Scientific Computing: An Introductory Survey
Chapter 6 – Optimization

Prof. Michael T. Heath

Department of Computer Science
University of Illinois at Urbana-Champaign

.

Selected subset of slides for CS 357 - page numbers will not necessarily be correct

Outline

1. Optimization Problems

2. One-Dimensional Optimization

3. Multi-Dimensional Optimization

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Optimization

- Given function $f \colon \mathbb{R}^n \to \mathbb{R}$, and set $S \subseteq \mathbb{R}^n$, find $\boldsymbol{x}^* \in S$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in S$

- $\boldsymbol{x}^*$ 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 $\boldsymbol{x} \in S$ are called *feasible* points

- If $S = \mathbb{R}^n$, problem is *unconstrained*

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Optimization Problems

- General continuous optimization problem:

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

  where $f \colon \mathbb{R}^n \to \mathbb{R}, \quad \boldsymbol{g} \colon \mathbb{R}^n \to \mathbb{R}^m, \quad \boldsymbol{h} \colon \mathbb{R}^n \to \mathbb{R}^p$

- *Linear programming* : f, $\boldsymbol{g}$, and $\boldsymbol{h}$ are all linear

- *Nonlinear programming* : at least one of f, $\boldsymbol{g}$, and $\boldsymbol{h}$ is nonlinear

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Examples: Optimization Problems

- Minimize weight of structure subject to constraint on its strength, or maximize its strength subject to constraint on its weight

- Minimize cost of diet subject to nutritional constraints

- Minimize surface area of cylinder subject to constraint on its volume:

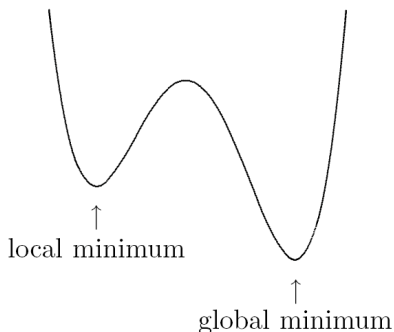$$\min_{x_1, x_2} f(x_1, x_2) = 2\pi x_1(x_1 + x_2)$$

  subject to $\quad g(x_1, x_2) = \pi x_1^2 x_2 - V = 0$

  where x_1 and x_2 are radius and height of cylinder, and V is required volume

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Local vs Global Optimization

- $x^* \in S$ 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^*

↑
local minimum

↑
global minimum

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Existence of Minimum

- If f is continuous on *closed* and *bounded* set $S \subseteq \mathbb{R}^n$, then f has global minimum on S

- If S is not closed or is unbounded, then f may have no local or global minimum on S

- Continuous function f on unbounded set $S \subseteq \mathbb{R}^n$ is *coercive* if

$$\lim_{\|\boldsymbol{x}\| \to \infty} f(\boldsymbol{x}) = +\infty$$

  i.e., $f(\boldsymbol{x})$ must be large whenever $\|\boldsymbol{x}\|$ is large

- If f is coercive on closed, unbounded set $S \subseteq \mathbb{R}^n$, then f has global minimum on S

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Level Sets

- *Level set* for function $f \colon S \subseteq \mathbb{R}^n \to \mathbb{R}$ is set of all points in S for which f has some given constant value

- For given $\gamma \in \mathbb{R}$, *sublevel set* is

$$L_\gamma = \{\boldsymbol{x} \in S : \ f(\boldsymbol{x}) \leq \gamma\}$$

- If continuous function f on $S \subseteq \mathbb{R}^n$ has nonempty sublevel set that is closed and bounded, then f has global minimum on S

- If S is unbounded, then f is coercive on S if, and only if, *all* of its sublevel sets are bounded

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Uniqueness of Minimum

- Set $S \subseteq \mathbb{R}^n$ is *convex* if it contains line segment between any two of its points

- Function $f \colon S \subseteq \mathbb{R}^n \to \mathbb{R}$ is *convex* on convex set S if its graph along any line segment in S lies *on or below* chord connecting function values at endpoints of segment

- Any local minimum of convex function f on convex set $S \subseteq \mathbb{R}^n$ is global minimum of f on S

- Any local minimum of *strictly* convex function f on convex set $S \subseteq \mathbb{R}^n$ is *unique* global minimum of f on S

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

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

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

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

- For continuously differentiable $f\colon S \subseteq \mathbb{R}^n \to \mathbb{R}$, any interior point $\boldsymbol{x}^*$ 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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Definitions
Existence and Uniqueness
Optimality Conditions

Second-Order Optimality Condition

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

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

which is symmetric

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

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 Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
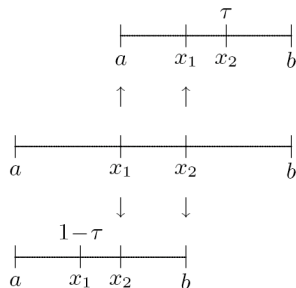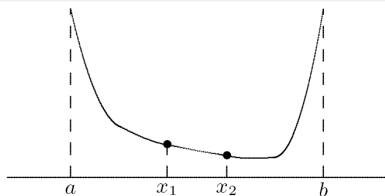Newton's Method

Golden Section Search, continued

- 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 $C \approx 0.618$

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

Golden Section Search, continued

$\tau = (\sqrt{5} - 1)/2$
$x_1 = a + (1 - \tau)(b - a); \; f_1 = f(x_1)$
$x_2 = a + \tau(b - a); \; f_2 = f(x_2)$
while $((b - a) > tol)$ **do**
    if $(f_1 > f_2)$ **then**
        $a = x_1$
        $x_1 = x_2$
        $f_1 = f_2$
        $x_2 = a + \tau(b - a)$
        $f_2 = f(x_2)$
    else
        $b = x_2$
        $x_2 = x_1$
        $f_2 = f_1$
        $x_1 = a + (1 - \tau)(b - a)$
        $f_1 = f(x_1)$
    end
end

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

Example: Golden Section Search

Use golden section search to minimize

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method
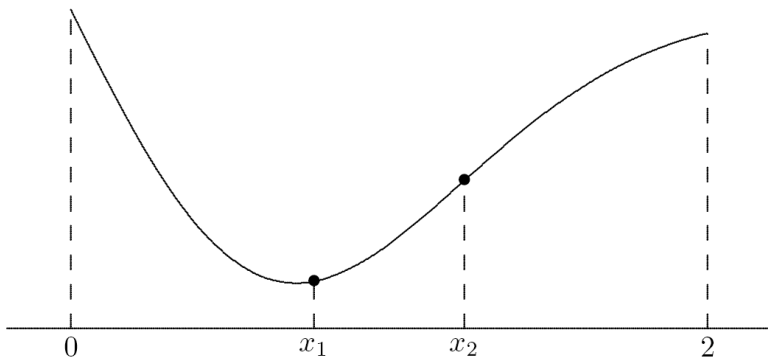
Example, continued

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

< interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

Newton's Method

- 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 enough to solution to converge   < interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

Example: Newton's Method

- 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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Golden Section Search
Successive Parabolic Interpolation
Newton's Method

Safeguarded Methods

- As with nonlinear equations in one dimension, slow-but-sure and fast-but-risky optimization methods can be combined to provide both safety and efficiency

- Most library routines for one-dimensional optimization are based on this hybrid approach

- Popular combination is golden section search and successive parabolic interpolation, for which no derivatives are required

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Direct Search Methods

- 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

< interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Steepest Descent Method

- Let $f \colon \mathbb{R}^n \to \mathbb{R}$ 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

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Steepest Descent, continued

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

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

  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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example: Steepest Descent

- Use steepest descent method to minimize

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

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

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

- Performing line search along negative gradient direction,

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

exact minimum along line is given by $\alpha_0 = 1/3$, so next
approximation is $\boldsymbol{x}_1 = \begin{bmatrix} 3.333 \\ -0.667 \end{bmatrix}$

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example, continued

$\boldsymbol{x}_k$		$f(\boldsymbol{x}_k)$	$\nabla f(\boldsymbol{x}_k)$	
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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example, continued

< interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization
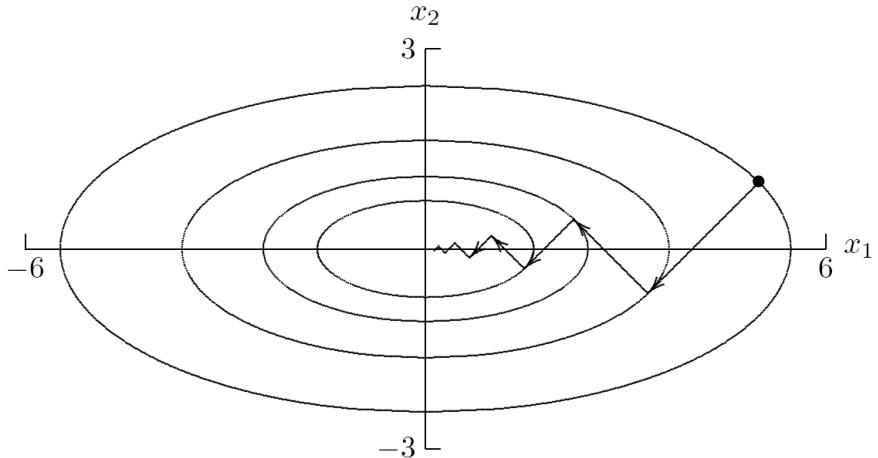
Newton's Method

- 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

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

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

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Newton's Method, continued

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

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

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

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

- 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

< interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example: Newton's Method

- Use Newton's method to minimize

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

- Gradient and Hessian are given by

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

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

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

  $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{s}_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} + \begin{bmatrix} -5 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, which is exact solution

  for this problem, as expected for quadratic function

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Newton's Method, continued

- In principle, line search parameter is unnecessary with Newton's method, since quadratic model determines length, as well as direction, of step to next approximate solution

- When started far from solution, however, it may still be advisable to perform line search along direction of Newton step s_k to make method more robust (*damped* Newton)

- Once iterates are near solution, then $\alpha_k = 1$ should suffice for subsequent iterations

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Newton's Method, continued

- If objective function f has continuous second partial derivatives, then Hessian matrix $\boldsymbol{H}_f$ 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, $\boldsymbol{H}_f(\boldsymbol{x}_k)$ may not be positive definite, so Newton step $\boldsymbol{s}_k$ may not be *descent direction* for function, i.e., we may not have

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

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

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
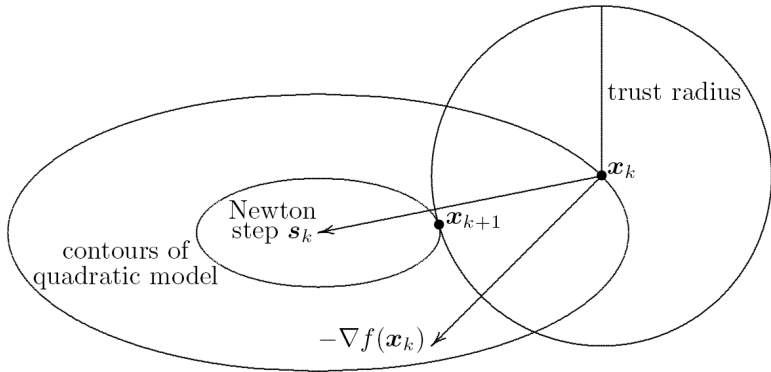Nonlinear Least Squares
Constrained Optimization

Trust Region Methods

- Alternative to line search is *trust region method*, in which approximate solution is constrained to lie within region where quadratic model is sufficiently accurate

- If current trust radius is binding, minimizing quadratic model function subject to this constraint may modify direction as well as length of Newton step

- Accuracy of quadratic model is assessed by comparing actual decrease in objective function with that predicted by quadratic model, and trust radius is increased or decreased accordingly

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Trust Region Methods, continued

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Quasi-Newton Methods

- Newton's method costs $\mathcal{O}(n^3)$ arithmetic and $\mathcal{O}(n^2)$ scalar function evaluations per iteration for dense problem

- Many variants of Newton's method improve reliability and reduce overhead

- *Quasi-Newton* methods have form

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

  where α_k is line search parameter and $\boldsymbol{B}_k$ is approximation to Hessian matrix

- Many quasi-Newton methods are more robust than Newton's method, are superlinearly convergent, and have lower overhead per iteration, which often more than offsets their slower convergence rate

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Conjugate Gradient Method

- Another method that does not require explicit second derivatives, and does not even store approximation to Hessian matrix, is *conjugate gradient* (CG) method

- CG generates sequence of conjugate search directions, implicitly accumulating information about Hessian matrix

- For quadratic objective function, CG is theoretically exact after at most n iterations, where n is dimension of problem

- CG is effective for general unconstrained minimization as well

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Conjugate Gradient Method, continued

$\boldsymbol{x}_0 = $ initial guess
$\boldsymbol{g}_0 = \nabla f(\boldsymbol{x}_0)$
$\boldsymbol{s}_0 = -\boldsymbol{g}_0$
for $k = 0, 1, 2, \ldots$
    Choose α_k to minimize $f(\boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k)$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k$
    $\boldsymbol{g}_{k+1} = \nabla f(\boldsymbol{x}_{k+1})$
    $\beta_{k+1} = (\boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1})/(\boldsymbol{g}_k^T \boldsymbol{g}_k)$
    $\boldsymbol{s}_{k+1} = -\boldsymbol{g}_{k+1} + \beta_{k+1} \boldsymbol{s}_k$
end

- Alternative formula for β_{k+1} is

$$\beta_{k+1} = ((\boldsymbol{g}_{k+1} - \boldsymbol{g}_k)^T \boldsymbol{g}_{k+1})/(\boldsymbol{g}_k^T \boldsymbol{g}_k)$$

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example: Conjugate Gradient Method

- Use CG method to minimize $f(\boldsymbol{x}) = 0.5x_1^2 + 2.5x_2^2$
- Gradient is given by $\nabla f(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix}$

- Taking $\boldsymbol{x}_0 = \begin{bmatrix} 5 & 1 \end{bmatrix}^T$, initial search direction is negative gradient,

$$\boldsymbol{s}_0 = -\boldsymbol{g}_0 = -\nabla f(\boldsymbol{x}_0) = \begin{bmatrix} -5 \\ -5 \end{bmatrix}$$

- Exact minimum along line is given by $\alpha_0 = 1/3$, so next approximation is $\boldsymbol{x}_1 = \begin{bmatrix} 3.333 & -0.667 \end{bmatrix}^T$, and we compute new gradient,

$$\boldsymbol{g}_1 = \nabla f(\boldsymbol{x}_1) = \begin{bmatrix} 3.333 \\ -3.333 \end{bmatrix}$$

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example, continued

- So far there is no difference from steepest descent method
- At this point, however, rather than search along new negative gradient, we compute instead

$$\beta_1 = (\boldsymbol{g}_1^T \boldsymbol{g}_1)/(\boldsymbol{g}_0^T \boldsymbol{g}_0) = 0.444$$

which gives as next search direction

$$\boldsymbol{s}_1 = -\boldsymbol{g}_1 + \beta_1 \boldsymbol{s}_0 = \begin{bmatrix} -3.333 \\ 3.333 \end{bmatrix} + 0.444 \begin{bmatrix} -5 \\ -5 \end{bmatrix} = \begin{bmatrix} -5.556 \\ 1.111 \end{bmatrix}$$

- Minimum along this direction is given by $\alpha_1 = 0.6$, which gives exact solution at origin, as expected for quadratic function

< interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Truncated Newton Methods

- Another way to reduce work in Newton-like methods is to solve linear system for Newton step by iterative method

- Small number of iterations may suffice to produce step as useful as true Newton step, especially far from overall solution, where true Newton step may be unreliable anyway

- Good choice for linear iterative solver is CG method, which gives step intermediate between steepest descent and Newton-like step

- Since only matrix-vector products are required, explicit formation of Hessian matrix can be avoided by using finite difference of gradient along given vector

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Nonlinear Least Squares

- Given data (t_i, y_i), find vector $\boldsymbol{x}$ of parameters that gives "best fit" in least squares sense to model function $f(t, \boldsymbol{x})$, where f is nonlinear function of $\boldsymbol{x}$

- Define components of *residual function*

$$r_i(\boldsymbol{x}) = y_i - f(t_i, \boldsymbol{x}), \quad i = 1, \ldots, m$$

so we want to minimize $\phi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})$

- Gradient vector is $\nabla\phi(\boldsymbol{x}) = \boldsymbol{J}^T(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})$ and Hessian matrix is

$$\boldsymbol{H}_\phi(\boldsymbol{x}) = \boldsymbol{J}^T(\boldsymbol{x})\boldsymbol{J}(\boldsymbol{x}) + \sum_{i=1}^{m} r_i(\boldsymbol{x})\boldsymbol{H}_i(\boldsymbol{x})$$

where $\boldsymbol{J}(\boldsymbol{x})$ is Jacobian of $\boldsymbol{r}(\boldsymbol{x})$, and $\boldsymbol{H}_i(\boldsymbol{x})$ is Hessian of $r_i(\boldsymbol{x})$

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Nonlinear Least Squares, continued

- Linear system for Newton step is

$$\left(\boldsymbol{J}^T(\boldsymbol{x}_k)\boldsymbol{J}(\boldsymbol{x}_k) + \sum_{i=1}^{m} r_i(\boldsymbol{x}_k)\boldsymbol{H}_i(\boldsymbol{x}_k) \right) \boldsymbol{s}_k = -\boldsymbol{J}^T(\boldsymbol{x}_k)\boldsymbol{r}(\boldsymbol{x}_k)$$

- m Hessian matrices $\boldsymbol{H}_i$ are usually inconvenient and expensive to compute

- Moreover, in $\boldsymbol{H}_\phi$ each $\boldsymbol{H}_i$ is multiplied by residual component r_i, which is small at solution if fit of model function to data is good

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Gauss-Newton Method

- This motivates Gauss-Newton method for nonlinear least squares, in which second-order term is dropped and linear system

$$\boldsymbol{J}^T(\boldsymbol{x}_k)\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{s}_k = -\boldsymbol{J}^T(\boldsymbol{x}_k)\boldsymbol{r}(\boldsymbol{x}_k)$$

  is solved for approximate Newton step $\boldsymbol{s}_k$ at each iteration

- This is system of normal equations for linear least squares problem

$$\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{s}_k \cong -\boldsymbol{r}(\boldsymbol{x}_k)$$

  which can be solved better by QR factorization

- Next approximate solution is then given by

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

  and process is repeated until convergence

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example: Gauss-Newton Method

- Use Gauss-Newton method to fit nonlinear model function

$$f(t, \boldsymbol{x}) = x_1 \exp(x_2 t)$$

to data

t	0.0	1.0	2.0	3.0
y	2.0	0.7	0.3	0.1

- For this model function, entries of Jacobian matrix of residual function $\boldsymbol{r}$ are given by

$$\{\boldsymbol{J}(\boldsymbol{x})\}_{i,1} = \frac{\partial r_i(\boldsymbol{x})}{\partial x_1} = -\exp(x_2 t_i)$$

$$\{\boldsymbol{J}(\boldsymbol{x})\}_{i,2} = \frac{\partial r_i(\boldsymbol{x})}{\partial x_2} = -x_1 t_i \exp(x_2 t_i)$$

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example, continued

- If we take $x_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, then Gauss-Newton step s_0 is given by linear least squares problem

$$\begin{bmatrix} -1 & 0 \\ -1 & -1 \\ -1 & -2 \\ -1 & -3 \end{bmatrix} s_0 \cong \begin{bmatrix} -1 \\ 0.3 \\ 0.7 \\ 0.9 \end{bmatrix}$$

  whose solution is $s_0 = \begin{bmatrix} 0.69 \\ -0.61 \end{bmatrix}$

- Then next approximate solution is given by $x_1 = x_0 + s_0$, and process is repeated until convergence

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example, continued

x_k		$\|r(x_k)\|_2^2$
1.000	0.000	2.390
1.690	−0.610	0.212
1.975	−0.930	0.007
1.994	−1.004	0.002
1.995	−1.009	0.002
1.995	−1.010	0.002

< interactive example >

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Gauss-Newton Method, continued

- Gauss-Newton method replaces nonlinear least squares problem by sequence of linear least squares problems whose solutions converge to solution of original nonlinear problem

- If residual at solution is large, then second-order term omitted from Hessian is not negligible, and Gauss-Newton method may converge slowly or fail to converge

- In such "large-residual" cases, it may be best to use general nonlinear minimization method that takes into account true full Hessian matrix

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Linear Programming

- One of most important and common constrained optimization problems is *linear programming*

- One standard form for such problems is

  $$\min f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \quad \text{and} \quad \boldsymbol{x} \geq \boldsymbol{0}$$

  where $m < n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^n$

- Feasible region is convex polyhedron in $\mathbb{R}^n$, and minimum must occur at one of its vertices

- *Simplex* method moves systematically from vertex to vertex until minimum point is found

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Linear Programming, continued

- Simplex method is reliable and normally efficient, able to solve problems with thousands of variables, but can require time exponential in size of problem in worst case

- *Interior point* methods for linear programming developed in recent years have polynomial worst case solution time

- These methods move through interior of feasible region, not restricting themselves to investigating only its vertices

- Although interior point methods have significant practical impact, simplex method is still predominant method in standard packages for linear programming, and its effectiveness in practice is excellent

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example: Linear Programming

- To illustrate linear programming, consider

$$\min_{\boldsymbol{x}} = \boldsymbol{c}^T \boldsymbol{x} = -8x_1 - 11x_2$$

subject to linear inequality constraints

$$5x_1 + 4x_2 \le 40, \quad -x_1 + 3x_2 \le 12, \quad x_1 \ge 0, \ x_2 \ge 0$$

- Minimum value must occur at vertex of feasible region, in this case at $x_1 = 3.79$, $x_2 = 5.26$, where objective function has value -88.2

Optimization Problems
One-Dimensional Optimization
Multi-Dimensional Optimization

Unconstrained Optimization
Nonlinear Least Squares
Constrained Optimization

Example, continued