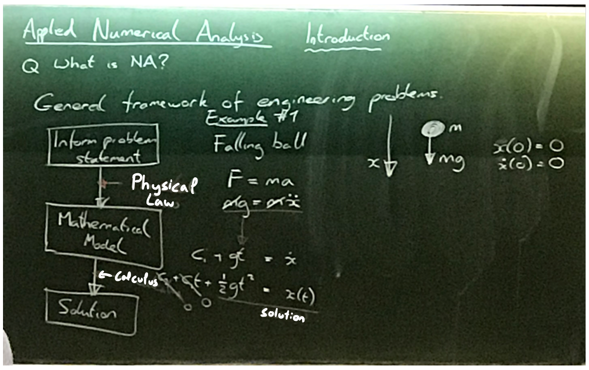
NUMERICAL ANALYSIS

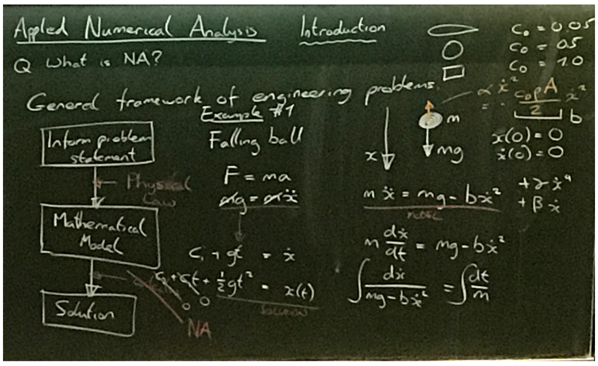
Applied Numerical Analysis

Introduction

What is Numerical Analysis?

General framework of engineering problems



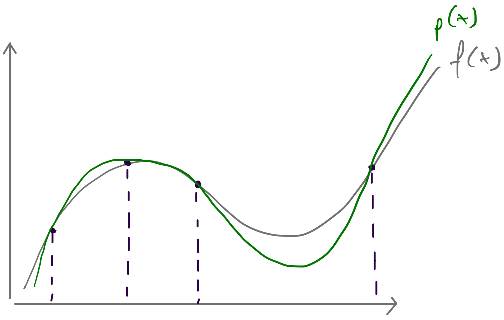


NA is a solution method (method → solution) using only operations (that a computer can perform) $+, -, \times, \div$, etc.

Example:

^ indicates that it is a numerical solution

$\dot{x} = f(x)$ we can aprox $x(\Delta t n) \simeq \hat{x}_n = \hat{x}_{n+1} + \Delta t f(\hat{x}_{n+1})$ Module 5



Modules:

Problem Statement.

such that

1:- Find x s.t. $f(x) = 0$ Root-finding.

2:- Find $p(x)$, s.t $p(x_i) = f(x_i)$ for all i. Interpolation.

3:- Find $p(x,y)$ st. $p(x_i, y_i) = f(x_i, y_i)$ $\forall i$. Multidimensional interpolation Advanced interpolation.

4.- Approximate $\hat{f}' \approx \frac{df}{dx}$ and $\hat{F} \approx \int_a^b f\,dx$ Numerical Calculus Differation + Integration.

5:- Find $u(x)$ s.t. $u' = f(u)$, $u(0) = u_0$ Numerical ODEs

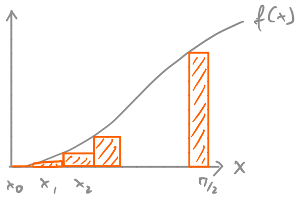
6:- Find $x \in [a,b]$ s.t. $f(x)$ is a minimum. Optimization.

Example — Statement + analysis of a numerical method.

Module 4 — $I(f) = \int_0^{\pi/2} f(x)\,dx \qquad f(x) = x\sin x$

Analytic solution $\quad I[f] = 1$

Numerically

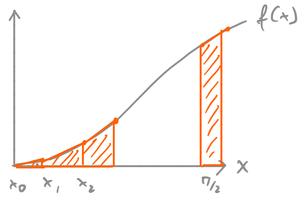


$$h = \frac{\frac{\pi}{2} - 0}{N} = \frac{\pi}{2N}$$

$$I(f) \approx \sum_{i=0}^{N-1} h f(x_i)$$

$$f_i = f(x_i)$$

Piecewise Constant Rule



$$I(f) = \sum_{i=0}^{N-1} h \, \frac{f(x_i) + f(}{2}$$

Trapezoidal Rule

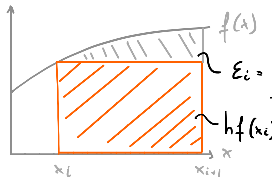
Error $\quad \varepsilon := |Q(f) - I(f)|$

Convergence rate $\quad$ Is the rate at which $\varepsilon \to 0$, as $h \to 0$ (or $h \to \infty$) Important

Eg. If $\varepsilon \sim h^{a}$ then we say the method is a-order accurate.

Consistency $\quad$ NM (Numerical Method.) is consistent if $\varepsilon \to 0$ as $h \to 0$

Conv. of P.C rule Q:



$$\varepsilon_i = \int_{x_i}^{x_{i+1}} f(x) - f_i\,dx = \int_0^h \theta f'(x_i)\,d\theta + \int_0^\theta \frac{\theta^2}{2} f''(x_i)\,d\theta + \dots$$

$$= \left[\frac{1}{2}\theta^2 f_i'\right]_0^h + \left[\frac{1}{6}\theta^3 f_i''\right]_0^h + \dots$$

$$= O(h^2) + \cancel{O(h^3) + \dots}$$

$hf(x_i)$

$$f(x) = f(x_i) + \theta f'(x_i) + \frac{\theta^2}{2} f''(x_i) + \dots$$

$$\theta = x - x_i$$

Order notation

Constants do not matter. All we are trying to prove is that $\varepsilon \propto h^{\alpha}$

Bigger order h do not matter either since h is small.

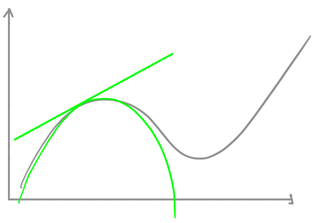
$$\varepsilon = \sum_{i=0}^{N-1} \varepsilon_i = \sum_{i=0}^{N-1} O(h^2) = O(Nh^2) = O\left(\frac{1}{h}h^2\right) = O(h)$$

$$h = \frac{b-a}{N} \qquad h \propto \frac{1}{N}$$

Taylor approximation $\boxed{\text{MAJOR}}$

Assume $f(x) \in C^{N+1}([a,b]) \to$ then $\to$

$$\to f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \sum_{j=3}^{N} \frac{1}{j!} f^{(j)}(x_0)(x - x_0)^j + O(x - x_0)^{N+1}$$



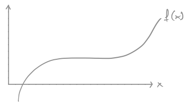
Module 1:- Root finding

<u>Problem statement</u>: Find $\tilde{x} \in \mathbb{R}$ s.t. $f(\tilde{x})=0$

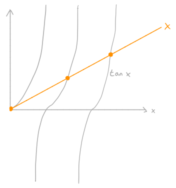
<u>Examples</u> <u>Polynomials</u>

$ax^2+bx+c=0$

$ax^5+bx^4+\dots+f=0$



<u>structural analysis</u> $\tan x = x$



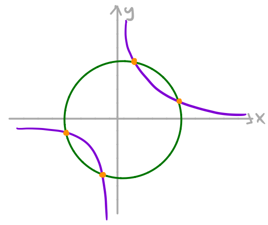
<u>More generally</u>

Find $\tilde{x} \in \mathbb{R}^M$ s.t. $f(\tilde{x})=0$ where $f: \mathbb{R}^m \longrightarrow \mathbb{R}^m$

Ex. - $Ax=b$

- $(x^2+y^2=1)$

$xy=\frac{1}{4}$

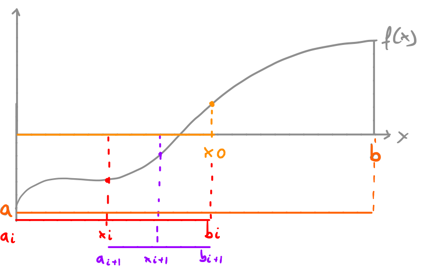


Does the solution exist?
Is it unique?

We are going to discuss three algorithms

Algorithm #1. Recursive bisection.

Assume $f \in C^0([a,b])$ and $f(a)<0$, $f(b)>0$



Given $[a_i, b_i]$

if $f(x_i)>0$: $a_{i+1}=a_i$
$b_{i+1}=x_i$

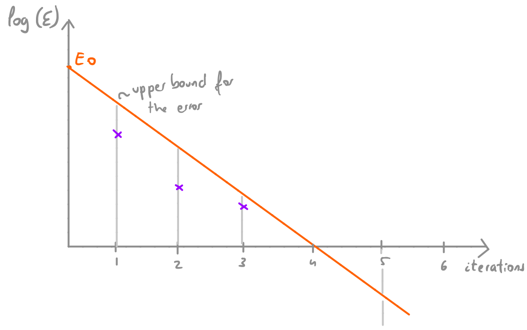
if $f(x_i)<0$: $a_{i+1}=x_i$
$b_{i+1}=b_i$

<u>Error</u> $\varepsilon_i = |x_i-\tilde{x}|$

It is guaranteed that $\tilde{x} \in [a_i, b_i]$, $\varepsilon \le \dfrac{|b_i-a_i|}{2}$ ← upper bound for the error.

Iteration $\varepsilon_0 \le \dfrac{|b_0-a_0|}{2} = E_0$ $\qquad E_{i+1} = \dfrac{E_i}{2}$

$\varepsilon_i \le E_i = \dfrac{E_{i-1}}{2} = \dots = \dfrac{E_0}{2^i}$



Method #2. Fixed-Point Iteration (FPI)

Rewrite $f(x)=0$ as $\varphi(x)=x$

$\boxed{\text{Iteration } x_{i+1}=\varphi(x_i)}$

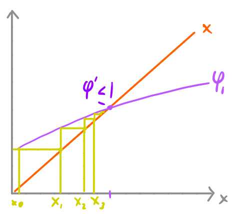
<u>Example</u> $f(x)=x^3+2=0$ $\qquad \varphi(x)=x^3+2+x$

$\varphi(x)=\dfrac{-2}{x^2}$

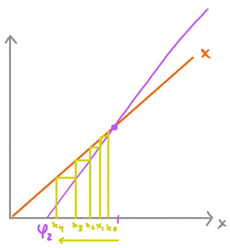
etc.

Observe that if $f(\tilde{x})=0 \Leftrightarrow \tilde{x}=\varphi(\tilde{x})$

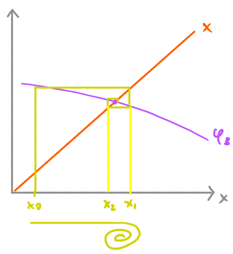
Observe that if $x_i=\tilde{x} \Rightarrow x_{i+1}=\tilde{x}$



Convergent



Divergent



Convergent (Oscillatory)

<u>Error</u> $\varepsilon_i = x_i - \tilde{x}$

The goal is to relate ε_{i+1} to ε_i

$$\varepsilon_{i+1} = x_{i+1}-\tilde{x} = \varphi(x_i)-\varphi(\tilde{x}) \longrightarrow$$

We make a trick and divide and multiply by $(x_i-\tilde{x})$

Fine as long as $x_i \neq \tilde{x}$

$$\longrightarrow \underbrace{\frac{\varphi(x_i)-\varphi(\tilde{x})}{(x_i-\tilde{x})}}_{\varphi'(\xi)} (x_i-\tilde{x}) = \varphi'(\xi)\varepsilon_i$$

$\xi \in [x_i, \tilde{x}]$

<u>Mean Value Theorem MVT</u>

$\varphi(x) \in C^1([a,b])$

$\varphi'(\xi)$ $\dfrac{\varphi(b)-\varphi(a)}{b-a}$ $\exists\, \xi \in [a,b]$

Assume $|\varphi'(x)|<k \quad \forall x \in [a,b]$

$$|\varepsilon_{i+1}| \le k|\varepsilon_i| \le k^2|\varepsilon_{i-1}| \le \dots \le k^i|\varepsilon_0|$$

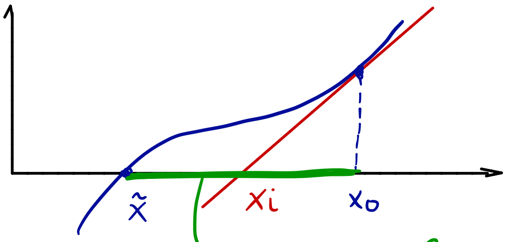
If $k<1 \Rightarrow$ convergence

$k>1 \Rightarrow$ divergence

Method #3 Newton

$$x_{i+1} = x_i - \underbrace{\frac{f(x_i)}{f'(x_i)}}$$

$$x_{i+1} = \varphi(x_i)$$



derivative less than one in this interval.

FPI $|\varphi'(\xi)| < 1$ $\quad \xi \in [x_i, \tilde{x}]$

$$\varphi' = \frac{f f''}{f'^2}$$

$\underline{[f' \neq 0]}$ ⟶ function can not touch the axis of the root.

Error behaviour:

$$\varepsilon_{i+1} = x_{i+1} - \tilde{x}$$

$$x_{i+1} = \varphi(x_i) = \underbrace{\varphi(\tilde{x} + \varepsilon_i)}$$

taylor expansion: nonlinear operator acting on $(a+b)$ when b is small.

$$x_{i+1} = \underbrace{\varphi(\tilde{x})}_{\tilde{x}} + \varphi'(\tilde{x})\varepsilon_i + \frac{1}{2}\varphi''(\tilde{x})\varepsilon_i^2 + \boxed{O(\varepsilon_i^3)}$$ ⟶ negligible because ε_i small

$$\varepsilon_{i+1} = \underbrace{\varphi'(\tilde{x})\varepsilon_i}_{\text{at root this is } = 0} + \frac{1}{2}\varphi''(\tilde{x})\varepsilon_i^2$$ ⟶ Quadratic convergence $\boxed{\varepsilon_{i+1} \propto \varepsilon_i^2}$

EXAMPLE:

$$f(x) = \cos(x) - 2x$$

Fixed point iteration $\varphi(x) = \frac{\cos x}{2}$ because needs to converge $\varphi'(x) = -\frac{\sin(x)}{2}$ ⟶ varies between $-\frac{1}{2}$ and $\frac{1}{2}$

FPI: $\varepsilon_{i+1} = \varphi'(\xi)\varepsilon_i$ ⟶ $|\varphi'(\xi)| < \frac{1}{2} \quad \forall \xi \in \mathbb{R}$ ⟶ error will reduce by ½ or more!

EXAMPLE:

MULTI-DIMENSIONAL ROOT-FINDING

Find $\tilde{x}$ s.t $f(\tilde{x}) = 0 \quad \underline{x} \in \mathbb{R}^n$

Taylor $f(x) = f(x_0) + \underbrace{f'}_{\text{JACOBIAN MATRIX}}(x_0)(\underline{x} - \underline{x_0}) = 0$ ⟶ $f = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \end{bmatrix} \quad \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}$

$$\underline{x} = x_0 - \frac{f(x_0)}{f'(x_0)} \longrightarrow \underline{x} = x_0 - (f'(x_0))^{-1} \cdot f(x_0)$$

TAYLOR APPROXIMATION

Theorem: Let $f \in C^{N+1}([x, x_0])$ then

continuous function if we differenciate

$$\boxed{f(x) = f(x_0) + f'(x_0)\cdot(x - x_0) + \frac{1}{2!} f(x_0)\cdot(x - x_0)^2 + \dots + \mathcal{O}(x - x_0)^{N+1}}$$

$$\boxed{f(x) = \sum_{n=0}^{N} \frac{1}{n!} f^{(n)}(x_0)(x - x_0)^n + \mathcal{O}\underbrace{(x - x_0)}^{N+1}}$$

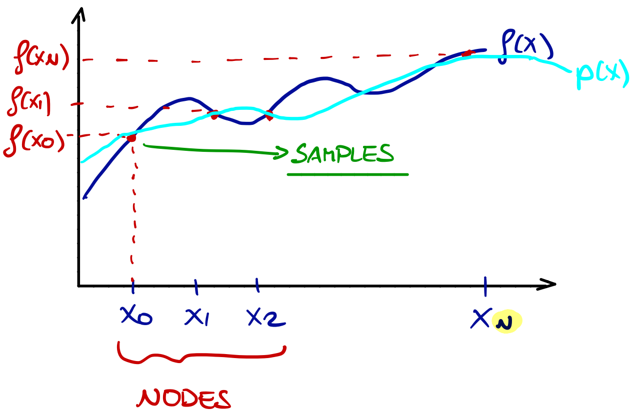
→ must be small in order to be useful.

Explicit form of $\mathcal{O}(x-x_0)^{N+1}$ $= \frac{1}{(N+1)!} f^{(N+1)}(\xi)(x - x_0)^{N+1}$ $\quad \xi \in [x - x_0]$

this is hard to find, so thats why $(x-x_0)$ must be small.

MODULE #2 | INTERPOLATION

Find a function $p(x)$ s.t. $f(x_i) = p(x_i)$ for $x_i \in [a,b] \subset \mathbb{R}$ $\quad i = 0, 1, \dots, N$



Find another function of p(x) that passes through the samples similar to EXCEL:

Interpolation means, find p(x) and worry later.

Usually polynomial functions.

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_N x^N = \sum_{i=0}^{N} a_i x^i$$

SOME MATEMATICAL NOTATIONS

$\forall$ for all $\qquad \Rightarrow$ implies

$\exists$ there exist $\qquad$ s.t. such that

$\mathcal{O}(h)$ order h $\qquad C^N(\mathbb{R})$ N times continuos differentiable function on $\mathbb{R}$

$f \in C^N$ f in C^N $\qquad A \subset B$ A subset of B

$\mathbb{P}^N$ Polynomial of degree N

Interpolation conditions

$p(x_i) = f(x_i) \quad \forall\, i \in \{\emptyset, \dots, N\}$

writing out.

1. $a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_N x_0^N = f(x_0) \quad i=0$
2. $a_0 + a_1 x_N + a_2 x_N^2 + \dots + a_N x_N^N = f(x_N) \quad i=N$

} linear in coefficients

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^N \\ 1 & x_1 & \cdots & x_1^N \\ \vdots & & \vdots & \\ 1 & x_N & \cdots & x_N^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}$$

Vandermonde matrix $\boxed{\underline{V}}$

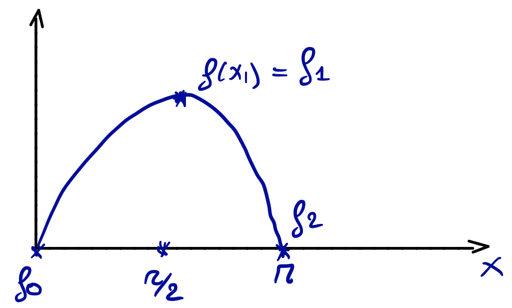
Properties of interpolation:

1. $p(x)$ to exist – Existance
2. $p(x)$ is unique – Uniqueness

When do we have this? If V is invertible

- $\det(V) = \prod_{0 \le i < j \le N} (x_j - x_i)$
- Implies $x_i \neq x_j \;\; \forall\, i \neq j$

Example $f(x) = \sin(x)$ on $[0, \pi]$



$\underline{x} = (0, \frac{\pi}{2}, \pi)$

$f = (0, 1, 0)$

$V\underline{a} = \underline{f}$

$$\boxed{V = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \pi/2 & \pi^2/4 \\ 1 & \pi & \pi^2 \end{bmatrix}}$$

Solve for a_i

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

$a_0 = 0$

$a_1 = 4/\pi$

$a_2 = -4/\pi^2$

$\boxed{p(x) = \frac{4}{\pi} x - \frac{4}{\pi^2} x^2}$ $\longrightarrow$ satisfies goes through the points.

Nyquist sample theorem: In an oscillating signal you need at least two samples per wavelength.

More generally:

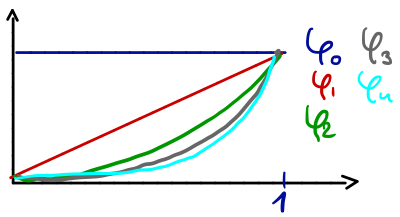
Interpolate $\phi(x) = \sum_{i=0}^{N} a_i \varphi_i(x)$

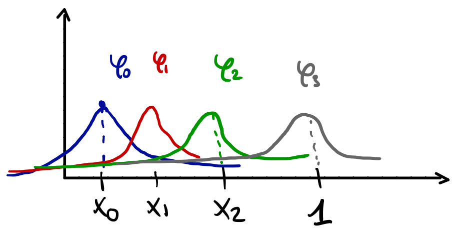
$\varphi_i \longrightarrow$ "basis functions"

e.g. $\varphi_i(x) = x^i$ monomial basis

$\varphi_i(x) = \sin\left(\frac{2\pi i x}{N}\right)$

$\varphi_i(x) = \exp\left[-(x - x_i)^2\right]$





Interp. conditions:

$$\left.\begin{aligned} a_0 \cdot \varphi_0(x_0) + a_1\varphi_1(x_0) + \dots + a_N \varphi_N(x_0) &= f(x_0) \\ a_0 \cdot \varphi_0(x_N) + a_1 \varphi_1(x_N) + \dots + a_N \varphi_N(x_N) &= f(x_N) \end{aligned}\right\} [N+1]$$

Matrix Equation:

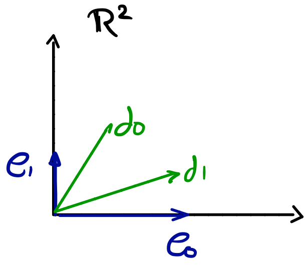
$$\begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) + \dots + & \varphi_N(x_0) \\ \varphi_0(x_N) & \varphi_1(x_N) + \dots + & \varphi_N(x_N) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_N \end{bmatrix}$$

Non singular

Polinomial case: comparison to linear algebra:

$\varphi_i(x) = x^i$

1. $p = a_0 \underline{e_0} + a_1 \underline{e_1}$
2. $p = b_0 \underline{d_0} + b_1 \underline{d_1}$



$\{e_0, e_1\}$ and $\{d_0, d_1\}$ span $\mathbb{R}^2$

Basis $\Big\}$ Similarly for polynomials:

Assuming x^i, $i=0,\dots,N$ forms a basis for $\mathbb{P}^N$. For any $p \in \mathbb{P}^N$ $p = \sum_{i=0}^{N} a_i x^i$

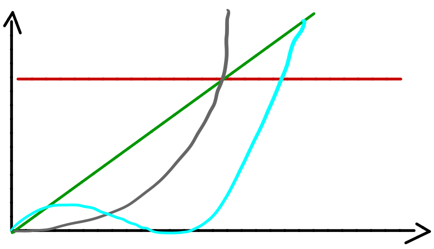
Newton Basis:

Def: $n_0(x) = 1$

$n_1(x) = x - x_0$

$n_2(x) = (x-x_0)(x-x_1)$

$\Big\}$ $n_k(x) = \prod_{j=k} (x - x_j)$



Property: $n_k(x_j) = 0$, $j<k$

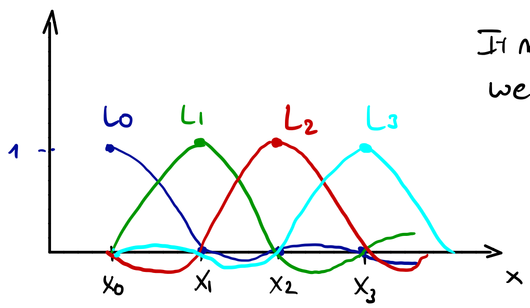
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & x-x_0 & 0 & 0 & 0 \\ 1 & x_i - x_0 & (x\ \ ) & 0 & 0 \\ \cdot & \cdot & \cdot & \ddots & 0 \end{bmatrix}$$

Lower triangular.

- Invertible
- Find coefficients easily!

Lagrange Basis:

Can we obtain $A = I$? Done with $L_i(x_j) = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$



It needs to be a cubic because we have three roots.

Directly:

$\hat{l}_0(x) = (x-x_1)\cdot(x-x_2)(x-x_3)$ three roots and 0 at locations

$\hat{l}_0(x_0) = 1 = (x_0-x_1)(x_0-x_2)(x_0-x_3) \neq 0 \longrightarrow l_0(x) = \dfrac{\hat{l}_0(x)}{\hat{l}_0(x_i)}$

$l_k(x) = \prod_{\substack{j \neq k \\ j=0}}^{N} \dfrac{(x-x_j)}{(x_k - x_j)}$ $\quad p(x) = \sum_{i=0}^{N} f_i l_i(x)$

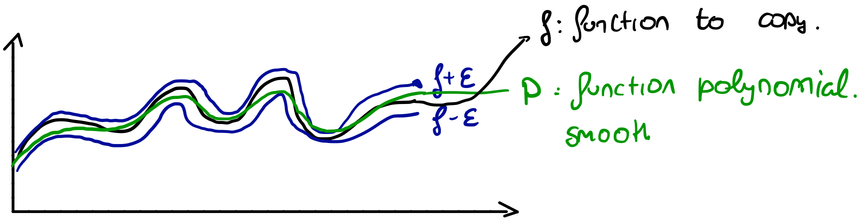
Why do we use polynomials:

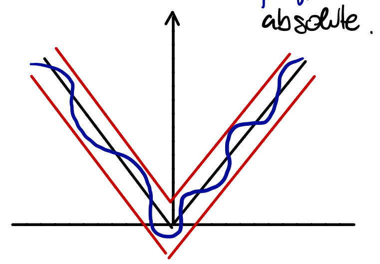
- smooth.
- easy to differentiate and integrate.
- unique interpolation.

Weierstrass Theorem:

$\exists p \in P^N$, the same N, st for any $\varepsilon > \emptyset$ for any $f \in C^0$

$$\max_x |f(x) - p(x)| < \varepsilon$$





Interpolation error:

If is $f \in C^{N+1}([a,b])$, $\quad p(x_i) = f(x_i) \ \forall i$

$$\varepsilon = f(x) - p(x) = \frac{f^{(N+1)}(\xi)}{N+1} \cdot W_{(N+1)}(x) \longrightarrow W_{N+1} = \prod_{i=0}^{N} (x - x_i) \text{ - Nodal poly}$$

Proof 1.

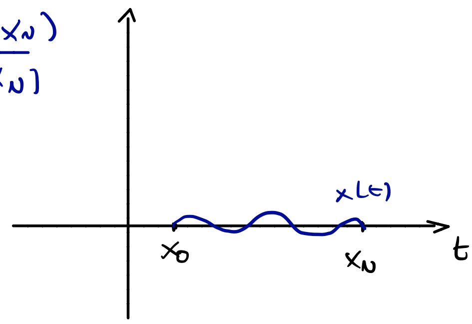
1. $x = x_0 \qquad f(x_i) - p(x_i) = 0 \qquad w(x_i) = 0$

2. $x \neq x_i \qquad g(t) = f(t) - p(t) - \left[f(x) - p(x)\right] \dfrac{(t - x_i)\dots(t - x_N)}{(x - x_0)\dots(x - x_N)}$

Properties of g, Roots of g? $t = x_0 \rightarrow g = 0$

On $[a,b]$ g has $N+2$ roots. $t = x \rightarrow g = 0$

g is $N+1$ times differentiable.



$$0 = g^{(N+1)}(\xi) = f^{(N+1)} - \underbrace{p^{(N+1)}}_{0} - \left(f(x) - p(x)\right)\frac{N!}{w(x)}$$

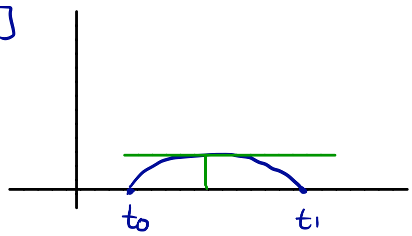
$$\Rightarrow f(x) - p(x) = \frac{f^{(N+1)}(\xi)}{(N+1)} w(x)$$

Rolle's Theorem

$g \in [a,b]$

$g(t_i) = 0 \quad i = 0, 1$

$g \in C^1 \longrightarrow g'(\xi) = 0$



Generalized:

N: $g(t_i) = 0 \quad i = 0, \dots, N$

$g \in C^{N+1} \rightarrow g^{(N+1)}(\xi) = 0$

Thеm (Control of ω)

Of all polynomials P^n with leading coefficient 1.

the best minimum absolute value of $[-1,1]$ is $\dfrac{T_N(x)}{2^{(N+1)}}$ → Chebisev. $x = \cos\theta$

Def: $T_n(\cos\theta) = \cos(N\theta) \longleftrightarrow T_n(x) = \cos(N\cos^{-1}(x))$

Possible $\cos(N\theta) = p(\cos\theta)$

$N=0$	$\cos\theta = 1$	$T_0 = 1$
1	$\cos\theta = \cos\theta$	$T_1 = x$
2	$\cos 2\theta = 2\cos^2\theta - 1$	$T_2 = 2x^2 - 1$

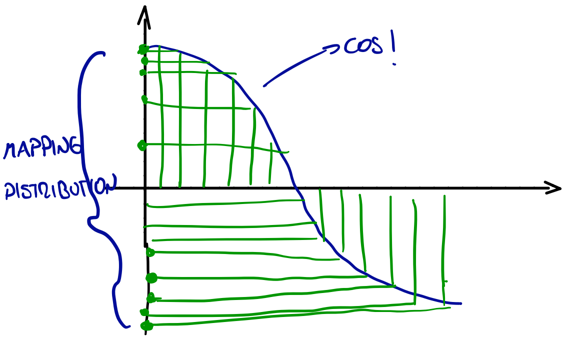
$$\boxed{\text{Property: } \max_x |T_N(x)| = 1}$$

$$\max_{\xi} \left| \frac{T_N(\xi)}{2^{N-1}} \right| = \frac{1}{2^{N-1}}$$

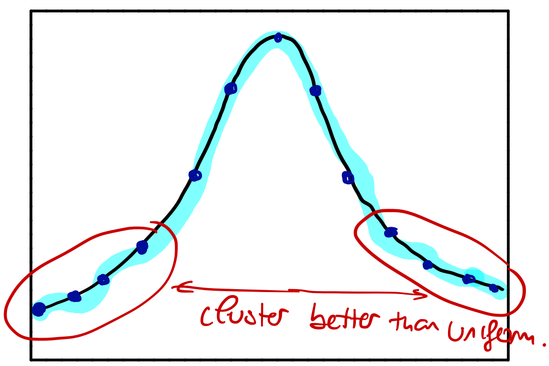
$$\boxed{|f(x) - p(x)| \leq \frac{f^{(N+1)}(\xi)}{(N+1)!} \cdot \frac{1}{2^{N+1}}}$$ bound of error

WHAT ARE THE T_N's

1. $T_N(x) = \cos(N\arccos(x))$
2. $T_N(\cos(x)) = \cos(Nx)$
3. Degree N polynomial with roots $\eta_i = \cos\left(\frac{2i-1}{2N}\pi\right)$ $i = 1, \dots, N$



EXAMPLE



PROBLEMS:

For large numbers of N. Oscillations on the sides may occur that is why we use piecewise polynomial interpolation.

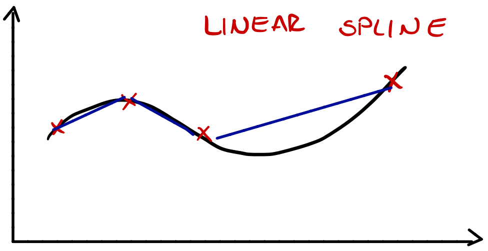
MODULE 3 — SPLINES, MULTI-DIMENSIONS AND RADIAL BASES

Spline of degree S: (datapoints $a = x_0 < x_1 < x_2 < \dots < x_N = b$)

S is a spline of degree D if:

1. S is a polinomial of degree $\le d$ on $[x_i, x_{i+1}]$
2. $S, S', S'', \dots, S^{(d-1)}$ continuous on the entire interval.

polynomial in each subinterval, but not in total.



$$S = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots \\ S_{N-1}(x) & x \in [x_{N-1}, x_N] \end{cases}$$

Cubic splines are more used because derivative is not immediately a constant.

$$\boxed{S_i(x) = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} \cdot (x - x_i) + f_i \qquad \text{for } x \in [x_i, x_{i+1}]}$$

CUBIC SPLINES:

$[a,b]$

$i = 0, 1, 2, 3, \dots, N-1 \rightarrow$ D.O.F $= \boxed{4 \cdot N}$

$4(a_i + b_i + c_i + d_i)$

1. $s \in C^2[a,b]$
2. $[x_i, x_{i+1}]$ $\quad S_i(x_i) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$

$$S_i'(x_i) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

$$S_i''(x_i) = 6a_i(x - x_i) + 2b_i$$ $\longrightarrow$ pretend we know this. $S_i''(x_i) = M_i$

$2b_i = M_i$

$$\boxed{b_i = \frac{M_i}{2}}$$

$S_i''(x_{i+1}) = M_{i+1} \longrightarrow 6a_i \underbrace{(x_{i+1} - x_i)}_{\ell_i \to \text{spacing}} + M_i = M_{i+1}$

$$\boxed{a_i = \frac{M_{i+1} - M_i}{6\ell_i}}$$

$$S_i(x) = \frac{M_{i+1} - M_i}{6\ell_i}(x - x_i)^3 + \frac{M_i}{2}(x - x_i)^2 + c_i(x - x_i) + d_i$$ $\longrightarrow$ need more constraints

$S_i(x_i) = f_i$ continuity of function. $\longrightarrow$ $\boxed{d_i = f_i}$

$S_i(x_{i+1}) = f_{i+1} \longrightarrow$

$$\frac{M_{i+1}}{6\ell_i}\ell_i^3 + \frac{M_i}{2}\ell_i^2 + c_i\ell_i + f_i = f_{i+1}$$

$$c_i = \frac{f_{i+1} - f_i}{\ell_i} - \frac{\ell_i}{3}M_i - \frac{\ell_i}{6}M_{i+1}$$

$$S_i(x) = \frac{M_{i+1} - M_i}{6 l_i} \cdot (x - x_i)^3 + \frac{M_i}{2}(x - x_i)^2 + \left(\frac{f_{i+1} - f_i}{l_i} - \frac{l_i}{3} M_i - \frac{l_i}{6} M_{i+1} \right)(x - x_i) + f_i$$

still to determine

First derivative

$$S_i'(x) = \frac{M_{i+1} - M_i}{2 l_i}(x - x_i)^2 + \frac{M_i}{4}(x - x_i) + \left(\frac{f_{i+1} - f_i}{l_i} - \frac{l_i}{3} M_i - \frac{l_i}{6} M_{i+1} \right)$$

$S_i'(x_i) = S_{i-1}'(x_i) \longrightarrow$ continuity.

$$S_i'(x_i) = \left(\frac{f_{i+1} - f_i}{l} - \frac{l_i}{3} M_i - \frac{l_i}{6} M_{i+1} \right)$$

$$S_{i-1}'(x) = \frac{M_i - M_{i-1}}{2 l_{i-1}}(x - x_{i-1})^2 + \frac{M_{i-1}}{4}(x - x_{i-1}) + \left(\frac{f_i - f_{i-1}}{l_{i-1}} - \frac{l_{i-1}}{3} M_{i-1} - \frac{l_{i-1}}{6} M_i \right) \quad =$$

$$S_{i-1}'(x_i) = \frac{M_i - M_{i-1}}{2 l_{i-1}}(l_{i-1})^2 + \frac{M_{i-1}}{4}(l_{i-1}) + \left(\frac{f_i - f_{i-1}}{l_{i-1}} - \frac{l_{i-1}}{3} M_{i-1} - \frac{l_{i-1}}{6} M_i \right)$$

$$= \frac{l_{i-1}}{6} M_{i-1} + \frac{l_{i-1}}{3} M_i + \frac{f_i - f_{i-1}}{l_{i-1}}$$

first derivative continuous.

$$\frac{l_{i-1}}{6} M_{i-1} + \frac{l_{i-1} + l_i}{3} M_i + \frac{l_i}{6} M_{i+1} = \frac{f_{i+1} - f_i}{l_i} - \frac{f_i - f_{i-1}}{l_{i-1}} \quad \text{for } i = 1, 2, 3 \dots N-1$$

We have $N+1$ unknowns, and we have less constraints.

USE: NATURAL CUBIC SPLINE $N+1$ unknowns $N-1$ constants.

- $M_0 = 0 \quad M_N = 0$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & & 0 \\ \frac{l_0}{6} & \frac{l_0 + l_1}{6} & \frac{l_1}{6} & 0 & \cdots & 0 \\ 0 & \frac{l_1}{6} & \frac{l_1 + l_2}{3} & \frac{l_2}{6} & 0 \cdots & 0 \\ 0 & 0 & 0 & \frac{l_{n-2}}{6} & \frac{l_{n-2} + l_{n-1}}{3} & \frac{l_{n-1}}{6} \\ 0 & 0 & 0 & \cdots & & 1 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{f_2 - f_1}{l_1} - \frac{f_1 - f_0}{l_0} \\ \frac{f_3 - f_2}{l_2} - \frac{f_2 - f_1}{l_1} \\ \vdots \\ \frac{f_N - f_{N-1}}{l_{N-1}} - \frac{f_{N-1} - f_{N-2}}{l_{N-2}} \\ 0 \end{bmatrix}$$

tridiagonal matrix

FORMALIZED SOLUTION

$A \cdot M = f \qquad A$ is tridiagonal $\qquad M = A^{-1} \cdot f$

if $|a_{ii}| > \sum_{j \neq i} |a_{ij}| \longrightarrow A$: strictly diagonally dominant. $\longrightarrow \boxed{\det(A) \neq 0}$ then we can invert.

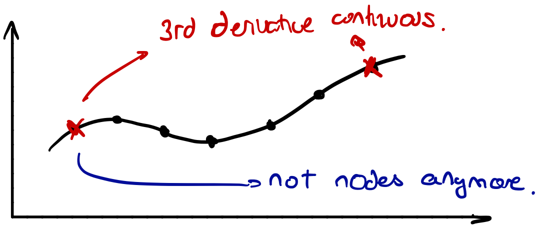
$$\underbrace{\left| \frac{h_{i-1} + h_i}{3} \right|}_{\text{general}} > \left| \frac{h_{i-1}}{6} \right| + \left| \frac{h_i}{6} \right| \longrightarrow \text{true? yes always true}$$

$\boxed{\text{TRUE}}$

Other options

$S_0''(x_0) = M_0 = f''(x_0)$ instead of $M_0 = 0$

not-a-knot spline



Bivariate Interpolation

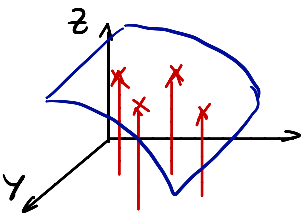
Set of data points:

$\{\underline{x} : i = 0, 1, \dots, n\} \in D \subset \mathbb{R}^2$

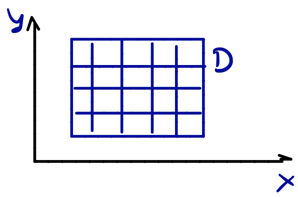
$\{f_i : i = 0, 1, \dots, n\}$

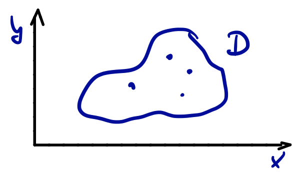
find a function that interpolates these data points.

$p(x_i) = f_i \quad i = 0, 1, \dots, n$



Build a surface which passes through all these points.



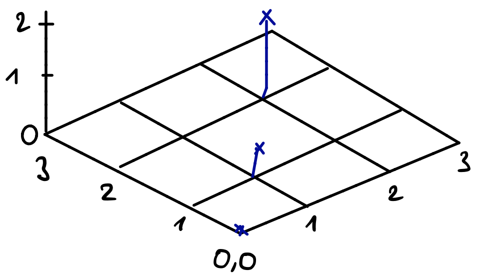


$$P(x,y) = \sum_{h=0}^{n} c_h \cdot b(x,y) \longrightarrow P(x_i) = f_i \longrightarrow \sum_{k=0}^{n} c_k b_k(x_i) = f_i$$

$A \cdot c = f \qquad c = A^{-1} \cdot f$

$A_{ij} = b_j(x_i) \qquad \det(A) \neq 0$

Example



$P(x,y) = a_0 + a_1 x + a_2 y$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \end{bmatrix}}_{A} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$a_0 = 0$

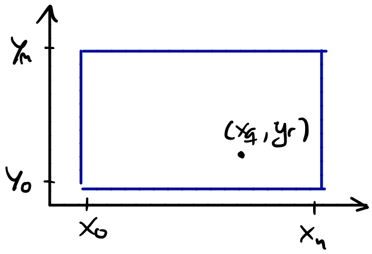
$a_1 + a_2 = 1$

$\det(A) = 0$

$P(x,y) = a_1 x + (1 - a_1) y$ is a solution, no unique solution.

TENSOR - PRODUCT INTERPOLATION

$$\{(x_i, y_j)\}$$



split between functions depending only on x and only on y.

$$b_{ij}(x,y) = \varphi_i(x) \cdot \psi_j(y)$$

$$p(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{m} \varphi_i(x) \cdot \psi_j(y) \cdot C_{ji}$$

$$\varphi_i(x) = \mathcal{L}_i^{(x)}(x) = \prod_{\substack{h=0 \\ h \neq i}}^{n} \frac{(x - x_h)}{(x_i - x_h)}$$

$$\psi_j(y) = \mathcal{L}_j^{(y)}(y) = \prod_{\substack{h=0 \\ h \neq j}}^{m} \frac{(y - y_h)}{(y_j - y_h)}$$

Basis function $b_{ij}(x_q, y_r)$?

$$\mathcal{L}_i^{(x)}(x_q) \cdot \mathcal{L}_j^{(y)}(y_r) = \begin{cases} 1 & q = i \text{ and } r = j \\ 0 & \text{otherwise} \end{cases}$$

$P(x_q, y_r) = \sum_{i=0}^{n} \sum_{j=0}^{m} b_{ij}(x_q, y_r) \cdot C_{ij} =$ C_{qr} is the only term that "survives" because the rest go to 0. $\longrightarrow = f_{qr} \rightarrow \boxed{C_{ij} = f_{ij}}$

due to the fact of using the lagrange basis.

$$P(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{m} \cdot f_{ij} \cdot \mathcal{L}_i^{(x)}(x) \cdot \mathcal{L}_j^{(y)}(y)$$

EXAMPLE:

find the bilinear polynomial $p(x,y)$ from the data.

$f(0,0) = 1 \quad f(1,0) = f(0,1) = f(1,1) = 0$

$\mathcal{L}_0(x) = \frac{x-1}{0-1} = 1 - x$	$\mathcal{L}_0(y) = \frac{y-1}{0-1} = 1 - y$
$\mathcal{L}_1(x) = \frac{x-0}{1-0} = x$	$\mathcal{L}_1(y) = \frac{y-0}{1-0} = y$

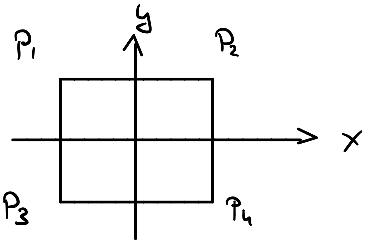
$$P(x,y) = f_{0,0}\, \mathcal{L}_0(x) \cdot \mathcal{L}_0(y) + \cancel{f_{1,0} \cdot \mathcal{L}_1(x)}^{0} \cdot \mathcal{L}_0(y) + \cancel{f_{0,1}}^{0} \cdot \mathcal{L}_0(x) \cdot \mathcal{L}_1(y) + \cancel{f_{1,1}}^{0} \cdot \mathcal{L}_1(x) \cdot \mathcal{L}_1(y)$$

$$P(x,y) = (1-x)(1-y) = \boxed{P\left(\tfrac{1}{2}, \tfrac{1}{2}\right) = \left(\tfrac{1}{4}\right)}$$

When n and m are increased a lot the runge fenomenon on the edges does not work.

Patch Interpolation Rectangular

- In 1D, to avoid oscillations of the interpolant we use splines.
- In 2D, we can use patch interpolation
- Perform the interpolation in rectangular or triangular pieces.



$P_1\ (-1,1) \longrightarrow f_1$
$P_2\ (1,1) \longrightarrow f_2$
$P_3\ (-1,-1) \longrightarrow f_3$
$P_4\ (1,-1) \longrightarrow f_4$

$$P(x,y) = a_0 + a_1 x + a_2 y + a_3 xy$$

$$\underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}}_{a} = \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}}_{f}$$

$$a^T = [a_0, a_1, a_2, a_3]$$

$$b^T(x,y) = [1, x, y, xy]$$

$$P(x,y) = b^T(x,y) \cdot a = \underbrace{b^T(x,y) \cdot A^{-1}}_{\text{shape function } S(x,y)} \cdot f$$

$\longrightarrow$ different notation.

$$S(x,y) = [1, x, y, xy] \cdot \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}$$

$$S(x,y) = \frac{1}{4} \cdot \begin{bmatrix} 1 & -x & +y & -xy \\ 1 & +x & +y & +xy \\ 1 & -x & -y & +xy \\ 1 & +x & -y & -xy \end{bmatrix} = \frac{1}{4} \begin{bmatrix} (1-x)(1+y) \\ (1+x)(1+y) \\ (1-x)(1-y) \\ (1+x)(1-y) \end{bmatrix}$$

$$P(x,y) = S(x,y) \cdot f = \sum_{i=1}^{4} S_i(x,y) \cdot f_i$$

Not always a rectangle, so we should mapp into a rectangle whatever the surface.

Example: 1

Find $p(x,y)$ of $f(0,0,1,2)$, on the unit rectangle

determine $p(0,0)$

$$p(x,y) = \frac{f_1}{4}(1-x)(1+y) + \frac{0}{4}(1+x)(1+y) + \frac{1}{4}(1-x)(1-y) + \frac{2}{4}(1+x)(1-y)$$

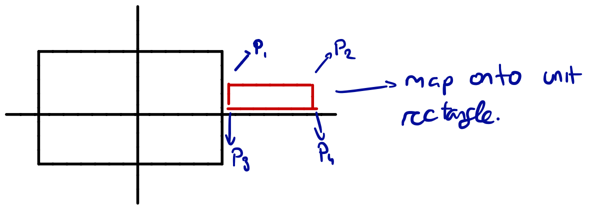
$$\boxed{p(0,0) = \frac{3}{4}}$$

EXAMPLE 2

$P_1(1, 1/2) \quad P_2 = (2, 1/2) \quad P_3 = (1, 1/5) \quad P_4 = (2, 1/5)$

$f_1 = 1 \quad f_2 = 2 \quad f_3 = 3 \quad f_4 = 4$

$\Big\} \; P = (3/2 \,,\, 1/4)\ ?$

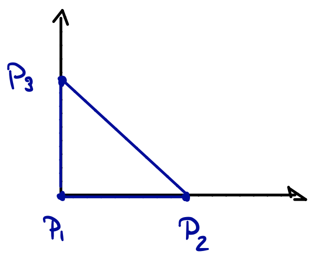


$x \in [a_x, b_x] \to \xi_x \in [-1, 1]$

$$\xi_{xp} = 2\,\frac{x_p - a_x}{b_x - a_x} - 1 = 2\,\frac{x_p - 1}{2 - 1} - 1 = 2x - 3$$

$$\xi_y = 2\,\frac{y - a_y}{b_y - a_y} - 1 = 2 \cdot \frac{y - 1/5}{1/2 - 1/5} - 1 = -\frac{2}{3}$$

PATCH INTERPOLATION: TRIANGULAR PATCH



$P_1 = (0,0) \longrightarrow f_1$

$P_2 = (1,0) \longrightarrow f_2$

$P_3 = (0, 1) \longrightarrow f_3$

$P(x,y) = a_0 + a_1 x + a_2 y$ 3 DOF - 3 points

INTERPOLATION CONDITION

$P(\underline{x}_i) = f_i \quad i = 1, 2, 3$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}}_{a} = \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}}_{f}$$

$P(x,y) = b^T(x,y) \cdot a \qquad b^T(x,y) = [1, x, y]$

$P(x,y) = \underbrace{b^T(x,y) \cdot A^{-1}}_{S(x,y)} \cdot f$

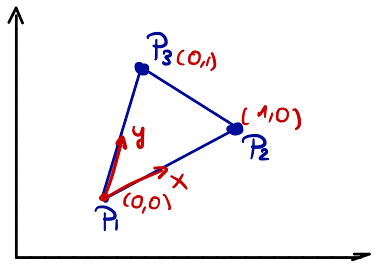
$\det(A) = \underline{1}$

$$A^* \cdot \begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad A^{-1} = \frac{1}{\det(A)} \cdot A^{*T} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$S(x,y) = [1, x, y] \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 - x - y \\ x \\ y \end{bmatrix}$$

$$\boxed{P(x,y) = \sum_{i=1}^{3} S_i(x,y) \cdot f_i = (1 - x - y) f_1 + x f_2 + y f_3}$$

How do we map triangles into unit triangles?



$P_i = (u_i, v_i)$ To make the transformation from $(x,y) \to (u,v)$

$$\begin{cases} u = u_1 + (u_2 \cdot u_1)x + (u_3 - u_1)y \\ v = v_1 + (v_2 \cdot v_1)x + (v_3 - v_1)y \end{cases}$$

we want to do the oposite

long equations

$x = x(u,v)$

$y = y(u,v)$

$$S(x,y) = S(x(u,v), y(u,v)) \Rightarrow S^*(u,v)$$

$$P(x,y) = P(u,v) = \sum_{i=1}^{3} S_i^*(u,v) \cdot f_i$$

Radial Basis Functions

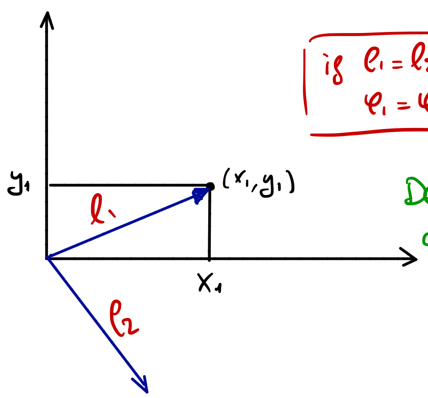
Scattered grid instead of regular, to get continuous and smoot approximation, same as in 2D.

Guarantees uniquenes and existence of bivariate interpolation for arbitrary distinct data points.

$\phi : \mathbb{R}^2 \to \mathbb{R}$ is a radial function if a function $\varphi\ [0,+\infty)$ exists such that

$(x,y) \longrightarrow$ real number

$$\phi(\underline{x}) = \varphi(r) \qquad r = \|x\| = \sqrt{x^2+y^2}$$



if $\ell_1 = \ell_2$
$\varphi_1 = \varphi_2$

if $\|x_1\| = \|x_2\|$
$\to \phi(x_1) = \phi(x_2)$

Depends on the norm and not on x and y separately.

$\{\underline{x_i} : i = 0, 1, \dots, n\}$

$\{f_i : i = 0, 1, \dots, n\}$

RBF interpolant:

$$\boxed{S(\underline{x}) = \sum_{J=0}^{N} a_J \cdot \phi(\|x - x_J\|)}$$

$$S(\underline{x_i}) = f_i \quad i = 0, 1, \dots, n$$

$$\sum_{J=0}^{n} a_J \cdot \phi(\|\underline{x_i} - x_J\|) = f_i$$

$$A = \begin{bmatrix} \phi(\|x_0 - x_0\|) & \phi(\|x_0 - x_1\|) & \cdots & \phi(\|x_0 - x_n\|) \\ \phi(\|x_1 - x_0\|) & \phi(\|x_1 - x_1\|) & \cdots & \phi(\|x_1 - x_n\|) \\ \vdots & \vdots & & \vdots \\ \phi(\|x_n - x_0\|) & \phi(\|x_n - x_1\|) & \cdots & \phi(\|x_n - x_n\|) \end{bmatrix}$$

$$A \cdot a = f \qquad A_{iJ} = \phi(\|x_i - x_J\|)$$

$A_{ii} = \phi(0) \longrightarrow$

$A_{ij} = A_{ji} \longleftrightarrow \|x_i - x_J\| = \|x_J - x_i\|$

Radial Function Interpolation

- Gaussian $\phi(r) = \exp\left[-(\gamma \cdot r)^2\right]$
- Inverse multiquadratic $\dfrac{1}{\sqrt{1+(\gamma \cdot r)^2}}$
- Inverse quadratic $\phi(r) = \dfrac{1}{1+(\gamma \cdot r)^2}$
- Linear $\phi(r) = r$

EXAMPLE:

$\phi(r) = \exp\left[-(\gamma \cdot r)^2\right]$ used to interpolate $x = \{1,3,7\}$ $f = \{1, 1, 1\}$

$x_0 - x_0 = 0 \longrightarrow \phi(0) = e^{-(\gamma \cdot 0)^2} = 1$

$x_1 - x_0 = 2 \longrightarrow \phi(2) = e^{-4\gamma^2}$

$x_2 - x_0 = 6 \longrightarrow \phi(6) = e^{-36\gamma^2}$

$x_2 - x_1 = 4 \longrightarrow \phi(4) = e^{-16\gamma^2}$

Sure the solution exists, but maybe is not a good solution!

- If $\gamma = 10$

$$\begin{bmatrix} 1 & \exp(-400) & \exp(-3600) \\ \exp(-400) & 1 & \exp(-1600) \\ \exp(-3600) & \exp(-1600) & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

NOT good approximation!

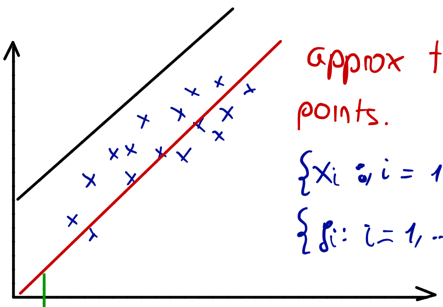
If $\gamma = 1 \rightarrow$ better
If $\gamma = 0.1 \rightarrow$ perfect.
} in the order of separation of datapoints.

Least Square Regression

Approximate the values of the data points instead of finding a function that passes through the data points.

$$\phi(x) = \sum_{i=1}^{m} c_i \, \varphi_i(x) \qquad m < n$$

it does not care about order, more DOF.



approx trend of data points.

$\{x_i : i = 1, \dots, n\}$
$\{f_i : i = 1, \dots, n\}$

$\rightarrow$ the blue is better than the black because the sum of the residuals square is less

$$R = \sum_{i=1}^{n} r_i^2 \;\Big|\; r_i = f'(x_i) - \phi(x_i) \quad i = 1, \dots, n$$

$\phi(x) = a_0 + a_1 x \qquad r_i = f_i - a_0 - a_1 x$

How we determine coefficients?

$$R = \sum_{i=1}^{m} (f_i - a_0 - a_1 x_i)^2 \rightarrow \frac{\partial R}{\partial a_0} = -2 \sum_{i=0}^{m} (f_i - a_0 - a_1 x_i) = 0$$

$$\sum_{i=1}^{n} a_0 + \sum_{i=1}^{n} x_i a_1 = \sum_{i=1}^{n} f_i \rightarrow n a_0 + \sum_{i=1}^{n} x_i a_1 = \sum_{i=1}^{n} f_i$$

$$\frac{\partial R}{\partial a_1} = 0 \qquad \sum x_i a_0 + \sum x_i^2 a_1 = \sum x_i f_i$$

$$n a_0 + \sum_{i=1}^{n} x_i a_i = \sum_{i=1}^{n} f_i$$

$$\sum x_i a_0 + \sum x_i^2 a_1 = \sum x_i f_i$$

} linear system of 2 eq w 2 unk

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum x_i f_i \end{bmatrix}$$

$f(x), g(x) \rightarrow$ scalar product of $\langle f, g \rangle = \sum_{i=1}^{n} f(x_i) \cdot g(x_i)$

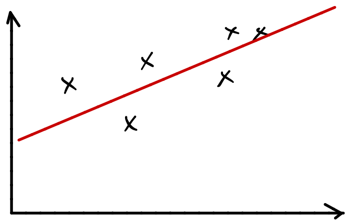
$$\begin{bmatrix} \langle 1,1 \rangle & \langle 1,x \rangle \\ \langle 1,x \rangle & \langle x,x \rangle \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \langle 1,f \rangle \\ \langle x,f \rangle \end{bmatrix}$$

$$\phi(x) = \sum_{j=1}^{m} a_j \varphi_j(x) \qquad m \leq n \qquad r_i = f_i - \phi(x_i) = f_i - \sum_{j=1}^{m} a_j \varphi_j(x_i)$$

$$R = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \left[f_i - \sum_{j=1}^{m} a_j f_j(x_i) \right]^2 \qquad \frac{\partial R}{\partial a_j} = 0 \qquad j = 1, \dots, m$$

$$\begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \langle \varphi_1, \varphi_2 \rangle & \cdots & \langle \varphi_1, \varphi_m \rangle \\ \langle \varphi_2, \varphi_1 \rangle & \vdots & & \vdots \\ \vdots & & & \\ \langle \varphi_m, \varphi_1 \rangle & \cdots & \cdots & \langle \varphi_m, \varphi_m \rangle \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \langle f, \varphi_m \rangle \end{bmatrix}$$

$$\langle \varphi_2, \varphi_1 \rangle = \sum_{i=1}^{n} \varphi_2(x_i) \cdot \varphi_1(x_i)$$

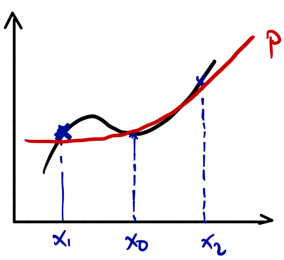


$$\boxed{\text{ROOT MEAN SQUARED ERROR} \quad \sigma_{RMS} = \sqrt{\frac{\langle r, r \rangle}{n}}}$$

How well the trend of the data points its approx by our approximant

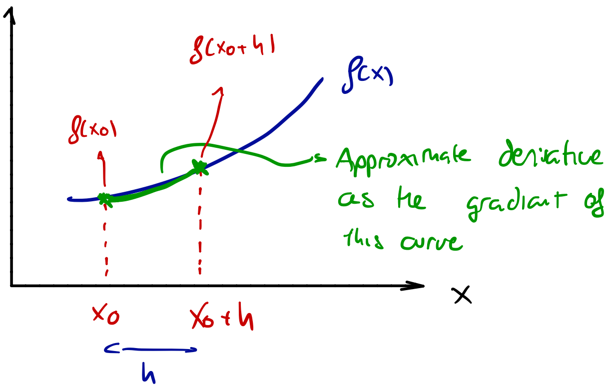
NUMERICAL DIFFERENTIATION

Approximate $\underbrace{f'(x)}_{\text{derivative}} \simeq \sum_{j=0}^{M} a_j f(x_j) \longrightarrow$ solve root finding faster.



$p(x) \rightarrow$ evaluate derivative $\rightarrow$ and it will be an approximation. If we make the points closer will be a good approximation.

METHOD 1: FORWARD DIFFERENCES



$$\boxed{f'(x_0) \simeq \underbrace{\frac{f(x_0+h) - f(x_0)}{h}}_{D_F}}$$

Error?

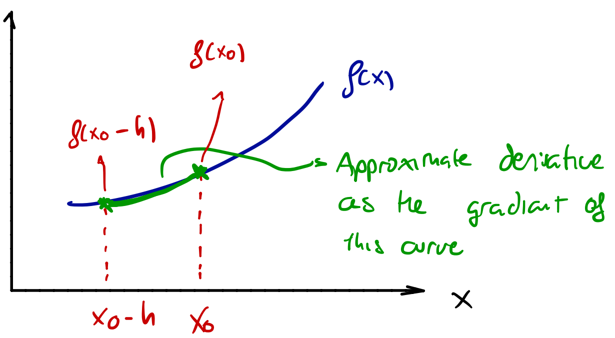
$$\boxed{\varepsilon = D_F(f) - f'(x_0)}$$

Apply Taylor series to $f(x_0+h) = f(x_0) + f'(x_0)h + \frac{1}{2} f''(x_0) h^2 + \mathcal{O}(h^3)$

Substitude

$$\underbrace{\frac{f(x_0+h) - f(x_0)}{h}}_{\text{approx of } f'(x)} = f'(x_0) + \underbrace{\frac{1}{2} f''(x_0) h + \mathcal{O}(h^2)}_{\text{Truncation error}}$$

METHOD 2: BACKWARD DIFFERENCES



$$f'(x_0) \simeq \frac{f(x_0) - f(x_0-h)}{h} \qquad D_B$$

order

ERROR

$$f(x_0-h) = f(x_0) - h f'(x_0) + \frac{1}{2} f''(x_0) + \mathcal{O}(h^3)$$

$$\frac{f(x_0) - f(x_0-h)}{h} = f'(x_0) - \frac{1}{2} f''(x_0) + \mathcal{O}(h^2)$$

Which one is better? both are the same.

METHOD 3: CENTRAL DIFFERENCES

Adding both methods together

$$f'(x_0) \simeq \frac{1}{2}\left(D_F(f) + D_B(f)\right) \simeq \frac{1}{2}\left[\frac{f(x_0+h) - f(x_0)}{h} + \frac{f(x_0) - f(x_0-h)}{h}\right]$$

$$\boxed{f'(x_0) \simeq \frac{f(x_0+h) - f(x_0-h)}{2h}}$$

Error:?

$$\left.\begin{array}{l} \text{Taylor } f(x_0+h) = f + hf + \frac{1}{2} h^2 f' + \frac{1}{3!} h^3 f''' + \mathcal{O}(h^4) \\ \text{Taylor } f(x_0-h) = f - hf' + \frac{1}{2} h^2 f'' - \frac{1}{3} h^3 f''' + \mathcal{O}(h^4) \end{array}\right\} \quad \cancel{2h} f' = \cancel{2h} f' + \left.\frac{1}{3!} h^2 f''' + \mathcal{O}(h^3)\right\} \text{error}$$

Generalized Difference Rules

$$D[f] = \sum_{i=0}^{n} a_i f(x_0 + ih) \longrightarrow \text{solve for coefficients} \qquad f'(x_0) = D[f] + \mathcal{O}(h^n)$$

find rules for highest possible orders $\quad \underline{n=3}$

$\boxed{}$ 4x4 invertible

$$\begin{array}{c|cccc}
n=3 & a_0 & a_1 & a_2 & a_3 \\
\text{Taylor of} & f(x_0) & f(x_0+h) & f(x_0+2h) & f(x_0+3h) \\
\hline
f(x_0) & 1 & 1 & 1 & 1 \\
hf'(x_0) & 0 & 1 & 2 & 3 \\
\frac{1}{2!}h^2 f'' & 0 & 1 & 4 & 9 \\
\frac{1}{3!}h^3 f''' & 0 & 1 & 8 & 27 \\
\frac{1}{4!}h^4 f'''' & 0 & 1 & 16 & 81
\end{array}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 4 & 9 \\ 0 & 1 & 8 & 27 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ h \\ 0 \\ 0 \end{bmatrix}$$

$$a = \frac{1}{6}(-11, 18, -9, 2) \longrightarrow \text{build difference rule}$$

$$f'(x_0) = \frac{1}{6}\left(-11 \cdot f(x_0) + 18 \cdot f(x_0+h) - 9 \cdot f(x_0+2h) + 2 f(x_0+3h)\right)$$

is it invertible?

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 \\ x_0^3 & x_1^3 & x_2^3 & x_3^3 \end{bmatrix}$$

Balancing Truncation Error + Rounding Error

$$\text{FD} \quad f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \underbrace{\frac{1}{2} h f''(x_0)}_{\text{truncation error}} + \mathcal{O}(h^2)$$

Assume rounding error in $f(x)$ of $\varepsilon(x)$

$|\varepsilon(x)| < E$, constant E, true for $\forall x$

we are evaluating

$$\frac{F(x_0+h) - F(x_0)}{h} = \frac{f(x_0+h) - f(x_0)}{h} + \frac{\varepsilon(x_0+h) - f(x_0)}{h}$$

$$\text{Error:} \quad \left| f' - \frac{f(x_0+h) - f(x_0)}{h} \right| = \left| \frac{\varepsilon(x+h) - \varepsilon(x)}{h} - \frac{1}{2} h f''(x_0) + \mathcal{O}(h^2) \right|$$

$$\leq \left| \frac{\varepsilon(x+h)}{h} \right| + \left| \frac{\varepsilon(x)}{h} \right| + \left| \frac{1}{2} h f'' \right| + \mathcal{O}(h^2)$$

$$\left|\frac{\varepsilon(x+h)}{h}\right| + \left|\frac{\varepsilon(x)}{h}\right| + \left|\frac{1}{2}hf''\right| + O(h^2) \le \left.\underbrace{\frac{2\epsilon}{h}}_{\text{rounding}} + \underbrace{\frac{1}{2}hM + O(h^2)}_{\text{truncation}}\right\} e(h)$$

Assume $f'' < M, \forall x$

At 64 bit FP $\rightarrow$ $\varepsilon_{mach} = 10^{-16}$ $\quad \epsilon = 10^{-16} \quad M = 1$ $\rightarrow$ not massive $\rightarrow$ independent of h

Choose h such that h s.t uper bound is minimum

$$\frac{de}{dh} = 0 = -\frac{2\epsilon}{h^2} + \frac{M}{2} \qquad h^2 \rightarrow \frac{4\epsilon}{M} = \boxed{h = \frac{2\sqrt{\epsilon}}{\sqrt{M}}} \longrightarrow \text{optimum } h \text{ to minimize error.}$$

$$\frac{2\cdot 10^{-8}}{1}$$

Richardson Extrapolation

General technique for increasing the order of the truncation method.

Rule $D_h(f)$ with step h, with order N $\qquad$ α, β are constants, unknown

$$D_h(f) = f' + \alpha \cdot h^N + \beta h^{N+1} + O(h^{N+2})$$

Compare:
$$D_h(f) = f' + \alpha h^N + \beta h^{N+1} + O(h^{N+2})$$
$$D_{h/2}(f) = f' + \alpha \frac{h^N}{2^N} + \beta \frac{h^{N+1}}{2^{N+1}} + O(h^{N+2})$$

$$2^N D_{\frac{h}{2}}(f) - D_h(f) = (2^N - 1)\cdot f' + \gamma h^{N+1} + O(h^{N+2})$$

$$\boxed{\frac{2^N D_{\frac{h}{2}}(f) - D_h(f)}{2^N - 1} = f' + \underbrace{\gamma h^{N+1}}_{\text{truncation}} + O(h^{N+2})}$$

Example: FD

$$D_h(f) = \frac{f(x+h) - f(x)}{h} + O(h) \qquad \boxed{N = 1}$$

$$D_{h/2}(f) = \frac{f(x+\frac{h}{2}) - f(x)}{(h/2)} + O(h)$$

Using (*)

$$D'(f) = \frac{2D_{\frac{h}{2}} - D_h}{2^N - 1} = \frac{4\cdot f(x+\frac{h}{2}) - 3\cdot f(x) - f(x+h)}{h} = f' + O(h^2)$$

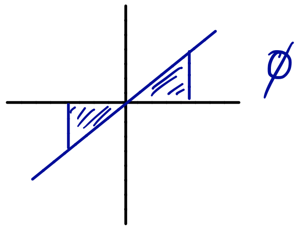
Module 4 Numerical Integration

$$I[f] = \int_a^b f(x)\,dx = \sum_{i=0}^{N} w_i\, f(x_i) = Q[f]$$

Degree of Precision (DoP)

Defn $Q[f]$ has DoP d:

- if $Q[p] = I[p] \quad \forall p \in \mathbb{P}^{d}$
- and $Q[q] \neq I[q] \quad \exists q \in \mathbb{P}^{d+1}$



Equivalent Q is exact $\forall\ a x^d + a x^{d-1} + \dots + a_0$

and Q is not exact for x^{d+1}

Linearity: $I[ap + bq] = a I[p] + b I[q]$

$Q[ap + bq] = a Q[p] + b Q[q]$

Integration conditions:

$$I[p] = Q[p]$$

- Q is exact for $x^0, \dots, x^d$

$p = 1 \qquad (b-a) = \int_a^b 1\,dx = \sum_{i=0}^{N} w_i$

$p = x \qquad \frac{b^2-a^2}{2} = \int_a^b x\,dx = \sum_{i=0}^{N} w_i x_i$

$p = x^2 \qquad \frac{b^3-a^3}{3} = \int_a^b x^2\,dx = \sum_{i=0}^{N} w_i x_i^2$

Linear system:

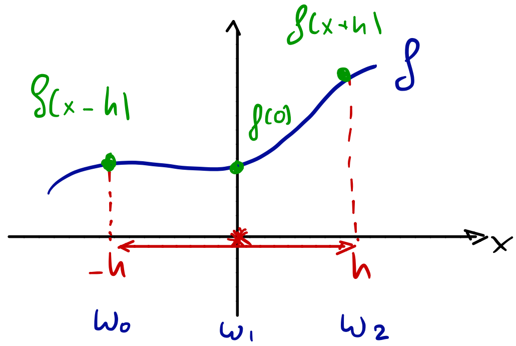
Vandermor $\longrightarrow$ invertible

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_0 & x_1 & \dots & x_N \\ x_0^2 & x_1^2 & \dots & x_N^2 \\ \vdots & \vdots & & \vdots \\ x_0^d & x_1^d & \dots & x_N^d \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} b-a \\ \frac{b^2-a^2}{2} \\ \frac{b^3-a^3}{3} \\ \frac{b^{d+1}-a^{d+1}}{d+1} \end{bmatrix}$$

Simpsons Rule

Aim for DoP 2.

$$Q[f] = w_0 f(-h) + w_1 f(0) + w_2 f(h)$$



$$\begin{array}{l} p=1 \\ p=x \\ p=x^2 \end{array} \begin{bmatrix} 1 & 1 & 1 \\ -h & 0 & h \\ h^2 & 0 & h^2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 2h \\ 0 \\ \frac{2h^3}{3} \end{bmatrix}$$

From below:

$$Q[f] = \frac{h}{3} f(-h) + \frac{4h}{3} f(0) + \frac{h}{3} f(h)$$

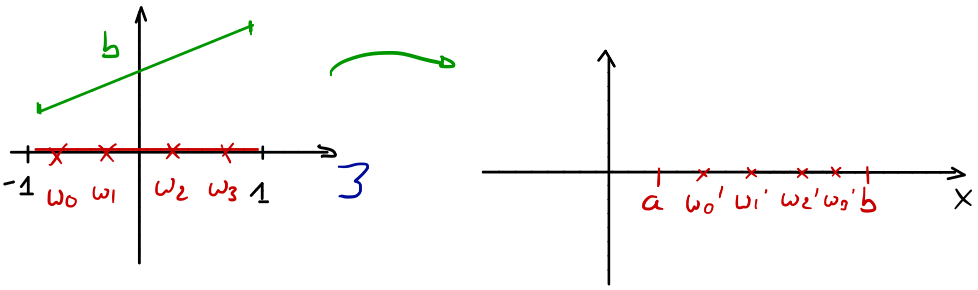
$\rightarrow w_0 = w_2$ (from $p = x$)

$(p = x^2) \Rightarrow h^2 w_0 + h^2 w_2 = \frac{2h^3}{3}$ $\qquad w_0 = w_2 = \frac{h}{3}$

$(p = 1) \Rightarrow w_1 = \frac{4h}{3}$

Interval Transformation

Unit-Space: $\qquad$ Physical space.

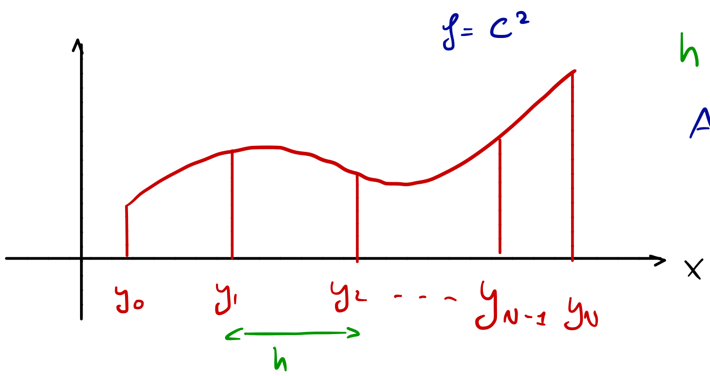


$$x = \left(\frac{a+b}{2}\right) + \left(\frac{b-a}{2}\right)\zeta \qquad \frac{dx}{d\zeta} = \frac{b-a}{2}$$

$$x_1 = (\qquad) + (\qquad)\zeta$$

$$I[f] = \int_a^b f(x)\,dx = \int_{-1}^{1} f(x(\zeta))\frac{dx}{d\zeta}d\zeta \simeq \left(\frac{b-a}{2}\right)\sum_{i=0}^{N} w_i f(x_i) = \sum_{i=0}^{N}\Big(\underbrace{\frac{b-a}{2}w_i}_{\boxed{w_i'}}\Big) f(x_i)$$

Composite Rules



$h = y_{i+1} - y_i$

Add up all the approximations of each interval.

devide into subinterval.

Reduce Error: by increasing d. Oscillation can be a problem. so beter not

$$I[f] = \int_a^b f(x)\,dx = \sum_{i=0}^{M-1}\int_{y_i}^{y_{i+1}} f(x)\,dx \simeq \underbrace{\sum_{i=0}^{M} Q[f; y_i, y_{i+1}]}$$

Composite rule — to not go to high degrees of polynomials.

Error | Behaviour with $h \to 0$ |

- Degree of $Q = d$
- Define $g_i(x) = f(y_i + x)$

Taylors expansion small, so "h".

$$\int_0^h g_i(x)\,dx = \int_0^h \left(g(0) + g'(0)x + \frac{1}{2}g''(0)x^2 + \dots + \frac{1}{d!}g^{d}(0)x^d\right)dx + \underbrace{\int_0^h \frac{1}{(d+1)!}g^{(d+1)}(\xi(x))\,x^{d+1}}_{\text{error term.}}\,dx$$

Error $\quad \mathcal{E}_i \leq A\int_0^h x^{d+1}\,dx = \frac{A}{d+2}h^{d+2} \sim \mathcal{O}(h^{d+2})$

Total error $\quad \boxed{\mathcal{E} = \sum_{i=0}^{M} \mathcal{E}_i \leq \frac{MA}{d+2}h^{d+2} + \mathcal{O}(h^{d+1}) \quad M \propto \frac{1}{h}}$

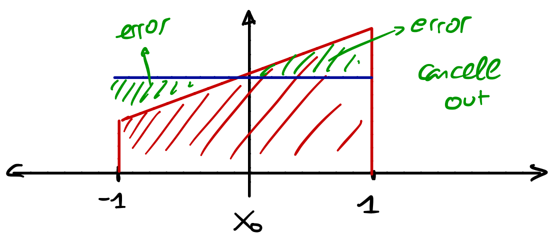
straight line.

TRAPEZOIDAL RULE
DoP : 1 Error: h^2

Gauss - Legendre Quadrature

Allow free choice of x_i, as well as w_i. DoF $\to$ $2(N+1)$

Solve DoP $\quad 2(N+1) - 1 = \boxed{2N+1}$



Do the same with different points.

$Q(1) = I(1)$

$Q(x) = I(x)$

$w_0 = 2$

$w_0 \cdot x_0 = 0$ ✓ $\boxed{x_0 = 0}$

$\boxed{Q_1[f] = 2f(0)}$

Quadrature rule 1

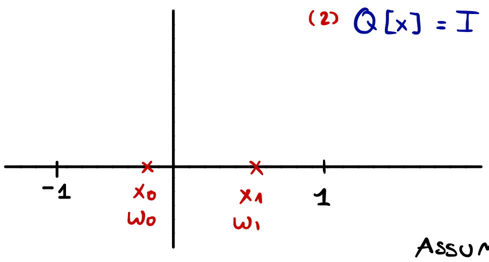
Exact $\int_{-1}^{1} ax + b\,dx = b \cdot 2$

$N = 0$

$N = 1$

(1) $Q[1] = I[1]$ (3) $Q[x^2] = I[x^2]$

(2) $Q[x] = I[x]$ (4) $Q[x^3] = I[x^3]$



(1) $w_0 \underbrace{f(x_0)}_{1} + w_1 \underbrace{f(x_1)}_{1} = 2$ ✓

(2) $w_0 x_0 + w_1 x_1 = 0$ ✓

(3) $w_0 x_0^2 + w_1 x_1^2 = 2/3$ ✓

(4) $w_0 x_0^3 + w_1 x_1^3 = 0$ ✓

NON LINEAR

NO STANDARD ALGEBRAIC TECHNIQUES.

USE NEWTONS METHOD

4 unknowns 4 equations.

Assume symmetry:

$w_0 = w_1 = 1$

$x_0 = -x_1$

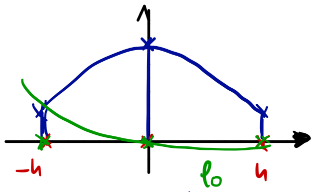
(3) $x_0^2 + x_0^2 = 2/3 = \boxed{\frac{1}{\sqrt{3}} = x_0}$

QUADRAURE RULE 2

$\boxed{Q_2[f] = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)}$

symmetry assumption result.

IDEA: POLYNOMIAL INTERPOLATION + INTEGRATE THE INTERPOLANT



$$Q[f] = \int_{-h}^{h} p(x)\,dx$$

Sympsons Rule:

Lagrange: $p(x) = \sum_{i=0}^{2} f_i \, \ell_i(x)$

$$Q[f] = \sum_{i=0}^{2} f_i \underbrace{\int_{-h}^{h} \ell_i(x)\,dx}_{\text{weight of node } w_i}$$

$$\ell_0(x) = \frac{(x)(x-h)}{2h^2}$$

$$w_0 = \int_{-h}^{h} \ell_0(x)\,dx = \frac{1}{2h}\left[\frac{1}{3}x^3 - \frac{h}{2}x^2\right]_{-h}^{h} = \boxed{\frac{h}{3}} \rightarrow \boxed{w_0 = w_2} \text{ same.}$$

MODULE 5: Numerical Solutions of ODEs

PROBLEM STATEMENT

Find $\underline{u}(t)$ satisfying $\frac{d\underline{u}}{dt} = \underline{f}(\underline{u})$ and $\underline{u}(0) = u_0$

takes some space u and maps it on to the derivative of u.

$\underline{u}$ is a function $u: \mathbb{R} \to \mathbb{R}^m$

$f: \mathbb{R}^n \to \mathbb{R}^m$

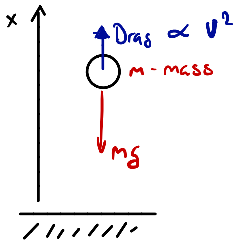
$$\boxed{\frac{du}{dt} = u'}$$

1-D case: $u' = f(u)$ $\quad u: \mathbb{R} \to \mathbb{R}$ $\quad f: \mathbb{R} \to \mathbb{R}$

Initial value problem, we know to and we want to know the end. Not boundary value.

- $g(u(0), u(T)) = \emptyset$ } NOT COVERED

EXAMPLE Body accelerating under gravity.



$$D = \frac{\overbrace{C \rho A}^{\hat{C}}}{2}(x')^2$$

NO AIR	AIR
$m x'' = -mg$	$m x' = -mg + \hat{C}(x')^2$
$u = \begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \end{pmatrix}$	$u = \begin{pmatrix} x \\ x' \end{pmatrix}$
$\cancel{m} u_1' = -\cancel{m}g$	$m u_1' = -mg + \hat{C} \cdot u_1^2$
$u_0' = u_1$	$u_0' = u_1$
$\underline{u}' = \underbrace{\begin{pmatrix} u_1 \\ -g \end{pmatrix}}_{f(\underline{u})}$	$\underline{u} = \begin{pmatrix} u_1 \\ -g + \frac{\hat{C}}{m} u_1^2 \end{pmatrix}$
$\underline{w} = f(\underline{u})$	$\underline{u}' = f(\underline{u})$

$\rightarrow$ transform to $\boxed{\frac{du}{dt} = u'}$

we need x and x'

Example Aerospace

Structural equations:

$$m h'' + S_\theta \cdot \theta'' + k_h \cdot h = -L(h', \theta)$$
$$S_\theta h'' + I_\theta \theta'' + k_\theta \cdot \theta = M(h', \theta)$$

S_θ = static mass moment

I_θ = mass moment of inertia

m = mass.

$$u = \begin{bmatrix} h \\ h' \\ \theta \\ \theta' \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

consider just h' and θ' $\boxed{u_0' = u_1}$ $\boxed{\theta_0' = \theta_1}$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & m & 0 & S_\theta \\ 0 & 0 & 1 & 0 \\ 0 & S_\theta & 0 & I_\theta \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}' + \begin{bmatrix} 0 & 1 & 0 & 0 \\ k_h & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & k_\theta & 0 \end{bmatrix} \bar{u} = \begin{bmatrix} 0 \\ -L \\ 0 \\ M \end{bmatrix}}_{u' = f(u)}$$

Non Autonomous ODEs

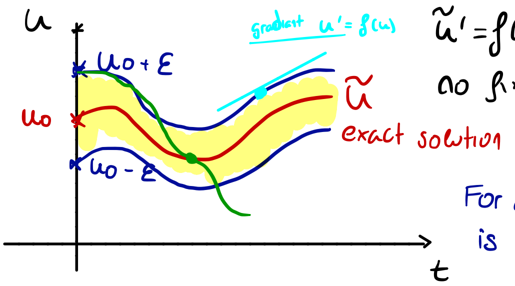
(Scalar) $u' = f(u, t)$ ⟶ rewrite it without a t. we can always do it.

explicit dependence on time.

Trick: fake variable:

$$\underline{u} = \begin{pmatrix} w_0 \\ u_1 \end{pmatrix} = \begin{pmatrix} u \\ u_1 \end{pmatrix} \qquad \begin{aligned} u_0' &= f(u_0, t) \longrightarrow u_0' = \underbrace{f(u_0, u_1)}_{\text{no dependence on } t.} \\ u_1' &= 1 = t \end{aligned}$$

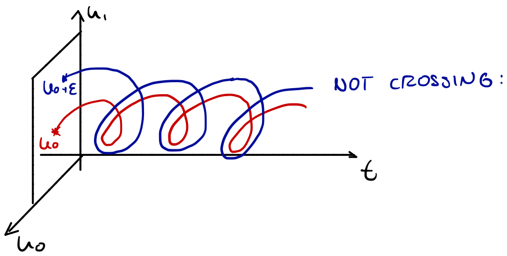
Graphically



$\tilde{u}' = f(u)$

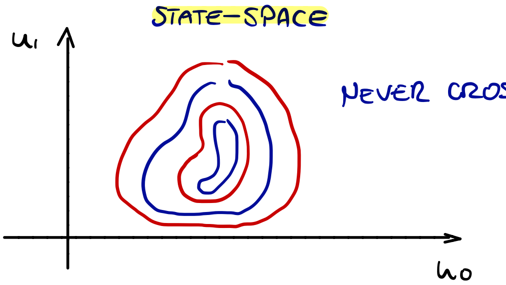
no fixed IC.

$u : \mathbb{R} \to \mathbb{R}^2$



For a scalar equation, this lines should never cross if they do the future is uncertain.

State-space

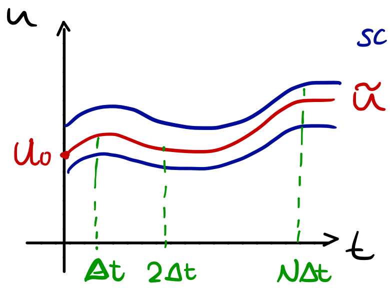


Never crossing.

Discretization of $u(t)$

- $u : \mathbb{R}^+ \to \mathbb{R}^M$ $\rightarrow$

SCALAR SYSTEM



$\tilde{u} = f(\tilde{u})$ (1)

Discretize solution:

$$\tilde{u} = \begin{bmatrix} \tilde{u}(0) \\ \tilde{u}(\Delta t) \\ \vdots \\ \tilde{u}(N\Delta t) \end{bmatrix} \in \mathbb{R}^{N+1} \quad (b/c \ u : \mathbb{R} \to \mathbb{R})$$

$\tilde{u}$ $\downarrow$ solution at time instant.

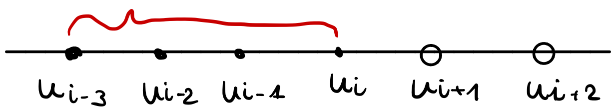
If $u : \mathbb{R} \to \mathbb{R}^M$ $\quad \tilde{u} \in \mathbb{R}^{(N+1)M}$

Reduced an infinite dimension $u \longrightarrow$ finite-dimension vector.

Numerical Method to find u_i

- Aiming to approximate $\tilde{u}(i\Delta t) \simeq u_i$
- Assume past is known but future is unknown $\quad u_0, u_1, \dots, u_i$ KNOWN
  $u_{i+1}, \dots, u_N$ UNKNOWN

predict the future using this

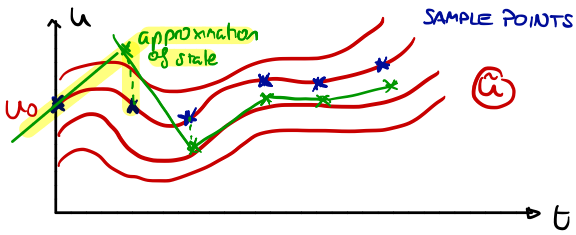


Find u_{i+1} in terms of $u_i, u_{i-1} \dots$

$$u'(t_i) \simeq \frac{u_{i+1} - u_i}{dt}$$

$\longrightarrow$ $f(u_i)$ from equation (1)

$u_{i+1} = u_i + \Delta t \cdot f(u_i)$ FORWARD EULER



$\tilde{u} = f(\tilde{u})$

F-Euler approximation of $\tilde{u}$ $\quad$ DOES NOT GOOD / VERY GOOD. $\}$ decrease Δt to improve

Error in F-E

Taylor expansion:

(+) $u'(t_i) = \dfrac{u_{i+1} - u_i}{\Delta t} + \varepsilon = \dfrac{u(t_i + \Delta t) - u(t_i)}{\Delta t} + \varepsilon$ $\quad$ $\longrightarrow$ taylor.

(#) $u(t_i + \Delta t) = u(t_i) + u'(t_i)\Delta t + \frac{1}{2} u''(\xi) \cdot \Delta t^2 \quad \xi \in [t_i, t_{i+1}]$

$$\varepsilon = -\frac{1}{2} u''(\xi) \cdot \Delta t$$

Rearrange (#) $\to$ (+)

Next step

$u_i' = f(u_i)$ $\qquad \dfrac{u_{i+1} - u_i}{\Delta t} + \varepsilon = f(u_i)$ $\qquad u_{i+1} = u_i + \Delta t f(u_i) - \underbrace{\Delta t \cdot \varepsilon}_{\text{error at each time step}}$

$\longrightarrow$ Proportional to $O(\Delta t^2)$

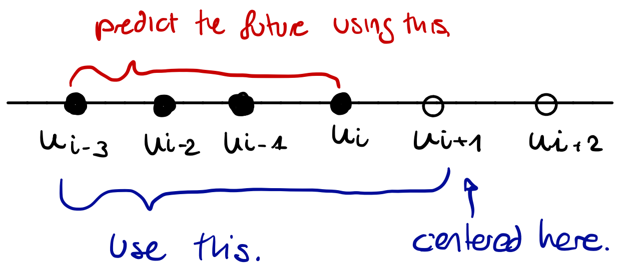
Error in u_N at $t = T$

Take N steps, each step error $\sim \mathcal{O}(\Delta t^2)$

$\varepsilon_T \sim N \cdot \mathcal{O}(\Delta t^2)$ $\quad$ N depends on $\Delta t \longrightarrow N \propto \frac{1}{\Delta t}$

$\sim \mathcal{O}(\Delta t) \longrightarrow$ F-E is a first order accurate method.

Backward Euler



$$f(u_{i+1}) \quad u'_{i+1} = \frac{u_{i+1} - u_i}{\Delta t} + \mathcal{O}(\Delta t)$$

Rearrange:

BACKWARD EULER

$$\boxed{u_{i+1} = u_i + \Delta t \, f(u_{i+1})}$$

How to use method:

- Assume f linear: $f(u) = Au + B$

$$u_{i+1} = u_i + \Delta t \cdot [A u_{i+1} + B] \qquad \underbrace{[I - \Delta t \cdot A]}_{J} u_{i+1} = u_i + \Delta t B$$

If J invertible

$$\boxed{u_{i+1} = \dots}$$

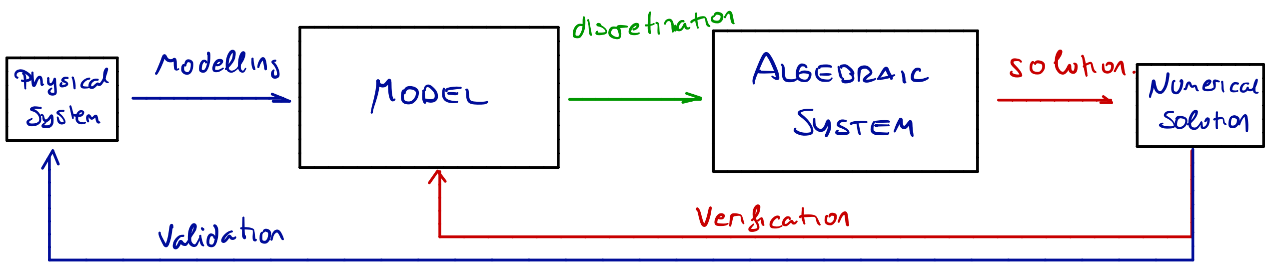
- Use root-finding at each time-step.
- Linearize f

$$f(u_{i+1}) = f(u_i + \Delta u) = f(u_i) + \underbrace{f'(u_i)}_{\text{Jacobian}} \Delta u + \cancel{\mathcal{O}(\Delta u^2)}$$

error O to linearize

Computational Modelling

Process



MODELLING

Uncertainties

Epistemic: Due to unrecognized or unanticipated behaviour.

Aleatory: Impractical to measure.

Model Errors

Due to the use of simplified equations, linearization and sub-models for difficult to compute phenomena.

DISCRETIZATION

Take a continuous function and make it discrete. Solve for discrete function it is easier for computers.

Discretization Error

Taylor: $$u_{i+1} = u_i + \Delta x \left.\frac{du}{dx}\right| + \underbrace{\frac{\Delta x^2}{2} \left.\frac{du}{dx}\right| + \dots}_{\text{discretization error.}}$$

SOLUTION

Iteration and Round-off errors.

Linear iteration error: error due to halting of the iterative procedure for the solution of a linear algebraic system. (< tol2)

Non-linear iteration error: error due to halting of the iterative solution of the non-linear problem (< tol1)

VERIFICATION

TYPE 1 Establishes consistency with the PDE + BCs done using exact solution.

TYPE 2 Provides error estimate. done with grid study.

VALIDATION

Stablishes the magnitude of model errors and uncertainties.

REQUIREMENTS

CONVERGENT: If when running the code if i increase the nodes, i get more close to the solution.

- Consistency: When substituting an exact solution into the discrete equations the only terms that remain are these that tend to 0.
- Stability: numerical solution is unique. Small changes to the input produce only small changes to the numerical solution.

It is possible to have a consistent algorithm which is not stable.

ACCURACY: It is possible to design methods with a certain order of accuracy.

AFFORDABILITY: Some solution methods are much faster than others.

Computers are not better but methods are.

PARTIAL DIFFERENTIAL EQUATIONS

HEAT EQUATION

$$\frac{\partial u}{\partial t} = V \cdot \frac{\partial^2 u}{\partial x^2}$$

CONVECTION EQUATION

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

- Unnattenuated
-

ADVECTION-DIFFUSSION EQUATION

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = V \cdot \frac{\partial^2 u}{\partial x^2}$$

- Combination of two previous
-

SECOND ORDER WAVE EQUATION

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$