Numerical Analysis and Computational Mathematics

Fall Semester 2025 – CSE Section

Prof. Laura Grigori

Assistant: Israa Fakih

Session 6 – October 15, 2025

Approximation of functions and data

Exercise I (MATLAB)

Consider a digital measuring instrument that samples a signal expressed by the function $g(x)$ for $x \in I = [a, b] = [0, 1]$, where $g(x) = 10x^2$. The output of the instrument is affected by noise and can be represented by a function $f(x) = g(x) + \varepsilon(x)$, where $|\varepsilon(x)| \leq 1$. In MATLAB, we consider the following commands:

```
g = @(x) 10 * x.^2;                % True signal
f = @(x) g(x) + 2*rand(size(x))-1; % Instrument output
```

The noise $\varepsilon(x)$ is represented by the command `2*rand(size(x))-1`, which returns a random vector (with values in $[-1, 1]$) of the same size as `x`. Note that the function `f` returns a different realization each time (even if the variable `x` is the same).

a) Compute the polynomial $\Pi_n f(x)$ for $n = 9$, interpolating $f(x)$ at $n+1$ uniformly spaced nodes over I by using the MATLAB functions `polyfit` and `polyval`. Then, by using the same nodes, compute the least-squares polynomial $\widetilde{f}_m(x)$ of degree $m = 2$ approximating $f(x)$. Plot in the same figure the functions $f(x)$, $g(x)$, and the polynomials $\Pi_n f(x)$ and $\widetilde{f}_m(x)$. Which method produces better representations of the original signal?

b) Use the polynomials $\Pi_9 f(x)$ and $\widetilde{f}_2(x)$ to extrapolate the value of $f(x)$ at $x = 2$. Compare and discuss the results obtained.

c) Due to the noise, repeating of the measurement typically yields a different signal $f(x)$. In MATLAB, at each call of the function `f`, we obtain different values due to the use of the function `rand`. In this way, we can study the stability of the polynomials $\Pi_n f(x)$ and $\widetilde{f}_m(x)$ by analyzing the variations in the results using different sets of data points $\{(x_i, f(x_i))\}_{i=0}^{9}$. Repeat the points a) and b) several times and discuss the results obtained.

Exercise II (MATLAB)

Consider the function $f(x) = e^{-x^2/2}$ over the interval $I = [a, b] = [-5, 5]$.

a) Using the MATLAB function `interp1`, compute and plot the piecewise linear interpolants $\Pi_1^H f(x)$ on n uniform sub-intervals of size $H = \frac{b-a}{n}$, for $n = 2, 7, 12, 22, 27, 32$. Compare graphically the interpolating functions with $f(x)$.

b) Following point a), compute the errors $e_1^H(f) := \max_{x \in I} |f(x) - \Pi_1^H f(x)|$ corresponding to the piecewise linear interpolants $\Pi_1^{H_n} f(x)$ for $n = 2, 3, \ldots, 32$ uniform sub-intervals of size H_n. Plot $e_1^{H_n}(f)$ vs. n. Motivate the result obtained.

c) The not-a-knot interpolating cubic spline $s_3(x)$ can be computed in MATLAB by using the command `spline` (see `help spline`). Compute the spline $s_3(x)$ approximating $f(x)$ by subdividing the interval I in $n = 7$ uniform sub-intervals. Compare it graphically with $f(x)$ and with the piecewise linear interpolant $\Pi_1^H f(x)$. Finally, plot the errors $e_{s_3}(f) := \max_{x \in I} |f(x) - s_3(x)|$ as a function of n for $n = 2, 3, \ldots, 32$.

Exercise III (Theoretical)

Consider the function $f(x) = \frac{1}{1+x}$ over the interval $I = [a, b] = [0, 5]$.

a) We consider the piecewise linear interpolant $\Pi_1^H f(x)$ over n uniform sub-intervals of size $H = \frac{b-a}{n}$. Compute the minimum number n such that the error associated to $\Pi_1^H f(x)$ is smaller than 10^{-3}.

b) Repeat point a) by considering the natural interpolating cubic spline $s_3(x)$. To this aim, recall that, for $f \in C^4(I)$,

$$\max_{x \in I} \left| f^{(r)}(x) - s_3^{(r)}(x) \right| \leq C_r H^{4-r} \max_{x \in I} \left| f^{(4)}(x) \right|, \qquad \text{for } r = 0, 1, 2,$$

for some positive constants C_r depending on r. For simplicity, you may assume that $C_r \simeq 1$.

What is the minimum number of uniform sub-intervals that ensures that the error associated to the first derivative of $s_3(x)$ is smaller than 10^{-3}?

c) Consider the nodes $x_0 = 0$, $x_1 = 1/2$, $x_2 = 1$. Calculate the least-squares degree-1 polynomial $\widetilde{f}_1(x) = a_0 + a_1 x$ that approximates $f(x)$ at such nodes. To this aim, you should find the coefficients a_0 and a_1 that minimize the discrete interpolation error:

$$\Phi(a_0, a_1) = \sum_{i=0}^{2} [f(x_i) - \widetilde{f}_1(x_i)]^2 = \sum_{i=0}^{2} [f(x_i) - a_0 - a_1 x_i]^2.$$

Exercise IV (MATLAB)

The left ventricle is the chamber of the heart responsible for pumping oxygenated blood into the major organs. We have obtained a time series of $n+1 = 11$ data points $\{(t_j, V_j)\}_{j=0}^n$ describing the volume V_j (in cm^3) of a left ventricle at time t_j (in s):

j	0	1	2	3	4	5	6	7	8	9	10
t_j [s]	0.00	0.07	0.14	0.21	0.28	0.35	0.42	0.49	0.56	0.63	0.70
V_j [cm^3]	194	184	177	156	142	160	168	166	170	178	187

We wish to find an interpolating curve $I_tV(t)$ that describes the volume $V(t)$ of the left ventricle as it contracts during the heartbeat. Since the volume of the ventricle $V(t)$ can be considered as a periodic function of the time t with period T equal to the length of the heartbeat (here $T = 0.77$ s), we decide to use *trigonometric interpolation*.

a) By using the MATLAB command `interpft`, interpolate the data points by means of trigonometric interpolation and evaluate the interpolant on 1100 uniformly spaced points. Plot the data points and the trigonometric interpolant $I_tV(t)$ on the same figure. Then, by using the trigonometric interpolant $I_tV(t)$, evaluate the *ejection fraction* (percentage of volume of blood ejected during one heartbeat) defined by the formula:

$$E_f = \frac{\max\limits_{t\in[0,T]} I_tV(t) - \min\limits_{t\in[0,T]} I_tV(t)}{\max\limits_{t\in[0,T]} I_tV(t)}.$$

b) We observe that in a real ventricle the volume $V(t)$ first decreases monotonically during the contraction of the muscle (for $t \in [0, 0.3]$ s) and then increases monotonically during the relaxation of the muscle (for $t \in (0.3, 0.77]$ s) due to the presence of two valves that block the flow in the wrong direction. This property is not satisfied by the interpolant $I_tV(t)$ due to the presence of the erroneous data point $(t_7, V_7) = (0.49, 166)$, which is probably a consequence of measurement error. In trigonometric interpolation, we assume that the nodes are uniformly spaced, so that we cannot simply remove the bad data point (t_7, V_7) from the data set. Instead, we define a new series of data points $\{(t_j, \widetilde{V}_j)\}_{j=0}^n$ by replacing the erroneous value with the average of its two nearest neighbors:

$$\widetilde{V}_j := \begin{cases} V_j & \text{if } j \neq 7, \\ \frac{V_{j-1}+V_{j+1}}{2} & \text{if } j = 7. \end{cases}$$

Compute the corrected trigonometric interpolant $I_t\widetilde{V}(t)$, evaluate the interpolant at 1100 equally spaced points, and plot it in the same figure as $I_tV(t)$. Compute also the new estimated ejection fraction

$$\widetilde{E}_f = \frac{\max\limits_{t\in[0,T]} I_t\widetilde{V}(t) - \min\limits_{t\in[0,T]} I_t\widetilde{V}(t)}{\max\limits_{t\in[0,T]} I_t\widetilde{V}(t)}.$$

Are the results considerably different from those from a)?

Exercise V (Theoretical)

Consider $B \in \mathbb{R}^{n\times m}$, $\mathbf{y} \in \mathbb{R}^n$, and $\mathbf{a} \in \mathbb{R}^m$, with $m < n$, defining the linear system

$$B\mathbf{a} = \mathbf{y}.$$

Show that the least-square solution $\tilde{\mathbf{a}} \in \mathbb{R}^m$ satisfies the normal equations

$$B^T B\tilde{\mathbf{a}} = B^T\mathbf{y}.$$