Problem Sheet 2 (CSE/Inf/Geo)

discussion: Tuesday, 24.10.2022

2.1. Use quadratic interpolation in the points $(0, 1), (1, e), (2, e^2)$ to approximate the function $f(x) = \exp(x)$ on $[0, 2]$. What is the error at $x = 1.2$? Compute an upper estimate for the pointwise error

$$|f(x) - p(x)|$$

for any fixed point $x \in [0, 2]$ using the error formula (Thm. 1.17) from the lecture notes. Also compute an upper estimate for the maximal error, i.e., for

$$||f - p||_{\infty,[0,2]}.$$

2.2. **a)** Write a `matlab` or `python` program that realizes the Neville scheme. Input are the vectors **x**, **f** (knots and data values) of length $n+1$. Output is an array N (size $(n+1)\times(n+1)$) that contains the columns of the Neville scheme.

Remark: use the algorithm described in the lecture; you could check, if you wish, your algorithm with your code of Problem 1.3.

b) Let, for a function f and a point x_0,

$$D_{sym}(h) := \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

be the symmetric difference quotient.

Use your program of a) to generate the first 4 columns of the Neville scheme of the extrapolation of $D_{sym}(0)$ for the function $f(x) = \tan(x)$ and $x_0 = \pi/4$ (*note:* $f'(x) = 1/\cos^2(x)$ so that $f'(\pi/4) = 2$) and $h_i = 2^{-i}$, $i = 0, \ldots, n_{max}$ with $n_{max} = 10$. Plot in a `loglog` plot the error versus h for these 4 columns, i.e., plot for $m \in \{0, 1, 2, 3\}$ the values $\texttt{abs}(N([1 : n_{max} + 1 - m], m) - 2)$ versus $h([1 : n_{max} + 1 - m])$. Include in the plot the auxiliary lines $h \mapsto h^2$, $h \mapsto h^3$, $h \mapsto h^4$. What convergence rates do you observe?

2.3. Consider the function $f(x) = \frac{1}{4-x^2}$.

Provide an upper estimate for the best-approximation error by polynomials of maximal degree n

$$\min_{q\in\mathcal{P}_n} \|f - q\|_{\infty,[-1,1]}$$

by selecting q to be the Taylor polynomial of f about a suitable point.

Plot semilogarithmically (`semilogy`) the error $\|f - I_n^{Cheb} f\|_{\infty,[-1,1]}$ versus $n \in \{1, \ldots, 20\}$, where $I_n^{Cheb} f$ is the Chebyshev interpolant of degree n. Herby, compute the quantity $\|f - I_n^{Cheb} f\|_{\infty,[-1,1]}$ approximatively by evaluating $f - I_n^{Cheb} f$ in 100 uniformly distributed points in the interval $[-1, 1]$ and take the maximum of these values (you can use the Neville scheme from Ex.2.2 for that!).

Include in your graph also the error $\|f - T_{2n} f\|_{\infty,[-1,1]}$ where $T_{2n} f$ is the Taylor polynomial of f about $x = 0$ of degree $2n$. Compare the approximation quality of the Taylor polynomial with that of the Chebyshev interpolant.

2.4. ("harmonic series") The goal is the efficient evaluation/approximation of

$$S(N) := \sum_{n=1}^{N} \frac{1}{n}$$

for large N. We use the fact that $S(N)$ can be written as

$$S(N) = \ln N + a_0 + \frac{a_1}{N} + \frac{a_2}{N^2} + \cdots \tag{1}$$

Determine the coefficients a_0, a_1, a_2 as follows: 1) Write a routine to evaluate $S(N)$. 2) Set up a linear system of equations for the coefficients a_0, a_1, a_2 that is obtained for $N = 10$, 100, 1000. (The terms

$+\cdots$ in (1) are simply ignored). Solve for the coefficients (in `matlab` this is achieved with \, in `python` this can be done with `numpy.linalg.solve`).

What is the error of your approximation for $N = 10^6$ and $N = 10^8$? What is the run time of your approximation for $N = 10^8$ and $N = 10^9$? What is the run time for the evaluation of $S(10^8)$ and $S(10^9)$ on your computer? (Use `tic`, `toc` or `time.time()`)

2.5. Let the data $(0,0),(1,1),(2,1),(3,2)$ be given.

a) How many unknowns does the cubic spline with a knot-a-knot condition have? Write down the linear system of equations that has to be solved to compute this cubic spline!

b) Solve the linear system (not necessarily by hand!) and write down and draw the cubic spline as a function.

c) Assume that the given values $f_i = f(x_i)$ above are evaluations of a given function f with $\|f^{(4)}\|_{\infty,[0,3]} \leq 2$. Write down an upper bound for the error

$$\|f - p\|_{\infty,[0,3]},$$

where p denotes the cubic spline.