Brief Answers to Quiz Quiz 03

Nov 17, 2020.

1. Let P_n be the degree n interpolating polynomial of $\cos(2x)$ on the uniformly spaced nodes $x_0, \cdots, x_n$ on $[0,1]$ with $x_j = jh$, $h = 1/n$. Is it true that

$$\max_{0\le x\le 1} |\cos(2x) - P_n(x)| \to 0 \quad \text{as } n \to \infty?$$

Give all details and explain.

Ans:

Yes. **(2 pts)**
Let $f(x) = \cos(2x)$. Then

$$\begin{aligned}
\max_{0\le x\le 1} |\cos(2x) - P_n(x)| &= \max_{0\le x\le 1} \left|\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n)\right| \textbf{ (8 pts)} \\
&\le \max_{0\le \xi\le 1} \left|\frac{f^{(n+1)}(\xi)}{(n+1)!}\right| \max_{0\le x\le 1} \left|(x-x_0)(x-x_1)\dots(x-x_n)\right| \\
&\le \frac{2^{n+1}}{(n+1)!} n! \frac{1}{n^{n+1}} \textbf{ (8 pts)} \\
&= \frac{1}{n+1}\left(\frac{2}{n}\right)^{n+1} \\
&\le \frac{1}{n+1} \to 0 \text{ as } n \to \infty \textbf{(2 pts)}.
\end{aligned}$$

2. Denote by $P_{0,1,\cdots,k}(x)$ the Lagrange interpolating polynomial on the data set $(x_0, f(x_0)), (x_1, f(x_1), \cdots (x_k, f(x_k))$. Express $P_{0,1,\cdots,k}$ in terms of $P_{0,1,\cdots,j-1,j+1,\cdots,k}$ and $P_{0,1,\cdots,i-1,i+1,\cdots,k}$. Then verify your answer is indeed the Lagrange interpolating polynomial.

Ans:

$$P_{0,1,\cdots,k}(x) = \frac{(x-x_j)P_{0,1,\cdots,j-1,j+1,\cdots,k}(x) - (x-x_i)P_{0,1,\cdots,i-1,i+1,\cdots,k}(x)}{x_i - x_j}. \textbf{ (8 pts)}$$

For $t \ne i$ and $t \ne j$,

$$P_{0,1,\cdots,k}(x_t) = \frac{(x_t-x_j)f(x_t) - (x_t-x_i)f(x_t)}{x_i - x_j} = f(x_t).$$

For $t = i$,

$$P_{0,1,\cdots,k}(x_i) = \frac{(x_i-x_j)f(x_i)}{x_i - x_j} = f(x_i).$$

For $t = j$,

$$P_{0,1,\cdots,k}(x_j) = \frac{-(x_j-x_i)f(x_j)}{x_i - x_j} = f(x_j).$$

Finally, $\deg P_{0,1,\cdots,j-1,j+1,\cdots,k} \le k-1$ and $\deg P_{0,1,\cdots,i-1,i+1,\cdots,k} \le k-1$ imply that

$$\deg P_{0,1,\cdots,k} \le k. \textbf{ (12 pts)}$$

3. A natural cubic spline S on $[0, 2]$ is defined by

$$S(x) = \begin{cases} S_0(x) = 1 + 2x - x^3, & 0 \le x \le 1 \\ S_1(x) = 2 + b(x-1) + c(x-1)^2 + d(x-1)^3, & 1 \le x \le 2 \end{cases}$$

Find b, c, d.

Ans: Compute S_0', S_1', S_0'', and S_1''. **(2 pts)**

$$S_0'(1) = S_1'(1) \text{ (3 pts)} \Rightarrow b = -1 \text{ (3 pts)}$$
$$S_0''(1) = S_1''(1) \text{ (3 pts)} \Rightarrow c = -3 \text{ (3 pts)}$$
$$S_1''(2) = 0 \text{ (3 pts)} \Rightarrow d = 1 \text{ (3 pts)}$$

4. Suppose that we are to construct a piecewise polynomial interpolation $S(x)$ on the data $(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n))$, with additional continuity conditions for S', S'' and S''' on the interior nodes $x_1, \cdots, x_{n-1}$. If we use polynomials of the same degree on each of the interval $[x_0, x_1], \cdots, [x_{n-1}, x_n]$, what is the minimal degree needed in each interval? How many additional end conditions are needed? Count carefully and explain (give details).

Ans:

Values of S at x_0, x_n: one condtion each.

Values of S at $x_1, \cdots x_{n-1}$: two conditions each.

Continuity of S' at $x_1, \cdots x_{n-1}$: one condition each.

Continuity of S'' at $x_1, \cdots x_{n-1}$: one condition each.

Continuity of S''' at $x_1, \cdots x_{n-1}$: one condition each.

(15 pts, 0 pts without details)

Total $5n - 3$ conditions. Therefore we require minimal $5n$ unknowns or degree 4 polynomials on each interval and additional 3 boundary conditions **(5 pts)**.

5. Given four data $(x_i, \exp(-2x_i))$: $(0.3, 0.5488)$, $(0.4, 0.4493)$, $(0.5, 0.3679)$ and $(0.6, 0.3012)$ (you should generate the data yourself to avoid typo in inputting data). Use Inverse Interpolation to find the root of $x = \exp(-2x)$. You can use any algorithm for Lagrange interpolation. Extra credits for Neville's method (or divided difference, if anyone knows it). After finding x, check yourself that $x = \exp(-2x)$ is indeed satisfied in case of a bug in your code. Need not show the last part.

Hand in code, put all data within the code so that it can be executed immediately.

Ans: Use Lagrange interpolation to interpolate the function $t = f(s)$ at $s = 0$ where $s_i = y_i - x_i$ and $t_i = x_i$ **(10 pts)**.

Result: $x^* = f(0) \approx 0.4263....$ **(10 pts)**.

C programming: **(extra 2 pts)**.