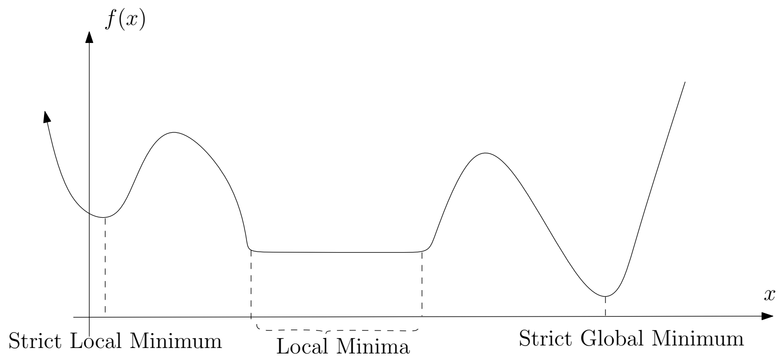
EN530.603 Applied Optimal Control
Lecture 2: Unconstrained Optimization Basics

September 8, 2014

Lecturer: Marin Kobilarov

1 Optimality Conditions

- Find the value of $x \in \mathbb{R}^n$ which minimizes $f(x)$
- We will generally assume that f is at least twice-differentiable
- Local and Global Minima



- Small variations Δx yield a cost variation (using a Taylor's series expansion)

$$f(x^* + \Delta x) - f(x^*) \approx \nabla f(x^*)^T \Delta x \geq 0,$$

to first order, or two second order:

$$f(x^* + \Delta x) - f(x^*) \approx \nabla f(x^*)^T \Delta x + \frac{1}{2}\Delta x^T \nabla^2 f(x^*) \Delta x \geq 0,$$

- Then $\nabla f(x^*) \Delta x \geq 0$ for arbitrary $\Delta x \quad \Rightarrow \quad \nabla f = 0$
- Then $\nabla f = 0 \quad \Rightarrow \quad \frac{1}{2}\Delta x^T \nabla^2 f(x^*) \Delta x \geq 0$ for arbitrary $\Delta x \quad \Rightarrow \quad \nabla^2 f(x^*) \geq 0$

Proposition 1. (Necessary Optimality Conditions) [1] *Let x^* be an unconstrained local minimum of $f : \mathbb{R}^n \to \mathbb{R}$ that it is continuously differentiable in a set S containing x^*. Then*

$$\nabla f = 0 \qquad \textit{(First-order Necessary Conditions)}$$

If in addition, f is twice-differentiable within S then

$$\nabla^2 f \geq 0 : \textit{positive semidefinite} \qquad \textit{(Second-order Necessary Conditions)}$$

Proof: Let $d \in \mathbb{R}^n$ and examine the change of the function $f(x + \alpha d)$ with respect to the scalar α

$$0 \leq \lim_{\alpha \to 0} \frac{f(x^* + \alpha d) - f(x^*)}{\alpha} = \nabla f(x^*)^T d,$$

The same must hold if we replace d by $-d$, i.e.

$$0 \leq -\nabla f(x^*)^T d \quad \Rightarrow \quad \nabla f(u)^T d \leq 0,$$

for all d which is only possible if $\nabla f(u) = 0$.

The second-order Taylor expansion is

$$f(x^* + \alpha d) - f(x^*) = \alpha \nabla f(x^*) d + \frac{\alpha^2}{2} d^T \nabla^2 f(x^*) d + o(\alpha^2)$$

Using $\nabla f(x^*) = 0$ we have

$$0 \leq \lim_{\alpha \to 0} \frac{f(x^* + \alpha d) - f(x^*)}{\alpha^2} = \frac{1}{2} d^T \nabla^2 f(x^*) d,$$

hence $\nabla^2 f$ must be positive semidefinite. □

Note: small-o notation means that $o(g(x))$ goes to zero faster than $g(x)$, i.e. $\lim_{g(x) \to 0} \frac{o(g(x))}{g(x)} = 0$

Proposition 2. (Second Order Sufficient Optimality Conditions) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable in an open set S. Suppose that a vector $x^* \in S$ satisfies the conditions*

$$\nabla f(x^*) = 0, \qquad \nabla^2 f(x^*) > 0 : \textit{positive definite}$$

Then, x^* is a strict unconstrained local minimum of f. In particular, there exist scalars $\gamma > 0$ and $\epsilon > 0$ such that

$$f(x) \geq f(x^*) + \frac{\gamma}{2} \|x - x^*\|^2, \qquad \forall x \quad \textit{with } \|x - x^*\| \leq \epsilon.$$

Proof: Let λ be the smallest eigenvalue of $\nabla^2 f(x^*)$ then we have

$$d^T \nabla^2 f(x^*) d \geq \lambda \|d\|^2 \quad \text{for all} \quad d \in \mathbb{R}^m,$$

The Taylor expansion, and using the fact that $\nabla f(x^*) = 0$

$$\begin{aligned} f(x^* + d) - f(x^*) &= \nabla f(x^*) d + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \\ &\geq \frac{\lambda}{2} \|d\|^2 + o(\|d\|^2) \\ &= \left(\frac{\lambda}{2} + \frac{o(\|d\|^2)}{\|d\|^2} \right) \|d\|^2. \end{aligned}$$

This is satisfied for any $\epsilon > 0$ and $\gamma > 0$ such that

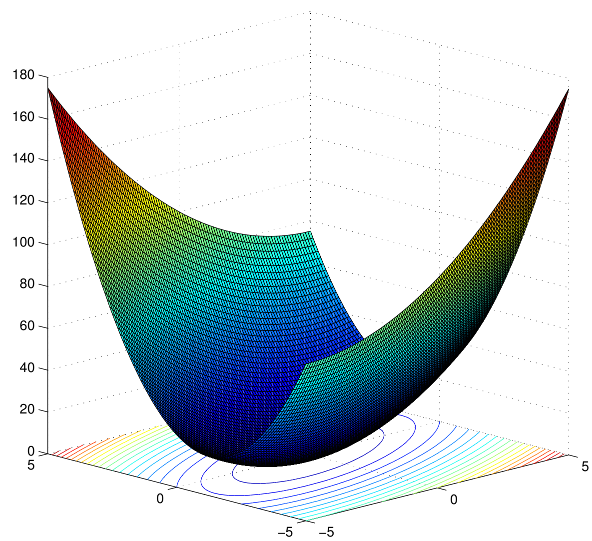
$$\frac{\lambda}{2} + \frac{o(\|d\|^2)}{\|d\|^2} \geq \frac{\gamma}{2}, \qquad \forall d \quad \text{with} \quad \|d\| \leq \epsilon.$$

□

1.1 Examples

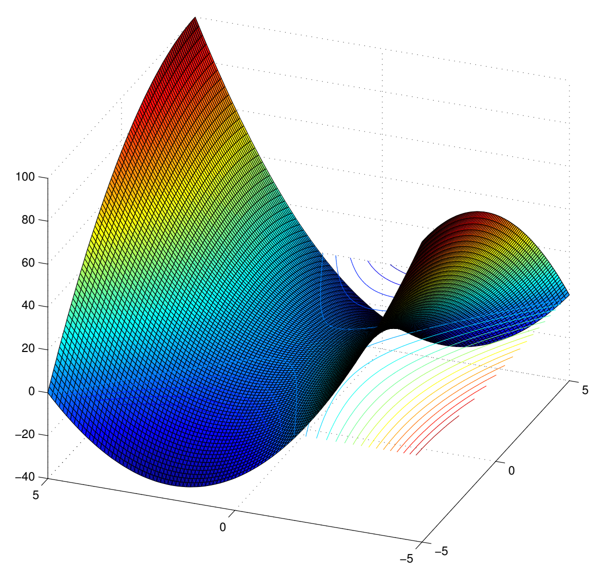
- Convex function with strict minimum

$$f(x) = x^T \begin{bmatrix} 1 & -1 \\ -1 & 4 \end{bmatrix} x$$



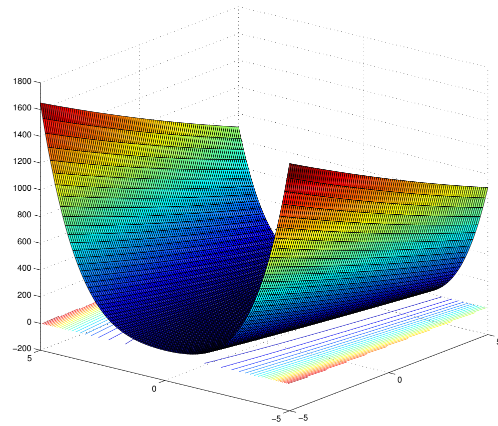
- Saddlepoint: one positive eigenvalue and one negative

$$f(x) = x^T \begin{bmatrix} -1 & 1 \\ 1 & 3 \end{bmatrix} x$$

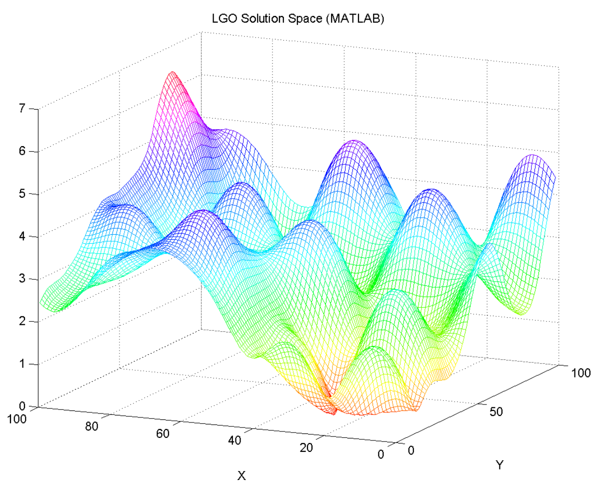


- Singular point: one positive eigenvalue and one zero eigenvalue

$$f(x) = (x_1 - x_2^2)(x_1 - 3x_2^2)$$



- a complicated function with multiple local minima



2 Numerical Solution: gradient-based methods

In general, optimality conditions cannot be solved in closed-form. It is necessary to use an iterative procedure starting with some initial guess $x = x^0$, i.e.

$$x^{k+1} = x^k + \alpha^k d^k, \quad k = 0, 1, \ldots$$

until $f(x^k)$ converges. Here $d^k \in \mathbb{R}^n$ is called the descent *direction* (or more generally "search direction") and $\alpha^k > 0$ is called the *stepsize*. The most common methods for finding α^k and d^k are gradient-based. Some use only first-order information (the gradient only) while other additionally use higher-order (gradient and Hessian) information.

- Gradient-based methods follow the general guidelines:
  1. Choose direction d^k so that whenever $\nabla f(x^k) \neq 0$ we have
  $$\nabla f(x^k)^T d^k < 0,$$
  i.e. the direction and *negative* gradient make an angle $< 90°$
  2. Choose stepsize $\alpha^k > 0$ so that
  $$f(x^k + \alpha d^k) < f(x^k),$$
  i.e. cost decreases
- Cost reduction is guaranteed (assuming $\nabla f(x^k) \neq 0$) since we have
$$f(x^{k+1}) = f(x^k) + \alpha^k \nabla f(x^k)^T d^k + o(\alpha^k)$$
and there always exist α^k small enough so that
$$\alpha^k \nabla f(x^k)^T d^k + o(\alpha^k) < 0.$$

2.1 Selecting Descent Direction d

Descent direction choices

- Many gradient methods are specified in the form
$$x^{k+1} = x^k - \alpha^k D^k \nabla f(x^k),$$
where D^k is positive definite symmetric matrix.
- Since $d^k = -D^k \nabla f(x^k)$ and $D^k > 0$ the descent condition
$$-\nabla f(x^k)^T D^k \nabla f(x^k) < 0,$$
is satisfied.

We have the following general methods:

Steepest Descent

$$D^k = I, \qquad k = 0, 1, \ldots,$$

where I is the identity matrix. We have

$$\nabla f(x^k)^T d^k = -\|\nabla f(x^k)\|^2 < 0, \quad \text{when} \quad \nabla f(x^k) \neq 0$$

Furthermore, the direction $\nabla f(x^k)$ results in the *fastest* decrease of f at $\alpha = 0$ (i.e. near x^k).

Newton's Method

$$D^k = [\partial^2 f(x^k)]^{-1}, \qquad k = 0, 1, \ldots,$$

provided that $\partial^2 f(x^k) > 0$.

- The idea behind Newton's method is to minimize a quadratic approximation of f around x^k

  $$f^k(x) = f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k),$$

  and solve the condition $\nabla f^k(x) = 0$

- This is equivalent to

  $$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0$$

  and results in the Newton iteration

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

Diagonally Scaled Steepest Descent

$$D^k = \begin{pmatrix} d_1^k & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & d_2^k & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & d_{n-1}^k & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & d_n^k \end{pmatrix} \equiv \operatorname{diag}([d_1^k, \cdots, d_n^k]),$$

for some $d_i^k > 0$. Usually these are the inverted diagonal elements of the hessian $\nabla^2 f$, i.e.

$$d_i^k = \left[\frac{\partial^2 f(x^k)}{(\partial x_i)^2}\right]^{-1}, \qquad k = 0, 1, \ldots,$$

Gauss-Newton Method

When the cost has a special *least squares form*

$$f(x) = \frac{1}{2}\|g(x)\|^2 = \frac{1}{2}\sum_{i=1}^{m} (g_i(x))^2$$

we can choose

$$D^k = \left[\nabla g(x^k) \nabla g(x^k)^T\right]^{-1}, \qquad k = 0, 1, \ldots$$

Conjugate-Gradient Methods

Idea is to choose linearly independent (i.e. conjugate) search directions d^k at each iteration. For quadratic problems convergence is guaranteed by at most n iterations. Since there are at most n independent directions, the independence condition is typically reset every $k \leq n$ steps for general nonlinear problems.

The directions are computed according to

$$d^k = -\nabla f(x^k) + \beta^k d^{k-1}.$$

The most common way to compute β^k is

$$\beta^k = \frac{\nabla f(x^k)^T \left(\nabla f(x^k) - \nabla f(x^{k-1})\right)}{\nabla f(x^{k-1})^T \nabla f(x^{k-1})}$$

It is possible to show that the choice β^k ensures the conjugacy condition.

2.2 Selecting Stepsize α

- *Minimization Rule*: choose $\alpha^k \in [0, s]$ so that f is minimized, i.e.

  $$f(x^k + \alpha^k d^k) = \min_{\alpha \in [0,s]} f(x^k + \alpha d^k)$$

  which typically involves a one-dimensional optimization (i.e. a line-search) over $[0, s]$.

- *Successive Stepsize Reduction - Armijo Rule:* idea is to start with initial stepsize s and if $x^k + sd^k$ does not improve cost then s is reduced:

  $$\begin{aligned}
  &\text{Choose: } s > 0,\ 0 < \beta < 1,\ 0 < \sigma < 1 \\
  &\text{Increase: } m = 0, 1, ... \\
  &\text{Until: } f(x^k) - f(x^k + \beta^m s d^k) \geq -\sigma \beta^m s \nabla f(x^k)^T d^k
  \end{aligned}$$

  where β is the rate of decrease (e.g. $\beta = .25$) and σ is the acceptance ratio (e.g. $\sigma = .01$).

- *Constant Stepsize:* use a fixed step-size $s > 0$

  $$\alpha^k = s, \qquad k = 0, 1, \ldots$$

  while simple it can be problematic: too large step-size can result in divergence; too small in slow convergence

- *Diminishing Stepsize:* use a stepsize converging to 0

  $$\alpha^k \to 0$$

  under a condition $\sum_{k=0}^{\infty} \alpha^k = \infty$, x^k will converge theoretically but in practice is slow.

2.3 Example

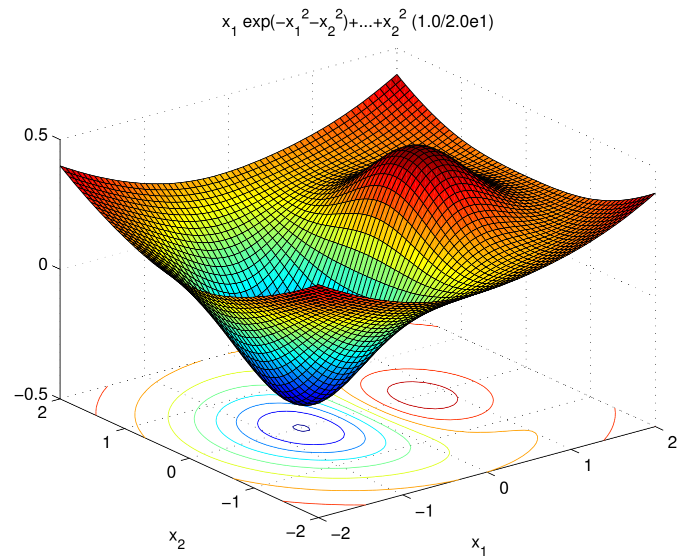
- Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$

  $$f(x) = x_1 \exp(-(x_1^2 + x_2^2)) + (x_1^2 + x_2^2)/20$$

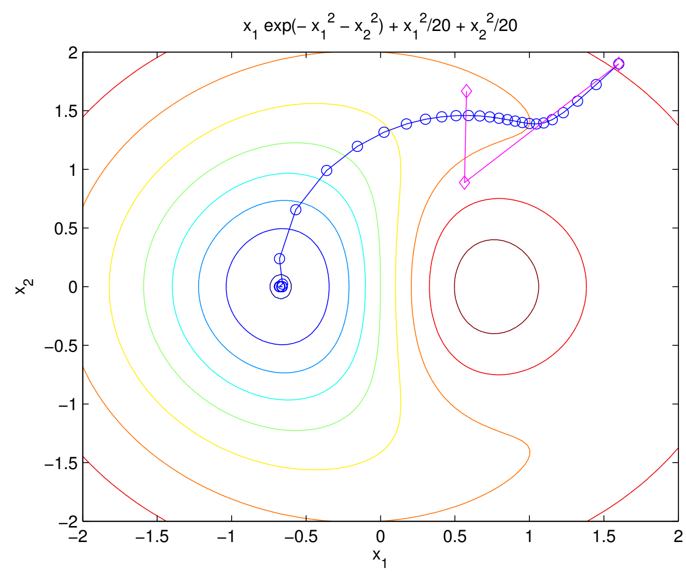
  The gradient and Hessian are

  $$\nabla f(x) = \begin{bmatrix} x_1/10 + \exp(-x_1^2 - x_2^2)(1 - 2x_1^2) \\ x_2/10 - 2x_1 x_2 \exp(-x_1^2 - x_2^2) \end{bmatrix},$$

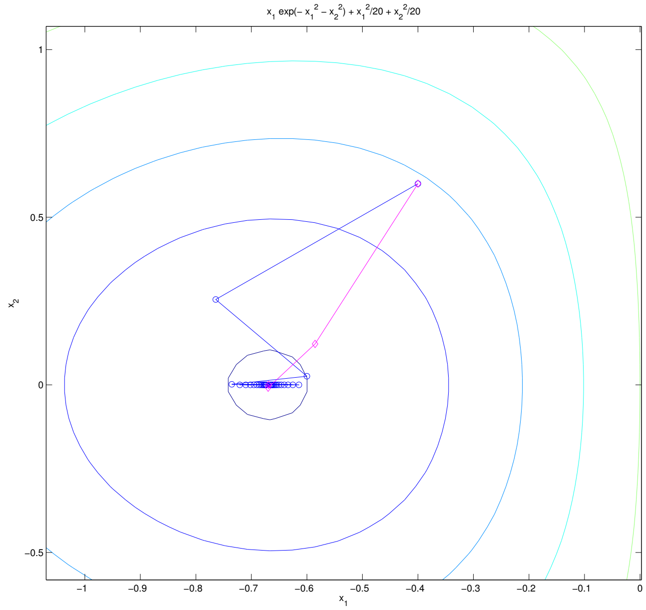
  $$\nabla^2 f(x) = \begin{bmatrix} (4x_1^3 - 6x_1)\exp(-x_1^2 - x_2^2) + 1/10 & (4x_1^2 x_2 - 2x_2)\exp(-x1^2 - x_2^2) \\ (4x_1^2 x_2 - 2x_2)\exp(-x_1^2 - x_2^2) & (4x_1 x_2^2 - 2x_1)\exp(-x_1^2 - x_2^2) + 1/10 \end{bmatrix}.$$



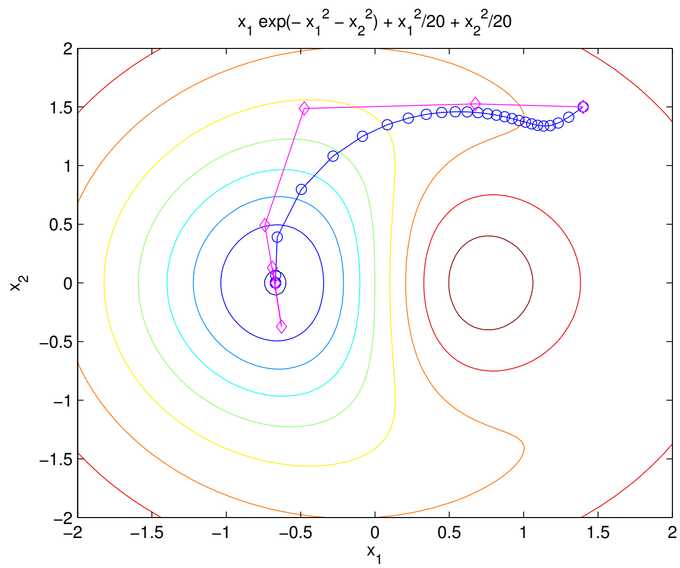
- The function has a strict global minimum around $x^* = (-2/3, 0)$ but also local minima
- There are also saddle points around $x = (1, 1.5)$
- We compare gradient-method (blue) and Newton method (magenta)
  - Gradient converges (but takes many steps); $\nabla^2 f$ is not p.d. and Newton get stuck



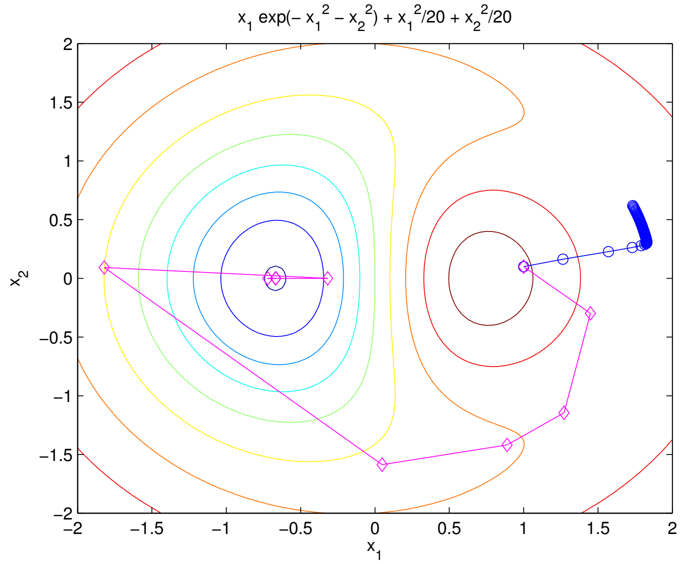
  - Both methods converge if started near optimum; gradient zigzags



– Newton's methods with regularization (trust-region) now works



– A bad starting guess causes gradient to converge to local minima



2.4 Regularized Netwon Method

The pure form of Newton's method has serious drawbacks:

- The inverse Hessian $\nabla^2 f(x)^{-1}$ might not be computable (e.g. if f were linear)
- When $\nabla^2 f(x)$ is not p.d. the method can be attracted by global maxima since it just solves $\nabla f = 0$

A simple approach to add a *regularizing* term to the Hessian and solve the system

$$(\nabla^2 f(x^k) + \Delta^k)d^k = -\nabla f(x^k)$$

where the matrix Δ^k is chosen so that

$$\nabla^2 f(x^k) + \Delta^k > 0.$$

There are several ways to choose Δ^k. In *trust-region* methods one sets

$$\Delta^k = \delta^k I,$$

where $\delta^k > 0$ and I is the identity matrix.

Newton's method is derived by finding the direction d which minimizes the local quadratic approximation f^k of f at x^k) defined by

$$f^k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d.$$

It can be shown that the resulting method

$$(\nabla^2 f(x^k) + \delta^k I)d^k = -\nabla f(x^k)$$

is equivalent to solving the the optimization problem

$$d^k \in \arg \min_{\|d\| \leq \gamma^k} f^k(d).$$

The *restricted direction* d must satisfy $\|d\| \leq \gamma^k$, which is referred to as the *trust region.*

References

[1] D. P. Bertsekas, *Nonlinear Programming, 2nd ed.* Belmont, MA: Athena Scientific, 2003.