SWFR ENG 4TE3 (6TE3)  
COMP SCI 4TE3 (6TE3)  
Continuous Optimization Algorithms  
Computing and Software  
McMaster University  

January 2015
Generic Algorithm

\[ \min \ f(x) \]
\[ \text{s.t. } \quad x \in C. \]

**Input:**
\( \epsilon > 0 \) is the accuracy parameter;
\( x^0 \) is a given (relative interior) feasible point;

**Step 0:** \( x := x^0, \ k = 0; \)

**Step 1:** Find search direction \( s^k \)
\[ \text{s.t. } \quad \delta f(x^k, s^k) < 0; \]
(this should be a descending feasible direction in the constrained case.)

**Step 1a:** If no such direction exists \textbf{STOP}, optimum found.

**Step 2:** Line search : find \( \lambda^k = \min \lambda f(x^k + \lambda s^k); \)

**Step 3:** \( x^{k+1} = x^k + \lambda^k s^k, \quad k = k + 1; \)

**Step 4:** If stopping criteria satisfied \textbf{STOP},
else GOTO Step 1.
Search direction

Gradient method

\[ s = -\nabla f(x^k) \]

**Steepest descent direction!**

\[
\delta f(x, -\nabla f(x)) = -\nabla f(x)^T \nabla f(x) \\
= \min_{||s||=||\nabla f(x)||} \{ \nabla f(x)^T s \}.
\]

**Cost of one iteration:** \( O(n) + \) line search.

The (negative) gradient is orthogonal to the level curves.
Not a finite algorithm, even not for quadratic functions.
Slow convergence, “zigg–zagging” is possible.
Theorem 1 Let $f$ be continuously differentiable. Starting from the initial point $x^0$ and using an exact line search, the gradient method produces a decreasing sequence $x^0, x^1, x^2, \ldots$ such that $f(x^k) > f(x^{k+1})$ for $k = 0, 1, 2, \ldots$. Assume that the level set $D = \{x : f(x) \leq f(x^0)\}$ is compact, then any accumulation point $\bar{x}$ of the generated sequence $x^0, x^1, x^2, \ldots, x^k, \ldots$ is a stationary point (i.e. $\nabla f(\bar{x}) = 0$) of $f$. If the function $f$ is convex, then $\bar{x}$ is a global minimizer of $f$.

Proof: Since $D$ is compact and $f$ is continuous, the function $f$ is bounded on $D$. Hence we have a convergent subsequence $x^{k_j} \rightarrow \bar{x}$ with $f(x^{k_j}) \rightarrow f^*$ as $k_j \rightarrow \infty$. Thus, $f(\bar{x}) = f^*$ by continuity of $f$. Since the search direction is the gradient of $f$, we have:

$$\bar{s} = \lim_{k_j \rightarrow \infty} s^{k_j} = -\lim_{k_j \rightarrow \infty} \nabla f(x^{k_j}) = -\nabla f(\bar{x}).$$

Composing with $\nabla f(\bar{x})$ yields:

$$\bar{s}^T \nabla f(\bar{x}) = -\nabla f(\bar{x})^T \nabla f(\bar{x}) \leq 0. \quad (*)$$

On the other hand, using the construction of the iteration sequence and the convergent subsequence, we can write

$$f(x^{k_{j+1}}) \leq f(x^{k_j+1}) \leq f(x^{k_j} + \lambda s^{k_j}).$$

Taking the limit in the last inequality yields:

$$f(\bar{x}) \leq f(\bar{x} + \lambda \bar{s})$$

which leads to $\delta f(\bar{x}, \bar{s}) = \bar{s} \nabla^T f(\bar{x}) \geq 0$. Thus, combining this result with $(*)$, $\nabla f(\bar{x}) = 0$; that is, the theorem holds. $\square$
The order of convergence

The order of convergence is only linear, speed depends on the conditioning of the Hessian. Let \( q(x) = \frac{1}{2}x^TAx - b^Tx \) with \( A : n \times n \) symmetric and positive definite, \( b \in R^n \).

Let \( 0 < \mu_1 \leq \mu_2 \leq \cdots \leq \mu_n \) be the eigenvalues of \( A \).

Let \( r = \frac{\mu_n}{\mu_1} \) (condition number of \( A \), degree of difficulty).

Let \( E(x) = q(x) - q(x^*) \) where \( x^* \) minimizes the function \( q(x) \); that is, \( E(x) = \frac{1}{2}(x - x^*)^TA(x - x^*) \)

**Theorem 2** Let \( x^0 \) be an arbitrary starting point of the steepest decent method applied to minimize \( E(x) \). Then the steepest descent method converges to the unique minimum \( x^* \) of \( E(x) \) (and \( q(x) \)), further

\[
E(x^{k+1}) \leq \left( \frac{r-1}{r+1} \right)^2 E(x^k) = \left( \frac{\mu_n - \mu_1}{\mu_n + \mu_1} \right)^2 E(x^k).
\]
Newton’s method

\( f \) is twice continuously differentiable and \textit{strictly} convex.

Newton’s method is based on minimizing the second order approximation of \( f \).

\[
q(x) := f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T \nabla^2 f(x^k)(x - x^k).
\]

The Hessian \( \nabla^2 f(x^k) \) is positive definite (PD), thus \( q(x) \) is strictly convex. Hence the minimum is attained when

\[
\nabla q(x) = \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0.
\]

Thus,

\[
x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k).
\]

\textbf{Local quadratic convergence with full step (without any line search!).}

Good starting point is essential.

If line search is applied \( \rightarrow \) \textit{damped} Newton method.

\textbf{Cost of one iteration:} \( O(n^3) \) + line search.

To reduce computational costs:

\textit{quasi-Newton} methods,

\textit{methods of conjugate directions}.

Hessian not PD: then do \textit{Trust-region method}. 
Descent directions (once more)

Hessian/modified Hessian

At a given point \( x \) in the directions \( s \) the directional derivative is:

\[
\delta f(x, s) = \nabla f(x)^T s.
\]

Let \( s = -H \nabla f(x) \) (like in Newton’s method), then \( s \) is a descent direction, i.e.,

\[
\delta f(x, s) = \nabla f(x)^T s = -\nabla f(x)^T H \nabla f(x) < 0
\]

iff \( H \) is positive definite. That is why a positive definite Hessian is needed for Newton’s method.

**Observe:** ANY positive definite matrix \( H \) gives a descent direction!

- If \( H = I \) then we got the gradient method.
- If \( H = \left( \nabla^2 f(x) \right)^{-1} \) then we have Newton’s method.
- If \( H = \left( \nabla^2 f(x) + \alpha I \right)^{-1} \) then we got the Trust Region method.
If the function \( f(x) \) is not strictly convex, or if the Hessian is ill-conditioned the Hessian is not (or hardly) invertible.

Remedy: trust-region method.

Quadratic approximation of the function \( f(x) \):

\[
q(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)^T (x - x^k).
\]

If the Hessian is not PD, we perturb it to PD:

\[
Q(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \left( \nabla^2 f(x^k)^T + \alpha I \right) (x - x^k).
\]

\[
Q(x) = q(x) + \frac{\alpha}{2} \| x - x^k \|^2
\]

The minimum of this (observe \( \nabla^2 f(x^k) \) is replaced by \( (\nabla^2 f(x) + \alpha I) \) comparing Newton);

\[
s^k(\alpha) = x^{k+1} - x^k = - \left( \nabla^2 f(x^k) + \alpha I \right)^{-1} \nabla f(x^k).
\]

Then \( x^k + s^k(\alpha) \) minimizes \( Q(x) \), but we want a decrease in \( q(x) \) as well:

\[
Q(x^k + s^k(\alpha)) \leq Q(x)
\]

\[
q(x^k + s^k(\alpha)) + \frac{\alpha}{2} \| s^k(\alpha) \|^2 \leq q(x) + \frac{\alpha}{2} \| x - x^k \|^2
\]

\[
q(x^k + s^k(\alpha)) \leq q(x) \quad \text{if} \quad \| x - x^k \| \leq \| s^k(\alpha) \| =: r_\alpha
\]
\[ r_\alpha := \| s^k(\alpha) \| = \| - \left( \nabla^2 f(x^k) + \alpha I \right)^{-1} \nabla f(x^k) \| \]

**Lemma 1** \( r_\alpha \) is a non-increasing function of \( \alpha \).

**Proof:** Let \( z_1, \ldots, z_n \) be an orthonormal eigenvector system of \( \nabla^2 f(x^k) \) with the ordered eigenvalues \( \mu_1 \leq \cdots \leq \mu_n \). Further, let \( \nabla f(x^k) = \sum_{i=1}^{n} c_i z_i \).

Then
\[
s^k(\alpha) = - \sum_{i=1}^{n} \frac{c_i}{\mu_i + \alpha} z_i
\]

\[
r_\alpha = \sqrt{\sum_{i=1}^{n} \frac{c_i^2}{(\mu_i + \alpha)^2}}
\]

is clearly non-increasing with \( \alpha \) increasing. \( \square \)

The bigger the \( \alpha \), the smaller the step!
The Trust-Region (TR) direction is a compromise between the gradient and Newton directions:

\[-\nabla f(x^k) - (\nabla^2 f(x^k) + \alpha I)^{-1} \nabla f(x^k)\]

If \( \alpha = 0 \) then we have the Newton step, as \( \alpha \to \infty \) then we approach a small multiple of the negative gradient.
Trust region method - IV

The update of $\alpha$:
Let $x^k$ and $\alpha > 0$ be the current iterate.
Let $0 < \mu < \eta < 1$ and $0 < \gamma_1 < 1 < \gamma_2$ be given.
Typical values: $\mu = \frac{1}{4}$, $\eta = \frac{3}{4}$, $\gamma_1 = \frac{1}{2}$, $\gamma_2 = 2$.
A usual starting value is $\alpha = 1$.

If $x^k$ satisfies termination criteria, then STOP.

Calculate $s^k(\alpha)$.

Compute
\[
\rho^k = \frac{f(x^k) - f(x^k + s^k(\alpha))}{f(x^k) - q(x^k + s^k(\alpha))} = \frac{\text{actual decrease}}{\text{predicted decrease}}
\]

If $\rho^k < \mu$ then (step failed)
\[
x^{k+1} := x^k \text{ and } \alpha := \gamma_2 \alpha.
\]

If $\mu \leq \rho^k \leq \eta$ then (step as predicted)
\[
x^{k+1} := x^k + s^k(\alpha) \text{ and } \alpha := \alpha.
\]

If $\rho^k > \eta$ then (step is very good)
\[
x^{k+1} := x^k + s^k(\alpha) \text{ and } \alpha := \gamma_1 \alpha.
\]

In order to avoid exact line search
$\alpha$ is dynamically increased and decreased.
Newton step was given by minimizing the quadratic approximation:
\[ q(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T \nabla^2 f(x^k)^T (x - x^k). \]

How far you TRUST that \( q(x) \) is a good approximation of \( f(x) \)? Not far, thus you replace the Newton sub-problem by the TR subproblem

\[
\begin{aligned}
\min & \quad q(x) \\
\text{s.t.} & \quad \| x - x^k \|^2 \leq \Delta_k^2.
\end{aligned}
\]

As we will see later (duality, Lagrange function) the optimum of this problem is obtained by minimizing the Lagrange function

\[
Q(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T (\nabla^2 f(x^k)^T + \alpha I)(x - x^k),
\]

where \( \alpha \) is an appropriate Lagrange multiplier. This results in

\[
x^{k+1} = x^k - \left( \nabla^2 f(x^k) + \alpha I \right)^{-1} \nabla f(x^k).
\]

It can be verified:
The bigger the \( \Delta_k \) is, the smaller \( \alpha \) is.

Thus if you solve the TR subproblem directly in a TR algorithm, you should decrease \( \Delta_k \) when \( \alpha \) is increased and increase \( \Delta_k \) when \( \alpha \) is decreased.
Let $LDL^T = (\nabla^2 f(x))^{-1}$, if it exists, where $D$ is a diagonal and $L$ is a lower triangular matrix.

Choose the search direction as follows:

- If the factorization exists and $\text{diag}(D)$ is positive, then use Newton, i.e., $s = -\left(\nabla^2 f(x)\right)^{-1}\nabla f(x)$.
- If the factorization exists and $D$ is not positive, then let $y = -\text{sign}(\text{sign}(\text{diag}(D)) - I)$. Then solve $L^T s = y$ which imply
  $$s^T (\nabla^2)^{-1} f(x) s = s^T LD L^T s < 0.$$  
  If $\delta f(x, s) < 0$ then use $s$;
  If $\delta f(x, -s) < 0$ then use $-s$;
  If $\delta f(x, s) = 0$ then use $s = -\nabla f(x, s)$.

- If the factorization does not exist then use steepest descent, i.e., $s = -\nabla f(x)$. 

Fiacco-McCormick’s modification

Newton-like algorithm for Nonconvex functions
Newton’s method for nonlinear equations:

\[ F(x) = 0 \]

Linearize at \( x^k \):

\[ F(x) \approx F(x^k) + JF(x^k)(x - x^k) \]

\[ JF(x)_{ij} = \frac{\partial F_i(x)}{\partial x_j} \quad \text{where} \quad i = 1, \ldots, m; \quad j = 1, \ldots, n. \]

\[ JF(x^k)(x^{k+1} - x^k) = -F(x^k). \]

Minimize \( f(x) \iff \nabla f(x) = 0. \)

\[ \nabla^2 f(x^k)(x^{k+1} - x^k) = -\nabla f(x^k). \]

The Jacobian of the gradient is exactly the Hessian of the function \( f(x) \) hence it is positive definite and we have

\[ x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1}\nabla f(x^k) \]

as we have seen above.