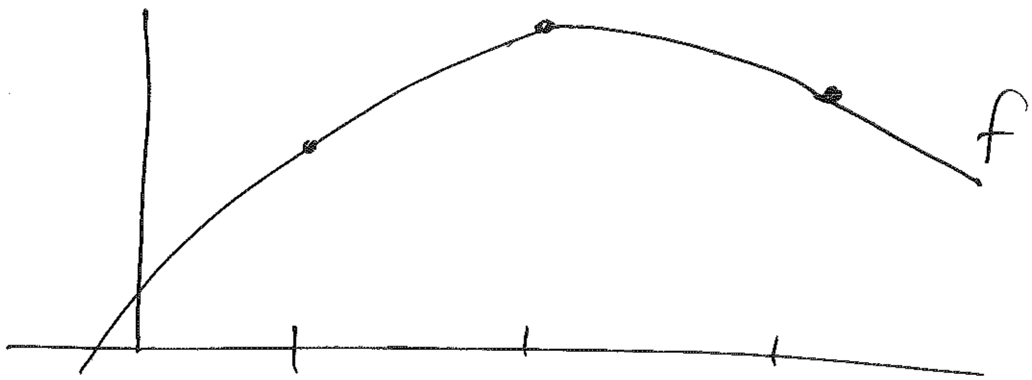
Interpolation



f <u>interpolant</u> or interpolating function.

Examples we have seen:

- <u>secant method</u> for nonlinear equations
- <u>successive parabolic interpolation</u> for minimization

<u>Problem</u>: Given data

$$(t_i, y_i) \qquad i = 1, \dots, m$$

$t_1 < t_2 < \dots < t_m$, find $f : \mathbb{R} \to \mathbb{R}$ such that

$$f(t_i) = y_i, \quad i = 1, \dots, m$$

· Note: could also impose other conditions: on derivatives, smoothness, convexity, etc.

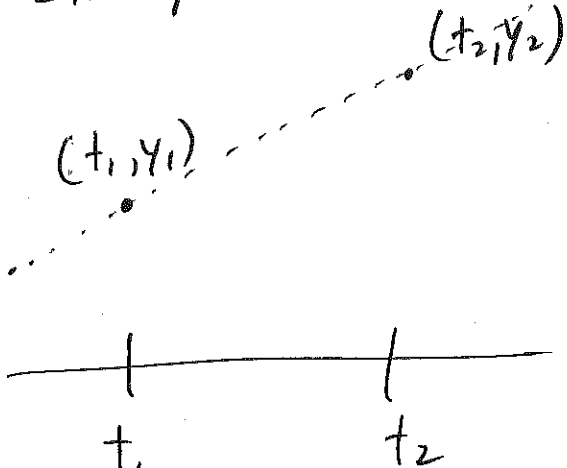
Purposes:

- plot smooth curve through data points
- read between lines of a table
- differentiate or integrate tabular data
- quick function evaluation
- replace complicated function by simpler one

Historical: compute approximate values for functions ~~for~~ from tables of data.

tool in approximating infinite-dimensional problems by finite-dimensional problems.

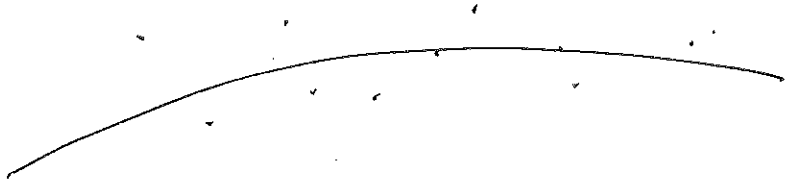
Example



$\Longleftrightarrow$ $f(t) = mt + b$

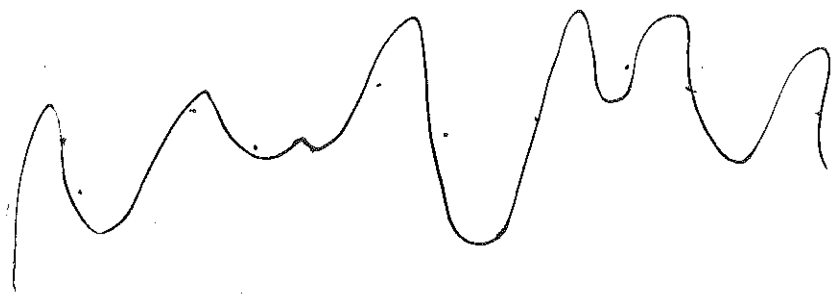
equivalent : f more useful as we can immediately evaluate at other points t, see slope, y-intercept.

Note: interpolation not always appropriate.



may want to smooth out noise

or



Many different function may interpolate same data.

<u>Examples</u>

- polynomials
- piecewise polynomials
- trigonometric functions
- exponential functions
- rational functions.

7.2. Existence, Uniqueness & Conditioning

match # of parameters to number of data pts.

too few params $\rightsquigarrow$ no interpolant

too many params $\rightsquigarrow$ not unique

$(t_i, y_i) \quad i = 1, \ldots, m$

interpolant is chosen from set of basis functions $\phi_1(t), \ldots, \phi_n(t)$

$$f(t) = \sum_{j=1}^{n} \underline{x_j}\, \phi_j(t)$$

parameters x_j are TBD

$$f(t_1) = \sum_{j=1}^{n} x_j\, \phi_j(t_1)$$

$$f(t_2) = \sum_{j=1}^{n} x_j\, \phi_j(t_2)$$

$$\vdots$$

$$f(t_m) = \sum_{j=1}^{n} x_j\, \phi_j(t_m)$$

$$\underbrace{\begin{pmatrix} \phi_1(t_1) & \phi_2(t_1) & \cdots & \phi_n(t_1) \\ \phi_1(t_2) & \phi_2(t_2) & \cdots & \phi_n(t_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(t_m) & \phi_2(t_m) & \cdots & \phi_n(t_m) \end{pmatrix}}_{m \times n} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}}_{n \times 1} = \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{m \times 1}$$

$Ax = b$

$a_{ij} = \phi_j(t_i)$

Cases:

- $\underline{m = n}$ ~~and~~ $\Rightarrow$ A square
  and A nonsingular
  unique solution for $x_1, \ldots, x_m$

- $\underline{m > n}$ overdetermined
  may not be exact solution
  e.g. Least Squares solution instead.

- $\underline{m < n}$ underdetermined
  impose additional conditions
  - monotonicity, convexity, smoothness, (examples).

Sensitivity of $\vec{x}$ to data depends on conditioning of A.

~~ex~~ different choice of basis functions even for same family of functions.

§7.3 Polynomial Interpolation

$\mathbb{P}_k$ vector space of polynomials of degree at most k ($k+1$ dimensional)

choice of basis functions affects
- cost of computing
- cost of evaluating
- sensitivity of parameters.

7.3.1 Monomial Basis

$$\phi_j(t) = t^{j-1} \qquad j = 1, \dots, n$$

i.e. $1, t, t^2, t^3, \dots, t^{n-1}$

$$p_{n-1}(t) = x_0 + x_1 t + x_2 t^2 + \dots + x_{n-1} t^{n-1}$$

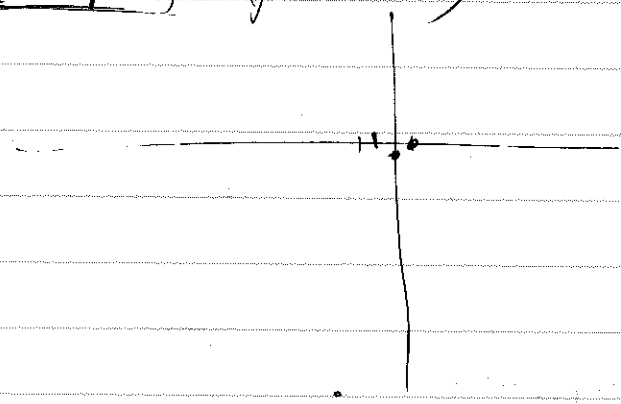
$$Ax = b$$

$$\underbrace{\begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & & & & \\ 1 & t_n & t_n^2 & \cdots & t_n^{n-1} \end{pmatrix}}_{\text{Vandermonde matrix}} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Necessarily non-singular if t_i are distinct.

Proof: $Az = 0 \Rightarrow n$ roots of degree $n-1$ polynomial $\Rightarrow \underbrace{\text{zero polynomial}}_{z=0}$

$\boxed{\text{Example}}$ (parabola)



$(-2,-27)$, $(0,-1)$, $(1,0)$

$$\cancel{p_2(t) = x_0 + x_1 t + x_2 t^2}$$

$$f(x) = a_0 + a_1 x + a_2 x^2$$

$$\begin{aligned} f(-2) &= a_0 + a_1 \cancel{x}(-2) + a_2\, 4 &&= -27 \\ f(0) &= a_0 + a_1 \cdot 0 + a_2\, 0 &&= -1 \\ f(1) &= a_0 + a_1 \cdot 1 + a_2 \cdot 1 &&= 0 \end{aligned}$$

$$\Rightarrow \begin{pmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -27 \\ -1 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 5 \\ -4 \end{pmatrix}$$

$$f(x) = -1 + 5x - 4x^2$$

<u>check</u>:

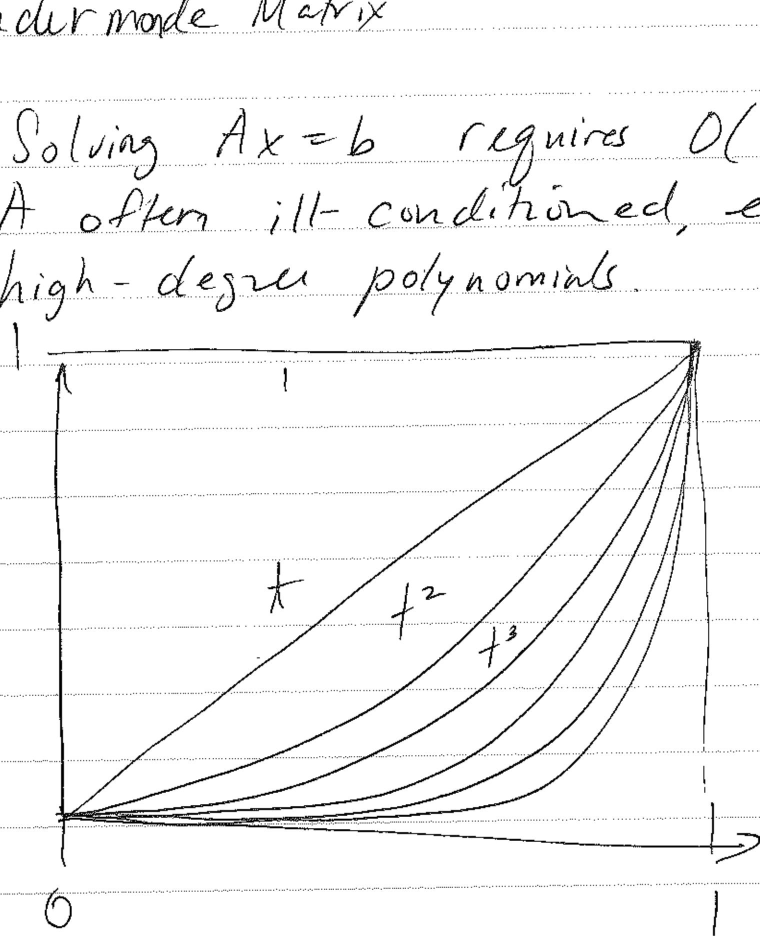
$f(-2) = -1 - 10 - 4 \cdot 4 = -27$ ✓

$f(0) = -1$ ✓

$f(1) = -1 + 5 - 4 = 0$ ✓

Vandermonde Matrix

- Solving $Ax=b$ requires $O(n^3)$ work
- A often ill-conditioned, especially for high-degree polynomials.



monomial basis functions become progressively less distinguishable, leading to columns in A that become progressively less distinguishable

for most choices of t_i, $K(A)$ grows at least exponentially w/ n.

A nonsingular in theory, but arbitrarily ill-conditioned.

- polynomial will still fit data points well, as G.E. w/ partial pivoting produces small residual.
- but values of coefficients will be poorly determined.

+ both conditioning and amount of work can be improved using a different basis interpolating polynomial is unique.

Proof:

Assume $p_1(x)$ degree $n-1$

$p_2(x)$ ~~with~~

are both interpolating (x_i, y_i)

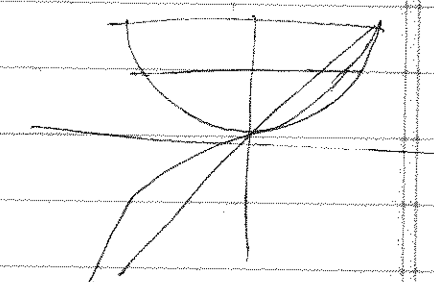
$p_2(x_i) = p_1(x_i) = y_i \quad , \quad i = 1, \ldots, n$

Then $\underbrace{p(x_i)}_{\text{degree} \le n-1} = \underbrace{p_1(x_i) - p_2(x_i)}_{\text{degree } n-1} = 0$ for $i = 1, \ldots, n$

w/ n roots $\Rightarrow$ 0 polynomial

Improved monomial basis, shifted and scaled,

$$\phi_j(t) = \left(\frac{t-c}{d}\right)^{j-1}$$



$$c = \frac{t_1 + t_n}{2} \quad , \quad d = \frac{t_n - t_1}{2}$$

Better, but still not well conditioned

Evaluation

- Efficient evaluation ~~of~~ using Horner's Method

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

$$p_n(x) = a_0 + x(a_1 + x(a_2 + x(\ldots (a_{n-1} + x a_n) \cdots)))$$

$n+1$ additions and $n+1$ multiplications

Example:

$1 - 4x + 5x^2 - 2x^3 + 3x^4 = \cancel{1 - 4(x + 5(x - 2x(x + 3x)}$

$\cancel{1 + 4 \quad 1 = x(4+}$

$1 + x(-4 + x(5 + x(-2 + 3x)))$

4 +'s and 4 x's

Same principle applies in forming Vandermonde matrix

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$a_{ij} = x_i \, a_{i,j-1}$$

Superior to explicit exponentiation

§7.3.2 Lagrange Interpolation

$(x_i, y_i) \qquad i = 1, \dots, n$

Lagrange basis functions for $\mathbb{P}_{n-1}$
(fundamental polynomials)

$$l_j(x) = \frac{\prod_{k \neq j} (x - x_k)}{\prod_{k \neq j} (x_j - x_k)}, \qquad j = 1, \dots, n$$

Note: $l_j(x)$ is a polynomial of degree $n-1$

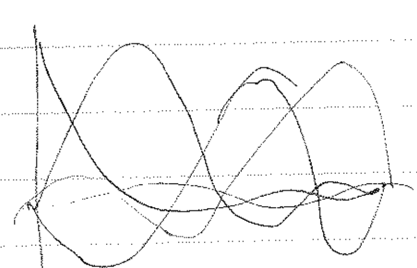
$$j = 1, \dots, n \qquad l_j(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Recall $a_{ij} = l_j(x_i) = \delta_{ij} = I$

$\Rightarrow$ coefficients are the y_i

$\cancel{p(x) = y_1}$

$$p(x) = y_1 l_1(x) + y_2 l_2(x) + \dots + y_n l_n(x)$$



+ easy to determine interpolating polynomial
+ parameters perfectly conditioned

− Lagrange form is more expensive to evaluate
− more difficult to differentiate, integrate, etc

Example Lagrange Interpolation

$(-2, -27)$, $(0, -1)$, $(1, 0)$

$$p(x) = -27 \frac{(x-0)(x-1)}{(-2-0)(-2-1)} + (-1) \frac{(x+2)(x-1)}{(0+2)(0-1)} + 0 \cdot (\quad)$$

$$\cancel{-27(}$$

$$= -27 \frac{x(x-1)}{6} + \frac{(x+2)(x-1)}{2}$$

§ 7.3.3 Newton Interpolation

data points (x_i, y_i) $i = 1, \dots, n$

basis functions $\pi_j(x) = \prod_{k=1}^{j-1} (x - x_k)$ $j = 1, \dots, n$

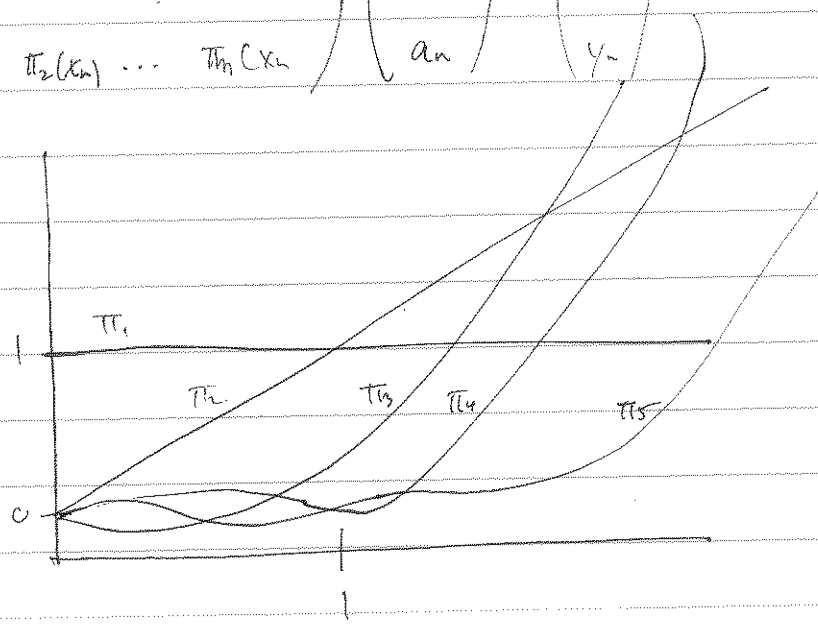
$$\pi_1(x) = 1$$
$$\pi_2(x) = (x - x_1)$$
$$\pi_3(x) = (x - x_1)(x - x_2)$$
$$\vdots$$
$$\pi_n(x) = (x - x_1)(x - x_2) \cdots (x - x_{n-1})$$

$$p(x) = a_1 \pi_1(x) + a_2 \pi_2(x) + \dots + a_n \pi_n(x)$$

$$\begin{pmatrix} \pi_1(x_1) & \pi_2(x_1) & \cdots & \pi_n(x_1) \\ \pi_1(x_2) & \pi_2(x_2) & \cdots & \pi_n(x_2) \\ \vdots & & & \\ \pi_1(x_n) & \pi_2(x_n) & \cdots & \pi_n(x_n) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Note $\pi_j(x_i) = 0$ when $i < j$ $\Rightarrow$

$$\begin{pmatrix} \pi_1(x_1) & & 0 & \\ \pi_1(x_2) & \pi_2(x_2) & & \\ \vdots & & \ddots & \\ \pi_1(x_n) & \pi_2(x_n) & \cdots & \pi_n(x_n) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$



Example $(-2,-27),\ (0,-1),\ (1,0)$

$$\pi_1(x) = 1$$

$$\pi_2(x) = x - x_1 = x+2$$

$$\pi_3(x) = (x-x_1)(x-x_2) = (x+2)(x)$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} -27 \\ -1 \\ 0 \end{pmatrix} \Rightarrow \begin{matrix} a_1 = -27 \\ a_2 = 13 \\ a_3 = -4 \end{matrix}$$

$$p(x) = -27 + 13(x+2) + -4(x+2)x$$

Evaluation: can be efficiently evaluated using Horner's rule.

$$\begin{aligned} p(x) &= a_1\pi_1(x) + a_2\pi_2(x) + \cdots + a_n\pi_n(x) \\ &= a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + \cdots + a_n(x-x_1)(x-x_2)\cdots(x-x_{n-1}) \\ &= a_1 + (x-x_1)\Big(a_2 + (x-x_2)\big(a_3 + (x-x_3)(\cdots(a_{n-1} + a_n(x-x_{n-1}))\cdots)\big)\Big) \end{aligned}$$

Newton Interpolation

incremental

$p_n(x)$ polynomial of degree $n-1$
interpolates (x_i, y_i), $i=1,\dots,n$

$p_{n+1}(x)$ polynomial of degree n
interpolates (x_i, y_i), $i=1,\dots,n$
<u>and</u> (x_{n+1}, y_{n+1})

$$p_{n+1}(x) = p_n(x) + a_{n+1}\pi_{n+1}(x)$$

(since $\pi_{n+1}(x_i) = 0 \Rightarrow p_{n+1}(x_i) = p_n(x_i) = y_i$)
$\forall\ i=1,\dots,n$

choose a_{n+1} to interpolate (x_{n+1}, y_{n+1})

$$p_{n+1}(x_{n+1}) = p_n(x_{n+1}) + a_{n+1}\pi_{n+1}(x_{n+1}) = y_{n+1}$$

$$\Rightarrow \quad a_{n+1} = \frac{y_{n+1} - p_n(x_{n+1})}{\pi_{n+1}(x_{n+1})}$$

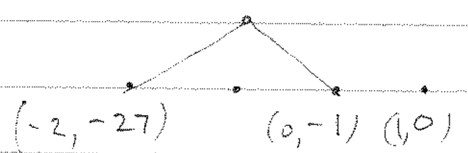
$(x_1, y_1), \ldots, (x_n, y_n)$

coefficients through divided differences

$$f[x_1, x_2, \ldots, x_k] = \frac{f[x_2, \ldots, x_k] - f[x_1, \ldots, x_{k-1}]}{x_k - x_1}$$

$$f[x_k] = y_k$$

<u>Example</u> $(-2, -27), (0, -1), (1, 0)$



7.3.4. Orthogonal Polynomials.

inner product:

$$(p,q) = \int_a^b p(t)\,q(t)\,w(t)\,dt$$

$w(t)$ non-negative weight function

p, q are orthogonal if $(p,q) = 0$

$\{p_i\}$ orthonormal if

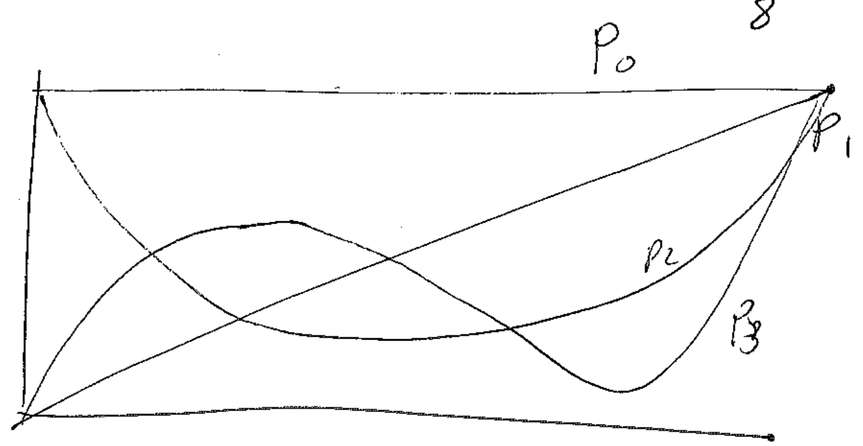
$$(p_i, p_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

Given set of polynomials, G-S orthogonalization process can be used to general orthonormal set spanning the same space.

E.g. take $w(t) = 1$, on $[-1, 1]$

apply G-S to monomials $1, t, t^2, t^3, \ldots$, and scale so that $p_k(1) = 1$, get Legendre polynomials

$$1,\ t,\ \frac{(3t^2-1)}{2},\ \frac{(5t^3-3t)}{2},\ \frac{(35t^4-30t^2+3)}{8},\ \frac{(63t^5-70t^3+15t)}{2}, \ldots$$



$1, t, t^2, t^3$ on $[-1, 1]$

first, orthonormal set:

$$\int_{-1}^{1} 1 = t\Big|_{-1}^{1} = 1 - -1 = 2 \Rightarrow p_0(t) = \frac{1}{2}$$

$$\left(\frac{1}{2}, t\right) = \int_{-1}^{1} \frac{1}{2} t \, dt = \frac{1}{2}\frac{t^2}{2}\Big|_{-1}^{1} = \frac{1}{4}\left[\frac{1}{2} - \frac{1}{2}\right] = 0$$

$$(t,t) = \int_{-1}^{1} t^2 dt = \frac{t^3}{3}\Big|_{-1}^{1} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3} \Rightarrow p_1(t) = \sqrt{\frac{3}{2}}\, t$$

$$\int_{-1}^{1} \frac{3}{2} t^2 \, dt = \frac{3}{2} \frac{t^3}{3}\Big|_{-1}^{1} = \frac{3}{2}\left[\frac{1}{3} + \frac{1}{3}\right] = \frac{\cancel{3}}{2}\frac{\cancel{2}}{\cancel{3}} = 1 \checkmark$$

Orthogonal polynomials satisfy 3-term recurrence:

$$p_{k+1}(t) = (\alpha_k t + \beta_k)\, p_k(t) - \gamma_k\, p_{k-1}(t)$$