Roots of Polynomials

$P_N(x) = 0$, $P_N(x)$: N-th degree polynomial

$N = 0, 1, 2, 3, 4 \rightarrow$ explicit (complicated) formulas

$N > 4 \rightarrow$ no formula exists

Trick: Use <u>linear algebra</u> to find

eigenvalues of a matrix with characteristic

polynomial $P_N(x)$

reminder: $A\vec{x} = \lambda\vec{x} \Leftarrow \underbrace{\det(A - \lambda \mathbb{1})}_{= \chi(\lambda)} = 0$

eigenvector ($\vec{x}$), eigenvalue (λ)

characteristic polynomial)

Companion matrix

$$P_N(x) = x^N + c_{N-1}x^{N-1} + \dots + c_1 x + c_0$$

$$C_p = \begin{pmatrix} 0 & \cdots & & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & 0 \cdots & 0 & -c_2 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & -c_{N-1} \end{pmatrix}$$

$$\det(C_p - x\,\mathbb{1})$$

$$= \begin{vmatrix} -x & 0 & \cdots & & 0 & -c_0 \\ 1 & -x & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & -x & 0 \;\cdots & 0 & -c_2 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ & & \ddots & 1 & -x & -c_{N-2} \\ 0 & \cdots & & 0 \quad 1 & -x & -c_{N-1} \end{vmatrix}$$

welop bottom row

$$)(-x - c_{N-1}) \underbrace{\begin{vmatrix} -x & 0 & \cdots & & 0 \\ 1 & -x & 0 & \cdots & 0 \\ 0 & & \ddots & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -x \end{vmatrix}}_{= (-x)^{N-1}} \quad \begin{matrix} -c_0 \\ -c_1 \end{matrix}$$

$$-1)^N \begin{vmatrix} -x & 0 & \cdots & 0 & -c_0 \\ 1 & -x & & & -c_1 \\ 0 & & \ddots & & \vdots \\ \vdots & & & & \\ 0 & & & & -c_{N-2} \end{vmatrix}$$

$$x^N + x^{N-1} c_{N-1} + (-1)^N \begin{vmatrix} -x & 0 & \cdots & 0 & -c_0 \\ 1 & -x & & & \vdots \\ 0 & & \ddots & & \\ \vdots & & & -x & -c_{N-3} \\ 0 & \cdots & 0 & 1 & -c_{N-2} \end{vmatrix}$$

$$^N + x^{N-1} c_{N-1} + (-1)^N (-1)^{N-2} c_{N-2} (-x)^{N-2} + (-1)^{N-3}$$

$$\times \begin{vmatrix} -x & 0 & \cdots & 0 & -c_0 \\ 1 & -x & 0 & \cdots & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & \ddots & -x & -c_{N-4} \\ 0 & \cdots & 0 & 1 & -c_{N-3} \end{vmatrix}$$

$$= x^N + x^{N-1} C_{N-1} + x^{N-2} C_{N-2} + \cdots$$

$$+ C_0 \quad \checkmark$$

Numerical Optimization

similar but distinct from root finding

given $f(\vec{x})$

find $\vec{x}^*$ such that $f(\vec{x}^*)$

is minimal / maximal

asic distinction: derivative-free vs. derivative methods

erivative-free method in 1D

ecall bisection search

$$[a_0, b_0] \to [a_1, b_1] \to \dots \to [a_n, b_n]$$

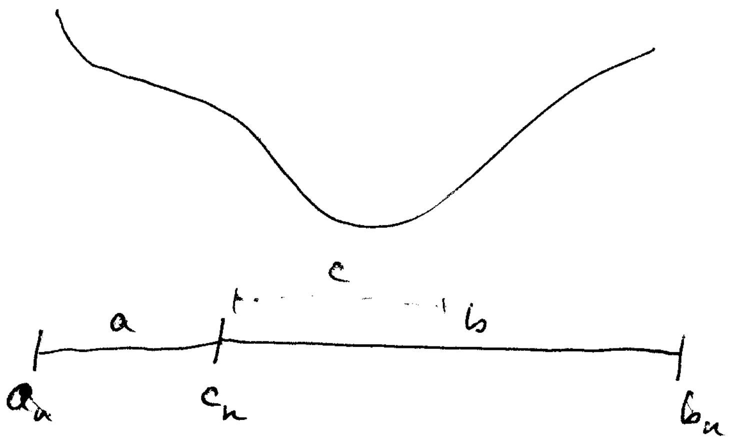
ith $\mathrm{sign}(f(a_n)) \neq \mathrm{sign}(f(b_n))$ always

olden Section Search is similar but uses triples

$$[a_n, c_n, b_n] \quad , \quad a_n < c_n < b_n$$

$$(*) \qquad f(a_n) > f(c_n) \quad , \quad f(c_n) < f(b_n)$$

Pick ratio between intervals lengths such that it remains constant always:



want $\frac{a}{b} = \frac{c}{b-c}$ and $\frac{c}{a} = \frac{a}{b}$

$$\rightarrow \quad \left(\frac{b}{a}\right)^2 - \frac{b}{a} = 1$$

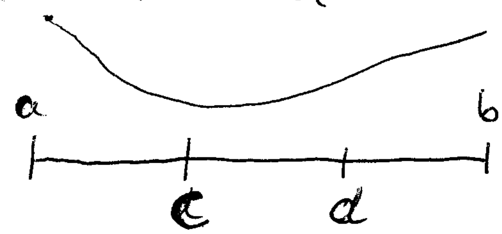
$$\rightarrow \quad \frac{b}{a} = \phi = \frac{1+\sqrt{5}}{2} \Rightarrow \frac{a}{b} = \gamma = \frac{2}{1+\sqrt{5}}$$

(golden ratio) ≈ 0.618

$\rightarrow$ ratio of new intervals always identical.

lgorithm:

tart from (a_0, b_0) bracketing a minimum



1. compute $c = b_n - \gamma(b_n - a_n)$

$$d = a_n + \gamma(b_n - a_n)$$

(probe points)

. if $f(c) < f(d)$

set the next bracket

$$a_{n+1} = a_n$$
$$b_{n+1} = d$$

(minimum must be near c)

else, $f(d) < f(c)$

set the next bracket

$$a_{n+1} = c$$
$$b_{n+1} = b$$

(minimum near d)

terate and finally estimat $x_{min} \approx \frac{1}{2}(a+b)$

Derivative method in 1D

Extrema of $f(x)$ are characterized

by $f'(x) = 0$

-> can apply a root-finding method to this.

-> Ex: Newton's method

$$x_{n+1} = x_n - \frac{f'(x)}{f''(x)}$$

equires second derivative, can be expensive

SET: method of gradient descent

$$x_{n+1} = x_n - \alpha_n f'(x)$$

n-D:

$$\vec{x}_{n+1} = \vec{x}_n - \underbrace{H^{-1}(\vec{x}_n)}_{\text{Hessian matrix}} \overbrace{\vec{\nabla} f(\vec{x}_n)}^{\text{gradient}}$$

computing / storing Hessian is often very computationally intensive

→ "Quasi-Newton" methods approximate either $H(\vec{x})$ or the product $H^{-1}\vec{\nabla} f$.

(BFGS, ...)

→ another option is to go down the gradient and optimize the step size by a line search

$$\min_{\alpha} f(\vec{x} - \alpha \vec{\nabla} f(x))$$

often, α is found approximately

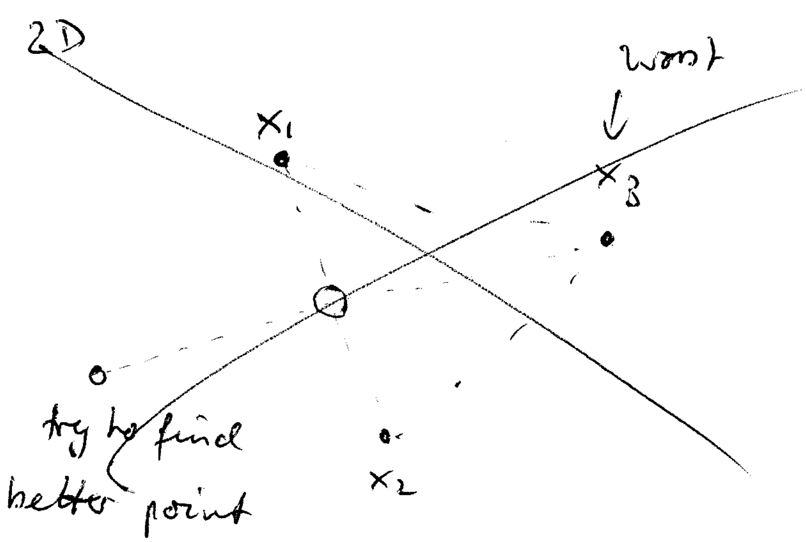
(ex: Conjugate-gradients, gradient descent)

Derivative - free methods in n-D

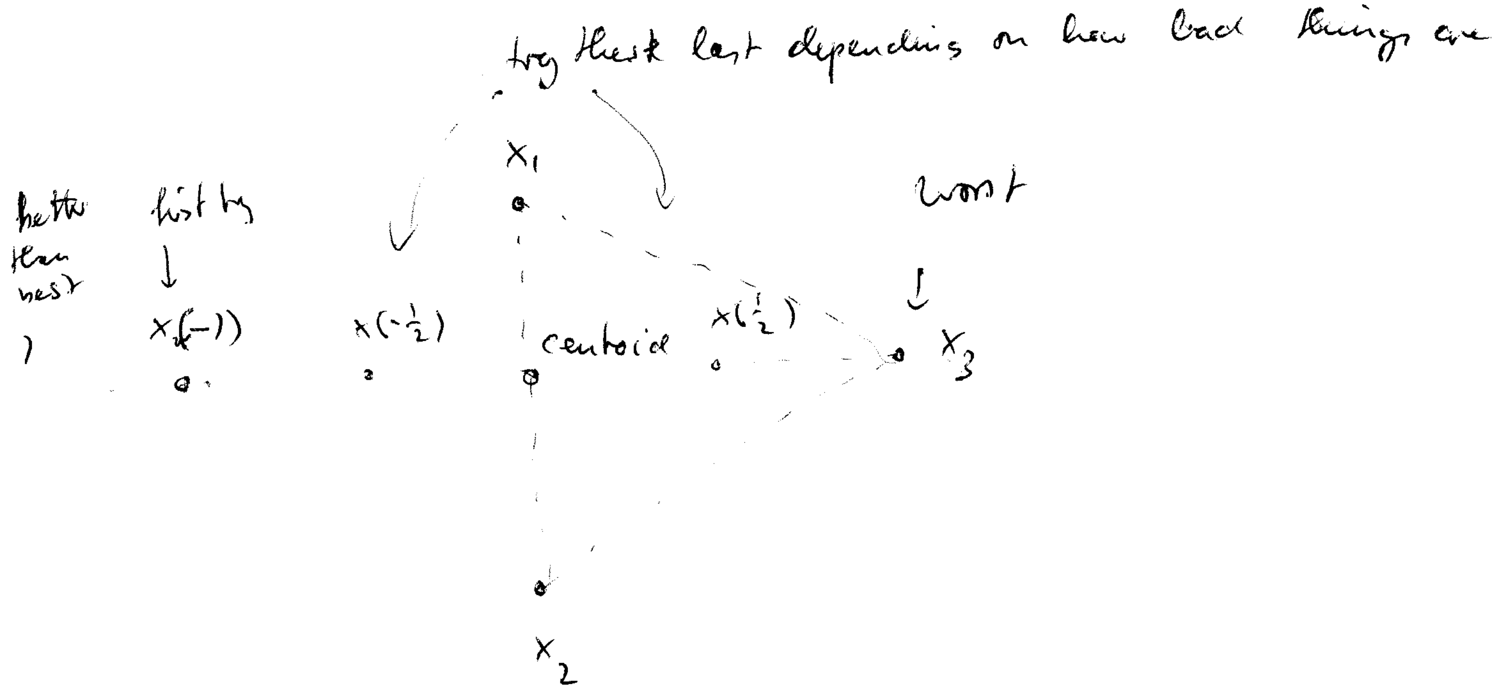
Nelder - Mead / Amoeba method

Idea:

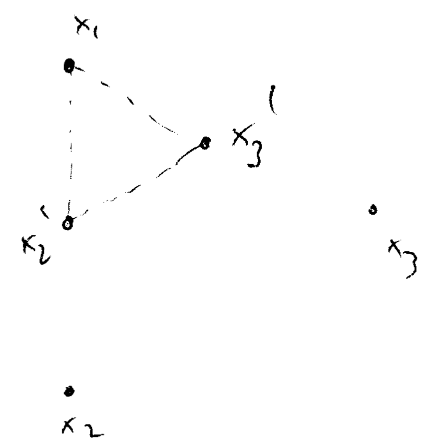
- sample $f(\vec{x})$ at $N+1$ points in N-D $\rightarrow$ define a simplex
- order by $f(\vec{x}_{N+1}) \geq f(\vec{x}_N) \geq \dots \geq f(\vec{x}_1)$
- decrease size of simplex by successively removing worst point ~~the~~ $\vec{x}_{N+1}$ and replacing by a better estimate found by reflecting $\vec{x}_{N+1}$ ~~on the~~ along centroid of other points



successively try



or if nothing works, shrink



relatively simple,

$O(N)$ function evaluation at each step

some convergence theory

usually linear rate of convergence