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:

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

Polynomial Interpolation

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

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

Conditioning of Interpolation

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

- The Vandermonde matrix tends to be ill-conditioned:

Lagrange Basis

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

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

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:
- The Newton basis yields a triangular Vandermonde system:

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*:

Legendre Polynomials

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

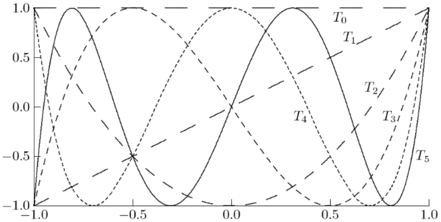
- The *Legendre polynomials* are obtained by Gram-Schmidt on the monomial basis, with $w(t) = \begin{cases} 1 : -1 \leq t \leq 1 \\ 0 : \text{otherwise} \end{cases}$ and normalized so $\hat{\phi}_i(1) = 1$.

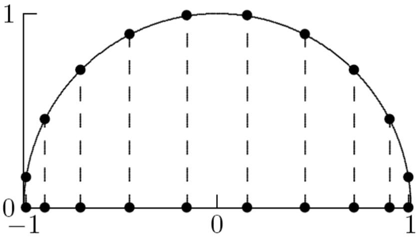
Chebyshev Basis

- *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* $t_1, \ldots, t_n$ along with a basis, to yield perfect conditioning:

Demo: *Jump with Chebyshev Nodes*

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?

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

Interpolation Error Bounds

- Consequently, polynomial interpolation satisfies the following error bound:

- 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: