AMSC/CMSC 460 Computational Methods, Fall 2007
UNIT 2: Polynomial Approximations

Dianne P. O'Leary

Polynomial Interpolation

Read: Chapter 2. Skip: 2.4.2, 2.4.4.

Caution: The numbering of coefficients is inconsistent in the chapter. See the difference between the top of p. 76 and the bottom. I'll use the Matlab notation (top of p. 76) rather than Van Loan's.

The Plan:

- Why polynomial interpolation is useful.
- Some facts about polynomials.
- Computing interpolating polynomials.
- Evaluating polynomials.
- Existence and uniqueness: Lagrange and Newton form of the polynomial.
- The error formula.
- 2-d interpolation.

Why polynomial interpolation is useful

Motivation

Problem 1: Given some experimental data (e.g., 1000 measurements of mortality rate as a function of fat consumption, or 20 yearly measurements of amount of pollution in a river), see how well the data fits a model such as a polynomial or a sum of exponentials.

Problem 2: Fit a model to data in order to reduce the effects of noise in the measurements.

We'll study this in Chapter 7.

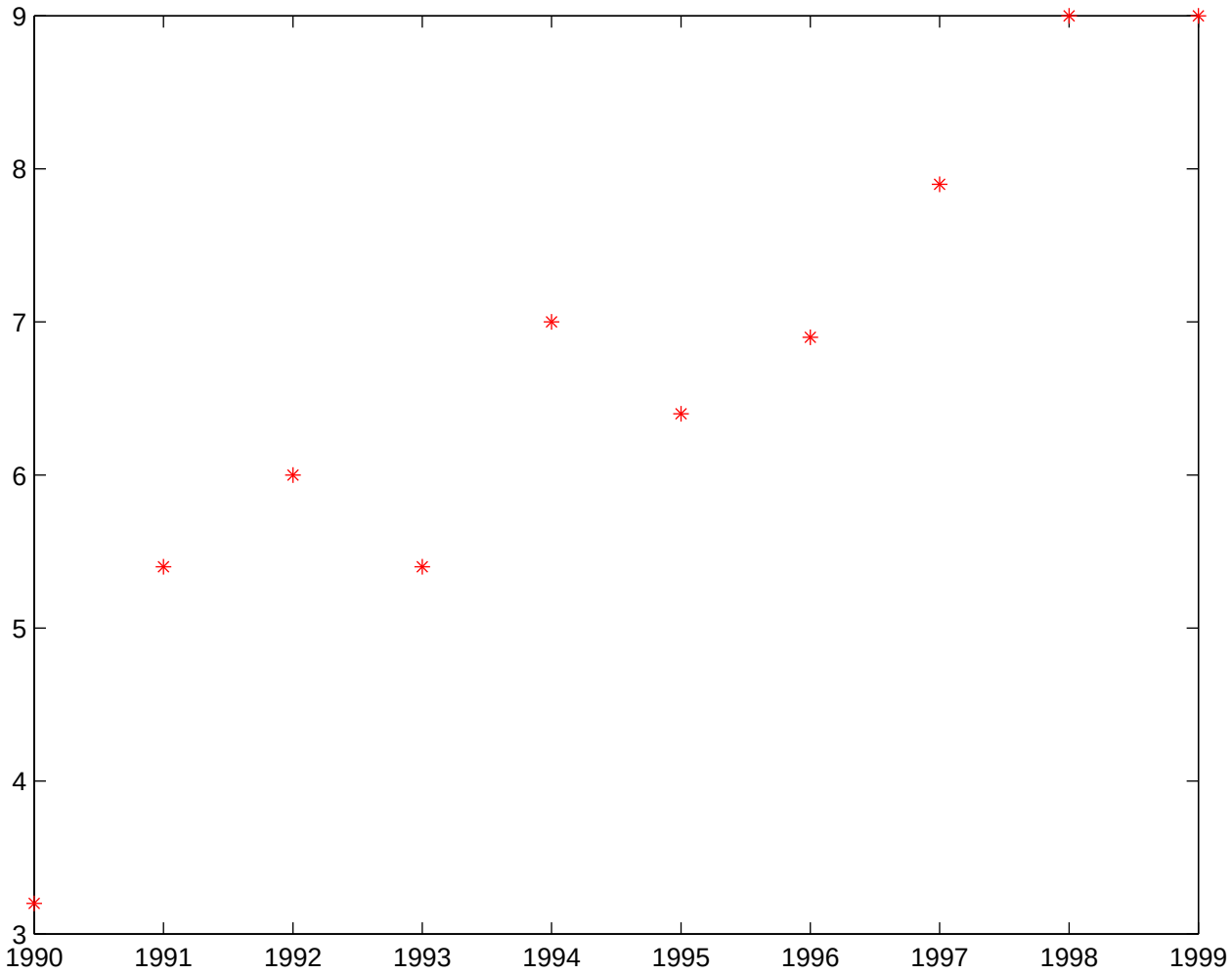

Figure 1: Pollution levels (ppm) vs. year. Does a polynomial fit this data?

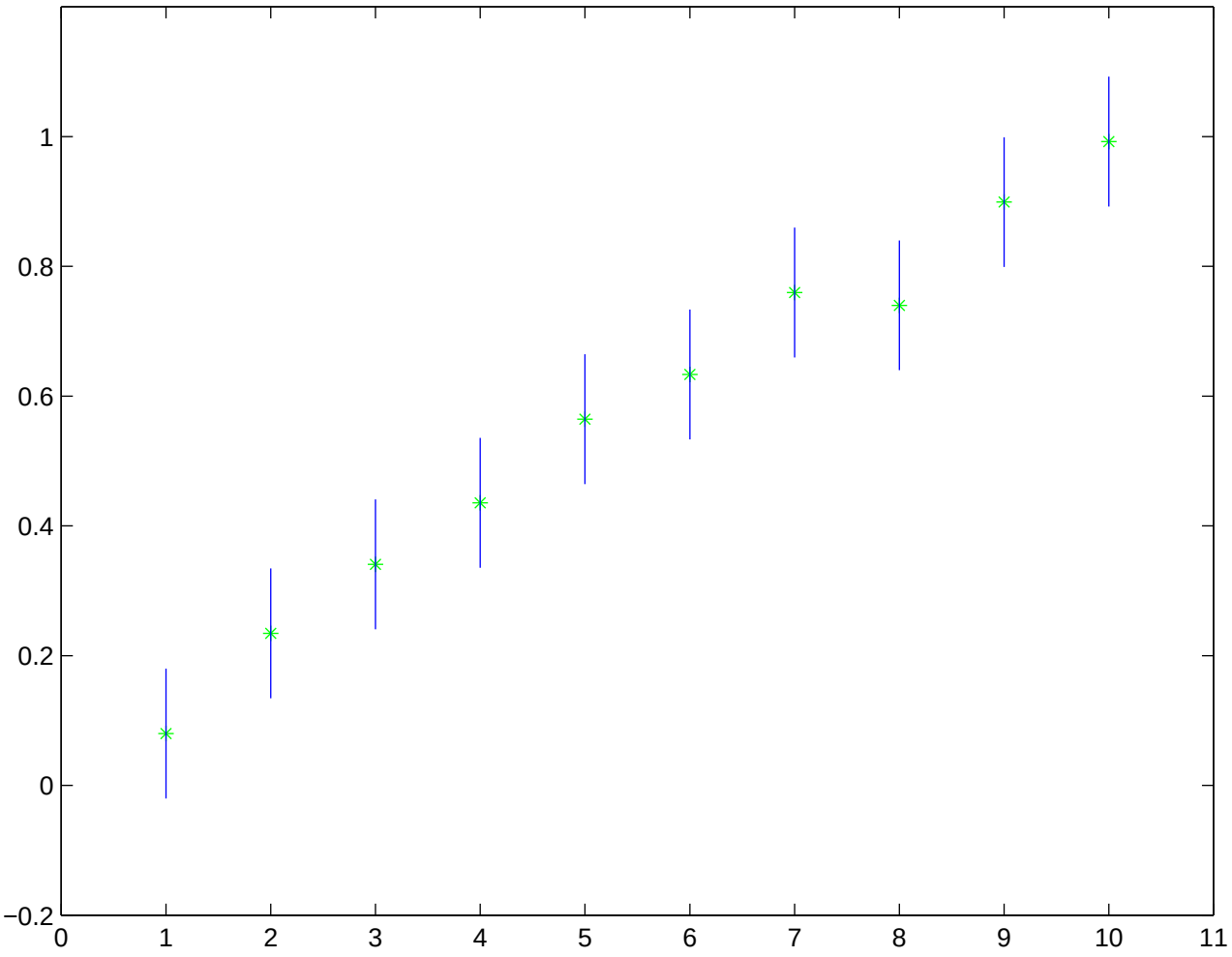

Figure 2: Is a straight line a good model to this data?

Problem 3: Fit a simple model, such as a polynomial, to a piece of a function that is expensive to calculate, so that we can use the simple model as a substitute.

Problem 4: Estimate the integral or the derivative of some function.

Polynomial Interpolation

Why polynomials?

- often a reasonable model.
- cheap to calculate and evaluate.
- important in solving ordinary differential equations (ode's) and nonlinear equations and in computing integrals.

How well can polynomials approximate functions?

To find out, we need to drop the requirement of interpolation.

Theorem: (Weierstrass) For all $f \in \mathcal{C}[a,b]$, for all $\epsilon > 0$, there exists a degree n and a polynomial p_n such that $||f - p||_\infty < \epsilon$.

Problems with Weierstrass way of constructing the polynomial:

- The degree can be quite high.
- The polynomial is not guaranteed to interpolate at any given set of points.

The standard polynomial interpolation problem:
Given: (x_i, f_i), $i = 1, \ldots, n$, $x_i \neq x_j$ for $i \neq j$
Find: a polynomial $p_{n-1}(x)$ of degree at most $n-1$ such that $p_{n-1}(x_i) = f_i$, $i = 1, \ldots, n$.

Some facts about polynomials

Some facts about polynomials

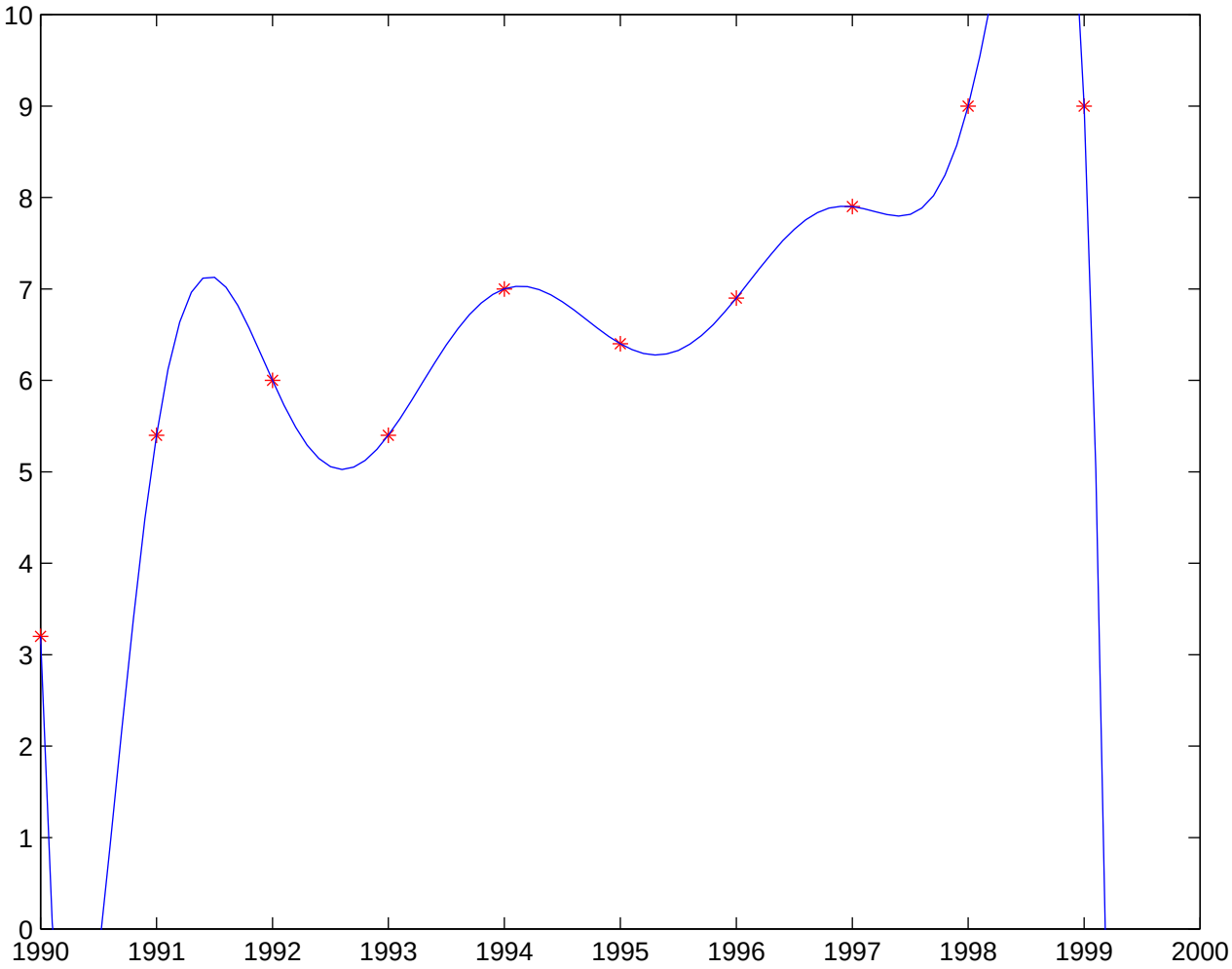

Figure 3: Polynomial fit to pollution data

- A polynomial of degree k has at most k distinct zeroes, unless it is the zero polynomial $z(x) = 0$ for all x.
- The sum of two polynomials of degree at most k is a polynomial of degree at most k.
- There are many ways to write a given polynomial.

  For example $(x-2)(x-5) = x^2 - 2x - 5(x-2) = x^2 - 7x + 10$.

  We have used three different sets of basis functions in this example:

  1. $(x-2)(x-5)$, $x-1$, and 1.
  2. x^2, x, and $x-2$.
  3. x^2, x, and 1.

  The third basis set is the most familiar one, called the power basis.

  How we write a polynomial – what set of basis functions we use – can be chosen for our convenience!

  We can use any basis functions, as long as they are independent.

Computing interpolating polynomials

Computing interpolating polynomials

Suppose we want to compute the polynomial of degree 3 that interpolates the data $(1,3), (2,5), (-1,4), (0,6)$. Let's use the power basis: $x^3, x^2, x, 1$, and let the polynomial be denoted as

$$p(x) = a_1 x^3 + a_2 x^2 + a_3 x + a_4.$$

Then the four conditions that our polynomial needs to satisfy are

$$\begin{aligned}
p(1) = 3: &\quad 1^3 a_1 + 1^2 a_2 + 1 a_3 + a_4 &&= 3,\\
p(2) = 5: &\quad 2^3 a_1 + 2^2 a_2 + 2 a_3 + a_4 &&= 5,\\
p(-1) = 4: &\quad (-1)^3 a_1 + (-1)^2 a_2 + (-1) a_3 + a_4 &&= 4,\\
p(0) = 6: &\quad 0^3 a_1 + 0^2 a_2 + 0 a_3 + a_4 &&= 6,
\end{aligned}$$

If we write this in matrix form, we have

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 4 \\ 6 \end{bmatrix}.$$

For conciseness, we can say

$$Va = y$$

where V is the matrix, a is the vector of coefficients of the polynomial, and y is the set of y coordinates for the data. The x coordinates determine the matrix V:

$$V = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix}.$$

We need to solve this linear system for the 4 values in the vector a.

This is such a commonly occurring problem that Matlab has a special function to do it:

```
x = [1 2 -1 0];
y=[3 5 4 6];
a = polyfit(x,y,3)
```

The third argument to `polyfit` gives the degree of the polynomial.

The matrix for the interpolation problem

The matrix in our polynomial interpolation problem has a very special form:

$$V = \begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \dots & x_1^1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \dots & x_2^1 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ x_n^{n-1} & x_n^{n-2} & \dots & x_n^1 & 1 \end{bmatrix}$$

and a special name, a Vandermonde matrix.

(To be precise, a Vandermonde matrix looks like this but with the order of the columns reversed.)

Evaluating polynomials

Evaluating polynomials

Now suppose someone asks what is the value of the polynomial at $x = 7$?

We can determine this by computing

$$a_1 * 7 * 7 * 7 + a_2 * 7 * 7 + a_3 * 7 + a_4.$$

Unquiz:

- Write an algorithm to implement this for a polynomial of degree $n - 1$.
- Count the number of multiplications that the program does.
- Think about how we might make the algorithm faster.

Evaluating polynomials using Horner's rule

Given x, we can evaluate $p(x)$ by the following recursion:

$p = a_1$

For $j = 2, \ldots, n$,

$$p = p * x + a_j.$$

end for

Unquiz: Count the number of multiplications and compare with the previous algorithm.

Existence and uniqueness: Lagrange and Newton form of the polynomial

---

Existence of interpolation polynomial

Can we guarantee that we can always find a unique polynomial of degree $n - 1$ that interpolates any given set of n data points?

In order to guarantee this, we would have to know that the linear system of equations $Va = y$ always has one and only one solution. In other words, we would need to know that V is a nonsingular matrix.

Although it is true that V is always nonsingular, as long as the values $x_1, \ldots x_n$ are distinct, this is not so easy to prove. So we use the fact that we can write the

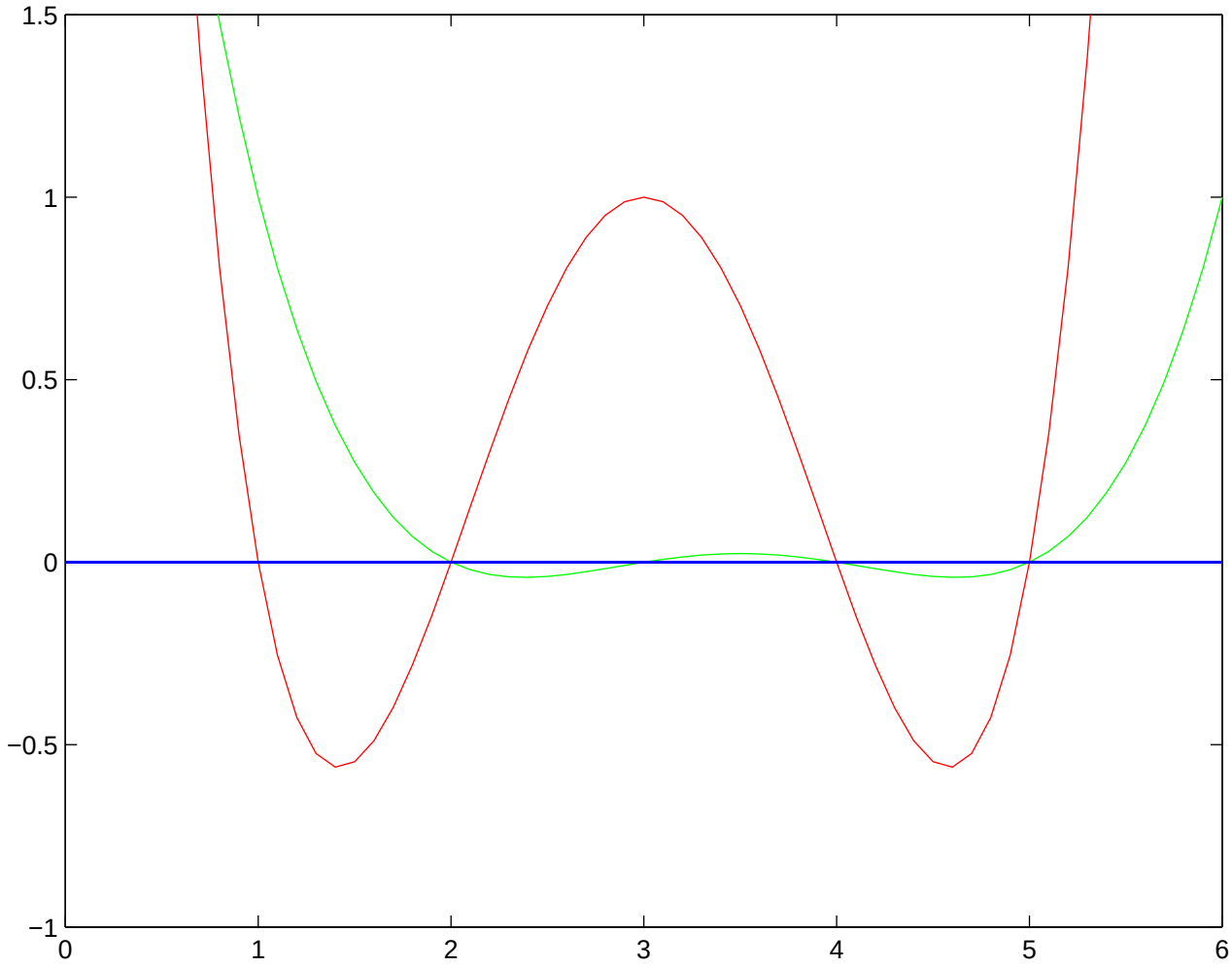

Figure 4: $L_{5,1}$ and $L_{5,3}$ for interpolation points 1, 2, 3, 4, 5.

polynomial in any basis, and choose one more convenient for the proof, in order to show existence. Then we will show uniqueness using properties of polynomials.

The most convenient set of basis functions for this proof: Lagrange basis: for $k = 1, 2, \ldots, n$

$$L_{n,k}(x) = \frac{\prod_{i=1, i\neq k}^{n}(x - x_i)}{\prod_{i=1, i\neq k}^{n}(x_k - x_i)}$$

$L_{n,k}$ is an interesting polynomial:

- $L_{n,k}$ is a polynomial of degree $n - 1$.
- $L_{n,k}(x_k) = 1$ and $L_{n,k}(x_i) = 0$, for $i \neq k$.

The Lagrange form of the interpolating polynomial

Let

$$p_{n-1}(x) = \sum_{i=1}^{n} f_i L_{n,i}(x)\,.$$

Then $p_{n-1}(x)$ is a polynomial of degree $n-1$, and $p_{n-1}(x_i) = f_i$.

So existence of the Lagrange interpolating polynomial is assured. []

Unquiz: Write the Lagrange form of the interpolating polynomial for the data $(1,3),(2,5),(-1,4),(0,6)$.

Uniqueness of the Lagrange interpolating polynomial

We know that the interpolating polynomial exists.

Suppose there are two polynomials $p_{n-1} \neq q_{n-1}$ and each of them solves the interpolation problem.

So the polynomials satisfy $p_{n-1}(x_i) = q_{n-1}(x_i) = f_i$, $i = 1,\ldots,n$.

So $p_{n-1} - q_{n-1}$, also a polynomial of degree $n-1$, has n distinct roots. Therefore, $p_{n-1} - q_{n-1}$ must be the zero polynomial, so $p_{n-1} = q_{n-1}$.

[]

Remaining tasks:

We now have existence and uniqueness of the solution to the interpolation problem.

We need

- an alternate representation for the interpolation polynomial.
- a rate of convergence of the polynomial to a function that we are approximating.

A second convenient basis: The Newton basis

Let

$$\begin{aligned}\phi_0(x) &= 1\,,\\ \phi_j(x) &= (x-x_1)(x-x_2)\dots(x-x_j)\,, j=1,\dots,n\end{aligned}$$

Note that ϕ_j is a polynomial of degree j.

We will express the interpolating polynomial as a weighted sum of these basis functions.

Define $f[x_1,\dots,x_j]$ to be the coefficient of ϕ_{j-1} in the interpolating polynomial. We will call the jth coefficient the jth divided difference.

Then

$$p_{n-1}(x) = f[x_1] + f[x_1,x_2](x-x_1) + f[x_1,x_2,x_3](x-x_1)(x-x_2)$$
$$+\dots+ f[x_1,x_2,x_3,\dots,x_n](x-x_1)(x-x_2)\dots(x-x_{n-1})$$

Our task: determine these coefficients and their properties.

What is a divided difference?

Let $I=[\min_j x_j, \max_j x_j]$, and assume that $f\in\mathcal{C}^n[I]$.

Property: If we reorder the points, the divided difference remains the same.

Proof: $f[x_1,x_2,x_3,\dots,x_n]$ is the coefficient of x^{n-1} in the interpolating polynomial. Since the polynomial is unique, the order of the points cannot matter. []

Property: If $x_1\neq x_n$, then

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

Note: This will be our computational formula!

Proof: Let $\hat{p}_{n-3}$ be the polynomial that satisfies all of the interpolation conditions except those for x_1 and x_n. By uniqueness,

$$\begin{aligned} p_{n-1}(x) &= \hat{p}_{n-3}(x) + f[x_2, \ldots, x_{n-1}, x_1](x - x_2) \ldots (x - x_{n-1}) \\ &\quad + f[x_2, \ldots, x_{n-1}, x_1, x_n](x - x_1) \ldots (x - x_{n-1}) \\ &= \hat{p}_{n-3}(x) + f[x_2, \ldots, x_{n-1}, x_n](x - x_2) \ldots (x - x_{n-1}) \\ &\quad + f[x_2, \ldots, x_{n-1}, x_n, x_1](x - x_2) \ldots (x - x_n) \end{aligned}$$

Therefore,

$$f[x_1, \ldots, x_n](x - x_2) \ldots (x - x_{n-1})[(x - x_1) - (x - x_n)] =$$

$$-f[x_1, \ldots, x_{n-1}](x - x_2) \ldots (x - x_{n-1}) + f[x_2, \ldots, x_n](x - x_2) \ldots (x - x_{n-1}),$$

so, after dividing by $(x - x_2) \ldots (x - x_{n-1})$, we obtain

$$f[x_1, \ldots, x_n][(x - x_1) - (x - x_n)] = -f[x_1, \ldots, x_{n-1}] + f[x_2, \ldots, x_n]. \; []$$

Computing the polynomial

See powerpoint presentation.

The error formula

The error formula

Recall Rolle's theorem: If $f(a) = f(b)$, then there exists $\psi \in (a, b)$ such that $f'(\psi) = 0$.

Theorem: If $f \in \mathcal{C}^n[I]$, then

$$f(x) - p_{n-1}(x) = \frac{(x - x_1) \ldots (x - x_n) f^{(n)}(\xi)}{n!}$$

for some point ξ in the interval containing I and x.

Proof: Given x, consider the function

$$E(t) = f(t) - p(t) - cL(t)$$

where c is a constant

$$c = \frac{f(x) - p(x)}{(x - x_1) \ldots (x - x_n)}$$

and

$$L(t) = (t - x_1) \ldots (t - x_n).$$

Note that $E(x) = 0$ and $E(x_i) = 0$, $i = 1, \ldots, n$.

Therefore, by Rolle's Theorem, in between each of these zeros, E' has a zero. So

- E' has at least n zeroes,
- E'' has at least $n-1$ zeroes,
- ...
- $E^{(n)}$ has at least 1 zero, call it ξ.

Let's consider $E^{(n)}$.

- Observe that $p^{(n)} = 0$ since p is a polynomial of degree $n-1$.
- $L^{(n)}(t) = n!$.

Therefore,

$$0 = E^{(n)}(\xi) = f^{(n)}(\xi) - cn!$$

Using our definition of c, we obtain

$$f(x) - p(x) = f^{(n)}(\xi)\frac{(x-x_1)\dots(x-x_n)}{n!}\,.$$

[]

We can get similar results for $f'(x) - p'(x)$.

Note: Let's derive an alternate form of the error, that will be useful to us when we study numerical integration. If we write down the polynomial that interpolates f at $x_1, \dots, x_n$ and also at $\hat{x}$, we get

$$\hat{p}(x) = p_{n-1}(x) + f[x_1, \dots, x_n, \hat{x}](x-x_1)\dots(x-x_n)\,.$$

Then, since $\hat{p}(\hat{x}) = f(\hat{x})$, we have

$$f(\hat{x}) - p_{n-1}(\hat{x}) = f[x_1, \dots, x_n, \hat{x}](\hat{x}-x_1)\dots(\hat{x}-x_n)$$

Therefore,

$$f[x_1, \dots, x_n, \hat{x}] = \frac{f^{(n)}(\xi)}{n!}$$

for some point ξ in the interval I.

Evaluating Newton polynomials using Horner's rule

For conciseness, let $c_k \equiv f[x_1, \dots, x_k]$.

Then, given x, we can evaluate $p_{n-1}(x)$ by the following recursion:

$p = c_n$

For $j = n-1, \dots, 1$,

$p = p * (x - x_j) + c_j$.

end for

Polynomial approximation

There exist functions for which polynomial interpolation does not get closer to the function as the number of interpolation points increases, because the derivatives are not well-behaved.

Van Loan gives an example, the Runge function, p. 91.

2-d interpolation

Interpolation in 2 dimensions

Suppose we want a polynomial of two variables that satisfies

$$\begin{aligned} p(0,0) &= 1 \\ p(0,1) &= 2 \\ p(1,0) &= -1 \\ p(1,1) &= 6 \end{aligned}$$

Then we need a polynomial with 4 parameters: for example:

$$p(x,y) = c_1 + c_2 x + c_3 y + c_4 xy$$

Unquiz: Find the conditions that these four coefficients must satisfy.

Final Words:

- Vandermonde matrices are very ill-conditioned. (We'll illustrate this in Chapter 7.) Therefore, the power basis is not a good one to work with.
- Polynomial interpolation should not be used unless you expect the function to be modeled very well by a low-order polynomial!
- Polynomial interpolation is critical in many applications: numerical integration, solution of differential equations, solution of nonlinear equations, etc. In each of these cases, the interval is very short, so a polynomial is a good model!