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:

$$f(t_i) = y_i \quad \forall i.$$

  - *The number of possible interpolant functions is infinite, but there is a unique degree $m-1$ polynomial interpolant.*
  - *Error of interpolant can be quantified with knowledge of true function g, (e.g. by considering $\max_{t \in [t_1, t_m]} |f(t) - g(t)|$) .*
- Interpolant is usually constructed as linear combinations of *basis functions* $\{\phi_j\}_{j=1}^n = \phi_1, \ldots, \phi_n$ so $f(t) = \sum_j x_j \phi_j(t)$.
  - *Interpolant exists if $n \geq m$ and is unique for a given basis if $n = m$.*
  - *Vandermonde-like matrix $\boldsymbol{A} = \boldsymbol{V}(\boldsymbol{t}, \{\phi_j\}_{j=1}^n)$ satisfies $a_{ij} = \phi_j(t_i)$ so $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.*
  - *Coefficients $\boldsymbol{x}$ of interpolant are obtained by solving Vandermonde system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$.*

Polynomial Interpolation

- The choice of *monomials* as basis functions, $\phi_j(t) = t^{j-1}$ yields a degree $n-1$ polynomial interpolant:
  - *Corresponding matrix is Vandermonde,* $\boldsymbol{A} = \boldsymbol{V}(\boldsymbol{t}, \{t^{j-1}\}_{j=1}^{n})$ *satisfies* $a_{ij} = t_i^{j-1}$*.*
- Polynomial interpolants are easy to evaluate and do calculus on:
  - *Horner's rule requires* n *products and* $n-1$ *additions:*

$$f(t) = x_1 + t(x_2 + t(x_3 + \ldots)).$$

  - $O(n)$ *work to determine new coefficients for differentiation and integration.*

Conditioning of Interpolation

- Conditioning of interpolation matrix $\boldsymbol{A}$ depends on basis functions and coordinates $t_1, \dots, t_m$:
  - *t_i defines the ith row, so columns tend to be nearly linearly-dependent if $t_i \approx t_{i+1}$*
  - *ϕ_j defines the jth column, so rows tend to be nearly linearly-dependent if ϕ_j is nearly in the span of the other basis functions: $span(\{\phi_i\}_{i=1,i\neq j}^{n})$*
- The Vandermonde matrix tends to be ill-conditioned:
  - *Monomials of increasing degree increasingly resemble one-another, so rows of $\boldsymbol{A}$ become nearly the same, and consequently $\kappa(\boldsymbol{A})$ grows.*
  - *The conditioning can be improved somewhat by shifting and scaling points so that each $t_i \in [-1, 1]$.*
  - *Consequently, we will consider alternative polynomial bases, seeking to improve the efficiency and conditioning associated with the Vandermonde matrix.*
  - *However, generally, we will obtain the same polynomial interpolant. To improve interpolant quality (e.g., avoid oscillations near endpoints 'Runge phenomenon'), the nodes and not the basis functions need to be changed.*

Lagrange Basis

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

$$\phi_j(t) = \underbrace{\prod_{k=1,k\neq j}^{n} (t - t_k)}_{\textbf{num}} \Big/ \underbrace{\prod_{k=1,k\neq j}^{n} (t_j - t_k)}_{\textbf{den}}$$

  - *Note that **den** is never 0,*
  - ***num** is 0 whenever $t = t_k$ for some k, so $\phi_j(t_i) = 0$ if $i \neq j$,*
  - *when $t = t_j$ then **num** and **den** are the same, so $\phi_j(t_j) = 1$,*
  - *consequently, the Lagrange Vandermonde matrix $\boldsymbol{V}(\boldsymbol{t}, \{\phi_j\}_{j=1}^{n}) = \boldsymbol{I}$.*
- Lagrange polynomials yield an ideal Vandermonde system, but the basis functions are hard to evaluate and do calculus on:
  - *Evaluation requires $O(n^2)$ work naively and may incur cancellation error.*
  - *Differentiation and integration are also harder than with monomials.*

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:
  - *Evaluation with Newton basis can use recurrence,*
    $$\phi_j(t) = \phi_{j-1}(t)(t - t_j).$$
  - *Divided difference recurrence enables fast computation of coefficients.*
- The Newton basis yields a triangular Vandermonde system:
  - *Note that $a_{ij} = \phi_j(t_i) = 0$ for all $i < j$, so $\boldsymbol{A}$ is lower-triangular.*
  - *Given $\boldsymbol{A}$, can use back-substitution to obtain the solution in $O(n^2)$ work.*
  - *Can use evaluation recurrence to compute $\boldsymbol{A}$ with $O(n^2)$ work, but divided difference recurrence is more stable than forming $\boldsymbol{A}$.*

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*:
  - *To compute overlap between basis functions, use a w-weighted integral as inner product,*
    $$\langle p, q\rangle_w = \int_{-\infty}^{\infty} p(t)q(t)w(t)dt.$$
  - $\{\phi_i\}_{i=1}^n$ *are orthonormal with respect to the above inner product if*
    $$\langle \phi_i, \phi_j\rangle_w = \delta_{ij} = \begin{cases} 1 & \textit{if } i = j \\ 0 & \textit{otherwise} \end{cases}.$$
  - *The corresponding norm is given by* $||f|| = \sqrt{\langle f, f\rangle_w}$.

Legendre Polynomials

- The Gram-Schmidt orthogonalization procedure can be used to obtain an orthonormal basis with the same span as any given arbitrary basis:
  Given orthonormal functions $\{\hat{\phi}_i\}_{i=1}^{k-1}$ *obtain* k*th function from* ϕ_k *via*

$$\hat{\phi}_k(t) = \frac{\psi_k(t)}{||\psi_k||}, \quad \psi_k(t) = \phi_k(t) - \sum_{i=1}^{k-1} \langle \phi_k(t), \hat{\phi}_i(t) \rangle_w \hat{\phi}_i(t)$$

- The *Legendre polynomials* are obtained by Gram-Schmidt on the monomial basis, with $w(t) = \begin{cases} 1 : -1 \leq t \leq 1 \\ 0 : \text{otherwise} \end{cases}$ and normalized so $\hat{\phi}_i(1) = 1$.
  For example, $\{\hat{\phi}_i(t)\}_{i=1}^{3} = \{1, t, (3t^2 - 1)/2\}$ *since*

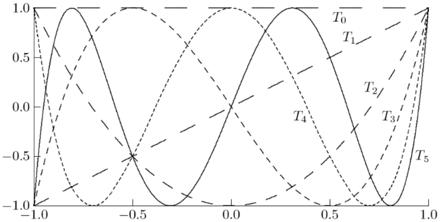
$$\psi_1(t) = 1, \quad \psi_2(t) = t \quad (\textit{as } \langle \phi_2(t), \hat{\phi}_1(t) \rangle_w / \|\hat{\phi}_1(t)\|^2 = 0)$$

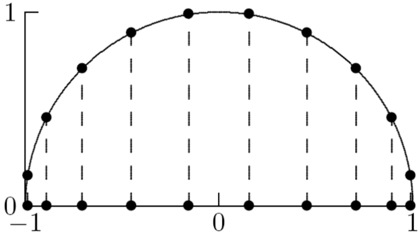
$$\psi_3(t) = t^2 - \frac{1}{2}\int_{-1}^{1} t^2 dt - t\frac{2}{3}\int_{-1}^{1} t^3 dt = t^2 - 1/3$$

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:
  - *They satisfy the recurrence* $\phi_1(t) = 1, \phi_2(t) = t, \phi_{i+1}(t) = 2t\phi_i(t) - \phi_{i-1}(t)$
  - *The Chebyshev basis functions are orthonormal with respect to*
  $$w(t) = \begin{cases} 1/(1-t^2)^{1/2} & : -1 < t < 1 \\ 0 & : \textit{otherwise} \end{cases}.$$
  - *The Chebyshev nodes ensure orthogonality of the columns of* $\boldsymbol{A}$, *since*
  $$\begin{aligned}\sum_{k=1}^{n} \phi_l(t_k)\phi_j(t_k) &= \sum_{k=1}^{n} \cos\left(\frac{(l-1)(2k-1)}{2n}\pi\right)\cos\left(\frac{(j-1)(2k-1)}{2n}\pi\right) \\ &= \frac{1}{2}\sum_{k=1}^{n}\left[\cos\left(\frac{(l-j)(2k-1)}{2n}\pi\right) + \cos\left(\frac{(j+l-2)(2k-1)}{2n}\pi\right)\right]\end{aligned}$$
  is zero whenever $j \neq l$ *due to periodicity of the summands (can be checked by evaluating geometric sums after applying Euler's formula).*

Chebyshev Nodes Intuition





- 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+1} and are abscissas of points uniformly spaced on the unit circle.

Chebyshev Basis: Why Polynomial?

- Why is $\phi_j(t) = \cos((j-1)\arccos(t))$ a polynomial?
  - *We have that*
    $$\phi_j(\cos(t)) = \cos((j-1)\cos(t))$$
  - *Further, multiple angle-formulae give that* $\cos((j-1)t) = p(\cos(t))$ *where* p *is a degree* j *polynomial (the form may be derived from De Moivdre's formula, but is complicated and not important here)*
  - *Hence, we have that* ϕ_j *is a polynomial in the domain* $[-1, 1]$
  - *The Chebyshev recurrence follows from the identity*
    $$\cos(nt) = 2\cos t \cos((n-1)t) - \cos((n-2)t)$$

Error in Interpolation

Given degree n polynomial interpolant $\tilde{f}$ of f the error $E(t) = f(t) - \tilde{f}(t)$ has n zeros $t_1, \ldots, t_n$. By induction on n, we show that there exist $y_1, \ldots, y_n \in [t_1, t_n]$ so

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

$$E(t) = E(t_1) + \int_{t_1}^{t} E'(w_0) dw_0 \qquad (2)$$

Now note that for each of $n-1$ consecutive pairs t_i, t_{i+1} we have

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

and so there are $n-1$ zeros $z_i \in (t_i, t_{i+1})$ such that $E'(z_i) = 0$.
The inductive hypothesis on E' then gives

$$E'(w_0) = \int_{z_1}^{w_0} \int_{y_2}^{w_1} \cdots \int_{y_n}^{w_{n-1}} f^{(n+1)}(w_n) dw_n \cdots dw_1 \qquad (3)$$

Substituting (3) *into* (2), *we obtain* (1) *with $y_1 = z_1$.*

Interpolation Error Bounds

- Consequently, polynomial interpolation satisfies the following error bound:

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

  Note that the Choice of Chebyshev nodes decreases this error bound at the extrema, equalizing it with nodes that are in the middle of the interval.

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

$$|E(t)| \leq \frac{\max_{s\in[t_1,t_n]} |f^{(n+1)}(s)|}{n!} h^n = O(h^n) \quad \textit{for} \quad t \in [t_1, t_n]$$

  Suggests that higher-accuracy can be achieved by
  - *adding more nodes (however, high polynomial degree can lead to unwanted oscillations)*
  - *shrinking interpolation interval (suggests piecewise interpolation)*

Piecewise Polynomial Interpolation

- The kth piece of the interpolant is typically chosen as polynomial on $[t_i, t_{i+1}]$
  - *Typically low-degree polynomial pieces used, e.g. cubic.*
  - *Degree of piecewise polynomial is the degree of its pieces.*
  - *Continuity is automatic, differentiability can be enforced by ensuring derivative of pieces is equal at knots (nodes at which pieces meet).*

$$f(t) = \begin{cases} t \in [t_1, t_2] & : f_1(t) \\ & \vdots \\ t \in [t_{n-1}, t_n] & : f_{n-1}(t) \end{cases}, \forall i \in [2, n-1], f_{i-1}(t_i) = f_i(t_i) = y_i$$

- *Hermite* interpolation ensures consecutive interpolant pieces have same derivative at each *knot* t_i:
  - *Hermite interpolation ensures differentiability of the interpolant* $\forall i \in [2, n-1], f'_{i-1}(t_i) = f'_i(t_i)$
  - *Various further constraints can be placed on the interpolant if its degree is at least 3, since otherwise the system is underdetermined.*

Spline Interpolation

- A *spline* is a $(k-1)$-time differentiable piecewise polynomial of degree k:
  Cubic splines are twice-differentiable (Hermite cubics may only be once-differentiable)
  - $2(n-1)$ *equations needed to interpolate data*
  - $n-2$ *to ensure continuity of derivative*
  - $n-2$ *to ensure continuity of second derivative for cubic splines*

  Overall there are $4(n-1)$ *coefficients in the interpolant.*
- The resulting interpolant coefficients are again determined by an appropriate *generalized Vandermonde system*:
  A natural spline obtains $4(n-1)$ *constraints by forcing* $f''(t_1) = f''(t_n) = 0$.
  Given cubic pieces $p(t)$ *and* $q(t)$ *and nodes* t_1, t_2, t_3 *(where* t_2 *is a knot) the generalized Vandermonde system for a two-piece cubic natural spline consists of 8 equations with 8 unknowns:*

$$
\begin{array}{cccc}
 & p(t_1) = y_1, & p''(t_1) = 0 & \\
p(t_2) = y_2, & q(t_2) = y_2, & p'(t_2) = q'(t_2), & p''(t_2) = q''(t_2) \\
 & q(t_3) = y_3, & q''(t_3) = 0 &
\end{array}
$$

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:

$$v_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i}, \qquad \phi_i^0(t) = \begin{cases} 1 & t_i \le t \le t_{i+1} \\ 0 & \textit{otherwise} \end{cases}$$

$$\phi_i^k(t) = v_i^k(t)\phi_i^{k-1}(t) + (1 - v_{i+1}^k(t))\phi_{i+1}^{k-1}(t), \quad f(t) = \sum_{i=1}^{n} c_i \phi_i^k(t)$$

- 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 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^k*s.*