Poon, Keegan
poonkeeg
1002423727
CSCD37: Assignment #1

1. Recall the full-Newton algorithm for solving the nonlinear system $F(x) = 0; F, x \in \mathcal{R}^n$:

   Generate an initial approximation $\hat{x}_0$
   for $k = 0, 1 \ldots$ until convergence
   compute $-F(\hat{x}_k)$
   compute $\frac{\partial F(\hat{x}_k)}{\partial x}$
   solve $\frac{\partial F(\hat{x}_k)}{\partial x}\Delta_k = -F(\hat{x}_k)$
   update $\hat{x}_{k+1} = \hat{x}_k + \Delta_k$
   end for

   (a) This algorithm is computationally expensive. Give a detailed analysis of the cost using standard big-oh notation. You may use results from CSCC37; e.e., the cost (measuring flops) of the LU-factorization of an $n \times n$ matrix is $(1/3)n^3 + \mathcal{O}(n^2)$.

   (b) We briefly discussed in lecture the "quasi-Newton algorithm" for solvling nonlinear systems. Modify the algorithm above to take advantage of the optimizations we discussed. You do not need to implement the modifications . . . pseudo-code will suffice. Be careful to discuss both flop optimizations and convergence issues ("X-test" and "F-test" tolerance, maximum number of iterations, condition of Jacobian, how long to hold Jacobian fixed, etc.).

2. In lecture we derived the divided-difference (Newton) form of the interpolating polynomial for the simple interpolation problem. This question will investigate how the Newton polynomial can be used for osculatory interpolation.

   (a) Prove that

$$\lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} y[x_{i+k}, x_{i+k-1}, \ldots, x_i] = \frac{y^{(k)}(x_i)}{k!}$$

   Provided $y \in \mathcal{C}^k$.

   Proof. This will be proven using induction on the elements of the divided difference, i.e. for an indexed set of elements $\{a_0, a_1, a_2, \ldots, a_k\}$ it will be on k. For the case $k = 0$,

$$\lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} y[x_{i+j}] = \lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} y(x_{i+j}) \overset{\text{cont}}{=} y(x_i) = \frac{y^{(0)}(x_i)}{1!}$$

   For the inductive case, assume it holds for $n < k$

$$\begin{aligned}\lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} y[x_{i+k}, x_{i+k-1}, \ldots, x_i] &= \lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} \frac{y[x_{i+k}, x_{i+k-1}, \ldots, x_{i+1}] - y[x_{i+k-1}, x_{i+k-1}, \ldots, x_i]}{x_{i+k} - x_i}\\ &= \lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} \frac{y^{(k-1)}(x_{i+1})}{(x_{i+k} - x_i)(k-1)!} - \frac{y^{(k-1)}(x_i)}{(x_{i+k} - x_i)(k-1)!}\\ &= \frac{1}{(k-1)!} \lim_{\substack{x_{i+j}\to x_i\\1\le j\le k}} \frac{y^{(k-1)}(x_{i+1}) - y^{(k-1)}(x_i)}{x_{i+k} - x_i}\end{aligned}$$

   $\square$

   (b) The result in (a) tells us that divided differences can be replaced with derivatives as data points coincide. Using this result, construct a divided difference table to find the coefficients of the Newton polynomial of degree 6 or less that satisfies the following interpolation conditions:

$p(-1) = 4$	$p(0) = 7$	$p(1) = 28$	$p(2) = 247$
	$p'(0) = 6$	$p'(1) = 56$	
		$p''(1) = 140$	

   (c) Use the Method of Undetermined Coefficients (i.e., as discusseds in lecture, construct and solve an appropriate Vandermonde system) to find the coefficients of the monomial-basis polynomial of degree 6 or less that satisfies the interpolation conditions specified in (b).

   You may use *MatLab* for this question if you wish. Verify that you have obtained the same polynomial as in (b).

3. Consider the function $y \in \mathcal{C}^{n+1}$ and the polynomial $p \in \mathcal{P}_n$ which satisfies

$$p^{(j)}(x_i) = y^{(j)}(x_i);\ j = 0, \ldots, j_i;\ i = 0, \ldots, k;\ \sum_{i=0}^{k}(j_i + 1) = n + 1;$$

with all of the x_i distinct. The error in this polynomial interpolant is given by

$$E(x) = y(x) - p(x) = \frac{y^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{j_0+1}(x - x_1)^{j_1+1} \cdots (x - x_k)^{j_k+1} \tag{1}$$

where $\xi \in \text{span}\{x_0, \ldots, x_k, x\} = [\min\{x_0, \ldots, x_k, x\}, \max\{x_0, \ldots, x_k, x\}]$ provided $y \in \mathcal{C}^{n+1}$ on $\text{span}\{x_0, \ldots, x_k, x\}$. This is a fundemental formula in Numerical Aprroximation. The following is an outline of a possible derivation of (1). In this question you will expand this outline by proving certain key statements.

If $x = x_i$, $i = 0, \ldots, k$, then $y(x) - p(x) = 0$ since p interpolates y at these $k+1$ points. Also, $E(x_i) = 0$ in (1) since $(x_i - x_i)^{j_1+1} = 0$. Therefore, (1) holds when $x = x_i$.

Now assume $x \neq x_i$ for any $i = 0, \ldots, k$ and consider x *fixed*. Let $F(t) = y(t) - p(t) - CW(t)$ where $C = [y(x) - p(x)]/W(x)$ is a constant and

$$W(t) = (t - x_0)^{j_0+1}(t - x_1)^{j_1+1} \cdots (t - x_k)^{j_k+1} \tag{2}$$

is a polynomial of degree $n + 1$. Clearly $F(x) = 0$, and also

$$F^{(j)}(x_1) = 0;\ j = 0, \ldots, j_i;\ i = 0, \ldots, k. \tag{3}$$

Therefore, counting multiplicities, $F(t)$ has at least $n + 2$ zeros in $\text{span}\{x_0, \ldots, x_k, x\}$, which implies $F^{(n+1)}(t)$ has at least 1 zero in $\text{span}\{x_0, \ldots, x_k, x\}$, or, in other words,

$$F^{(n+1)}(\xi) = 0, \xi \in \text{span}\{x_0, \ldots, x_k, x\}. \tag{4}$$

But

$$F^{(n+1)}(t) = \frac{d^{n+1}}{dt^{n+1}}[y(t) - p(t) - CW(t)] = y^{(n+1)}(t) - (n + 1)!C. \tag{5}$$

Therefore,

$$F^{(n+1)}(\xi) = 0 \implies y^{(n+1)}(\xi) - (n + 1)!C = 0 \implies C = \frac{y^{(n+1)}(\xi)}{(n+1)!}.$$

But

$$C = \frac{y(x) - p(x)}{W(x)} \implies y(x) - p(x) = \frac{y^{(n+1)}(\xi)}{(n+1)!}W(x).$$

Now for the statements you must prove:

(a) Prove that $W(t)$ in (2) is a polynomial of degree $n + 1$.

(b) Prove (3)

(c) Explain how (4) follows from $F(t)$ having at least $n + 2$ zeros in $\text{span}\{x_0, \ldots, x_k, x\}$.

(d) Prove (5).

4. In lecture we proved that the roots of the Chebyshev polynomial

$$T_k(x) = \cos(k \cos^{-1}(x)), k = 0, 1, \ldots \tag{6}$$

are the optimal interpolation points on [-1,1].

(a) Prove that (6) is a polynomial of degree k for all $k \geq 0$.

Proof. For the case of $k = 0, 1$, we have that

$$\begin{aligned} \cos(k \arccos(x))|_{k=0} &= \cos(0) = 1 \\ \cos(k \arccos(x))|_{k=1} &= \cos(\arccos x) = x \end{aligned}$$

So these cases hold. Using induction, assume that $T_n(x)$ is a polynomial for $n < k$, then there are two inductive cases:
First $k = 2t$ is even

$$\begin{aligned} \cos(k \arccos(x)) &= \cos(2(t \arccos(x))) \\ &= 2(\cos(t \arccos(x)))^2 - 1 \text{ [Double angle identity]} \\ &= 2(T_t(x))^2 - 1 \end{aligned}$$

This is again a polynomial since T_t is one by IH, so this case holds.
Second $k = 2t + 1$ is odd

$$\begin{aligned} \cos(k \arccos(x)) &= \cos(2(t \arccos(x)) + \arccos(x)) \\ &\overset{\text{angle sum}}{=} \cos(2t \arccos(x))(\cos(\arccos(x))) - \sin(2t \arccos(x))(\sin(\arccos(x))) \\ &= xT_{2t}(x) - \sin(2t \arccos(x))(\sin(\arccos(x))) \\ &\overset{\text{double angle}}{=} xT_{2k}(x) - 2\cos(t \arccos(x)) \sin(t \arccos(x))(\sin(\arccos(x))) \\ &= xT_{2t}(x) - 2T_t(x) \sin(t \arccos(x))(\sin(\arccos(x))) \\ &\overset{\text{angle prod}}{=} xT_{2k}(x) - T_t(x)(\cos((t-1) \arccos(x)) - \cos((t+1) \arccos(x))) \\ &= xT_{2t}(x) - T_t(x)(T_{t-1} - T_{t+1}) \end{aligned}$$

Which is again a polynomial all the lower degrees of T are polynomials by IH, so both cases hold. $\square$

(b) Derive the leading coefficient of (6) (i.e., the coefficient of x^k).
The coefficient is 2^{k-1} for $k \geq 1$ and 1 for $k = 0$.

Proof. Again, using induction, the case of 1 and 0 are trivial, as $T_0(x) = 1$ and $T_1(x) = x$. For the inductive case, consider again even and odd, and assume that it holds for $n < k$.
For $k = 2t$ even, the leading coefficient $\mathcal{LC}(T_k)$:

$$\begin{aligned} \mathcal{LC}(T_k) &= 2\mathcal{LC}(T_t)^2 \text{ From the recurrence in (a)} \\ &= 2(2^{t-1})^2 = 2(2^{2t-2}) = 2^{k-1} \text{ By IH} \end{aligned}$$

For $k = 2t + 1$ odd,

$$\begin{aligned} \mathcal{LC}(T_k) &= \mathcal{LC}(T_{k-1}) + \mathcal{LC}(T_t)\mathcal{LC}(T_{t+1}) \text{ From the other recurrence in (a)} \\ &= 2^{k-2} + (2^{t-1})(2^t) \text{ From IH} \\ &= 2^{k-2} + 2^{2t-1} \\ &= 2^{k-2} + 2^{k-2} = 2^{k-1} \end{aligned}$$

□

(c) Derive the roots of (6) (i.e., the so-called Chebyshev points).
The roots for a polynomial of degree k are $x_i = \cos(\frac{(2i+1)\pi}{2k})$ where $i = 0, \dots k-1$.

Proof. Given the definition for T_k, plugging in x_i gives

$$\begin{aligned}
\cos(k \arccos(x_i)) &= \cos(k \arccos(\cos\left(\frac{(2i+1)\pi}{2k}\right))) \\
&= \cos\left(k \frac{(2i+1)\pi}{2k}\right) \\
&= \cos\left(\frac{(2i+1)\pi}{2}\right) \\
&= 0 \text{ When } i \in \mathbb{N}
\end{aligned}$$

□

(d) Complete the proof showing why the Chebyshev points are optimal.

5. Consider the definite integral $I(f) = \int_{-1}^{1} f(x)\, dx$.

   (a) Construct the interpolatory quadrature rule for this integral based on nodes $-1, -\frac{1}{2}, \frac{1}{2}, 1$.

   (b) What is the precision of the quadrature rule derived in (a)? **Justify your answer.**

   (c) Find as good an error bound as you can for your quadrature rule, assuming f is as smooth as required for your analysis.