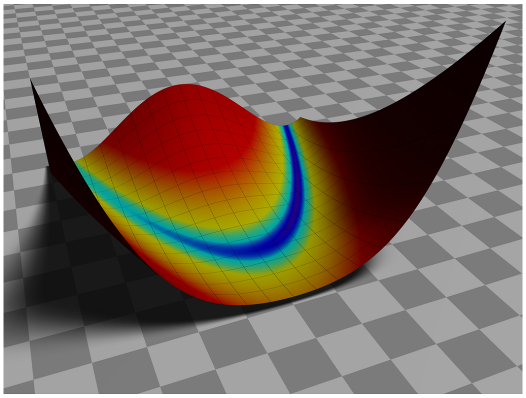
# Line Search 2

Lecture 3



ME EN 575
Andrew Ning
aning@byu.edu

## Outline

1D Optimization Methods

Inexact Line Search

# Golden Section Search

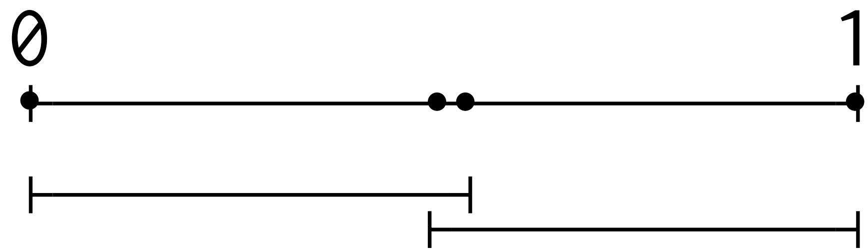
Consider methods based on minimization directly, rather than on root finding. Let's start with a bracketing approach.

In root finding, a bracket only requires two points, but in minimization it requires three. Why?

Dividing the interval in half resulting in the fastest approach to root finding (bisection), what about for minimization?

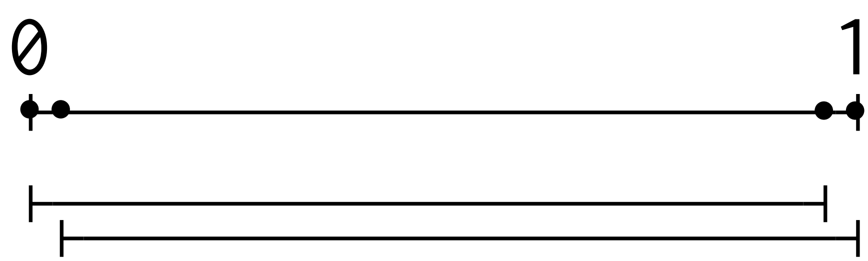
Remember, we want to use as few function calls as possible. Four points is the minimum to allow division into two regions. Why?

Extreme 1: bisection-like

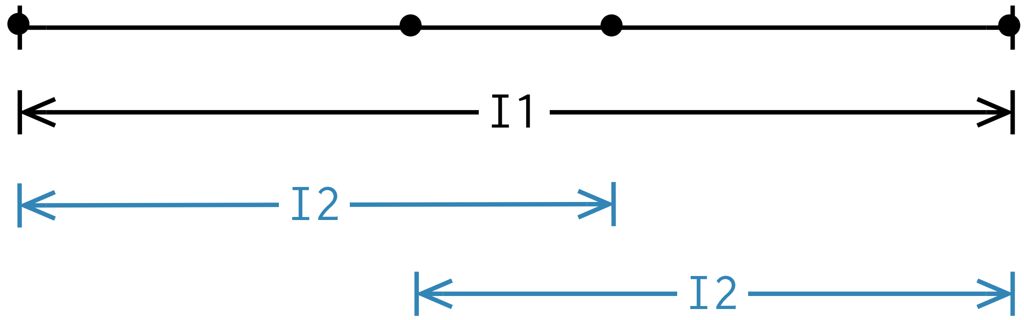



reduces interval significantly, but wastes function calls

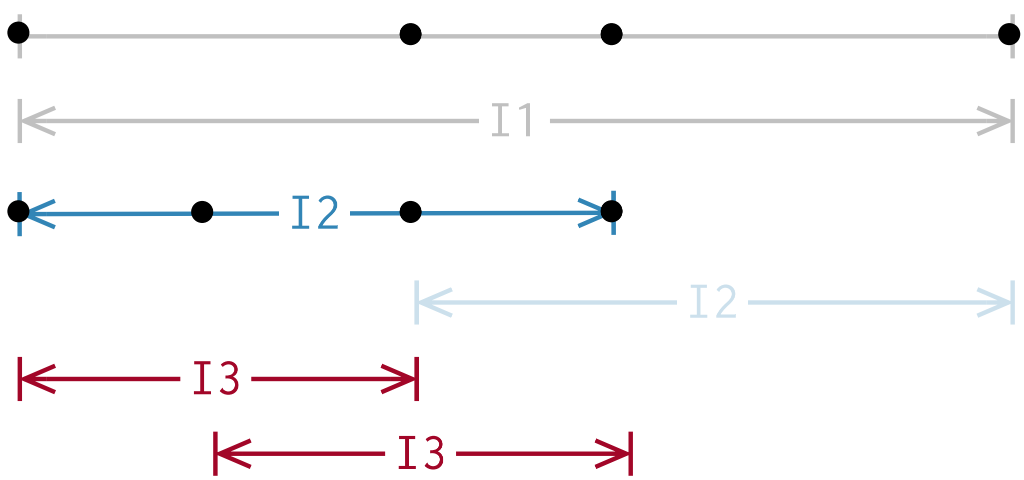
Extreme 2: small improvement




reusable function call, but incremental improvement




$$I_2 = \tau I_1$$




$$I_3 = \tau I_2$$

$$
\begin{aligned}
I_1 &= I_2 + I_3 \\
I_1 &= \tau I_1 + \tau^2 I_1 \\
1 &= \tau^2 + \tau
\end{aligned}
$$

Solution:

$$
\tau = \frac{\sqrt{5} - 1}{2} = 0.618\ldots
$$

(inverse of golden ratio)

# Polynomial Methods

Approximate function locally as a polynomial (in this case quadratic):

$$
\tilde{f} = \frac{1}{2} a x^2 + b x + c
$$

If $a > 0$, the minimum of this function is $x^* = -b/a$.

Taylor's series:

$$\tilde{f}(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$$

$$\begin{aligned} x^* &= -b/a \\ \Rightarrow x_{k+1} - x_k &= -\frac{f'(x_k)}{f''(x_k)} \\ \Rightarrow &\boxed{x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}} \end{aligned}$$

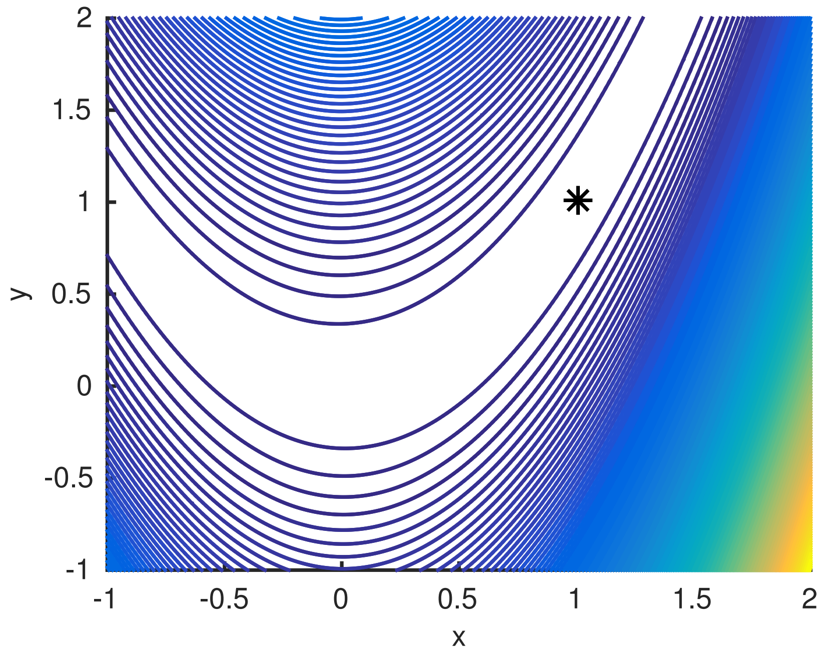
Identical to Newton's method used to find a root (assuming a quadratic polynomial).

# Brent's Method

Combines quadratic polynomial method with golden section search.

# Inexact Line Search

Recall the goals of line search:




At iteration k, assume we are at some point $x_k$, and are given a search direction $p_k$. How far should we go. In other words, we need to choose $\alpha_k$ such that

$$x_{k+1} = x_k + \alpha_k p_k$$

Goals and requirements:

- Funciton must decrease (guaranteed by choosing $p_k$ appropriately).
- Function must decrease enough.
- Use as few function calls as possible.
- Larger $\alpha$ is preferable, but not too large.

## Wolfe Conditions

Sufficient decrease (don't waste time iterating, prevent too large of a step):

$$f(x_k + \alpha p_k) \leq f(x_k) + \mu_1 \alpha g_k^T p_k$$

Typical value: $\mu_1 = 10^{-4}$

Intuition: $\alpha g_k^T p_k$ is the expected decrease. If we get even a small fraction of expected decrease, we will take it and move on. Also if our step size is large we better get a larger decrease (prevent steps that are too large).

Curvature condition (not too small of a step):

$$g(x_k + \alpha p_k)^T p_k \geq \mu_2 g_k^T p_k$$

Typical value: $\mu_2 = 0.9$

Intuition: Slope at new point must be less negative (if not we've taken a step that is too small and are leaving too much on the table).

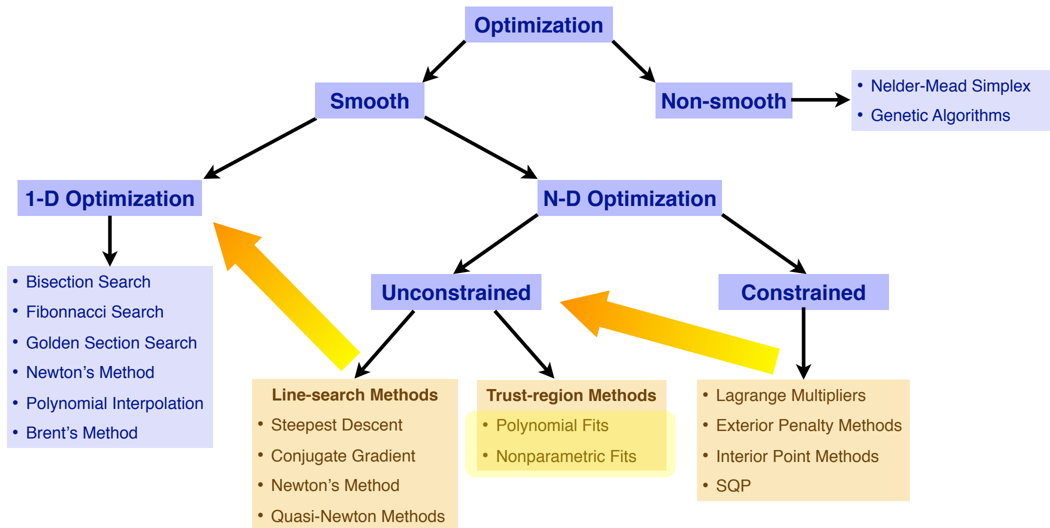
## Backtracking line search

Ridiculously simple (but of course not the most effective):

1. Check sufficient decrease.
   $f(x_k + \alpha p_k) \leq f(x_k) + \mu_1 \alpha g_k^T p_k$
2. Decrease step size. $\alpha = \rho\alpha$

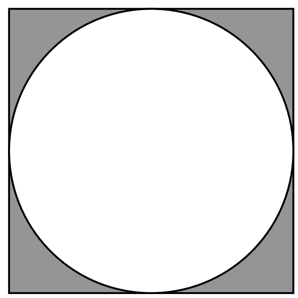
Better line searches combine 1D minimization methods with backtracking while enforcing the Wolfe conditions.

# Where are we going?

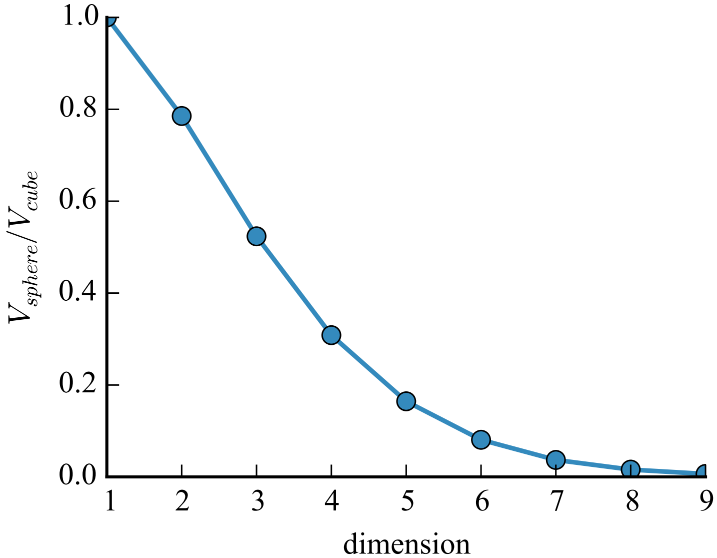



# Intuition in Higher Dimensions

Consider a hypersphere inscribed inside a hypercube



volume of sphere ? volume of cube


1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
dimension
$V_{sphere}/V_{cube}$