Numerical Analysis I
Math 373

Surya Teja Gavva
Rutgers University

July 19 2018

Polynomial Interpolation

What have we done so far?
Lagrange Interpolation:

$$P_n(x) = \sum_{k=0}^{n} f(x_k) L_{k,n}(x)$$

$$L_k(x) = \prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)}$$

$$= \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

Polynomial Approximation

Neville's method: To compute interpolation values $P_n(x)$ iteratively.

$P_{k_1,k_2,\ldots,k_r}$ be the Lagrange interpolation of $f(x)$ at $x_{k_1}, x_{k_2}, \ldots, x_{k_r}$.

We have the formula

$$P_{0,1,2,..,n}(x) = \frac{(x-x_0)P_{1,2,..,n} - (x-x_n)P_{0,1,2,..,n-1}(x)}{x_n - x_0}$$

$$\begin{array}{cccccc}
x_0 & P_0 & & & \\
 & & P_{0,1} & & \\
x_1 & P_1 & & P_{0,1,2} & \\
 & & P_{1,2} & & P_{0,1,2,3} \\
x_2 & P_2 & & P_{1,2,3} & \\
 & & P_{2,3} & & \\
x_3 & P_3 & & &
\end{array}$$

Newton Divided Differences

$$f[x_0, x_1, .., x_n] = \frac{f[x_1, x_2, x_3, .., x_n] - f[x_0, x_1, x_2, .., x_{n-1}]}{x_n - x_0}$$

are defined recursively starting with $f[x_i] = f(x_i)$

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots$$

$$+ f[x_0, x_1, .., x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1})$$

Hermite Interpolation

Problem: Find a polynomial of degree $2n + 1$ that satisfies
$p(x_i) = f(x_i)$ and $p'(x_i) = f'(x_i)$
We want a polynomial that passes through our points and also having the same slope as the original function at these points.

Hermite Interpolation: Using Newton Divided differences

Apply divided differences method to the points $x_0, x_0, x_1, x_1, x_2, x_2, ..., x_n, x_n$. So, we repeat each point twice

$$
\begin{array}{cccccc}
x_0 & f[x_0] & & & & \\
 & & f[x_0,x_0] & & & \\
x_0 & f[x_0] & & f[x_0,x_0,x_1] & & \\
 & & f[x_1,x_1] & & f[x_0,x_0,x_1,x_1] & \\
x_1 & f[x_1] & & f[x_0,x_1,x_1] & & f[x_0,x_0,x_1,x_1,x_2] \\
 & & f[x_1,x_1] & & f[x_0,x_1,x_1,x_2] & \qquad f[x_0,x_0,x_1,x_1,x_2,x_2] \\
x_1 & f[x_1] & & f[x_1,x_1,x_2] & & f[x_0,x_1,x_1,x_2,x_2] \\
 & & f[x_1,x_2] & & f[x_1,x_1,x_2,x_2] & \\
x_2 & f[x_2] & & f[x_1,x_2,x_2] & & \\
 & & f[x_2,x_2] & & & \\
x_2 & f[x_2] & & & &
\end{array}
$$

Hermite Interpolation

But how do we calculate $f[x_0, x_0]$?
if we compute by known formula, we get $\frac{f[x_0]-f[x_0]}{x_0-x_0} = \frac{0}{0}$
We interpret this in a limit sense– we think of the second x_0 as $x_0 + h$ where $h \to 0$
This gives $\lim_{h\to 0} \frac{f[x_0+h]-f[x_0]}{x_0+h-x_0} = f'(x_0)$
So $f[x_i, x_i] = f'(x_i)$
With this definitions, we know how to calculate every term in the above divided difference table.

Hermite Interpolation

We can generalize to interpolation of more dervatives.
We take $f[x_0, x_0, x_0] = \frac{f''(x_0)}{2!}, f[x_0, x_0, x_0, x_0] = \frac{f'''(x_0)}{3!}$, and for $n+1$ of $x_0's$

$$f[x_0, x_0, x_0, \cdots, x_0] = \frac{f^n(x_0)}{n!}$$

Example: If we want $p(x)$ with
$p(x_0) = f(x_0), p'(x_0) = f'(x_0), p''(x_0) = f''(x_0)$,
$p(x_1) = f(x_1), p'(x_1) = f'(x_1)$, $p(x_2) = f(x_2), p'(x_2) = f'(x_2)$, we repeat x_0 thrice, x_1, x_2 twice. The interpolating points are
$x_0, x_0, x_0, x_1, x_1, x_2, x_2$

Hermite Interpolation

Example: If we want $p(x)$ with
$p(x_0) = f(x_0), p'(x_0) = f'(x_0), p''(x_0) = f''(x_0)$,
$p(x_1) = f(x_1), p'(x_1) = f'(x_1)$, $p(x_2) = f(x_2), p'(x_2) = f'(x_2)$, we repeat x_0 thrice, x_1, x_2 twice. The interpolating points are $x_0, x_0, x_0, x_1, x_1, x_2, x_2$

x_0	$f[x_0]$					
		$f[x_0, x_0]$				
x_0	$f[x_0]$		$f[x_0, x_0, x_0]$			
		$f[x_0, x_0]$		$f[x_0, x_0, x_0, x_1]$		
x_0	$f[x_0]$		$f[x_0, x_0, x_1]$		$f[x_0, x_0, x_0, x_1, x_1]$	
		$f[x_0, x_1]$		$f[x_0, x_0, x_1, x_1]$		$f[x_0, x_0, x_0, x_1, x_1, x_2]$
x_1	$f[x_1]$		$f[x_0, x_1, x_1]$		$f[x_0, x_0, x_1, x_1, x_2]$	
		$f[x_1, x_1]$		$f[x_0, x_1, x_1, x_2]$		$f[x_0, x_0, x_1, x_1, x_2, x_2]$
x_1	$f[x_1]$		$f[x_1, x_1, x_2]$		$f[x_0, x_1, x_1, x_2, x_2]$	
		$f[x_1, x_2]$		$f[x_1, x_1, x_2, x_2]$		
x_2	$f[x_2]$		$f[x_1, x_2, x_2]$			
		$f[x_2, x_2]$				
x_2	$f[x_2]$					

Hermite Interpolation: Using Osculating Polynomials

Suppose we want to find $p(x)$ with $p(x_i) = f(x_i)$ and $p'(x_i) = f'(x_i)$
"Lagrange type form"

$$H_{2n+1}(x) = \sum_{k=0}^{n} f(x_k) H_{k,n}(x) + \sum_{k=0}^{n} f'(x_k) \hat{H}_{k,n}(x)$$

where

$$H_{k,n}(x) = [1 - 2(x - x_k) L'_{k,n}(x)] L^2_{k,n}(x)$$

$$\hat{H}_{k,n}(x) = (x - x_k) L^2_{n,k}(x)$$

Polynomial Interpolation

Hermite interpolations tend to behave badly as $n \to \infty$ at the endpoints like we discussed in Runge's example

So, how do we rectify this problem?

Just use the polynomial to approximate in a part of the interval!

We split the interval into various parts and approximate the function on each interval by interpolation