CS 450: Numerical Anlaysis[1]

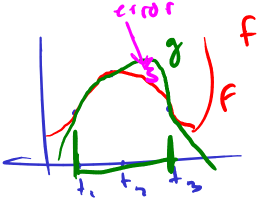
Interpolation

University of Illinois at Urbana-Champaign

[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

Interpolation

- Given $(t_1, y_1), \ldots, (t_m, y_m)$ with *nodes* $t_1 < \cdots < t_m$ an *interpolant* f satisfies:



$f(t_i) = y_i$

· infinite # of interpolant

· (m-1)-degree polynomial interpolant is unique

· error $\max_{t \in [t_1, t_m]} |f(t) - g(t)|$

true function

- Interpolant is usually constructed as linear combinations of *basis functions* $\{\phi_j\}_{j=1}^n = \phi_1, \ldots, \phi_n$ so $f(t) = \sum_j x_j \phi_j(t)$.

$1, t, t^2, \ldots, t^{n-1}$

coefficients

$\varphi(t) = y$

· interpolant exists and is unique if $n = m$

· Vandermonde matrix A, $a_{ij} = \varphi_j(t_i)$

$y = Ax$ $\qquad y_i = \sum_j \varphi_j(t_i) x_j$

Polynomial Interpolation

- The choice of *monomials* as basis functions, $\phi_j(t) = t^{j-1}$ yields a degree $n-1$ polynomial interpolant:

$a_{ij} = t_i^{j-1}$

- Polynomial interpolants are easy to evaluate and do calculus on:

Horner's rule

$f(t) = x_1 + t(x_2 + t(x_3 + \dots))$

n products and n-1 additions

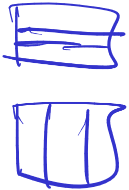
Conditioning of Interpolation

- Conditioning of interpolation matrix $\boldsymbol{A}$ depends on basis functions and coordinates $t_1, \dots, t_m$:

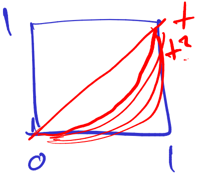
  $\varphi_1, \dots, \varphi_n$

  $t_i \approx t_j \Rightarrow \varphi_k(t_i) \approx \varphi_k(t_j)$

  $\varphi_i \approx \varphi_j \Rightarrow \varphi_i(t_k) \approx \varphi_j(t_k)$



- The Vandermonde matrix tends to be ill-conditioned:



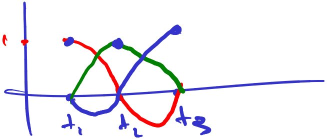
$\kappa(A) \sim \Theta(2^n)$

Lagrange Basis

$A = I \Leftarrow e_j(t_j) = 1, \; e_j(t_i) = 0 \; i \neq j$

- n-points fully define the unique $(n-1)$-degree polynomial interpolant in the *Lagrange basis*:

$$e_j(t) = \begin{cases} 1 & \text{if } t = t_j \\ 0 & \text{if } t = t_i \text{ for } i \neq j \end{cases}$$



$Ix = y$

$x = y$

well

$$e_j(t) = \prod_{k=1, k \neq j}^{n} (t - t_k) \Big/ \prod_{k=1, k \neq j}^{n} (t_j - t_k)$$

- Lagrange polynomials yield an ideal Vandermonde system, but the basis functions are hard to evaluate and do calculus on:

naively, evaluation requires $O(n^2)$ work

Newton Basis

- The *Newton basis* functions $\phi_j(t) = \prod_{k=1}^{j-1}(t - t_k)$ with $\phi_1(t) = 1$ seek the best of monomial and Lagrange bases:

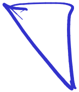
$$e_j(t) = e_{j-1}(t)(t - t_j) \qquad O(n)$$

. enables fast computation of coefficients

. divided difference formula $O(n^2)$

- The Newton basis yields a triangular Vandermonde system:

a_{ij} for $i < j$? $\quad e_j(t_i) = \prod_{k=1}^{j-1}(t_i - t_j) = 0$

$A =$ 

$Ax = b$ with cost $O(n^2)$

Orthogonal Polynomials

- Recall that good conditioning for interpolation is achieved by constructing a well-conditioned Vandermonde matrix, which is the case when the columns (corresponding to each basis function) are orthonormal. To construct robust basis sets, we introduce a notion of *orthonormal functions*:

$$\langle p, q\rangle_{\overset{\uparrow}{w}\text{ weight-function}} = \int_{-\infty}^{\infty} p(t)\, q(t)\, w(t)\, dt$$

$e_1 \ldots e_n$ are orthonormal if

$$\langle e_i, e_j\rangle_w = \delta_{ij} = \begin{bmatrix} \ddots \end{bmatrix}$$

$$\|e\| = \sqrt{\langle e, e\rangle_w}$$

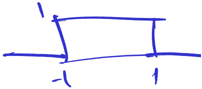
Legendre Polynomials

- The Gram-Schmidt orthogonalization procedure can be used to obtain an orthonormal basis with the same span as any given arbitrary basis:

$\psi_1, \dots, \psi_n$ (non-orthonormal basis)

$\hat{\phi}_k(t) = \frac{\psi_k(t)}{\|\psi_k(t)\|}$, $e_k(t) = \hat{e}_k(t) - \sum_{i=1}^{k-1} \langle \hat{\psi}_k(t), \phi_i(t) \rangle_w e_i(t)$

- The *Legendre polynomials* are obtained by Gram-Schmidt on the monomial basis, with $w(t) = \begin{cases} 1 : -1 \le t \le 1 \\ 0 : \text{otherwise} \end{cases}$ and normalized so $\hat{\phi}_i(1) = 1$.



$\{1, t, t^2\}$

$e_1(t) = \frac{1}{2}$, $e_2(t) = t - \underbrace{\int_{-1}^{1} \frac{1}{2} \cdot t}_{0} = t$

$(3t^2 - 1)/2$

$e_3(t) = t^2 - \int_{-1}^{1} \frac{1}{2} \cdot t^2 \, dt - \int_{-1}^{1} t \cdot t^2 = t^2 - 1/3$

Chebyshev Basis

- *Chebyshev polynomials* $\phi_j(t) = \cos((j-1)\arccos(t))$ and *Chebyshev nodes* $t_i = \cos\left(\frac{2i-1}{2n}\pi\right)$ provide a way to pick *nodes* $t_1, \dots, t_n$ along with a basis, to yield perfect conditioning:

- $\varphi_1(t) = 1$, $\varphi_2(t) = t$, $\varphi_{i+1}(t) = 2t\,\varphi_i(t) - \varphi_{i-1}(t)$

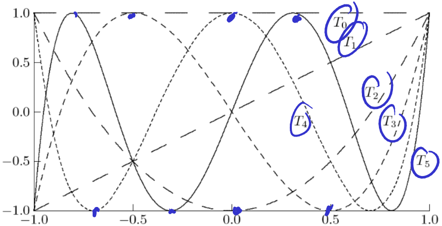
- orthonormal w.r.t.

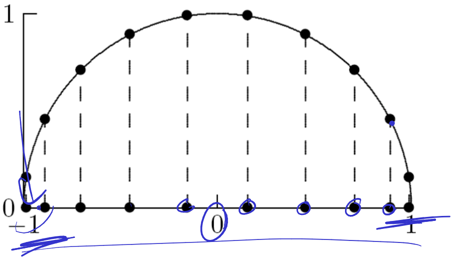
$$w(t) = \begin{cases} 1/(1-t^2)^{1/2} & : -1 \le t \le 1 \\ 0 & : \text{otherwise} \end{cases}$$

- columns of Vandermonde matrix w. Chebyshev basis & nodes are orthogonal

Chebyshev Nodes Intuition

***Demo:** Jump with Chebyshev Nodes*





- Note *equi-oscillation* property, successive extrema of $T_k = \phi_k$ have the same magnitude but opposite sign.
- Set of k Chebyshev nodes of are given by zeros of T_k and are abscissas of points uniformly spaced on the unit circle.

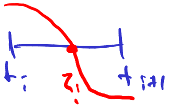
Error in Interpolation

We show by induction that given degree n polynomial interpolant $\tilde{f}$ of f the error $E(t) = f(t) - \tilde{f}(t)$ has n zeros $t_1, \ldots, t_n$ and there exist $y_1, \ldots, y_n$ so

$$E(t) = \int_{t_1}^{t} \int_{y_1}^{w_0} \cdots \int_{y_n}^{w_{n-1}} f^{(n+1)}(w_n) dw_n \cdots dw_0 \tag{1}$$

$$E(t) = \underbrace{E(t_1)}_{=0} + \int_{t_1}^{t} E'(w_0)\, dw_0$$

$E'(t)$



for each (t_i, t_{i+1})

$$\int_{t_i}^{t_{i+1}} E'(t)\, dt = \underbrace{E(t_{i+1})}_{=0} - \underbrace{E(t_i)}_{=0} = 0$$

there are $n-1$ zeros $z_i \in (t_i, t_{i+1})$, $E'(z_i) = 0$

$$E'(w_0) = \int_{z_1}^{w_0} \int_{y_2}^{w_1} \cdots \int_{y_n}^{w_{n-1}} f^{(n+1)}(w_n)\, dw_n \cdots dw_1$$

Interpolation Error Bounds

- Consequently, polynomial interpolation satisfies the following error bound:

$$|E(t)| \le \max_{s \in [t_1, t_n]} \frac{|f^{(n+1)}(s)|}{n!} \prod_{i=1}^{n} (t - t_i)$$

$h = t_n - t_1$

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

- Letting $h = t_n - t_1$ (often also achieve same for h as the node-spacing $t_{i+1} - t_i$), we obtain

Piecewise Polynomial Interpolation

- The kth piece of the interpolant is typically chosen as polynomial on $[t_i, t_{i+1}]$

- *Hermite* interpolation ensures consecutive interpolant pieces have same derivative at each *knot* t_i:

Spline Interpolation

- A *spline* is a $(k-1)$-time differentiable piecewise polynomial of degree k:

- The resulting interpolant coefficients are again determined by an appropriate *generalized Vandermonde system*:

B-Splines

B-splines provide an effective way of constructing splines from a basis:

- The basis functions can be defined recursively with respect to degree:
- ϕ_i^1 is a linear hat function that increases from 0 to 1 on $[t_i, t_{i+1}]$ and decreases from 1 to 0 on $[t_{i+1}, t_{i+2}]$.
- ϕ_i^k is is positive on $[t_i, t_{i+k+1}]$ and zero elsewhere.
- The B-spline basis spans all possible splines of degree k with nodes $\{t_i\}_{i=1}^n$.
- The B-spline basis coefficients are determined by a Vandermonde system that is lower-triangular and banded (has k subdiagonals), and need not contain differentiability constraints, since $f(t)$ is a sum of ϕ_i^ks.