

You are encouraged to work with other students on this assignment but you are expected to write and work on your own answers. You don't need to provide the name of students you worked with.

You are expected to submit

- (a) a zipped folder containing all your work online on Sakai so the grader can reproduce your results using only your online submission. **Due on Sakai, Wednesday, Nov. 6.**
- (b) a print-out copy of your obtained results (don't print out the *.dat files) for each problem. **Due in class, Wednesday, Nov. 6.**

Problem 1 : Lagrange interpolation (Gibbs phenomenon)

Recall the Lagrange interpolant of a function $f(x)$ can be expressed as the original form

$$P_n(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^n f(x_k)L_{n,k}(x) \quad (3.1),$$

or the Newton's divided difference form

$$P_n(x) = f[x_0] + \sum_{k=1}^n f[x_0, \dots, x_k](x - x_0) \cdots (x - x_{k-1}) \quad (3.10).$$

In this example, you will approximate the following Heaviside step function

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

on the interval $[-1, 1]$ using Lagrange interpolation. You may choose to either implement formula (3.1) or formula (3.10). (*Fact: the Heaviside step function has a discontinuity at $x = 0$. But Lagrange polynomial interpolation approximate the discontinuities poorly, which is the well-known Gibbs phenomenon.*)

- (a) Generate the Lagrange interpolant with $n = 5, 10$ using uniform $n + 1$ nodes on $[-1, 1]$, i.e., $x_k = -1 + \frac{2k}{n}$, $\forall k = 0, 1, \dots, n$.
 - Evaluate each Lagrange interpolant at the points `linespace(-1, 1, 100)` and save these values as a column vector in `A11_n5.dat` and `A11_n10.dat`, respectively.
 - Plot on the same figure the original function $f(x)$, along with the two Lagrange interpolants. Use legend to distinguish among curves.
 - Calculate the maximum absolute error of your two Lagrange interpolants. Do you observe the error decrease when n increase from 5 to 10?
- (b) Generate the Lagrange interpolant with $n = 20, 40$ using following non-uniform $n + 1$ nodes (*known as the Chebyshev nodes*) on $[-1, 1]$: $x_k = -\cos(\frac{k\pi}{n})$, $\forall k = 0, 1, \dots, n$.
 - Evaluate each Lagrange interpolant at the points `linespace(-1, 1, 200)` and save these values as a column vector in `A12_n20.dat` and `A12_n40.dat`, respectively.
 - Plot on the same figure the original function $f(x)$, along with the two Lagrange interpolants. Use legend to distinguish among curves.
 - Calculate the maximum absolute error of your two Lagrange interpolants. Do you observe the error decrease when n increase from 20 to 40?

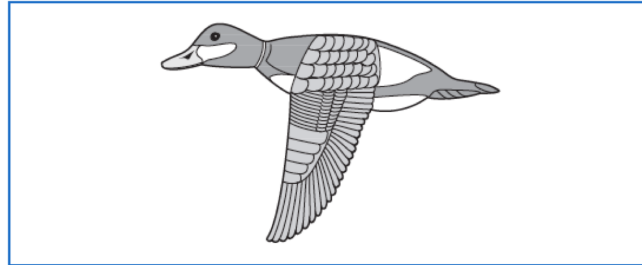
For this problem, please (1) include all your code and generated .dat files in the online submission, and (2) report the plots and answers to the last bullet of (a) and (b) in your print-out submission.

Problem 2 : cubic spline vs. Lagrange interpolation

(This problem is the illustration example in Page 154 of the textbook)

The top portion of the following picture of a flying duck is to be approximated by a curve, which is either a natural cubic spline or a Lagrange polynomial interpolation.

Figure 3.11

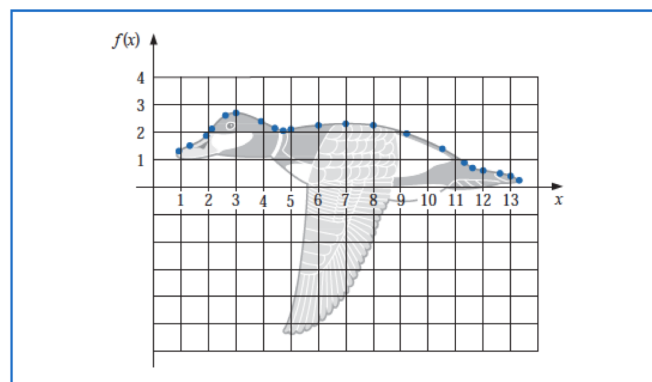


The following table lists the coordinates of 21 data points relative to the coordinate system shown in Figure 3.12.

Table 3.18

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

Figure 3.12

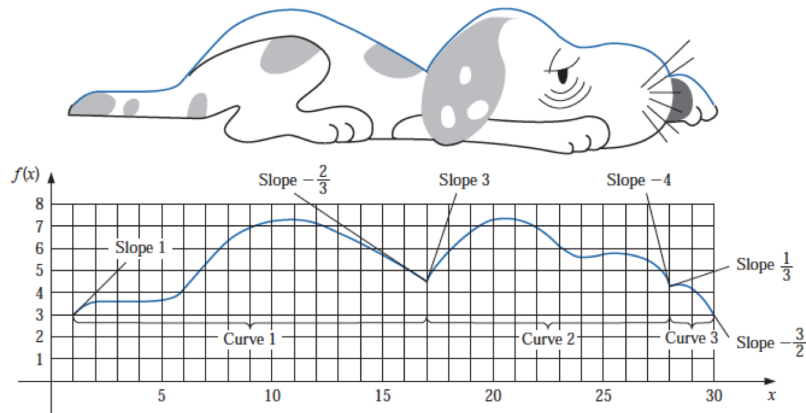


- Generate a natural cubic spline for the above data in Table 3.18. Evaluate the curve at the points `linspace(0.9,13.3,100)` and save these values as a column vector in `A21.dat`. Plot your obtained curve along with the interpolation nodes, which shall be similar to Figure 3.13 in the textbook.
NOTE: The coefficients for the natural cubic spline are also provided in Table 3.19 in the textbook. This table can serve as a means to validate the correctness of your implementation, but you are not allowed to copy data from Table 3.19. You will receive no points for this problem if data is simply copied from Table 3.19.
- Generate a 20th order Lagrange interpolant for the above data in Table 3.18. Evaluate the curve at the points `linspace(0.9,13.3,100)` and save these values as a column vector in `A22.dat`. Plot your obtained curve along with the interpolation nodes, which shall be similar to Figure 3.14 in the textbook.

For this problem, please (1) include all your code and generated `.dat` files in the online submission, and (2) report the plots in your print-out submission.

Problem 3 : Clamped cubic spline

The top portion of this picture of a dog is to be approximated by *three curves*, each of which is a *clamped cubic spline interpolant*.



The set of coordinates along the profile that we want to approximate is given in the following table.

Curve 1				Curve 2				Curve 3			
i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$
0	1	3.0	1.0	0	17	4.5	3.0	0	27.7	4.1	0.33
1	2	3.7		1	20	7.0		1	28	4.3	
2	5	3.9		2	23	6.1		2	29	4.1	
3	6	4.2		3	24	5.6		3	30	3.0	-1.5
4	7	5.7		4	25	5.8					
5	8	6.6		5	27	5.2					
6	10	7.1		6	27.7	4.1	-4.0				
7	13	6.7									
8	17	4.5	-0.67								

The point (17, 4.5) separates Curve 1 and Curve 2; the point (27.7, 4.1) separates Curve 2 and Curve 3. The slopes of each curve at its two endpoints are shown in the diagram.

- Evaluate Curve 1 at the points `linspace(1,17,80)` and save these values as a column vector in `A31.dat`
- Evaluate Curve 2 at the points `linspace(17,27.7,80)` and save these values as a column vector in `A32.dat`
- Evaluate Curve 3 at the points `linspace(27.7,30,30)` and save these values as a column vector in `A33.dat`
- Plot your obtained curve using all the above points, along with the interpolation nodes on the same figure. Your curve shall be an approximation of the curve in the above figure. (Please add the line `axis equal; axis([0, 30, 0,8]); grid on;` after your plot command for better visualization.)

For this problem, please (1) include all your code and generated `.dat` files in the online submission, and (2) report the plot in your print-out submission.