Numerical Analysis

Math 370 Spring 2009 MWF 11:30am - 12:25pm Fowler 110
 http://faculty.oxy.edu/ron/math/370/09/

Worksheet 16

SUMMARY Newton Polynomials, Divided Differences and Chebyshev Nodes
READING Recktenwald, pp. 521–538; Mathews & Fink 220–227, 230–239

Newton Polynomials and Divided Differences
Previously we have shown that it is a relatively trivial task to write down the Lagrange polynomials for a set of nodes $(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n)$

The only problem is that these polynomials are rather annoying to evaluate because they have expressions like: $(X - x_0)(X - x_1)\ldots(X - x_{i-1})(X - x_{i+1})\ldots(X - x_n)$ which can easily overflow or underflow if n gets large.

One solution of this is to use **Newton interpolating polynomials** as basis functions instead of the Lagrange interpolating polynomials. (Yes, this is the same Sir Isaac Newton who co-invented calculus, Newton's Method, Laws of Gravitation, *et cetera.*)

The general form of the interpolant $\mathcal{F}(x)$ when using Newton interpolating polynomials of degree n is

$$\mathcal{F}(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

where a_k for $k = 0$ to n are "appropriate constants".
NOTE: The neat thing about this form of the interpolating polynomial is that computing the appropriate constants a_k is actually quite easy. (Though not as easy to find as they were with Lagrange polynomials.)

1. **Question**: What are the coefficents a_k when interpolating using Lagrange polynomials as a basis? **Answer:** ______________
In addition, Newton polynomials have the property that they can be written in **nested form** very easily. For example,

$$P_4(x) = [[[[a_4] * (x - x_3) + a_3] * (x - x_2) + a_2] * (x - x_1) + a_1] * (x - x_0) + a_0$$

Here P_4 just means that this is the unique interpolating polynomial of degree 4 for the points (x_0, y_0), (x_1, y_1), (x_2, y_2) and $x_3, y_3)$.
This form is useful because it means that $P_N(x)$ can be expressed in terms of $P_{N-1}(x)$ very easily, i.e. $P_N(x) = P_{N-1}(x) + a_N(x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{N-1})$

Exercise

2. Show that the nested form of the Newton Polynomial is exactly the same as the "natural" (un-nested) form of the Newton Polynomial (which is shown below). (*Which form uses more arithmetic operations when evaluated?*)

EXAMPLE

Considering nodes at $x_0 = 1$, $x_1 = 2$ and $x_2 = 4$ let's write down the Newton polynomials $P_1(x)$, $P_2(x)$ and $P_3(x)$. What's $P_0(x)$?

RECALL

$P_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \ldots + a_n(x-x_0)(x-x_1)(x-x_2)(x-x_3)\ldots(x-x_{n-1})$

Exercise

Our goal is to obtain a general formula for a_k in the Newton Polynomial formula given a set of $n+1$ nodes $(x_k, y_k)_{k=0}^{n}$

3. For the general Newton Polynomial $P_n(x)$ what is the value of a_0?

4. Using your answer for a_0 find a formula for a_1

5. Can you surmise a formula for a_2?

6. Do you see a pattern for a formula for a_k?

Divided Differences

The **zeroth divided difference** of the function $f(x)$ with respect to x_i is denoted by $f[x_i]$ and is given by $f(x_i)$. In other words, $f[x_i]$ is just the function $f(x)$ evaluated at x_i,

$$f[x_i] = f(x_i)$$

The **first divided difference** of f with respect to x_i and x_{i-1} is denoted by $f[x_{i-1}, x_i]$ and defined recursively in terms of zeroth divided differences:

$$f[x_{i-1}, x_i] = \frac{f[x_1] - f[x_{i-1}]}{x_i - x_{i-1}}$$

The **second divided difference** of f is denoted $f[x_{i-2}, x_{i-1}, x_i]$ and defined as

$$f[x_{i-2}, x_{i-1}, x_i] = \frac{f[x_{i-1}, x_i] - f[x_{i-2}, x_{i-1}]}{x_i - x_{i-2}}$$

The **third divided difference** of f is denoted $f[x_{i-2}, x_{i-1}, x_i]$ and defined as

$$f[x_{i-3}, x_{i-2}, x_{i-1}, x_i] = \frac{f[x_{i-2}, x_{i-1}, x_i] - f[x_{i-3}, x_{i-2}, x_{i-1}]}{x_i - x_{i-3}}$$

As you may have guessed, in general the **kth divided difference** of f with respect to $x_i, x_{i-1}, \ldots, x_{i-k}$ is given by

$$f[x_{i-k}, x_{i-k+1}, \ldots, x_i] = \frac{f[x_{i-k+1}, x_{i+2}, \ldots, x_{i-1}, x_i] - f[x_{i-k}, \ldots, x_{i-1}]}{x_i - x_{i-k}}$$

Newton Polynomial formula

The value of a_k, the coefficient of x^k of the Newton Polynomial is exactly the **kth divided difference**.

GroupWork

7. Use a divided difference table to compute the coefficients of the Newton interpolating polynomial for the data $(-1, 1/2), (0, 1), (1, 2)$ and then extrapolate the value at $x = 1.5$. (you may want to explore the Matlab function `newtint`)

Chebyshev Polynomials

The Chebyshev Polynomials are given by the recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ for $k = 2, 3, \ldots$ where $T_0(x) = 1$ and $T_1(x) = x$.

Exercise

8. Write down $T_4(x)$, the 4th degree Chebyshev polynomial.

Lagrange-Chebyshev Polynomial Interpolation

It turns out that the zeroes of the Chebyshev polynomial are exactly the optimal place to put nodes to minimize Lagrange Polynomial interpolation error. So, if one has the interval $[a, b]$ one should place $N + 1$ uniformly distributed nodes in the following locations

$$x_k = \frac{b-a}{2} t_k + \frac{a+b}{2} \text{ where } t_k = \cos\left((2N+1-2k)\frac{\pi}{2N+2}\right) \text{ for } k = 0, 1, 2, \ldots, N$$

In this case, the Lagrange-Chebyshev Polynomial Interpolant $P_N(x)$ is able to approximate a function $f(x)$ that is $N+1$-times differentiable on the interval (a, b) within an error defined as

$$|f(x) - P_N(x)| \leq \frac{2(b-a)^{N+1}}{4^{N+1}(N+1)!} \max_{a \leq \xi \leq b} |f^{N+1}(\xi)|$$

EXAMPLE

Mathews & Fink 4.15, page 237. 9. For $f(x) = \sin(x)$ on $[0, \pi/4]$, find the Chebyshev nodes and the error bound for the Lagrange-Chebyshev polynomial $P_5(x)$.