Interpolation

Question 1 **50 marks**

Consider the problem of interpolating the data

k	0	1	2	3
x_k	-1.20	0.500	2.20	3.100
y_k	-0.400	1.20	5.50	10.2

using a polynomial interpolant $P_3(x)$

(**a**) Write out the interpolating conditions that $P_3(x)$ satisfies. Follow the recipe on slide 11 of lecture 14.

Taking the basis $\phi_0(x) = 1$, $\phi_1(x) = x + 1.2$, $\phi_2(x) = (x + 1.2)(x - 0.5)$, $\phi_3(x) = (x + 1.2)(x - 0.5)(x - 2.2)$, and $\Pi_3(x) = a_0\phi_0(x) + a_1\phi_1(x) + a_2\phi_2(x) + a_3\phi_3(x)$, we obtain

$$\begin{aligned}
\Pi_3(x_0) &= a_0 = y_0 = -0.4\\
\Pi_3(x_1) &= a_0 + 1.70a_1 = y_1 = 1.20\\
\Pi_3(x_2) &= a_0 + 3.40a_1 + 5.78a_2 = y_2 = 5.50\\
\Pi_3(x_3) &= a_0 + 4.30a_1 + 11.18a_2 + 10.062a_3 = y_3 = 10.2
\end{aligned}$$

(**b**) Construct the linear system for the interpolation data in the table above.

$$\begin{bmatrix} 1 & 0 & 0 & 0\\ 1 & 1.70 & 0 & 0\\ 1 & 3.40 & 5.78 & 0\\ 1 & 4.30 & 11.18 & 10.062 \end{bmatrix} \begin{bmatrix} a_0\\ a_1\\ a_2\\ a_3 \end{bmatrix} = \begin{bmatrix} -0.400\\ 1.20\\ 5.50\\ 10.2 \end{bmatrix}$$

(**c**) Solve the linear system for the interpolation coefficients (you may use NumpPy).

With forward substitution, starting from $a_0 = -0.4$ and working our way down. We find $a_0 = -0.4$, $a_1 = 0.9412$, $a_2 = 0.4671$, $a_3 = 0.1322$. I have kept one digit more than then specified for the interpolation data to avoid unnecessary round-off error.

(**d**) Produce a plot that shows the interpolation data in the table with points as well as the graph of the interpolant. Make sure the axes are labeled and the scale and limits of the axes are appropriate (extending slightly to the left of x_0 and to the right of x_3). Include the figure in your LaTeX submission and add a caption that explains the figure in one or two sentences.

Question 2 **50 marks**

Consider the functions

$$f(x) = \frac{1}{x+1} \qquad g(x) = \exp(-x)$$

And consider the interpolation nodes $x_j = j/(2N)$ for $j = 0, \ldots, N$.

(**a**) For each function, use the theorem in lecture 14 to find an upper bound for the interpolation errors

$$E_f = \max_{x\in[0,1/2]} |P_N(x) - f(x)| \qquad E_g = \max_{x\in[0,1/2]} |P_N(x) - g(x)|$$

Make sure your error bound decreases at least exponentially with N. Remember that *the extrema of a differentiable function of a single variable on a finite interval occur there where its derivative is zero and/or at the end points of the interval.*

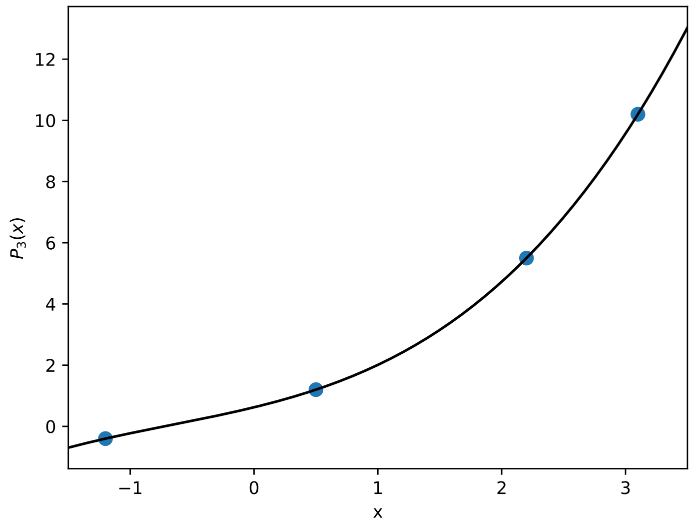


Figure 1: The interpolation data from the table in question 1 along with the graph of the interpolating polynomial $P_3(x)$.

For the derivatives, we have

$$\frac{d^k f}{dx^k} = (-1)^k \frac{k!}{(x+1)^{k+1}} \qquad \frac{d^k g}{dx^k} = (-1)^k \exp(-x)$$

Inserting this into the expression for the error, we find for f that

$$\begin{aligned} E_f &= \max_{x\in[0,1/2]} |P_N(x) - f(x)| = \max_{x,\xi\in[0,1/2]} \left| \frac{f^{(N+1)}(\xi)}{(N+1)!}\, x \left(x - \frac{1}{2N}\right) \cdots \left(x - \frac{1}{2}\right) \right| \\ &= \max_{x,\xi\in[0,1/2]} \frac{1}{(\xi+1)^{N+2}}\, x \left|x - \frac{1}{2N}\right| \cdots \left|x - \frac{1}{2}\right| \leq \left(\frac{1}{2}\right)^{N+1} \end{aligned}$$

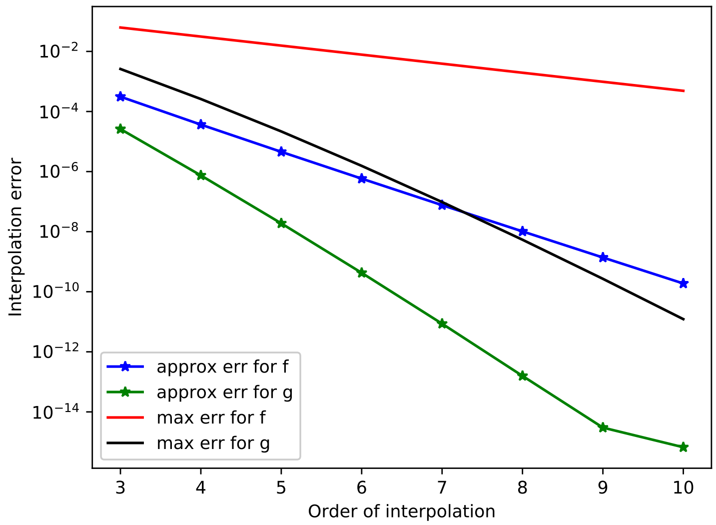
we I exploited the fact that the function of ξ is monotonically decreasing and $|x - j/(2N)| \leq 1/2$ for $x \in [0, 1/2]$ and $j = 0, \ldots, N$.
For g, we find analogously

$$\begin{aligned} E_g &= \max_{x\in[0,1/2]} |P_N(x) - g(x)| = \max_{x,\xi\in[0,1/2]} \left| \frac{g^{(N+1)}(\xi)}{(N+1)!}\, x \left(x - \frac{1}{2N}\right) \cdots \left(x - \frac{1}{2}\right) \right| \\ &= \max_{x,\xi\in[0,1/2]} \frac{\exp(-\xi)}{(N+1)!}\, x \left|x - \frac{1}{2N}\right| \cdots \left|x - \frac{1}{2}\right| \leq \frac{1}{(N+1)!} \left(\frac{1}{2}\right)^{N+1} \end{aligned}$$

(**b**) Write a script that does the following for $N = 3, \ldots, 20$:

1. compute the interpolating polynomials $(\Pi_N f)(x)$ and $(\Pi_N g)(x)$ (we will write code for this purpose in class);
2. estimates the interpolation error for each function by finding the maximum of $E(x)$ on a fine grid;
3. evaluates your upper bounds from part (a) and stores them along with the error estimate from (b2);
4. after completing the loop over values for N, produces a plot of the error bounds and estimates as a function of N.

See the course_codes repository. Here is the result:



Note, that I plotted on a lin-log scale because or error bounds have a factor exponential in N. Apparently, the (approximate) actual error also decreases exponentially to a good approximation.

(**c**) Based on (a)-(b), which function do you think produces interpolation data more amenable to polynomial interpolation?

Clearly, both the actual error and the upper bound for the error decrease much faster for the exponential function than for the rational function.