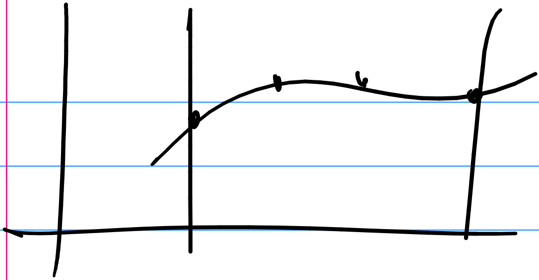
Overview

nd solve

opt

1D

nD



$$V\vec{\alpha} = \vec{f}$$

$$\tilde{f}(x) = \alpha_0 \varphi_0(x) + \cdots + \alpha_{n-1}\varphi_{n-1}(x)$$

$$\int \tilde{f}(x)\,dx = \alpha_0 \int \varphi_0 + \cdots + \alpha_{n-1}\int \varphi_{n-1}$$

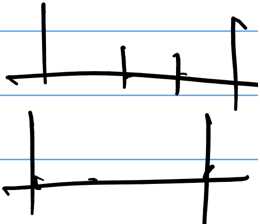
$$\vec{I} = \begin{pmatrix} \int \varphi_0 \\ \vdots \\ \int \varphi_{n-1} \end{pmatrix}$$

$$I^T \left(V^{-1} \vec{f} \right) \leftarrow n^2/n^3$$

$$= \underbrace{\left(I^T V^{-1} \right)}_{\text{row}} \vec{f} \leftarrow n$$

$\vec{w}$

precompute!



Solving Nonlinear Equations

What is the goal here?

Demo: Three quadratic functions (click to visit)

Newton's method

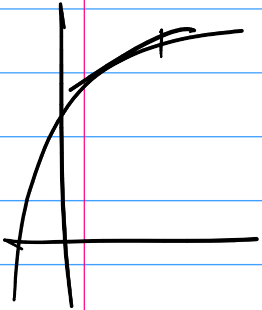
What does Newton's method look like in n dimensions?

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \leftarrow$$

$$f\begin{pmatrix} a \\ ? \end{pmatrix} = \begin{pmatrix} <0 \\ \end{pmatrix} \leftarrow$$

$$f\begin{pmatrix} b \\ ? \end{pmatrix} = \begin{pmatrix} >0 \\ \end{pmatrix}$$

$$\vec{x}_{k+1} = \vec{x}_k + \vec{h}$$



$$\vec{f}(\vec{x}_k + \vec{h}) = \vec{f}(\vec{x}_k) + \begin{pmatrix} \partial f_1/\partial x_1 & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \end{pmatrix} \vec{h}$$

says "how much does f_n change if h_1 changes

$$\vec{0} = \vec{f}(\vec{x}_k + \vec{h}) \approx \vec{f}(\vec{x}_k) + J_f(\vec{x}_k) \cdot \vec{h}$$

$$\vec{0} = \vec{f}(\vec{x}_k) + J_f(x_k) \cdot \vec{h}$$

$$J_f^{-1}(x_k) \cdot \left(-\vec{f}(\vec{x}_k)\right) = \vec{h}$$

$$\vec{x}_{k+1} = \vec{x}_k - J^{-1}(x_k) \cdot f(x_k)$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton: Example

Set up Newton's method to find a root of

$$\mathbf{f}(x, y) = \begin{pmatrix} x + 2y - 2 \\ x^2 + 4y^2 - 4 \end{pmatrix} \doteq 0$$

$\partial f_1/\partial x$ $\partial f_1/\partial y$

Demo: Newton's method in n dimensions (click to visit)

$$J_f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2x & 8y \end{pmatrix}$$

$\mathcal{J}_{\varphi}$ ($\swarrow$) $\leftarrow$ 1x1 1D

n

$\mathcal{J}_{\varphi} = n$ $\underbrace{\begin{pmatrix} & \\ & \end{pmatrix}}_{n \to 1000}$ nD

Secant in n dimensions?

What would the secant method look like in n dimensions?

It doesn't.

$\leftarrow\quad f(x_k) \qquad f(x_{k+1}) \rightarrow O(n)$

$J_f(x_k) \rightarrow O(n^2)$

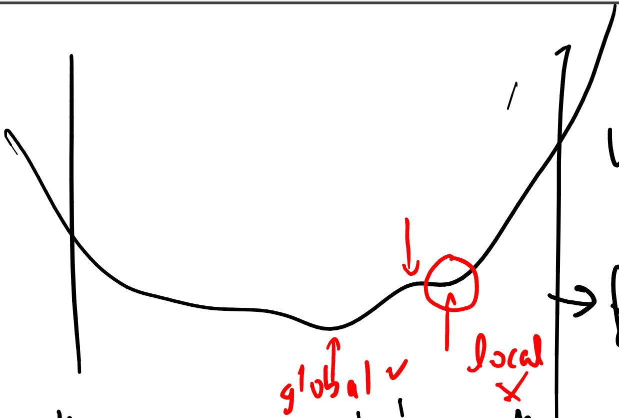
$$f(x) = (x-15)(x-10)(x-5)$$

$$g(x) = \frac{f(x)}{(x-15)}$$

Outline

Optimization

State the problem.



"Find the minimum"

Simplest Idea: Use Newton to solve: $f'(x) = 0$

Necessary condition

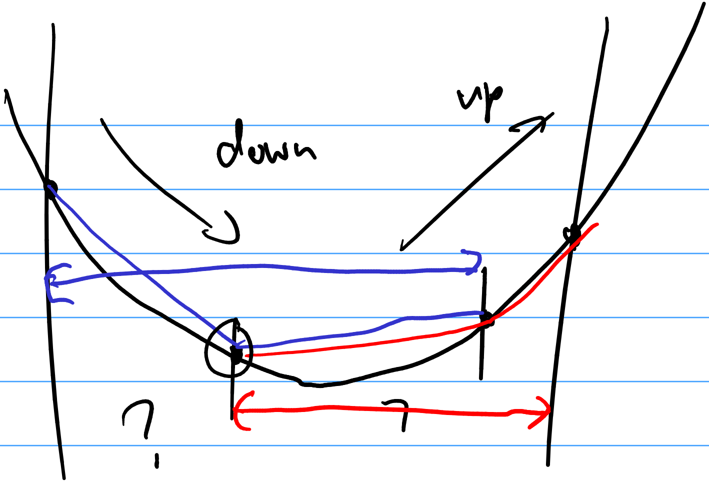
Sufficient cond: $f''(x) > 0$

Newton for 1D opt?

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

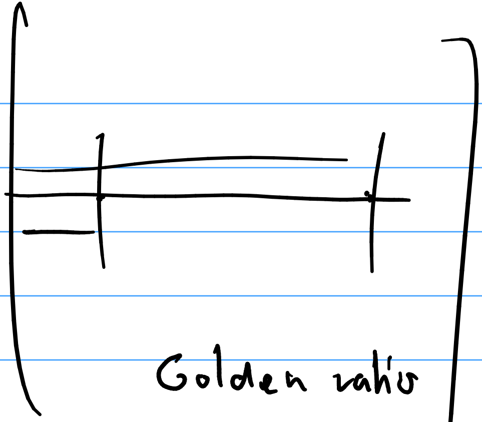
still quadratically convergent.

Um, no.



"unimodal"

$\frac{1}{3}$ $\frac{?}{3}$



Golden ratio

Golden section

Golden section search

Optimization: What could go wrong?

What are some potential problems in optimization?

doesn't have
min

- local minimum

Optimization: What is a solution?

How can we tell that we have a (at least local) minimum? (Remember calculus!)

Newton's Method

Let's steal the idea from Newton's method for equation solving:
Build a simple version of f and minimize that.

Demo: Newton's Method in 1D (click to visit)

In-class activity: Optimization Methods

Golden Section Search

Would like a method like bisection, but for optimization.
In general: No invariant that can be preserved.
Need *extra assumption.*

Demo: Golden Section Search Proportions (click to visit)

Outline

Optimization in n dimensions: What is a solution?

How can we tell that we have a (at least local) minimum? (Remember calculus!)

Find $\min f(x, y)$

Steepest Descent

Given a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $\boldsymbol{x}$, which way is down?