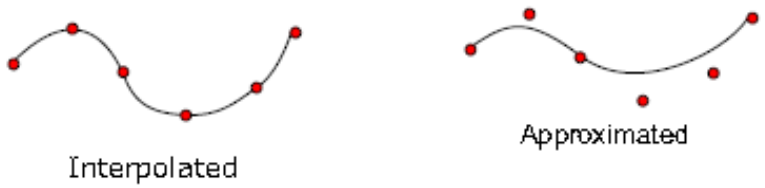
2) Interpolation and Approximation

(Grasselli & Pelinovsky - Chapter 5)

Basic Definitions: What is the difference between the two?



Given a set of data points $\{x_i, y_i\}_{i=1}^N$ and a function $f(x) = \sum_{i=1}^M a_i \phi_i(x)$ where $\{\phi\}_{i=1}^M$ are suitably chosen basis functions.

- the **interpolation problem**: find $\{a_i\}_{i=1}^M$ such that $f(x_i) = y_i, i = 1,...,M$
- the **approximation problem: find** $\{a_i\}_{i=1}^M$ such that $\sum_{k=1}^N [y_k - f(x_k)]^2 = \min$

Note that usually N>>M

Questions:
- what are good choices of the basis function ϕ_i ?
  - easy to compute
  - fast decrease of errors with $M \to \infty$
- how to determine the expansion coefficients $a_i,...,a_M$?
- Estimates of integration/approximation errors
- computational cost

Polynomials - a first natural choice of basis functions
- MATLAB convention for indexing coefficients $P_n(x) = c_0 x^n + c_1 x^{n-1} + ... + c_{n-1} x + c_n$
- P_n is determined by a coefficient vector $c \in \mathbb{R}^n$. Hence, polynomials P_n can be identified with a finite-dimensional vector space $P_n \in V = \mathbb{R}^{n+1}$
- trigonometric functions can also be integrated as (complex) polynomials
  set $z = e^{i\varphi} = \cos\varphi + i\sin\varphi$ then $z^k = e^{ik\varphi} = \cos(k\varphi) + i\sin(k\varphi)$
- roots of polynomials

Fundamental Theorem of Algebra: a polynomial of degree n has exactly n possibly multiple roots $p_n(x) = c_1(x - x_1)^{m_1}(x - x_2)^{m_2} \cdot ... \cdot (x - x_k)^{m_k}$, $x_1,..,x_k$ - distinct roots with multiplicities $m_1,..,m_k$ and $m_1 + ... + m_k = n$

Remark: Given the numbers $c_0,...,c_n \in \mathbb{R}$, consider an (nxn) **companion matrix**

$$A_n = \frac{1}{c_0}\begin{pmatrix} 0 & 0 & \dots & 0 & -c_n \\ c_0 & 0 & \cdots & 0 & -c_{n-1} \\ 0 & c_0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & c_2 \\ 0 & \cdots & 0 & c_0 & c_1 \end{pmatrix}$$

It can be shown that $P_n(\lambda) = \det(A - \lambda I)$ and the roots of $P_n(\lambda)$ are given by the eigenvalues of the companion matrix A. Solving the eigenvalue problem for the companion matrix A the preferred way of finding the roots of a polynomial (MATLAB's function roots), because

- all roots are found in the computation (both real and complex)
- it is numerically stable and efficient procedure (no issue of the initial guess etc.)

- Representation of polynomials: power series (coefficients) vs. factorised (roots + c_1)
- Differentiation and integration of polynomials is trivial

$$P_n'(x) = nc_0x^{n-1} + (n-1)c_1x^{n-2} + ... + c_n$$

$$\int P_n(x).dx = \frac{c_0x^{n+1}}{n+1} + \frac{c_1x^n}{n} + ... + c_nx + c_{n+1}$$

- Multiplication of polynomials $P_n(x) \cdot P_m(x) = P_{n+m}(x)$

$$(\sum_{i=0}^{n} a_i x^{n-i})(\sum_{j=0}^{m} b_j x^{m-j}) = \sum_{l=0}^{n+m} c_l x^{n+m-l}$$ c_l-complicated functions of $\{a_i\}$ and $\{b_i\}$

- Division of polynomials leads to **rational functions** (not polynomials anymore) $R(x) = \frac{P_n(x)}{P_m(x)}$

Weierstrass Approximation Theorem (1885)
Let f be a continuos function on [a,b] and let $\varepsilon > 0$ be arbitrary. Then, there exists a polynomial P_n such that

$$\max_{a \le x \le b} | P_n(x) - f(x) | < \varepsilon$$

Remarks:
- Central result of approximation theory
- the theorem is not consrtuctive; it's not even known what the degree of the polynomial should be

2.1 Integrating Polynomials

Given (N+1) pairs $\{(x_i, y_i)\}_{i=0}^{N}$, find a degree N polynomial passing through all these points.

$g(x) = \sum_{k=0}^{N} a_k x^k$ has (N+1) unknown coefficients which can be determined using the following conditions

$$\begin{cases} \sum_{k=0}^{N} a_k {x_0}^k = y_0 \\ \sum_{k=0}^{N} a_k {x_1}^k = y_1 \\ \vdots \\ \sum_{k=0}^{N} a_k {x_N}^k = y_N \end{cases}$$

$$\begin{pmatrix} 1 & x_0^1 & \cdots & x_{N_0} \\ 1 & x_1^1 & \cdots & x_1^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N^1 & \cdots & x_N^N \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix}$$

Vandermonde Matrix V

Vandermonde determinant: $\det(v) = \prod_{i=0}^{N} \prod_{j=i+1}^{N} (x_i - x_j)$.

Thus, for distinct integration points $x_i \neq y_i, i \neq j$, $\det(V) \neq 0$ and the Vendermonde matrix is nonsingular $\Rightarrow$ unique solutions exist.

Theorem: There exists a unique interpolating polynomial $P_n(x)$ iff the data points $x_o,...,x_N$ are distinct.

However, the Vandermonde matrix is extremely ill-conditioned, to the point that it is very difficult to use in practice. This is because the basis functions are monomials $x^k, k = 0,..,N$ which for large k look alike; in analogy with vectors in $\mathbb{R}^n$, they are almost collinear (linearly dependent).

Lagrange Interpolating Polynomials

Choose P_n interpolating polynomial is a special form:

$P_n(x) = \sum_{k=0}^{n} \phi_k(x) y_k$ where

(*) $\phi_k(x) = \dfrac{\prod_{i=0, i\neq k}^{n} (x - x_i)}{\prod_{i=0, i\neq k}^{n} (x_k - x_i)}$ <— Lagrange/cardinal polynomial

(the terms i=k are omitted in both the numerator and denominator)

For example (with n=3)

$$P_3(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} y_0 + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} y_1 + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_0-x_3)} y_2 + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} y_3$$

Remarks:

- the interpolating polynomial is a sum of n+1 n-degree polynomials (Lagrange functions)
- the Lagrange functions have the property $\phi_k(x_i) = \delta_{ik}, i,k = 0,...,n$,
  i.e. have unit value at the corresponding node and zero at all other nodes.

Thanks to this property, the interpolation conditions $P_n(x_i) = y_i, i = 0,...,n$ are satisfied automatically (ly construction).

- the interpolating polynomial is easy to construct (no need to solve an algebra system)
- computational cost: $\left\{ \begin{matrix} O(n) \text{ operations to evaluate} \\ \phi_i(x) \text{ for each x} \end{matrix} \right\} \cdot n = O(n^2)$

Can we do better than that?

Barycentric Interpolation Formula

Examples:
- Lagrangian polynomial
- Lagrangian inrerpolation

For all k, the numerators in (*) are the same, except that they miss different factors $(x-x_k)$. One should take advantage of that.
Define the **node polynomial** for the given grid

$$\zeta(x)=\prod_{i=0}^{n}(x-x_k)$$

Then, the Lagrange polynomials become $\phi_k(x)=\dfrac{\ell(x)}{\ell'(x_k)(x-x_k)}$.

We can also define $\lambda_k=\dfrac{1}{\prod_{i=0}^{n}(x_k-x_i)}=\dfrac{1}{\ell'(x_k)}$ so that we have $\phi_k(x)=\ell(x)\dfrac{\lambda_k}{x-x_k},k=0,...,n$.

Then, the Lagrange interpolation formula becomes

$$\underbrace{p_n(x)=\ell(x)\sum_{k=0}^{n}\frac{\lambda_k}{x-x_k}y_k}$$

The First Form of the Barycentric Interpolation formula

Remarks:
- single dependence on x in side the sum
- if the weights λ_k are known, the formula produces a value for $p(x)$ is just O(n) operations
- evaluation of the weights λ_k requires $O(n^2)$ computation, but this is independent of x and hence can be performed just once at the beginning (for special grids the weights are known analytically)
- The barycentric formula is numerically stable (w.r.t round-off errors)
- further modifications exist, e.g for chebyshev interpolation

Analysis of Errors of Polynomial Interpolation
Assume the data is obtained using the function $f(x)$ as $y_i=f(x_i),i=0,\ldots,n$
Consider the **error function**: $E(x)=f(x)-P_n(x)$

It vanishes at $x_0,x_1,...,x_n$, thus
$$\begin{aligned}E(x)&=f(x)-P_n(x)\\&=(x-x_0)(x-x_1)\cdot...\cdot(x-x_n)g(x)\end{aligned}$$
g(x) accounts for the behaviour between the nodes. Then,
$$\begin{aligned}f(x)-P_n(x)-E(x)&=f(x)-P_n(x)-(x-x_o)\cdot...\cdot(x-x_n)g(x)\\&=0\end{aligned}$$
Need to determine g(x). For this purpose we will define a near function W(t) (depending on the variable t)
$$W(t)=f(t)-P_n(x)-(t-x_o)(t-x_1)\cdot...\cdot(t-x_n)g(x)$$
W(t) has n+2 roots: $t=x_0,x_1,...,x_n$ and t=x
Assume W(t) is **continuous and differentiable**

Mean Value Thereom: There is a root of the derivative W'(t) between every two roots of W(t); thus, altogether, there are n+1 roots of the derivative

W''(t) - n roots
W'''(t) - (n-1) roots
⋮
$W^{(n+1)}(t)$ - 1 root (denoted ξ) in the interval bounded by $\{x_0, x_n, x\}$. Thus,

$$W^{(n+1)}(\xi) = 0$$

$$= \frac{d^{(n+1)}}{dt^{(n+1)}}[f(t) - P_n(t) - (t-x_0)(t-x_1)\cdot\ldots\cdot(t-x_n)]\,|_{t=\xi}$$

$$= f^{(n+1)}(\xi) - 0 - (n+1)!\,g(x)$$

$$= 0$$

$\Rightarrow g(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}$, where ξ varies between $\{x_0, x_1, x\}$. Therefore $E(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{k=0}^{n}(x - x_k)$

Remarks
- since f(x) is usually unknown, the expression for E(x) has only **qualitative** meaning
- if $f(x) = P_n(x), m \le n$, then $E(x) \equiv 0$
- smooth functions f(x) $\Rightarrow$ smaller errors
- for large n the error E(x) increases at the end joints of the interval (Runge Phenomenon); do not take n larger then 5,6,..

Another look at the Errors of Polynomial Interpolation

$$E_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\underbrace{\prod_{k=0}^{n}(x - x_k)}_{w_{n+1}(x)}$$

Assume $x \in [a,b]$

Then $x_k = a + kh,\ k = 0,\ldots,n,\ h = \frac{b-a}{h}$. Let $M_n = \underbrace{\max}_{a \le x \le b} |f^{(n)}(x)|$ - upper bound on the n-th derivative

$$c_n = \underbrace{\max}_{0 \le x \le n} \prod_{k=0}^{n} |z - k|$$

Thus: $E_{\max} = \underbrace{\max}_{a \le x \le b} |E_n(x)| \le \frac{1}{(n+1)!} c_n M_{n+1} h^{n+1}$

For a fixed degree of interpolating polynomial $E_{\max} \le ch^{n+1},\ h << 1$

How do reduce the error: $h \to 0$
- shrink the interval $|b-a| \to 0$, while keeping n unchanged
- keep the interval, but increase n then we need to control the derivatives $\lim_{n\to\infty} M_n$.

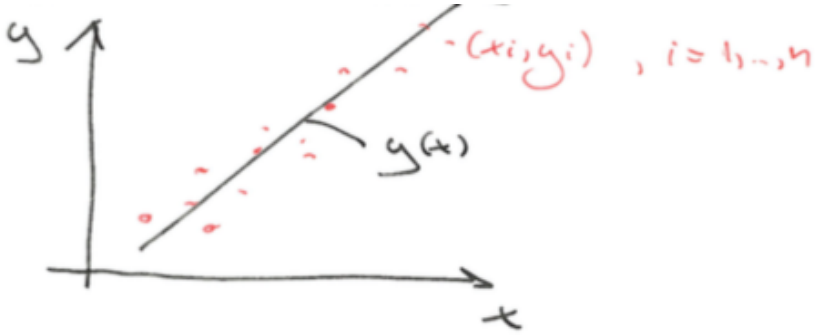
- The rate of decay (increase) of M_n depends on the analytical (regularity) properties of the interpolation function

- Runge's function $f(x) = \frac{1}{1+25x^2},\ \ x \in [-1,1]$

Least -Squares Approximation

Applicable when,
- the number of samples is much larger than the number of parameters in the approximating function

- the data is approximate only, so it is not necessary for the approximating function to go through every point



The approximating function y(x) should be chosen to minimise the deviation from the data in some suitable sense (expressing using different vector norms)

$$S_p = ||\vec{e}||_p , \; e_i = y_i - y(x_i)$$

- when p is odd $||-||_p$ is not differentiable (problem hard to solve)
- p=2 - the most common choice (least squares approximation)

Example

Given data $\{x_i, y_i\}_{i=1}^n$

Approximating function y=ax+b

$$S(a,b) = ||e||_2^2 = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

Want to find $\min_{a,b}\{S(a,b)\}$

$$\left.\begin{aligned} \frac{\partial s}{\partial a} &= \sum_{i=1}^{n} 2(y_i - ax_i - b)(-x_i) = 0 \\ \frac{\partial s}{\partial b} &= \sum_{i=1}^{n} 2(y_i - ax_i - b)(-1) = 0 \end{aligned}\right\} \Rightarrow \underbrace{\begin{cases} a\sum_i x_i^2 + b\sum_i x_i = \sum_i x_i y_i \\ a\sum_i x_i + bn = \sum_i y_i \end{cases}}_{\text{system of normal equations easily solved for (a,b)}}$$

The approach can be generalised to higher order approximating polynomials, e.g.

$y(x) = a_0 + a_1 x + \ldots + a_n x^m$ (degree m, m<<n)

Then $S(a_0, a_1, \ldots, a_m) = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - \ldots - a_m x_n^m)^2$

Conditions $\frac{\partial s}{\partial a_j} = 0, \; j = 0, \ldots, m$ give rise to the normal system

$$\underbrace{\begin{pmatrix} N & \sum x_i & \cdots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{2m} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^n y_i \end{pmatrix}}_{b}$$

Structure of the normal system

$Ax = B$

$A = V^T V$, where $V: \underbrace{\mathbb{R}^{m+1}}_{\text{degree of the polynomial}} \rightarrow \underbrace{\mathbb{R}^{n}}_{\text{number of data points}}$

V- rectangular nx(m+1) Vandermode matrix

$$V = \begin{pmatrix} 1 & x_1 & \cdots & \sum x_1^{\,m} \\ 1 & x_2 & \cdots & \sum x_2^{\,m+1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & \sum x_n^{\,m} \end{pmatrix} \text{ and } b = V^T b$$

The normal system $V^T V x = V^T b$ is thus $(m+1)\mathrm{x}(m+1)$. The problem $Vx = b$ has dimension $(m+1)\mathrm{x}n$ and is therefore **overdetermined** when n>m+1. When n=m+1, a standard interpolation is recovered.

Remarks:

- The normal system $V^T V$ of the normal system is symmetric (good!) but ill-conditioned (bad!) due to poor conditioning, m should not exceed ~10
- When the data exhibits special trends, other (non polynomial) approximating functions can be used e.g. $y(x) = ax^b$. Then

$$\begin{aligned} e_i &= \log(y_i) - \underbrace{\log(a)}_{c} + b\log(x_i) \\ &= \log(y_i) - c - b\log(x_i) \end{aligned}$$

- Poor conditioning of the normal matrix can be eliminated by using combinations of **orthogonal polynomials** as approximating functions

Orthogonal Polynomials (Chapter 17)

Consider definition of a **weighted inner product** for functions $f, g : [a,b] \rightarrow \mathbb{R}$.

The weight function: $w \in C^1(a,b), w(x) > 0 \underset{x \in (a,b)}{\forall} \int_{-1}^{1} w(x).dx < \infty$

$$(f,g)_w = \int_a^b f(x)g(x)w(x).dx$$

The functions f and g are orthogonal on (a,b) wrt the weight w(x) iff $(f,g)_w = 0$.

Consider a family of degree n polynomials $P_n(x)$ defined on [a,b]. For a given weight w(x), one can obtain a family of orthogonal polynomials $p_o, p_1, p_2, \ldots$ by performing the Gram-Schmidt orthogonalisation procedure. They satisfy the relations $(p_i, p_k)_w = 0$, k≠j.