

CS 357: Numerical Methods

Optimization

Eric Shaffer

Optimization Problems

- Given function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, and set $S \subseteq \mathbb{R}^n$, find $x^* \in S$ such that $f(x^*) \leq f(x)$ for all $x \in S$
- x^* is called *minimizer* or *minimum* of f
- It suffices to consider only minimization, since maximum of f is minimum of $-f$
- *Objective* function f is usually differentiable, and may be linear or nonlinear
- *Constraint* set S is defined by system of equations and inequalities, which may be linear or nonlinear
- Points $x \in S$ are called *feasible* points
- If $S = \mathbb{R}^n$, problem is *unconstrained*

Optimization Problems

- General continuous optimization problem:

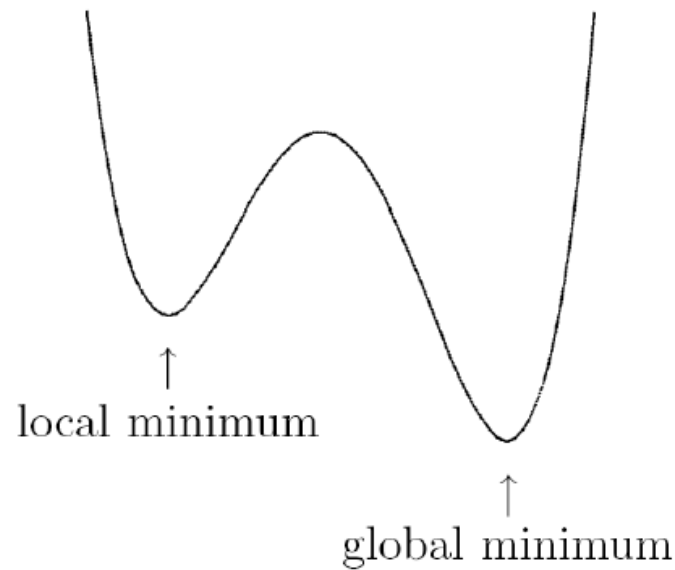
$$\min f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \quad \text{and} \quad \mathbf{h}(\mathbf{x}) \leq \mathbf{0}$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{h}: \mathbb{R}^n \rightarrow \mathbb{R}^p$

- *Linear programming*: f , \mathbf{g} , and \mathbf{h} are all linear
- *Nonlinear programming*: at least one of f , \mathbf{g} , and \mathbf{h} is nonlinear

Global versus Local Minimum

- $x^* \in S$ is *global minimum* if $f(x^*) \leq f(x)$ for all $x \in S$
- $x^* \in S$ is *local minimum* if $f(x^*) \leq f(x)$ for all feasible x in some neighborhood of x^*



Global Optimization

- Finding, or even verifying, global minimum is difficult, in general
- Most optimization methods are designed to find local minimum, which may or may not be global minimum
- If global minimum is desired, one can try several widely separated starting points and see if all produce same result
- For some problems, such as linear programming, global optimization is more tractable



First Order Optimality Condition

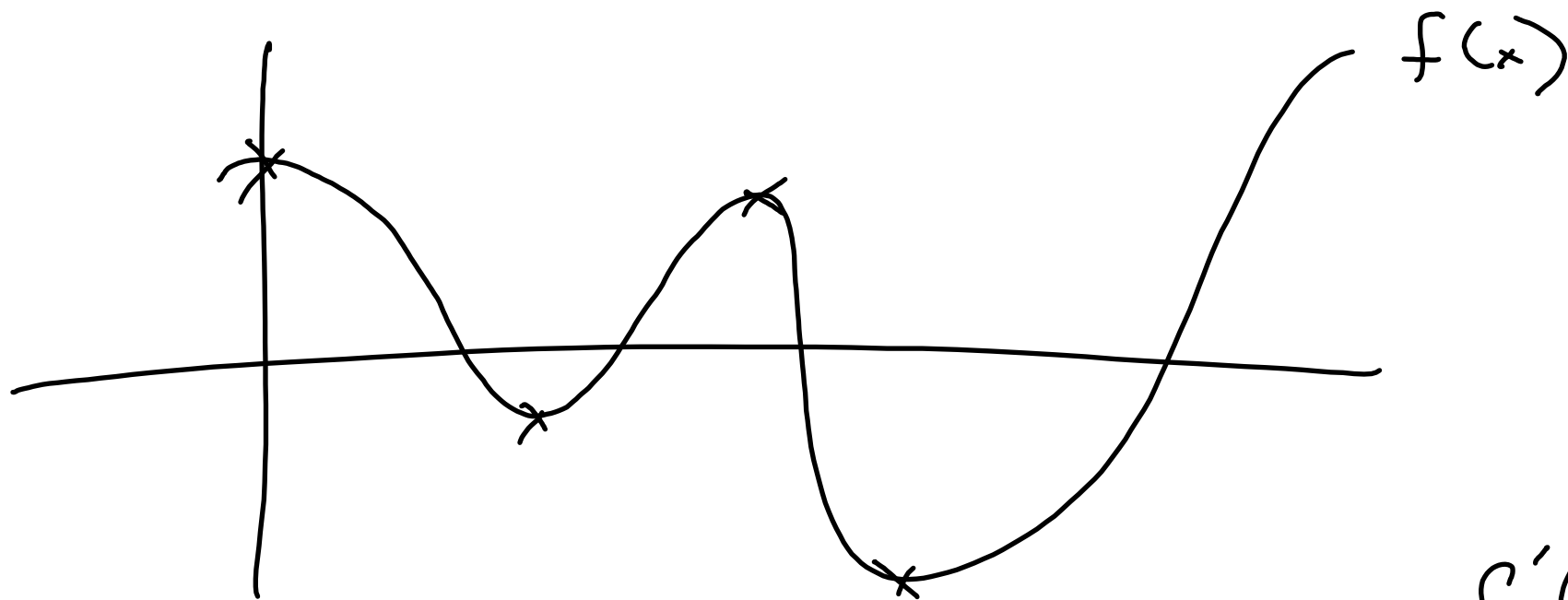
- For function of one variable, one can find extremum by differentiating function and setting derivative to zero
- Generalization to function of n variables is to find *critical point*, i.e., solution of nonlinear system $f(x_1, x_2)$

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

$$\nabla f(\mathbf{x}) = \left\langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right\rangle$$

where $\nabla f(\mathbf{x})$ is *gradient* vector of f , whose i th component is $\partial f(\mathbf{x})/\partial x_i$

- For continuously differentiable $f: S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, any interior point \mathbf{x}^* of S at which f has local minimum must be critical point of f
- But not all critical points are minima: they can also be maxima or saddle points



$f'(x) = 0$
 $\rightarrow x$
 critical
) necessary
 condition of
 $x = \text{min}$

sufficient conditions $\left\{ \begin{array}{l} f'(x) = 0 \\ f''(x) > 0 \end{array} \right.$


Second Order Optimality Condition

- For twice continuously differentiable $f: S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, we can distinguish among critical points by considering *Hessian matrix* $\mathbf{H}_f(x)$ defined by

$$\{\mathbf{H}_f(x)\}_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

which is symmetric

- At critical point x^* , if $\mathbf{H}_f(x^*)$ is
 - positive definite, then x^* is minimum of f
 - negative definite, then x^* is maximum of f
 - indefinite, then x^* is saddle point of f
 - singular, then various pathological situations are possible

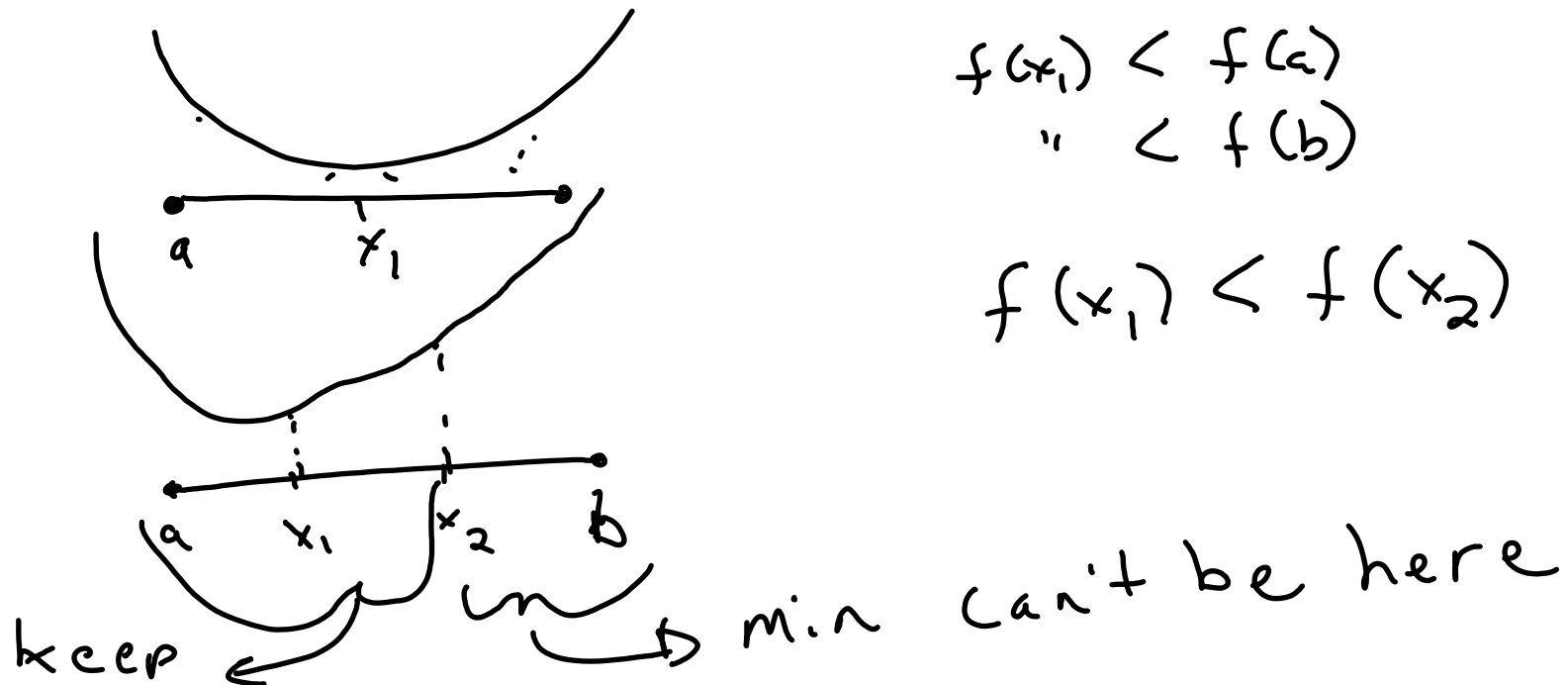
 unimodal not
unimodal

Unimodality

- For minimizing function of one variable, we need “bracket” for solution analogous to sign change for nonlinear equation
- Real-valued function f is *unimodal* on interval $[a, b]$ if there is unique $x^* \in [a, b]$ such that $f(x^*)$ is minimum of f on $[a, b]$, and f is strictly decreasing for $x \leq x^*$, strictly increasing for $x^* \leq x$
- Unimodality enables discarding portions of interval based on sample function values, analogous to interval bisection

Optimization and Root-finding

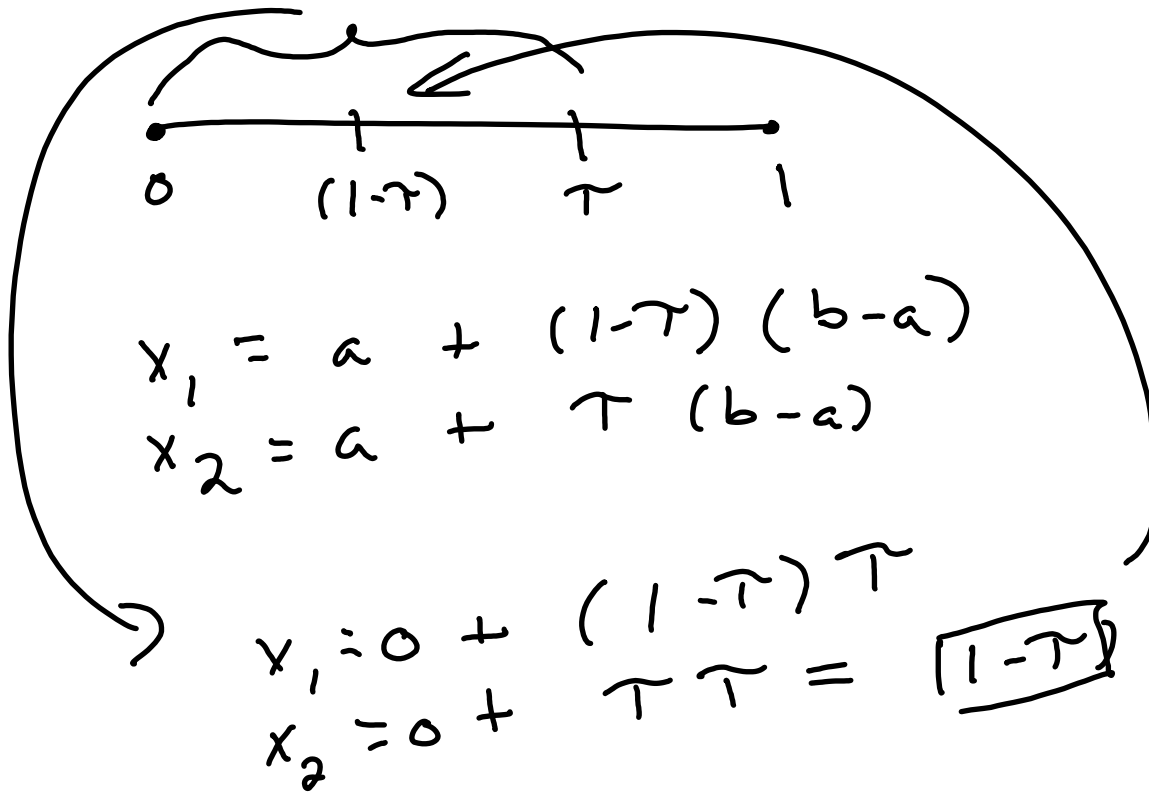
- In both problems, searching for specific function values
- Can we create an optimization algorithm like Bisection?



Golden Section Search

- Suppose f is unimodal on $[a, b]$, and let x_1 and x_2 be two points within $[a, b]$, with $x_1 < x_2$
- Evaluating and comparing $f(x_1)$ and $f(x_2)$, we can discard either $(x_2, b]$ or $[a, x_1)$, with minimum known to lie in remaining subinterval
- To repeat process, we need compute only one new function evaluation
- To reduce length of interval by fixed fraction at each iteration, each new pair of points must have same relationship with respect to new interval that previous pair had with respect to previous interval

Golden Section Search



$$\tau^2 = (1-\tau)$$
$$\tau \approx 0.618$$

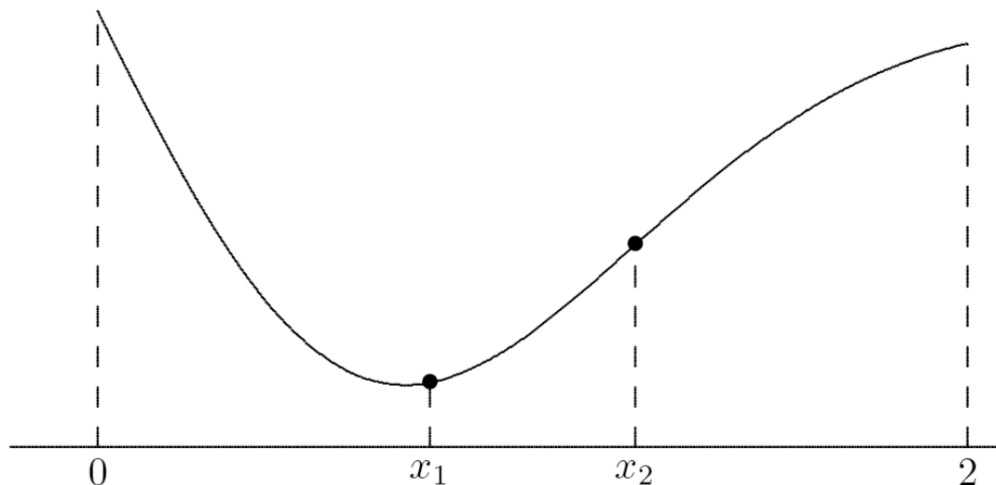
Golden Section Search

- To accomplish this, we choose relative positions of two points as τ and $1 - \tau$, where $\tau^2 = 1 - \tau$, so $\tau = (\sqrt{5} - 1)/2 \approx 0.618$ and $1 - \tau \approx 0.382$
- Whichever subinterval is retained, its length will be τ relative to previous interval, and interior point retained will be at position either τ or $1 - \tau$ relative to new interval
- To continue iteration, we need to compute only one new function value, at complementary point
- This choice of sample points is called *golden section search*
- Golden section search is safe but convergence rate is only linear, with constant $C \approx 0.618$

Golden Section Search: Example

Use golden section search to minimize

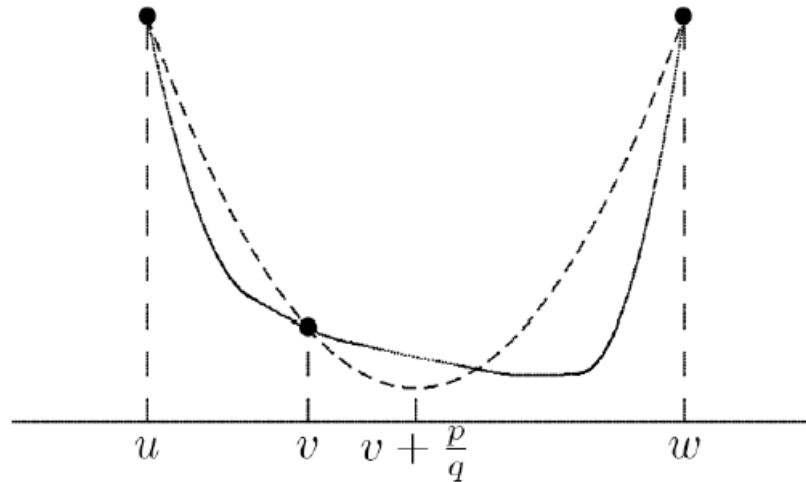
$$f(x) = 0.5 - x \exp(-x^2)$$



x_1	f_1	x_2	f_2
0.764	0.074	1.236	0.232
0.472	0.122	0.764	0.074
0.764	0.074	0.944	0.113
0.652	0.074	0.764	0.074
0.584	0.085	0.652	0.074
0.652	0.074	0.695	0.071
0.695	0.071	0.721	0.071
0.679	0.072	0.695	0.071
0.695	0.071	0.705	0.071
0.705	0.071	0.711	0.071

Successive Parabolic Interpolation

- Fit quadratic polynomial to three function values
- Take minimum of quadratic to be new approximation to minimum of function



- New point replaces oldest of three previous points and process is repeated until convergence
- Convergence rate of successive parabolic interpolation is superlinear, with $r \approx 1.324$

Newton's Method (for optimization)

- Another local quadratic approximation is truncated Taylor series

$$f(x+h) \approx f(x) + f'(x)h + \frac{f''(x)}{2}h^2$$

- By differentiation, minimum of this quadratic function of h is given by $h = -f'(x)/f''(x)$
- Suggests iteration scheme

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

which is *Newton's method* for solving nonlinear equation $f'(x) = 0$

- Newton's method for finding minimum normally has quadratic convergence rate, but must be started close

Newton's Method: Example

- Use Newton's method to minimize $f(x) = 0.5 - x \exp(-x^2)$
- First and second derivatives of f are given by

$$f'(x) = (2x^2 - 1) \exp(-x^2)$$

and

$$f''(x) = 2x(3 - 2x^2) \exp(-x^2)$$

- Newton iteration for zero of f' is given by

$$x_{k+1} = x_k - (2x_k^2 - 1) / (2x_k(3 - 2x_k^2))$$

- Using starting guess $x_0 = 1$, we obtain

x_k	$f(x_k)$
1.000	0.132
0.500	0.111
0.700	0.071
0.707	0.071

$$e_{k+1} \leq e_k^2$$

$$e_0 = 10^{-1}$$

4

$$\begin{bmatrix} 10^{-1} \\ 10^{-2} \\ 10^{-4} \\ 10^{-8} \\ 10^{-16} \\ 10 \end{bmatrix}$$

$$e_i = 10^{-(2^i)}$$

$$10^{-16} > 10^{-(2^i)}$$

$$-16 > -(2^i)$$

$$10 > 2^i$$

$$\lg 10 > i$$

$$\lceil \lg 10 \rceil = i$$

$$4 = i$$

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

Multi-Dimensional Optimization

Direct Search Methods: Nelder-Mead

- Direct search methods for multidimensional optimization make no use of function values other than comparing them
- For minimizing function f of n variables, *Nelder-Mead* method begins with $n + 1$ starting points, forming *simplex* in \mathbb{R}^n
- Then move to new point along straight line from current point having highest function value through centroid of other points
- New point replaces worst point, and process is repeated
- Direct search methods are useful for nonsmooth functions or for small n , but expensive for larger n

$$f(x_1, x_2)$$

$$\nabla f = \left\langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right\rangle$$

Steepest Descent

- Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be real-valued function of n real variables
- At any point x where gradient vector is nonzero, negative gradient, $-\nabla f(x)$, points downhill toward lower values of f
- In fact, $-\nabla f(x)$ is locally direction of steepest descent: f decreases more rapidly along direction of negative gradient than along any other
- **Steepest descent** method: starting from initial guess x_0 , successive approximate solutions given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)$$

where α_k is **line search** parameter that determines how far to go in given direction

$$\begin{array}{c} \bullet \\ \downarrow \\ x_k \end{array} \xrightarrow{\nabla f} \equiv x_k - \alpha \nabla f = g(\alpha)$$

Steepest Descent

- Given descent direction, such as negative gradient, determining appropriate value for α_k at each iteration is one-dimensional minimization problem

$$\min_{\alpha_k} f(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k))$$

that can be solved by methods already discussed

- Steepest descent method is very reliable: it can always make progress provided gradient is nonzero
- But method is myopic in its view of function's behavior, and resulting iterates can zigzag back and forth, making very slow progress toward solution
- In general, convergence rate of steepest descent is only linear, with constant factor that can be arbitrarily close to 1

Steepest Descent: Example

- Use steepest descent method to minimize

$$f(\mathbf{x}) = 0.5x_1^2 + 2.5x_2^2$$

- Gradient is given by $\nabla f(\mathbf{x}) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix}$

- Taking $\mathbf{x}_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$, we have $\nabla f(\mathbf{x}_0) = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$

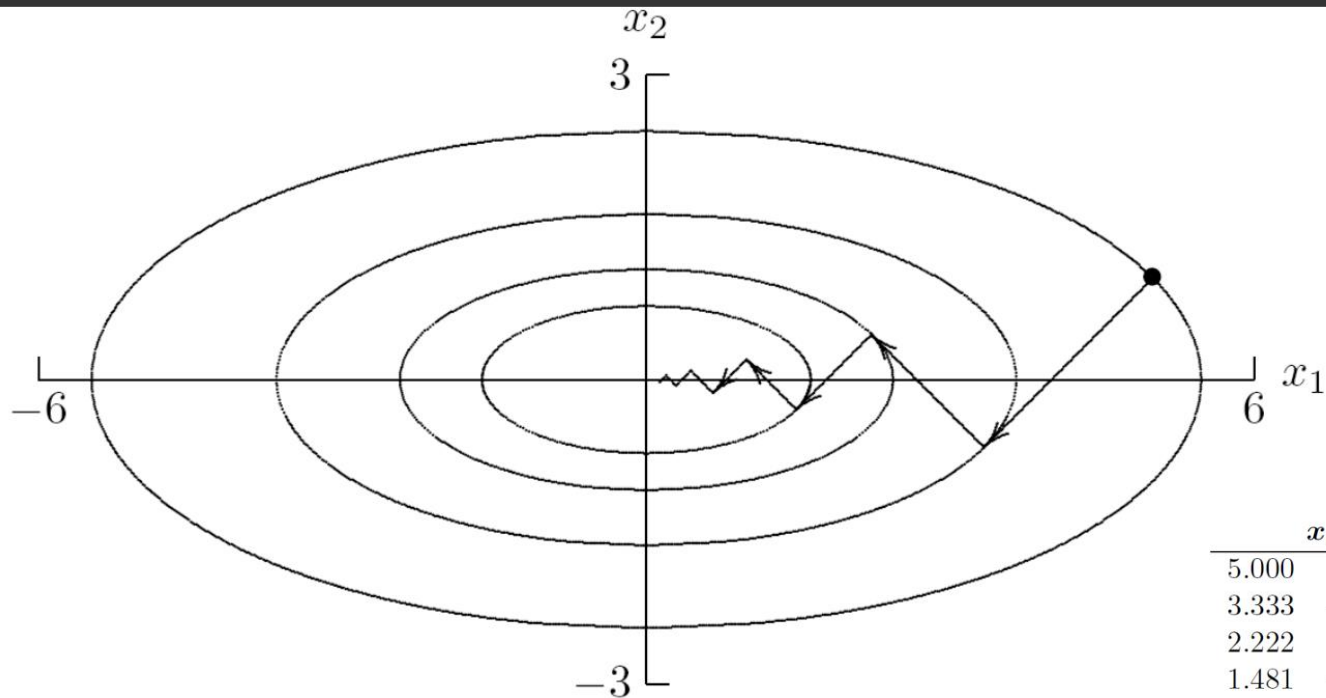
- Performing line search along negative gradient direction,

$$\min_{\alpha_0} f(\mathbf{x}_0 - \alpha_0 \nabla f(\mathbf{x}_0))$$

exact minimum along line is given by $\alpha_0 = 1/3$, so next

approximation is $\mathbf{x}_1 = \begin{bmatrix} 3.333 \\ -0.667 \end{bmatrix}$

Steepest Descent: Example



\mathbf{x}_k		$f(\mathbf{x}_k)$	$\nabla f(\mathbf{x}_k)$	
5.000	1.000	15.000	5.000	5.000
3.333	-0.667	6.667	3.333	-3.333
2.222	0.444	2.963	2.222	2.222
1.481	-0.296	1.317	1.481	-1.481
0.988	0.198	0.585	0.988	0.988
0.658	-0.132	0.260	0.658	-0.658
0.439	0.088	0.116	0.439	0.439
0.293	-0.059	0.051	0.293	-0.293
0.195	0.039	0.023	0.195	0.195
0.130	-0.026	0.010	0.130	-0.130

Multi-Dimensional Optimization: Newton's Method

- Broader view can be obtained by local quadratic approximation, which is equivalent to Newton's method
- In multidimensional optimization, we seek zero of gradient, so *Newton iteration* has form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_f^{-1}(\mathbf{x}_k) \nabla f(\mathbf{x}_k)$$

where $\mathbf{H}_f(\mathbf{x})$ is *Hessian* matrix of second partial derivatives of f ,

$$\{\mathbf{H}_f(\mathbf{x})\}_{ij} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$$

Multi-Dimensional Optimization: Newton's Method

- Do not explicitly invert Hessian matrix, but instead solve linear system

$$\mathbf{H}_f(\mathbf{x}_k) \mathbf{s}_k = -\nabla f(\mathbf{x}_k)$$

for Newton step \mathbf{s}_k , then take as next iterate

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$$

- Convergence rate of Newton's method for minimization is normally quadratic
- As usual, Newton's method is unreliable unless started close enough to solution to converge

Multi-Dimensional Optimization: Newton's Method

- If objective function f has continuous second partial derivatives, then Hessian matrix \mathbf{H}_f is symmetric, and near minimum it is positive definite
- Thus, linear system for step to next iterate can be solved in only about half of work required for LU factorization
- Far from minimum, $\mathbf{H}_f(\mathbf{x}_k)$ may not be positive definite, so Newton step \mathbf{s}_k may not be *descent direction* for function, i.e., we may not have

$$\nabla f(\mathbf{x}_k)^T \mathbf{s}_k < 0$$

- In this case, alternative descent direction can be computed, such as negative gradient or direction of negative curvature, and then perform line search

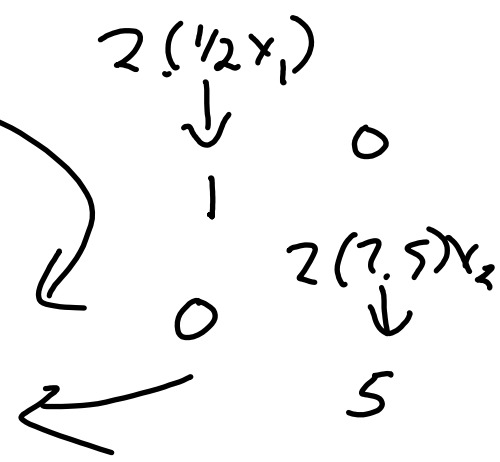
Multi-Dimensional Optimization: Newton's Method

- Use Newton's method to minimize

$$f(\mathbf{x}) = 0.5x_1^2 + 2.5x_2^2$$

- Gradient and Hessian are given by

$$\nabla f(\mathbf{x}) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_f(\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$



- Taking $\mathbf{x}_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$, we have $\nabla f(\mathbf{x}_0) = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$

- Linear system for Newton step is $\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix} \mathbf{s}_0 = \begin{bmatrix} -5 \\ -5 \end{bmatrix}$, so

$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{s}_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} + \begin{bmatrix} -5 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, which is exact solution for this problem, as expected for quadratic function