Lecture 6: 1d Interpolation

Assume we want to interpolate a function $f(x)$ by an N^{th} degree interpolant I_N s.t.

$$I_N\left(f(x_i)\right) = f(x_i) \quad \text{where } x_i \text{ are } i=0,\dots,N$$

distinct points where we sample the function. Here, we have chosen to match the function exactly at all x_i points. Between points x_i we write:

$$(6.1) \quad f(x) = \sum_{i=0}^{N} \psi_i(x)\,\tilde{f}_i + e_N(x) \quad \text{where } e_N(x) \text{ is}$$

the error incurred by approximating $f(x)$ with only N terms in the series. We can use two types of interpolation: Modal and Nodal.

Modal Interpolation

For the domain $x \in [-1,+1]$ w/o periodicity, we obtain the SLO

$$\frac{d}{dx}\left[(1-x^2)\frac{d\varphi}{dx}\right] + \lambda\varphi = 0 \quad \text{where } p(x) = 1-x^2$$
$$\& \quad w(x) = 1$$

which admits solutions of the type:

$$\varphi(x) = \sum_{n=0}^{\infty} a_n x^n$$

& using Gram-Schmidt, we can derive the Legendre polynomials (see ch. 18.5.2) where they define an O.G. set as follows:

$$\int_{-1}^{+1} \omega(x)\, P_i(x)\, P_j(x)\, dx = \delta_{ij} \quad \forall\ (i,j) = 0, \ldots, N$$

when $\omega(x) = 1$ it $P_i(x)$ have been orthonormalized.

How Modal Functions Work

Note that the first 3 Legendre polynomials are $P_0(x) = 1$, $P_1(x) = x$, & $P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}$ (unnormalized).

If we wish to approximate the function $f(x) = a + bx + cx^2$ then we would find:

$$f(x) \equiv \sum_{j=0}^{N} P_j(x)\, \tilde{f}_j = P_0(x)\,\tilde{f}_0 + P_1(x)\,\tilde{f}_1 + P_2(x)\,\tilde{f}_2$$

$$= \tilde{f}_0 + \tilde{f}_1 x + \tilde{f}_2\left(\tfrac{3}{2}x^2 - \tfrac{1}{2}\right)$$

$$= \left(\tilde{f}_0 - \tfrac{1}{2}\tilde{f}_2\right) + \tilde{f}_1 x + \tilde{f}_2\left(\tfrac{3}{2}x^2\right)$$

$\therefore\quad a = \tilde{f}_0 - \frac{1}{2}\tilde{f}_2, \quad b = \tilde{f}_1, \quad c = \frac{3}{2}\tilde{f}_2$

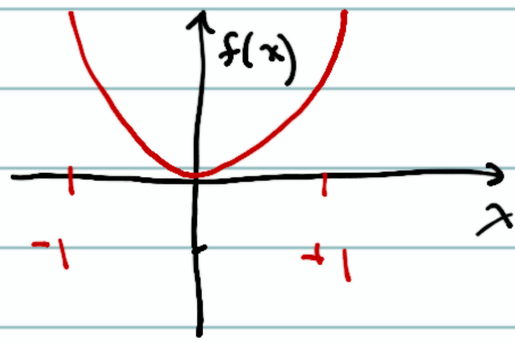
resulting in:

$\tilde{f}_0 = a + \frac{1}{3}c$

$\tilde{f}_1 = b$

$\tilde{f}_2 = \frac{2}{3}c$ & we obtain the modal approximation:

$\boxed{\text{Ex 6.1}}$ Use the $N=2$ Legendre polynomials to approximate $f(x) = x^2$ in $x \in [-1, +1]$



To find $f(x)$ for any x we need to sum all N terms in the expansion:

$$f(x) = P_0(x)\tilde{f}_0 + P_1(x)\tilde{f}_1 + P_2(x)\tilde{f}_2$$

e.g. Let $x_0 = -1$, $x_1 = 0$, & $x_2 = +1$

we can write:

$$f(x_0) = P_0(x_0)\tilde{f}_0 + P_1(x_0)\tilde{f}_1 + P_2(x_0)\tilde{f}_2 \equiv 1$$
$$f(x_1) = P_0(x_1)\tilde{f}_0 + P_1(x_1)\tilde{f}_1 + P_2(x_1)\tilde{f}_2 \equiv 0$$
$$f(x_2) = P_0(x_2)\tilde{f}_0 + P_1(x_2)\tilde{f}_2 + P_2(x_2)\tilde{f}_2 \equiv 1$$

or:

$$\begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix} = \begin{pmatrix} P_0(x_0) & P_1(x_0) & P_2(x_0) \\ P_0(x_1) & P_1(x_1) & P_2(x_1) \\ P_0(x_2) & P_1(x_2) & P_2(x_2) \end{pmatrix} \begin{pmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \end{pmatrix}$$

which can be generalized for x_i & $P_j(x_i)$ $i,j = 0,\dots,N$

Vandermonde Matrix

In R^n we can write the Legendre interpolation as follows:

$$\begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{pmatrix} = \underbrace{\begin{pmatrix} P_0(x_0) & P_1(x_0) & \cdots & P_N(x_0) \\ P_0(x_1) & P_1(x_1) & \cdots & P_N(x_1) \\ \vdots & & & \\ P_0(x_N) & P_1(x_N) & \cdots & P_N(x_N) \end{pmatrix}}_{\text{Vandermonde Matrix}} \begin{pmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \vdots \\ \tilde{f}_N \end{pmatrix}$$

which we write as:

$$f_i = V_{ij} \tilde{f}_j \quad \rightarrow \text{Legendre Transform map}$$

Note that V provides the map b/w modal & nodal space since given $\tilde{f}_i$ we now know f_i. And vice-versa, given f_i we know $\tilde{f}_i$ from:

$$\underbrace{\tilde{f}_i = V^{-1}_{ij} f_j}_{\text{Inverse Legendre transform map.}} \quad \text{for } \det(V) \neq 0$$

Nodal Interpolation

Let us write:

$$f(x) = \sum_{j=0}^{N} L_j(x)\, f_j \quad \text{when} \quad f_j = f(x_j) \rightarrow \text{i.e.,}$$

f_j are the values of f at x_j

Therefore we now insist that the interpolant match the function exactly at certain locations x_j, $j = 0, \ldots, N$. This then means that $L_j(x)$ are cardinal functions w/ the following property:

$$L_j(x_i) \equiv L_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \equiv \delta_{ij}$$

w/ the Partition of Unity property: $\sum_{j=0}^{N} L_j(x) = 1 \quad \forall\, x \in [-1, +1]$

How to Derive Lagrange Polynomial

If $\varphi_i(x)$ defines the O.G. polynomial which are the solutions to the SLO (i.e., $\varphi_i(x)$ are the eigenfunctions & natural basis on this domain) then we can use Lagrange polynomials to write:

$$(6.2) \qquad \varphi_i(x) = \sum_{j=0}^{N} \varphi_i(x_j)\, L_j(x) \quad \text{where}$$

x_j are any set of points when $\varphi_i(x)$ have been

sampled & used to define $L_j(x)$.
In fact, note that $\varphi_i(x_j) = V_{ij}$ is in fact the generalized Vandermonde matrix & so we can rewrite (6.2) as follows:

$$(6.3) \qquad V_{ij} L_j(x) = \varphi_i(x)$$

If V_{ij} is non-singular then we can left-multiply by V^{-1} to get:

$$(6.4) \qquad L_i(x) = V_{ij}^{-1} \varphi_j(x)$$

$\rightarrow$ This is the general that we could use in multi-dimensions.

Lagrange Polynomials in 1D In 1d $\exists$ a simpler form given by:

$$(6.5) \qquad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \left(\frac{x - x_j}{x_i - x_j} \right) = \frac{(x-x_0)(x-x_1)\dots(x-x_N)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_N)}$$

where if $x = x_i$, $L_i(x_i) = 1$

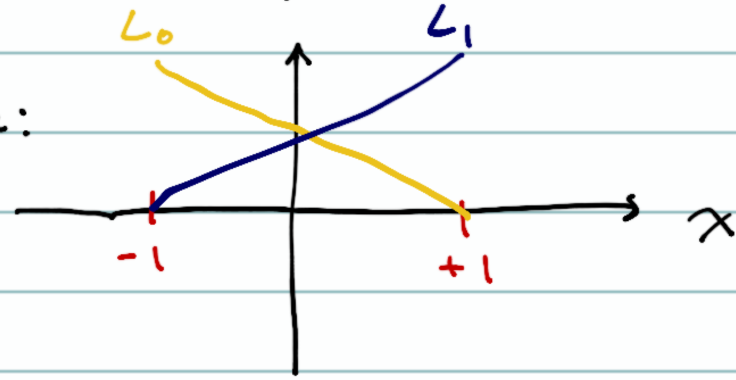
$\boxed{\text{Ex 6.2}}$ For $N=1$ we get $x_0 = -1$ & $x_1 = +1$ & so:

$$L_i(x) = \frac{(x-x_0)(x-x_1)}{(x_i-x_0)(x_i-x_1)} \qquad j \neq i$$

$$\therefore \quad L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)} = \frac{x - 1}{-1 - 1} = \tfrac{1}{2}(1 - x)$$

$$L_1(x) = \frac{(x - x_0)}{(x_1 - x_0)} = \frac{x + 1}{2} = \tfrac{1}{2}(1 + x)$$

which look like:



Note that $L_0(x_0) = \tfrac{1}{2}(1+1) = 1$ & $L_0(x_1) = \tfrac{1}{2}(1-1) = 0$

& $L_1(x_0) = 0$ & $L_1(x_1) = \tfrac{1}{2}(2) = 1$

Cardinality

&

$$\sum_{i=0}^{N} L_i(x) \equiv L_0(x) + L_1(x) = \tfrac{1}{2}(1-x) + \tfrac{1}{2}(1+x)$$
$$= 1 \rightarrow \text{partition of unity}$$

$\boxed{\text{HW 1}}$ Construct $L_i(x)$ for $i = 0, .., 2$ w/ $x_0 = -1$, $x_1 = 0$, & $x_2 = +1$. Plot them & show Cardinality & Partition of Unity.

Constructing Good Interpolants

In 1D, we construct Lagrange polynomials using:

$$L_i = \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j}$$ when we need to know

a priori the points (called roots) x_i $i=0,\dots,N$.
So far we have used equi-spaced points (e.g., $x_0=-1$, $x_1=0$, $x_2=+1$)

But continuing in this way for large N is not a good idea, as we shall see. Before describing the issue & the solution, let's introduce the Lebesgue function.

Lebesgue Function For the Lagrange polynomials L_i the Lebesgue function:

$$\Lambda_N(x) = \sum_{i=0}^{N} |L_i(x)|$$

& the

Lebesgue constant is: $\Lambda_N \equiv \max\left(\Lambda_N(x)\right) = \max\left(\sum_{i=0}^{N} |L_i(x)|\right)$

Constructing Good Interpolation Points

To construct good interpolation points, let's look at the roots of O.G. polynomials.

$\boxed{\text{Chebyshev Points}}$ The roots of these polynomials are obtained in closed-form solution as follows:

$$x_i = \cos\left(\frac{2i+1}{2N+2}\pi\right) \quad \text{for } i=0,\dots,N$$

These points are optimal for interpolation but do not include the endpoints $x=\pm 1$

Legendre Points The roots of these polynomials are obtained via Newton's method as such:

$$\varphi_N\left(x^{(n+1)}\right) = \varphi_N\left(x^{(n)}\right) + \left(x^{(n+1)} - x^{(n)}\right)\frac{d\varphi_N}{dx}(x) \equiv 0$$

or

$$x^{(n+1)} = x^{(n)} - \frac{\varphi_N\left(x^{(n)}\right)}{\varphi_N'\left(x^{(n)}\right)}$$

which requires a good starting value $x^{(0)}$ = Chebyshev points.

These points are very good but do not include the endpoints $x = \pm 1$.

Lobatto Points If we require the endpoints, then we require a different set of points.

Simply writing: $P_N^m(x) = (1+x)(1-x)P_{N-2}(x)$ will certainly now include the endpoints but the polynomials are no longer O.G.

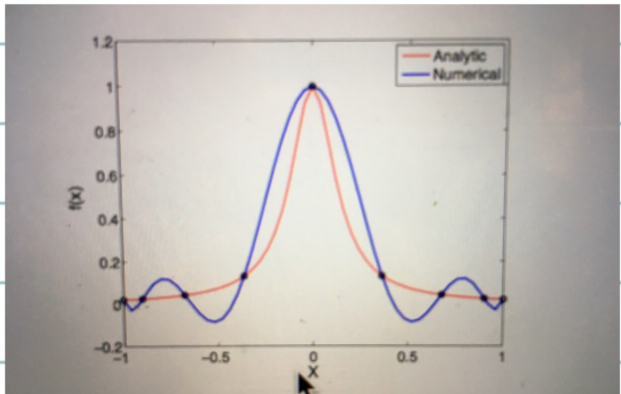
Instead, we can define:

$$L_N(x) = (1-x^2)\,P_{N-1}'(x) \rightarrow \text{Lobatto polynomials}$$

are O.G. s.t. $\int_{-1}^{+1} L_i(x)L_j(x)\,dx = c\,\delta_{ij}$

& also include the endpoints.

The zeros of $L_N(x)$ are endpoints + roots of $P'_{N-1}(x)$
The zeros/roots of $P'_{N-1}(x)$ is when the extrema of $P_N(x)$ occur. Therefore, using these points to construct the Lagrange polynomials ensure good behavior (no wild oscillations, see Fig 3.4 in text)



(b) Runge Function