Honors Numerical Analysis

Lecture 16

There are a few questions that can be asked about p_n at this point:

$\boxed{Q1}$ If the points (x_j, y_j) come from a <u>smooth</u> function, f what is the error between p_n and the function f:

$\boxed{Q2}$ What is the cost of evaluating p_n? If a new data point is added, (x_{n+1}, y_{n+1}), what is the cost of updating p_n?

$\boxed{Q3}$ In floating point arithmetic, is the evaluation of p_n stable?

$\boxed{Q1}$ Note, if $y_j = f(x_j)$, then $p_n(x_j) = y_j = f(x_j)$ by <u>construction</u>. If $x \neq x_j$, then

<u>Theorem</u>: Let $f \in C^{n+1}[a,b]$. For $x \in [a,b]$, there exists a $\xi = \xi(x) \in (a,b)$ such that

$$f(x) - p_n(x) = \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}}_{\text{Similar to Taylor's thm}} \underbrace{\prod_{j=0}^{n} (x - x_j)}_{\text{Depends highly on the choice of interpolation points.}}$$

Exact pointwise error.

Moreover:

$$\left| f(x) - p_n(x) \right| \le \frac{M_{n+1}}{(n+1)!} \left| \Pi_{n+1}(x) \right|$$

where $M_{n+1} = \max_{t \in [a,b]} \left| f^{(n+1)}(t) \right|$

$$\Pi_{n+1}(x) = \prod_{j=0}^{n} (x - x_j)$$

Proof is detailed, will not go through it, see text.

Two takeaway points:

1. Only useful if M_{n+1} can be computed.
2. The interpolation error highly depends on where the nodes x_j are located.

This will be very important later on.

$\boxed{Q2}$ The cost of evaluating p_n depends on the form it is written in.

Lagrange Form: $p_n(x) = \underbrace{\sum_{k=0}^{n} y_k L_k(x)}_{\substack{n+1 \text{ mult} \\ n \text{ adds}}}$, $L_k(x) = \underbrace{\prod_{\substack{j=0 \\ j \neq k}}^{n} \underbrace{\frac{x - x_j}{x_k - x_j}}_{3 \text{ flops}}}_{3(n-1) \text{ flops.}}$

$\Rightarrow (n+1)\,3(n-1)$ flops to evaluate all L_k's.

$\Rightarrow$ overall, $O(n^2)$ flops to evaluate p_n in Lagrange form.

Compare this with <u>Horner's Method</u>:

If the coefficients $a_0, \dots, a_n$ are known in

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$ then we can

rewrite p_n as:

$$p_n(x) = a_0 + x\left(a_1 + a_2 x + \dots + a_n x^{n-1}\right)$$

$$= a_0 + x\left(a_1 + x\left(a_2 + a_3 x + \dots + a_n x^{n-2}\right)\right)$$

$$= a_0 + x\left(a_1 + x\left(a_2 + x\underbrace{\underbrace{\left(\cdots\right)}_{b_{n-1} = a_{n-1} + a_n x}}_{b_{n-2} = a_{n-2} + b_{n-1} x}\right)\right)$$

$b_{n-1} = a_{n-1} + a_n x$ (1 mult, 1 add)

$b_{n-2} = a_{n-2} + b_{n-1} x$ (1 mult, 1 add)

$\vdots$

$b_0 = a_0 + b_1 x$ (1 mult, 1 add)

$= p_n(x) \Rightarrow \boxed{2n \text{ flops}}$.

This means that the Lagrange Form is very inefficient.

Is there a better form?

$\boxed{Q3}$ The numerical stability of evaluating p_n in Lagrange form:

Short story: The basic Lagrange form $p_n(x) = \sum_{k=0}^{n} y_k L_k(x)$ <u>can be</u> unstable (i.e. have large condition number).

(<u>Ex</u>: overflow/underflow, roundoff error, etc.)

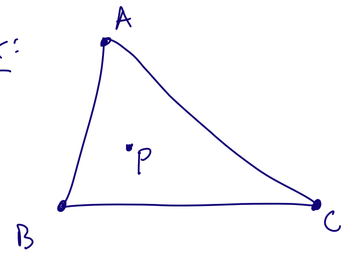
Alternative form next class.

Barycentric Form(s) of Interpolation

The numerical stability of evaluating an interpolating polynomial can be fixed by rearranging its terms – this does not change what the actual interpolant is.

As motivation: Examine the barycentric coordinates on a triangle.

Ex:



The barycentric coordinates of a point P inside a triangle with vertices A, B, C are given by:

$$P = \alpha A + \beta B + \gamma C \qquad (\alpha, \beta, \gamma) \text{ coordinates}$$

$$\text{with} \quad \alpha + \beta + \gamma = 1, \quad \alpha \geq 0, \ \beta \geq 0, \ \gamma \geq 0.$$

The center of mass of the triangle is the given by

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}.$$

Idea: Replace A, B, C with functions that sum to 1.

Start with the Lagrange Form: (and rewrite)

$$p_n(x) = \sum_{k=0}^{n} \underbrace{\left(\prod_{j \neq k} \frac{(x-x_j)}{(x_k - x_j)} \right)}_{L_k(x).} y_k$$

$$= \sum_{k=0}^{n} \underbrace{\left(\prod_{j=0}^{n} (x - x_j) \right)}_{\text{does not depend on } k} \frac{1}{x - x_k} \left(\prod_{j \neq k} \frac{1}{(x_k - x_j)} \right) y_k$$

$$= \underbrace{\left(\prod_{j=0}^{n} (x - x_j) \right)}_{\varphi(x)} \sum_{k=0}^{n} \frac{1}{x - x_k} \underbrace{\left(\prod_{j \neq k} \frac{1}{x_k - x_j} \right)}_{w_k} y_k$$

$$= \varphi(x) \sum_{k=0}^{n} \frac{w_k}{x - x_k} y_k$$

(Modified Lagrange Form)

First Barycentric Formula

We can even further simplify this form by "dividing by 1".

The polynomial interpolant of the function $\underline{1}$ at the same nodes x_j is simply:

$$\underline{1} = \varphi(x) \sum_{k=0}^{n} \frac{w_k}{x - x_k} \qquad (\text{since } y_k = 1).$$

This interpolant is mathematically equivalent to 1.

Then $$p_n(x) = \frac{\cancel{\varphi(x)} \sum_{k=0}^{n} \frac{w_k}{x - x_k} y_k}{\cancel{\varphi(x)} \sum \frac{w_k}{x - x_k}} = \frac{\sum_{k=0}^{n} w_k / x - x_k \cdot y_k}{\sum_{k=0}^{n} w_k / x - x_k}$$

Second Barycentric Formula.

This form is "stable for any reasonable choice of x_j" (2004, Higham).

One should always use this form to do polynomial interpolation.

Convergence of Polynomial Interpolation

Let's examine the question of what happens as $n \to \infty$, i.e.

$$\lim_{n\to\infty} \underbrace{\max_x |f(x) - p_n(x)|}_{\text{this is the } \infty\text{-norm.}} = ?$$

The pointwise error is approximately:

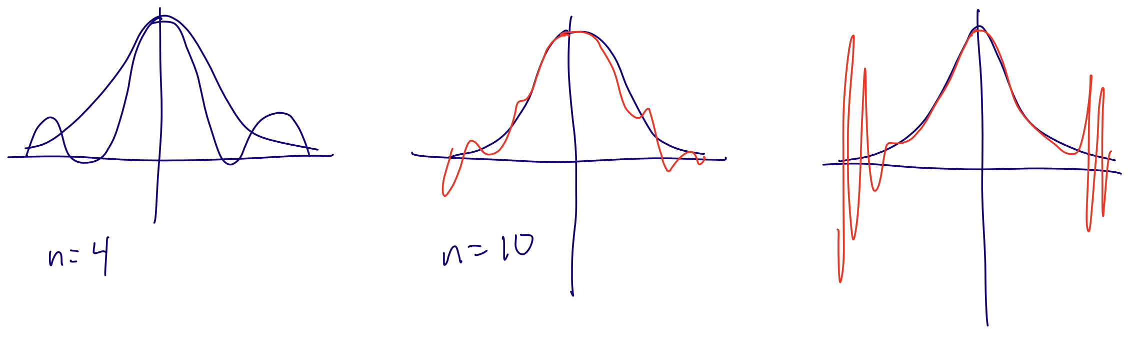
$$\max_\xi \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \cdot \max_x \prod_{j=0}^{n} |x - x_j|$$

It's not obvious if this increases or decreases as $n \to \infty$...

Example

Runge's Function $\quad f(x) = \dfrac{1}{1+(3x)^2}$

The Runge Effect



This behavior is related to the fact that the function $f(x) = \frac{1}{1+x^2}$ has a singularity at $x = \pm i$ in the complex plane. $f(i) = \frac{1}{1+i\cdot i} = \frac{1}{1+-1} = \frac{1}{0} = \infty$.

This dictates the radius of convergence of its Taylor series:

$$f(x) = 1 - x^2 + x^4 - x^6 + x^8 - x^{10} + \ldots$$

(can be fixed, we'll see later on)