MTH8408A

Méthodes Numériques d'Optimisation et de Contrôle Optimal

Introduction to Unconstrained Optimization

Dominique Orban
dominique.orban@polymtl.ca

Mathématiques et Génie Industriel
École Polytechnique de Montréal

Hiver 2017

The Problem

We seek a *vector* $x \in \mathbb{R}^n$ that solves the problem

$$\underset{x}{\text{minimize}} \;\; f(x),$$

where we assume that $f : \mathbb{R}^n \to \mathbb{R}$ is continuous and has continuous first (and perhaps second) partial derivatives.

We say that $x^* \in \mathbb{R}^n$ is a local minimizer if there exists $r > 0$ such that

$$f(x) \geq f(x^*) \quad \text{for all } x \in B(x^*; r).$$

We say that $x^* \in \mathbb{R}^n$ is a global minimizer if

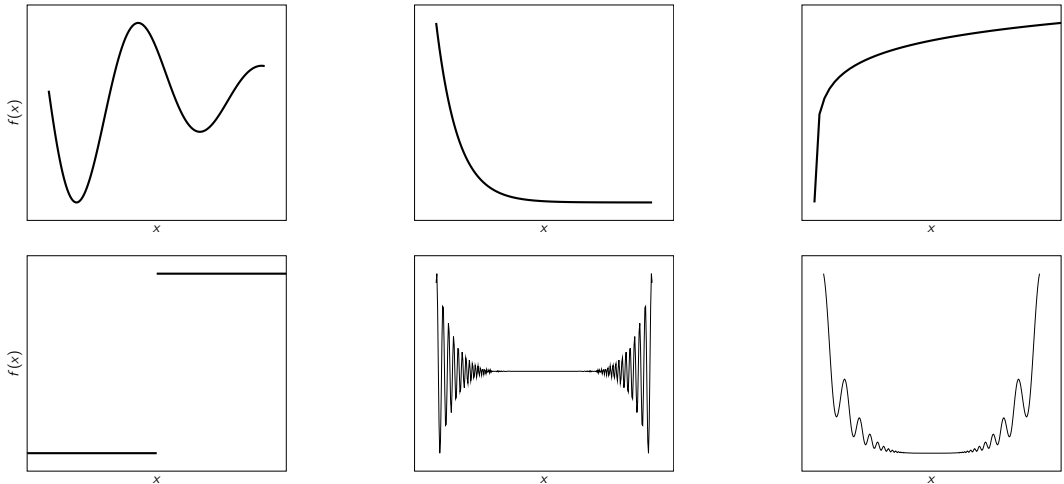
$$f(x) \geq f(x^*) \quad \text{for all } x \in \mathbb{R}^n.$$

Under our assumption it is possible to state first-order necessary optimality conditions.

Classes of Problems Not Considered in this Course

- Problems without derivatives or whose derivatives are not computable,
- multi-objective problems: $\underset{x \in \mathbb{R}^n}{\text{minimize}} \ \{f_1(x), \ldots, f_m(x)\}$
- multi-level problems: $\underset{x \in \mathbb{R}^n, y \in \mathbb{R}^m}{\text{minimize}} \ f_1(x, y)$ subject to y solves $\underset{y \in \mathbb{R}^m}{\text{minimize}} \ f_2(x, y)$
- problems whose variables are matrices (semi-definite optimization, cone optimization)
- problems involving stochastic variables (stochastic optimization)
- problems involving discrete variables (integer or categorical)
- ...

What is a "Solution"? What Can we Expect of a Solver?



First- and Second-Order Optimality Conditions

The following results can be established using Taylor's theorem:

N1 if x^* is a local minimum and $f \in \mathcal{C}^1$, then $\nabla f(x^*) = 0$
N2 if x^* is a local minimum and $f \in \mathcal{C}^2$, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$
S2 if $f \in \mathcal{C}^2$, $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, then x^* is a local minimum

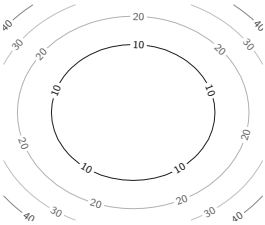
If $\nabla f(x^*) = 0$, x^* is called a *first-order critical point*, or a *stationary* point.

Exercise: establish the results above using

$$f(x^* + d) - f(x^*) = \nabla f(x^*)^T d + \tfrac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2).$$

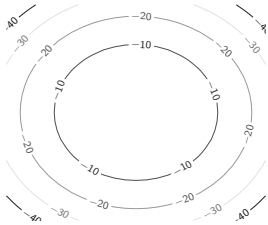
Minimum, Maximum and Saddle Point

Local Minimum



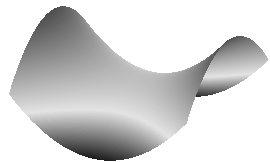
$\nabla^2 f(x_*) \succ 0$

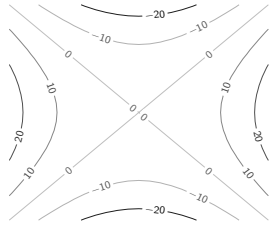
Local Maximum



$\nabla^2 f(x_*) \prec 0$

Saddle Point





$\nabla^2 f(x_*)$ indefinite

Example

The stationary points of

$$f : \mathbb{R}^2 \to \mathbb{R}, \qquad f(x) = 2x_1^3 - 3x_1^2 - 6x_1x_2(x_1 - x_2 - 1)$$

are

- $(0, 0)$: a saddle point
- $(1, 0)$: a local minimum
- $(-1, -1)$: a local maximum
- $(0, -1)$: a saddle point.

Optimization $\neq$ Nonlinear Equations

Optimization does NOT consist in solving the equations $\nabla f(x) = 0$!!!

(c.f. notebook about Newton's method)

Special Case: Quadratic Objective

The quadratic model

$$q(s) = g^T s + \tfrac{1}{2} s^T H s \quad \text{with } H = H^T$$

is important because $f(x+s) \approx q(s)$ if $f \in \mathcal{C}^2$.

The optimality conditions yield

N2 if s^* is a local minimum, then $Hs^* = -g$ and $H \succeq 0$

S2 if $Hs^* = -g$ and $H \succ 0$ then s^* is a local minimum.

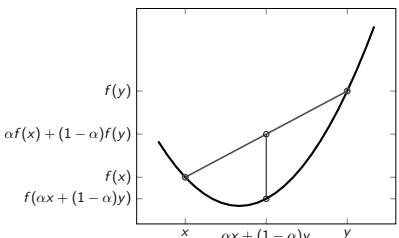
Exercise: In S2, show that s^* is in fact a *global* minimum!

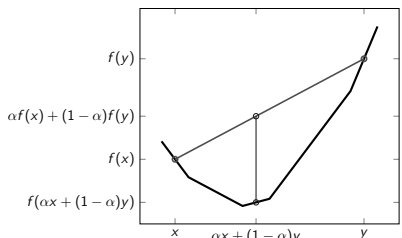
In particular

1. if $H \not\succeq 0$, q is not bounded below
2. if $H \succ 0$, then minimize $q(s) \Leftrightarrow Hs = -g$.

Special Case: Convex Objective

- If $f \in \mathcal{C}^2$, f is *convex* if $\nabla^2 f(x) \succeq 0$ for all $x \in \mathbb{R}^n$,
- if $f \in \mathcal{C}^1$, f is *convex* if $f(y) \geq f(x) + \nabla f(x)^T (y - x)$ for all x, $y \in \mathbb{R}^n$,
- f is convex if $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$ for all $\alpha \in [0, 1]$ and x, $y \in \mathbb{R}^n$.





If $\nabla f(x) = 0$, then $f(y) \geq f(x)$ for all $y \in \mathbb{R}^n$ and x is a *global* minimum!

If x is a local minimum, then x is a *global* minimum!

Examples of Convex Functions

1. e^{ax}, x^a ($a \leq 0$ or $a \geq 1$), $|x|^p$ for $p \geq 1$ are convex on $\mathbb{R}$
2. $-\log(x)$ is convex on $\mathbb{R}_0^+$,
3. $\|x\|_p$ is convex on $\mathbb{R}^n$ for $p \geq 1$ and $p = \infty$,
4. $-\log(\det(X))$ where X is a symmetric positive definite matrix
5. $q(s) = g^T s + \frac{1}{2} s^T H s$ when $H \succeq 0$,
6. if f and g are convex and $\lambda \geq 0$, then $f + \lambda g$ and $\max\{f, g\}$ are convex,
7. if f is linear and g is convex, then $g \circ f$ is convex,
8. ...

Excellent reference:
S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004
Online: `stanford.edu/~boyd/cvxbook`

Problems with Fixed Variables

In variational calculus, we often encounter problems of the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to } \; x_j = a_j \text{ (given)}, \; j \in \mathcal{J},$$

where $\mathcal{J} \subseteq \{1, \ldots, n\}$.

After substituting each of those x_j with a_j, we recover an unconstrained problem.

In variational calculus, fixed variables often appear from initial/final conditions.

Modeling languages such as AMPL are able to perform the substitution automatically.

Introduction to Algorithms for Unconstrained Optimization

Almost all numerical methods that we will encounter fit the following framework.

Algorithm 1 Template Descent Method

1: Choose an initial guess $x_0 \in \mathbb{R}^n$, set $k = 0$.
2: If x_k is optimal, stop.
3: From x_k generate x_{k+1} such that $f(x_{k+1}) < f(x_k)$.
4: $k \leftarrow k + 1$, return to Step 2.

Stopping condition?

1. $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$
2. number of evaluations of f exceed the budget
3. time runs out
4. an error occurs
5. ...

Coordinate Descent and Direct Search

- Do not use gradient information
- in Coordinate Descent, the objective is improved along one component (or one group of components) at a time
- in Direct Search methods, a similar idea is applied with directions other than the coordinate directions.

Linesearch Methods

At x_k, identify a search direction d_k and a stepsize $t_k > 0$ such that

$$f(x_k + t_k d_k) < f(x_k).$$

By Taylor's Theorem, that will always be possible if d_k is a descent direction.

Definition

The direction d_k is a descent direction for f at x_k if $\nabla f(x_k)^T d_k < 0$.

Examples:

1. $d_k = -\nabla f(x_k)$
2. $d_k = -B_k \nabla f(x_k)$ where $B_k = B_k^T \succ 0$
3. there are infinitely many choices as long as $\nabla f(x_k) \neq 0$
4. numerically, it is important that $\nabla f(x_k)^T d_k \leq -\epsilon \|\nabla f(x_k)\| \|d_k\| < 0$.

Terminology

Once a descent direction d_k has been identified, we say that we use

1. an *exact* linesearch if t_k solves

$$\underset{t>0}{\text{minimize}} \;\; f(x_k + td_k),$$

2. no linesearch if $t_k = 1$.

In practice, an exact linesearch is too costly and does not pay off, except in specific cases (e.g., f is quadratic).

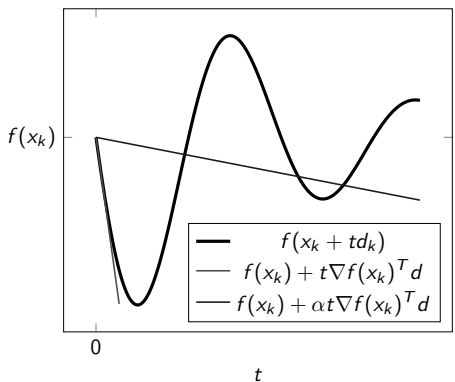
Not using a linesearch normally does not guarantee that our algorithm will converge (typically, it won't!).

Armijo Backtracking Linesearch

Theorem

If $f \in \mathcal{C}^2$ and $\alpha \in (0,1)$, there is an interval $(0, T]$ such that for all $t \in (0, T]$,

$$f(x_k + td_k) \leq f(x_k) + \alpha t \nabla f(x_k)^T d_k.$$



Strategy:

1. choose an initial $t > 0$ (e.g., $t = 1$)
2. check Armijo condition
3. decrease t (e.g., $t \leftarrow t/2$)
4. repeat

The first t to satisfy the Armijo condition is our t_k.

Other Linesearch Methods

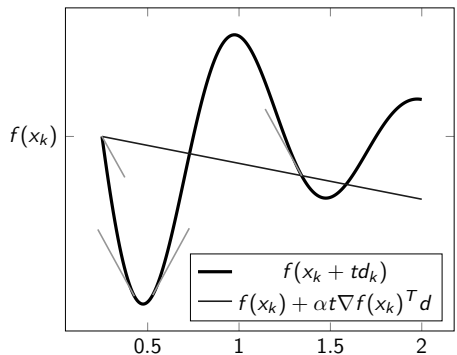
We focus on the Armijo linesearch for simplicity but other linesearch methods are important in theory and in practice:

1. the Wolfe linesearch
2. the strong Wolfe linesearch
3. the Armijo-Goldstein linesearch
4. non-monotone linesearch methods (i.e., that permit temporary increases in f)

The Strong Wolfe Linesearch

With $0 < \alpha < \gamma < 1$, seek $t > 0$ such that

$$f(x_k + td_k) \leq f(x_k) + \alpha t \nabla f(x_k)^T d_k$$
$$|\nabla f(x_k + td_k)^T d_k| \leq \gamma |\nabla f(x_k)^T d_k|$$



Implementation requires minimizing and updating a cubic interpolant.

A General Convergence Result

Algorithm 2 Descent Method with Armijo Linesearch

1: Choose an initial guess $x_0 \in \mathbb{R}^n$, $\cos_{\min} > 0$, set $k = 0$.
2: If $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, stop.
3: Compute a descent direction d_k such that $\cos(\nabla f(x_k), d_k) \geq \cos_{\min}$.
4: Obtain t_k from an Armijo linesearch along d_k
5: Set $x_{k+1} = x_k + t_k d_k$, $k \leftarrow k + 1$, return to Step 2.

Theorem

Assume f is $\mathcal{C}^2$ and $\{x_k\}$ is generated with Algorithm 2. Then either

1. $f(x_k) \to -\infty$, or
2. $\nabla f(x_k) \to 0$.

The Steepest Descent Method

Algorithm 3 Steepest Descent Method with Armijo Linesearch

1: Choose an initial guess $x_0 \in \mathbb{R}^n$, set $k = 0$.
2: If $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, stop.
3: Set $d_k = -\nabla f(x_k)$.
4: Obtain t_k from an Armijo linesearch along d_k.
5: Set $x_{k+1} = x_k + t_k d_k$, $k \leftarrow k + 1$, return to Step 2.

- the simplest gradient-based method
- $d_k = -\nabla f(x_k)$ solves the problem

$$\underset{d \in \mathbb{R}^n}{\text{minimize}} \ \nabla f(x_k)^T d \quad \text{subject to } \|d\| = \|\nabla f(x_k)\|$$

- may require many iterations even if f is quadratic
- geometrically, d_k is orthogonal to the level curve at x_k

Properties of an Idealized Steepest Descent Method: Quadratic

Theorem

Suppose $f(x) = g^T x + \frac{1}{2} x^T H x$ *with* $H = H^T \succ 0$ *and let* $x^* := -H^{-1} g$. *If we apply the steepest-descent method with exact linesearch to* f, *then*

$$f(x_{k+1}) - f(x^*) \leq \left(\frac{\lambda_{\max}(H) - \lambda_{\min}(H)}{\lambda_{\max}(H) + \lambda_{\min}(H)} \right)^2 (f(x_k) - f(x^*)), \quad k = 0, 1, \ldots$$

In particular,

- if $\lambda_{\max}(H) = \lambda_{\min}(H)$: one iteration!
- if $\lambda_{\max}(H) \gg \lambda_{\min}(H)$: very slow progress.

Properties of an Idealized Steepest Descent Method: General

Theorem

If $f \in \mathcal{C}^3$, let $\{x_k\}$ be generated by the steepest-descent method with exact linesearch. Suppose $\{x_k\} \to x^*$ where $\nabla f(x^*) = 0$ and $H := \nabla^2 f(x^*) \succ 0$. For sufficiently large k,

$$f(x_{k+1}) - f(x^*) \leq r^2 \, (f(x_k) - f(x^*)), \quad k = 0, 1, \ldots,$$

where r is any number such that

$$\frac{\lambda_{\max}(H) - \lambda_{\min}(H)}{\lambda_{\max}(H) + \lambda_{\min}(H)} < r < 1.$$

Newton's Method

Algorithm 4 Newton's Method with Armijo Linesearch

1: Choose an initial guess $x_0 \in \mathbb{R}^n$, set $k = 0$.
2: If $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, stop.
3: Solve $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$ for d_k.
4: Obtain t_k from an Armijo linesearch along d_k.
5: Set $x_{k+1} = x_k + t_k d_k$, $k \leftarrow k + 1$, return to Step 2.

- Is d_k a descent direction?
- if $\nabla^2 f(x_k) \succ 0$, d_k solves the problem

$$\underset{d \in \mathbb{R}^n}{\text{minimize}} \ \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_k) d$$

- what if $\nabla^2 f(x_k) \not\succ 0$?

Properties of Newton's Method

Globally (i.e., far from x^*), the difficulty is that $\nabla^2 f(x_k)$ may not be positive definite.

Locally (i.e., close to x^*), we have

Theorem

If $f \in \mathcal{C}^3$, $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, suppose $\{x_k\}$ is generated using Newton's method without linesearch. Then

1. *if x_0 is sufficiently close to x^*, $\{x_k\} \to x^*$*
2. *$\|x_{k+1} - x^*\| \leq C_1 \|x_k - x^*\|^2$ for some $C_1 > 0$ and all sufficiently large k,*
3. *$\|\nabla f(x_{k+1})\| \leq C_2 \|\nabla f(x_k)\|^2$ for some $C_2 > 0$ and all sufficiently large k.*

If the linesearch method somehow converges to such an x^*, we eventually expect $t_k = 1$.

The Modified Newton Method

If $\nabla^2 f(x_k) \not\succ 0$, it is still possible that d_k be a descent direction. But if it isn't, we modifiy the Hessian and solve the modified system

$$(\nabla^2 f(x_k) + E_k)d = -\nabla f(x_k),$$

where E_k can be

- a multiple of the identity to obtain a positive-definite system
  - simplest strategy to implement
  - costly: a new system must be solved after each attempt
- implicit from a modified factorization of $\nabla^2 f(x_k)$
  - succeeds at the first attempt
  - can consume large amounts of memory
  - $\nabla^2 f(x_k)$ must be explicity available as a matrix.

To retain convergence, we must ensure that $\text{cond}(\nabla^2 f(x_k) + E)$ is bounded.

Overview of Quasi-Newton Methods

If $\nabla^2 f(x)$ is not available, too costly, or modifying it is not feasible, ...

Algorithm 5 Quasi-Newton Method with Armijo Linesearch

1: Choose an initial guess $x_0 \in \mathbb{R}^n$, set $k = 0$.
2: If $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, stop.
3: Choose $B_k = B_k^T \succ 0$ and solve $B_k d_k = -\nabla f(x_k)$ for d_k.
4: Obtain t_k from an Armijo linesearch along d_k.
5: Set $x_{k+1} = x_k + t_k d_k$, $k \leftarrow k + 1$, return to Step 2.

- If $B_k \succ 0$, d_k is always a descent direction
- if $B_k \succ 0$, d_k solves the problem

$$\underset{d \in \mathbb{R}^n}{\text{minimize}} \ \nabla f(x_k)^T d + \tfrac{1}{2} d^T B_k d$$

- how to choose B_k? (Later ...)

Properties of Quasi-Newton Methods

Theorem

Suppose $f \in \mathcal{C}^2$ and $\{x_k\}$ is generated using a Wolfe linesearch with $\alpha \leq \frac{1}{2}$. Suppose $\{x_k\} \to x^*$ where $\nabla f(x^*) = 0$ and $H := \nabla^2 f(x^*) \succ 0$, and B_k satisfies the Dennis-Moré condition

$$\lim_{k \to \infty} \frac{\|(B_k - \nabla^2 f(x_k))d_k\|}{\|d_k\|} = 0.$$

Then

1. *for all sufficiently large k, $t_k = 1$ is accepted*
2. $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$.

Notes:

- The result applies to Newton's method if $\nabla^2 f(x_k) \succ 0$
- typically $\alpha \approx 10^{-4}$
- the Dennis-Moré condition is necessary and sufficient for superlinear convergence
- several choices of B_k are known to satisfy it.

A Few Details on Quasi-Newton Approximations

The Hessian $\nabla^2 f(x_k)$ satisfies

$$\nabla^2 f(x_k) s_k = y_k + o(s_k),$$

where $s_k := x_k - x_{k-1}$ and $y_k := \nabla f(x_k) - \nabla f(x_{k-1})$.

At iteration k, the idea is to look for $B_k = B_k^T \approx \nabla^2 f(x_k)$ that satisfies

$$B_k s_k = y_k,$$

called the secant equation.

Here, there are $n(n+1)/2$ unknowns. Even if we impose $B_k \succ 0$, there remain $n(n-1)/2$ unknowns.

A possibility is to update B_k from B_{k-1} to satisfy the secant equation and an optimality property. Typically, $B_k = B_{k-1} + \text{rank}(1)$ or $\text{rank}(2)$.

Quasi-Newton Methods: Main Idea

The general idea is to obtain B_k as the solution of an optimization problem:

$$\underset{B}{\text{minimize}} \ \tfrac{1}{2}\|B - B_{k-1}\|_W \quad \text{subject to } \ B = B^T, \ Bs_k = y_k,$$

where $\|\cdots\|_W$ is a well-chosen norm.

The classic choice of W leads to the DFP formula, of the form $B_k = B_{k-1} + \text{rank}(2)$. Unfortunately, DFP does not perform well in practice.

A superior choice arises when the problem is formulated in terms of $H := B^{-1}$:

$$\underset{H}{\text{minimize}} \ \tfrac{1}{2}\|H - H_{k-1}\|_W \quad \text{subject to } \ H = H^T, \ Hy_k = s_k,$$

which leads to the BFGS update formula.

The BFGS Update Formula

The solution of the previous problem can be shown to be

$$H_k = (I - \rho_k s_k y_k^T) H_{k-1} (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \qquad \rho_k := 1/y_k^T s_k.$$

It is interesting that we have derived an update of the *inverse* Hessian because

$$B_k d_k = -\nabla f(x_k) \quad \text{becomes} \quad d_k = H_k \cdot (-\nabla f(x_k)).$$

It is possible to show that

$$B_k = B_{k-1} - \frac{(B_{k-1}s_k)(B_{k-1}s_k)^T}{s_k^T B_{k-1} s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

(see the exercices).

In addition, if $B_{k-1} \succ 0$ and $y_k^T s_k > 0$, then $B_k \succ 0$.

A Property on Convex Quadratic Problems

Theorem

Suppose we apply the (inverse) BFGS method to

$$\underset{x}{\text{minimize}} \ -b^T x + \tfrac{1}{2} x^T A x,$$

where $A = A^T \succ 0$ from any x_0 and using any $B_0 = B_0^T \succ 0$. Assume that an exact linesearch is performed at each step. Then

1. *the method converges in at most n iterations,*
2. *the secant equation is satisfied for all previous search directions:*

$$B_k s_j = y_j, \quad j = 1, 2, \ldots, k,$$

3. *if $B_0 = I$, the iterates are identical to those generated by the conjugate gradient method and the search directions are conjugate, i.e., $s_k^T A s_j = 0$ for $k \neq j$,*
4. *$B_n = A$.*

Global Convergence

Theorem

Assume $f \in C^2$ and x_0 is chosen so the level set

$$\mathcal{L}_0 := \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$$

is convex. Assume also that there are constants $M > m > 0$ such that for all $x \in \mathcal{L}_0$,

$$m \|z\|^2 \leq z^T \nabla^2 f(x) z \leq M \|z\|^2 \qquad \text{for all } z.$$

Then

1. *f has unique minimizer $x^* \in \mathcal{L}_0$,*
2. *$\{x_k\} \to x^*$,*
3. *the convergence is superlinear if $\nabla^2 f(x*) \succ 0$.*

Note: BFGS satisfies the Dennis-Moré condition!

The Problem with Quasi-Newton Udpates

The main issue is that if a has many nonzero entries, aa^T is nearly dense. There is no reason why $B_{k-1}s_k$ and y_k should have many zeros.

For a large sparse problem, approximating $\nabla^2 f(x_k)$ with a dense matrix seems unreasonable. It is also inefficient!

But why should we include in B_k the influence of old pairs $\{s_i, y_i\}$ for $i \ll k$?

The Limited-Memory BFGS Update

We have seen that

$$H_k = V_k^T H_{k-1} V_k + \rho_k s_k s_k^T, \qquad V_k := I - \rho_k y_k s_k^T, \quad \rho_k := 1/y_k^T s_k.$$

We can apply the update recursively m times:

$$\begin{aligned}
H_k &= V_k^T V_{k-1}^T H_{k-2} V_{k-1} V_k + \rho_{k-1} V_k^T s_{k-1} s_{k-1}^T V_k + \rho_k s_k s_k^T \\
&= \dots \\
&= (V_k^T \dots V_{k-m+1}^T) H_{k-m} H_k^0 \quad (V_{k-m+1} \dots V_k) \\
&\quad + \rho_{k-m+1} (V_k^T \dots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \dots V_k) \\
&\quad + \dots \\
&\quad + \rho_{k-1} V_k^T s_{k-1} s_{k-1}^T V_k \\
&\quad + \rho_k s_k s_k^T.
\end{aligned}$$

Computing a Matrix-Vector Product

Based on the previous formula, the following algorithm computes a product $H_k v$.

Algorithm 6 Two-Loop Recursion

$q = v$
for $i = k, k-1, \ldots, k-m+1$ **do**
    $\alpha_i = \rho_i s_i^T q$
    $q = q - \alpha_i y_i$
$r = H_k^0 q$
for $i = k-m+1, k-m+2, \ldots, k$ **do**
    $\beta = \rho_i y_i^T r$
    $r = r + s_i(\alpha_i - \beta)$
return r

General Strategy

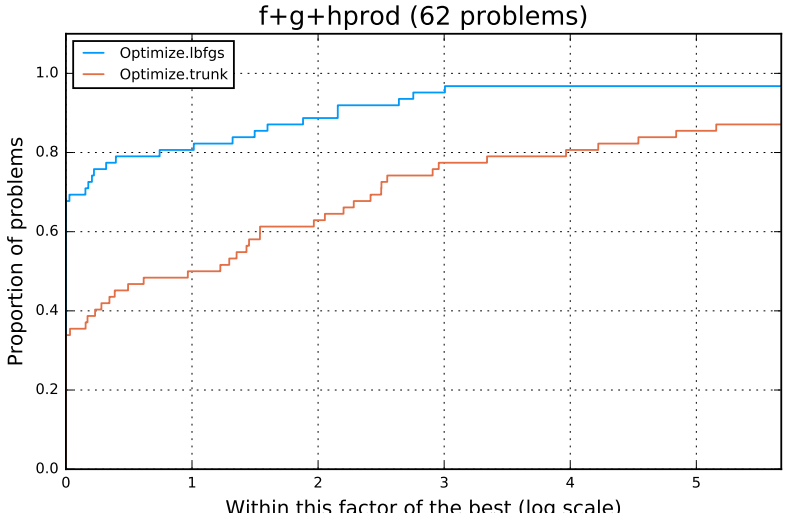
Small values of m are generally effective ($m = 5$, 10 or 20)

For $k \geq m$, the most recent $\{s, y\}$ pair replaces the oldest one in a circular stack.

Similar updates and formulas exist for limited-memory versions of other quasi-Newton updates.

Comparison on Problems with $\sim$ 100 Variables

L-BFGS with memory 5 vs. Trunk (a kind of modified Newton method).



Symmetric Positive Definite Systems

When we see $Ax = b$ with $A = A^T \succ 0$, we read it as the optimality conditions of

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ -b^T x + \tfrac{1}{2} x^T A x.$$

In optimization, we mentally associate $b \leftarrow -g$ and $A \leftarrow H$ (or $B \approx H$).

- Direct method: Cholesky factorization
  - advantages: usually accurate, fast if good ordering can be found
  - disadvantages: can be memory intensive, need A explicitly
- iterative method: conjugate gradient
  - advantages: cheap, does not require A explcitly
  - disadvantages: speed and accuracy depend on a good preconditioner.

Cholesky Factorization

If $A = A^T \succ 0$, it may be written $A = LDL^T$ with

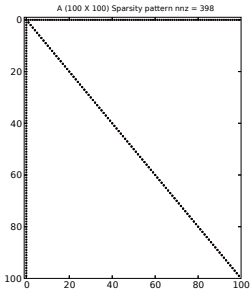
- L unit lower triangular (i.e., $\ell_{ii} = 1$)
- D diagonal and positive definite.

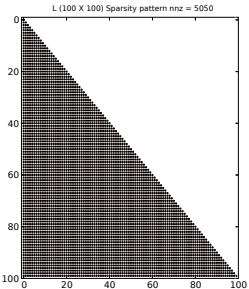
Algorithm 7 Cholesky Factorization: *LDL* Version

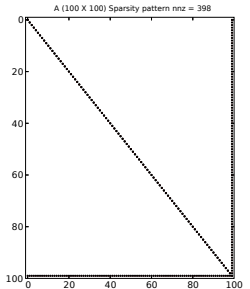
1: **for** $i = 1, \ldots, n$ **do**
2: $\quad d_i = a_{ii} - \sum_{k=1}^{i-1} \ell_{ik}^2 d_k$
3: $\quad$ **for** $j = i+1, \ldots, n$ **do**
4: $\quad\quad \ell_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} \ell_{jk} \ell_{ik} d_k \right) / d_i$

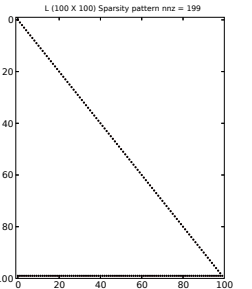
Sparse version must be more sophisticated.

Sparse Cholesky: Importance of Ordering





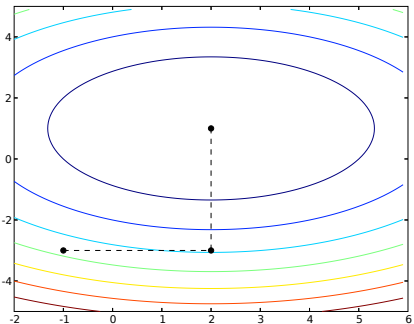


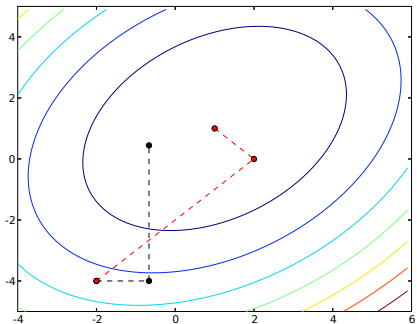


The Conjugate Gradient (CG) Method

Solve $Ax = b$ with $A = A^T \succ 0$, or, equivalently, $\underset{x \in \mathbb{R}^n}{\text{minimize}} \; q(x) := -b^T x + \frac{1}{2} x^T A x$.

Main idea: similar to coordinate search but along well-chosen directions!





It seems possible to solve the problem in at most n steps!

Conjugate Directions

Nonzero vectors $p_1, \ldots, p_k$ are conjugate (with respect to A) if $p_i^T A p_j = 0$ when $i \neq j$.

Example: eigenvectors of A.

Exercise: show that conjugate vectors are necessarily linearly independent.

The main idea of CG is to construct conjugate descent directions one at a time and perform an exact linesearch on $q(x)$:

1. begin with $x_0 := 0$ and $p_0 := b = -\nabla q(x_0)$
2. at each iteration k, denote the optimality residual $r_k := \nabla q(x_k)$
3. given x_k and a descent direction p_k, perform an exact linearch on

$$q(x_k + \alpha p_k) = q(x_k) + \alpha r_k^T p_k + \tfrac{1}{2}\alpha^2 p_k^T A p_k, \quad \text{i.e, } \alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k},$$

4. generate the next conjugate direction in the form $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$.

CG: Algorithm (Hestenes and Stiefel, 1957)

Algorithm 8 The Conjugate Gradient Method

1: $x_0 := 0$, $p_0 := b$, $r_0 := -b$, $k = 0$ // *Start with steepest descent dir*
2: If $\|r_k\| \leq \epsilon_a + \epsilon_r \|r_0\|$, stop
3: $\alpha_k = r_k^T r_k / p_k^T A p_k$ // *Exact linesearch*
4: $x_{k+1} = x_k + \alpha_k p_k$ // *Update* x_k
5: $r_{k+1} = r_k + \alpha_k A p_k$ // *Update* $\nabla q(x_{k+1})$
6: $\beta_{k+1} = r_{k+1}^T r_{k+1} / r_k^T r_k$
7: $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$ // β_{k+1} *ensures* $p_{k+1}^T A p_k = 0$
8: $k \leftarrow k + 1$ and return to Step 2

Other stopping conditions are possible, e.g., based on the value of $q(x_k)$.

Exercice: explain geometrically why $r_{k+1} \perp r_k$, i.e., $r_{k+1}^T r_k = 0$.

Exercise: find a recurrence to update $q(x_k)$ in the algorithm.

CG: Most Important Properties for Optimization

1. By design, $q(x_k)$ is decreasing
2. $\|r_k\|$ is **not** monotonic!
3. $r_k \perp r_j$ for $j = 0, \ldots, k-1$
4. $p_k^T A p_j = 0$ for $j = 0, \ldots, k-1$
5. in exact arithmetic, the search ends after at most n iterations
6. in practice, CG can require much more than n iterations if A is ill conditioned.

Exercise: $A \succ 0$ if and only if $p_k^T A p_k > 0$ for all $k = 0, \ldots, n-1$.

CG: Rate of Convergence

Theorem

If $\kappa := cond(A)$, then

$$q(x_{k+1}) - q(x^*) \leq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)\left(q(x_k) - q(x^*)\right)$$

(c.f., steepest descent).

Theorem

If A has m distinct eigenvalues, CG terminates in at most m iterations.

For performance, it is important to *precondition* the system to cluster its eigenvalues, i.e., replace $Ax = b$ with the equivalent

$$C^T A C x = C^T b,$$

for a well chosen matrix C.

CG May Fail!

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Application I: Linear Least Squares

It is natural to want to use CG to solve

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\|Ax - b\|^2$$

(which is a convex quadratic) because the first-order optimality conditions are the normal equations:

$$A^T A x = A^T b.$$

That can be a bad idea because $\text{cond}(A^T A) = \text{cond}(A)^2$, though CG often behaves well.

However, the intent is right, and there are more stable implementations of CG that should, be used:

- CGLS (CG on $A^T A x = A^T b$ using only A and never $A^T A$, very simple)
- LSQR (different mechanism but equivalent in exact arithmetic, best option)

(c.f., Cholesky vs. QR).

Why Least Squares: Statistical Estimation

Assume the unknown parameters x are related to the observations b via the linear model

$$Ax = b + \epsilon,$$

where ϵ is a vector of random errors. In the *standard linear model*, we assume that

$$\mathbb{E}(\epsilon) = 0, \qquad \mathbb{V}(\epsilon) = \sigma^2 I.$$

Theorem (Gauss-Markov)

If A is m-by-n of rank n, then the minimum-variance unbiased estimator of any linear function $c^T x$ of the unknown parameters is $c^T \hat{x}$ where $\hat{x}$ is the unique solution of the linear least-squares problem.
Moreover, $\mathbb{V}(\hat{x}) = \sigma^2 (A^T A)^{-1}$.

Application II: Inexact Newton Methods

Main idea: $\nabla^2 f(x_k) d_k \approx -\nabla f(x_k)$.

In the local regime, no linesearch is performed: $x_{k+1} = x_k + d_k$.

Theorem

Assume $f \in C^2$, $\nabla f(x^*) = 0$, and $\nabla^2 f(x^*)$ is positive definite. Suppose d_k is computed according to

$$\nabla^2 f(x_k) d_k \approx -\nabla f(x_k), \qquad \|\nabla^2 f(x_k) d_k + \nabla f(x_k)\| \leq \eta_k \|\nabla f(x_k)\|,$$

where $0 < \eta_k \leq \bar{\eta} < 1$ for all k. Then

1. $\{x_k\} \to x^*$,
2. *the convergence is superlinear if $\{\eta_k\} \to 0$,*
3. *the convergence is quadratic if $\eta_k = O(\|\nabla f(x_k)\|)$.*

In the global regime: d_k must be a descent direction!

Generalization: Nonlinear Conjugate Gradient Methods

Main idea: imitate CG for $\min_{x \in \mathbb{R}^n} f(x)$.

Main differences:

1. the residual r_k becomes $\nabla f(x_k)$;
2. the exact linesearch is replaced with a Wolfe linesearch ;
3. the computation of β_{k+1} can be done in various ways:

$$\beta_{k+1}^{\mathrm{FR}} := \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)} \qquad \text{(Fletcher-Reeves)}$$

$$\beta_{k+1}^{\mathrm{PR}} := \frac{\nabla f(x_{k+1})^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla f(x_k)^T \nabla f(x_k)} \qquad \text{(Polak-Ribière)}$$

$$\beta_{k+1}^{\mathrm{PR}+} := \max(\beta_{k+1}^{\mathrm{PR}}, 0) \qquad \text{(Polak-Ribière+)}$$

$$\ldots$$

Nonlinear Least Squares

We write the problem as

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \ \frac{1}{2}\sum_{j=1}^{m} f_j(x)^2.$$

Define

$$F(x) := \begin{bmatrix} f_1(x_1,\ldots,x_n) \\ \ldots \\ f_m(x_1,\ldots,x_m) \end{bmatrix} \in \mathbb{R}^m \qquad J(x) := \begin{bmatrix} \nabla f_1(x)^T \\ \ldots \\ \nabla f_m(x)^T \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Then,

$$f(x) = \tfrac{1}{2}\|F(x)\|^2,$$

$$\nabla f(x) = \sum_{j=1}^{n} f_j(x)\nabla f_j(x) = J(x)^T F(x),$$

$$\nabla^2 f(x) = \sum_{j=1}^{n} \nabla f_j(x)\nabla f_j(x)^T + \sum_{j=1}^{m} f_j(x)\nabla^2 f_j(x) = J(x)^T J(x) + \sum_{j=1}^{m} f_j(x)\nabla^2 f_j(x)$$

The Gauss-Newton Method

Newton's method requires $\nabla^2 f_j(x)$ for $j = 1, \ldots, m$.

When we evaluate $J(x)$, we have the $J(x)^T J(x)$ part of $\nabla^2 f(x)$ for free!

The Gauss-Newton method consists in using $\nabla^2 f(x) \approx J(x)^T J(x)$.

At iteration k, we solve the subproblem

$$\underset{d \in \mathbb{R}^n}{\text{minimize}} \ \ \tfrac{1}{2} \|J(x_k)d + F(x_k)\|^2.$$

Exercice: the result is always a descent direction, unless x_k is stationary.

Typically, $m > n$ but not always.

Convergence of Gauss-Newton

We expect good convergence properties when $F(x^*) \approx 0$ or F is nearly linear.

Theorem

Assume each $f_j \in C^2$ and $J(x)$ has uniform full rank. Then if $\{x_k\}$ is generated from the Gauss-Newton method with Wolfe linesearch, we have

$$\lim_{k \to \infty} \nabla f(x_k) = 0.$$

The rank condition is the most limitating. We will resolve it later with trust-region methods.

Problems with Large Residuals

Gauss-Newton does not perform well when $F(x^*) \neq 0$ or F is highly nonlinear.

One solution is to use a quasi-Newton approximation

$$\nabla^2 f(x_k) = J(x_k)^T J(x_k) + \sum_{j=1}^{m} f_j(x_k) \nabla^2 f_j(x_k) \approx J(x_k)^T J(x_k) + B_k.$$

Taking advantage of the structure requires structured quasi-Newton updates.

The method NL2SOL of Gay, Dennis and Welsch set B_k using a BFGS-like formula.

Trust-Region Methods

Instead of searching in a single well-chosen direction, trust-region method allow exploration of the whole space around x_k but control the maximum step size:

1. we construct a model of $f(x_k + s)$ (typically quadratic but not always):

$$f(x_k + s) \approx m_k(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \tfrac{1}{2} s^T B_k s,$$

   where $B_k \approx \nabla^2 f(x_k)$,
2. the model is *trusted* in a region $\{x_k + s \mid \|s\| \leq \Delta_k\}$ around x_k
3. we compute a step as an (approximate) solution of

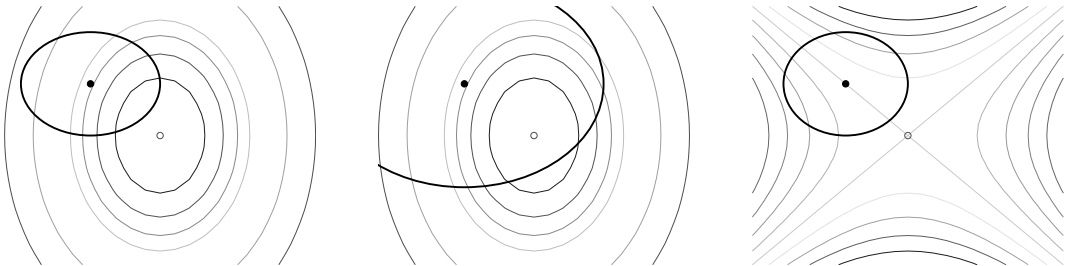
$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \;\; m_k(x_k + s) \quad \text{subject to} \;\; \|s\| \leq \Delta_k,$$

4. depending on the improvement in f achieved along s, we either accept or reject the step and update Δ_k.

Remarks on the Trust-Region Subproblem

1. The subproblem is constrained;
2. $B_k = 0$ is allowed;
3. if $m_k(x_k + s)$ is convex, the subproblem is convex;
4. however, using nonconvex models poses no difficulty;
5. different norms may be used depending on the context: typically $\|\cdot\|_2$ and $\|\cdot\|_\infty$;
6. when using $\|\cdot\|_2$ we know how to find a *global* minimizer, even when $m_k(x_k + s)$ is not convex;
7. the conjugate-gradient method is well-suited to solve the subproblem!

Three Main Situations

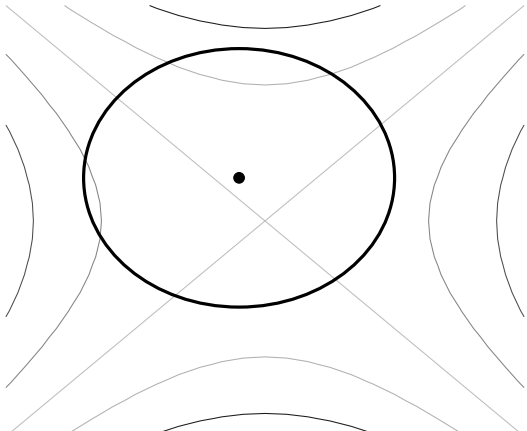


(a) convex model, unconstrained solution outside trust region;

(b) convex model, unconstrained solution inside trust region;

(c) nonconvex model.

One Potential Advantage on Linesearch Methods



Escape from saddle points if nonconvex models are allowed!

A Basic Trust-Region Algorithm

Algorithm 9 Basic Trust-Region Algorithm

1: Choose an initial guess $x_0 \in \mathbb{R}^n$, $\Delta_0 > 0$, set $k = 0$.

2: If $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, stop.

3: Choose B_k and compute a solution s_k of: $\underset{s \in \mathbb{R}^n}{\text{minimize}}\ m_k(x_k + s)$ subject to $\|s\| \leq \Delta_k$.

4: Compute the ratio

$$\rho_k := \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

5: Set

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k \geq 10^{-4} \\ x_k & \text{otherwise} \end{cases}, \qquad \Delta_{k+1} = \begin{cases} 3\Delta_k & \text{if } \rho_k \geq 0.99 \\ \frac{1}{3}\Delta_k & \text{if } \rho_k < 10^{-4} \\ \Delta_k & \text{otherwise.} \end{cases}$$

6: $k \leftarrow k + 1$ and return to Step 2.

Computing a Step

To ensure convergence to a stationary point, the step s_k must satisfy the *sufficient decrease* condition

$$m_k(x_k) - m_k(x_k + s_k) \geq \tfrac{1}{2}\|\nabla f_k\| \min\left(\frac{\|\nabla f_k\|}{1 + \|B_k\|}, \Delta_k\right),$$

similar in spirit to the Armijo condition.

The condition is satisfied with

1. the minimizer of $m_k(x_k + s)$ in the steepest descent direction and inside the trust region;
2. the global minimizer of $m_k(x_k + s)$ in the trust region.

Characterization of a Global Minimizer: ℓ_2 Norm

Theorem (Gay and Sorensen)

A step s is a global minimizer of $m_k(x_k + s)$ subject to $\|s\|_2 \leq \Delta_k$ if and only if

$$(B_k + \lambda I)s = -\nabla f(x_k),$$

where $\lambda \geq 0$ is such that

1. $B_k + \lambda I \succeq 0$, *and*
2. $\lambda(\|s\|_2 - \Delta_k) = 0$.

If $B_k + \lambda I \succ 0$, the global minimizer is unique.

The trust-region subproblem is one of the rare nonconvex optimization problems for which we have necessary and sufficient conditions for *global* optimality.

The theorem is the basis for the method of Moré and Sorensen to identify the global minimizer.

The Truncated Conjugate Gradient Method: Main Idea

Apply CG to $\underset{s \in \mathbb{R}^n}{\text{minimize}}\ m_k(x_k + s)$ unconstrained.

Important properties of CG:

1. the first step occurs in the steepest descent direction,
2. $m_k(x_k + s_{j+1}) \leq m_k(x_k + s_j)$, $j = 1, 2, \ldots$ as long as $p_j^T B_k p_j > 0$,
3. $\|s_{j+1}\|_2 \geq \|s_j\|_2$, $j = 1, 2, \ldots$ as long as $p_j^T B_k p_j > 0$,

Consequences:

1. the sufficient decrease condition is certainly satisfied!
2. Subsequent steps further decrease the model,
3. if the CG iterates leave the trust region, they will never return.

We stop only if $\nabla_s mk(x_k + s) = 0$ or we cross the trust-region boundary.

The Truncated Conjugate Gradient Method

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ m_k(x_k+s)\ \text{ subject to }\ \|s\|_2 \le \Delta_k$$

Algorithm 10 The Truncated Conjugate Gradient Method

1: $s_0 := 0$, $p_0 := -\nabla f(x_k)$, $r_0 := \nabla f(x_k)$, $j = 0$ *// Start with steepest descent dir*
2: If $\|r_j\| \le \epsilon_a + \epsilon_r\|r_0\|$, stop
3: compute $\gamma > 0$ such that $\|s_j + \gamma p_j\|_2 = \Delta_k$ *// Step to boundary*
4: if $p_j^T B_k p_j \le 0$, set $s_{j+1} = s_j + \gamma p_j$ and exit *// Nonconvex model*
5: $\alpha_j = r_j^T r_j / p_j^T B_k p_j$ *// Exact linesearch*
6: if $\alpha_j > \gamma$, set $s_{j+1} = s_j + \gamma p_j$ and exit *// CG iterate is outside*
7: $s_{j+1} = s_j + \alpha_j p_j$
8: $r_{j+1} = r_j + \alpha_j B_k p_j$
9: $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$
10: $p_{j+1} = -r_{j+1} + \beta_{j+1} p_j$
11: $j \leftarrow j+1$ and return to Step 2

Main Property in the Convex Case

Theorem

Assume $B_k \succeq 0$. Let s_k^{CG} be the iterate returned by the truncated conjugate-gradient method and s_k^* be a global minimizer of

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \;\; m_k(x_k + s) \;\; \text{subject to} \;\; \|s\|_2 \leq \Delta_k.$$

Then

$$m_k(x_k) - m_k(x_k + s_k^{CG}) \geq \tfrac{1}{2} \left(m_k(x_k) - m_k(x_k + s_k^*) \right).$$

The Method of Levenberg and Marquardt

In nonlinear least-squares problems where $J(x)$ may be rank deficient, the method of Levenberg and Marquardt solves the trust-region subproblem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\|J(x_k)s + F(x_k)\|^2 \quad \text{subject to } \|s\| \leq \Delta_k.$$

According to the Gay and Sorensen theorem, either the solution is interior or there is $\lambda > 0$ such that

$$\left(J(x_k)^T J(x_k) + \lambda I\right) s = -J(x_k)^T F(x_K),$$

which are the optimality conditions of

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\left\| \begin{bmatrix} J(x_k) \\ \sqrt{\lambda} I \end{bmatrix} s + \begin{bmatrix} F(x_k) \\ 0 \end{bmatrix} \right\|^2.$$

Here we would use CGLS or LSQR in place of CG.