Announcements

Exam 3
Recitation section
  Monday 2:30  Loomis 151

Review
  Valm, condition
  Runge's phenomenon

Goals
Lagrange Polynomials: General Form

\[ \varphi_j(x) = \frac{\prod_{k=1, k \neq j}^{m} (x - x_k)}{\prod_{k=1, k \neq j}^{m} (x_j - x_k)} \]

Write down the Lagrange interpolant for nodes \((x_i)_{i=1}^{m}\) and values \((y_i)_{i=1}^{m}\).

\[ p_{m-1}(x) = \sum_{j=1}^{m} y_j \varphi_j(x) \]
Newton Interpolation

Find a basis so that $V$ is triangular.

$$
\phi_j(x) = \prod_{k=1}^{j-1} (x - x_k) \quad (j = 1 \ldots n)
$$

forward subsh on $\Delta$ vdm: $O(n^2)$

Why not Lagrange/Newton?
Newton Interpolation

Find a basis so that $V$ is triangular.

\[ \phi_j(x) = \prod_{k=1}^{j-1} (x-x_k) \quad (j=1, \ldots, n) \]

forward subsh on $\vartriangle$ vdm: $O(n^2)$

Why not Lagrange/Newton?

Cheap to form, expensive to evaluate, expensive to do calculus on.
Better conditioning: Orthogonal polynomials

What caused monomials to have a terribly conditioned Vandermonde?

What’s a way to make sure two vectors are *not* like that?

But polynomials are functions!
Better conditioning: Orthogonal polynomials

What caused monomials to have a terribly conditioned Vandermonde?

Being close to linearly dependent.

What’s a way to make sure two vectors are \textit{not} like that?

Orthogonality

But polynomials are functions!
Orthogonality of Functions

How can functions be orthogonal?

\[ \langle f, g \rangle = \int f(x)g(x)dx \]

\[ \langle f, g \rangle = \int f(x)g(x)dx \]

\[ \|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int f^2(x)dx} \]
Constructing Orthogonal Polynomials

How can we find an orthogonal basis?

Demo: Orthogonal Polynomials [cleared] — Got: Legendre polynomials. But how can I practically compute the Legendre polynomials?

Three-term recurrence:

\[ P_0 = 1 \quad P_1 = x \]

\[ (n+1) \cdot P_{n+1} = (2n+1) \cdot x \cdot P_n - n \cdot P_{n-1} \]

Idea: Generalize the inner product to include weight:

\[ (f, g)_w = \int \frac{f(x) \cdot g(x) \cdot w(x) \, dx}{w(x)} \]
Three equivalent definitions:

- Result of Gram-Schmidt with weight $\frac{1}{\sqrt{1-x^2}}$. What is that weight?
- $T_k(x) = \cos(k \cos^{-1}(x))$
- $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ plus $T_0 = 1$, $T_1 = x$

(Like for Legendre, you won’t exactly get the standard normalization if you do this.)

half-circle
Chebyshev Interpolation

What is the Vandermonde matrix for Chebyshev polynomials?

Let $\theta_i = \cos\left(\frac{i}{k} \pi\right)$ for $i = 0, 1, \ldots, k$.

These are minimal maxima of $\Gamma_k$.

$V_{ij} = T_j(\theta_i) = \cos\left(j \cos^{-1}\left(\cos\left(\frac{i}{k} \pi\right)\right)\right)

= \cos\left(\frac{i \cdot j}{k} \pi\right)$

To create cosine transform matrix $c$ and inverse available in $O(k \log k)$ time.
Chebyshev Nodes

Might also consider roots (instead of extrema) of $T_k$:

$$x_i = \cos \left( \frac{2i - 1}{2k} \pi \right) \quad (i = 1 \ldots, k).$$

Vandermonde for these (with $T_k$) can be applied in $O(N \log N)$ time, too.

Edge-clustering seemed like a good thing in interpolation nodes. Do these do that?

**Yes**

**Demo:** Chebyshev Interpolation [cleared] (Part I-IV)
Truncation Error in Interpolation

If $f$ is $n$ times continuously differentiable on a closed interval $I$ and $p_{n-1}(x)$ is a polynomial of degree at most $n$ that interpolates $f$ at $n$ distinct points $\{x_i\}$ ($i = 1, \ldots, n$) in that interval, then for each $x$ in the interval there exists $\xi$ in that interval such that

$$f(x) - p_{n-1}(x) = \frac{f^{(n)}(\xi(x))}{n!}(x - x_1)(x - x_2)\cdots(x - x_n).$$

$$R(x) := f(x) - p_{n-1}(x)$$

$$\gamma_x(t) = R(t) - \frac{R(x)}{W(x)}W(t)$$

(\text{let } x \in I \setminus \{x_1, \ldots, x_n\}).$$
Truncation Error in Interpolation: cont’d.

\[ R(t) = \phi(t) - \rho_{n-1}(t) \]

\[ Y_x(t) = R(t) - \frac{R(x)}{W(x)} W(t) \quad \text{where} \quad W(t) = \prod_{i=1}^{n} (t - x_i) \]

- \( x_i \) are roots of \( R(t) \) and \( W(t) \). \( Y_x(x) = 0 \). \( Y_x \) has \( n+1 \) roots. \((x\) and the nodes\)
- \( Y_x' \) has \( n \) roots \( \ldots \). \( Y_x^{(n)} \) has 1 root in \( I \).

Let’s call that root \( \xi \).

\[ Y_x^{(n)}(t) = \frac{\phi^{(n)}(t)}{W(x)} - \frac{R(x)}{W(x)} n! \quad \text{where} \quad W(x) \frac{\phi^{(n)}(\xi)}{n!} = R(x) \]

\[ 0 = Y_x^{(n)}(\xi) = \phi^{(n)}(\xi) - \frac{R(x)}{W(x)} n! \]
Error Result: Simplified Form

Boil the error result down to a simpler form.

\[ f(x) - p_{n-1}(x) = \frac{f^{(n)}(\xi(x))}{n!} (x - x_1)(x - x_2) \cdots (x - x_n). \]

Assume \( x_i < \cdots < x_n \),
\[ |p^{(n-1)}(x)| \leq M \quad (x \in I) \]
\( n = \text{length of the interval} \)

\[ \max_{x \in I} |f(x) - p_{n-1}(x)| \leq C M h^n \]

▶ **Demo:** Interpolation Error [cleared]

\[ \mathcal{E}(h) = O(h^n) \]

n-th order convergence
Going piecewise: Simplest Case

Construct a piecewise linear interpolant at four points.

<table>
<thead>
<tr>
<th>$x_0, y_0$</th>
<th>$x_1, y_1$</th>
<th>$x_2, y_2$</th>
<th>$x_3, y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1 = a_1 x + b_1$</td>
<td>$f_2 = a_2 x + b_2$</td>
<td>$f_3 = a_3 x + b_3$</td>
<td></td>
</tr>
<tr>
<td>2 unk.</td>
<td>2 unk.</td>
<td>2 unk.</td>
<td></td>
</tr>
<tr>
<td>$f_1(x_0) = y_0$</td>
<td>$f_2(x_1) = y_1$</td>
<td>$f_3(x_2) = y_2$</td>
<td></td>
</tr>
<tr>
<td>$f_1(x_1) = y_1$</td>
<td>$f_2(x_2) = y_2$</td>
<td>$f_3(x_3) = y_3$</td>
<td></td>
</tr>
<tr>
<td>2 eqn.</td>
<td>2 eqn.</td>
<td>2 eqn.</td>
<td></td>
</tr>
</tbody>
</table>

Why three intervals?

(to be covered after break)
Piecewise Cubic (‘Splines’)

\[\begin{align*}
\text{x}_0, \text{y}_0 & & \text{x}_1, \text{y}_1 & & \text{x}_2, \text{y}_2 & & \text{x}_3, \text{y}_3 \\
& & f_1 & & f_2 & & f_3 \\
& & \frac{a_1 x^3 + b_1 x^2 + c_1 x + d_1}{x_1 - x_0} & & \frac{a_2 x^3 + b_2 x^2 + c_2 x + d_2}{x_2 - x_1} & & \frac{a_3 x^3 + b_3 x^2 + c_3 x + d_3}{x_3 - x_2} \\
& & \frac{x - x_0}{x_1 - x_0} & & \frac{x - x_1}{x_2 - x_1} & & \frac{x - x_2}{x_3 - x_2}
\end{align*}\]

(to be covered after break)
### Piecewise Cubic (‘Splines’): Accounting

<table>
<thead>
<tr>
<th>$x_0, y_0$</th>
<th>$f_1$</th>
<th>$x_1, y_1$</th>
<th>$f_2$</th>
<th>$x_2, y_2$</th>
<th>$f_3$</th>
<th>$x_3, y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a_1 x^3 + b_1 x^2 + c_1 x + d_1$</td>
<td></td>
<td>$a_2 x^3 + b_2 x^2 + c_2 x + d_2$</td>
<td></td>
<td>$a_3 x^3 + b_3 x^2 + c_3 x + d_3$</td>
<td></td>
</tr>
</tbody>
</table>

(to be covered after break)
Outline

Introduction to Scientific Computing

Systems of Linear Equations

Linear Least Squares

Eigenvalue Problems

Nonlinear Equations

Optimization

Interpolation

Numerical Integration and Differentiation
   Numerical Integration
   Quadrature Methods
   Accuracy and Stability
   Gaussian Quadrature
   Composite Quadrature
   Numerical Differentiation
   Richardson Extrapolation

Initial Value Problems for ODEs

Boundary Value Problems for ODEs

Partial Differential Equations and Sparse Linear Algebra

Fast Fourier Transform

Additional Topics
Derive the (absolute) condition number for numerical integration.

(to be covered after break)
Interpolatory Quadrature: Examples

\[ f(x) \approx \sum_{i=1}^{n} f(x_i) \ell_i(x) \]

\[ \int_{a}^{b} f(x) \, dx \approx \sum_{i=1}^{n} f(x_i) \int_{a}^{b} \ell_i(x) \, dx \]

"Quadrature rule"  
\( x_i \): nodes  
\( w_i \): weights

"Square" = "Quadrat" in German
Interpolatory Quadrature: Computing Weights

How do the weights in interpolatory quadrature get computed?

Demo: Newton-Cotes weight finder [cleared]