Chapter 10: Polynomial Interpolation

Uri M. Ascher and Chen Greif
Department of Computer Science
The University of British Columbia
{ascher,greif}@cs.ubc.ca

Slides for the book
A First Course in Numerical Methods (published by SIAM, 2011)
`http://www.ec-securehost.com/SIAM/CS07.html`

This chapter is mainly from Dr. Peter Arbenz Lecture notes at ETH

Topics of today

- What is interpolation
- Monomial interpolation

References

- U. Ascher & C. Greif: Numerical methods. SIAM 2011. Chapter 10.1–10.3.

Interpolating data

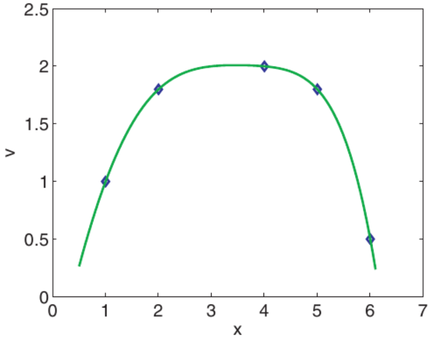
We are given a collection of data samples $\{(x_i, y_i)\}_{i=0}^{n}$

- The $\{x_i\}_{i=0}^{n}$ are called the abscissae, the $\{y_i\}_{i=0}^{n}$ are called the data values.
- Want to find a function $v(x)$ which can be used to estimate sampled function for $x \neq x_i$.

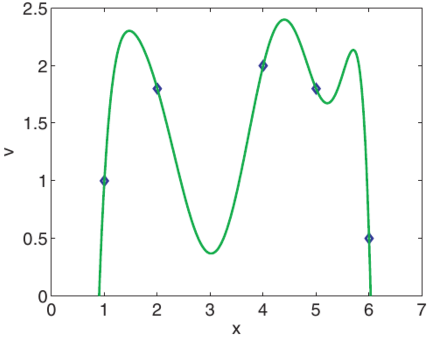
$$\boxed{\text{Interpolation:} \qquad v(x_i) = y_i, \quad i = 0, 1, \ldots, n.}$$

- Why? We often get discrete data from sensors or computation, but we want information as if the function were not discretely sampled.
- Want a *reasonable* looking interpolant. If possible, $v(x)$ should be inexpensive to evaluate for a given x.

Interpolating data (cont.)



(a) Reasonable.



(b) Unreasonable.

Interpolating functions

A function $f(x)$ may be given explicitly or implicitly. Want interpolant $v(x)$ such that

$$v(x_i) = f(x_i), \qquad i = 0, 1, \ldots, n.$$

Same algorithms as for interpolating data, but here

- May be able to choose abscissae x_i.
- May be able to estimate interpolation error.

There are lots of ways to define a function $v(x)$ to interpolate the data: polynomials, trigonometric functions, exponential, rational functions (fractions), wavelets, etc.

Interpolation and approximation

- Interpolation means that the approximating function g coincides with the function at prescribed points:
$$v(x_i) = f(x_i) = y_i,\ i = 0, \ldots, n.$$
- Approximation means that some norm $\|\mathbf{v} - \mathbf{y}\|$ of the difference of the vectors $\mathbf{v} = [v(x_0), \ldots, v(x_n)]$ and $\mathbf{y} = [y_0, \ldots, y_n]$ is minimized, or the difference of the given function $f(x)$ and $v(x)$.
- Quality of approximation/interpolation depends on the method and the choice of v.
- Interpolation: $g(z)$ is computed at some new value $z \neq x_i$ inside the range of the interpolation points $x_0, \ldots, x_n$.
- Extrapolation: the new value z is outside this range.

The need for interpolation/extrapolation

- For *prediction*: we can use $v(x)$ to find approximate values of the underlying function at locations x other than the data abscissae, $x_0, \ldots, x_n$.
  - Tabulated numbers
  - Stock performance
- For *manipulation*: an instance is finding approximations for derivatives and integrals of the underlying function.

- The interpolating function should be easy to evaluate and manipulate, and of course to generate.

Interpolation formulation

Assume a *linear form* of interpolation

$$v(x) = \sum_{j=0}^{n} c_j \phi_j(x) = c_0 \phi_0(x) + \cdots + c_n \phi_n(x)$$

where $\{c_i\}_{i=0}^{n}$ are *unknown* coefficients or parameters and $\{\phi_i(x)\}_{i=0}^{n}$ are predetermined basis functions.

The basis functions are assumed to be *linearly independent*.

There are $n+1$ coefficients to be determined by $n+1$ equations.

We assume : number of basis functions = number of data points n+1

Interpolation formulation (cont.)

The $n+1$ equations $v(x_i) = y_i$, $i = 0, 1, \ldots, n$ yield

$$\begin{pmatrix} \phi_0(x_0) & \phi_1(x_0) & \phi_2(x_0) & \cdots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \vdots & \vdots & & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \phi_2(x_n) & \cdots & \phi_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Choices for the $\phi_j(x)$:

- Polynomial interpolation with monomial basis $\phi_j(x) = x^j$.
- Piecewise polynomial interpolation: soon to come.
- Trigonometric interpolation: $\phi_j(x) = \cos(jx)$.

Example 1: linear data fitting

$$\{(x_i, y_i)\} = \{(2, 14), (6, 24), (4, 25), (7, 15)\}$$

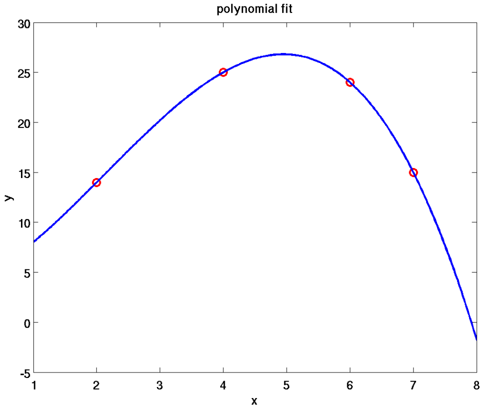
- Requires four basis functions: $\{\phi_j(x)\} = \{1, x, x^2, x^3\}$.
  The interpolant will be $\{p(x) = c_0 1 + c_1 x + c_2 x^2 + c_3 x^3\}$.
- Construct linear system

$$A = \begin{pmatrix} 1 & 2 & 4 & 8 \\ 1 & 6 & 36 & 216 \\ 1 & 4 & 16 & 64 \\ 1 & 7 & 49 & 343 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 14 \\ 24 \\ 25 \\ 15 \end{pmatrix}$$

- Solve `c = A\y`. We find $\mathbf{c} \approx (3.800, 2.767, 1.700, -0.267)^T$.

Example 1: linear data fitting (cont.)

```
x=[2 6 4 7];
y=[14 24 25 15];
c=polyfit(x,y,3)
xx=[1:.02:8];
yy=polyval(c,xx);
plot(x,y,'ro',xx,yy)
```



Example 2: fitting a rational function

Consider the function

$$f(x) = \int_0^x e^{\sin t} dt$$

some values of which are tabulated:

x	0.4	0.5	0.6	0.7	0.8	0.9
y	0.4904	0.6449	0.8136	0.9967	1.1944	1.4063

What is the value $f(0.66)$?
Let's choose

$$g(x) = \frac{a}{x-b}$$

Example 2: fitting a rational function (cont.)

We determine the parameters a and b by requiring that g interpolates at both neighboring points of z:

$$g(0.6) = \frac{a}{0.6 - b} = 0.8136,$$
$$g(0.7) = \frac{a}{0.7 - b} = 0.9967.$$

The result is $a = -0.4429$ and $b = 1.1443$ thus

$$g(0.66) = -\frac{0.4429}{0.66 - 1.1443} = 0.9145 \approx f(0.66) = 0.9216$$

Interpolation error is rather large: $|g(0.66) - f(0.66)| = 0.0071$.

Choice of the model function g was not clever: linear function $g(x) = ax + b$ leads to smaller error 0.0019.

Polynomial interpolation

- A common choice of model functions for interpolation is polynomials, which are easy to evaluate and smooth, i.e., infinitely differentiable ($\in C^\infty$).
- Given $n+1$ points x_i, we are looking for a polynomial $p(x)$ such that

$$p(x_i) = f(x_i) = y_i, \quad i = 0, \dots, n. \qquad (*)$$

- As we have $n+1$ constraints to satisfy, we need $n+1$ degrees of freedom. Consider the n-th degree polynomial

$$p_n(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} + c_n x^n.$$

The $n+1$ coefficients c_i have to be determined in such a way that $(*)$ is satisfied.

Polynomial interpolation (cont.)

This leads to the linear system of equations

$$
\begin{aligned}
c_0 + c_1 x_0 + \cdots + c_{n-1} x_0^{n-1} + c_n x_0^n &= y_0 \\
c_0 + c_1 x_1 + \cdots + c_{n-1} x_1^{n-1} + c_n x_1^n &= y_1 \\
&\vdots \\
c_0 + c_1 x_n + \cdots + c_{n-1} x_n^{n-1} + c_n x_n^n &= y_n.
\end{aligned}
$$

In matrix form, the system reads

$$
\underbrace{\begin{bmatrix} 1 & x_0 & \dots & x_0^{n-1} & x_0^n \\ 1 & x_1 & \dots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & \dots & x_n^{n-1} & x_n^n \end{bmatrix}}_{V}
\underbrace{\begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}}_{\mathbf{c}}
\underset{=}{=}
\underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}}
\tag{1}
$$

Polynomial interpolation (cont.)

- V is called a Vandermonde matrix. Its elements are the powers of the nodes.
- The determinant of V is

$$\det(V) = \prod_{i \neq j} (x_i - x_j)$$

- If all the nodes x_i are distinct, then V is nonsingular and the linear system of equations has a unique solution.

Uniqueness of interpolation

The polynomial $p_n(x)$ in $(*)$ is the only polynomial of degree n that interpolates f at the $n+1$ distinct points $x_0, \ldots, x_n$.

Proof.

If $q_n(x) \in \mathbb{P}_n$ was another such polynomial:

$$p_n(x_k) = q_n(x_k) = y_k, \qquad k = 0, 1, \ldots, n.$$

Then the difference polynomial $d(x) := p_n(x) - q_n(x) \in \mathbb{P}_n$ has $n+1$ zeros $x_0, \ldots, x_n$. By the fundamental theorem of algebra a nonzero polynomial of degree $\leq n$ has at most n zeros. But d has $n+1$ distinct zeros; hence, it must be identically zero, meaning that $q_n(x) \equiv p_n(x)$. $\square$

Note: this statement proves nonsingularity of Vandermonde matrix.

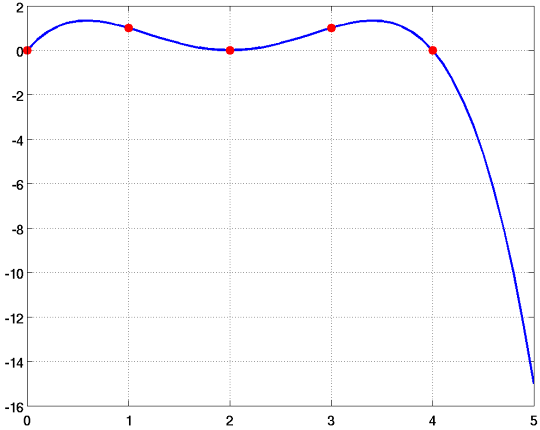
Polynomials in MATLAB

MATLAB displays polynomials as row vectors containing the coefficients ordered by descending powers.

$$p(x) = c_0 + c_1 x + \ldots + c_n x^n \quad \Longleftrightarrow \quad \mathbf{c} = [c_n, \ldots, c_1, c_0]$$

```
>> x=[0:4]'; y=[0 1 0 1 0]';
>> V=vander(x)
V =
     0      0      0      0      1
     1      1      1      1      1
    16      8      4      2      1
    81     27      9      3      1
   256     64     16      4      1
>> a=V\y;
>> X=[0:.1:5]';  plot(X, polyval(a,X))
```

Polynomials in MATLAB (cont.)



Monomial basis assessment

- Simple!
- Matrix A is a *Vandermonde matrix*: nonsingular.
  Hence uniqueness: there is precisely one interpolating polynomial.
- Construction cost $\mathcal{O}(n^3)$ flops (high if n is large)
- Evaluation cost using Horner, $\mathcal{O}(n)$ flops (low)
- Coefficients c_j not indicative of $f(x)$ and all change if data are modified.
- Potential stability difficulties if degree is large or abscissae spread apart.

see Ex 1.4 on page 10 in your textbook