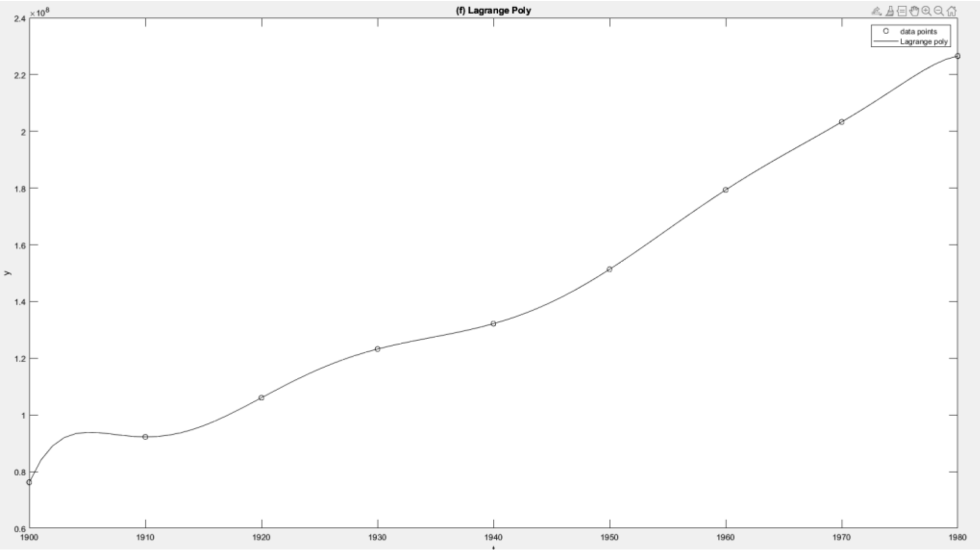

(d) cubic spline and (b), (c)
$\times 10^8$
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
y
1900
1910
1920
1930
1940
1950
1960
1970
1980
t
data points
polynomial
Hermite cubic
Cubic spline

e)

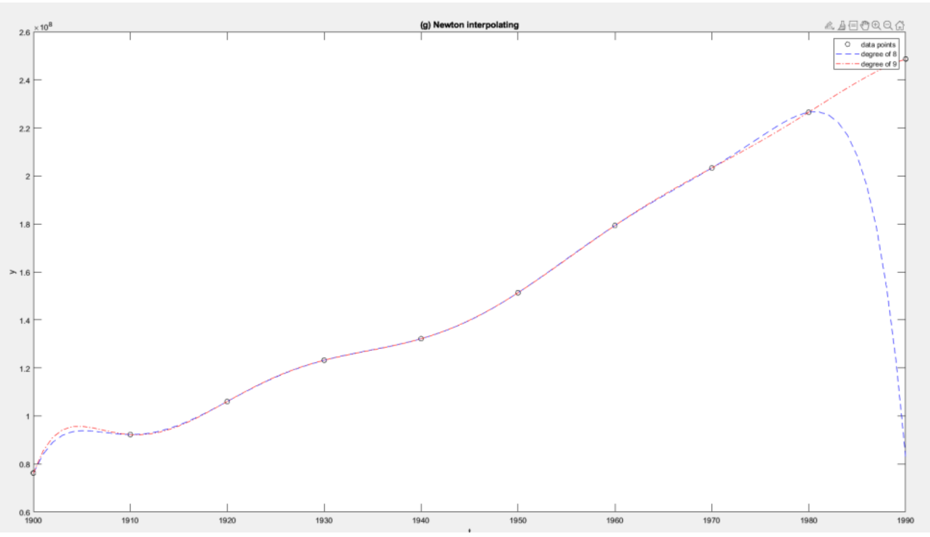

(e) Extrapolation to 1990
$\times 10^8$
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
y
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
t
data points
polynomial
Hermite cubic
Cubic spline

f)

```
(f) Timing of each basis:
Monomial basis: 0.0019931
Hermite cubic: 0.0006004
Cubic spline: 0.0007787
Newton basis: 0.0027882
Lagrange basis: 0.0022419
```



g)



h) All the coefficients is decreased.

```
(h) Coefficients:
Coefficients of original input
   1.0e+08 *

    1.3216
    0.4613
    1.0272
    1.8253
   -3.7461
   -3.4267
    6.0629
    1.8918
   -3.1518
```

```
Coefficients of rounded input
   1.0e+08 *

    1.3200
    0.4596
    1.0014
    1.8111
   -3.5676
   -3.3849
    5.7031
    1.8692
   -2.9420
```