OPTIMIZATION THEORY FUNDAMENTALS

Kostas S. Tsakalis
Arizona State University
Electrical Engineering
August 95

Part I: Outline

- Descent Algorithms: Basic Properties
  - — The Typical Optimization Problem
  - — Background Material: Gradient, Hessians, Convexity, Projections, Fixed Point Theorem
  - — Conditions for Local Minima
  - — Descent Directions and Convergence Theorems
  - — Line Search
  - — Steepest Descent, Newton and modifications
- Conjugate Direction Methods and variations
- Quasi-Newton Methods

The Typical Optimization Problem

$$\min f(x)$$
$$\text{subject to } x \in \Omega \subseteq \mathbf{R}^n$$

- $f(x) : \mathbf{R}^n \mapsto \mathbf{R}$; usually $f \in C^1$ or C^2 (once or twice continuously differentiable)

  f (and, possibly, its derivatives) can be evaluated at any given point.
- Ω : A set of constraints; e.g., $g_i(x) \leq 0$, $h_j(x) = 0$, $i, j = 1, 2, \ldots$
- Iterative Optimization: $x_{k+1} = \mathcal{A}[x_k]$. $\mathcal{A}$ is an algorithm such that

  $f(x_k) \to \min_{x\in\Omega} f(x)$

  $x(k) \to x_* \stackrel{\Delta}{=} \arg\min_{x\in\Omega} f(x)$ (possibly set-valued)

1 Fundamental Questions

- Convergence (local, global, region of attraction).
- Speed of convergence (in terms of x_k or $f(x_k)$); required number of iterations and computations.
- Robustness/sensitivity of solutions: How small perturbations in x affect the minimum value of f; how small perturbations in f affect the minimizing x.

  Remark: The first two questions are algorithm dependent; the last is problem dependent. (Is the optimization problem well-posed?)

2 Typical Iterative Solutions

- $x_{k+1} = x_k + a_k d_k$, where x_k is the current estimate of the minimizer, d_k is a "descent" direction and a_k is a scaling factor. The objective is to ensure that $f(x_{k+1}) < f(x_k)$ whenever x_k is "is not a minimizer."

  Basic descent algorithms (steepest descent, Newton-like, etc.) deal with the selection of a suitable descent direction and step size to achieve this objective.
- $X_{k+1} = \mathcal{A}(X_k)$, where X_k is the current estimate of a set containing the minimizing x. The objective here is to ensure that the "size" of X decreases whenever it contains non-minimizing points. (This idea finds applications in convex optimization.)

3 Some Applications

• **Production/Manufacturing:**

$$\min f(x) \quad \text{s.t. } x \in \Omega$$

f: Cost, Ω: Constraints (feasibility of solution)

• **Approximation:** Given pairs x_i, g_i, $i = 1, \ldots, n$ find parameters $a = [a_j]_{j=0}^m$ so as to minimize

$$\|g_i - h(x_i; a)\|$$

e.g., $h(x, a) = a_m x^m + \cdots + a_0$,
$h(x, a) = B/Ax^2 + 1/Ax + A^2$,
$A = a_1 \exp(-a_2/T), B = a_3 \exp(-a_4/T)$

• **Projection** Given a convex set $\mathcal{M}$ and a point $y \notin \mathcal{M}$, find a point $x \in \mathcal{M}$ such that the distance $|x - y|$ is minimum.

Background Material

4 Gradient, Hessians

• C^p: *The space of functions with continuous derivatives of order p.*

• *Gradient:* For $f \in C^1$,

$$\nabla f(x) = \frac{\partial f(x)}{\partial x} = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}\right]$$

For vector valued f,

$$\nabla f(x) = \begin{bmatrix} \nabla f_1(x) \\ \vdots \\ \nabla f_m(x) \end{bmatrix}$$

• *Hessian:* For $f \in C^2$,

$$\nabla^2 f(x) = F(x) = \nabla(\nabla f)(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]$$

Remark: For vector valued functions, the Hessian is a "third-order tensor." Most of the associated computations rely on the property: Given $\lambda \in \mathbf{R}^m$,
$\nabla(\lambda^\top f)(x) = \lambda^\top \nabla f(x)$ and

$$\nabla^2(\lambda^\top f)(x) = \lambda^\top F(x) = \sum_{i=1}^m \lambda_i \nabla^2 f_i(x)$$

5 Taylor's Theorem

If $f \in C^1$, there exists $\theta \in [0,1]$ s.t.

$$f(x_2) = f(x_1) + \nabla f(x_\theta)(x_2 - x_1)$$

If $f \in C^2$, there exists $\theta \in [0,1]$ s.t.

$$\begin{aligned} f(x_2) &= f(x_1) + \nabla f(x_1)(x_2 - x_1) \\ &\quad + \frac{1}{2}(x_2 - x_1)^\top \nabla^2 f(x_\theta)(x_2 - x_1) \end{aligned}$$

where $x_\theta = \theta x_1 + (1-\theta)x_2$.

6 Convex Sets

• *A set $M \subseteq \mathbf{R}^n$ is convex if for every $x_1, x_2 \in M$ and every real number $\theta \in [0,1]$, the point $\theta x_1 + (1-\theta)x_2 \in M$.*

The intersection of any collection of convex sets is convex.

7 Hyperplanes and Polytopes

• *A Hyperplane in* $\mathbf{R}^n$ is an (n-1)-dimensional linear variety (translated subspace).

$$H = \{x \in \mathbf{R}^n : a^\top x = c\}$$

a: a vector in $\mathbf{R}^n$; c: a real constant.

A Hyperplane divides $\mathbf{R}^n$ into two half spaces:

$$H_+ = \{x \in \mathbf{R}^n : a^\top x \geq c\} \quad \text{and} \quad H_- = \{x \in \mathbf{R}^n : a^\top x \leq c\}$$

Hyperplanes and the corresponding half spaces are convex sets.

• *A polytope* is an intersection of a finite number of closed half spaces.

Polytopes are convex sets and can be described as

$$H = \{x \in \mathbf{R}^n : Ax \leq b\}$$

where A is a matrix, b is a vector and the inequality is defined in terms of rows.

8 Ellipsoids

• *Ellipsoids* are sets described as

$$E = \{x \in \mathbf{R}^n : (x-a)^\top P(x-a) \leq 1\}$$

where P is a positive definite matrix ($P = P^\top > 0$) and a is a vector (center or centroid). If P is positive semidefinite ($P \geq 0$), E is a degenerate ellipsoid.

Ellipsoids are convex sets.

9 Some Examples

• Visualize the sets:

1. $\{x \in \mathbf{R}^3 : Ax \leq b\}$, $A = diag(1,1,1)$, $b = (-1,1,2)$
2. $\{x \in \mathbf{R}^3 : x^\top P x \leq 1\}$, $P = diag(1,2,0)$.

• Describe the set $\{x \in \mathbf{R}^2 : |x_1 + 2x_2| < 1\}$ by an ellipsoid.

• Describe the hyperplane in $\mathbf{R}^3$ that passes through the points $(1,0,0), (0,1,0), (0,0,2)$

10 Separating and Supporting Hyperplanes

• Given a convex set M and a point y exterior to $\bar{M}$ (the closure of M), there exists a hyperplane containing y and containing M in one of its open half spaces.

Given M convex and $y \notin \bar{M}$, there exists a such that $a^\top y < \inf_{x \in M} a^\top x$.

If y is a boundary point of M then there exists a hyperplane containing y and containing M in one of its closed half spaces.

• *Supporting Hyperplanes of a convex set M*: A hyperplane containing M in one of its closed half spaces and containing a boundary point of M.

11 Convex Functions

f defined on a convex set M is convex if, for every $x_1, x_2 \in M$ and every $\theta \in [0,1]$, there holds

$$f(x_\theta) \leq \theta f(x_1) + (1-\theta) f(x_2)$$

where $x_\theta = \theta x_1 + (1-\theta)x_2$.

(Strictly convex: strict inequality with $\theta \in (0,1)$ and $x_1 \neq x_2$. Concave if $-f$ is convex.)

• **Properties of Convex Functions**

1. For a convex function f on M, the set $\{x \in M : f(x) \leq c\}$ is convex for every real c.
2. For a convex function $f \in C^1$ over a convex set M

   $$f(y) - f(x) \geq \nabla f(x)(y-x) \; ; \quad \forall x, y \in M$$

   (the converse is also true)
3. For a convex function $f \in C^2$ over a convex set M (containing an interior point), the Hessian $\nabla^2 f(x)$ is positive semidefinite throughout M.

   (the converse is also true)
4. Let f convex on the convex set M. The set $\Gamma \subseteq M$ where f achieves its minimum is convex and any local minimum of f is a global minimum.

   If, in addition, $f \in C^1$ and there is $x_* \in M$ such that for all $y \in M$, $\nabla f(x_*)(y - x_*) \geq 0$, then x_* is a global minimum of f over M.
5. Let f convex on a bounded, closed, convex set M. If f has a maximum over M, it is achieved at an extreme point of M.

12 Example

Let $f \in C^1$ be a convex function and consider the convex set $\{x : f(x) \leq c\}$. A supporting hyperplane at a point $x_* : f(x_*) = c$ is the tangent hyperplane

$$\{x : \nabla f(x_*)x = \nabla f(x_*)x_*\}$$

For an ellipsoid $\{x : (x-a)^\top P(x-a) \leq 1\}$, we have

$$\nabla f(x) = 2(x-a)^\top P$$

Hence, a supporting hyperplane at the boundary point x_* is

$$\{x : 2(x_* - a)^\top P x = 2(x_* - a)^\top P x_*\}$$

Basic Properties of Solutions

13 Minima and Feasible directions

• x_* is a *relative (local) minimum* point of f over Ω if there exists $\epsilon > 0$ such that for any $|x - x_*| \leq \epsilon$, $f(x) \geq f(x_*)$. x_* is a *global minimum* of f over Ω if for any $x \in \Omega$, $f(x) \geq f(x_*)$.

(strict minimum for strict inequality and $x \neq x_*$)

• Given $x \in \Omega$, a vector d is a *feasible direction* at x if there is an $a_* > 0$ such that $x + ad \in \Omega$, for all $a \in [0, a_*]$.

14 1st Order Necessary Conditions

• Let $f \in C^1$ on Ω. If x_* is a local minimum of f over Ω, then for any feasible direction d at x_*, we have

$$\nabla f(x_*)d \geq 0$$

• (Unconstrained case) if x_* is an interior point of Ω, then

$$\nabla f(x_*) = 0$$

15 2nd Order Necessary Conditions

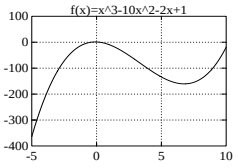
• Let $f \in C^2$ on Ω. If x_* is a local minimum of f over Ω, then for any feasible direction d at x_*, we have
1. $\nabla f(x_*)d \geq 0$;
2. If $\nabla f(x_*)d = 0$, then $d^\top \nabla^2 f(x_*)d \geq 0$.
• (Unconstrained case) if x_* is an interior point of Ω, then $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*) \geq 0$ (positive semi-definite).

16 Sufficient Conditions

• Let $f \in C^2$ on Ω. Suppose that x_* is an interior point of Ω and such that
1. $\nabla f(x_*) = 0$;
2. $\nabla^2 f(x_*) > 0$ (positive definite).
Then x_* is a strict local minimum of f.

17 Example

• Let $f(x) = x^3 - 10x^2 - 2x + 1$. Then
$\nabla f(x) = 3x^2 - 20x - 2$; (roots at $6.76, -0.098$)
$\nabla^2 f(x) = 6x - 20$
$\nabla f(6.76) > 0, \nabla f(-0.098) > 0$



f has a local min. at 6.76 and a local max. at -0.098
In the interval [0,10] f has a local max. at 0 and at 10
$-\nabla f(0)d = 2d > 0, d > 0$,
$-\nabla f(10)d = -98d > 0, d < 0$
In the interval [0,5] f has a local minimum at 5
$\nabla f(5)d = -27d > 0, d < 0$

18 Algorithms and Descent Functions

• A general setting to study properties of algorithms
• In a general framework, an Algorithm $\mathcal{A}$ is a point-to-set mapping.
A typical iteration step is $x_{k+1} \in \mathcal{A}(x_k)$.
• Let Γ be a "solution set" (e.g., where $\nabla f(x) = 0$). A continuous real-valued function Z is a *descent function for* Γ *and* $\mathcal{A}$ if
1. if $x \notin \Gamma$ and $y \in \mathcal{A}(x)$, then $Z(y) < Z(x)$
2. if $x \in \Gamma$ and $y \in \mathcal{A}(x)$, then $Z(y) \leq Z(x)$
(e.g., $|\nabla f(x)|$, or $f(x)$ could serve as descent functions)

• *Closed Mapping*: The assumptions
1. $x_k \to x \in X$, 2. $y_k \to y$, $y_k \in \mathcal{A}(x_k)$
imply $y \in \mathcal{A}(x)$
• For point-to-point mappings, continuity implies closedness

19 Global Convergence

Let $\mathcal{A}$ be an algorithm on a set X, $\{x_k\}_{k=0}^{\infty}$ be a sequence generated by $x_{k+1} \in \mathcal{A}(x_k)$ and Γ be a solution set. Suppose that
1. all points x_k are contained in a compact set
2. there exists a descent function for γ and $\mathcal{A}$
3. $\mathcal{A}$ is closed at points outside Γ
Then the limit of any convergent subsequence of $\{x_k\}$ is a solution. If, in addition, Γ consists of a single point x_*, then $x_k \to x_*$.

20 Speed of Convergence

Here $\{r_k\}$ denotes a sequence of real numbers converging to r_*.
• Linear (geometric) Convergence:
$\lim_{k\to\infty} \frac{|r_{k+1} - r_*|}{|r_k - r_*|} = \beta < 1$. β is called the convergence ratio. ($|r_k - r_*| \leq c\beta^k$)
• Superlinear Convergence: $\beta = 0$
• Convergence of order p: $\lim\limits_{k\to\infty} \dfrac{|r_{k+1} - r_*|}{|r_k - r_*|^p} < \infty$.

21 Banach Fixed Point Theorem

• Determines when a *fixed point of* f $x_* = f(x_*)$ can be found iteratively by $x_{k+1} = f(x_k)$.
Provides only sufficient (but elegant) conditions
• Let S be a subset of a normed space X and let T be a map $S \mapsto S$ (i.e., $T(S) \subseteq S$). Then T is a *contraction mapping* if there is a $\rho \in [0, 1)$ such that

$$\|T(x_1) - T(x_2)\| \leq \rho \|x_1 - x_2\|, \quad \forall x_1, x_2 \in S$$

• *If T is a contraction on a closed subset S of a Banach (complete) space, there is a unique vector $x_* \in S$ satisfying $x_* = T(x_*)$. Furthermore, x_* can be obtained iteratively by starting from an arbitrary point $x_0 \in S$ and forming the sequence*

$$x_{k+1} = T(x_k), \quad k = 0, 1, \ldots$$

The convergence of the sequence is linear with ratio ρ.
• *Remarks:*
1. $\rho \leq \sup_{x\in S} \|\nabla T(x)\|$
2. The theorem is NOT valid with the weaker condition $\|T(x_1) - T(x_2)\| < \|x_1 - x_2\|$
3. Other (weaker) local versions do exist

22 BFP Example

Classical applications of the Banach fixed point theorem include the solution of linear and nonlinear algebraic equations, integral and differential equations (Picard's iterations).
One such example is the solution of $f(x) = 0$ using Newton's method. In particular, consider the problem of computing the square root of a positive real number, i.e., find x s.t. $x^2 - a = 0$.
• Formulation
Rewrite the equation as $x - x/2 = a/(2x)$ or $x = \frac{1}{2}(x + a/x)$
(Motivation: Newton's method
$x_{k+1} = x_k - [\nabla f(x_k)]^{-1} f(x_k)$)

That is, $T(x) = \frac{1}{2}(x + a/x)$ and $\nabla T(x) = \frac{1}{2}(1 - a/x^2)$
• Analysis
Contraction Constant: $|\nabla T(x)| < \frac{1}{2}$ in $[\sqrt{a/2}, \infty)$
Define $S = [\sqrt{a/2}, \infty)$ (S^c open $\Rightarrow S$ closed)
Then $T(S) \subset S$
□ $T(x) > \sqrt{a/2} : x^4 + a^2 > 0$
• Result
T satisfies the fixed point theorem; hence starting with $x_0 \in S$, the sequence $x_{k+1} = T(x_k)$ converges to x_* at least as fast as 0.5^k.
• *Remarks*
1. Newton's algorithm guarantees the existence of a sufficiently small S around simple solutions. For this particular example, any positive x_0 yields $x_1 \in S$. The algorithm converges for any $x_0 > 0$ This "nice" behavior is not typical for the standard Newton algorithm.
2. Convergence is, in fact, of order 2.
As a numerical example, let's use the algorithm to compute
$\sqrt{2}0 = 4.47213595499958$

```
MATLAB Commands
x=1
x=(x+20/x)/2

x0 = 1
x1= 10.50000000000000
x2 = 6.20238095238095
x3 = 4.71347454528837
x4 = 4.47831444547438
x5 = 4.47214021706570
x6 = 4.47213595500161
x7 = 4.47213595499958
x8 = 4.47213595499958
```

Optimization in Vector Spaces
• A special but very important class of problems is: given a subspace $\mathcal{M}$ and a point x, to find $m_* \in \mathcal{M}$ minimizing the distance $\|x - m\|$.
• A general solution can be obtained iteratively by convex optimization. However, when the distance is induced by an inner product, the solution can be explicitly characterized by means of orthogonality and alignment concepts and computed using projection operators.
• The associated theory is very rich and powerful, applicable to very general settings of vector spaces. Typical examples include minimum norm and least squares problems in $\mathbf{R}^n$, general approximation theory (e.g., Fourier), optimal control and estimation problems (in function spaces), system identification and more.
• Here, we summarize some of the basic results, aiming to gain working knowledge for the simple cases and some intuition about the more general ones (these require a more extensive background on real and functional analysis).

23 The Classical Projection Theorem
Let X be a Hilbert space and $\mathcal{M}$ a closed subspace of X. Corresponding to any vector $x \in X$ there is a unique vector $m_* \in \mathcal{M}$ such that $\|x - m_*\| \leq \|x - m\|$, for all $m \in \mathcal{M}$. Furthermore, $m_* \in \mathcal{M}$ is the unique minimizer if and only if $x - m_*$ is orthogonal to $\mathcal{M}$.
• The minimizer is computed by means of a projection operator ($\mathcal{P}_{\mathcal{M}} : X \mapsto \mathcal{M}$). The condition $x - m_* \in \mathcal{M}^\perp$, means that $m_* = \mathcal{P}_{\mathcal{M}}(x)$. Let us use this to derive the solution to two classical optimization problems.
Notation:
$A^\top$: the transpose of A (in general, the adjoint)
$(x|y)$: inner product
$X^\perp$: the orthogonal complement of X:
$\{y : (x|y) = 0, \forall x \in X\}$
$\mathcal{R}(A)$,$\mathcal{N}(A)$: the range and null spaces of A, respectively

CPT Examples

---

24 Least Squares
Find x such that $\|Ax - b\|$ is minimum.
• Setting $y = Ax$, $\mathcal{M} = \mathcal{R}(A)$, we want to minimize $\|y - b\|$, s.t. $y \in \mathcal{M}$. From the CPT, the minimizer is such that
$y - b \in [\mathcal{R}(A)]^\perp = \mathcal{N}(A^\top)$, i.e.,

$$A^\top y = A^\top Ax = A^\top b$$

These are the celebrated *Normal Equations*. Any solution of them is a solution of our LS minimization problem. If $A^\top A$ is invertible then a simple formula for x_* is

$$x_* = (A^\top A)^{-1} A^\top b$$

Alternatively,
if $A^\top A$ is invertible, $\mathcal{P}_{\mathcal{R}(A)} = A(A^\top A)^{-1}A^\top$. Since $\mathcal{P}_{\mathcal{R}(A)}(y - b) = y - \mathcal{P}_{\mathcal{R}(A)}(b)$ the optimal y is $\mathcal{P}_{\mathcal{R}(A)}(b)$. That is, $Ax_* = A(A^\top A)^{-1}A^\top b$. By the CPT, the obvious solution $x_* = (A^\top A)^{-1}A^\top b$ is also unique.

25 Minimum Norm
Find the minimum norm x subject to $Ax = b$.
Let x_o be a solution, i.e., $Ax_o = b$. Define $\hat{x} = x - x_o$. Then the MN problem becomes

$$\min_{\hat{x} \in \mathcal{N}(A)} \|\hat{x} + x_o\|$$

Assuming that $\mathcal{R}(A)$ is closed and $AA^\top$ is invertible, we have that the optimizing $\hat{x}$ should satisfy $\hat{x}_* + x_o \in [\mathcal{N}(A)]^\perp = \mathcal{R}(A^\top)$ Hence, there exists z such that $\hat{x}_* + x_o = x_* = A^\top z$. Since $Ax_* = b$, we have that $A(\hat{x}_* + x_o) = b = AA^\top z$ and therefore $z = (AA^\top)^{-1}b$. Thus,

$$x_* = A^\top (AA^\top)^{-1} b$$

Alternatively, since $\hat{x}_* \in \mathcal{N}(A)$,

$$\mathcal{P}_{\mathcal{N}(A)}(\hat{x}_* + x_o) = \hat{x}_* + \mathcal{P}_{\mathcal{N}(A)}(x_o) = 0$$

Therefore, $x_* = (I - \mathcal{P}_{\mathcal{N}(A)})(x_o) = \mathcal{P}_{[\mathcal{N}(A)]^\perp}(x_o) = \mathcal{P}_{\mathcal{R}(A^\top)}(x_o)$ The last projection has the form

$$\mathcal{P}_{\mathcal{R}(A^\top)} = A^\top (AA^\top)^{-1} A$$

Hence, $x_* = A^\top (AA^\top)^{-1} Ax_o$. While x_o is unspecified, $Ax_o = b$, yielding $x_* = A^\top (AA^\top)^{-1} b$
• *Remark:* Modulo technicalities, all of the above translate to very general settings. For example, in the problem of finding the minimum norm input transferring the state of a linear dynamical system ($\dot{x} = Fx + Gu$) to zero, A is a convolution operator ($A : L_2 \mapsto \mathbf{R}^n$), whose adjoint is a

multiplier, yielding the well-known controllability Grammian as $AA^\top$.

26 Summary of Projections

• $m_* \in \mathcal{M}$ minimizes $\|x - m\|, m \in \mathcal{M}$ iff $x - m_* \in \mathcal{M}^\perp$.
• LS Solution for A 1-1: $x_* = (A^\top A)^{-1} A^\top b$
• MN Solution for A onto: $x_* = A^\top (AA^\top)^{-1} b$
• General LS solution(s): $(A^\top A) x_{LS} = A^\top b$
• A 1-1: $\mathcal{P}_{\mathcal{R}(A)} = A(A^\top A)^{-1} A^\top$
• A onto: $\mathcal{P}_{\mathcal{R}(A^\top)} = A^\top (AA^\top)^{-1} A$
• Orthogonal Projections: $P = P^2, P = P^\top$;
$P_{X^\perp} = I - P_X$ [1]
• $\mathcal{P}_{[\mathcal{N}(A)]^\perp} = \mathcal{P}_{\mathcal{R}(A^\top)}$

27 Example 1: Function Approximation

Let $b(t) = t$, $g_1(t) = 1$, $g_2(t) = \{-1$ if $t < 0.5, 1$ if $t > 0.5\}$, $t \in [0,1]$. Find $y(t) = \sum_{i=1}^2 x_i g_i(t)$ minimizing

$$\|y - b\|^2 \triangleq \int_0^1 |y(t) - b(t)|^2 \, dt$$

• *Formulation*
$L_2([0,1])$: square-integrable, real functions on [0,1]. It is a Hilbert space with inner product and norm:

$$(b|y) = \int_0^1 b(t) y(t) \, dt \; ; \quad \|b\|_2 = (b|b)^{1/2}$$

Define the operator $A : \mathbf{R}^2 \mapsto L_2$ by $Ax = \sum_{i=1}^2 x_i g_i(t)$. Our problem translates into

$$\min_{x \in \mathbf{R}^2} \|Ax - b\|_2$$

• *Solution*
After verifying that the CPT conditions hold (not trivial!), the solution is found as

$$(A^\top A) x_* = A^\top b \quad \text{or} \quad x_* = (A^\top A)^{-1} A^\top b$$

All that is left is to compute the operator $A^\top : L_2 \mapsto \mathbf{R}^2$.
• *Computations*
Here, for notational simplicity, we used $A^\top$ to denote the *adjoint* operator of A (usual symbol: A^*), defined by

$$(z|Ax)_{L_2} = (A^\top z|x)_{\mathbf{R}^2}, \forall z, x$$

Letting $z \in L_2$, $(z|Ax) = \int z \sum x_i g_i = \sum x_i (z|g_i) = (A^\top z|x)$. Hence,

$$A^\top z = \begin{bmatrix} (z|g_1) \\ (z|g_2) \end{bmatrix} \in \mathbf{R}^2$$

$$A^\top A = \begin{bmatrix} (g_1|g_1) & (g_2|g_1) \\ (g_1|g_2) & (g_2|g_2) \end{bmatrix} \; (: \mathbf{R}^2 \mapsto \mathbf{R}^2)$$

Computation of the various entries of $A^\top A$ and $A^\top b$
$(g_1|g_1) = \int_0^1 (1)(1) = 1$
$(g_1|g_2) = (g_2|g_1) = \int_0^{0.5} (-1)(1) + \int_{0.5}^1 (1)(1) = 0$
$(g_2|g_2) = \int_0^{0.5} (-1)(-1) + \int_{0.5}^1 (1)(1) = 1$
$(b|g_1) = \int_0^1 (t)(1) = 0.5$
$(b|g_2) = \int_0^{0.5} (t)(-1) + \int_{0.5}^1 (t)(1) = 0.25$
Substitution into the normal equations:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x = \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix}$$

• *Final Result*
$y(t) = 0.5 + 0.25 g_2(t)$
(a staircase or piecewise constant approximation)

28 Example 2: LS Data Fit

Given N pairs (x_i, y_i) find a line $\hat{y} = ax + b$ yielding minimum sum of square errors, i.e. $\|Az - y\|, z = [a, b]^\top$. In an expanded form, we want to solve, in a LS sense,

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \end{bmatrix} z = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}$$

The normal equations now become

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{bmatrix} z = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

Analytical inversion of the 2×2 matrix yields

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{N \sum x_i^2 - (\sum x_i)^2} \begin{bmatrix} N \sum x_i y_i - \sum x_i \sum y_i \\ \sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i \end{bmatrix}$$

a solution that can easily be implemented in a recursive form in a small calculator, using five storage locations.

29 Example 3: Minimum distance of a hyperplane from the origin

• Find the minimum distance of the hyperplane passing through (1,0,0), (0,1,0), (0,0,1) from the origin.
• *Formulation*
Evaluating the hyperplane equation $a^\top x = c$ at the given points, we find the following relations for a and c:

$$a_1 = a_2 = a_3 = c$$

Normalizing $c = 1$, our hyperplane is described by

$$\{x \in \mathbf{R}^3 : a^\top x = 1\} \; , \quad a^\top = (1,1,1)$$

Note: a is the normal vector to the hyperplane.
Thus, our problem becomes

$$\min_{x \in \mathbf{R}^3} \|x\| \; , \quad \text{s.t. } a^\top x = 1$$

• *Solution*
Using the MN formula, the minimizing x is

$$x_* = a(a^\top a)^{-1}(1) = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$$

whose norm is $\|x_*\| = 1/\sqrt{3}$. This is the minimum distance of our hyperplane from the origin.

[1] This is useful when projecting a vector in $\mathbf{R}^n$ on an $(n-1)$-dimensional subspace: Instead of considering the subspace as the range of an $(n-1)$-rank matrix (that would require the inversion of the $(n-1) \times (n-1)$ $A^\top A$ matrix) we think of it as the orthogonal complement of its normal vector, i.e., all $x : n^\top x = 0$. In this case, $P_X = I - P_n = I - nn^\top/(n^\top n)$, requiring only a scalar division.

(The same result is obtained from elementary Euclidean geometry.)

Basic Descent Methods: Line Search

- $x_{k+1} = x_k + a_k d_k$, d_k is a descent direction (e.g., $-\nabla f$)
- a_k computed by *Line Search*

$$a_k = \arg\min_{a \geq 0} f(x_k + a d_k)$$

30 Fibbonacci and Golden Section Search

- Suppose f is unimodal in $[x_{min}, x_{max}]$ (one minimum).

Fibonacci: Select N successive measurements to determine the smallest region where the minimum must lie.
Golden Section: Fibonacci with $N \to \infty$.
Generate 4 points $x_1, \ldots, x_4$ and evaluate $f(x_1), \ldots, f(x_4)$
If $f(x_1) \geq f(x_2) \geq f(x_3)$ then exclude $[x_1, x_2)$
If $f(x_4) \geq f(x_3) \geq f(x_2)$ then exclude $(x_3, x_4]$
else the function not unimodal
In the reduced interval, generate one new point preserving the symmetry properties of the initial four points.

$$x_1 = x_{min}, x_4 = x_{max}$$

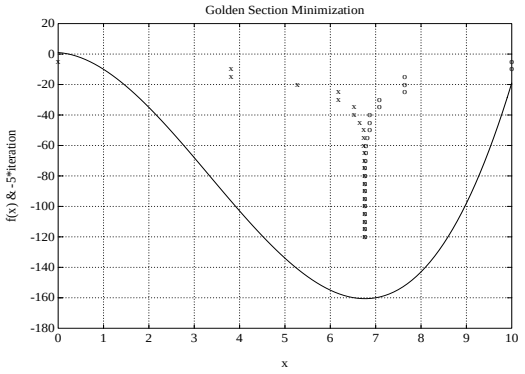
$$x_2 = x_1 + \tau(x_4 - x_1) \;\; ; \;\; x_3 = x_1 + (1-\tau)(x_4 - x_1)$$

Exclude $[x_1, x_2)$: $x_1 \leftarrow x_2, x_2 \leftarrow x_3, x_4 \leftarrow x_4, x_3 \leftarrow x_1 + \tau(x_4 - x_1)$
Exclude $(x_3, x_4]$: $x_4 \leftarrow x_3, x_3 \leftarrow x_2, x_1 \leftarrow x_1, x_3 \leftarrow x_4 - \tau(x_4 - x_1)$

31 Golden Section Example

- Minimize $f(x) = x^3 - 10x^2 - 2x + 1$, $x \in [0, 10]$ (Ex. 17)

```
[amin,index,xs]=linsea('fun1',0,10)
plot xs
```



32 Newton

- Taylor expansion around x_k

$$f(x) = f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}\nabla^2 f(x_k)(x - x_k)^2 + H.O.T.$$

Determine x_{k+1} so that it is the desired solution if the H.O.T. are ignored.

□ Minimize f approximated by its first 3 Taylor terms

□ Solve $g(x) \stackrel{\Delta}{=} \nabla f(x) = 0$. $(\nabla f(x) = \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) + H.O.T.)$

$$x_{k+1} = x_k - \frac{\nabla f(x_k)}{\nabla^2 f(x_k)}$$

- **Newton's method for solving** $g(x) = 0$

For $g \in C^2$, $x_* : g(x_*) = 0, \nabla g(x_*) \neq 0$ (simple root), Newton's iteration converges locally to x_* with order 2.
- *Remark:* Local convergence: x_0 sufficiently close to x_*.
- *Analysis*

Using the fixed point theorem with $T(x) = x - \frac{g(x)}{\nabla g(x)}$

$$\nabla T(x) = 1 - \frac{\nabla g(x)}{\nabla g(x)} + \frac{g(x)\nabla^2 g(x)}{[\nabla g(x)]^2}$$

To show that T is a contraction in $S = \{x : |x - x_*| \leq \epsilon\}$ note that

$$\sup_{x \in S} |\nabla T(x)| \leq \frac{\sup_{x \in S} |g(x)| \sup_{x \in S} |\nabla^2 g(x)|}{\inf_{x \in S} |\nabla g(x)|^2} \leq \rho < 1$$

for sufficiently small $\epsilon(\rho)$. ($\nabla^2 g(x)$ bounded in S and, by continuity, $1/\nabla g(x)$ bounded in S, $g(x) \leq O(\epsilon)$ in S)
$T(S) \subseteq S$:
In S, $0 = g(x_*) = g(x) + \nabla g(x)(x_* - x) + O(x - x_*)^2$.
Hence,

$$(x_{k+1} - x_*) = (x_k - x_*) + \frac{\nabla g(x_k)(x_* - x_k)}{\nabla g(x_k)} + \frac{O(x_k - x_*)^2}{\nabla g(x_k)}$$

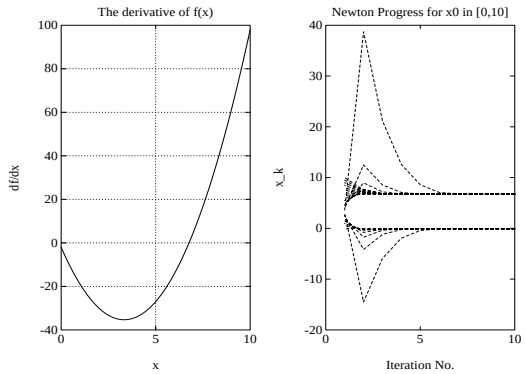
$$|x_{k+1} - x_*| \leq CO(x - x_*)^2 \leq CO\epsilon^2 \leq \epsilon$$

for suff. small ϵ; hence $x_k \in S \Rightarrow x_{k+1} \in S$.

33 Newton Example

- Minimize $f(x) = x^3 - 10x^2 - 2x + 1$ (Ex. 17)
- Solve $g(x) = \nabla f(x) = 0$

```
[amin,xs]=newtmn('fun2',5)
plot xs
```



34 False Position

Newton with approximate derivative:
$\nabla^2 f(x_k) = \frac{\nabla f(x_k) - \nabla f(x_{k-1})}{x_k - x_{k-1}}$
Local convergence (like Newton) with order 1.618 (golden mean)

35 Polynomial Fit

• Cubic fit of f and ∇f at x_k, x_{k-1}. Local convergence of order 2

• Quadratic fit of f at x_k, x_{k-1}, x_{k-2}. Local convergence of order 1.3

□ Polynomial fit methods are usually preferred since they can be implemented in a relatively efficient and reliable way. Newton's method, on the other hand may exhibit unreliability problems, especially if it is implemented without constraints.

36 Stopping Criteria

Line search is only a part of a minimization algorithm. Convergence to the exact minimizer a_* of $\phi(a) = f(x_k + ad_k)$ should not be required. Stopping criteria should ensure that "adequate" convergence has been achieved.

• a should not be too large or too small. This idea can be justified by examining the expansion of $\phi(a)$ around 0,

$$\phi(a) = \phi(0) + \nabla f(x_k)ad_k + \frac{1}{2}a^2 d_k^2 \nabla^2 f$$

Since $\nabla f(x_k)d_k < 0$, $\phi(a) - \phi(0) < 0$ for a satisfying

$$0 < a < \frac{-2\nabla f(x_k)d_k}{d_k^2 \sup|\nabla^2 f|}$$

• Percentage test: $|a - a_*| \leq ca_*$, $c \simeq 0.1$ or less. Requires an a priori estimate of a/a_*. Suitable for algorithms that produce intervals containing the minimizer.

• Armijo: $n = 2$ or 10, $\epsilon = 0.2$

$$\phi(a) \leq \phi(0) + \epsilon(\nabla f(x_k)d_k)a, \quad a \text{ not too large}$$

$$\phi(na) > \phi(0) + \epsilon(\nabla f(x_k)d_k)na, \quad a \text{ not too small}$$

(sometimes used as a search technique)

• Goldstein: Like Armijo's ($\epsilon \in (0, 1/2)$) except

$$\phi(a) > \phi(0) + (1-\epsilon)(\nabla f(x_k)d_k)a, \quad a \text{ not too small}$$

• Wolfe: Like Armijo's ($\epsilon \in (0, 1/2)$) except

$$(\nabla f(x_k)d_k)a > (1-\epsilon)(\nabla f(x_k)d_k), \quad a \text{ not too small}$$

• *Remarks:*

□ The "cost" of evaluating functions or function derivatives is an important factor. A natural trade-off exists between spending time in objective optimization versus line searches. Sometimes an algorithm is applied without line searches by using a conservative rule to choose the step size (e.g., adaptive estimation).

□ Closedness of the line search is important the theoretical analysis of algorithms.

Basic Descent Methods: Steepest Descent

---

37 The Steepest Descent Method

• Select $g_k = -\nabla f(x_k)^\top$ as a descent direction and compute x_{k+1} by

$$x_{k+1} = x_k - a_k g_k \ , \quad a_k = \arg\min_{a \geq 0} f(x_k - ag_k)$$

• Global convergence to the solution set $x : \nabla f(x) = 0$. (via the Global Convergence theorem and the closedness of the line search algorithm)

• Linear Convergence near the minimizer with rate

$$\left(\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}\right)^2 = \left(\frac{r-1}{r+1}\right)^2 \ ; \quad r = \frac{\lambda_{max}}{\lambda_{min}}$$

where $\lambda_{max}, \lambda_{min}$ are the maximum and minimum eigenvalues of the Hessian $\nabla^2 f(x_*)$ and r is its condition number.

(Convergence may be very poor for badly conditioned problems)

38 Steepest Descent: Quadratic case

• $f(x) = \frac{1}{2}x^\top Q x - x^\top b$, $Q = Q^\top > 0$. From the 2nd order sufficient conditions the minimizer is $x_* = Q^{-1}b$.

• Define the error functional $E(x) = \frac{1}{2}(x - x_*)^\top Q(x - x_*)$. (Note that $E(x) = f(x) + \frac{1}{2}x_*^\top Q x_*$.)

• Gradient of f and E: $\nabla f(x) = x^\top Q - b^\top$. Hessian: $\nabla^2 f(x) = Q$.

• Steepest Descent: $g_k = \nabla f(x_k)^\top$, $x_{k+1} = x_k - a_k g_k$

$a_k = \arg\min_{a \geq 0} f(x_k - ag_k)$ can be found explicitly: $a_k = \frac{g_k^\top q_k}{g_k^\top Q q_k}$

• *Analysis*

□ $E(x_k)$ is decreasing:

$$E(x_{k+1} = \left[1 - \frac{(g_k^\top q_k)^2}{g_k^\top Q q_k g_k^\top Q^{-1} q_k}\right] E(x_k)$$

□ *Kantorovich inequality*: (Q positive definite)

$$\frac{(x^\top x)^2}{x^\top Q x x^\top Q^{-1} x} \geq \frac{4\lambda_{min}\lambda_{max}}{(\lambda_{min} + \lambda_{max})^2}$$

where $\lambda_{min}, \lambda_{max}$ are the minimum and maximum eigenvalues of Q.

Combining the above expressions we arrive at the desired result.

• Nonquadratic case analysis: Quadratic (Taylor) approximation of f, locally around x_*.

• Scaling: $x = Ty$; then $\nabla_y^2 f(Ty) = T^\top \nabla^2 f(x) T$.

$$y_{k+1} = y_k - a_k T g_k \Rightarrow x_{k+1} = x_k - a_k T^2 g_k$$

□ $T = (\sqrt{\nabla^2 f(x)})^{-1}$ attempts to change the coordinates so that the surfaces $E = const.$ look like spheres. Other scaling factors that yield approximately the same effect may also produce considerable speedup.

• **Coordinate Descent:** Perform a line search with respect to one coordinate only. E.g.:

□ Cyclic: $x_1, \ldots x_n, x_1, \ldots, x_n, \ldots$.

□ Gauss-Southwell: largest in absolute value component of ∇f

These are simple algorithms, especially useful for "manual" optimization.

Basic Descent Methods: Examples

• Minimize $f(x) = x_1^2 + x_2^4 - 5x_1x_2 - 25x_1 - 8x_2$

The minimum of this function occurs at (20,3) where the condition Number of Hessian $\simeq 61$.

39 Steepest Descent

□ From the previous analysis we expect that the Steepest Descent should converge near the optimum as 0.967^k (relatively slow).

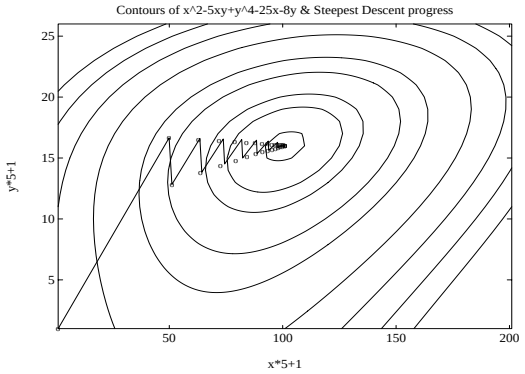
□ The MATLAB function *stdes* solves this problem using either Newton or Golden Section line search.

□ Letting (0,0) be our starting point the Steepest Descent converges in 31 steps (default tolerance).

```
[x,niter,xs]=stdes([0;0]);
plot(xs(:,1),xs(:,2),'+')
```

□ The results are shown in the figure below where the steepest descent progress is plotted together with the level curves of f. (Note that the coordinates are not Cartesian).

□ Observe the "zig-zag" pattern near the minimum, caused by the long valleys in the function level curves and indicated by the condition number of the Hessian. (Condition numbers can get much worse too!)



Basic Descent Methods: Newton's Method

40 Newton's Method

Approximate f by its truncated Taylor series around x_k

$$f(x) \simeq f(x_k) + \nabla f(x_k) + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k)$$

Choose x_{k+1} so as to minimize the right-hand side

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

(Local convergence by the Fixed Point Theorem; in fact the order of convergence is two; $f \in C^3, \nabla^2 f(x_*) > 0$)

41 Variations of Newton's Method

• A General Class of Algorithms:

$$x_{k+1} = x_k - a_k S_k g_k \ ; \quad (g_k = \nabla f(x_k)^\top)$$

• $-S_k g_k$ is a descent direction for $S_k = S_k^\top > 0$
• a_k from line search, minimizing $f(x_k - a S_k g_k)$
• $S_k = I$: Steepest Descent
• $S_k = [\nabla^2 f(x_k)]^{-1}$: avoids some of the problems due to H.O.T. but can fail when the Hessian is not positive definite (local convergence)
• $S_k = [\epsilon_k I + \nabla^2 f(x_k)]^{-1}$: ϵ_k is the smallest positive constant making $[\epsilon_k I + \nabla^2 f(x_k)]$ positive definite with minimum eigenvalue δ. This algorithm achieves global convergence and Newton-type scaling if $\nabla^2 f(x_k)$ is "sufficiently p.d." and partial scaling if not. (If $\epsilon_k \not\to 0$, the convergence is linear, at least as good as steepest descent.)

□ $\epsilon_k = \delta - \min \text{eig}(\nabla^2 f(x_k))$ is an easy selection. The computation of eigenvalues can be avoided through Cholesky factorization ($\nabla^2 f(x_k) = GG^\top$) and recursive definition of ϵ_k (Levenberg-Marquardt).

• S_k recursively updated through gradient information (see quasi-Newton methods).

Basic Descent Methods: Examples

42 Scaled Steepest Descent Example

To improve the convergence rate we can define the diagonal scaling factor $x = Ty$ where

$$T^{-2} = 2 \begin{bmatrix} 1 & 0 \\ 0 & x(2)^2 + 0.01 \end{bmatrix}$$

□ Steepest Descent with scaling yields convergence in 11 steps.

```
[x,niter,xs]=gnewt([0;0], 3);
plot(xs(:,1),xs(:,2),'+')
```

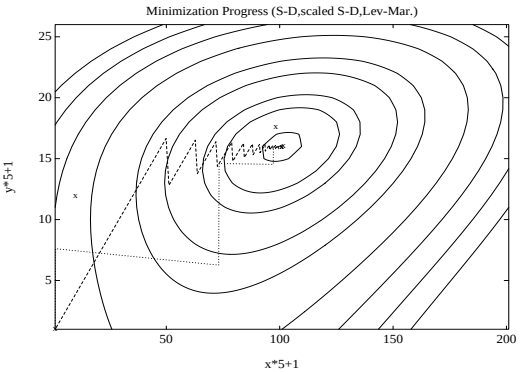
43 Newton-type Algorithms Example

□ While the unmodified Newton method fails to converge (the Hessian at (0,0) is indefinite), a Levenberg-Marquardt-type converges in 5 steps.

```
[x,niter,xs]=gnewt([0;0], 2);
plot(xs(:,1),xs(:,2),'+')
```

□ The results are shown in the figure below where the progress of the various algorithms is plotted together with the level curves of f.

□ Observe the reduction of the "zig-zag" pattern near the minimum, achieved by using some information about the Hessian.



Conjugate Direction Methods

Attempting to accelerate the slow convergence of Steepest Descent Methods but without resorting in "expensive" Hessian evaluations and inversions.

Most of the development is performed for quadratic functions

$$f(x) = \frac{1}{2}x^\top Qx - b^\top x, \quad Q = Q^\top > 0$$

but the results can be extended for general functions via Taylor approximations.

44 Conjugate Directions in the Quadratic Case

• Q-orthogonal (conjugate) vectors: $d_i^\top Q d_j = 0$.

• n Q-orthogonal vectors form a basis in $\mathbf{R}^n$.

• A Q-orthogonal basis provides a natural expansion for the minimizer $x_* = Q^{-1}b$.

$$x_* = \sum_i \frac{d_i d_i^\top b}{d_i^\top Q d_i}$$

• *Conjugate Direction Algorithm:* $x_{k+1} = x_k - d_k \frac{\nabla f(x_k) d_k}{d_k^\top Q d_k}$ where $\{d_k\}$ are Q-orthogonal; convergence in n steps.

□ Precise movement in each direction. x_k minimizes $f(x)$ on the whole variety $x_0 + span(d_0, \ldots, d_{k-1})$. The k-th gradient is orthogonal to all previous directions.

45 Conjugate Gradient Method

• Idea: Start with ∇f as a direction and perform a line search; extract previous component from the new gradient (in Q-orthogonal coordinates) and continue.

• **C-G Algorithm:** Let $x_0 \in \mathbf{R}^n$, $d_0 = -g_0 = -Qx_0 + b$, and repeat the sequence:

$$\begin{aligned} a_k &= -\frac{g_k^\top d_k}{d_k^\top Q d_k}; \quad x_{k+1} = x_k + a_k d_k \\ g_{k+1} &= [\nabla f(x_{k+1})]^\top = Qx_{k+1} - b \\ \beta_k &= \frac{g_{k+1} Q d_k}{d_k^\top Q d_k}; \quad d_{k+1} = -g_{k+1} + \beta_k d_k \end{aligned}$$

• Optimality of the C-G algorithm in the reduction of the error functional $E(x_{k+1}) = \frac{1}{2}(x_{k+1} - x_*)^\top Q(x_{k+1} - x_*)$ over all algorithms of the form $x_{k+1} = x_k + P_k(Q)g_0$, where P_k is a polynomial of degree k.
(Note: Cayley-Hamilton: $Q^{-1} = P_*(Q)$, $degP_* \leq n-1$.)

• *Partial C-G Methods:* Restart every m steps. Applications in cases where Q has clustered eigenvalues (e.g., as they appear in penalty function augmentation) and the general nonlinear case.

□ Suppose $Q > 0$ has $n - m$ eigenvalues in $[a, b]$ and the remaining in (b, ∞). Then the C-G method, restarted every $m + 1$ steps yields

$$E(x_{k+1}) \leq \left(\frac{b-a}{b+a}\right)^2 E(x_k)$$

where x_{k+1} is found from x_k by taking $m + 1$ C-G steps.

□ **C-G Algorithm** for nonquadratic problems:
Given x_0, compute $g_0 = [\nabla f(x_0)]^\top$, $d_0 = -g_0$.

$$\begin{aligned} a_k &= -\frac{g_k^\top d_k}{d_k^\top \nabla^2 f(x_k) d_k}; \quad x_{k+1} = x_k + a_k d_k \\ g_{k+1} &= [\nabla f(x_{k+1})]^\top \\ &\text{If} \quad k < n-1 \\ &\qquad \beta_k = \frac{g_{k+1} \nabla^2 f(x_k) d_k}{d_k^\top \nabla^2 f(x_k) d_k}; \quad d_{k+1} = -g_{k+1} + \beta_k d_k \\ &\text{else} \\ &\qquad x_0 \leftarrow x_n, \quad d_0 \leftarrow (-g_n), \quad \text{and repeat} \end{aligned}$$

• *Remarks:*

□ Restarts are preferred over continuous application of C-G.

□ Still requires $\nabla^2 f(x_k)$ and is only locally convergent

• **Important Variants:** Use line search to find a_k and alternative formulae to compute β_k (all equivalent in the quadratic case).

1. *Fletcher-Reeves:* $\beta_k = \frac{g_{k+1}^\top g_{k+1}}{g_k^\top g_k}$
2. *Polak-Ribiere:* $\beta_k = \frac{(g_{k+1} - g_k)^\top g_{k+1}}{g_k^\top g_k}$

□ Global Convergence using spacer steps arguments

□ Order 2 convergence with respect to restarts (n C-G steps = 1 Newton step)

• **Parallel Tangents (PARTAN) Algorithm:**

$$\begin{aligned} x_0 &\rightarrow x_1 \quad \text{(S-D)} \\ x_1 &\rightarrow y_1 \quad \text{(S-D)} \\ x_0, y_1 &\rightarrow x_2 \quad \text{(Line search)} \\ x_2 &\rightarrow y_2 \quad \text{(S-D)} \\ x_1, y_2 &\rightarrow x_3 \quad \text{(Line search)} \\ &\vdots \\ &\text{(Restart after } n \text{ steps)} \end{aligned}$$

□ Not very sensitive on line minimization, in contrast to other C-G methods.

□ Simple implementation.

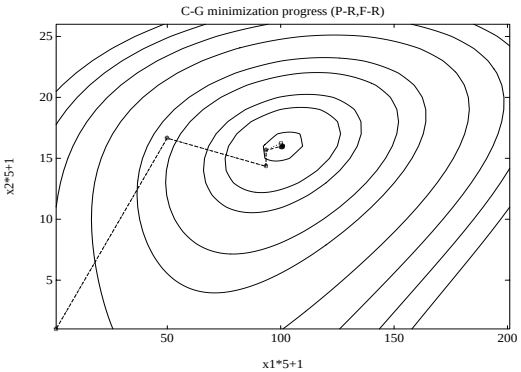
Conjugate Gradient Methods: Examples

46 C-G Methods (P-R, F-R)

For our example, both the P-R and F-R versions of Conjugate Gradient converged in 6 steps. (Sensitivity to line search accuracy.)

```
[x,niter,xs]=congr([0;0]  [,m,method]);
```

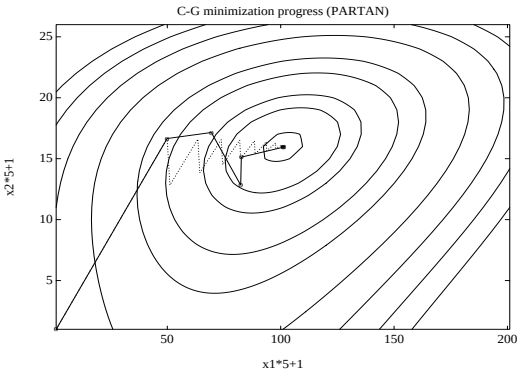
□ Select *method* = 0 (P-R) or 1 (F-R).
Steps to restart m (0=default); n or $n + 1$ are common choices for non-partial methods.



47 PARTAN Example

• The PARTAN algorithm also yields good convergence rates for the same simple example.

```
[x,niter,xsg]=partan([0;0], -1);  % S-D
[x,niter,xs]=partan([0;0]);
plot(xsg(:,1),xsg(:,2),':',xs(:,1),xs(:,2),'+')
```



Quasi-Newton Methods

• Attempting to build up an approximation to the inverse Hessian

$$x_{k+1} = x_k - a_k S_k [\nabla f(x_k)]^\top$$

□ S_k should be positive definite to ensure that $S_k \nabla f_k^\top$ is a descent direction.

□ Observation: For quadratic problems the minimizing a_k is

$$a_k = \frac{g_k^\top S_k g_k}{g_k^\top S_k Q S_k g_k}$$

Convergence ratio depends on the conditioning number of $S_k Q$.

□ If the eigenvalues of $S_k Q$ approach 1, we expect good linear or, even, superlinear convergence.

Notation:

□ $g_k = \nabla f(x_k)^\top$

□ $q_k = g_{k+1} - g_k$

□ $p_k = x_{k+1} - x_k$

□ $F_k = \nabla^2 f(x_k)$

48 Construction of the Inverse: Basic Ideas

• Taylor expansion of ∇f to first order:

$$\nabla f(x_{k+1})^\top \simeq \nabla f(x_k)^\top + \nabla^2 f(x_k)(x_{k+1} - x_k)$$

Assuming that the Hessian is (approximately) constant,

$$q_{k+1} = F p_k$$

Then n linearly independent p_k's (q_k's) would allow the construction of F (F^{-1}).

Consider a recursion of the form (rank one correction) [2]

$$S_{k+1} = S_k + a_k z_k z_k^\top$$

Construction of the inverse: some algebra

We want to find a_k, z_k such that $S_{k+1} q_i = p_i$, $i \le k$

$$p_k = S_{k+1} q_k = S_k q_k + a_k z_k z_k^\top q_k$$

1. Construct $a_k z_k z_k^\top$: $p_k - S_k q_k = a_k z_k z_k^\top q_k \Rightarrow$

$$(p_k - S_k q_k)(p_k - S_k q_k)^\top = a_k^2 z_k (z_k^\top q_k q_k^\top z_k) z_k^\top)$$

$$a_k z_k z_k^\top = \frac{(p_k - S_k q_k)(p_k - S_k q_k)^\top}{a_k (z_k^\top q_k)^2}$$

2. Compute $a_k (z_k^\top q_k)^2$: Premultiply with q_k

$$q_k^\top (p_k - S_k q_k) = a_k (z_k^\top q_k)^2$$

Thus,

$$S_{k+1} = S_k + \frac{(p_k - S_k q_k)(p_k - S_k q_k)^\top}{q_k^\top (p_k - S_k q_k)}$$

□ *Remark:* Clever but cannot ensure $S_k > 0$.

49 Davidon-Fletcher-Powell Method

• Rank two correction:

$$S_{k+1} = S_k + \frac{p_k p_k^\top}{p_k^\top q_k} - \frac{S_k q_k q_k^\top S_k}{q_k^\top S_k q_k}$$

□ Movement in F-conjugate directions (exact for quadratic functions). Relies on line search to show $p_k^\top g_{k+1} = 0$.

□ $S_k > 0 \Rightarrow S_{k+1} > 0$ provided that $p_k^\top q_k > 0$. The latter can be guaranteed with a sufficiently accurate line search.

50 Broyden-Fletcher-Goldfarb-Shanno Method

• Use similar ideas to develop an update for the Hessian itself instead of its inverse.

$$B_{k+1} = B_k + \frac{q_k q_k^\top}{q_k^\top p_k} - \frac{B_k p_k p_k^\top B_k}{p_k^\top S_k p_k}$$

□ Translate the result to an inverse Hessian update [3]

$$S_{k+1} = S_k + \left(\frac{1 + q_k^\top S_k q_k}{q_k^\top p_k}\right) \frac{p_k p_k^\top}{p_k^\top q_k} - \frac{p_k q_k^\top S_k + S_k q_k p_k^\top}{q_k^\top p_k}$$

Quasi-Newton Methods: The Broyden Family

• A collection of update laws combining DFP and BFGS methods:

$$S^\phi = (1 - \phi) S^{DFP} + \phi S^{BFGS}$$

After a lot of algebra,

$$S_{k+1}^\phi = \underbrace{S_k + \frac{p_k p_k^\top}{p_k^\top q_k} - \frac{S_k q_k q_k^\top S_k}{q_k^\top S_k q_k}}_{S_{k+1}^{DFP}} + \phi v_k v_k^\top$$

$$v_k = \sqrt{q_k^\top S_k q_k} \left(\frac{p_k}{p_k^\top q_k} - \frac{S_k q_k}{q_k^\top S_k q_k}\right)$$

□ Differences among methods are important only with inaccurate line search.

[2] Symmetric matrices can be diagonalized via a unitary similarity transformation ($Z^\top = Z^{-1}$) $S = Z^\top \Lambda Z = \sum_i \lambda_i z_i z_i^\top$.

[3] A useful identity: $(A + ab^\top) = A^{-1} - (A^{-1} a b^\top A^{-1})/(1 + b^\top A^{-1} a)$ (Matrix Inversion Lemma, or Sherman-Morrison formula).

□ $\phi \geq 0$ to ensure $S_k > 0$.
□ BFGS ($\phi \simeq 1$) is generally a preferred method
□ Partial updates (with restarts every $m < n$ steps), to alleviate storage requirements for large problems (save p_k, q_k only).

Self-Scaling Quasi-Newton Methods

• With partially updated Broyden methods, progress within a cycle may not be uniform; sensitivity to scaling.
□ E.g., Consider a quadratic function where F has large eigenvalues. Starting a Broyden method with $S_0 = I$, after m steps $S_m F$ will have m eigenvalues $\simeq 1$ and the rest will be large. Hence, the condition number of $S_m F$ will be large and the convergence of the partial Broyden method will be poor, even worse than steepest descent.

• To alleviate this problem, S_k can be rescaled at every step so as to ensure that the interval of eigenvalues of $S_k F$ contains 1.

51 A Self-Scaling Quasi-Newton Family

• With $\phi \in [0, 1]$ and v_k as before, select

$$S_{k+1} = \left(S_k - \frac{S_k q_k q_k^\top S_k}{q_k^\top S_k q_k} + \phi v_k v_k^\top \right) \gamma_k + \frac{p_k p_k^\top}{p_k^\top q_k}$$

where $\gamma_k > 0$. A suitable choice is:

$$\gamma_k = \frac{p_k^\top q_k}{q_k^\top S_k q_k}$$

□ $S_k > 0 \Rightarrow S_{k+1} > 0$ provided that $p_k^\top q_k > 0$.
□ F-orthogonality (conjugate behavior) is preserved, but $S_n \neq F^{-1}$.

52 Other Quasi-Newton Variants

• Memoryless: Reset $S_k = I$ after each step. (BFGS $\equiv$ C-G/P-R)
• Steepest Descent + Newton: Use Newton to minimize f on a linear variety $(x_k + Bu_k)$ where $\nabla^2 f$ is easily computed and then apply a steepest descent step.

Quasi-Newton Methods: Examples

53 Broyden Family Example

For our example, (with high-accuracy line search) both the BFGS and DFP algorithms converged in 6 steps, with or without scaling. (Very low tolerance may cause finite arithmetic-related problems near the end of the search.)

```
[x,niter,xs]=broyden([0;0]  [,m,method,scaling]);
```

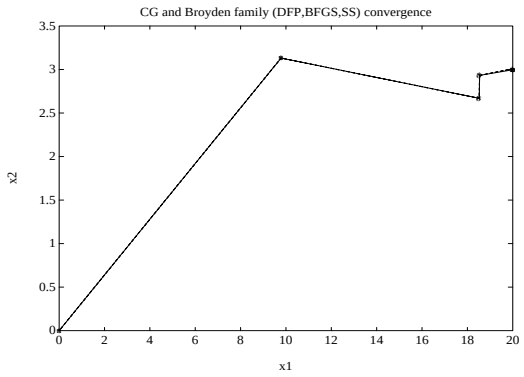
□ Select Steps to restart m (0=default). *method* = 0 (DFP) or 1 (BFGS); *scaling* = 0 (no) or 1 (yes).

54 A Classical "Nasty" Example

□ Minimize the *Rosenbrock's* function,

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

starting with $x = [-1.9; 2]$.
This is one of the classical benchmark optimization problems. The function contours not only contain a narrow valley (check $\nabla^2 f$ at the minimum [1;1]) but the valley is also "warped" as we move away from the minimum. This picture can be generated using the CONTOUR function of MATLAB:




```
[xt,yt]=meshdom(-3:0.2:3,-3:0.2:3);
z=100*(yt-xt.^2).^2+(1-xt).^2;
contour(z,[5,20,])
```

You may try the various algorithms on this example by setting

```
index=2;
```

in the functions *tfun* and *invhess*. Remember to *hold* the contours if you want to see the optimization progress against the contour plot. Also, to plot the optimization sequence in the same coordinates as the contours, use

```
plot((xs(:,1)+3)*5+1,(xs(:,2)+3)*5+1,'+')
```

Note that the sequential optimization converges quickly to the valley, where Hessian information is essential to avoid slow convergence.

55 A Least Squares Fit Example

• Here we are faced with a standard parameter estimation problem: *Given "input-output" measurements of a process, find a model that describes it.*
• Common approach: Begin by assuming a certain model structure, parametrized by an adjustable parameter vector (a.k.a. DOF), and then estimate these parameters so that the predicted input-output pairs agree (as much as possible) with the actual measurements.
• There are several important theoretical issues associated with this approach.
□ Sufficient Degrees of Freedom, Parsimony, Excitation
□ Type of Approximation
□ Measurement Noise/modeling error properties
□ Interpolation/Extrapolation error properties
Without getting too deep in the details, let us work on a practical example.

56 Silicon Oxidation Model Identification

• A simplifed model for the silicon oxidation process is (Deal-Grove)

$$\dot{z} = \frac{B}{2z + A}$$

where $z(t)$ is the oxide thickness as a function of time. Assuming $z(0) = 0$, the solution of the diff.eqn. is

$$t = \frac{z^2(t)}{B} + \frac{Az(t)}{B}$$

A, B are (Arrhenius-type) functions of the processing temperature T with nominal expressions

$$A = 3500e^{-5300/T} \qquad B = 3.0E11e^{-25000/T}$$

Thus, a model for the oxidation process, under constant temperature can be written as $f(x, \theta) = 0$, where

$$f(x, \theta) = x_2^2 e^{-\theta_1 + \frac{\theta_2}{x_3}} + x_2 e^{-\theta_3 + \frac{\theta_4}{x_3}} - x_1$$

and $x_1 = t$, $x_2 = z$, $x_3 = T$. The nominal value of θ is

$$\theta_* = [26.427, 25, 18.2665, 19.7]^\top$$

• We are asked to find the model parameters, given several input-output measurements (t, z, T) produced, for example, by a sequence of batch experiments.
• In an ideal setting where $f(x(k), \theta_*) = 0$, for all k, this problem can be formulated as a solution of a set of nonlinear equations $f(x(k), \theta) = 0$.
• Basic questions: uniqueness of the solution and its smooth dependence on the measurements. An affirmative answer is guaranteed when the vectors $\nabla_\theta f(x(k), \theta_*)$ are linearly independent. [4]
• In a more realistic situation where the measurements are "noisy," we expect that the ideal model satisfies $f(x(k), \theta_*) \simeq 0$ in some sense. A "reasonable" modeling objective is to find θ so as to minimize an error functional of the form $E(\theta) = \|\{f(x(k), \theta)\}_k\|$, e.g.,

$$E(\theta) = \sum_k |f(x(k), \theta)|^2$$

• The gradient and Hessian of $E(\theta)$ are computed as follows:

$$\nabla_\theta E(\theta) = 2 \sum_k f(x(k), \theta) \nabla_\theta f(x(k), \theta)$$

$$\begin{aligned} \nabla_\theta^2 E(\theta) \quad &= \quad 2 \sum_k [\nabla_\theta f(x(k), \theta)]^\top \nabla_\theta f(x(k), \theta) \ldots \\ &\quad +2 \sum_k f(x(k), \theta) \nabla_\theta^2 f(x(k), \theta) \end{aligned}$$

• Checking the sufficient conditions for a minimum at θ_*, we observe that in the ideal case, $\nabla_\theta E(\theta_*) = 0$ and

$$\nabla_\theta^2 E(\theta_*) = 2 \sum_k [\nabla_\theta f(x(k), \theta_*)]^\top \nabla_\theta f(x(k), \theta_*)$$

The latter is a sum of rank-one matrices that is guaranteed to be positive definite under the condition that the gradients $\nabla_\theta f(x(k), \theta_*)$ are linearly independent. Continuity also implies that for sufficiently low noise levels (n), the positive definiteness will be preserved in the non-ideal case (note that the minimizer may also be biased by $O(n)$ in the noisy case).

[4] This question is addressed by the implicit function theorem; alternatively, the Jacobian of $f(x(k), \theta_*)$ has null space $\{0\}$ or, simply, it is invertible when the number of measurements equals the number of parameters.

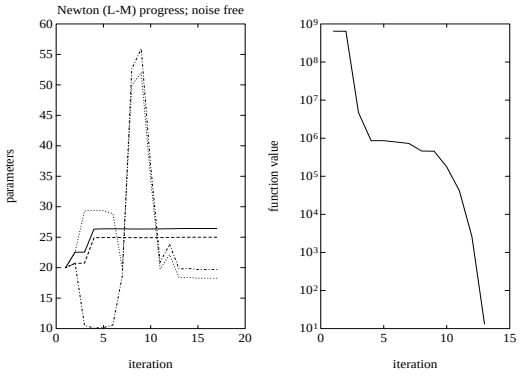
• Implications on the selection of experimental measurements:

$$[\nabla_\theta f(x(k), \theta)]^\top = \begin{bmatrix} -z(k)^2 e^{-\theta_1 + \frac{\theta_2}{T(k)}} \\ \frac{z(k)^2}{T(k)} e^{-\theta_1 + \frac{\theta_2}{T(k)}} \\ -z(k) e^{-\theta_3 + \frac{\theta_4}{T(k)}} \\ \frac{z(k)}{T(k)} e^{-\theta_3 + \frac{\theta_4}{T(k)}} \end{bmatrix}$$

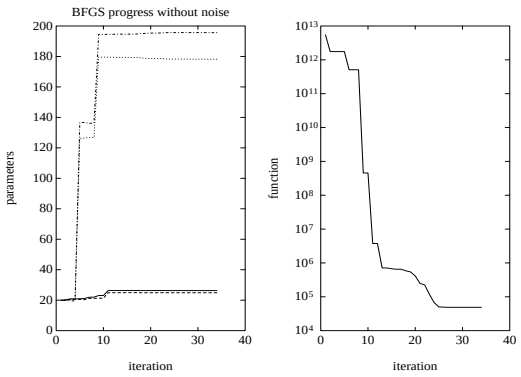
i.e., must have measurements at different temperatures. (reasonable!)
This agrees with the intuitive concept that sufficient information (independent measurements) must be collected in order to solve for the parameters (parsimony, persistent excitation in recursive identifiers)
• It is now a straightforward task to apply our minimization algorithms to solve this problem.

```
th0=[20;20;20;20];
[th,n,ths,es]=sgnewt(th0);
[th,n,ths,es]=sbroyden(th0);
```

• **Noise-Free Case Observations:**
□ The Hessian is very ill-conditioned and becomes indefinite quickly as we move away from the solution.
□ Newton's method (L-M) requires about 15 iterations for convergence. Scaled Broyden-type algorithms require very low tolerance (1.e-8) to converge near the minimizer in 500 or more steps. However, essential convergence to "low functional error" values is achieved much faster.
□ Newton's progress towards the minimum is considerably more "violent" than C-G and Broyden.

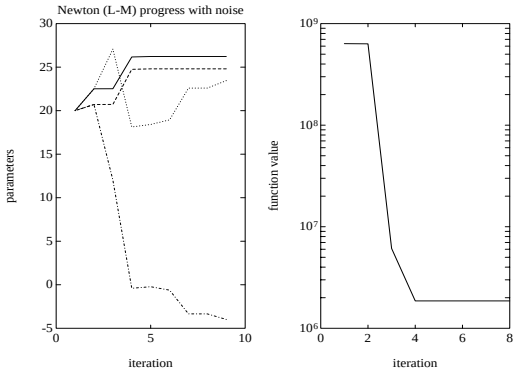


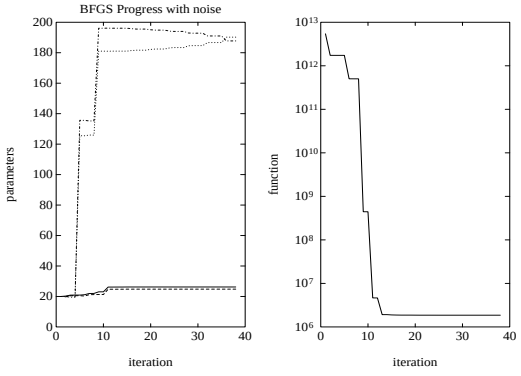
• **Noisy Case Observations:**
□ For noisy measurements, change $data(0)$ to $data(0.05)$ in $totfun$.
□ The noise introduces bias in the solution. $(\nabla_\theta^2 E(\theta_*) \not> 0)$
□ Newton's (L-M) algorithm converges in 8 steps, while 34 steps are required for the BFGS and 51 for its autoscaling version (again, essential convergence of $E(\theta)$ is considerably faster).
□ While all algorithms achieve roughly the same level of error $E(\theta)$ the corresponding minimizers are quite different in the third and fourth components. This is related to the poor conditioning of the Hessian and is indicative



of a poor "signal-to-noise" ratio in the corresponding directions. It also raises questions on the suitability of the model for the assumed range of operating conditions and noise properties.

□ From a different point of view, we can say that the model attempts to capture a very fine detail of the process that is well below the noise level





Part II: Outline

- Constrained Minimization
  - — Equality and Inequality Constraints
  - — First and Second Order Conditions
  - — Lagrange Multipliers, Sensitivity
- Basic Algorithms: Generalities
  - — Feasible Directions
  - — Active Set Methods, Gradient Projection
  - — Penalty and Barrier Methods

Constrained Minimization: Equality Constraints

• Basic Problem:

$$\begin{array}{ll} \min_{x\in\Omega\subseteq\mathbf{R}^n} & f(x) \\ \text{s.t.} & h_i(x)=0,\ i=1,\dots,m \end{array}$$

57 Equality Constraints: 1st Order Necessary Conditions

• Optimality on the surface $S=\{x : h(x)=0\}$: $\nabla f(x)d \geq 0$ for any feasible direction d.

□ For small deviations, S "looks" approximately like its tangent plane (TP_S). Since the latter contains both d and $-d$, we get that $[\nabla f(x)]^\top$ should be perpendicular to TP_S.

□ Assuming that x is a *regular* point of the constraints,[5]

$$TP_S = M = \{y : \nabla h(x)y = 0\}$$

The orthogonal complement of M, where $[\nabla f(x)]^\top$ should belong, is the span of the columns of $[\nabla h(x)]^\top$. Thus:

• Let x_* be a local extremum of f subject to $h(x)=0$ and a regular point of the constraints. Then, there is a $\lambda \in \mathbf{R}^m$ s.t.

$$\nabla f(x_*) + \lambda^\top \nabla h(x_*) = 0$$

• **Lagrangian:** $\ell(x,\lambda) \triangleq f(x) + \lambda^\top h(x)$

□ Local extrema of the constrained problem are stationary points of the Lagrangian:

$$\nabla_x \ell(x,\lambda) = 0 \ \ ; \ \ \nabla_\lambda \ell(x,\lambda) = 0$$

58 Equality Constraints: 2nd Order Necessary Conditions

• Let x_* be a local minimum of f s.t. $h(x)=0$ and a regular point. Then, there is a $\lambda \in \mathbf{R}^m$ s.t.

$$\nabla f(x_*) + \lambda^\top \nabla h(x_*) = 0$$

$$L(x_*) \triangleq \nabla^2 f(x_*) + \lambda^\top \nabla^2 h(x_*) \geq 0 \text{ on } M$$

i.e., for all $y \in M$, $y^\top L(x_*) y \geq 0$.

59 Equality Constraints: 2nd Order Sufficient Conditions

• Suppose there exists an x_* s.t. $h(x_*)=0$ and a $\lambda \in \mathbf{R}^m$ s.t.

$$\nabla f(x_*) + \lambda^\top \nabla h(x_*) = 0$$

$$L(x_*) \triangleq \nabla^2 f(x_*) + \lambda^\top \nabla^2 h(x_*) > 0 \text{ on } M$$

[5] Regular point: $h(x)=0$ and $\nabla h_i(x)$ are linearly independent. For example, with $x \in \mathbf{R}^2$, $h(x)=x_1$, S is the x_2-axis, $\nabla h = (1,0)$ and every point on S is regular; if, instead, $h(x)=x_1^2$, S is again the x_2-axis but $\nabla h = (0,0)$ and no point is regular.

i.e., for all $y \in M$, $y \neq 0$, there holds $y^\top L(x_*)y > 0$. Then x_* is a strict local minimum of f subject to $h(x) = 0$.

60 Eigenvalues in the Tangent Subspace

We seek to determine whether $x^\top Ax > 0$ for all $x \neq 0$, $h^\top x = 0$. $(h^\top : m \times n)$. Let N be a basis of the null space of $h^\top$. Then all $x \in \mathcal{N}(h^\top)$ can be written as Ny where $y \in \mathbf{R}^m$. Then the condition $x^\top Ax > 0$ becomes $y^\top(N^\top AN)y > 0$, for all y, or, $N^\top AN$ is a positive definite matrix.

□ A basis of the null space can be computed using the command $NULL$ in MATLAB. Alternatively, one could form the matrix $[h|I_{n\times n}]$ and perform a Gram-Schmidt orthogonalization on its columns. The last $n - m$ vectors of this procedure form a basis of $\mathcal{N}(h^\top)$.

61 Sensitivity

• Let $f, h \in C^2$ and consider the family of the problems

$$\min f(x) \ ; \quad \text{s.t. } h(x) = c$$

Suppose that for $c = 0$, x_* is a local, regular solution that, together with the associated Lagrange multiplier λ satisfy the 2nd order sufficient conditions. Then for $|c|$ sufficiently small, there is a local minimum $x(c)$ depending continuously on c and such that $x(0) = x_*$. Furthermore,

$$\nabla_c f(x(c))|_{c=0} = -\lambda^\top$$

Constrained Minimization: Inequality Constraints

• Basic Problem:

$$\begin{array}{ll} \min_{x\in\Omega\subseteq\mathbf{R}^n} & f(x) \\ \text{s.t.} & h_i(x) = 0, \ i = 1, \ldots, m \\ & g_i(x) \leq 0, \ i = 1, \ldots, p \end{array}$$

• A parallel development of conditions.

• Active Inequality Constraints: $g_j(x_*) = 0, \ j \in J$.

□ *Regular points:* $\nabla h_i(x_*)$ and $\nabla g_j(x_*) = 0, \ j \in J$ are linearly independent.

62 1st Order Necessary Conditions

• *Kuhn-Tucker:* Let x_* be a local minimum of f subject to $h(x) = 0$, $g(x) \leq 0$ and a regular point of the constraints. Then, there is a $\lambda \in \mathbf{R}^m$ and a $\mu \in \mathbf{R}^p$, $\mu \geq 0$ s.t.

$$\begin{aligned} \nabla f(x_*) + \lambda^\top \nabla h(x_*) + \mu^\top \nabla g(x_*) &= 0 \\ \mu^\top g(x_*) &= 0 \end{aligned}$$

□ Since $\mu \geq 0$ and $g(x) \leq 0$, a component of μ may be nonzero only if the corresponding constraint is active.

□ $\nabla f(x_*) + \lambda^\top \nabla h(x_*)$ must be zero in the "direction" of non-active constraints.

63 2nd Order Necessary Conditions

• Let x_* be a local minimum of f s.t. $h(x) = 0, g(x) \leq 0$ and a regular point. Then, there is a $\lambda \in \mathbf{R}^m$ and a $\mu \in \mathbf{R}^p$ s.t. in addition to the first order necessary conditions, there holds

$$L(x_*) \triangleq \nabla^2 f(x_*) + \lambda^\top \nabla^2 h(x_*) + \mu^\top \nabla^2 g(x_*) \geq 0 \text{ on } M$$

where M is the tangent subspace of the active constraints at x_*

64 2nd Order Sufficient Conditions

• Suppose there exists an x_* s.t. $h(x_*) = 0$ and a $\lambda \in \mathbf{R}^m$ and a $\mu \in \mathbf{R}^p$ s.t.

$$\begin{aligned} \nabla f(x_*) + \lambda^\top \nabla h(x_*) + \mu^\top \nabla g(x_*) &= 0 \\ \mu^\top g(x_*) &= 0 \\ \mu &\geq 0 \\ L(x_*) \triangleq \nabla^2 f(x_*) + \lambda^\top \nabla^2 h(x_*) + \mu^\top \nabla^2 g(x_*) &> 0 \text{ on } M' \end{aligned}$$

where $M' = \{y : \nabla h(x_*)y = 0, \ \nabla g_j(x_*)y = 0, \ j \in J'\}$ and $J' = \{j : g_j(x) = 0, \ \mu_j > 0\}$.

Then x_* is a strict local minimum of f subject to $h(x) = 0, g(x) \leq 0$.

□ Positive definiteness of L is required on the *larger* subspace that is tangent to all active constraints excluding any *degenerate* ones, i.e., inequality constraints with zero Lagrange multipliers.

65 Sensitivity

• Let $f, h \in C^2$ and consider the family of the problems

$$\min f(x) \ ; \quad \text{s.t. } h(x) = c, g(x) \leq d$$

Suppose that for $c = 0, d = 0$, x_* is a local, regular solution that, together with the associated Lagrange multipliers λ, μ satisfy the 2nd order sufficient conditions and no active inequality constraints are degenerate. Then for $|c|, |d|$ sufficiently small, there is a local minimum $x(c, d)$ depending continuously on c and d and such that $x(0, 0) = x_*$. Furthermore,

$$\nabla_c f(x(c, d))|_{c=0,d=0} = -\lambda^\top$$

$$\nabla_d f(x(c))|_{c=0,d=0} = -\mu^\top$$

Constrained Minimization: Example

66 Minimum distance from a set

• Let $M = \{x \in \mathbf{R}^2 : x^\top Px \leq 1\}$, $P > 0$. Given a point $x_0 \notin M$, find a point $x_* \in M$ such that the distance (Euclidean) of x_* form x_0 is minimum.

$$\begin{array}{ll} \min & (x - x_0)^\top (x - x_0) \\ \text{s.t.} & x^\top Px \leq 1 \end{array}$$

• Lagrangian: $\ell(x, \lambda) = (x - x_0)^\top (x - x_0) + \lambda(x^\top Px - 1)$

□ Stationarity conditions:

$$\begin{aligned} [\nabla_x \ell(x, \lambda)]^\top = 2[(x - x_0) + \lambda Px] &= 0 \\ \lambda(x^\top Px - 1) &= 0 \\ \lambda &\geq 0 \end{aligned}$$

□ The Hessian $L(x) = 2(I + \lambda P)$ is p.d. for any $\lambda \geq 0$.

□ The optimum x is: $x_* = (I + \lambda P)^{-1} x_0$.

• Note that $\lambda = 0$ is not a solution since $x_0 \notin M$. Hence, the constraint must be active ($x_*^\top Px_* = 1$).

□ Substituting, we obtain an equation for λ, whose solution will determine x_*.

$$x_0(I + \lambda P)^{-1} P (I + \lambda P)^{-1} x_0 = 1$$

• Identical results would be obtained by using convexity and orthogonality ideas.
• Properties of solutions: There exists a unique $\lambda > 0$ satisfying the above equation.

□ Let $q(\lambda) = x_0(I + \lambda P)^{-1}P(I + \lambda P)^{-1}x_0 - 1$. Its derivative is [6]

$$\frac{dq(\lambda)}{d\lambda} = -2x_0(I + \lambda P)^{-1}P(I + \lambda P)^{-1}P(I + \lambda P)^{-1}x_0$$

which is negative for all $x_0 \neq 0$ and all $\lambda \geq 0$. Hence $q(\lambda)$ is strictly decreasing and can have only one zero in $\lambda \geq 0$.

□ The solution for λ can be found iteratively by using Newton's method. A minor difficulty appears in that other solutions may exist for $\lambda < 0$. To ensure that λ_k remains positive for all k we may implement a constrained version of the algorithm as follows:

```
dk = q(lam)/dq(lam);
    while lam - dk <= 0,
         dk=dk/2;
     end
lam = lam - dk
```

□ This problem reappears in a more interesting way when the distance from M is to be maximized. An added difficulty in this case is that the optimum is not necessarily unique.
• **Numerical Example:** Let $P = \text{diag}(1, 2)$ and $x_0 = [1, 1]^\top$. Solving the necessary conditions for λ we need to find the real roots of

$$\frac{1}{(1 + \lambda)^2} + \frac{2}{(1 + 2\lambda)^2} = 1$$

which translate to finding the roots of the polynomial

$$4\lambda^4 + 12\lambda^3 + 7\lambda^2 - 2\lambda - 2$$

Its real roots are -2.1127 and 0.4666. The positive root yields the desired minimizer $x_* = [0.6818, 0.5173]^\top$, while the negative root produces the maximizer $[-0.8987, 0.3100]$.
• This algorithm is implemented as a MATLAB function to compute projections on a possibly degenerate, ellipsoid (see *orppr*1).

Algorithms for Constrained Optimization

67 Primal Methods: Generalities

• Searching for the solution in the feasible region directly (i.e., working with x).

□ Generate feasible points at every iteration.

□ Can often guarantee that if the sequence has a limit, then that is a local constrained minimum.

□ Do not require special problem structure (e.g., convexity).

□ Competitive convergence rates especially for linear constraints.

□ Require a feasible point to start.

[6] Differenciate the identity $A(\ell)A^{-1}(\ell) = I$ to obtain $dA^{-1}(\ell)/d\ell = -A^{-1}(\ell)[dA(\ell)/d\ell]A^{-1}(\ell)$.

□ In general, remaining in the feasible set requires elaborate precautions.

68 Feasible Direction Methods

• $x_{k+1} = x_k + a_k d_k$; d_k: feasible direction.
• $\min f(x)$; s.t. $Ax \leq b$.

□ Compute a feasible direction as the solution of the linear program

$$\begin{aligned} \min \quad & \nabla f(x_k)d \\ \text{s.t.} \quad & A_i d \leq b_i \ , \quad i \in I \\ & \sum |d_i| = 1 \end{aligned}$$

where I is the set of indices of the active constraints.

□ Susceptible to "jamming." Possible, but complicated to avoid.

69 Active Set Methods

• Inactive constraints are essentially ignored.

□ Determine current working set, a subset of the current active constraints

□ Movement on the tangent surface of the working set to an improved point.

□ At new boundaries constraints are added to the working set.

□ After minimization, constraints are dropped if their Lagrange multipliers are negative.

70 Gradient Projection

• For linear constraints, project $-\nabla f(x_k)$ onto the tangent subspace of the active constraints.

□ Let A_q be the $q \times n$ sub-matrix of the active constraints. Then

$$P_k = I - A_q^\top (A_q A_q^\top)^{-1} A_q$$

is the projection matrix to the tangent subspace and

$$d_k = -P_k \nabla f(x_k)$$

is a descent direction if non-zero ($\nabla f(x_k)d_k = -|d_k|^2$).
• Select the step size so as to either minimize f or encounter a new constraint.
• Termination or constraint elimination depends on the sign of the Lagrange multipliers.
• *Algorithm:*
1. Find M, the tangent subspace of active constraints
2. Set $d_k = -P_k \nabla f(x_k)$
3. If $d \neq 0$ compute:
   $a_1 = \max\{a : x_k + ad_k \text{ is feasible}\}$
   $a_2 = \arg\min_{a \in [0, a_1]} f(x_k + ad_k)$
   Set $x_{k+1} = x_k + a_2 d_k$ and return to (1).
4. If $d = 0$, compute $\lambda = -(A_q A_q^\top)^{-1} A_q \nabla f(x_k)^\top$
   If $\lambda_j \geq 0$ for all active constraints, stop.
   Otherwise drop the constraint with the most negative component and return to (2).

□ Since only one constraint is added/deleted each time, the projection matrix can be computed recursively (cmp. recursive least squares.)
• For Nonlinear Constraints, project the negative gradient on the tangent subspace to find an approximately feasible descent direction, d_k. If movement along d_k produces non-feasible points, return to the constraint surface by moving orthogonally to the tangent subspace.

□ This idea can be described as an implicit line search:

$$\min_{a\geq 0} f[x_k + ad_k - \nabla h(x_k)^\top \beta(a)]$$

where $\beta(a)$ is defined such that $[x_k+ad_k-\nabla h(x_k)^\top\beta(a)] \in \{x : h(x) = 0\}$ (the latter set describing both equality and active inequality constraints).

□ A simple algorithm to compute β relies on successive approximation: Letting $y = x_k + ad_k$,

$$h(y - \nabla h(x_k)^\top \beta) \simeq h(y) - \nabla h(x_k)\nabla h(x_k)^\top \beta$$

Hence, a suitable first approximation of β is

$$\beta_1 = [\nabla h(x_k)\nabla h(x_k)^\top]^{-1} h(y)$$

Successive substitution yields

$$y_{j+1} = y_j - \nabla h(x_k)^\top [\nabla h(x_k)\nabla h(x_k)^\top]^{-1} h(y_j)$$

a sequence that will converge to the desired point for a small enough.

• Asymptotic rate of convergence: $\left(\frac{\lambda_{max}-\lambda_{min}}{\lambda_{max}+\lambda_{min}}\right)^2$, where $\lambda_{max}, \lambda_{min}$ are the maximum and minimum eigenvalues of $L(x_*)$, the Hessian of the Lagrangian, restricted to the tangent subspace.
• Satisfactory general routines implementing these concepts are quite complex.

Penalty and Barier Methods

---

71 Penalty Methods

• Replace the original constrained problem $\min_{x\in S} f(x)$, by an unconstrained problem that puts a large penalty on the points outside the set.

$$\min f(x) + cP_S(x) \; ; \quad c > 0$$

• Solve a sequence of such unconstrained problems, parametrized by c, with $c_k \to \infty$. The sequence of solutions "should" also converge to the solution of the constrained problem.
• Conditions: (i) P continuous, (ii) $P(x) \geq 0$ for all x, (iii) $P(x) = 0$ if and only if $x \in S$.
Further, $\{c_k\}$ should be strictly increasing, tending to infinity.
• Penalty Functions: Define $g_i^+(x) = \max[0, g_i(x)]$.
Then $P(x) = \gamma(g^+(x))$ where $\gamma(y) = y^\top \Gamma y$, $\Gamma > 0$ or $\gamma(y) = \sum y_i^\epsilon$, $\epsilon > 0$
• Limit points of the sequence generated by the penalty method are solutions of the original problem.
• The Hessian (possibly discontinuous) has eigenvalues that tend to infinity as $c_k \to \infty$.

72 Barrier Methods

• Replace the original constrained problem $\min_{x\in S} f(x)$, by an unconstrained problem that puts a large penalty on the points approaching the boundary of the set.

$$\min f(x) + \frac{1}{c} B_S(x) \; ; \quad c > 0$$

• Solve a sequence of such unconstrained problems, parametrized by c, with $c_k \to \infty$. The sequence of solutions "should" also converge to the solution of the constrained problem.
• Conditions: (i) B continuous, (ii) $B(x) \geq 0$ for all x, (iii) $B(x) \to \infty$ as $x \to \partial S$.
Further, $\{c_k\}$ should be strictly increasing, tending to infinity.
• Typical Barrier Function: $B(x) = -\sum_i \frac{1}{g_i(x)}$
• Restricted to sets S with non-empty interior and such that any boundary point can be approached by a sequence of interior points.
• Limit points of the sequence generated by the barrier method are solutions of the original problem.
Remarks:

□ Penalty methods approach the constrained minimum from the outside while barrier methods approach it from the inside.

□ As $c_k \to \infty$ the Hessians in both penalty and barrier methods become extremely ill-conditioned. Only C-G and Quasi-Newton type methods are suitable for numerical implementation.

□ Barrier methods with a log barrier function have found important applications in convex optimization. Fast and reliable convergence has been achieved using Newton methods to iterate *both* the minimizer for each c_k *as well as* the initial guess (i.e., searching for the value of the function $x_*(c_k)$ at $c_k = 0$ where x_* is is the minimization argument of the unconstrained problem).

Part III: Outline

Special Topics

- Least Squares Problems with Constraints
  - — Simple Constraint Sets (hyperplanes,half-spaces,ellipsoids)
  - — Oblique Projections
- Convex Optimization
  - — Cutting Plane Algorithm
  - — Ellipsoid Algorithm
- Quadratic Programming
  - — Linear Constraints
  - — The Role of QP in General Constrained Minimization

Constrained Least Squares

• Problem Statement:

$$\begin{array}{ll} \min & \|Ax - b\|_2^2 \\ \text{s.t.} & x \in S \end{array}$$

where S is a (simple) convex set, e.g., a Hyperplane or an Ellipsoid. For simplicity, we will also assume that $A^\top A$ is invertible (A is 1-1).[7]

□ Observation:

$$\|Ax-b\|_2^2 = (Ax-b)^\top(Ax-b) = x^\top A^\top Ax - 2x^\top A^\top b + b^\top b$$

The above expression can be written in a "canonical" quadratic form

$$(x-c)^\top Q(x-c) + const.$$

with $Q = A^\top A$, $c = Q^{-1}A^\top b$.
Thus, the original minimization is equivalent to

$$\min_{x\in S} \quad (x-c)^\top Q(x-c)$$

Since $A^\top A$ is nonsingular (hence p.d.),

$$\|x-c\|_Q \stackrel{\Delta}{=} [(x-c)^\top Q(x-c)]^{1/2}$$

is a *weighted* Euclidean norm. In such cases, it is convenient to change coordinates as $\bar{x} = \sqrt{Q}x$.

73 S is a Subspace

• Here $S = \{x : Px = 0\}$, P is onto, i.e., $PP^\top$ is invertible. (Shifted Subspaces can be handled similarly by an appropriate translation of the origin).

□ Using $\sqrt{Q}$ for coordinate transformation, our problem becomes

$$\begin{array}{ll} \min & \|\bar{x}-\bar{c}\|_2^2 \\ \text{s.t.} & \bar{x} \in \{\bar{x} : \bar{P}\bar{x} = 0\} \end{array}$$

where $\bar{x} = \sqrt{Q}x$, $\bar{c} = \sqrt{Q}c$, $\bar{P} = P\sqrt{Q}^{-1}$.
The last is a classical minimum norm problem, with solution[8]

$$\bar{x} = [I - \bar{P}^\top(\bar{P}\bar{P}^\top)^{-1}\bar{P}]\bar{c}$$

Translating the result back to the original coordinates,

$$x = [I - Q^{-1}P^\top(PQ^{-1}P^\top)^{-1}P]c$$

□ The above formula defines an *oblique projection* of c onto S, the null space of P.

74 S is a Half-space

• Here $S = \{x : p^\top x \le 0\}$, $p \neq 0$. (Again, shifted half-spaces can be handled similarly.)

□ The solution is a simple extension of the previous case:

$$x = \begin{cases} c & \text{if } p^\top c \le 0 \\ \left[I - \frac{Q^{-1}pp^\top}{p^\top Q^{-1}P}\right]c & \text{otherwise} \end{cases}$$

□ Unfortunately, such a simple relation is not available for polytopes where the "edges" or "corners" introduce problems. In such a case, general QP or convex optimization methods are applicable.

[7] The same principles apply in the general case, but the formulae are not simple; general solutions are obtained via, e.g., SVD or QR decompositions.

[8] See also the later section on QP where the same solution is obtained through Lagrange multipliers.

75 S is an Ellipsoid

• Here $S = \{x : x^\top Px \le 1\}$, $P = P^\top > 0$. (The usual remark for shifted ellipsoids...) Using our $\sqrt{Q}$ coordinate transformation, the original problem is equivalent to

$$\begin{array}{ll} \min & \|\bar{x}-\bar{c}\|_2^2 \\ \text{s.t.} & \bar{x} \in \{\bar{x} : x^\top \bar{P}\bar{x} \le 1\} \end{array}$$

where $\bar{P} = \sqrt{Q}^{-1}P\sqrt{Q}^{-1}$. This problem was solved (efficiently) in Example 66. Note that, again, by re-expressing the solution in the original coordinates, the recursion can be performed without computing matrix square-roots.

• Observe that in all cases, the same pattern emerges; that is, the constrained solution is an *oblique* projection of the LS solution onto the convex set. Furthermore, the weighting matrix of the oblique projection is the matrix $Q = A^\top A$ appearing in the quadratic cost objective. [9]

Convex Optimization

• Problem Statement

$$\min_y f(y), \quad \text{s.t.} y \in R$$

where f is a convex function and R is a convex set.

• Whenever necessary, the above problem can always be transformed to its canonical equivalent:

$$\min_x c^\top x, \quad \text{s.t.} x \in S$$

where S is a convex set. Such a transformation is obtained by noting that the original problem is equivalent to

$$\min_{r,y} r, \quad \text{s.t.} f(y) \le r \text{ and } y \in R$$

with $c^\top = [1, 0, \ldots, 0]$, $x = [r, y^\top]^\top$ and

$$S = \{x : f(y) \le r\} \cap \{x : y \in R\}$$

(Recall that since f is convex, $\{f(y) \le r\}$ is convex, and the intersection of convex sets is convex.)

76 Cutting Plane Methods: Kelley's Algorithm

• Here we construct an improving series of approximating linear programs (LP) whose solution converges to that of the original problem.

• **Kelley's Algorithm:**

□ Let P_k be a polytope containing S; ($S = \{x : g(x) \le 0\}$).

□ Solve the LP:

$$\min_x c^\top x, \quad \text{s.t.} x \in P_k$$

producing a solution, say, x_k.

□ If $x_k \in S$ stop; the solution is optimal.

□ If $x_k \notin S$ find a separating hyperplane (supporting S) for x_k and S and append it to the constraints:

[9] It is also interesting to extrapolate the form of the solution in the case where Q is singular. The latter will be a minimum distance problem between the convex set S and the linear variety containing the (now multiple) LS solutions of the unconstrained problem.

By convexity, $g(x) \geq g(w)\nabla g(w)(x-w)$. Hence, setting

$$i_* = \arg\max_i g_i(x_k)(> 0)$$

$$H_k = \{x : \nabla g_{i_*}(x_k)(x - x_k) + g_{i_*}(x_k) \leq 0\}$$

we have that H_k is such a separating hyperplane. We may now define $P_{k+1} = P_k \cap H_k$ (by appending the new inequality constraint in the polytope matrix) and repeat the process.

• The polytope matrix is increasing in size, linearly with every step. Efficient "book-keeping" may be necessary to delete redundant constraints and maintain a matrix that is as small as possible. Although the problem may seem intractable in the general case, in practice an adequately accurate solution is obtained before the storage and computational requirements become excessive.

• Differentiability of g is not required. The gradient used in the above expressions may be converted to a *subgradient* of g (any vector such that $g(x) \geq g(w)\nabla g(w)(x-w)$). Computing subgradients is similar and only incrementally harder than computing ordinary gradients.

77 Ellipsoid Method

• Here we approximate the solution set by an ellipsoid E_k so that at every step $x_* \in E_k$. E_k describes the "uncertainty" about the solution at the k-th step.

• Using convexity we construct an algorithm such that if the center of the ellipsoid is not a solution, E_{k+1} is "smaller" than E_k. The comparison is in terms of volumes which is proportional to the determinant of P_k^{-1}, for an ellipsoid

$$E_k = \{x : (x - c_k)^\top P_k (x - c_k) \leq 1\}$$

Note that E_{k+1} is not necessarily contained in E_k.

• The ellipsoid updates are performed by updating its center c_k and matrix P_k^{-1} (storage and computational requirements are fixed). The idea behind the updates is to use the objective function gradient (if $c_k \in S$) or the constraint gradient (if $c_k \notin S$) to define a half space where the optimum cannot lie. In this manner, the original ellipsoid is cut in half and the resulting set is then enclosed in another ellipsoid of minimum volume. [10]

• The sequence of ellipsoids is converging (in volume) and eventually collapses to a point or a variety. In any case, the algorithm provides a lower and an upper bound of the minimum value of the objective and the iteration is stopped when these values are sufficiently close to each other and $c_k \in S$.

• Although not very efficient, the algorithm yields an approximate solution (to a desired accuracy level) in polynomial time. Its convergence rate is defined in terms of ellipsoid volumes and satisfies

$$Vol(E_{k+1}) \leq Vol(E_k) e^{-\frac{1}{2n}}$$

where n is the dimension of x. Typically, its performance is adequate for small-to-medium size problems.

[10] For the case $c_k \notin S$, the value $g(c_k)$ can also be used to define a "deep-cut" (similar to Kelley's algorithm) and, thus, speed-up the convergence.

• The ellipsoid recursions are straightforward, though laborious, to derive. They are given next with the notation $Q_k = P_k^{-1}$ since only the inverse of P_k is updated.

78 Ellipsoid Algorithm

For the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in S = \{x : g(x) \leq 0\} \end{array}$$

f, g convex. Define

□ *I. Constraint Iteration*

1. Let $h = \nabla g_i(c_k)$ where $i = \arg\max_i [g_i(c_k)]$.
2. If $g_i(c_k) > \sqrt{h^\top Q_k h}$, quit; the feasible set is empty.
3. Else, define the "deep-cut" parameter $a = g_i(c_k)/\sqrt{h^\top Q_k h}$. $(a \in (0, 1])$
4. Set $\tilde{h} = h/\sqrt{h^\top Q_k h}$.

□ *II . Objective Iteration*

1. Let $h = \nabla f(c_k)^\top$.
2. Set the "deep-cut" parameter $a = 0$.
3. Set $\tilde{h} = h/\sqrt{h^\top Q_k h}$.

□ *III. Main Algorithm*[11]

0. c_0, Q_0: An ellipsoid containing feasible minimizers; (usually $Q_0 = \rho I$, $\rho \gg 1$).

Set, $n = dim(x) > 1$, $k = 0, 1, \ldots$ and repeat:

1. If $g(c_k) \not\leq 0$, perform a "constraint iteration"

else

2. Perform an "objective iteration"
3. $c_{k+1} = c_k - \frac{1+na}{n+1} Q_k \tilde{h}$
4. $Q_{k+1} = \frac{n^2(1-a^2)}{n^2-1}\left(Q_k - \frac{2(1+na)}{(n+1)(1+a)} Q_k \tilde{h}\tilde{h}^\top Q_k\right)$

until, $g(c_k) \leq 0$ and $\sqrt{\nabla f(c_k) Q_k \nabla f(c_k)^\top} \leq \epsilon$, where ϵ is the desired accuracy level.

□ Upper and lower bounds on the minimum of the objective function can be computed as

$$U = \min_k f(c_k^f) \;\; ; \;\; L = \max_k \left(f(c_k^f) - \sqrt{\nabla f(c_k^f) Q_k \nabla f(c_k^f)^\top} \right)$$

where these computations use only feasible points c_k^f.

79 Ellipsoid Method: Example

The problem of Example 66 is revisited with M now being the intersection of two ellipsoids. Even though the constraint set does not have a smooth boundary, convex optimization is still applicable.

□ To run the example, use:

```
lmiinit           %  Plot the constraint set
LMI_proj([10;10],1.e-4)
       %  Uses LMI_upd to perform the updates
```

□ The projection set is defined in *setcon*. If altered, make the corresponding changes in *lmiinit*. When finished, issue a *holdoff* command to release the previously held plots (by *lmiinit*).

[11] The algorithm is applicable for $dim(x) \geq 2$. The case $dim(x) = 1$ must be handled separately, by an interval search.

Quadratic Programming

• Problem Statement:

$$\min_x \frac{1}{2}x^\top Qx + x^\top c, \quad \text{s.t. } Ax \leq b$$

• For equality constraints, the Lagrange multiplier method yields the following conditions:

$$\begin{aligned} Qx + c + A^\top \lambda &= 0 \\ Ax - b &= 0 \end{aligned}$$

The solvability of this equation and optimality of the extremum is guaranteed by Q being p.d. on the null space of A.

□ In the special case $Q > 0$, the minimizing x can be obtained explicitly:

$$\begin{aligned} x &= -[I - Q^{-1}A^\top (AQ^{-1}A^\top)^{-1}](Q^{-1}c) \\ &\quad +Q^{-1}A^\top (AQ^{-1}A^\top)^{-1}b \end{aligned}$$

which is an appropriately shifted, oblique projection of the unconstrained minimizer.

• When inequalities are present, the projection on the polytope of the constraints can be efficiently computed by an active set method.

□ Letting $x_{k+1} = x_k + d_k$, a suitable descent direction is found by solving

$$\min_x \frac{1}{2}d_k^\top Q d_k + d_k^\top g_k, \quad \text{s.t. } A_w d_k = 0$$

where $g_k = (c + Qx_k)$ and A_w denotes the matrix of the active constraints.

□ If the solution for d_k, (obtained by, e.g., the oblique projection formula) is $d_k = 0$ then x_k is optimal for the working set. Compute the associated Lagrange multipliers and set $\lambda_q = \min_i \lambda_i$ where i takes values in the working set of inequalities. If $\lambda_q \geq 0$, stop; x_k is optimal. Otherwise, drop the constraint q from the working set and repeat the first step.

□ Find the maximum $a_k \in (0, 1]$ such that $A(x_k + a_k d_k) \leq b$.

□ Set $x_{k+1} = x_k + a_k d_k$. If $a_k < 1$, add the corresponding new constraint to the working set and repeat the first step.

• A general constrained minimization problem, can be solved as a sequence of QP's (Recursive Quadratic Programming), each one being solved to determine a suitable descent direction and step size. This approach is motivated by the observation that the recursive (Newton) solution of the 1st order necessary conditions can be written as a QP with $Q = \nabla_x^2 \ell$, A being the concatenation of the constraint gradients $(\nabla h, \nabla g)$ and b being the concatenation of the constraint values $(h,\ g)$, all evaluated at the current x_k, λ_k, μ_k. Variations of this method include Quasi-Newton principles applied to update $\nabla_x^2 \ell$ recursively (see, e.g., BFGS method).

References

1. D.G. Luenberger, *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachussetts, 1984.

2. D.G. Luenberger, *Optimization by Vector Space Methods.* John Wiley and Sons, New York, 1969.

3. A.V. Fiacco and G.P. Mc Cormick, *Nonlinear Programming. Sequential Unconstrained Minimization Techniques.* SIAM, Classics in Applied Mathematics, Philadelphia, 1990. (includes applications to constrained problems)

4. A. Bjork, *Least Squares Methods.* Handbook of Numerical Analysis, Vol.1, Ciarlet and Lions Editors, Elsevier, N. Holland, 1987.
(excellent treatment of least squares with computationally efficient solutions)

5. S.P. Boyd and C.H. Barratt, *Linear Controller Design. Limits of Performance.* Prentice Hall, 1991.
(applications of convex analysis in control problems)

6. S.P. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory.* SIAM, Studies in Applied Mathematics, Philadelphia, 1994.
(formulation of control problems as convex minimization problems)