Announcements

Exam 3

Recitation section

Monday 7:30 Loomis 151

Review

Vdm, conditioning

Runge's phenomena

Goals

Lagrange Polynomials: General Form

$V = I$

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

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.

$$\varphi_j(x) = \prod_{k=1}^{j-1} (x - x_k) \qquad (j = 1 \dots n)$$

empty product $(j=1)$: 1

forward subst on $\triangle$ Vdm: $O(n^2)$

Why not Lagrange/Newton?

Newton Interpolation

Find a basis so that V is triangular.

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

empty product $(j=1)$: 1

forward subst on $\triangle$ 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 *not* like that?

Orthogonality

But polynomials are functions!

Orthogonality of Functions

$[-1,1]$

$f(x)\ ,\ g(x)$

How can functions be orthogonal?

$$\vec{f}\cdot\vec{g} = \sum f_i g_i = (\vec{f},\vec{g})$$

$$(f,g) = \int_{-1}^{1} f(x)\, g(x)\, dx$$

$$\|f\| = \sqrt{(f,f)} = \sqrt{\int f^2(x)\, dx}$$

Constructing Orthogonal Polynomials

How can we find an orthogonal basis?

Gram-Schmidt

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

Three-term recurrence:

$$P_0 = 1 \qquad P_1 = x$$

$$(n+1) P_{n+1} = (2n+1) x P_n - n P_{n-1}$$

Idea: Generalize the inner product to include weight:

$$(f, g)_w = \int_{-1}^{1} f(x) g(x) w(x) \, dx$$

Chebyshev Polynomials: Definitions

Three equivalent definitions:

- Result of Gram-Schmidt with weight $1/\sqrt{1-x^2}$. What is that weight?

  half-circle

  (Like for Legendre, you won't exactly get the standard normalization if you do this.)
- $T_k(x) = \cos(k\cos^{-1}(x))$
- $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ plus $T_0 = 1,\ T_1 = x$

Chebyshev Interpolation

What is the Vandermonde matrix for Chebyshev polynomials?

What nodes?

$$T_k(x) = \cos(k \cos^{-1}(x))$$

$$x_i = \cos\left(\frac{i}{k}\pi\right) \qquad (i = 0, 1, \dots, k)$$

These are minima/maxima of T_k

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

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

Discrete cosine transform

matrix and inverse available in $O(k \log k)$ time.

Chebyshev Nodes

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

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

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 $\{x_i\}$ $(i = 1, ..., n)$ in that interval, then for each x in the interval there exists ξ in that interval such that

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

$W(x)$

error is zero at nodes!

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

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

Let $x \in I \setminus \{x_1, \dots, x_n\}$.

Truncation Error in Interpolation: cont'd.

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

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

- 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 .... $Y_x^{(n)}$ has 1 root in I.

  Let's call that root ξ.

$$Y_x^{(n)}(t) = f^{(n)}(t) - \frac{R(x)}{W(x)} n! \qquad W(x) \frac{f^{(n)}(\xi)}{n!} = R(x)$$

$$0 = Y^{(n)}(\xi) = f^{(n)}(\xi) - \frac{R(x)}{W(x)} n!$$

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 $x_1 < \cdots < x_n$

$|f^{(n)}(x)| \le M \quad (x \in I)$

h = length of the interval

Observe: $|x - x_i| \le h$ ($x \in I$)

$$\max_x |f(x) - p_{n-1}(x)| \le C M h^n$$

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

$E(h) = O(h^n)$

n-th order convergence

Going piecewise: Simplest Case

Construct a pieceweise linear interpolant at four points.

x_0, y_0		x_1, y_1		x_2, y_2		x_3, y_3
	$f_1 = a_1 x + b_1$		$f_2 = a_2 x + b_2$		$f_3 = a_3 x + b_3$	
	2 unk.		2 unk.		2 unk.	
	$f_1(x_0) = y_0$		$f_2(x_1) = y_1$		$f_3(x_2) = y_2$	
	$f_1(x_1) = y_1$		$f_2(x_2) = y_2$		$f_3(x_3) = y_3$	
	2 eqn.		2 eqn.		2 eqn.	

Why three intervals?

(to be covered after break)

Piecewise Cubic ('Splines')

x_0, y_0		x_1, y_1		x_2, y_2		x_3, y_3
|	f_1	|	f_2	|	f_3	|
|	$a_1x^3 + b_1x^2 + c_1x + d_1$	|	$a_2x^3 + b_2x^2 + c_2x + d_2$	|	$a_3x^3 + b_3x^2 + c_3x + d_3$	|

(to be covered after break)

Piecewise Cubic ('Splines'): Accounting

x_0, y_0		x_1, y_1		x_2, y_2		x_3, y_3
	f_1		f_2		f_3	
	$a_1x^3 + b_1x^2 + c_1x + d_1$		$a_2x^3 + b_2x^2 + c_2x + d_2$		$a_3x^3 + b_3x^2 + c_3x + d_3$	

(to be covered after break)

Outline

Conditioning

Derive the (absolute) condition number for numerical integration.

(to be covered after break)

Interpolatory Quadrature: Examples

l_i : Lagrange basis

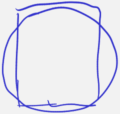
$$f(x) \approx \sum_{i=1}^{n} f(x_i)\, l_i(x)$$

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} f(x_i) \underbrace{\int_a^b l_i(x)\,dx}_{w_i}$$

"Quadrature rule"

x_i : nodes

weights



"square" = "Quadrat" in German

Interpolatory Quadrature: Computing Weights

How do the weights in interpolatory quadrature get computed?

Demo: Newton-Cotes weight finder [cleared]