Numerical Analysis, lecture 11:
Approximation
(textbook sections 9.1-3)

- Problem formulation
- Least squares fitting
How to approximate $f$ by a simple continuous function? (p. 261-263)

two curve-fitting problems:

continuous $f$

discrete $f$

We want to determine (the parameters of) $f^*$ so that $f^*$ is “close to” $f$
We’ll need a few abstract concepts (p. 264)

\[ f \text{ is an element of a linear space} \]

- \( C[a,b] \) space of continuous functions on \([a,b]\)
- \( \mathbb{R}^m \) space of \( m \)-dimensional vectors

\[ f^* \text{ is an element of a } n+1 \text{-dimensional subspace} \]

linear combinations of a set of \( n+1 \) functions: \( \text{span}\{\phi_0,\ldots,\phi_n\} \subseteq C[a,b] \)

\( \Pi_n = \text{polynomials of degree } \leq n \)

linear combinations of a set of \( n+1 \) vectors: \( \text{span}\{\phi_0,\ldots,\phi_n\} \subseteq \mathbb{R}^m \) \( (n < m) \)

\( \Pi_n|_G = \text{polynomials evaluated at } G = \{x_1,\ldots,x_m\} \)
The norm of a linear space element  

4 axioms

\[ \|f\| \geq 0, \quad \|f\| = 0 \Rightarrow f = 0, \quad \|\alpha f\| = |\alpha| \cdot \|f\|, \quad \|f + g\| \leq \|f\| + \|g\| \]

Euclidean norm on \( C[a,b] \)

\[ \|f\| = \sqrt{\int_a^b w(x) (f(x))^2 \, dx} \quad \text{where } w > 0 \text{ in } (a,b) \]

Euclidean norm on \( \mathbb{R}^m \)

\[ \|f\| = \sqrt{\sum_{i=1}^m w_i f_i^2} \quad \text{where } w_i > 0 \text{ for } i = 1, \ldots, m \]

Approximation problem

Find \( f^* \) in the subspace that minimizes \( \|f - f^*\| \)
The euclidean norm is associated to a scalar product \( (p. 267, 271) \)

3 axioms

A scalar product is a real-valued function of \( f \) and \( g \) [denoted \( (f,g) \)] such that

\[
(f,g) = (g,f), \quad (f, \alpha h + \beta g) = \alpha (f,h) + \beta (f,g), \quad f \neq 0 \Rightarrow (f,f) > 0
\]

a scalar product in \( C[a,b] \)

\[
(f,g) = \int_a^b w(x)f(x)g(x) \, dx
\]

a scalar product in \( \mathbb{R}^m \)

\[
(f,g) = \sum_{i=1}^m w_i f_i g_i = f^T W g
\]

W is the diagonal matrix with \( w_i \) on the diagonal

2 facts

The euclidean norm \( \|f\| \) is defined as \(\sqrt{(f,f)}\)

\[
(f,g) = 0 \quad \Rightarrow \quad \|f + g\|^2 = \|f\|^2 + \|g\|^2 \quad \text{(pythagorean law)}
\]
The best fit's error is orthogonal to the subspace (p. 269-270)

**Theorem**

\( f^* \) is the unique element of \( \text{span}\{\phi_0, \ldots, \phi_n\} \) that is closest to \( f \)

if and only if \( (f - f^*, \phi_k) = 0 \) for \( k = 0, 1, 2, \ldots, n \).

**Proof (\( \Longleftrightarrow \))**

\[
g = \sum_{j=0}^{n} c_j \phi_j \quad \Rightarrow \quad (f - f^*, f^* - g) = 0
\]

\[
\|f - g\|^2 = \|f - f^* + f^* - g\|^2 = \|f - f^*\|^2 + \|f^* - g\|^2 \geq \|f - f^*\|^2
\]

**Proof (\( \Rightarrow \))**

\[
0 = \frac{\partial}{\partial c_k} \|f - f^*\|^2 = \frac{\partial}{\partial c_k} \left\| f - \sum_{j=0}^{n} c_j \phi_j \right\|^2 = 2(f^* - f, \phi_k)
\]
The approximation’s functions should be linearly independent  

(p. 268-269)

**linear independence definition:**

\[ \sum_{j=0}^{n} c_j \phi_j = 0 \quad \Rightarrow \quad c_0 = \cdots = c_n = 0 \]

the standard polynomial basis set is linearly indept.

If \( \phi_i(x) = x^i \quad (x \in [a,b]) \)

then \( \{ \phi_0, \phi_1, \ldots, \phi_n \} \) is a linearly independent set in \( C[a,b] \)

the sampled std. poly. basis set is lin. indepent.

If \( \phi_i(x) = x^i \quad (x \in G = \{x_1, \ldots, x_m\}) \) and \( n < |G| \)  

\[ |G| \text{ is the number of distinct } x_i \]

then \( \{ \phi_0|_G, \phi_1|_G, \ldots, \phi_n|_G \} \) is a linearly independent set of \( \mathbb{R}^m \) vectors
**Linear independence ensures that the normal equations have a unique solution** (p. 270-271)

**Theorem**

If $\phi_0, \ldots, \phi_n$ are linearly independent then there is a unique $[c_0, \ldots, c_n]$ such that $(f - \sum_{j=0}^{n} c_j \phi_j, \phi_k) = 0$ for $k = 0, 1, 2, \ldots, n$.

**Proof**

$$0 = \left(f - \sum_{j=0}^{n} c_j \phi_j, \phi_k \right) = (f, \phi_k) - \sum_{j=0}^{n} c_j (\phi_j, \phi_k)$$

**Normal equations coefficient matrix is non-singular when $\Phi$’s are lin. indept because**

$$\sum_{j=0}^{n} c_j (\phi_j, \phi_k) = 0 \quad \Rightarrow \quad \left\| \sum_{j=0}^{n} c_j \phi_j \right\|^2 = 0 \quad \Rightarrow \quad \sum_{j=0}^{n} c_j \phi_j = 0 \quad \Rightarrow \quad c_0 = \cdots = c_n = 0$$
The least squares approximation can be found by solving the normal equations (p. 272)

normal equations

\[
\begin{bmatrix}
(\phi_n, \phi_n) & \cdots & (\phi_0, \phi_n) \\
\vdots & \ddots & \vdots \\
(\phi_n, \phi_0) & \cdots & (\phi_0, \phi_0)
\end{bmatrix}
\begin{bmatrix}
c_n \\
\vdots \\
c_0
\end{bmatrix}
=
\begin{bmatrix}
(f, \phi_n) \\
\vdots \\
(f, \phi_0)
\end{bmatrix}
\]

example (p. 272)

\( f \in C[0,1], \ f(x) = \sin(\pi x/2), \ f^* \in \Pi_1, \ w \equiv 1 \)

i.e. "find \( c_0 \) and \( c_1 \) to minimize \[ \frac{\| f - c_0 \phi_0 - c_1 \phi_1 \|}{\sqrt{\int_0^1 \left( \frac{\sin(\pi x)}{2} - c_0 - c_1 x \right)^2 dx}} \]"

\[
\begin{bmatrix}
1/3 & 1/2 \\
1/2 & 1
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_0
\end{bmatrix}
= 
\begin{bmatrix}
4/\pi^2 \\
2/\pi
\end{bmatrix}
\Rightarrow 
\begin{bmatrix}
c_1 \\
c_0
\end{bmatrix}
= 
\begin{bmatrix}
1.0437 \\
0.1148
\end{bmatrix}
\]

\[ \therefore \ f^*(x) = 0.1148 + 1.0437x \]
The discrete least squares approximation can be found by solving the normal equations (p. 273)

normal equation \[ A^T W A c = A^T W f \]

\[
\begin{bmatrix}
    c_n \\
    \vdots \\
    c_0
\end{bmatrix}
= \begin{bmatrix}
    f(x_1) \\
    \vdots \\
    f(x_m)
\end{bmatrix}, \quad
\begin{bmatrix}
    w_1 \\
    \vdots \\
    w_m
\end{bmatrix}
= \begin{bmatrix}
    \phi_n(x_1) & \cdots & \phi_0(x_1) \\
    \vdots & \ddots & \vdots \\
    \phi_n(x_m) & \cdots & \phi_0(x_m)
\end{bmatrix}
\]

example \[ G = \{0, 1/4, 1/2, 3/4, 1\}, \quad f \in C[0,1]|_G, \quad f^* \in \Pi_1|_G, \quad \sum_{i=1}^{5} w_i \left( \sin\left( \frac{\pi x_i}{2} \right) - c_1 x_i - c_0 \right)^2 \]

i.e. find \( c_1, c_0 \) to minimize

\[
\begin{bmatrix}
    \sin(0) \\
    \sin(\pi/8) \\
    \sin(\pi/4) \\
    \sin(3\pi/8) \\
    \sin(\pi/2)
\end{bmatrix}
= \begin{bmatrix}
    0.3438 & 0.5 \\
    0.5 & 1
\end{bmatrix}
\begin{bmatrix}
    c_1 \\
    c_0
\end{bmatrix}
= \begin{bmatrix}
    0.4105 \\
    0.6284
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
    c_1 \\
    c_0
\end{bmatrix}
= \begin{bmatrix}
    1.0275 \\
    0.1147
\end{bmatrix}
\]
The discrete least squares polynomial fit in Matlab

here's the previous slide's example

```matlab
>> m = 5;
>> n = 1;
>> x = linspace(0,1,m)';
>> f = sin(pi*x/2);
>> A = vander(x); A = A(:,end-n:end)
>> w = [0.5 ones(1,m-2) 0.5]'; W = diag(w);
>> c = (A'*W*A)\(A'*W*f)

c =
  1.0275
  0.1147
```

$f^*(x) = 1.0275x + 0.1147$
what happened, what’s next

- seek \( f^* = \sum_{j=0}^{n} c_j \phi_j \) to minimize \( ||f - f^*|| \)
- continuous and discrete euclidean norms
- \( ||f - f^*|| \) is minimized when \( (f - f^*, \varphi_k) = 0 \)
- discrete polynomial normal equation is nonsingular if degree < # distinct x-values

Next lecture: orthogonal polynomials (§9.4 – 6)
Numerical Analysis, lecture 12:

Approximation II

(textbook sections 9.4-6)

- orthogonal functions
- orthogonal polynomials
The best least-squares fit is determined by orthogonality (normal equations)

approximation problem

Given \( f \), find \( f^* \in \text{span}\{\phi_0, \ldots, \phi_n\} \)

that minimizes \( \|f - f^*\| = \sqrt{(f - f^*, f - f^*)} \)

where \( (f, g) = \int_a^b w(x)f(x)g(x)\,dx \) or \( (f, g) = \sum_{i=1}^m w_i f_i g_i \)

its solution

If \( \phi_0, \ldots, \phi_n \) are linearly independent then \( f^* = \sum_{j=0}^n c_j \phi_j \)

where \( (f - \sum_{j=0}^n c_j \phi_j, \phi_k) = 0 \) i.e. \[
\begin{bmatrix}
(\phi_n, \phi_n) & \cdots & (\phi_0, \phi_n) \\
\vdots & \ddots & \vdots \\
(\phi_n, \phi_0) & \cdots & (\phi_0, \phi_0)
\end{bmatrix}
\begin{bmatrix}
c_n \\
\vdots \\
c_0
\end{bmatrix}
= \begin{bmatrix}
(f, \phi_n) \\
\vdots \\
(f, \phi_0)
\end{bmatrix}
\]
Discrete LS-fitting with repeated x-values

f need not be a “sampled C[0,1] function” —
the normal eqn. is nonsingular if |G| > n
(i.e. if the number of distinct nodes is > the degree of fitting polynomial)

example

<table>
<thead>
<tr>
<th>k</th>
<th>x_k</th>
<th>f_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>7.97</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
<td>10.2</td>
</tr>
<tr>
<td>3</td>
<td>2.4</td>
<td>14.2</td>
</tr>
<tr>
<td>4</td>
<td>2.4</td>
<td>14.1</td>
</tr>
<tr>
<td>5</td>
<td>3.2</td>
<td>16.0</td>
</tr>
<tr>
<td>6</td>
<td>4.0</td>
<td>21.2</td>
</tr>
<tr>
<td>7</td>
<td>4.0</td>
<td>21.2</td>
</tr>
</tbody>
</table>

\[
f^* \in \Pi_1|_G, \ w_{1:7} = [1,1,\ldots,1]
\]

\[
\begin{bmatrix}
56.96 & 18.4 \\
18.4 & 7
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_0
\end{bmatrix} =
\begin{bmatrix}
311.416 \\
104.870
\end{bmatrix}
\]

\[c_1 = 4.1606, \ c_0 = 4.0449\]

\[f^*(x) = 4.1606x + 4.0449\]
The standard polynomial basis gives an ill-conditioned Gram matrix (p. 275)

\textbf{example} (p. 275) \quad f \in C[0,1], f^* \in \Pi_n, \ w \equiv 1

\text{i.e. find } c_0, \ldots, c_n \text{ to minimize } \sqrt{\int_0^1 \left( c_0 + \cdots + c_n x^n - f(x) \right)^2} \, dx

\left( \phi_i, \phi_j \right) = \int_0^1 x^i x^j \, dx = \frac{1}{i + j + 1}

\text{>> } \text{hilb}(4)

\text{ans =}

\begin{bmatrix}
1.0000 & 0.5000 & 0.3333 & 0.2500 \\
0.5000 & 0.3333 & 0.2500 & 0.2000 \\
0.3333 & 0.2500 & 0.2000 & 0.1667 \\
0.2500 & 0.2000 & 0.1667 & 0.1429
\end{bmatrix}

\text{>> } \text{cond(hilb}(4))

\text{ans =}

\begin{bmatrix}
1.5514e+04
\end{bmatrix}

\text{>> } \text{cond(hilb}(10))

\text{ans =}

\begin{bmatrix}
1.6025e+13
\end{bmatrix}
The standard polynomial basis functions are “nearly” linearly dependent.
Least-squares approximation is best done with orthogonal basis functions (p. 275)

**orthogonality** \[ i \neq j \implies (\phi_i, \phi_j) = 0 \]

**normal eqns**

\[
\begin{bmatrix}
(\phi_n, \phi_n) & 0 & 0 & 0 \\
0 & \ddots & \vdots & \vdots \\
0 & \cdots & (\phi_1, \phi_1) & 0 \\
0 & \cdots & 0 & (\phi_0, \phi_0)
\end{bmatrix}
\begin{bmatrix}
c_n \\
c_1 \\
c_0
\end{bmatrix}
= 
\begin{bmatrix}
(f, \phi_n) \\
(f, \phi_1) \\
(f, \phi_0)
\end{bmatrix}
\]

**best LS approximation**

If \( \phi_0, \ldots, \phi_n \) are nonzero & orthogonal then

\[
f^* = \sum_{j=0}^{n} c_j \phi_j \quad \text{where} \quad c_j = \frac{(f, \phi_j)}{(\phi_j, \phi_j)}
\]
Orthogonal basis = set of smallest monic polynomials = Gram-Schmidt method (p. 283-284)

**Theorem** (p. 283)

For $k = 0, 1, \ldots, n$, let $P_k$ be the smallest monic polynomial of degree $k$. Then $\{P_0, \ldots, P_n\}$ is orthogonal.

**Proof**

$P_k$ is the smallest monic polynomial of degree $k$

$\iff P_k$ is the monic polynomial of degree $k$ that minimizes $\|x^k - (x^k - P_k)\|$

$\iff x^k - P_k$ is the polynomial of degree $\leq k - 1$ that is closest to $x^k$

$\iff x^k - (x^k - P_k)$ is orthogonal to every polynomial of degree $\leq k - 1$

A set $\{P_0, \ldots, P_n\}$ of polynomials with $\deg(P_k) = k$ is orthogonal if and only if each $P_k$ is orthogonal to all polynomials of degree $< k$.

**Gram-Schmidt**

The polynomial of degree $\leq k - 1$ that is closest to $x^k$ is

$$P_k = x^k - \sum_{j=0}^{k-1} \frac{(P_j, x^k)}{(P_j, P_j)} P_j$$
Orthogonal polynomials can be found using the Gram-Schmidt procedure

If \( \phi_0, \ldots, \phi_n \) are linearly independent then \( P_0, \ldots, P_n \) defined by

\[
P_k = \phi_k - \sum_{j=0}^{k-1} \frac{(P_j, \phi_k)}{(P_j, P_j)} P_j \quad (k = 0, 1, \ldots, n)
\]

are orthogonal and \( \text{span}\{P_0, \ldots, P_n\} = \text{span}\{\phi_0, \ldots, \phi_n\} \).

**Example** For inner product \( (f, g) = \int_{-1}^{1} f(x)g(x) \, dx \),

\[
P_0(x) = 1
\]

\[
P_1(x) = x - c_0 P_0(x)
\]

\[
c_0 = \frac{(x, P_0)}{(P_0, P_0)} = 0 \quad \Rightarrow \quad P_1(x) = x
\]

\[
P_2(x) = x^2 - (c_0 P_0(x) + c_1 P_1(x))
\]

\[
c_0 = \frac{(x^2, P_0)}{(P_0, P_0)} = \frac{1}{3}, \quad c_1 = \frac{(x^2, P_1)}{(P_1, P_1)} = 0 \quad \Rightarrow \quad P_2(x) = x^2 - \frac{1}{3}
\]

\[
P_3(x) = x^3 - (c_0 P_0(x) + c_1 P_1(x) + c_2 P_2(x))
\]

\[
c_0 = \frac{(x^3, P_0)}{(P_0, P_0)} = 0, \quad c_1 = \frac{(x^3, P_1)}{(P_1, P_1)} = \frac{2}{5} = \frac{3}{5}, \quad c_2 = \frac{(x^3, P_2)}{(P_2, P_2)} = 0 \quad \Rightarrow \quad P_3(x) = x^3 - \frac{3}{5} x
\]
Legendre polynomials are orthogonal on [-1,1] with unit weight (p. 281–283)

Recursion

\[ P_0(x) = 1, \quad P_1(x) = x, \]
\[ P_{n+1}(x) = \frac{2n+1}{n+1} xP_n(x) - \frac{n}{n+1} P_{n-1}(x), \quad n = 1, 2, \ldots \]

First five

\[ P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \frac{1}{2}(3x^2 - 1), \]
\[ P_3(x) = \frac{1}{2}(5x^2 - 3x), \quad P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3) \]

Orthogonality

\[
\int_{-1}^{1} P_k(x)P_n(x) \, dx = \begin{cases} 
0 & \text{for } k \neq n, \\
\frac{2}{2n+1} & \text{for } k = n.
\end{cases}
\]
Orthogonal polynomials can also be generated by a three-term recurrence formula (p. 277-279)

\[ P_0(x) = A_0 \]
\[ P_1(x) = (\alpha_0 x - \beta_0)P_0(x) \]
\[ P_{k+1}(x) = (\alpha_k x - \beta_k)P_k(x) - \gamma_k P_{k-1}(x), \quad k = 1, 2, \ldots \]

where

\[ \alpha_k = \frac{A_{k+1}}{A_k}, \quad k = 0, 1, 2, \ldots \]
\[ \beta_k = \frac{\alpha_k (xP_k, P_k)}{(P_k, P_k)}, \quad k = 0, 1, 2, \ldots \]
\[ \gamma_k = \frac{\alpha_k (P_k, P_k)}{\alpha_{k-1} (P_{k-1}, P_{k-1})}, \quad k = 1, 2, \ldots \]
Discretely-orthogonal polynomials can be computed using the 3-term recurrence (p. 279)

**example**

Find monic polynomials of degree 0, 1, 2 that are orthogonal with respect to 
\[(f,g) = \frac{1}{2} f(0)g(0) + f(\frac{1}{4})g(\frac{1}{4}) + f(\frac{1}{2})g(\frac{1}{2}) + f(\frac{3}{4})g(\frac{3}{4}) + \frac{1}{2} f(1)g(1)\]
and use them to approximate \(f(x) = \sin \pi x/2\) on \([0,1]\).

\[
P_0 = 1
\]

\[
(P_0,P_0) = 4, \quad (xP_0,P_0) = 2, \quad \beta_0 = \frac{1}{2}, \quad P_1(x) = x - \frac{1}{2}
\]

\[
(P_1,P_1) = \frac{3}{8}, \quad (xP_1,P_1) = \frac{3}{16}, \quad \beta_1 = \frac{1}{2}, \quad \gamma_1 = \frac{3}{32}, \quad P_2(x) = (x - \frac{1}{2})(x - \frac{1}{2}) - \frac{3}{32}
\]

\[
c_0 = \frac{(f,P_0)}{(P_0,P_0)} = 0.6284, \quad c_1 = \frac{(f,P_1)}{(P_1,P_1)} = 1.0275, \quad c_2 = \frac{(f,P_2)}{(P_2,P_2)} = -0.8248
\]

\[
f^*(x) = 0.6284 + 1.0275(x - 0.5) - 0.8248\left((x - 0.5)^2 - 0.09375\right)
\]
Discretely-orthogonal polynomials
in Matlab  (p. 279-280)

function \([b,g,c] = \text{orthpolfit}(x,y,w,n)\)
\[x = x(:); y = y(:); w = w(:);\]
\[m = \text{length}(x);\]
\[b = \text{zeros}(1,\text{max}(1,n)); \quad g = b;\]
\[P = [\text{zeros}(m,1) \ \text{ones}(m,1)];\]
\[s = [\text{sum}(w) \ \text{zeros}(1,n)];\]
\[c = [\text{sum}(w.*y)/s(1) \ \text{zeros}(1,n)];\]
for \(k = 1:n\)
    \[b(k) = \text{sum}(w .* x .* P(:,2).^2) / s(k);\]
    if \(k == 1\)
        \[g(k) = 0;\]
    else
        \[g(k) = s(k)/s(k-1);\]
    end
    \[P = [P(:,2) \ (x-b(k)).*P(:,2)-g(k)*P(:,1)];\]
    \[s(k+1) = \text{sum}(w .* P(:,2).^2);\]
    \[c(k+1) = \text{sum}(w .* P(:,2).*y) / s(k+1);\]
end

\[f^*(x) = 0.6284 + 1.0275(x - 0.5)\]
\[- 0.8248((x - 0.5)^2 - 0.09375)\]

>> m = 5;
>> x = linspace(0,1,m);
>> f = sin(pi*x/2);
>> w = [0.5 ones(1,m-2) .5];
>> \([b,g,c] = \text{orthpolfit}(x,f,w,2)\)
\[b = \]
\[0.5000 \quad 0.5000\]
\[g = \]
\[0 \quad 0.09375\]
\[c = \]
\[0.6284 \quad 1.0275 \quad -0.8248\]
The linear combination of orthogonal polynomials can be efficiently computed (p. 280-281)

Clenshaw’s algorithm

\[ c_0 + \sum_{k=1}^{n} c_k P_k(x) = u_0 \] where \( u_n = c_n \), \( u_{n-1} = c_{n-1} + (x - \beta_{n-1})u_n \),

and \( u_j = c_j + (x - \beta_j)u_{j+1} - \gamma_{j+1}u_{j+2} \) for \( j = n-2, n-1, \ldots, 0 \).

function \( u = \text{orthpolval}(b,g,c,x) \)

\begin{verbatim}
function u = orthpolval(b,g,c,x)
n = length(c)-1;
u = c(end)*ones(size(x));
if n > 0
    ujp1 = u;
    u = c(end-1) + (x-b(n)).*ujp1;
    for j = n-2:-1:0
        ujp2 = ujp1;
        ujp1 = u;
        u = c(j+1) + (x-b(j+1)).*ujp1 ...
            - g(j+2)*ujp2;
    end
end
end
\end{verbatim}

example (cont’d)

\[ t = \text{linspace}(0,1,50); \]
\[ \text{plot}(t,\sin(pi*t/2)-\text{orthpolval}(b,g,c,t)) \]
what happened, what’s next

• least-squares approximation is best done with orthogonal basis functions
  ‣ standard polynomial basis gives normal equation with ill-conditioned full matrix

• orthogonal polynomials can be found using
  ‣ Gram-Schmidt or
  ‣ 3-term recurrence

Next lecture: solving differential equations (§10.1 – 4)