

Numerical Optimisation: Background

Marta M. Betcke

m.betcke@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 1

Mathematical optimisation problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{aligned} c_i(x) &= 0, & i \in \mathcal{E} \\ c_i(x) &\geq 0, & i \in \mathcal{I} \end{aligned}$$

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$: objective function
- $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$: constraint function,
 $i \in \mathcal{E}$ equality constraints,
 $i \in \mathcal{I}$ inequality constraints.
- $x \in \mathbb{R}^n$: optimisation variable

Optimal solution x^* has the smallest value of f among all x which satisfy the constraints.

Example: geodesics

Geodesics are the shortest surface paths between two points.

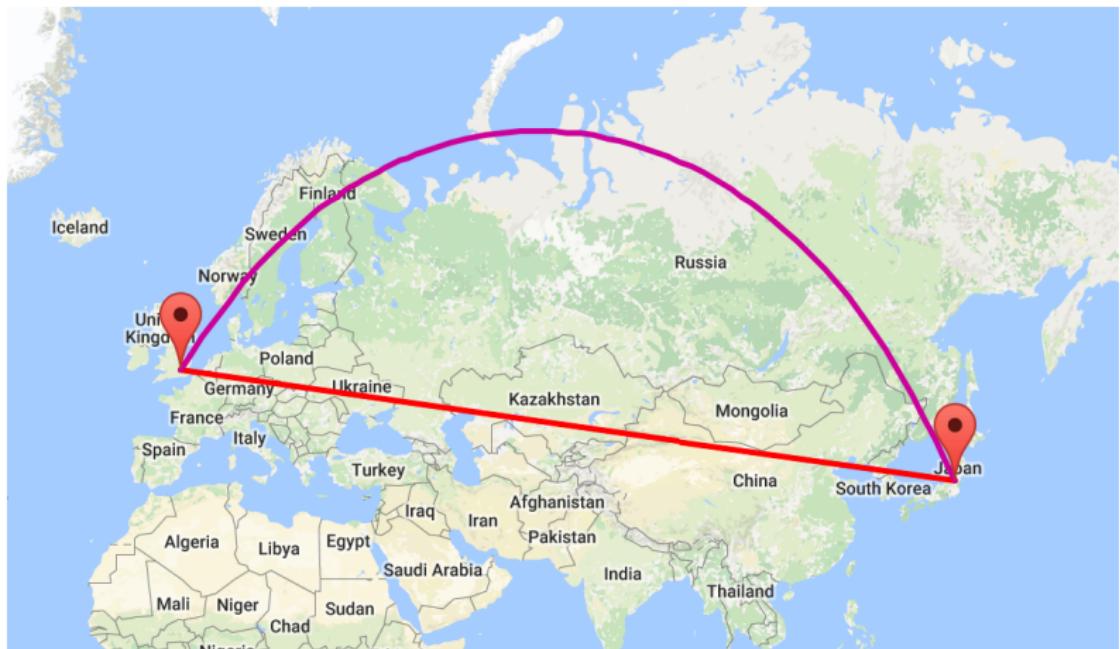


Figure: <https://academo.org/demos/geodesics/>

A very short and incomplete early history

Source <http://www.mitrikitti.fi/opthist.html>

- **Antiquity: geometrical optimisation problems**

300 BC Euclid: considers the minimal distance between a point and a line; proves that a square has the greatest area among the rectangles with given total length of edges

- **before Calculus of Variations: isolated optimization problems**

1615 J. Kepler: optimal dimensions of wine barrel (with smallest variation of volume w.r.t. barrel parameters).¹ Early version of the secretary problem (optimal stopping problem) when he started to look for a new wife

1636 P. Fermat shows that at the extreme point the derivative of a function vanishes. In 1657 Fermat shows that light travels between two points in minimal time.

¹ <http://www.maa.org/press/periodicals/convergence/>

kepler-the-volume-of-a-wine-barrel-solving-the-problem-of-maxima-wine-barrel-design

Source <http://www.mitrikitti.fi/opthist.html>

- **Calculus of Variations**

I. Newton (1660s) and G.W. von Leibniz (1670s) create mathematical analysis that forms the basis of calculus of variations (CoV). Some separate finite optimization problems are also considered

1696 Johann and Jacob Bernoulli study Brachistochrone's problem, calculus of variations is born

1740 L. Euler's publication begins the research on general theory of calculus of variations

A very short and incomplete early history cont.

Source <http://www.mitrikitti.fi/opthist.html>

- **Least squares**

1806 A.-M. Legendre presents the least square method, which also J.C.F. Gauss claims to have invented. Legendre made contributions in the field of CoV, too

- **Linear Programming**

1826 J.B.J. Fourier formulates LP-problem for solving problems arising in mechanics and probability theory

1939 L.V. Kantorovich presents LP-model and an algorithm for solving it. In 1975 Kantorovich and T.C. Koopmans get the Nobel price in economics for their contributions to LP-problems

1947 G. Dantzig, who works for US air-force, presents the Simplex method for solving LP-problems, von Neumann establishes the theory of duality for LP-problems

Example: transportation problem

- 2 factories, F_i
- 12 retail outlets, R_j
- each factory F_i can produce up to a_i tones of a certain compound per week
- each retail outlet R_j has a weekly demand of b_j tones of the compound
- the cost of shipping of one tone of the compound from F_i to R_j is c_{ij}

Goal: what is the optimal amount to ship from each factory to each outlet which satisfies demand at minimal cost.

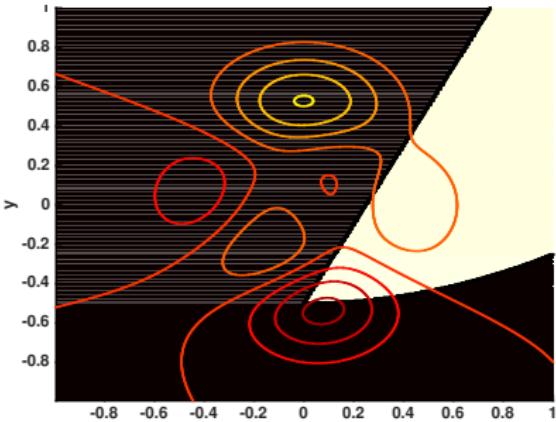
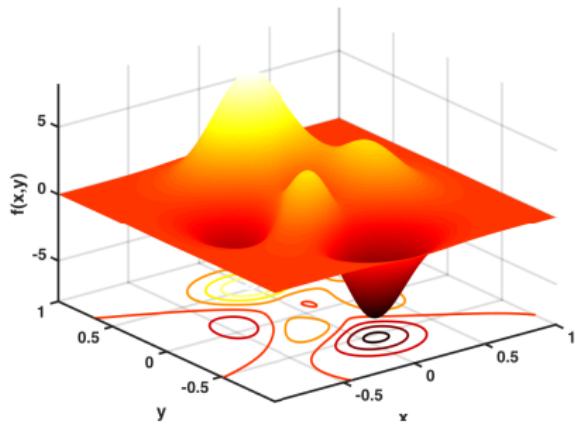
Example: transportation problem cont.

$$\begin{aligned} & \min \quad \sum_{ij} c_{ij} x_{ij} \\ \text{subject to } & \sum_{j=1}^{12} x_{ij} \leq a_i, \quad i = 1, 2 \\ & \sum_{i=1}^2 x_{ij} \geq b_j, \quad j = 1 \dots 12 \\ & x_{ij} \geq 0, \quad i = 1, 2, \quad j = 1 \dots 12. \end{aligned}$$

Linear programming problem because the objective function and all constraints are linear.

Example: nonlinear optimisation

$$\begin{aligned} & \min f(x, y) \\ \text{subject to } & -y + 2x - \frac{1}{2} \geq 0 \\ & y - \frac{1}{4}x^2 + \frac{1}{2} \geq 0. \end{aligned}$$



Convexity

A set $\mathbb{S} \subset \mathbb{R}^n$ is **convex** if for any two points $x, y \in \mathbb{S}$ the line segment connecting them lies entirely in \mathbb{S}

$$\alpha x + (1 - \alpha)y \in \mathbb{S}, \quad \forall \alpha \in [0, 1].$$

Examples:

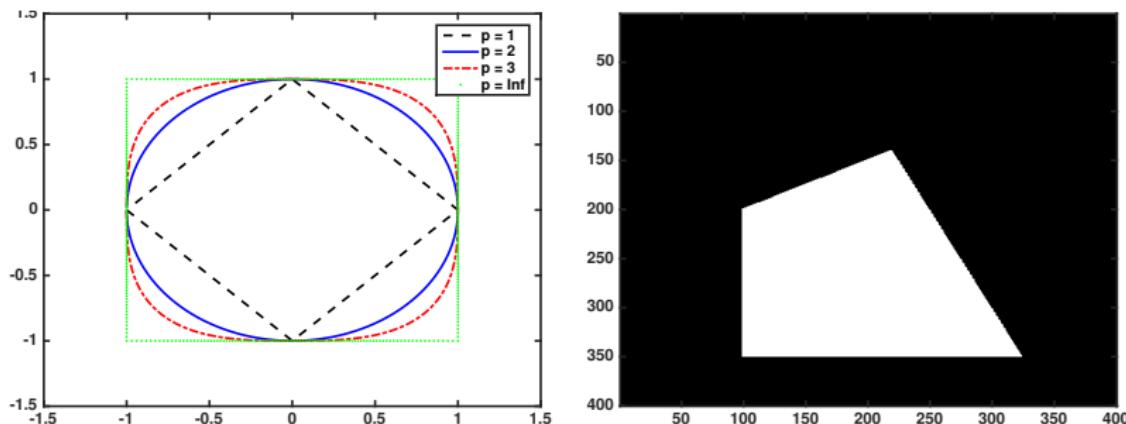


Figure: (a) unit ball $\{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$, $p \geq 1$; (b) polyheadron $\{x \in \mathbb{R}^n : Ax = b, Cx \leq d\}$

Convexity

A function f is **convex** if

- its domain \mathbb{S} is a convex set,
- for any two points $x, y \in \mathbb{S}$,

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall \alpha \in [0, 1].$$

A function f is **strictly convex** if for $x \neq y$

$$f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y), \quad \forall \alpha \in (0, 1).$$

A function f is **concave** if $-f$ is convex.

Examples:

- linear function $f(x) = c^T x + \alpha$, where $c \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$
- convex quadratic function $f(x) = x^T H x$, where $H \in \mathbb{R}^{n \times n}$ symmetric positive (semi)definite

Classification of optimisation problems

- **convex vs non-convex**
- **smooth vs non-smooth**
- **constrained vs unconstrained**
- **linear vs quadratic vs nonlinear**
- **small vs large scale**
- **local vs global**
- stochastic vs **deterministic**
- discrete vs **continuous**

Unconstraint minimisation

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a smooth function for which we can evaluate f and its derivatives at any given point $x \in \Omega \subseteq \mathbb{R}^n$.

Unconstraint optimisation problem

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} f(x). \quad (1)$$

A point x^* is a **global minimiser** if

$$f(x^*) \leq f(x), \quad \forall x \in \Omega \subseteq \mathbb{R}^n.$$

A point x^* is a **local minimiser** if

$$\exists \mathcal{N}(x^*) : f(x^*) \leq f(x), \quad \forall x \in \mathcal{N}(x^*),$$

$\mathcal{N}(y)$ is a neighbourhood of y (an open set which contains y).

Unconstraint minimisation

A point x^* is a **strict (or strong) local minimiser** if

$$\exists \mathcal{N}(x^*) : f(x^*) < f(x), \forall x \in \mathcal{N}(x^*), x \neq x^*.$$

Examples:

- $f(x) = 2$: every point is a (weak) local minimiser
- $f(x) = (x - 2)^4$: $x^* = 2$ is a strict local minimiser (also a global one)
- $f(x) = \cos(x)$: $x^* = \pi + 2k\pi, k \in \mathbb{Z}$, are all strict local minimisers (but not strict global on \mathbb{R})

Unconstraint minimisation

A point x^* is an **isolated local minimiser** if

$$\exists \mathcal{N}(x^*) : x^* \text{ is the only local minimiser in } \mathcal{N}(x^*).$$

Some strict local minimisers are not isolated e.g.

$$f(x) = x^4 \cos(1/x) + 2x^4, \quad f(0) = 0$$

has a strict local minimiser at $x^* = 0$ but there are strict local minimisers (albeit $f(x_j) \geq x_j^4 > 0 = f(x^*)$) at nearby points $x_j \rightarrow 0, j \rightarrow \infty$.

However, all isolated local minimisers are strict.

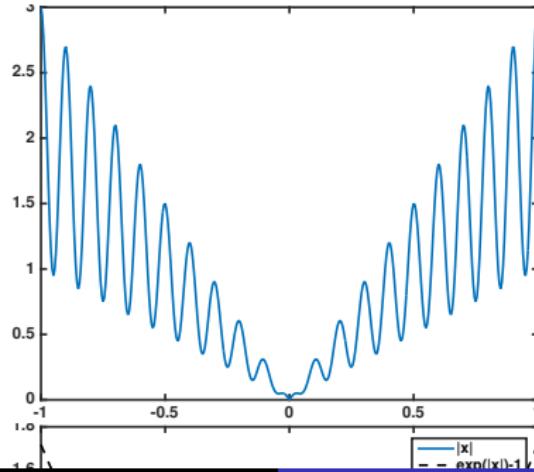
Unconstraint minimisation

Difficulties with global minimisation:

$$f(x) = (\cos(20\pi x) + 2)|x|$$

has a unique global minimiser $x^* = 0$, but the algorithms usually get trapped into one of the many local minima.

For convex functions, every local minimiser is also a global minimiser.



Taylor theorem

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable. Then for $p \in \mathbb{R}^n$ we have

$$f(x + p) = f(x) + p^T \nabla f(x + tp)$$

for some $t \in (0, 1)$.

If moreover f is twice continuously differentiable, we also have

$$\nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 f(x + tp)p dt.$$

and

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

for some $t \in (0, 1)$.

Theorem [1st order necessary condition]

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable in an open neighbourhood of a **local minimiser** x^* , then $\nabla f(x^*) = 0$.

Proof: [by contradiction]

Suppose that $\nabla f(x^*) \neq 0$ and define $p = -\nabla f(x^*)$. Note that $p^T \nabla f(x^*) = -\|\nabla f(x^*)\|_2^2 < 0$. Furthermore, as ∇f is continuous near x^* , there exists $T > 0$ such that

$$p^T \nabla f(x^* + tp) < 0, \quad t \in [0, T].$$

By Taylor's theorem, for any $\bar{t} \in (0, T]$ we have

$$f(x^* + \bar{t}p) = f(x^*) + \bar{t}p^T \nabla f(x^* + tp), \quad t \in (0, \bar{t}).$$

Hence $f(x^* + \bar{t}p) < f(x^*)$ for all $\bar{t} \in (0, T]$, and we have found a direction leading away from x^* along which f decreases which is in contradiction with x^* being a local minimiser.

1st oder necessary condition

We call x^* a **stationary point** if $\nabla f(x^*) = 0$.

By Theorem [1st order necessary condition] any local minimiser is a stationary point. The converse is in general not true.

Theorem [2nd order necessary condition]

If x^* is a **local minimiser** of f and $\nabla^2 f$ exists and is continuous in an open neighbourhood of x^* , then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.

Proof: [by contradiction]

By Theorem [1st order necessary condition] we have $\nabla f(x^*) = 0$.

Assume $\nabla^2 f(x^*)$ is not positive semidefinite. Then there exists a vector p such that $p^T \nabla^2 f(x^*) p < 0$, and because $\nabla^2 f$ is continuous near x^* , there exists $T > 0$ such that

$$p^T \nabla^2 f(x^* + tp) p < 0 \text{ for all } t \in [0, T].$$

By Taylor theorem, we have for any $\bar{t} \in (0, T]$ and some $t \in (0, \bar{t})$

$$f(x^* + \bar{t}p) = f(x^*) + \bar{t}p^T \nabla f(x^*) + \frac{1}{2}\bar{t}^2 p^T \nabla^2 f(x^* + tp)p < f(x^*).$$

We have found a decrease direction for f away from x^* which contradicts x^* being a local minimiser.

Theorem [2nd order sufficient condition]

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with $\nabla^2 f$ continuous in an open neighbourhood of x^* . If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then x^* is a **strict local minimiser** of f .

Proof:

Because the Hessian $\nabla^2 f$ is continuous and positive definite at x^* , we can choose a radius $r > 0$ so that $\nabla^2 f(x)$ remains positive definite for all x in an open ball $\mathcal{B}_2(x^*, r) = \{y : \|y - x^*\|_2 < r\}$. For any nonzero vector $p \neq 0$, $\|p\|_2 < r$, $x^* + p \in \mathcal{B}_2(x^*, r)$ and

$$f(x^* + p) = f(x^*) + p^T \nabla f(x^*) + \frac{1}{2} p^T \nabla^2 f(x^* + tp)p \quad (2)$$

$$= f(x^*) + \frac{1}{2} p^T \nabla^2 f(x^* + tp)p, \quad (3)$$

for some $t \in (0, 1)$.

Furthermore, $x^* + tp \in \mathcal{B}_2(x^*, r)$ thus $p^T \nabla^2 f(x^* + tp)p > 0$ and therefore $f(x^* + p) > f(x^*)$.

necessary vs sufficient condition

2nd order sufficient condition guarantees a stronger statement than the necessary conditions (strict local minimiser).

A strict local minimiser may fail to satisfy the sufficient conditions:

$$f(x) = x^4, \quad f'(x) = 4x^3, \quad f''(x) = 12x^2$$

$x^* = 0$ is a strict local minimiser while $f''(x^*) = 0$ thus it satisfies the necessary but not the sufficient conditions.

Implications of convexity

If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex, any local minimiser x^* is also a global minimiser of f . If, f is also differentiable, then any stationary point x^* is a global minimiser.

Proof: [by contradiction]

Suppose x^* is a local but not a global minimiser. Then

$\exists z \in \mathbb{R}^n : f(z) < f(x^*)$. For all x on a line segment joining x^* and z i.e.

$$\mathcal{L}(x^*, z) = \{x : x = \lambda z + (1 - \lambda)x^*, \quad \lambda \in (0, 1]\},$$

by convexity of f we have

$$f(x) \leq \lambda f(z) + (1 - \lambda)f(x^*) < f(x^*).$$

For any neighbourhood $\mathcal{N}(x^*) \cap \mathcal{L}(x^*, z) \neq \emptyset$, hence
 $\exists x \in \mathcal{N}(x^*) : f(x) < f(x^*)$ and x^* is not a local minimiser.

Implications of convexity

Proof: cont.

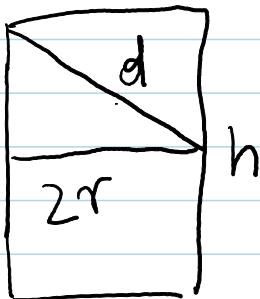
For the second part, we suppose that x^* is not a global minimiser.
For all z chosen as before by convexity of f it follows

$$\begin{aligned}\nabla f(x^*)^T(z - x^*) &= \frac{d}{d\lambda} f(x^* + \lambda(z - x^*)) \Big|_{\lambda=0} \\ &= \lim_{\lambda \rightarrow 0} \frac{f(x^* + \lambda(z - x^*)) - f(x^*)}{\lambda} \\ &\leq \lim_{\lambda \rightarrow 0} \frac{\lambda f(z) + (1 - \lambda)f(x^*) - f(x^*)}{\lambda} \\ &= f(z) - f(x^*) < 0.\end{aligned}$$

Hence $\nabla f(x^*) \neq 0$ and x^* is not a stationary point.

Example : Kepler's barrel

$$V = \pi r^2 h$$



$$d^2 = \left(\frac{h}{2}\right)^2 + (2r)^2$$

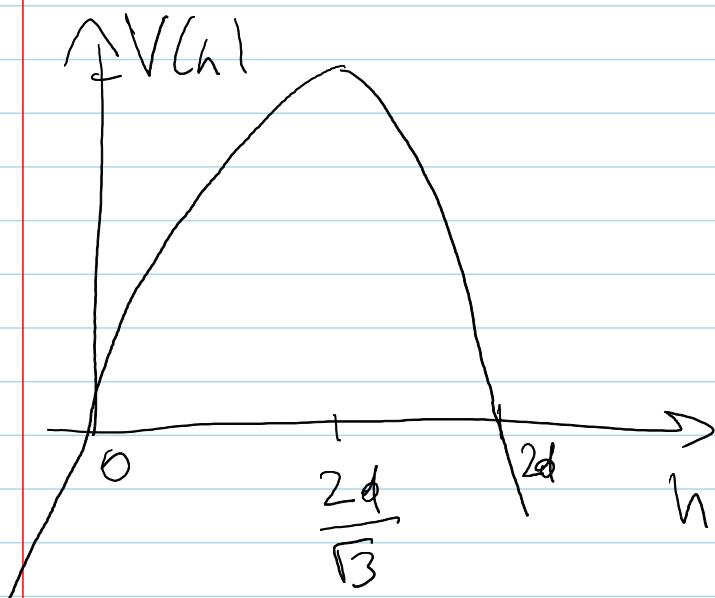
$$\rightarrow r^2 = d^2/4 - h^2/16$$

$$V(h) = \frac{\pi h}{4} \left(d^2 - \frac{h^2}{4} \right)$$

$$V'(h) = \frac{\pi}{4} d^2 - \frac{3\pi}{16} \cdot h^2$$

$$V'(h) = \frac{\pi}{4} \left(d^2 - \frac{3}{4} h^2 \right) = 0$$

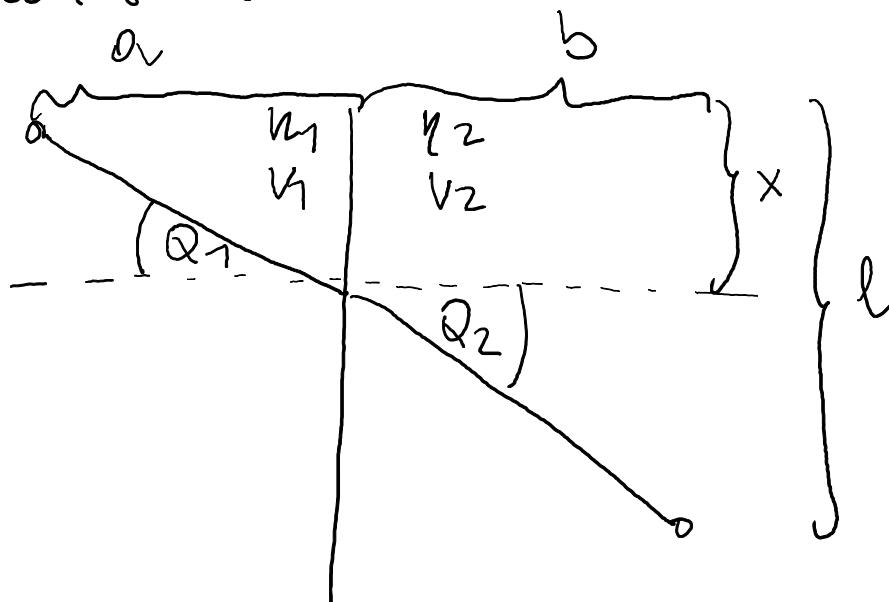
$$\rightarrow h = \pm \frac{2d}{\sqrt{3}}$$



Example: Fermat's principle

(light travel along the fastest path)

Snell's law:



$$v_1 = \frac{c}{n_1}$$

$$v_2 = \frac{c}{n_2}$$

c - speed of sound in vacuum

$$t = \frac{s}{v}$$

$$\min_x t = \frac{\sqrt{x^2 + a^2}}{v_1} + \frac{\sqrt{(l-x)^2 + b^2}}{v_2}$$

$$\frac{dt}{dx} = 0$$

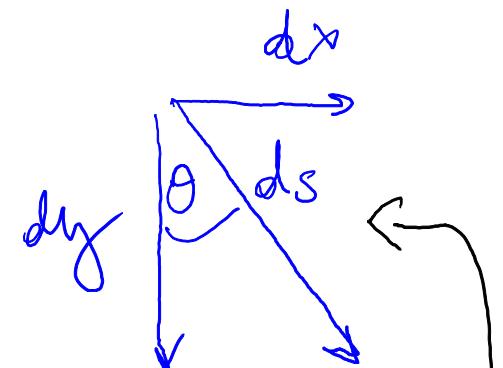
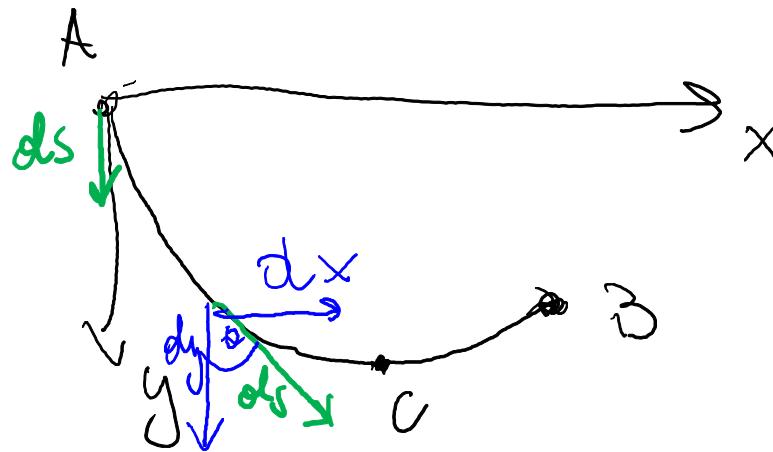
$$0 \Leftrightarrow \frac{dt}{dx} = \frac{1}{v_1} \frac{x}{\sqrt{x^2 + a^2}} + \frac{1}{v_2} \frac{l-x}{\sqrt{(l-x)^2 + b^2}}$$

$$\sin \theta_1 \quad \quad \quad -\sin \theta_2$$

$$\Rightarrow \frac{\sin \theta_1}{v_1} = \frac{\sin \theta_2}{v_2}$$

Example : Brachistochrone

Path of the fastest descent of a point mass "m" under gravity



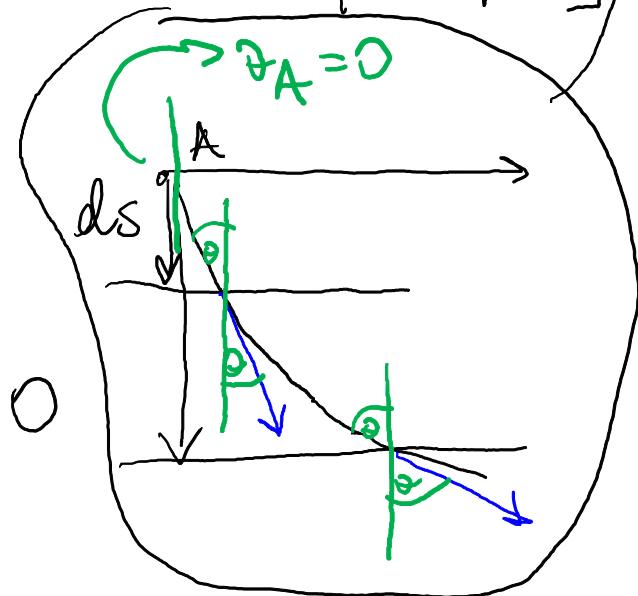
Energy conservation: $\frac{1}{2}mv^2 = m \cdot g \cdot y$

$$\frac{\sin \theta}{v} = \text{const}$$

[Fermat's principle]

$$\Rightarrow \frac{\sin \theta_{\max}}{v_{\max}}$$

$$ds \downarrow \Rightarrow \theta_A = 0, v_A = 0$$



C

$$\xrightarrow{ds} \Rightarrow \theta_{\max} = \frac{\pi}{2}, v < v_{\max}$$

$$\frac{1}{v_{\max}} = \frac{1}{v} \cdot \frac{dx}{ds}$$

$$\sin \theta$$

Get differential form of curve $y(x)$

$$v = \sqrt{2gy}$$

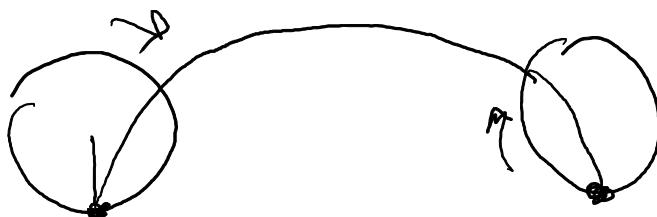
$$ds = \sqrt{dx^2 + dy^2}$$

$$\sqrt{2} ds^2 = v_{max}^2 dx^2$$

$$(v_{max}^2 - v^2) dx^2 = v^2 dy^2$$

$$dx = \sqrt{\frac{2gy}{v_{max}^2 - y}} dy$$

→ cycloid generated by a circle
with diameter y_{max}



Calculus of variations (CoV)

Lagrange, Euler 1756 [name giving lecture
analytic approach]

$$J[y] = \int_{x_1}^{x_2} L(x, y(x), y'(x)) dx$$

If $J[y]$ attains a local min at f

$$J[f] \leq J[f + \varepsilon \eta], \quad \varepsilon > 0, \text{ small}$$

where $\eta(x)$ a function $\eta(x_1) = \eta(x_2) = 0$

$$\Phi(\varepsilon) = J[f + \varepsilon \eta]$$

and $\Phi(\varepsilon)$ attains local min at $\varepsilon=0$

$$\Phi'(0) = \left. \frac{d\Phi}{d\varepsilon} \right|_{\varepsilon=0} = \int_{x_1}^{x_2} \left. \frac{dL}{d\varepsilon} \right|_{\varepsilon=0} dx = 0$$

$$\begin{aligned} \frac{dL}{d\varepsilon} &= \underbrace{\frac{\partial L}{\partial y} \frac{dy}{d\varepsilon}}_{\eta} + \underbrace{\frac{\partial L}{\partial y'} \cdot \frac{dy'}{d\varepsilon}}_{=\eta'} \\ &= \eta' \end{aligned}$$

$$\begin{aligned} \text{Hence: } \int_{x_1}^{x_2} \left. \frac{dL}{d\varepsilon} \right|_{\varepsilon=0} dx &= \int_{x_1}^{x_2} \left(\frac{\partial L}{\partial f} \eta + \frac{\partial L}{\partial f'} \cdot \eta' \right) dx \\ &= \int_{x_1}^{x_2} \frac{\partial L}{\partial f} \eta dx + \underbrace{\left. \frac{\partial L}{\partial f'} \eta \right|_{x_1}^{x_2}}_{=0} - \int_{x_1}^{x_2} \eta \frac{d}{dx} \frac{\partial L}{\partial f'} dx \end{aligned}$$

$$= \int_{x_1}^{x_2} \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \frac{\partial L}{\partial f'} \right) \eta(x) dx = 0$$

by fundamental thm. of calculus

$$\frac{\partial L}{\partial f} - \frac{d}{dx} \frac{\partial L}{\partial f'} = 0 \quad (\text{E.L. eq})$$

Fundamental thm. of calculus:

For $f \in C([a, b])$:

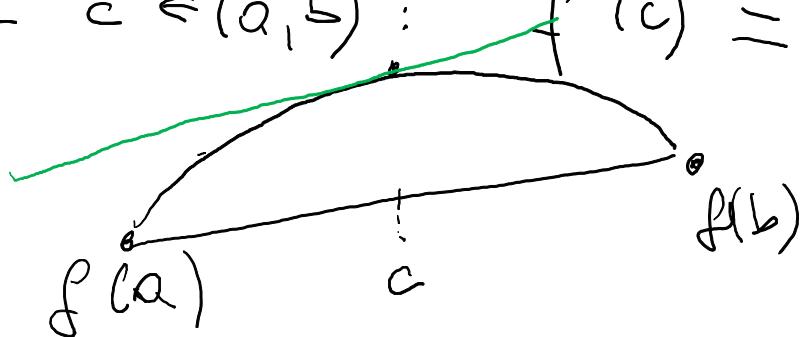
$$\int_a^b f(x) \cdot h(x) dx = 0$$

$$\forall h \in C_0^*(a, b) \quad * : \{\infty, 2, 1, 0\}$$

then $f \equiv 0$.

Mean value thm.

Let $f: [a, b] \rightarrow \mathbb{R}$ be a continuous on $[a, b]$ and diff. on (a, b) , then $\exists c \in (a, b)$: $f'(c) = \frac{f(b) - f(a)}{b - a}$.



Numerical Optimisation: Line search methods

Marta M. Betcke

`m.betcke@ucl.ac.uk`,

Kiko Rullan, Bolin Pan

`f.rullan@cs.ucl.ac.uk`, `bolin.pan.15@ucl.ac.uk`

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 2 & 3

Descent direction

Descent direction is a vector $p \in \mathbb{R}^n$ for which the function decreases.

From Taylor's theorem

$$\begin{aligned} f(x_k + \alpha p) &= f(x_k) + \alpha p^T \nabla f(x_k) + \frac{1}{2} \alpha^2 p^T \nabla^2 f(x_k + tp)p, \quad t \in (0, \alpha) \\ &= f(x_k) + \alpha \underbrace{p^T \nabla f(x_k)}_{<0} + O(\alpha^2) \end{aligned}$$

Thus for $\alpha > 0$ small enough, $f(x_k + \alpha p) < f(x_k)$ implies

$$p^T \nabla f(x_k) = \|p\| \|\nabla f(x_k)\| \cos \theta < 0 \Leftrightarrow |\theta| > \pi/2,$$

where θ is the angle between p and $\nabla f(x_k)$.

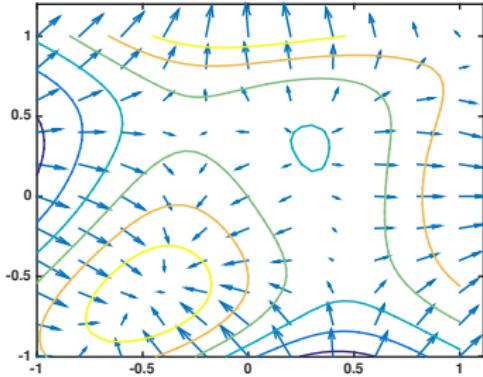
Steepest descent direction

Steepest descent direction p

$$\min_p p^T \nabla f(x_k), \quad \text{subject to } \|p\| = 1.$$

$$\min_p p^T \nabla f(x_k) = \min_p \|p\| \|\nabla f(x_k)\| \cos(\theta) = -\|\nabla f(x_k)\|,$$

attained for $\cos(\theta) = -1$ and $p = -\nabla f(x_k)/\|\nabla f(x_k)\|$, where θ is the angle between p and $\nabla f(x_k)$.



Newton direction

Consider the second order Taylor polynomial

$$f(x_k + p) \approx f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p =: m_2(p)$$

and assume $\nabla^2 f(x_k)$ is positive definite.

Newton direction minimises the second order Taylor polynomial m_2 . Setting

$$m'_2(p) = \nabla^2 f(x_k) p + \nabla f(x_k) = 0,$$

yields

$$p = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

The Newton direction is reliable when $m_2(p)$ is a close approximation to $f(x_k + p)$ i.e. $\nabla^2 f(x_k + tp)$, $t \in (0, 1)$ and $\nabla^2 f(x_k)$ are close. This is the case if $\nabla^2 f$ is sufficiently smooth and the difference is of order $\mathcal{O}(\|p\|^3)$.

$$p^T \nabla f(x_k) = -p^T \nabla^2 f(x_k) p \leq -\sigma \|p\|^2$$

for some $\sigma > 0$. Thus unless $\nabla f(x_k) = 0$ (and hence $p = 0$), $p^T \nabla f(x_k) < 0$ and p is a descend direction.

The step length 1 is optimal for $f(x_k + p) = m_2(p)$, thus 1 is used unless it does not produce a satisfactory reduction of f .

If $\nabla^2 f(x_k)$ is not positive definite, the Newton direction may not be defined: if $\nabla^2 f(x_k)$ is singular, $\nabla^2 f(x_k)^{-1}$ does not exist. Otherwise, p may not be a descent direction which can be remedied.

Fast local convergence (quadratic) close to the solution.

Computing the Hessian is expensive.

Quasi-Newton direction

Use symmetric positive definite (s.p.d.) approximation B_k to the Hessian $\nabla^2 f(x_k)$ in the Newton step

$$p = -B_k^{-1} \nabla f(x_k), \quad \nabla f(x_k)^T B_k^{-1} \nabla f(x_k) > 0,$$

such that superlinear convergence is retained.

B_k is updated in each step taking into account the additional information gathered during that step. The updates make use of the fact that changes in gradient provide information about the second derivative of f along the search direction.

Secant equation

$$\nabla f(x_k + p) = \nabla f(x_k) + B_k p$$

This equation is underdetermined, different quasi-Newton methods differ in the way they solve it.

Given the search direction p the optimal reduction of the f amounts to minimising the function of one variable

$$\phi(\alpha) := f(x_k + \alpha p), \quad \alpha > 0.$$

This is in general too expensive (even the local minimiser), hence *inexact* line search is of interest.

Choice of step size is important. Too small steps mean slow convergence, too large steps may not lead to reduction of the objective function f .

Conditions for decrease

Simple condition: require $f(x_k + \alpha p) < f(x_k)$.

Consider a sequence $f(x_k) = 5/k$, $k = 1, 2, \dots$. This sequence is decreasing but its limiting value is 0, while the minimum of a convex function can be smaller than 0.

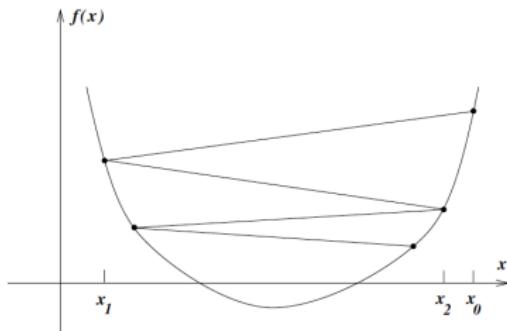


Figure: Nocedal Wright Fig 3.2

The decrease is insufficient to converge to the minimum of a convex function. Hence we need conditions for sufficient decrease.

Sufficient decrease condition

Armijo condition

$$f(x_k + \alpha p) \leq f(x_k) + c_1 \alpha p^T \nabla f(x_k) =: \ell(\alpha),$$

for some $c_1 \in (0, 1)$. [Typically small, $c_1 = 10^{-4}$]

$\ell(\alpha)$ is a linear function with negative slope $c_1 p^T \nabla f(x_k) < 0$,

$$\ell(\alpha) = f(x_k) + c_1 \alpha p^T \nabla f(x_k) > f(x_k) + \alpha p^T \nabla f(x_k) = \phi(0) + \alpha \phi'(0).$$

From Taylor Thm $\phi(\alpha) = \phi(0) + \alpha \phi'(0) + \alpha^2 \phi''(\xi)$, $\xi \in (0, \alpha)$,
thus for sufficiently small $\alpha > 0$, $\ell(\alpha) > \phi(\alpha)$.

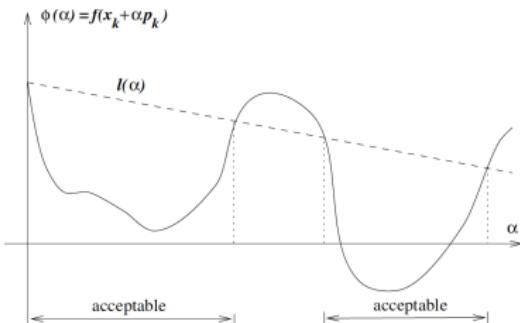


Figure: Nocedal Wright Fig 3.3

Curvature condition

As Armijo condition is satisfied for all sufficiently small α , we need another condition to avoid very small steps.

Curvature condition

$$\underbrace{p^T \nabla f(x_k + \alpha p)}_{\phi'(\alpha)} \geq c_2 \underbrace{p^T \nabla f(x_k)}_{\phi'(0)}, \quad c_2 \in (c_1, 1).$$

- Ensures, that we progress far enough along a *good* direction p .
- If $\phi'(\alpha)$ is strongly negative, there is a good prospect of significant decrease along p .
- If $\phi'(\alpha)$ is slightly negative (or even positive) we have a prospect of little decrease and hence we terminate the line search.
- Typically $c_2 = 0.9$ for a Newton or quasi Newton direction, $c_2 = 0.1$ for nonlinear conjugate gradient.

Curvature condition

Curvature condition

$$\underbrace{p^T \nabla f(x_k + \alpha p)}_{\phi'(\alpha)} \geq c_2 \underbrace{p^T \nabla f(x_k)}_{\phi'(0)}, \quad c_2 \in (c_1, 1).$$

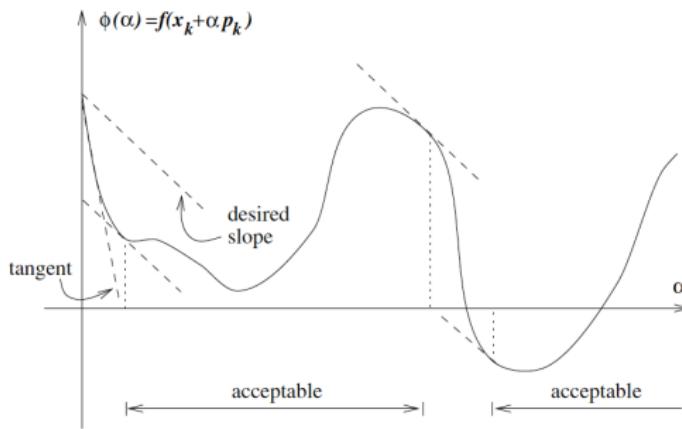


Figure: Nocedal Wright Fig 3.4

Wolfe conditions

The sufficient decrease (Armijo rule) and curvature conditions together are called **Wolfe conditions**

$$\begin{aligned} f(x_k + \alpha p) &\leq f(x_k) + c_1 \alpha p^T \nabla f(x_k), \\ p^T \nabla f(x_k + \alpha p) &\geq c_2 p^T \nabla f(x_k), \end{aligned}$$

for $0 < c_1 < c_2 < 1$.

Possibly includes points far away from stationary points, hence **strong Wolfe conditions** to disallow “too positive” values of $\phi'(\alpha)$

$$\begin{aligned} f(x_k + \alpha p) &\leq f(x_k) + c_1 \alpha p^T \nabla f(x_k), \\ |p^T \nabla f(x_k + \alpha p)| &\leq c_2 |p^T \nabla f(x_k)|, \end{aligned}$$

for $0 < c_1 < c_2 < 1$.

Wolfe conditions: existence

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuously differentiable and p be a descent direction at x_k . If f is bounded below along the ray $\{x_k + \alpha p \mid \alpha > 0\}$, then there exists an interval of step lengths satisfying both the Wolfe conditions and strong Wolfe conditions.

Proof:

For $\alpha > 0$, $\phi(\alpha) = f(x_k + \alpha p)$ is bounded below while $\ell(\alpha) = f(x_k) + \alpha c_1 p^T \nabla f(x_k)$ is unbounded below as $c_1 p^T \nabla f(x_k) < 0$ and for small α , $\ell(\alpha) > \phi(\alpha)$ as $c_1 < 1$. Thus $\ell(\alpha)$ has to intersect $\phi(\alpha)$ at least once. Let α' be the smallest value for which

$$\phi(\alpha') = f(x_k + \alpha' p) = f(x_k) + \alpha' c_1 p^T \nabla f(x_k) = \ell(\alpha').$$

Then the sufficient decrease condition holds for all $\alpha \leq \underline{\alpha'}$.

Furthermore, by the mean value theorem

$$\exists \alpha'' \in (0, \alpha') : f(x_k + \alpha' p) - f(x_k) = \alpha' p^T \nabla f(x_k + \alpha'' p)$$

and we obtain

$$p^T \nabla f(x_k + \alpha'' p) = c_1 p^T \nabla f(x_k) > c_2 p^T \nabla f(x_k)$$

since $c_1 < c_2$ and $p^T \nabla f(x_k) < 0$ and therewith α'' satisfies Wolfe conditions. As the inequality in curvature condition for α'' holds strictly, by continuity of ∇f , the inequality (and hence Wolfe conditions) also holds in an interval containing α'' . Furthermore, as all terms in the last equation are negative strong Wolfe conditions hold for the same interval.

Wolfe conditions are scale-invariant in the sense that are unaffected by scaling the function or affine change of variables. They can be used in most line search methods and are particularly important for quasi-Newton methods.

Goldstein conditions

$$f(x_k) + (1 - c)\alpha p^T \nabla f(x_k) \leq f(x_k + \alpha p) \leq f(x_k) + c\alpha p^T \nabla f(x_k)$$

with $0 < c < 1/2$.

The second inequality is the sufficient decrease condition. The first inequality controls the step length from below. Disadvantage w.r.t. Wolfe conditions is that it can exclude all minimisers of ϕ .

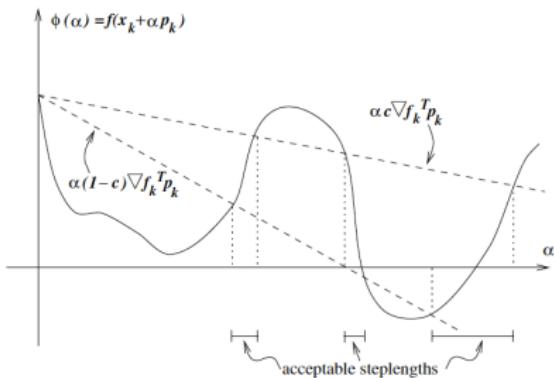


Figure: Nocedal Wright Fig 3.6

Backtracking line search

- 1: Choose $\bar{\alpha} > 0, \rho \in (0, 1), c \in (0, 1)$
- 2: Set $\alpha = \bar{\alpha}$
- 3: **repeat**
- 4: $\alpha = \rho\alpha$
- 5: **until** $f(x_k + \alpha p) \leq f(x_k) + c\alpha p^T \nabla f(x_k)$

- Terminates in finite number of steps: α will eventually become small enough to satisfy sufficient decrease condition.
- Prevents too short step lengths: the accepted α is within factor ρ of the previous value, α/ρ , which was rejected for violating the sufficient decrease condition i.e. being too long.
- ρ can vary in $[\rho_{\min}, \rho_{\max}] \subset (0, 1)$ between iterations.
- In Newton and quasi-Newton methods $\bar{\alpha} = 1$, but different values can be appropriate for other algorithms.
- Well suited for Newton methods, less appropriate for quasi-Newton and conjugate gradient methods.

Convergence of line search methods [Zoutendijk]

Consider an iteration

$$x_{k+1} = x_k + \alpha_k p_k, \quad k = 0, 1, \dots,$$

where p_k is a descent direction and α_k satisfies the Wolfe conditions.

Let f be bounded below in \mathbb{R}^n and continuously differentiable in an open set \mathcal{M} containing the level set $\{x : f(x) \leq f(x_0)\}$.

If ∇f is Lipschitz continuous on \mathcal{M} i.e.

$$\exists L > 0 : \|\nabla f(x) - \nabla f(\bar{x})\| \leq L\|x - \bar{x}\|, \quad \forall x, \bar{x} \in \mathcal{M}$$

then

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty,$$

where $\theta_k = \angle(p_k, -\nabla f(x_k))$.

Convergence of line search methods [Zoutendijk]

Subtracting $p_k^T \nabla f(x_k)$ from both sides of curvature condition

$$p_k^T \nabla f(\underbrace{x_k + \alpha p_k}_{=x_{k+1}}) \geq c_2 p_k^T \nabla f(x_k)$$

we obtain

$$p_k^T (\nabla f(x_{k+1}) - \nabla f(x_k)) \geq (c_2 - 1) p_k^T \nabla f(x_k).$$

On the other hand the Lipschitz condition implies

$$p_k^T (\nabla f(x_{k+1}) - \nabla f(x_k)) \leq \|\nabla f(x_{k+1}) - \nabla f(x_k)\| \|p_k\| \leq \alpha_k L \|p_k\|^2.$$

Combining the last two inequalities we obtain a lower bound on the step size

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{p_k^T \nabla f(x_k)}{\|p_k\|^2}.$$

Substituting this inequality into the sufficient decrease condition

$$f(x_{k+1}) \leq f(x_k) + c_1 \frac{c_2 - 1}{L} \frac{(p_k^T \nabla f(x_k))^2}{\|p_k\|^2}$$

with $\cos \theta_k = -\frac{p_k^T \nabla f(x_k)}{\|\nabla f(x_k)\| \|p_k\|}$ yields

$$f(x_{k+1}) \leq f(x_k) - c \cos^2 \theta_k \|\nabla f(x_k)\|^2,$$

where $c = c_1(1 - c_2)/L$.

Summing over all indices up to k we obtain

$$f(x_{k+1}) \leq f(x_0) - c \sum_{j=0}^k \cos^2 \theta_j \|\nabla f(x_j)\|^2$$

and since f is bounded from below

$$\sum_{j=0}^k \cos^2 \theta_j \|\nabla f(x_j)\|^2 \leq (f(x_0) - f(x_{k+1}))/c < C$$

where $C > 0$ is some positive constant. Taking limits
 $\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty$.

Global convergence

Goldstein or strong Wolfe conditions also imply the Zoutendijk condition

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty.$$

The Zoutendijk condition implies

$$\cos^2 \theta_k \|\nabla f(x_k)\|^2 \rightarrow 0,$$

which can be used to derive *global* convergence results for line search algorithms.

If the method ensures that $\cos \theta_k \geq \delta > 0$, $\forall k$ i.e. θ_k is bounded away from $\pi/2$, it follows that

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

This is the strongest global convergence result that can be obtained for such iteration (convergence to a stationary point) without additional assumptions.

In particular, the steepest descent ($p_k = -\nabla f(x_k)$) produces a gradient sequence which converges to 0 if it uses a line search satisfying Wolfe or Goldstein conditions.

For some algorithms e.g. nonlinear conjugate gradient methods, only a weaker result can be obtained

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$$

i.e. only subsequence of gradient norms $\|\nabla f(x_{k_j})\|$ converges to 0 rather than the whole sequence.

Those limits can be proved by contradiction:

Suppose that $\|\nabla f(x_k)\| \geq \gamma$ for some $\gamma > 0$ for all k sufficiently large. Then from $\cos^2 \theta_k \|\nabla f(x_k)\|^2 \rightarrow 0$ we conclude that $\cos \theta_k \rightarrow 0$ i.e. the entire sequence $\{\cos \theta_k\}$ converges to 0.

Thus to show the weak convergence result it is enough to show that a subsequence $\{\cos \theta_{k_j}\}$ is bounded away from 0.

Consider any algorithm which

- (i) decreases the objective function in each iteration,
- (ii) every m th iteration is a steepest descent step with step length satisfying the Wolfe or Goldstein conditions.

Since $\cos \theta_k = 1$ for steepest descent steps, this provides the subsequence bounded away from 0. The algorithm can do something better in remaining $m - 1$ iterates, while the occasional steepest descent step will guarantee the overall (weak) global convergence.

Rate of convergence

Unfortunately, rapid convergence sometimes conflicts with global convergence.

Example: Steepest descent is globally convergent (with appropriate step sizes) but it can be very slow in practice. On the other hand, while Newton iteration converges rapidly when we are close to the solution, the Newton step may not even be a descent direction far away from the solution.

The challenge: design algorithms with good global convergence properties and rapid convergence rate.

Steepest descent

Steepest descent with exact line search for strictly convex quadratic function

$$f(x) = \frac{1}{2}x^T Qx - b^T x,$$

where Q is symmetric positive definite.

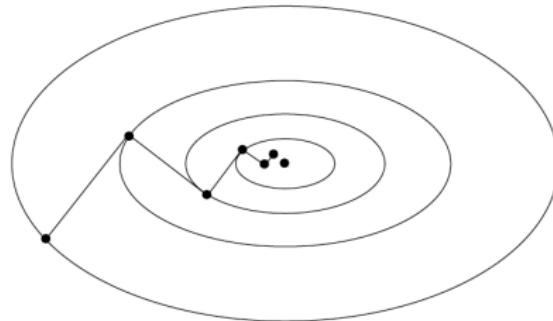


Figure: Nocedal Wright Fig 3.7

Steepest descent

Characteristic zig-zag due to elongated shape of the ellipse. If the level sets were circles instead, the steepest descent would need one step only.

Convergence rate of steepest descent with exact line search:

$$\underbrace{\|x_{k+1} - x^*\|_Q^2}_{=f(x_{k+1})-f(x^*)} \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \underbrace{\|x_k - x^*\|_Q^2}_{=f(x_k)-f(x^*)},$$

where $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ are the eigenvalues of Q , and $\|x\|_Q^2 = x^T Q x$. Note: for quadratic strictly convex function we obtain objective function and (for free) iterate convergence rates!

The objective function convergence rate is essentially the same for steepest descent with exact line search when applied to a twice continuously differentiable nonlinear function satisfying sufficient conditions at x^* .

Local quadratic convergence: Newton methods

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable with Lipschitz continuous Hessian in a neighbourhood of the solution x^* satisfying the sufficient conditions. Note that the Hessian $\nabla^2 f$ is positive definite also in the vicinity of the solution x^* .

The iterates x_k computed by the Newton method ([note step length 1](#))

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

converge locally quadratically i.e. for starting point x_0 sufficiently close to x^* .

The sequence of gradient norms $\|\nabla f(x_k)\|$ also converges quadratically to 0.

Local convergence: note that away from the solution ∇f_k may not be positive definite and hence p_k may not be a descent direction.
Global convergence with Hessian modification is discussed later.

Superlinear convergence: Newton-type methods with line search with W.C.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable.

Let $\{x_k\}$ be a sequence generated by a descent method

$$x_{k+1} = x_k + \alpha_k p_k$$

for step sizes satisfying Wolfe conditions with $c_1 \leq 1/2$.

If the sequence $\{x_k\}$ converges to a point x^* satisfying the sufficient conditions and the search direction satisfies

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k)p_k\|}{\|p_k\|} = 0,$$

then for all $k > k_0$, the step length $\alpha_k = 1$ is admissible, and for that choice of $\alpha_k = 1, k > k_0$, the sequence $\{x_k\}$ converges to x^* superlinearly.

Note: once close enough to the solution so that $\nabla^2 f(x_k)$ became s.p.d., the limit is trivially satisfied and for $\alpha_k = 1$ we recover local quadratic convergence.

Superlinear convergence: Quasi-Newton methods

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable. Let $\{x_k\}$ be a sequence generated by a quasi-Newton method ([note step length 1](#), B_k s.p.d.)

$$x_{k+1} = x_k - \underbrace{B_k^{-1} \nabla f(x_k)}_{p_k}.$$

Assume the sequence $\{x_k\}$ converges to a point x^* satisfying the sufficient conditions. Then $\{x_k\}$ converges superlinearly [if and only if](#)

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|} = 0$$

(Quasi-Newton version of the equivalent condition in the previous theorem).

Note: the superlinear convergence rate can be attained even if the sequence $\{B_k\}$ does not converge to $\nabla^2 f(x^*)$. It suffices that B_k becomes increasingly accurate approximation to $\nabla^2 f(x^*)$ along the search direction p_k . Quasi-Newton methods use it to construct B_k .

Hessian modifications

Away from the solution, the Hessian may not be positive definite, and the Newton direction may not be a descent direction. The general solution is to consider positive definite approximations.

$B_k = \nabla^2 f(x_k) + E_k$, where E_k is chosen to ensure that B_k is sufficiently positive definite.

Global convergence results can be established for Newton method with Hessian modification and step satisfying Wolfe or Goldstein or Armijo backtracking conditions provided that:

$\kappa(B_k) = \|B_k\| \|B_k^{-1}\| \leq C$ for some $C > 0$ and all k whenever the sequence of the Hessians $\{\nabla^2 f(x_k)\}$ is bounded.

Hessian modifications

Eigenvalue decomposition

$$\nabla^2 f(x_k) = Q \Lambda Q^T = \sum_{i=1}^n \lambda_i q_i q_i^T.$$

Example:

$$\nabla^2 f(x_k) = \text{diag}(10, 3, -1), \quad \nabla f(x_k) = (1, -3, 2)^T$$

$$Q = I, \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$$

The Newton step: $p_k = (-0.1, 1, 2)^T$

As $p_k^T \nabla f(x_k) > 0$, it is not a descent direction.

Eigenvalue modifications (not practical):

Replace all negative eigenvalues with $\delta = \sqrt{\mathbf{u}} = 10^{-8}$, where $\mathbf{u} = 10^{-16}$ is the machine precision.

$$B_k = \sum_{i=1}^2 \lambda_i q_i q_i^T + \delta q_3 q_3^T = \text{diag}(10, 3, 10^{-8})$$

B_k is s.p.d. and curvature along q_1 , q_2 is preserved, however the direction is dominated by q_3 :

$$p_k = -B_k^{-1} \nabla f_k = -\sum_{i=1}^2 \frac{1}{\lambda_i} q_i q_i^T \nabla f_k - \frac{1}{\delta} q_3 q_3^T \nabla f_k \approx -(2 \times 10^8) q_3.$$

p_k is a descent direction but the length is very large, not in line with local validity of the Newton approximation. Thus p_k may be ineffective.

Adapt choice of δ to avoid excessive lengths. Even $\delta = 0$ which eliminates direction q_3 .

Let A is symmetric $A = Q\Lambda Q^T$.

The correction matrix ΔA of minimum Frobenius norm that ensures $\lambda_{\min}(A + \Delta A) \geq \delta$ is given by

$$\Delta A = Q \operatorname{diag}(\tau_i) Q^T, \quad \text{with} \quad \tau_i = \begin{cases} 0, & \lambda_i \geq \delta \\ \delta - \lambda_i, & \lambda_i < \delta. \end{cases}$$

and the modified matrix is

$$A + \Delta A = Q(\Lambda + \operatorname{diag}(\tau_i))Q^T.$$

Frobenius norm is defined $\|A\|_F^2 = \sum_{i,j=1}^n a_{ij}^2 = \sum_{i=1}^n \lambda_i^2$.

The correction matrix ΔA of minimum Euclidean norm that satisfies $\lambda_{\min}(A + \Delta A) \geq \delta$ is given by

$$\Delta A = \tau I, \quad \text{with} \quad \tau = \max(0, \delta - \lambda_{\min}(A)).$$

and the modified matrix has the form $A + \tau I$.

Cholesky factorisation of $A + \tau I$

Simple idea:

If $\min_i a_{ii} \leq 0$ set $\tau_0 = -\min_i a_{ii} + \beta$ for some small $\beta > 0$ (e.g. 10^{-3}), otherwise $\tau_0 = 0$.

Attempt the Cholesky algorithm to obtain $MM^T = A + \tau_k I$

If not successful, increase $\tau_{k+1} = \max(2\tau_k, \beta)$ and reattempt.

Drawback: possibly multiple failed attempts to factorise.

Cholesky decomposition

Consider the case $n = 3$. The equation $A = LDL^T$ is given by

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{bmatrix}.$$

(The notation indicates that A is symmetric.) By equating the elements of the first column, we have

$$\begin{aligned} a_{11} &= d_1, \\ a_{21} &= d_1 l_{21} \quad \Rightarrow \quad l_{21} = a_{21}/d_1, \\ a_{31} &= d_1 l_{31} \quad \Rightarrow \quad l_{31} = a_{31}/d_1. \end{aligned}$$

Proceeding with the next two columns, we obtain

$$\begin{aligned} a_{22} &= d_1 l_{21}^2 + d_2 \quad \Rightarrow \quad d_2 = a_{22} - d_1 l_{21}^2, \\ a_{32} &= d_1 l_{31} l_{21} + d_2 l_{32} \quad \Rightarrow \quad l_{32} = (a_{32} - d_1 l_{31} l_{21}) / d_2, \\ a_{33} &= d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \quad \Rightarrow \quad d_3 = a_{33} - d_1 l_{31}^2 - d_2 l_{32}^2. \end{aligned}$$

Figure: Nocedal Wright Ex. 3.1

Cholesky decomposition of indefinite matrix

For A indefinite:

- The factorisation $A = LDL^T$ may not exist.
- Even if it does exist, the algorithm can be unstable i.e. elements of L and D can become arbitrarily large.
- Posterior modification of D to force the elements to be positive may break down or result in a matrix very different to A .
- Instead, modify A during the factorisation to achieve that the elements of D are sufficiently positive and the elements of L and D are not too large.

Modified Cholesky decomposition

Choose $\delta, \beta > 0$. While computing j th column of L, D ensure

$$d_j \geq \delta, \quad |m_{ij}| \leq \beta, \quad i = j+1, j+2, \dots, n,$$

where $m_{ij} = l_{ij}\sqrt{d_j}$.

To satisfy these bounds we only need to change how d_j is computed, from $d_j = c_{jj}$ to

$$d_j = \max \left(|c_{jj}|, \frac{\theta_j^2}{\beta^2}, \delta \right), \quad \text{with } \theta_j = \max_{j < i \leq n} |c_{ij}|,$$

where $c_{ij} = l_{ij}d_j$. Note: θ_j can be computed before d_j because computing $c_{ij}, j < i \leq n$ only needs previous columns!

Verification:

$d_j \geq \delta$ due to taking maximum

$$|m_{ij}| = |l_{ij}\sqrt{d_j}| = \frac{|c_{ij}|}{\sqrt{d_j}} \leq \frac{|c_{ij}|\beta}{\theta_j} \leq \beta, \quad \forall i > j.$$

Modified Cholesky decomposition

Properties:

- Modifies the Hessian during factorization where necessary.
- The modified Cholesky factors exist and are bounded relative to the norm of the actual Hessian.
- It does not modify Hessian if it is sufficiently positive definite.

This is the basis for the modified Cholesky factorisation which also introduces symmetric row and column permutations to reduce the size of the modification.

$$PAP^T + E = LDL^T = MM^T,$$

where E is a nonnegative diagonal matrix that is zero if A is sufficiently positive definite.

It has been shown, that the matrices obtained by this modified Cholesky algorithm to the exact Hessian $\nabla^2 f(x_k)$ have bounded condition numbers, hence some global convergence results can be obtained.

Step length selection

How to find a step length satisfying one of the termination conditions e.g. Wolfe etc. for

$$\phi(\alpha) = f(x_k + \alpha p_k),$$

where p_k is a descent direction i.e. $\phi'(0) = p_k^T f'(x_k) < 0$.

If f is a convex quadratic function $f(x) = \frac{1}{2}x^T Qx - b^T x$, it has a global minimiser along the ray $x_k + \alpha p_k$ which can be calculated analytically

$$\alpha_k = -\frac{p_k^T \nabla f(x_k)}{p_k^T Q p_k}.$$

For general nonlinear functions iterative approach is necessary.

Line search algorithms can be classified according to the information they use:

Methods using only function evaluations can be very inefficient as they need to continue iterating until a very small interval has been found.

Methods using gradient information can determine whether the current step length satisfies e.g. Wolfe or Goldstein conditions which require gradients to evaluate.

Typically, they consist of two phases: *bracketing phase* which finds an interval containing acceptable step lengths and the *selection phase* which locates the final step in the interval.

Examples: via interpolation, line search method in Nocedal Wright (see tutorial).

Initial step length

For Newton and quasi Newton $\alpha_0 = 1$. This ensures that unit step length will be taken whenever they satisfy the termination conditions and allow for quick convergence.

For methods which do not produce well scaled search direction like steepest descent or conjugate gradient it is important to use available information to make the initial guess e.g.:

- First order change in function at iterate x_k will be the same as that obtained at previous step

i.e. $\alpha_0 p_k^T \nabla f(x_k) = \alpha_{k-1} p_{k-1}^T \nabla f(x_{k-1})$

$$\alpha_0 = \alpha_{k-1} \frac{p_{k-1}^T \nabla f(x_{k-1})}{p_k^T \nabla f(x_k)}.$$

- Interpolate quadratic to data $f(x_{k-1}), f(x_k)$ and $p_{k-1}^T \nabla f(x_{k-1})$ and define α_0 to be its minimiser

$$\alpha_0 = \frac{2(f(x_k) - f(x_{k-1}))}{\phi'(0)}.$$

It can be shown that if $x_k \rightarrow x^*$ superlinearly, then the ratio converges to 1. If we adjust by setting $\alpha_0 = \min(1, 1.01\alpha_0)$ we find that the unit step length will eventually always be tried and accepted and the superlinear convergence will be observed.

Numerical Optimisation: Trust region methods

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 4

Trust region: idea

- Choose a region around the current iterate $f(x_k)$ in which we trust a model.
- Choose a relatively easy solvable model which we trust is an adequate representation of f in this region.
- Compute the direction and step length which minimise the model in the trust region.
- The size of the trust region is critical to effectiveness. If the region is too small, the algorithm will make little progress. If the region is too large, the minimiser of the model can be far away from the minimiser of f .
- If the model is consistently reliable, the trust region may be increased.
- If the step length is not acceptable, reduce the size of the trust region and find a new minimiser. In general both the direction and step length change when the trust region changes.

Trustregion: model

Here we assume a quadratic model based on Taylor expansion of f at x_k

$$m_k(p) = f(x_k) + g_k^T p + \frac{1}{2} p^T B_k p,$$

where $g = \nabla f(x_k)$, and B_k is a symmetric approximation to the Hessian $\nabla^2 f(x_k)$.

In general we only assume symmetry and uniform boundedness for B_k . The difference between $\nabla^2 f(x_k + tp)$, $t \in (0, 1)$ and $m_k(p)$ is $\mathcal{O}(\|p\|^2)$.

The choice of $B_k = \nabla^2 f(x_k)$ leads to **trust region Newton methods** and the model accuracy is $\mathcal{O}(\|p\|^3)$.

In each step we solve

$$\min_{p \in \mathbb{R}^n} m_k(p) = f(x_k) + g_k^T p + \frac{1}{2} p^T B_k p, \quad \text{s.t. } \|p\| \leq \Delta_k, \quad (\text{CM})$$

and $\Delta_k > 0$ is the radius of the trust region. The constraint can be equivalently written $p^T p \leq \Delta_k^2$.

If B_k is positive definite the minimum of the unconstrained quadratic model problem m_k is $p^B = -B_k^{-1}g_k$. If $\|p^B\| = \|B_k^{-1}g_k\| \leq \Delta_k$ this is also the solution to the constrained problem and we call p^B a **full step**.

Solution in other cases is less straight forward but can usually be obtained at moderate computational cost. In particular, only **approximate** solution is necessary to obtain convergence and good practical behaviour.

Trust region vs line search

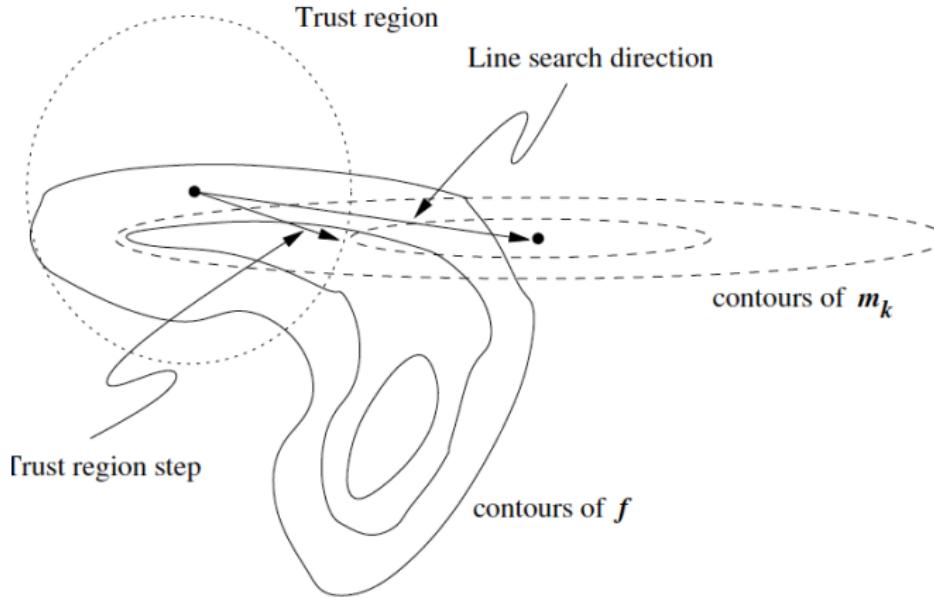


Figure: Nocedal Wright Fig 4.1

Choice of trust region radius Δ_k

Compare the actual reduction in objective function to the predicted reduction i.e. reduction in the model m_k .

$$\rho_k = \frac{f(x_k) - f(x_k + p)}{m_k(0) - m_k(p)}$$

- $\rho_k < 0$: $f(x_k) < f(x_k + p)$ – reject step, shrink trust region and try again
- $\rho_k > 0$, small – accept step, shrink trust region for next iteration
- $\rho_k > 0$, but significantly smaller than 1 – accept the step and do not alter trust region
- $\rho \approx 1$: good agreement between f and m_k – accept step and expand trust region for next iteration.

Algorithm: Trust region

```
1: Given  $\hat{\Delta} > 0$ ,  $\Delta_0 \in (0, \hat{\Delta})$  and  $\eta \in [0, \frac{1}{4})$ 
2: for  $k = 1, 2, 3, \dots$  do
3:   Obtain  $p_k$  by (approximatively) solving (CM)
4:   Evaluate  $\rho_k = \frac{f(x_k) - f(x_k + p)}{m_k(0) - m_k(p)}$ 
5:   if  $\rho_k < \frac{1}{4}$  then
6:      $\Delta_{k+1} = \frac{1}{4}\Delta_k$ 
7:   else
8:     if  $\rho_k > \frac{3}{4}$  and  $\|p_k\| = \Delta_k$  then
9:        $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$ 
10:    else
11:       $\Delta_{k+1} = \Delta_k$ 
12:    end if
13:  end if
14:  if  $\rho_k > \eta$  then
15:     $x_{k+1} = x_k + p_k$ 
16:  else
17:     $x_{k+1} = x_k$ 
18:  end if
19: end for
```

Theorem [More, Sorensen]

p^* is a global solution of the trust region problem (CM)

$$\min_{p \in \mathbb{R}^n} m_k(p) = f(x_k) + g_k^T p + \frac{1}{2} p^T B_k p, \quad \text{s.t. } \|p\| \leq \Delta_k$$

if and only if p^* is feasible and there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:

$$(B + \lambda I)p^* = -g_k, \tag{1a}$$

$$\lambda(\Delta - \|p^*\|) = 0, \tag{1b}$$

$$B + \lambda I \text{ is positive semidefinite.} \tag{1c}$$

Solution of (CM) for different radii

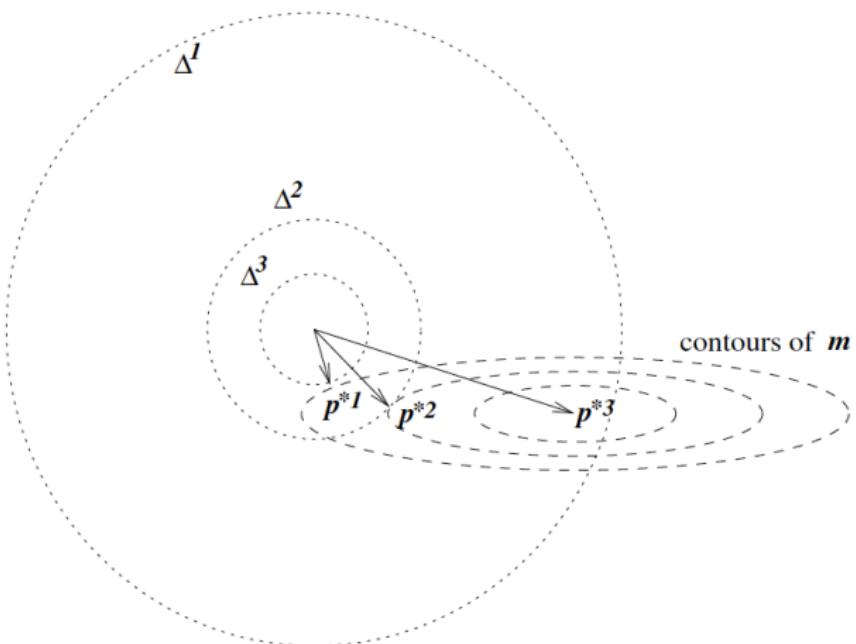


Figure: Nocedal Wright Fig 4.2 (note that p_3^* and p_1^* should be swapped)

Solution of (CM) for different radii

For Δ_1 , $\|p^*\| < \Delta$ hence $\lambda = 0$ and so

$$Bp^* = -g_k$$

with B positive semidefinite from (1)(a,c).

For Δ_2, Δ_3 the solution lies on the boundary of the respective trust region, hence $\|p^*\| = \Delta$ and $\lambda \geq 0$. From (1)(a) we have

$$\lambda p^* = -Bp^* - g_k = -\nabla m_k(p^*).$$

Thus if $\lambda > 0$, p^* is collinear with the negative gradient of m_k and normal to its contours.

Cauchy point

Cauchy point p^C is the minimiser of m_k along the steepest descent direction $-g_k$ subject to the trust region bound.

Find p^s :

$$p^s = \arg \min_{p \in \mathbb{R}^n} f(x_k) + g_k^T p, \quad \text{s.t. } \|p\| \leq \Delta_k$$

Calculate the scalar $\tau_k > 0$:

$$\tau_k = \arg \min_{\tau \geq 0} m_k(\tau p^s) \quad \text{s.t. } \|\tau p^s\| \leq \Delta_k.$$

Set $p^C = \tau_k p^s$.

The solution to the first problem can be written down explicitly, simply by going as far as allowed in the steepest descent direction

$$p^s = -\frac{\Delta_k}{\|g\|} g.$$

To obtain τ_k we substitute $p^s = -\frac{\Delta_k}{\|g_k\|}g_k$ into the second problem we obtain

$$\arg \min_{\tau} m_k(\tau p^s) = f(x_k) - \tau \underbrace{\frac{\Delta_k}{\|g_k\|} g_k^T g_k}_{\geq 0} + \frac{1}{2} \tau^2 \frac{\Delta_k^2}{\|g_k\|^2} g_k^T B_k g_k$$

subject to $\|\tau g_k \frac{\Delta_k}{\|g_k\|}\| \leq \Delta_k \Leftrightarrow \tau \in [-1, 1]$.

$m_k(\tau p^s)$ is a parabola with vertex at $\tau_v = \|g_k\|^3 / (\Delta_k g_k^T B_k g_k)$. We consider two cases:

- $g_k^T B_k g_k \leq 0$: $m_k(\tau p^s)$ concave function with $\tau_v < 0$. $m_k(\tau p^s)$ decreases monotonically in $(\tau_v, 1] \supset [0, 1]$ whenever $g_k \neq 0$. Hence, the minimum is attained for largest $\tau \in [-1, 1]$ i.e. $\tau = 1$.
- $g_k^T B_k g_k > 0$: is strictly convex quadratic function in $\tau_v > 0$, thus the minimum is either the unconstraint minimiser whenever in $(0, 1] \subset [-1, 1]$ or otherwise 1 ($\arg \min_{\tau=\{-1,1\}} m_k(\tau p^s)$)

$$\tau_k = \begin{cases} 1 & g_k^T B_k g_k \leq 0 \\ \min \left\{ \|g_k\|^3 / (\Delta_k g_k^T B_k g_k), 1 \right\} & g_k^T B_k g_k > 0. \end{cases}$$

Significance of Cauchy point

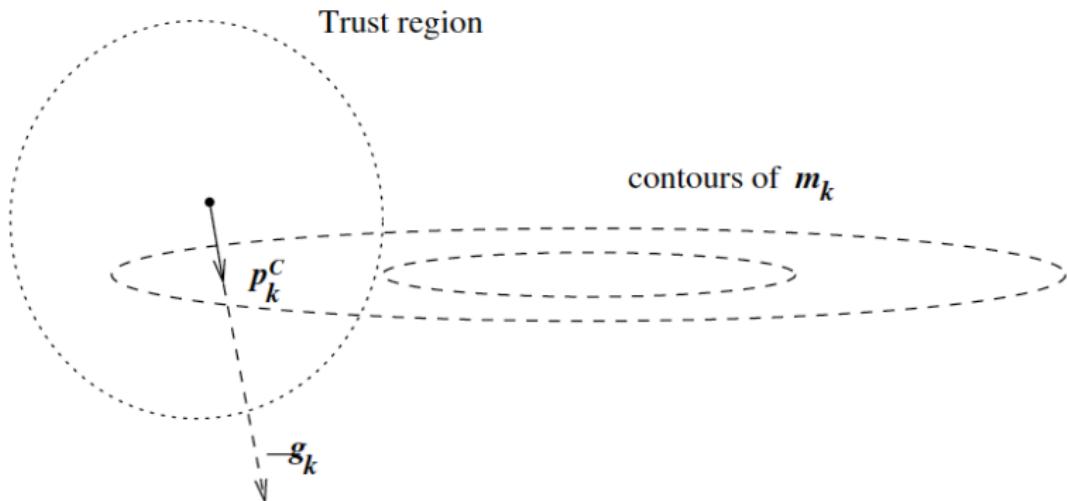


Figure: Nocedal Wright Fig 4.3, Example: Cauchy point for positive definite B_k

Sufficient reduction in the model is a reduction of at least a positive fraction of that achieved by the Cauchy point p^C .

Improvement on Cauchy point

- Cauchy point p^C provides sufficient reduction to yield global convergence.
- Cauchy point is cheap to compute.
- Cauchy point essentially corresponds to the steepest descent method with a particular choice of step length. Steepest descent performance can be very poor even with optimal step length.
- For Cauchy point the information in B is only used to compute the step length. Superlinear convergence can only be expected when B is used to compute both the descend direction and the step length.
- A number of trust region methods compute the Cauchy point and then attempt to improve on it. Often, the full step i.e. $p^B = -B^{-1}g_k$ is chosen whenever B is positive definite and $\|p^B\| \leq \Delta_k$. When $B = \nabla^2 f(x_k)$ or a quasi-Newton approximation, this strategy can be expected to yield superlinear convergence.

The dogleg method

Assumption: B positive definite.

If $p^B = -B^{-1}g_k$ with $\|p^B\| \leq \Delta_k$ it is just the unconstrained minimum

$$p^* = p^B, \quad \|p^B\| \leq \Delta_k$$

On the other hand if Δ is small w.r.t. $\|p^B\|$ the restriction to $\|p^B\| \leq \Delta$ ensures that the quadratic term in m_k has little effect on the solution of (CM) and it could be omitted i.e.

$$p^* \approx -\frac{\Delta_k}{\|g_k\|}g_k, \quad \Delta_k \ll \|p^B\|.$$

For intermediate values of Δ_k , the solution $p^*(\Delta_k)$ typically follows a curved trajectory (Fig. 4.4 Nocedal, Wright).

Dogleg approximation to $p(\Delta_k)$

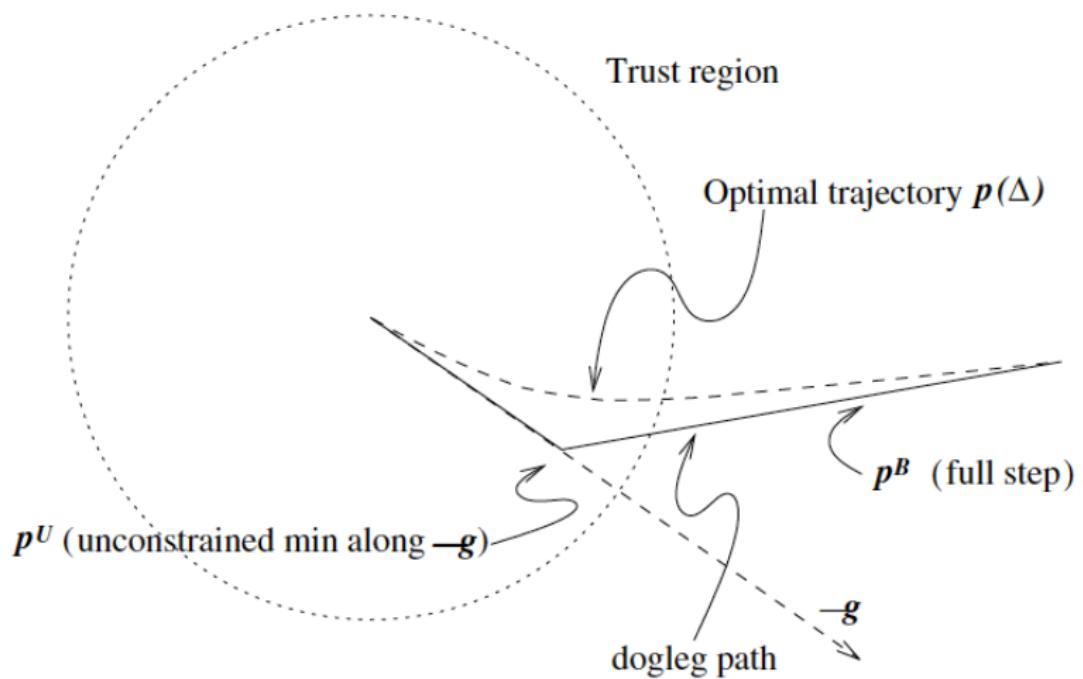


Figure: Nocedal Wright Fig 4.4

The dogleg method replaces the curved trajectory with path consisting of two line segments.

The first line segment runs from the origin to the minimiser of m_k along the steepest descent direction

$$p^U = -\frac{g_k^T g_k}{g_k^T B g_k} g_k.$$

The second line segment runs from p^U to p^B (the unconstraint minimum or full step).

Formally, the trajectory can be written as

$$\tilde{p}(\tau) = \begin{cases} \tau p^U, & 0 \leq \tau \leq 1 \\ p^U + (\tau - 1)(p^B - p^U), & 1 \leq \tau \leq 2. \end{cases}$$

The dogleg method chooses p to minimise the model m_k along this path subject to the trust region bound.

The minimum along the dogleg can be found easily because

- (i) $\|\tilde{p}(\tau)\|$ is an increasing function of τ
- (ii) $m(\tilde{p}(\tau))$ is a decreasing function of τ

Proof: For $\tau \in [0, 1]$ it follows from definition of p^U . For $\tau \in [1, 2]$ can be shown computing the derivative and showing that it is nonnegative (i), nonpositive (ii).

Intuition:

- (i) The length of \tilde{p} could only decrease with τ if $\tilde{p}(\tau)$ turns back at $\tau = 1$ i.e. the vector $p^B - p^U$ makes an angle larger than $\pi/2$ with p^U which is not possible for the steepest descent solution.
- (ii) $m(\tilde{p}(2))$ is the minimum of a strictly convex function, hence $m(\tilde{p}(1)) > m(\tilde{p}(2))$ and the function decreases for $\tau \in [1, 2]$.

As a consequence the path $\tilde{p}(\tau)$ intersects the trust region boundary at exactly one point if $\|p^B\| \geq \Delta$ and the intersection point can be computed solving the quadratic equation

$$\|p^U + (\tau - 1)(p^B - p^U)\|^2 = \Delta^2.$$

In case the exact Hessian $\nabla^2 f(x_k)$ is available, if it is positive definite, we set $B = \nabla^2 f(x_k)$ and the resulting procedure is a **Newton dogleg method**. If $\nabla^2 f(x_k)$ is not positive definite, we could use one of the modified Hessians and close to the solution we will recover the Newton step. However, the somewhat arbitrary perturbation introduced by the modification can interfere with the benefits of the trust region methods. In fact, the trust region introduces its own modification (1)(a,c) thus the dogleg method is most appropriate when B is positive semidefinite.

2D subspace minimisation

An extension of the dogleg method

$$\begin{aligned} \min_p m_k(p) &= f(x_k) + g_k^T p + \frac{1}{2} p^T B p \\ \text{s.t. } &\|p\| \leq \Delta_k, \quad p \in \text{span}[g, B^{-1}g]. \end{aligned}$$

The obtained minimiser is an improvement on the dogleg solution as $\tilde{p} \in \text{span}[g, B^{-1}g]$. Furthermore, the reduction in the model m_k is often close to that achieved solving the full problem (CM) (not on the subspace). The subspace $\text{span}[g, B^{-1}g]$ is a good one for looking for a minimiser for a quadratic model (Taylor theorem).

The 2D subspace minimisation method can be modified for indefinite B .

For more details see the corresponding tutorial.

Cauchy point reduction of the model m_k

The Cauchy point p^C satisfies the sufficient reduction condition

$$m_k(0) - m_k(p) \geq c_1 \|g_k\| \min \left(\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right) \quad (\text{SR})$$

with $c_1 = \frac{1}{2}$.

Proof: Use the definition of the Cauchy point p^C and check the inequality case by case.

If a vector p with $\|p\| \leq \Delta_k$ satisfies

$$m_k(0) - m_k(p) \geq c_2(m_k(0) - m_k(p^C))$$

then it satisfies (SR) with $c_1 = c_2/2$

$$m_k(0) - m_k(p) \geq c_2(m_k(0) - m_k(p^C)) \geq \frac{1}{2}c_2 \|g_k\| \min \left(\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right).$$

In particular, if p is the exact solution p^* of (CM), then it satisfies (SR) with $c_1 = \frac{1}{2}$. Note that both the dogleg and 2d-subspace minimisation algorithms satisfy (SR) with $c_1 = \frac{1}{2}$ because the both produce approximate solutions p for which $m_k(p) \leq m_k(p^C)$.

Let $\|B_k\| \leq \beta$ for some constant $\beta > 0$ and f be bounded below on the level set $S = \{x : f(x) \leq f(x_0)\}$ and Lipschitz continuously differentiable in the neighbourhood of S , $\mathcal{N}(S, R_0)$, $R_0 > 0$ and all the approximate solutions p_k of (CM) satisfy the inequalities (SR) for some $c_1 > 0$ and $\|p_k\| \leq \gamma \Delta_k$, $\gamma \geq 1$ (slight relaxation of trust region). We then have for

- $\eta = 0$ in Algorithm:Trust region

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

- $\eta \in (0, \frac{1}{4})$ in Algorithm:Trust region

$$\lim_{k \rightarrow \infty} g_k = 0.$$

Superlinear local convergence

Let f be twice Lipschitz continuously differentiable in the neighbourhood of a point x^* at which the second order sufficient conditions are satisfied. Suppose that the sequence $\{x_k\}$ converges to x^* and that for all k sufficiently large, the trust region algorithm based on (CM) with $B_k = \nabla^2 f(x_k)$ chooses steps p_k that satisfy the Cauchy point based sufficient reduction criteria (SR) and are asymptotically similar to Newton steps p_k^N whenever $\|p_k^N\| \leq \frac{1}{2}\Delta_k$ i.e.

$$\|p_k - p_k^N\| = o(\|p_k^N\|).$$

Then the trust region bound Δ_k becomes inactive for all k sufficiently large and the sequence $\{x_k\}$ converges superlinearly to x^* .

Numerical Optimisation: Conjugate gradient methods

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 5 & 6

Conjugate gradient: CG

- The linear CG method was proposed by Hestens and Stiefel in 1952 as a [direct](#) method for solution of linear systems of equations with positive definite matrix. (It was used to solve 106 difference equations on the Zuse computer at ETH (with a sufficiently accurate answer obtained in 90 iterations each approximately taking 2h 20 minutes.)
- In 1950 Lanczos iteration (including orthogonality of the basis and 3-term recurrence) applied to eigenvalue problems.
- Renaissance in early 1970 work by John Reid brought the connection to [iterative](#) methods. Game change: performance of CG is determined by the distribution of the eigenvalues of the matrix (preconditioning).
- In top 10 algorithms of 20th century.
- Nonlinear conjugate gradient method proposed by Fletcher and Reeves 1960.

Solution of linear system

$$Ax = b,$$

with A is symmetric positive definite matrix is equivalent to the quadratic optimisation problem

$$\min \phi(x) = \frac{1}{2}x^T Ax - b^T x.$$

Both have the same unique solution. In fact

$$\nabla \phi(x) = Ax - b =: r(x)$$

thus the linear system is the 1st order necessary condition (which is also sufficient for strictly convex function ϕ).

Conjugate directions

A set of non-zero vectors $\{p_0, p_1, \dots, p_{n-1}\}$ is said to be **conjugate** with respect to the symmetric positive definite matrix A if

$$p_i^T A p_j = 0, \quad i \neq j, \quad i, j = 1, \dots, n-1.$$

Conjugate directions are linearly independent.

Proof: [contradiction]: Assume $\{p_0, p_1, \dots, p_{n-1}\}$ are linearly dependent. Then

$$\exists \alpha_i > 0 : \alpha_0 p_0 + \alpha_1 p_1 + \dots + \alpha_{n-1} p_{n-1} = 0.$$

Multiplying with $p_i^T A$ from the left and using the conjugacy we obtain $\alpha_i p_i^T A p_i = 0$. Hence $\alpha_i > 0$, $p_i^T A p_i = 0$ contradicting the positive definiteness of A . \square .

Conjugacy enables us to minimise ϕ in n steps by successfully minimising it along the individual directions in the set.

Conjugate direction method

Given a starting point x_0 and the set of conjugate directions $\{p_0, p_2, \dots, p_{n-1}\}$ let us generate the sequence $\{x_k\}$

$$x_{k+1} = x_k + \alpha_k p_k,$$

where α_k is the one dimensional minimiser of the quadratic function along p_k , $\phi(x_k + \alpha p_k)$

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}.$$

For any $x_0 \in \mathbb{R}^n$ the sequence converges to the solution x^* in at most **n** steps.

Proof: Because $\text{span}\{p_0, p_2, \dots, p_{n-1}\} = \mathbb{R}^n$

$$x^* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + \cdots + \sigma_{n-1} p_{n-1}.$$

Multiplying from the left by $p_k^T A$ and using the conjugacy property we obtain σ_k as

$$\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k}, \quad k = 0, \dots, n-1.$$

On the other hand, in the k th iteration the method generates approximation

$$x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \cdots + \alpha_{k-1} p_{k-1}.$$

Multiplying from the left by $p_k^T A$ and using the conjugacy property we have $p_k^T A(x_k - x_0) = 0$ and

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k) = p_k^T (b - Ax_k) = -p_k^T r_k.$$

And hence $\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k} = -\frac{p_k^T r_k}{p_k^T A p_k} = \alpha_k$. \square

Theorem: [Expanding subspace minimisation]

For any starting point x_0 for the sequence $\{x_k\}$ generated by the conjugate direction method it holds

$$r_k^T p_i = 0, \quad i = 0, 1, \dots, k-1,$$

and x_k is the minimiser of $\phi(a) = \frac{1}{2}x^T Ax - b^T x$ over the set

$$\{x : x = x_0 + \text{span}\{p_0, p_1, \dots, p_{k-1}\}\}.$$

Proof: First let's characterise the minimiser on $\{p_0, p_1, \dots, p_{k-1}\}$. To this end define

$$h(\sigma) = \phi(x_0 + \sigma_0 p_0 + \dots + \sigma_{k-1} p_{k-1}),$$

where $\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{k-1})^T$. Since $h(\sigma)$ is a strictly convex quadratic, it has a unique minimiser σ^* that satisfies

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = 0, \quad i = 0, \dots, k-1.$$

Using the chain rule we obtain

$$\nabla \phi(x_0 + \sigma^* p_0 + \dots + \sigma_{k-1}^* p_{k-1})^T p_i = 0, \quad i = 0, 1, \dots, k-1.$$

Recalling that $r(x) = \nabla \phi(x)$, we have

$\tilde{x} = x_0 + \sigma_0^* p_0 + \dots + \sigma_{k-1}^* p_{k-1}$ is the minimiser on $\{p_0, p_1, \dots, p_{k-1}\}$ iff (if and only if) $r(\tilde{x})^T p_i = 0$ as claimed.

Use this characterisation of the minimiser to follow by induction:

For $k = 1$, from $x_1 = x_0 + \alpha_0 p_0$ being a minimiser of ϕ along p_0 it follows $r_1^T p_0 = 0$.

Suppose that $r_{k-1}^T p_i = 0$ for $i = 0, 1, \dots, k-2$.

$$r_k = Ax_k - b = A(x_{k-1} + \alpha_{k-1} p_{k-1}) - b = r_{k-1} + \alpha_{k-1} Ap_{k-1}.$$

and for $p_i, i = k-1$

$$p_{k-1}^T r_k = p_{k-1}^T r_{k-1} + \alpha_{k-1} p_{k-1}^T Ap_{k-1} = 0$$

by the definition of $\alpha_{k-1} = -\frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T Ap_{k-1}}$.

For any other $p_i, i = 0, 1, \dots, k-2$ we have

$$p_i^T r_k = p_i^T r_{k-1} + \alpha_{k-1} p_i^T Ap_{k-1} = 0,$$

where the first term disappears because of the induction hypothesis and the second because of the conjugacy of p_i . Thus we have shown $r_k^T p_i = 0$ for $i = 0, 1, \dots, k-1$ and the proof is complete.

□

Conjugate gradient vs conjugate direction

- So far the discussion was valid for any set of conjugate direction.
- An example are eigenvectors of a symmetric positive definite matrix A which are orthogonal and conjugate w.r.t. A .
Computation of full set of eigenvectors is expensive. Similarly, Gram Schmidt orthogonalisation process could be adopted to produce conjugate directions, however it is again expensive as it requires to store all the directions to orthogonalise against.
- Conjugate gradient (CG) method has a very special property, **it can compute a new vector p_k using only the previous vector p_{k-1}** i.e. it does not need to know the vectors p_0, p_1, \dots, p_{k-2} while p_k is automatically conjugate to those vectors. This makes CG particularly cheap in terms of computation and memory.

Conjugate gradient

In CG each new direction is chosen as

$$p_k = -r_k + \beta_k p_{k-1},$$

where

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}},$$

follows from requiring that p_{k-1}, p_k be conjugate
i.e. $p_{k-1}^T A p_k = 0$.

We initialise p_0 with the steepest descent direction at x_0 .

As in the conjugate direction method, we perform successive one dimensional minimisation along each of the search directions.

Given x_0

Set $r_0 = Ax_0 - b$, $p_0 = -r_0$, $k = 0$

while $r_k \neq 0$ **do**

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = Ax_{k+1} - b$$

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$$

$$k = k + 1$$

end while

Given x_0

Set $r_0 = Ax_0 - b$, $p_0 = -r_0$, $k = 0$

while $r_k \neq 0$ **do**

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$$

$$k = k + 1$$

end while

Theorem:

For the k th iterate of the conjugate gradient method, $x_k \neq x^*$ the following hold:

$$r_k^T r_i = 0, \quad i = 0, 1, \dots, k-1 \quad (1)$$

$$\text{span}\{r_0, r_1, \dots, r_k\} = \underbrace{\text{span}\{r_0, Ar_0, \dots, A^k r_0\}}_{=: \mathcal{K}_k(A, r_0)} \quad (2)$$

$$\text{span}\{p_0, p_1, \dots, p_k\} = \text{span}\{r_0, Ar_0, \dots, A^k r_0\} \quad (3)$$

$$p_k^T A p_i = 0, \quad i = 0, 1, \dots, k-1. \quad (4)$$

Therefore, the sequence $\{x_k\}$ converges to x^* in at most n steps.

The proof of this theorem relies on $p_0 = -r_0$. The result does not hold for other choices of p_0 .

Note that the gradients r_k are actually orthogonal, while the directions p_k are conjugate, thus the name of conjugate gradients is actually a misnomer.

Rate of convergence

From the properties of the $k + 1$ st iterate we have

$$x_{k+1} = x_0 + \alpha_0 p_0 + \cdots + \alpha_k p_k \quad (5)$$

$$= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0 \quad (6)$$

for some $\gamma_i, i = 0, \dots, k$.

Let P_k denote the k th degree polynomial

$$P_k(\lambda) = \gamma_0 + \gamma_1 \lambda + \cdots + \gamma_k \lambda^k$$

then

$$x_{k+1} = x_0 + P_k(A)r_0.$$

Recall that CG minimises the quadratic function ϕ over $x_0 + \text{span}\{p_0, \dots, p_k\}$ which is the same as $x_0 + \mathcal{K}_k(A, r_0)$ i.e.

$$\begin{aligned} \arg \min \phi(x) &= \arg \min \phi(x) - \phi(x^*) \\ &= \arg \min \frac{1}{2}(x - x^*)^T A(x - x^*) = \arg \min \frac{1}{2} \|x - x^*\|_A^2 \end{aligned}$$

CG vs steepest descent with optimal step length

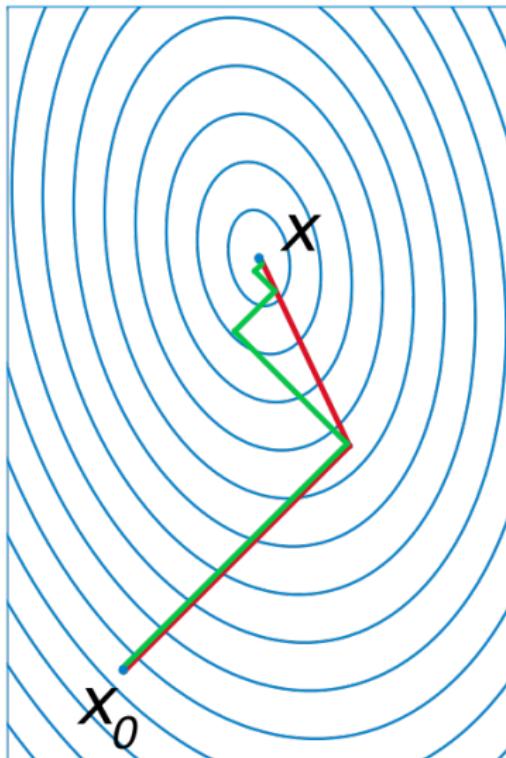


Figure: Wiki: Conjugate gradient method

Thus CG computes the minimising polynomial over all polynomials of degree k

$$\min_{P_k} \|x_0 + P_k(A)r_0 - x^*\|_A.$$

Observe that similar expressions hold for the error

$$\begin{aligned} x_k - x^* &= x_0 + P_{k-1}(A)r_0 - x^* = x_0 - x^* + P_{k-1}(A) \underbrace{A(x_0 - x^*)}_{=r_0} \\ &= [I + AP_{k-1}(A)](x_0 - x^*) \end{aligned}$$

and the residual

$$\begin{aligned} r_k &= Ax_k - b = A(x_k - x^*) = A(x_0 + P_{k-1}(A)r_0 - x^*) \\ &= \underbrace{A(x_0 - x^*)}_{=r_0} + AP_{k-1}(A)r_0 = [I + AP_{k-1}(A)]r_0 \end{aligned}$$

Let the eigenvalue decomposition of the symmetric positive definite matrix

$$A = V^T \Lambda V = \sum_i^n \lambda_i v_i v_i^T,$$

with $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ and $v_i, i = 1, \dots, n$ the corresponding orthogonal eigenvectors.

Since V is a basis for any vector in \mathbb{R}^n , in particular for $x_0 - x^* = \sum_{i=1}^n \xi_i v_i$.

Notice that any eigenvector v_i of A is also an eigenvector of $P_k(A)$ with the corresponding eigenvalue $P_k(\lambda_i)$

$$P_k(A)v_i = P_k(\lambda_i)v_i, \quad i = 1, \dots, n.$$

Hence

$$x_{k+1} - x^* = \sum_{i=1}^n [1 + \lambda_i P_k(\lambda_i)] \xi_i v_i$$

and

$$\|x_{k+1} - x^*\|_A^2 = \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2$$

Since P_k is optimal w.r.t. this norm we have

$$\begin{aligned}\|x_{k+1} - x^*\|_A^2 &= \min_{P_k} \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2 \\ &\leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \left(\sum_{i=1}^n \lambda_i \xi_i^2 \right) \\ &= \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|_A^2.\end{aligned}$$

Theorem If A has only r distinct eigenvalues, then CG will converge to the solution in at most r iterations.

Proof: Suppose the eigenvalues take on distinct r values $\tau_1 < \dots < \tau_r$ and define a polynomial

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \dots \tau_r} (\lambda - \tau_1) \dots (\lambda - \tau_r)$$

and note that $Q_r(\lambda_i) = 0, i = 1, \dots, n$ and $Q(0) = 1$. Then

$$\bar{P}_{r-1}(\lambda) = (Q_r(\lambda) - 1)/\lambda$$

is of degree $r - 1$ and we have

$$\begin{aligned} 0 &\leq \min_{P_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 \\ &\leq \max_{1 \leq i \leq n} [1 + \lambda_i \bar{P}_{r-1}(\lambda_i)]^2 = \max_{1 \leq i \leq n} Q_r^2(\lambda_i) = 0 \end{aligned}$$

and $\|x_r - x^*\|_A^2 = 0$ and hence $x_r = x^*$.

Convergence rate

Theorem If A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, we have that

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x^*\|_A^2.$$

Proof idea: Choose polynomial \bar{P}_k such that

$Q_{k+1}(\lambda) = 1 + \lambda \bar{P}_k(\lambda)$ has roots at the k largest eigenvalues $\lambda_n, \lambda_{n-1}, \dots, \lambda_{n-k+1}$ and at the midpoint between λ_{n-k} and λ_1 . It can be shown that the maximum value attained by Q_{k+1} on the remaining eigenvalues $\lambda_1, \dots, \lambda_{n-k}$ is $\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}$.

Theorem In terms of condition number

$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \lambda_n / \lambda_1$, we have that

$$\|x_k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A.$$

Preconditioning

We can accelerate CG through transformations which cluster eigenvalues. This process is known as **preconditioning**.

We perform a change of variables $\hat{x} = Cx$.

Then the quadratic function ϕ in terms of \hat{x} reads

$$\hat{\phi}(\hat{x}) = \frac{1}{2}\hat{x}^T(C^{-T}AC^{-1})\hat{x} - (C^{-T}b)^T\hat{x}.$$

Minimising $\hat{\phi}$ is equivalent to solving the system of normal equations

$$(C^{-T}AC^{-1})\hat{x} = (C^{-T}b)$$

and the convergence rate of CG depends on the eigenvalues of $C^{-T}AC^{-1}$.

It is not necessary to carry out the transforms explicitly. We can apply CG to $\hat{\phi}$ in terms of \hat{x} and then invert the transformations to reexpress all the equations in terms of the original variable x .

In fact, the **preconditioned CG** algorithm does not use the factorisation $M = C^T C$ explicitly, only M .

If we set $M = I$ we recover unpreconditioned CG algorithm.

The properties of CG generalise, in particular for PCG it holds

$$r_i^T M^{-1} r_j = 0, \quad \forall i \neq j.$$

Preconditioned CG (PCG)

Given x_0 , preconditioner M

Set $r_0 = Ax_0 - b$,

Solve $My_0 = r_0$

$p_0 = -y_0$, $k = 0$

while $r_k \neq 0$ **do**

$$\alpha_k = \frac{r_k^T y_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

Solve $My_{k+1} = r_{k+1}$

$$\beta_{k+1} = \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$$

$$p_{k+1} = -y_{k+1} + \beta_{k+1} p_k$$

$$k = k + 1$$

end while

Nonlinear conjugate gradient

Recall that CG can be interpreted as a minimiser of a quadratic convex function

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b.$$

Can the algorithm for ϕ be generalised to a nonlinear function f ?

Recall that

- step length α_k minimises ϕ along p_k .
For general f : compute α_k using line search

$$\alpha_k = \min_{\alpha} f(x_k + \alpha p_k)$$

- $r = Ax - b = \nabla \phi(x)$.
For general function f : $r \rightarrow \nabla f$

Fletcher Reeves

Given x_0

Evaluate $f_0 = f(x_0)$, $\nabla f_0 = \nabla f(x_0)$

Set $p_0 = -\nabla f_0$, $k = 0$

while $\nabla f_k \neq 0$ **do**

 Compute α_k using line search, $\alpha_k = \min_{\alpha} f(x_k + \alpha p_k)$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$\beta_{k+1} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$$

$$p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$$

$$k = k + 1$$

end while

Descent direction

Is p_k a descent direction?

$$\nabla f_k^T p_k = -\nabla f_k^T \nabla f_k + \beta_k \nabla f_k^T p_{k-1} \stackrel{?}{<} 0$$

If α_{k-1} is a local minimiser along p_{k-1} , $\nabla f_k^T p_{k-1} = 0$ and

$$\nabla f_k^T p_k = -\nabla f_k^T \nabla f_k < 0$$

thus p_k is a descent direction.

If the linear search is not exact, due to the second term $\beta_k \nabla f_k^T p_{k-1}$, p_k may fail to be a descent direction. This can be avoided by requiring that the step length α_{k-1} satisfies the strong Wolfe conditions

$$\begin{aligned} f(x_k + \alpha_k p_k) &\leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k, \\ |\nabla f(x_k + \alpha_k p_k)^T p_k| &\leq -c_2 \nabla f_k^T p_k \end{aligned}$$

with $0 < c_1 < c_2 < \frac{1}{2}$.

Lemma [1]

Let f be twice continuously differentiable, and the level set $\{x : f(x) \leq f(x_0)\}$ is bounded. If the step length α_k in the FR algorithm satisfies strong Wolfe conditions with $0 < c_2 < \frac{1}{2}$ then the method generates descent directions p_k that satisfy

$$\underbrace{-\frac{1}{1-c_2}}_{\in (-2,-1)} \leq \frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2} \leq \underbrace{\frac{2c_2 - 1}{1-c_2}}_{\in (-1,0)}, \quad k = 1, 2, \dots \quad (7)$$

Proof: First note that the upper bound $(2c_2 - 1)/(1 - c_2)$ monotonically increases for $c_2 \in (0, \frac{1}{2})$ and $-1 < (2c_2 - 1)/(1 - c_2) < 0$. Thus Lemma [1] implies that p_k is a descent direction $\nabla f_k^T p_k < 0$.

The inequalities can be shown by induction using the form of the update along with the second strong Wolfe condition.

Induction:

$$k = 0 : \quad p_0 = -\nabla f_0 \rightarrow \frac{\nabla f_0^T p_0}{\|\nabla f_0\|^2} = -1 \text{ and (7) holds.}$$

Assume (7) holds for some $k \geq 1$.

$$\frac{\nabla f_{k+1}^T p_{k+1}}{\|\nabla f_{k+1}\|^2} = -1 + \beta_{k+1}^{FR} \frac{\nabla f_{k+1}^T p_k}{\|\nabla f_{k+1}\|^2} = -1 + \frac{\nabla f_{k+1}^T p_k}{\|\nabla f_k\|^2},$$

where we used $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k$.

Plugging curvature Wolfe condition $|\nabla f_{k+1}^T p_k| \leq -c_2 \nabla f_k^T p_k$ into last equation (note $\nabla f_k^T p_k < 0$ by induction hypothesis) we obtain

$$-1 + c_2 \frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2} \leq \frac{\nabla f_{k+1}^T p_{k+1}}{\|\nabla f_{k+1}\|^2} \leq -1 - c_2 \frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2}.$$

Substituting the lower bound for $\frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2}$ from induction hypothesis we obtain (7) for $k + 1$

$$-1 - \frac{c_2}{1 - c_2} \leq \frac{\nabla f_{k+1}^T p_{k+1}}{\|\nabla f_{k+1}\|^2} \leq -1 + \frac{c_2}{1 - c_2}. \quad \square$$

Weakness of FR algorithm

If FR generates a bad direction and a tiny step, then the next direction and the next step are also likely to be poor.

Let $\theta_k = \angle(p_k, -\nabla f_k)$,

$$\cos \theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\| \|p_k\|}.$$

A bad direction p_k is almost orthogonal to $-\nabla f_k$ and $\cos \theta_k \approx 0$.
Multiplying (7) by $-\|\nabla f_k\|/\|p_k\|$ we obtain

$$\frac{1 - 2c_2}{1 - c_2} \frac{\|\nabla f_k\|}{\|p_k\|} \leq \cos \theta_k \leq \frac{1}{1 - c_2} \frac{\|\nabla f_k\|}{\|p_k\|}, \quad k = 1, 2, \dots$$

Thus $\cos \theta_k \approx 0$ if and only if $\|\nabla f_k\| \ll \|p_k\|$.

Since p_k is almost orthogonal to $-\nabla f_k$, the step from x_k to x_{k+1} is likely tiny, i.e. $x_{k+1} \approx x_k$. Consequently, $\nabla f_k \approx \nabla f_{k+1}$ then $\beta_{k+1} \approx 1$ and finally given $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| \ll \|p_k\|$, $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k \approx p_k$ - stagnation i.e. the following updates are unproductive.

Polak-Ribière:

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \quad (8)$$

If f is strongly convex quadratic function and the line search is exact (FR itself reduces to CG), $\nabla f_{k+1} \perp \nabla f_k$ and $\beta_{k+1}^{PR} = \beta_{k+1}^{FR}$.

For general nonlinear functions and inexact line search, numerical experience indicates that PR algorithm is more robust and efficient.

As is, the strong Wolfe conditions do not guarantee that p_k is always a descent direction. For $\beta_{k+1} = \max\{\beta_{k+1}^{PR}, 0\}$, simple adaptation of strong Wolfe conditions ensures the descent property.

Other choices of β_k

Hestenes - Stiefel (similar to PR in both theory and practical performance):

Consecutive directions are conjugate wrt *average Hessian*

$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau$. From Taylor's theorem we have

$\nabla f_{k+1} = \nabla f_k + \alpha_k \bar{G}_k p_k$. Solving $p_{k+1}^\top \bar{G}_k p_k = 0$ where

$p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$ for β_{k+1} yields

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^\top (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^\top p_k} \quad (9)$$

Two competitive with PR choices which guarantee p_k to be descent direction under (standard) Wolfe conditions on α_k :

$$\beta_{k+1} = \frac{\|\nabla f_{k+1}\|^2}{(\nabla f_{k+1} - \nabla f_k)^\top p_k} \quad (10)$$

$$\beta_{k+1} = \left(y_k - 2p_k \frac{\|y_k\|^2}{y_k^\top p_k} \right)^\top \frac{\nabla f_{k+1}}{y_k^\top p_k} \text{ with } y_k = \nabla f_{k+1} - \nabla f_k. \quad (11)$$

Restarts

Set $\beta_k = 0$ in every n th step i.e. take steepest descent step.
Restarting serves to refresh the algorithm erasing old information
that may be not beneficial. Such restarting leads to n step
quadratic convergence $\|x_{k+n} - x^*\| = \mathcal{O}(\|x_k - x^*\|^2)$.

Consider function which is strongly convex quadratic close to the
solution x^* but non-quadratic elsewhere. Once close to the
solution the restart will allow the method to behave like linear
conjugate gradients, in particular with finite termination within n
steps from the restart (recall that the finite termination property
for linear CG only holds if initiated with $p_0 = -\nabla f_0$).

In practice, conjugate gradient methods are usually used when n is
large, hence n steps are never taken. Observe that the gradients
are mutually orthogonal when f is a quadratic function. Restart
when two consecutive gradients are far from orthogonal

$$\frac{|\nabla f_k^T \nabla f_{k+1}|}{\|\nabla f_k\|^2} \geq \nu, \text{ with } \nu \text{ typically } 0.1.$$

When for some search direction p_k , $\cos \theta_k \approx 0$ and the subsequent step is small, substituting $\nabla f_{k+1} \approx \nabla f_k$ into β_{k+1}^{PR} results in $\beta_{k+1}^{PR} \approx 0$ and the next direction $p_{k+1} \approx -\nabla f_{k+1}$ the steepest descent direction. Therefore the PR algorithm essentially performs a restart after it encounters a bad direction.

The same argument applies to HS, and PR+.

FR algorithm requires some restart.

Hybrid FR-PR:

Global convergence can be guaranteed if $|\beta_k| \leq \beta_k^{FR}$ for all $k \geq 2$.

This suggest following strategy

$$\beta_k = \begin{cases} -\beta_k^{FR}, & \beta_k^{PR} < -\beta_k^{FR} \\ \beta_k^{PR}, & |\beta_k^{PR}| \leq \beta_k^{FR} \\ \beta_k^{FR}, & \beta_k^{PR} > \beta_k^{FR} \end{cases} \quad (12)$$

Assumptions:

- i) The level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ be bounded.
- ii) In some open neighbourhood \mathcal{N} of \mathcal{L} , the objective function f is Lipschitz continuously differentiable.

These assumptions imply that there is a constant γ such that

$$\|\nabla f(x)\| \leq \gamma, \quad \forall x \in \mathcal{L}.$$

Global convergence - **restarted** CG method

From Zoutenjik's lemma it follows that any line search iteration $x_{k+1} = x_k + \alpha_k p_k$ where p_k is a descent direction and the step length α_k satisfies Wolfe conditions gives the limit

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty.$$

Similarly, to the global convergence for line search, global convergence for **restarted** conjugate gradient algorithms periodically setting $\beta_k = 0$ (hence $\cos \theta_{k+1} = 1$) can be proven in a subsequence

$$\liminf_{k \rightarrow \infty} \|\nabla f_k\| = 0.$$

Theorem: [Al-Baali] Suppose that the assumptions i) and ii) hold and FR algorithm is implemented with line search that satisfies strong Wolfe conditions with $0 < c_1 < c_2 < \frac{1}{2}$. Then

$$\liminf_{k \rightarrow \infty} \|\nabla f_k\| = 0.$$

Proof: By contradiction (assume $\|\nabla f_k\| \geq \gamma > 0$). Substitute into Zoutenjik's result and use definition of p_k and upper bound in Lemma [1] recursively to show that the assumed to converge sequence is lower bounded by harmonic series which is divergent hence contradiction.

This global convergence result can be extended to any method satisfying $|\beta_k| \leq \beta_k^{FR}$ for all $k \geq 2$.

If constants $c_4, c_5 > 0$ exist such that

$$\cos \theta_k \geq c_4 \frac{\|\nabla f_k\|}{\|p_k\|}, \quad \frac{\|\nabla f_k\|}{\|p_k\|} \geq c_5 > 0, \quad k = 1, 2, \dots$$

it follows from Zoutenjik's result that

$$\lim_{k \rightarrow \infty} \|\nabla f_k\| = 0.$$

This result can be established for PR for f strongly convex and exact line search.

For general nonconvex functions it is not possible even though PR performs better in practice than FR. PR method can cycle infinitely even if ideal line search is used i.e. line search which returns α_k that is the first positive stationary point of $f(x_k + \alpha p_k)$. Example relies on $\beta_k < 0$ which motivated the modification $\beta_k^+ = \max\{0, \beta_k\}$.

The simplifications for final version of CG.

$$\alpha_k = -\frac{r_k^T P_k}{P_k^T A P_k} = -\frac{r_k^T (-r_k + \beta_k p_{k-1})}{P_k^T A P_k}$$

$$= +\frac{r_k^T r_k}{P_k^T A P_k} - \beta_k \frac{r_k^T p_{k-1}}{P_k^T A P_k} \quad \text{⊗} = 0$$

$$r_{k+1} = Ax_{k+1} - b = A(x_k + \alpha_k p_k) - b = \underbrace{Ax_k - b}_{r_k} + \alpha_k A P_k$$

$$\text{⊗} \quad r_k^T p_{k-1} = (r_{k-1} + \alpha_{k-1} A p_{k-1})^T p_{k-1} = \\ = r_{k-1}^T p_{k-1} + \alpha_{k-1} p_{k-1}^T A p_{k-1} \\ = r_{k-1}^T p_{k-1} + \frac{-r_{k-1}^T p_{k-1}}{P_{k-1}^T A P_{k-1}} \cdot P_{k-1}^T A P_{k-1} = 0$$

$$\beta_k = \frac{r_{k+1}^T A P_k}{P_k^T A P_k} = \frac{r_{k+1}^T \frac{1}{\alpha_k} (r_{k+1} - r_k)}{P_k^T A P_k} \quad \leftarrow \text{subst } \square$$

$$= \frac{r_{k+1}^T r_{k+1}}{P_k^T A P_k} \cdot \frac{1}{\alpha_k} + \frac{r_{k+1}^T r_k}{P_k^T A P_k} \cdot \frac{1}{\alpha_k} \xrightarrow{\text{conjugate rel.}} 0$$

$$= \frac{r_{k+1}^T r_{k+1}}{P_k^T A P_k} \quad \frac{P_k^T A P_k}{r_k^T r_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$\text{Δ} \quad r_{k+1}^T r_k = \underbrace{(r_k + \alpha_k A P_k)^T r_k}_{\text{subst } \square} = r_k^T r_k + \alpha_k P_k^T A r_k \\ = r_k^T r_k + \alpha_k P_k^T A \cancel{P_k} (-P_k + \beta_k P_{k-1})$$

$$r_{k+1}^T r_k = r_k^T r_k + \alpha_k P_k^T A P_k + \alpha_k \underbrace{P_k^T A P_{k-1}}_{=0 \text{ conj. } P} \cdot \beta_{k-1} \\ = r_k^T r_k - \frac{r_k^T r_k}{P_k^T A P_k} P_k^T A P_k = 0$$

Numerical Optimisation: Quasi-Newton methods

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 7 & 8

- First idea by William C. Davidon in mid 1950, who was frustrated by performance of coordinate descent.
- Quickly picked up by Fletcher and Powell who demonstrated that the new algorithm was much faster and more reliable than existing methods.
- Davidon's original paper was not accepted for publication. More than 30 years later it appeared in the first issue of the SIAM Journal on Optimization in 1991.
- Like steepest gradient, Quasi Newton methods only require the gradient of the objective function at each iterate. Measuring changes in gradient they build a model of the objective function which is good enough to produce superlinear convergence.
- As the Hessian is not required, Quasi-Newton methods can be more efficient than Newton methods which take a long time to evaluate the Hessian and solve for the Newton direction.

Quasi-Newton

Quadratic model of the objective function at x_k :

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p,$$

where $B_k \in \mathbb{R}^{n \times n}$ symmetric positive definite which will be updated during the iteration.

The minimiser of m_k can be written explicitly

$$p_k = -B_k^{-1} \nabla f_k.$$

p_k is used as a search direction and the next iterate becomes

$$x_{k+1} = x_k + \alpha_k p_k.$$

The step length α_k is chosen to satisfy the Wolfe conditions.

The iteration is similar to the line search Newton with the key difference that the Hessian B_k is an approximation.

B_k update

Davidon proposed to update B_k in each iteration instead of computing it anew.

Question: Having computed the new iterate x_{k+1} , when we construct the new model

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p,$$

what requirements should we impose on B_{k+1} based on the knowledge gathered in the last step?

Require: gradient of m_{k+1} should match the gradient of f at the last two iterates x_k, x_{k+1} .

- i) At x_{k+1} : $p_{k+1} = 0$,
 $\nabla m_{k+1}(0) = \nabla f_{k+1}$ is satisfied automatically.
- ii) At $x_k = x_{k+1} - \alpha_k p_k$:

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k.$$

By rearranging ii) we obtain

$$B_{k+1}\alpha_k p_k = \nabla f_{k+1} - \nabla f_k.$$

Define vectors

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \quad y_k = \nabla f_{k+1} - \nabla f_k,$$

ii) becomes the *secant equation*

$$B_{k+1}s_k = y_k.$$

As B_{k+1} is symmetric positive definite, this is only possible if the *curvature condition* holds

$$s_k^T y_k > 0,$$

which can be easily seen multiplying the secant equation by s_k^T from the left.

If f is strongly convex $s_k^T y_k > 0$ is satisfied for any x_k, x_{k+1} . However, for nonconvex functions in general this condition will have to be enforced explicitly by imposing restrictions on the line search.

$s_k^T y_k > 0$ is guaranteed if we impose Wolfe or strong Wolfe conditions:

From the 2nd Wolfe condition $s_k^T \nabla f_{k+1} \geq c_2 s_k^T \nabla f_k$, $c_1 < c_2 < 1$ it follows

$$s_k^T y_k \geq (c_2 - 1) \alpha_k p_k^T \nabla f_k > 0,$$

since $c_2 < 1$ and p_k is a descent direction, and the curvature condition holds.

Davidon Fletcher Powell (DFP)

When $s_k^T y_k > 0$, the secant equation always has a solution B_{k+1} .

In fact the secant equation is heavily underdetermined: a symmetric matrix has $n(n + 1)/2$ dofs, secant equation: n conditions, positive definiteness: n inequalities.

Extra conditions to obtain unique solutions: we look for B_{k+1} close to B_k in a certain sense.

DFP update:

$$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T \quad (\text{DFP B})$$

with $\rho_k = 1/y_k^T s_k$.

The inverse $H_k = B_k^{-1}$ can be obtained with Sherman-Morrison-Woodbury formula

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}. \quad (\text{DFP H})$$

Sherman-Morrison-Woodbury formula

This formula can be extended to higher-rank updates. Let U and V be matrices in $\mathbb{R}^{n \times p}$ for some p between 1 and n . If we define

$$\hat{A} = A + UV^T,$$

then \hat{A} is nonsingular if and only if $(I + V^T A^{-1} U)$ is nonsingular, and in this case we have

$$\hat{A}^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^TA^{-1}. \quad (\text{A.28})$$

Figure: Nocedal Wright (A.28)

Applying the same argument directly to the inverse of the Hessian H_k . The updated approximation H_{k+1} must be symmetric and positive definite and must satisfy the secant equation

$$H_{k+1}y_k = s_k.$$

BFGS update:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T)^T + \rho_k s_k s_k^T \quad (\text{BFGS})$$

with $\rho_k = 1/y_k^T s_k$.

How to choose H_0 ? Depends on the situation, information about the problem e.g. start with an inverse of an approximated Hessian calculated by a finite difference at x_0 . Otherwise, we can set H_0 to identity or diagonal matrix to reflect the scaling of the variables.

- 1: Given x_0 , inverse Hessian approximation H_0 , tolerance $\varepsilon > 0$
- 2: Set $k = 0$
- 3: **while** $\|\nabla f_k\| > \varepsilon$ **do**
- 4: Compute search direction

$$p_k = -H_k \nabla f_k$$

- 5: $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed with a line search procedure satisfying Wolfe conditions
- 6: Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$
- 7: Compute H_{k+1} using (BFGS)
- 8: $k = k + 1$
- 9: **end while**

- Complexity of each iteration is $\mathcal{O}(n^2)$ plus the cost of function and gradient evaluations.
- There are no $\mathcal{O}(n^3)$ operations such as linear system solves or matrix-matrix multiplications.
- The algorithm is robust and the rate of convergence is superlinear. In many cases it outperforms Newton method, which while converging quadratically, has higher complexity per iteration (Hessian computation and solve).
- A BFGS version with the Hessian approximation B_k rather than H_k . The update for B_k is obtained by applying Sherman-Morrison-Woodbury formula to (BFGS)

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

An $\mathcal{O}(n^2)$ implementation can be achieved based on updates of LDL^T factors of B_k (with possible diagonal modification for stability) but no computational advantage is observed on above algorithm using (BFGS) to update H_k .

- The positive definiteness of H_k is not explicitly forced, but if H_k is positive definite so will be H_{k+1} .
- What happens if at some iteration H_k becomes as poor approximation to the true inverse Hessian e.g. if $s_k^T y_k$ is tiny (positive) than the elements of H_{k+1} get very large.
It turns out that BFGS has effective self correcting properties, and H_k tends to recover in a few steps. The self correcting properties hold only when adequate line search is performed. In particular Wolfe conditions ensure that the gradients are sampled at points which allow the model m_k to capture the curvature information.
- On the other hand DFP method is less effective in correcting itself.
- DFP and BFGS are dual in the sense that they can be obtained by switching $s \leftrightarrow y, B \leftrightarrow H$.

- $\alpha_k = 1$ should always be tried first, because this step length will eventually be accepted (under certain conditions), thereby producing super linear convergence.
- Computational evidence suggests that it is more economical (in terms of function evaluations) to perform fairly inaccurate line search.
- $c_1 = 10^{-4}$, $c_2 = 0.9$ are commonly used with Wolfe conditions.

Heuristic for scaling H_0

Choice $H_0 = \beta I$ is popular, but there is no good strategy for estimating β .

If β is too large, the first step $p_0 = -\beta g_0$ is too long and line search may require many iterations to find a suitable step length α_0 .

Heuristic: estimate β after the first step has been computed (using $H_0 = I$ amounts to steepest descent step) but before the H_0 update (in step 7) and change the provisional value by setting $H_0 = \frac{s_k^T y_k}{y_k^T y_k} I$. This scaling attempts to approximate scaling with an eigenvalue of the inverse Hessian: from Taylor theorem

$$y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k$$

we have that the secant equation is satisfied for average Hessian

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau.$$

Symmetric rank-1 (SR-1) update

Both BFGS and DFP methods perform a rank-2 update while preserving symmetry and positive definiteness.

Question: Does a rank-1 update exist such that the secant equation is satisfied and the symmetry and definiteness are preserved?

Rank-1 update:

$$B_{k+1} = B_k + \sigma v v^T, \quad \sigma \in \{+1, -1\}$$

and v is chosen such that B_{k+1} satisfies the secant equation

$$y_k = B_{k+1} s_k.$$

Substituting the explicit rank-1 form into the secant equation

$$y_k = B_k s_k + \underbrace{(\sigma v^T s_k)}_{:=\delta^{-1}, \delta \neq 0} v$$

we see that v must be of the form $v = \delta(y_k - B_k s_k)$.

Substituting the explicit form of $v = \delta(y_k - B_k s_k)$ back into the last equation we obtain

$$y_k - B_k s_k = \sigma \delta^2 [s_k^T (y_k - B_k s_k)] (y_k - B_k s_k)$$

which is satisfied if and only if

$$\sigma = \text{sign}[s_k^T (y_k - B_k s_k)], \quad \delta = \pm |s_k^T (y_k - B_k s_k)|^{-1/2}.$$

Hence, the only symmetric rank-1 update satisfying the secant equation is

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

Applying the Sherman-Morrison-Woodbury formula we obtain the inverse Hessian update

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \quad (\text{SR-1})$$

SR-1 update does not preserve the positive definiteness. It is a drawback for line search methods but could be an asset for trust region as it allows to generate indefinite Hessians.

SR-1 breakdown

The main drawback of SR-1 is that $(y_k - B_k s_k)^T s_k$ (same for H_k) can become 0 even for a convex quadratic function i.e. there may be steps where there is no symmetric rank-1 update which satisfies the secant equation.

Three cases:

- $(y_k - B_k s_k)^T s_k \neq 0$, unique symmetric rank-1 update satisfying secant equation exists.
- $y_k = B_k s_k$, then the only update is $B_{k+1} = B_k$.
- $(y_k - B_k s_k)^T s_k = 0$ and $y_k \neq B_k s_k$, there is no symmetric rank-1 update satisfying secant equation.

Remedy: Skipping i.e. apply update only if

$$|(y_k - B_k s_k)^T s_k| \geq r \|s_k\| \|y_k - B_k s_k\|,$$

where $r \in (0, 1)$ is a small number (typically $r = 10^{-8}$), otherwise set $B_{k+1} = B_k$.

SR-1 applicability

- This simple safeguard adequately prevents the breakdown.
Recall: for BFGS update skipping is not recommended if the curvature condition $s_k^T y_k > 0$ fails. Because it can occur often by e.g. taking too small step if the line search does not impose the Wolfe conditions. For SR-1 $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently as it requires near orthogonality of s_k and $y_k - B_k s_k$ and moreover it implies that $s_k^T \bar{G}_k s_k \approx s_k^T B_k s_k$, where \bar{G}_k is the average Hessian over the last step meaning that the curvature approximation along s_k is essentially already correct.
- The Hessian approximations generated by SR-1 are good, often better than those by BFGS.
- When the curvature condition $y_k^T s_k > 0$ cannot be imposed e.g. constraint problems or partially separable functions, where indefinite Hessian approximations are desirable as they reflect the indefiniteness of the true Hessian.

SR-1 trust-region method

```
1: Given  $x_0$ ,  $B_0$ ,  $\Delta$ ,  $\eta \in (0, 10^{-3})$ ,  $r \in (0, 1)$  and  $\varepsilon > 0$ 
2: Set  $k = 0$ 
3: while  $\|\nabla f_k\| > \varepsilon$  do
4:    $s_k = \arg \min_s s^T \nabla f_k + \frac{1}{2} s^T B_k s$ , s. t.  $\|s\| \leq \Delta_k$ ,  $B_k$  indefinite
5:    $y_k = \nabla f(x_k + s_k) - \nabla f_k$ 
6:    $\rho_k = (f_k - f(x_k + s_k)) / -(s_k^T \nabla f_k + \frac{1}{2} s_k^T B_k s_k)$ 
7:   if  $\rho_k > \eta$  then
8:      $x_{k+1} = x_k + s_k$ 
9:   else
10:     $x_{k+1} = x_k$  (failed step)
11:   end if
12:   Update  $\Delta_k$  in dependence of  $\rho_k$ ,  $\|s_k\|$  (as in trust-region methods)
13:   if  $|(y_k - B_k s_k)^T s_k| \geq r \|s_k\| \|y_k - B_k s_k\|$  then
14:     Update  $B_{k+1}$  using (SR-1) (even if  $x_{k+1} = x_k$  to improve bad
        approximation along  $s_k$ )
15:   else
16:      $B_{k+1} = B_k$ 
17:   end if
18:    $k = k + 1$ 
19: end while
```

Theorem: Hessian approximation for quadratic function

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a strongly quadratic function

$f(x) = b^T x + \frac{1}{2} x^T A x$ with A symmetric positive definite. For any starting point x_0 and any symmetric initial matrix H_0 , the iterates

$$x_{k+1} = x_k + p_k, \quad p_k = -H_k \nabla f_k,$$

where H_k is updated with (SR-1), converge to the minimiser in at most n steps provided that $(s_k - H_k y_k)^T y_k \neq 0$ for all k . After n steps, if the search directions p_k are linearly independent,
 $H_n = A^{-1}$.

Proof idea: Show by induction that the secant equation $H_k y_j = s_j$ is satisfied for all $j = 1, \dots, k-1$ i.e. H_k (not merely the last one $k-1$). Use that for such quadratic function it holds $y_i = A s_i$.

For SR-1 $H_k y_j = s_j$, $j = 1, \dots, k-1$ holds regardless how the line search is performed. In contrast for BFGS, it can only be shown under the assumption that the line search is exact.

Theorem: Hessian approximation for general function

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ twice continuously differentiable with the Hessian bounded and Lipschitz continuous in a neighbourhood of a point $x^* \in \mathbb{R}^n$ and $\{x_k\}$ a sequence of iterates such that $x_k \rightarrow x^*$. Suppose that

$$|(y_k - B_k s_k)^T s_k| \geq r \|s_k\| \|y_k - B_k s_k\|$$

holds for all k and some $r \in (0, 1)$ and that the steps s_k are uniformly independent (steps do not tend to fall in a subspace of dimension less than n).

Then the matrices B_k generated by the update (SR-1) satisfy

$$\lim_{k \rightarrow \infty} \|B_k - \nabla^2 f(x^*)\| = 0.$$

The Broyden class

Broyden class is a family of updates of the form

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k^T} + \frac{y_k y_k^T}{y_k^T s_k} + \tau_k (s_k^T B_k s_k) v_k v_k^T, \quad (\text{Broyden})$$

where τ_k is a scalar parameter and

$$v_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}.$$

For $\tau_k = 0$ we recover BFGS and for $\tau_k = 1$ we DFP.

Hence we can write (Broyden) as a linear combination of the two

$$B_{k+1} = (1 - \tau_k) B_{k+1}^{\text{BFGS}} + \tau_k B_{k+1}^{\text{DFP}}.$$

Since both BFGS and DFP satisfy secant equation so does the whole Broyden class.

Since BFGS and DFP updates preserve positive definiteness of the Hessian when $s_k^T y_k > 0$, so does the **restricted Broyden class** which is obtained by restricting $0 \leq \tau_k \leq 1$.

Theorem: monotonicity of eigenvalue approximation

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the strongly convex quadratic function

$f(x) = b^T x + \frac{1}{2} x^T A x$ with A symmetric positive definite. Let B_0 any symmetric positive matrix and x_0 be any starting point for the iteration

$$x_{k+1} = x_k + p_k, \quad p_k = -B_k^{-1} \nabla f_k,$$

where B_k is updated with (Broyden) with $\tau_k \in [0, 1]$.

Denote with $\lambda_1^k \leq \lambda_2^k \leq \dots \leq \lambda_n^k$ the eigenvalues of

$$A^{1/2} B_k^{-1} A^{1/2}.$$

Then for all k , we have

$$\min\{\lambda_i^k, 1\} \leq \lambda_i^{k+1} \leq \max\{\lambda_i^k, 1\}, \quad i = 1, \dots, n.$$

The interlacing property does not hold if $\tau_k \notin [0, 1]$.

Consequence: The eigenvalues λ_i^k converge monotonically (but not strictly monotonically) to 1, which are the eigenvalues when $B_k = A$. Significantly, the result holds even if the line search is not exact.

So do the best updates belong to the restricted Broyden class?

We recover SR-1 formula for

$$\tau_k = \frac{s_k^T y_k}{s_k^T y_k - s_k^T B_k s_k},$$

which does not belong to the restricted Broyden class as τ_k may fall outside of $[0, 1]$.

It can be shown that for B_0 symmetric positive definite, if for all k $s_k^T y_k > 0$ and $\tau_k > \tau_k^c$, then all B_k generated by (Broyden) remain symmetric and positive definite. Here

$$\tau_k^c = (1 - \mu_k)^{-1} \leq 0, \quad \mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2} \geq 1.$$

When the [line search is exact](#) all the methods in the Broyden class with $\tau_k \geq \tau_k^c$ generate the same sequence of iterates, even for nonlinear functions because the directions differ only by length and this is compensated by the exact line search.

Thm: Properties of Broyden class for quadratic function

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be the strongly convex quadratic function

$f(x) = b^T x + \frac{1}{2} x^T A x$ with A symmetric positive definite. Let x_0 be any starting point and B_0 any symmetric positive definite matrix. Assume that α_k is the exact step length and $\tau_k \geq \tau_k^c$ for all k . Then it holds

- (i) The iterates are independent of τ_k and converge to the solution in at most n iterations.
- (ii) The secant equation is satisfied for all previous search directions

$$B_k s_j = y_j, \quad j = 1, \dots, k-1.$$

- (iii) If $B_0 = I$, then the sequence of iterates $\{x_k\}$ is identical to that generated by the conjugate gradient method, in particular the search directions s_k are conjugate

$$s_i^T A s_j = 0, \quad i \neq j.$$

- (iv) If n iterations are performed, we have $B_n = A$.

- The theorem can be slightly generalised to hold if the Hessian approximation remains nonsingular but not necessarily positive definite i.e. τ_k could be smaller than τ_k^c provided the chosen value did not produce singular updated matrix.
- (iii) can be generalised to $B_0 \neq I$, then the Broyden class method is identical to preconditioned conjugate gradient method with the preconditioner B_0 .
- The theorem is mainly of theoretical interest as the inexact line search used in practice significantly alters the performance of the methods. This type of analysis however, guided much of the development in quasi-Newton methods.

Global convergence

For general nonlinear objective function, there is no global convergence result for quasi-Newton methods i.e. convergence to a stationary point from any starting point and any suitable Hessian approximation.

Theorem: [BFGS global convergence]

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable and x_0 be a starting point for which the level set $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is convex and there exist two positive constants m, M such that

$$m\|z\|^2 \leq z^T \nabla^2 f(x) z \leq M\|z\|^2, \quad \forall z \in \mathbb{R}^n, x \in \mathcal{L}.$$

Then for any symmetric positive definite matrix B_0 the sequence $\{x_k\}$ generated by BFGS algorithm (with $\varepsilon = 0$) converges to the unique minimizer x^* of f in \mathcal{L} , ($\nabla^2 f(x)$ is s.p.d. in \mathcal{L}).

This results can be generalised to the restricted Broyden class with $\tau_k \in [0, 1]$ i.e. except for DFP method.

Theorem: Superlinear local convergence of BFGS

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable and the sequence of iterates generated by BFGS algorithm converge to $x^* \in \mathbb{R}^n$ such that the Hessian $\nabla^2 f$ is Lipschitz continuous at x^*

$$\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq L \|x - x^*\|, \quad \forall x \in \mathcal{N}(x^*), \quad 0 < L < \infty,$$

and that it holds

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty,$$

then x_k converges to x^* at a superlinear rate.

Theorem: SR-1 trust region convergence

Let $\{x_k\}$ be the sequence of iterates generated by the SR-1 trust region method. Suppose the following conditions hold:

- the sequence $\{x_k\}$ does not terminate, but remains in a closed bounded convex set D on which f is twice continuously differentiable and in which f has a unique stationary point x^* ;
- $\nabla^2 f(x^*)$ is positive definite and $\nabla^2 f(x)$ is Lipschitz continuous in $\mathcal{N}(x^*)$;
- the sequence $\{B_k\}$ is bounded in norm;
- $|(y_k - B_k s_k)^T s_k| \geq r \|s_k\| \|y_k - B_k s_k\|$, $r \in (0, 1)$, $\forall k$.

Then for the sequence $\{x_k\}$ we have $\lim_{k \rightarrow \infty} x_k = x^*$ and

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+n+1} - x^*\|}{\|x_k - x^*\|} = 0 \quad (n+1\text{-step superlinear rate}).$$

Remarks:

- SR-1 update does not maintain positive definiteness of B_k . In practice B_k can be indefinite at any iteration (trust region bound may continue to be active for arbitrarily large k) but it can be shown that (asymptotically) B_k remains positive definite most of the time regardless whether the initial approximation B_0 was positive definite or not.
- The theorem does not require exact solution of the trust region subproblem.

Numerical Optimisation: Large scale methods

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 9

Issues arising from large scale

- Hessian solve: Line search and trust region methods require factorisation of the Hessian. For large scale it is infeasible and has to be performed using large scale techniques such as sparse factorisations or iterative methods.
- Hessian computation and storage: Hessian approximations generated in quasi-Newton methods are usually dense even if the true Hessian is sparse. Limited-memory variants have been developed, where the Hessian approximation can be stored using only few vectors (slow convergence).
Approximated Hessians preserving sparsity.
- Special structure properties of the objective function like *partial separability* i.e. the function can be decomposed into a sum of simpler functions each depending only on a small subspace of \mathbb{R}^n .

Inexact Newton methods

Solve the Newton step system

$$\underbrace{\nabla^2 f_k}_{:=A} p_k^{iN} = -\underbrace{\nabla f_k}_b$$

using iterative method: CG or Lanczos with a modification to handle negative curvature.

Implementation can be done matrix free i.e. the Hessian does not need to be calculated or stored explicitly, we only require a routine which executes the Hessian matrix vector product.

Question: How does the inexact solve impact on the local convergence of the Newton methods?

Most of the termination rules for iterative methods are based on the residual

$$r_k = \nabla^2 f_k p_k^{\text{IN}} + \nabla f_k,$$

where p_k^{IN} is the inexact Newton step.

Usually we terminate CG when

$$\|r_k\| \leq \eta_k \|\nabla f_k\|, \quad (\text{IN-STOP})$$

where $\{\eta_k\}$ is some sequence $0 < \eta_k < 1$.

For the moment we assume that step of length $\alpha_k = 1$ is taken i.e. globalisation strategies do not interfere with the inexact-Newton step.

Theorem: local convergence

Suppose $\nabla^2 f(x)$ exists and is continuous in the neighbourhood of a minimiser x^* , with $\nabla^2 f(x^*)$ positive definite.

Consider the inexact Newton iteration with step length $\alpha_k = 1$, $x_{k+1} = x_k + p_k$, with a starting point x_0 sufficiently close to x^* , terminated with the stopping (iN-STOP) with $\eta_k \leq \eta$ for some constant $\eta \in [0, 1)$.

Then the sequence $\{x_k\}$ converges to x^* and satisfies

$$\|\nabla^2 f(x^*)(x_{k+1} - x^*)\| \leq \hat{\eta} \|\nabla^2 f(x^*)(x_k - x^*)\|,$$

for some constant $\hat{\eta}$: $\eta < \hat{\eta} < 1$.

Remark: This result provides convergence for $\{\eta_k\}$ bounded away from 1.

Proof idea convergence (superlinear):

Continuity of $\nabla^2 f(x)$ in a neighbourhood $\mathcal{N}(x^*)$ of x^* implies

$$\nabla f(x_k) = \nabla^2 f(x^*)(x_k - x^*) + o(\|x_k - x^*\|),$$

thus show instead $\|\nabla f(x_{k+1})\| \leq \hat{\eta} \|\nabla f(x_k)\|$.

Continuity and positive definiteness of $\nabla^2 f(x)$ in $\mathcal{N}(x^*)$ implies
 $\exists L \in \mathbb{R} > 0 : \|\nabla^2 f(x_k)^{-1}\| \leq L, \forall x_k \in \mathcal{N}(x^*)$ and hence

$$\|p_k\| \leq L(\|\nabla f(x_k)\| + \|r_k\|) \leq 2L\|\nabla f(x_k)\|.$$

From Taylor theorem and continuity of $\nabla^2 f(x)$ in $\mathcal{N}(x^*)$ we have

$$\begin{aligned}\nabla f(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k)p_k + \int_0^1 [\nabla^2 f(x_k + tp_k) - \nabla^2 f(x_k)]p_k dt \\ &= \nabla f(x_k) + \nabla^2 f(x_k)p_k + o(\|p_k\|) \\ &= \nabla f(x_k) - (\nabla f(x_k) - r_k) + o(\|\nabla f(x_k)\|) = r_k + o(\|\nabla f(x_k)\|)\end{aligned}$$

$$\|\nabla f(x_{k+1})\| \leq \eta_k \|\nabla f(x_k)\| + o(\|\nabla f(x_k)\|) \leq (\eta_k + o(1))\|\nabla f(x_k)\|$$

with $\eta_k = O(\sqrt{\|\nabla f(x_k)\|})$, $\leq o(\|\nabla f(x_k)\|)$.

Proof idea convergence (quadratic):

Continuity of $\nabla^2 f(x)$ in a neighbourhood $\mathcal{N}(x^*)$ of x^* implies

$$\nabla f(x_k) = \nabla^2 f(x^*)(x_k - x^*) + o(\|x_k - x^*\|),$$

thus show instead $\|\nabla f(x_{k+1})\| \leq \hat{\eta} \|\nabla f(x_k)\|$.

Continuity and positive definiteness of $\nabla^2 f(x)$ in $\mathcal{N}(x^*)$ implies
 $\exists L \in \mathbb{R} > 0 : \|\nabla^2 f(x_k)^{-1}\| \leq L, \forall x_k \in \mathcal{N}(x^*)$ and hence

$$\|p_k\| \leq L(\|\nabla f(x_k)\| + \|r_k\|) \leq 2L\|\nabla f(x_k)\|.$$

From Taylor theorem and Lipschitz continuity of $\nabla^2 f(x)$ in $\mathcal{N}(x^*)$

$$\begin{aligned}\nabla f(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k)p_k + \int_0^1 [\nabla^2 f(x_k + tp_k) - \nabla^2 f(x_k)]p_k dt \\ &= \nabla f(x_k) + \nabla^2 f(x_k)p_k + \mathcal{O}(\|p_k\|^2) \\ &= \nabla f(x_k) - (\nabla f(x_k) - r_k) + \mathcal{O}(\|\nabla f(x_k)\|^2) = r_k + \mathcal{O}(\|\nabla f(x_k)\|^2) \\ \text{with } \eta_k &= \mathcal{O}(\|\nabla f(x_k)\|)\end{aligned}$$

$$\|\nabla f(x_{k+1})\| \leq \eta_k \|\nabla f(x_k)\| + \mathcal{O}(\|\nabla f(x_k)\|^2) \leq \mathcal{O}(\|\nabla f(x_k)\|^2).$$

Theorem: superlinear (quadratic) convergence

Suppose $\nabla^2 f(x)$ exists and is continuous in the neighbourhood of a minimiser x^* , with $\nabla^2 f(x^*)$ positive definite.

Let the sequence $\{x_k\}$ generated by the inexact Newton iteration with step length $\alpha_k = 1$, $x_{k+1} = x_k + p_k$ with stopping (iN-STOP) and $\eta_k \leq \eta$ for some constant $\eta \in [0, 1)$ and a starting point x_0 sufficiently close to x^* , converge to x^* .

Then the rate of convergence is superlinear if $\eta_k \rightarrow 0$.

If in addition $\nabla^2 f(x)$ is Lipschitz continuous for $x \in \mathcal{N}(x^*)$ and $\eta_k = \mathcal{O}(\|\nabla f_k\|)$, then the convergence is quadratic.

Remark: To obtain superlinear convergence we can set e.g. $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$. The choice $\eta_k = \min(0.5, \|\nabla f_k\|)$ would yield quadratic convergence.

Line search Newton CG

Also called *truncated Newton method*. The key differences to standard Newton line search method:

- Solve the Newton step with CG with initial guess 0 and the termination criterium (iN-STOP) with the suitable choice of η_k , e.g. $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$ for superlinear convergence. Note, that if we are close enough to the solution the stopping tolerance decreases in each outer (line search) iteration.
- The inner CG iteration can be preconditioned.
- Away from the solution x^* the Hessian may not be positive definite. Therefore, we terminate CG whenever a direction of non-positive curvature is generated $d_j^T \nabla^2 f_k d_j \leq 0$. This guarantees that the produced search direction is a descent direction and preserves the fast pure Newton convergence rate provided $\alpha_k = 1$ is used whenever it satisfies the acceptance criteria.

Weakness: Performance when Hessian is nearly singular.

Algorithm 7.1 (Line Search Newton–CG).

Given initial point x_0 ;

for $k = 0, 1, 2, \dots$

 Define tolerance $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|}) \|\nabla f_k\|$;

 Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;

for $j = 0, 1, 2, \dots$

 if $d_j^T B_k d_j \leq 0$

 if $j = 0$

return $p_k = -\nabla f_k$;

 else

return $p_k = z_j$;

 Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;

 Set $z_{j+1} = z_j + \alpha_j d_j$;

 Set $r_{j+1} = r_j + \alpha_j B_k d_j$;

 if $\|r_{j+1}\| < \epsilon_k$

return $p_k = z_{j+1}$;

 Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;

 Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;

end (for)

 Set $x_{k+1} = x_k + \alpha_k p_k$, where α_k satisfies the Wolfe, Goldstein, or
 Armijo backtracking conditions (using $\alpha_k = 1$ if possible);

end

Trust region Newton CG

Use a special CG variant to solve the quadratic trust region model problem

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p, \quad \text{subject to } \|p\| \leq \Delta_k.$$

Modifications:

- Use the termination criterium (iN-STOP) as in line search variant with a suitable choice of η_k ,
e.g. $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$ for superlinear convergence.
- If CG generates direction of non-positive curvature
i.e. $d_j^T \nabla^2 f_k d_j \leq 0$, stop and return $p_k = z_j + \tau d_j$ which minimises $m_k(p_k)$ along d_j and satisfies $\|p_k\| = \Delta_k$.
- If the current iterate violates the trust region constraint
i.e. $\|z_{j+1}\| \geq \Delta_k$, stop and return $p_k = z_j + \tau d_j$, $\tau \geq 0$ which satisfies $\|p_k\| = \Delta_k$.

Algorithm 7.2 (CG-Steihaug).

Given tolerance $\epsilon_k > 0$;

Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;

if $\|r_0\| < \epsilon_k$

return $p_k = z_0 = 0$;

for $j = 0, 1, 2, \dots$

if $d_j^T B_k d_j \leq 0$

Find τ such that $p_k = z_j + \tau d_j$ minimizes $m_k(p_k)$ in (4.5)

and satisfies $\|p_k\| = \Delta_k$;

return p_k ;

Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;

Set $z_{j+1} = z_j + \alpha_j d_j$;

if $\|z_{j+1}\| \geq \Delta_k$

Find $\tau \geq 0$ such that $p_k = z_j + \tau d_j$ satisfies $\|p_k\| = \Delta_k$;

return p_k ;

Set $r_{j+1} = r_j + \alpha_j B_k d_j$;

if $\|r_{j+1}\| < \epsilon_k$

return $p_k = z_{j+1}$;

Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;

Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;

end (for).

The initialisation $z_0 = 0$ is crucial:

- Whenever $\|r_k\| \geq \varepsilon_k$, the algorithm terminates at a point p_k for which $m_k(p_k) \leq m_k(p_k^C)$ that is when the reduction in the model is at least that of the Cauchy point.
 - If $d_0^T B_k d_0 = \nabla f_0^T B_0 \nabla f_0 \leq 0$, the first **if** is activated and the algorithm returns the Cauchy point $p = -(\Delta_0 / \|\nabla f_0\|) \nabla f_0$.
 - Otherwise, the algorithm defines

$$z_1 = \alpha_0 d_0 = \frac{r_0^T r_0}{d_0^T B_k d_0} d_0 = -\frac{\nabla f_0^T \nabla f_0}{\nabla f_0^T B_0 \nabla f_0} \nabla f_0.$$

- If $\|z_1\| < \Delta_0$ then z_1 is exactly the Cauchy point. Subsequent steps ensure that the final p_k satisfies $m_k(p_k) \leq m_k(z_1)$.
- When $\|z_1\| \geq \Delta_0$, the second **if** is activated and the algorithm terminates at the Cauchy point.

Therefore, it is **globally convergent**.

- $\|z_{k+1}\| > \|z_k\| > \dots > \|z_1\|$ as a consequence of the initialisation $z_0 = 0$. Thus we can stop as soon as the boundary of trust region has been reached, because no further iterates giving a lower value of m_k will lie inside the trust region.

- Preconditioning can be used, but requires change of trust region definition, which can be reformulated in the standard form in terms of a variable $\hat{p} = Dp$ and modified $\hat{g}_k = D^{-T}\nabla f_k$ and $\hat{B}_k = D^{-T}(\nabla^2 f_k)D^{-1}$. Of particular interest is incomplete Cholesky factorisation (Algorithm 7.3 in Nocedal and Wright).
- The limitation of the algorithm is that it accepts any direction of negative curvature, even if this direction gives insignificant reduction in the model. To improve performance, CG can be replaced by Lanczos method (which can be seen as generalisation of CG which works for indefinite system albeit is more computationally expensive) for which techniques from exact trust region can be applied to compute a direction to quickly move away from stationary points which are not minimisers.

Limited memory quasi-Newton methods

Recall the BFGS formula

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k} \quad (\text{BFGS})$$

with $s_k = x_{k+1} - x_k = \alpha_k p_k$, $y_k = \nabla f_{k+1} - \nabla f_k$. Application of BFGS Hessian approximation can be efficiently implemented storing the list of vector pairs (s_k, y_k) .

The limited memory version can be obtained by restricting the total number of vectors used to construct the Hessian approximation to the last $m \ll n$. After the m th update, the oldest pair in the list makes space for the new pair.

Same strategy can be applied to the other quasi-Newton schemes (including updating B_k for use with e.g. trust region methods rather than line search methods which require H_k).

Application: large, non-sparse Hessians.

Convergence: often linear convergence rate.

Theoretical connection to CG methods

Consider the *memoryless* BFGS

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k} \right) \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right) + \frac{s_k s_k^T}{y_k^T s_k}$$

i.e. the previous Hessian is reset to identity, $H_k = I$.

If the memoryless BFGS is applied in conjunction with an **exact line search** i.e. $p_k^T \nabla f_{k+1} = 0$ for all k , we then obtain

$$p_{k+1} = -H_{k+1} \nabla f_{k+1} = -\nabla f_{k+1} + \frac{y_k^T \nabla f_{k+1}}{y_k^T p_k} p_k,$$

which is exactly the Hestens-Stiefel formula, which reduces to Polak-Ribiere when $\nabla f_{k+1}^T p_k = 0$

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{p_k^T (\nabla f_{k+1} - \nabla f_k)}, \quad \beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}.$$

Compact representation of BFGS update

Let B_0 be symmetric positive definite and assume that the vector pairs $\{s_i, y_i\}_{i=0}^{k-1}$ satisfy $s_i^T y_i > 0$. Applying k BFGS updates with these vector pairs to B_0 yields

$$B_k = B_0 - \begin{bmatrix} B_0 S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k^T B_0 S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^T B_0 \\ Y_k^T \end{bmatrix}$$

where S_k and Y_k are the $n \times k$ matrices defined by

$$S_k = [s_0, \dots, s_{k-1}], \quad Y_k = [y_0, \dots, y_{k-1}],$$

while L_k and D_k are the $k \times k$ matrices

$$(L_k)_{i,j} = \begin{cases} s_{i-1}^T y_{j-1} & \text{if } i > j, \\ 0 & \text{otherwise,} \end{cases}$$

$$D_k = \text{diag}[s_0^T y_0, \dots, s_{k-1}^T y_{k-1}].$$

- In limited memory version we replace the columns or diagonal entries in the matrices cyclically (keeping m last columns).
- Since the dimension of the middle matrix is small, the factorisation cost is negligible.
- Cost of an update: $2mn + \mathcal{O}(m^3)$
- Cost of $B_k v$: $(4m + 1)n + \mathcal{O}(m^3)$, (for $B_0 = \delta_k I$)
- This approximation can be used in trust region methods for unconstrained problems, but also in methods for constrained optimisation.
- Similar compact representation can be derived for H_k
- Compact representation can also be derived for SR-1

$$B_k = B_0 + (Y_k - B_0 S_k)(D_k + L_k + L_k^T - S_k^T B_0 S_k)^{-1} (Y_k - B_0 S_k)^T$$

with S_k, Y_k, D_k, L_k as before. The inverse formula for H_k can be obtained by swapping $B \leftrightarrow H, s \leftrightarrow y$, however limited memory SR-1 can be less effective than BFGS.

Sparse quasi-Newton updates

We require the quasi-Newton approximation to the Hessian B_k to have the same (or similar) sparsity pattern as the true Hessian. Suppose that we know which components of the Hessian are non-zero

$$\Omega = \{(i, j) : [\nabla^2 f(x)]_{ij} \neq 0 \text{ for some point } x \text{ in the domain of } f\},$$

and suppose that the current approximation B_k mirrors this sparsity structure. Such sparse update can be obtained as a solution of the following quadratic program

$$\min_B \|B - B_k\|_F^2 = \sum_{(i,j) \in \Omega} [B_{ij} - (B_k)_{ij}]^2,$$

$$\text{subject to } Bs_k = y_k, B = B^T, B_{ij} = 0 \forall (i,j) \notin \Omega.$$

It can be shown that the solution of this problem can be obtained solving an $n \times n$ linear system with sparsity pattern Ω . B_{k+1} is not guaranteed to be positive definite. The new B_{k+1} can be used within a trust region.

Unfortunately, this approach has several drawbacks, it is not scale invariant under linear transformations and the performance is disappointing. The fundamental weakness is that the closeness in Frobenius norm is an inadequate model and the produced approximations can be poor.

An alternative approach is to relax the secant equation making sure that it is approximately satisfied at the m last steps (as opposed to holding strictly in the last step) and solve

$$\begin{aligned} \min_B \quad & \|BS_k - Y_k\|_F^2, \\ \text{subject to} \quad & B = B^T, \quad B_{ij} = 0 \quad \forall (i,j) \notin \Omega, \end{aligned}$$

with S_k, Y_k containing the last m of s_i, y_i , respectively.

This convex optimisation problem has a solution but it is not easy to compute. Furthermore, it can produce singular and poorly conditioned Hessian approximations. Even though it frequently outperforms the previous approach, its performance is still not impressive for large scale problems.

Partially separable functions

An unconstrained optimisation problem is **separable** if the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ can be decomposed in a sum of independent functions e.g.

$$f(x) = f_1(x_1, x_3) + f_2(x_2, x_4, x_6) + f_3(x_5).$$

The optimal value can be found optimising each function independently, which is in general much less expensive.

In many large scale problems the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is not separable but it still can be written as a sum of simpler component functions. Each such component has the property that it only changes in a small number of directions while for other directions is remains constant. We call such functions **partially separable**.

All functions which have a sparse Hessian are partially separable, but there are many partially separable functions with dense Hessians. Partial separability allows for economical representation and effective quasi-Newton updating.

Consider an objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$f(x) = \sum_{i=1}^{\ell} f_i(x),$$

where each f_i depends only on a few components of x . For such f_i , its gradient and Hessian contain only few non-zeros.

For the function f we have by linearity of differentiation

$$\nabla f(x) = \sum_{i=1}^{\ell} \nabla f_i(x), \quad \nabla^2 f(x) = \sum_{i=1}^{\ell} \nabla^2 f_i(x)$$

thus we can maintain an quasi-Newton approximation to each individual component Hessian $\nabla^2 f_i(x)$ instead of approximating the entire Hessian $\nabla^2 f(x)$.

Example: partially separable approximation

Consider a partially separable objective function

$$f(x) = \underbrace{(x_1 - x_3^2)^2}_{f_1(x)} + \underbrace{(x_2 - x_4^2)^2}_{f_2(x)} + \underbrace{(x_3 - x_2^2)^2}_{f_3(x)} + \underbrace{(x_4 - x_1^2)^2}_{f_4(x)}.$$

Each f_i depends on two components only, all have the same form.

Denote

$$x^{[1]} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \quad U_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad x^{[1]} = U_1 x, \quad \phi_1(z_1, z_2) = (z_1 - z_2^2)^2.$$

Then $f_1(x) = \phi(U_1 x)$ and using chain rule we obtain

$$\nabla f_1(x) = U_1^T \nabla \phi_1(U_1 x), \quad \nabla^2 f_1(x) = U_1^T \nabla^2 \phi_1(U_1 x) U_1.$$

For the Hessians $\nabla^2\phi_1$ and ∇^2f_1 we have

$$\nabla^2\phi_1(U_1x) = \begin{bmatrix} 2 & -4x_3 \\ -4x_3 & 12x_3^2 - 4x_1 \end{bmatrix}, \quad \nabla^2f_1(x) = \begin{bmatrix} 2 & 0 & -4x_3 & 0 \\ 0 & 0 & 0 & 0 \\ -4x_3 & 0 & 12x_3^2 - 4x_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Idea: maintain quasi-Newton approximation $B^{[1]}$ to 2×2 Hessian $\nabla^2\phi_1$ and lift it up to ∇^2f_1 .

After a step from x_k to x_{k+1}

$$s_k^{[1]} = x_{k+1}^{[1]} - x_k^{[1]}, \quad y_k^{[1]} = \nabla\phi_1(x_{k+1}^{[1]}) - \nabla\phi_1(x_k^{[1]}),$$

and we use BFGS or SR-1 updating to obtain the new approximation $B_{k+1}^{[1]}$ of the small, dense Hessian $\nabla^2\phi_1$ and we lift it back using

$$\nabla^2f_1(x) \approx U_1^T B_{k+1}^{[1]} U_1.$$

We do the same for all component functions and we obtain

$$\nabla^2f \approx B = \sum_{i=1}^{\ell} U_i^T B^{[1]} U_i.$$

- The approximated Hessian may be used in trust region algorithm, obtaining an approximate solution to

$$B_k p_p = -\nabla f_k.$$

B_k does not need to be assembled explicitly but conjugate gradient method can be used and the products $B_k v$ can be performed directly using matrices U_i and $B^{[i]}$.

- This approach is particularly useful for large number of variables with very small dependence of component functions. Then each respective component Hessian can be much faster approximated by the iterative method (small problem requires few directions) and the so obtained full Hessian approximation is usually much better than one obtained by a quasi-Newton method applied to the problem ignoring the partially separable structure (large Hessian requires a lot of directions to approximate the curvature).

- It is not always possible for BFGS to update the partial Hessian $B^{[1]}$, as the curvature condition $(s^{[1]})^T y^{[1]} > 0$ may not be satisfied even if the full Hessian is at least positive semidefinite. This can be overcome applying SR-1 update to the component Hessians, which proved effective in practice.
- The limitation of this quasi-Newton approach is the cost of computing the step, which is comparable to the cost of Newton step, thus it may be beneficial to actually take the Newton step.
- Another problem is the difficulty of identifying the partially separable structure of a function. The performance of quasi-Newton methods is satisfactory provided that we find the *finest* partially separable decomposition.

Numerical Optimisation: Least squares

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 10 & 11

Least squares problem

Least squares is a problem where the objective function has the following special form

$$f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x),$$

where each r_j is a smooth function from $\mathbb{R}^n \rightarrow \mathbb{R}$. We refer to each of the r_j as a *residual*, and we assume that $m \geq n$. This is an overdetermined least square problem as opposed to an underdetermined problem if $m < n$).

Least squares problems are ubiquitous in applications, where the discrepancy between the model and the observed behaviour is minimised.

Let's assemble the individual components r_j into the *residual vector* $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$r(x) = (r_1(x), r_2(x), \dots, r_m(x))^T.$$

Using this vector, f becomes $f(x) = \frac{1}{2} \|r(x)\|_2^2$. The derivatives of f can be calculated with help of the Jacobian

$$J(x) = \left[\frac{\partial r_j}{\partial x_i} \right]_{ij} = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}.$$

$$\nabla f(x) = \sum_{j=1}^m r_j(x) \nabla r_j(x) = J(x)^T r(x),$$

$$\begin{aligned} \nabla^2 f(x) &= \sum_{j=1}^m \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x), \end{aligned}$$

Example

A model of a concentration of a drug in the bloodstream

$$\phi(x; t) = x_1 + tx_2 + t^2x_3 + x_4 \exp^{-x_5 t}.$$

Find a set of parameters x to that the model best matches the data solving

$$\frac{1}{2} \sum_{j=1}^m (\underbrace{\phi(x, t_j) - y_j}_{=: r_j(x)})^2.$$

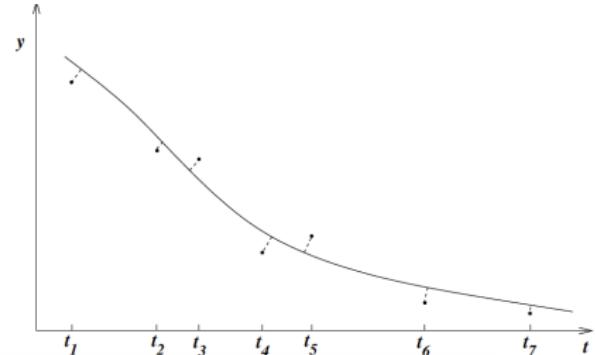
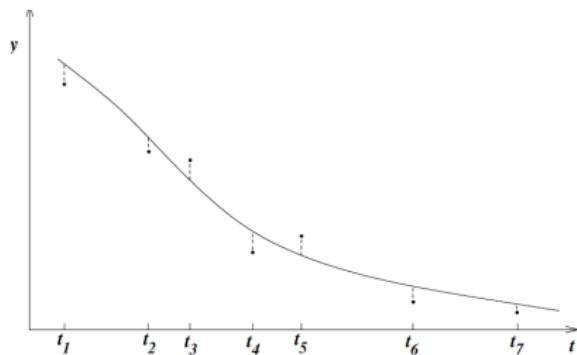


Figure: Nocedal Wright Fig 10.1 (left), Fig 10.2 (right)

Bayesian perspective

Bayes' theorem

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \propto \pi(y|x)\pi(x).$$

Denote the discrepancy between the model and the measurement at data point t_j as

$$\epsilon_j = \phi(x; t_j) - y_j.$$

Assume that ϵ_j are independent and identically distributed with variance σ^2 and probability density function $g_\sigma(\cdot)$. The *likelihood* of a particular set of observations $y_j, j = 1, 2, \dots, m$ given the parameter vector x is

$$\pi(y|x) = \prod_{j=1}^m g_\sigma(\epsilon_j) = \prod_{j=1}^m g_\sigma(\phi(x; t_j) - y_j).$$

The *maximum a posteriori probability* (MAP) estimate vs the *maximum likelihood* estimate

$$x_{\text{ML/MAP}} = \max_x \pi(x|y) = \max_x \pi(y|x)\pi(x).$$

If g_σ is a *normal* distribution

$$g_\sigma(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

and x is uniformly distributed i.e. $\pi(x) = \text{const}$, the ML equals the MAP estimate

$$\begin{aligned} x_{\text{ML/MAP}} &= \max_x \frac{1}{(\sqrt{2\pi}\sigma)^m} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^m (\phi(x; t_j) - y_j)^2\right) \\ &= \min_x \sum_{j=1}^m (\phi(x; t_j) - y_j)^2. \end{aligned}$$

Linear least squares

If $\phi(x; t)$ is linear function in x , the least squares problem becomes linear:

The residuals $r_j(x) = \phi(x; t_j) - y_j$ are linear and in the vectorized notation

$$r(x) = Jx - y,$$

where

- the vector of measurements $y = (y_1, y_2, \dots, y_m)^T = r(0)^T$
- the matrix J with rows $J_{j,:}$: $\phi(x; t_j) = J_{j,:}x$

are both independent of x .

The linear least squares has the form

$$f(x) = \frac{1}{2} \|Jx - y\|^2.$$

The gradient and Hessian are

$$\nabla f(x) = J^T(Jx - y), \quad \nabla^2 f(x) = J^T J.$$

Note: $f(x)$ is convex i.e. the stationary point is the global minimiser $\nabla f(x^*) = 0$

Normal equations

$$\nabla f(x^*) = J^T(Jx^* - y) = 0 \quad \Leftrightarrow \quad \textcolor{blue}{J^T J x^* = J^T y}.$$

Roadmap: solution of linear least squares

- Solve the normal equations $J^T J x^* = J^T y$
 - + If $m \gg n$, computing $J^T J$ explicitly results in a smaller matrix easier to store than J . This can be solved by e.g. Cholesky decomposition.
 - Formulating $J^T J$ squares the condition number.
 - + For regularised *ill-posed* problems squaring of the condition number may not be an issue.
 - If J is rank deficient (or ill conditioned) Cholesky decomposition will require pivoting.
- Solve the least squares $x^* = \arg \min_x \|Jx - b\|^2$
 - + If J is of moderate size you can use direct methods like QR decomposition or SVD decomposition.
 - + If J is large and sparse or given in operator form i.e. $x \rightarrow Jx$ use iterative methods like CGLS or LSQR.
 - + Does not square the condition number.
 - + In particular SVD and iterative methods e.g. LSQR can easily deal with ill-conditioning.

QR factorization

Let

$$JP = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \left[\underbrace{Q_1}_{\in \mathbb{R}^{m \times n}} \quad \underbrace{Q_2}_{\in \mathbb{R}^{m \times (m-n)}} \right] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R,$$

where

- $P \in \mathbb{R}^{n \times n}$ column permutation matrix (hence orthogonal),
- $Q \in \mathbb{R}^{m \times m}$ orthogonal matrix,
- $R \in \mathbb{R}^{n \times n}$ upper triangular with positive diagonal.

Recall: Multiplication with orthogonal matrix preserves $\|\cdot\|_2$.

$$\begin{aligned}\|Jx - y\|_2^2 &= \|Q^T(J \underbrace{PP^T}_{=I} x - y)\|_2^2 = \|(\underbrace{Q^T JP}_{=[R, 0]^T})P^T x - Q^T y\|_2^2 \\ &= \|RP^T x - Q_1^T y\|_2^2 + \|Q_2^T y\|_2^2.\end{aligned}$$

Solution: $x^* = PR^{-1}Q_1^T y$. In practice we perform backsubstitution on $Rz = Q_1^T y$ and permute for $x^* = Pz$.

Singular value decomposition

Let

$$J = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = \left[\underbrace{U_1}_{\in \mathbb{R}^{m \times n}} \quad \underbrace{U_2}_{\in \mathbb{R}^{m \times (m-n)}} \right] \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = U_1 S V^T,$$

where

- $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices,
- $S \in \mathbb{R}^{n \times n}$ diagonal matrix with elements $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$.

$$\begin{aligned} \|Jx - y\|_2^2 &= \|U^T(J \underbrace{VV^T}_{=I} x - y)\|_2^2 = \|(\underbrace{U^T JV}_{=[S,0]^T})V^T x - U^T y\|_2^2 \\ &= \|SV^T x - U_1^T y\|_2^2 + \|U_2^T y\|_2^2. \end{aligned}$$

Solution: $x^* = VS^{-1}U_1^T y = \sum_{i=1}^n \frac{u_i^T y}{\sigma_i} v_i$. If σ_i are small, they would undue amplify the noise and can be omitted from the sum.
Picard condition: $|u_i^T y|$ should decay faster than σ_i .

LSQR applied to

$$\min_f \|Af - g\|_2^2 + \tau \|f\|_2^2 = \min_f \left\| \begin{bmatrix} A \\ \sqrt{\tau}I \end{bmatrix} f - \begin{bmatrix} g \\ 0 \end{bmatrix} \right\|_2^2,$$

where τ is an optional damping parameter, is analytically equivalent to CG applied to the normal equations. However, it avoids forming them (hence no condition number squaring) using the Golub–Kahan bidiagonalization [Golub,Kahan '65].

G-K bidiagonalisation yields the projected least squares problem

$$\min_{y_i} \left\| \begin{bmatrix} B_i \\ \sqrt{\tau}I \end{bmatrix} y_i - \beta_1 e_1 \right\|, \quad (\text{P-LS})$$

which is then solved using QR decomposition yielding the approximation for the solution of the original problem, $f_i = V_i y_i$.

Arnoldi [Templates for the Solution of Algebraic Eigenvalue Problems]

<http://www.netlib.org/utk/people/JackDongarra/etemplates/node216.html>

The screenshot shows a web browser window with the URL <http://www.netlib.org/utk/people/JackDongarra/etemplates/node216.html>. The page title is "Basic Algorithm". On the right, there are tabs for "Lanczos Method" and "A. Ruhe". The main content area contains the following pseudocode for the Arnoldi Procedure:

```
ALGORITHM 7.32: Arnoldi Procedure
(1)    $v_1 = v / \|v\|_2$  for the starting vector  $v$ 
(2)   for  $j = 1, 2, \dots, m$  do
(3)      $w := Av_j$ 
(4)     for  $i = 1, 2, \dots, j$  do
(5)        $h_{ij} = w^* v_i$ 
(6)        $w := w - h_{ij} v_i$ 
(7)     end for
(8)      $h_{j+1,j} = \|w\|_2$ 
(9)     if  $h_{j+1,j} = 0$ , stop
(10)     $v_{j+1} = w / h_{j+1,j}$ 
(11)  end for
```

The above procedure will stop if the vector w computed in line (8) vanishes. The vectors v_1, v_2, \dots, v_m form an orthonormal system by construction and are called *Arnoldi vectors*. An easy induction argument shows that this system is a basis of the Krylov subspace $\mathcal{K}^m(A, v)$.

Next we consider a fundamental relation between quantities generated by the algorithm. The following equality is readily derived:

$$Av_j = \sum_{i=1}^{j+1} h_{ij} v_i, \quad j = 1, 2, \dots, m. \quad (122)$$

If we denote by V_m the $n \times m$ matrix with column vectors v_1, \dots, v_m , and by H_m the $m \times m$ Hessenberg matrix whose nonzero entries h_{ij} are defined by the algorithm, then the following relations hold:

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^*, \quad (123)$$

$$V_m^* A V_m = H_m. \quad (124)$$

ALGORITHM 6.16 Lanczos Method for Linear Systems

1. Compute $r_0 = b - Ax_0$, $\beta := \|r_0\|_2$, and $v_1 := r_0/\beta$
2. For $j = 1, 2, \dots, m$ Do:
 3. $w_j = Av_j - \beta_j v_{j-1}$ (If $j = 1$ set $\beta_1 v_0 \equiv 0$)
 4. $\alpha_j = (w_j, v_j)$
 5. $w_j := w_j - \alpha_j v_j$
 6. $\beta_{j+1} = \|w_j\|_2$. If $\beta_{j+1} = 0$ set $m := j$ and go to 9
 7. $v_{j+1} = w_j/\beta_{j+1}$
8. EndDo
9. Set $T_m = \text{tridiag } (\beta_i, \alpha_i, \beta_{i+1})$, and $V_m = [v_1, \dots, v_m]$.
10. Compute $y_m = T_m^{-1}(\beta e_1)$ and $x_m = x_0 + V_m y_m$

$$AV_i = V_i T_i + \beta_{i+1} v_{i+1} e_i^T$$

$$V_i^T AV_i = T_i$$

$$T_i = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{i+1} \\ & & & \beta_{i+1} & \alpha_{i+1} \end{bmatrix}, \quad V_i = [v_1, v_2, \dots, v_i], \quad V_i^T V_i = I.$$

Lanczos for NE vs. LSQR

Consider the system

$$\begin{bmatrix} I & A \\ A^T & -\tau I \end{bmatrix} \begin{bmatrix} r \\ f \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix} \quad (1)$$

which is equivalent to

$$\begin{aligned} r + Af &= g \leftrightarrow r = g - Af \\ A^T r - \tau f &= 0 \leftrightarrow A^T(Af - g) + \tau f = 0 \\ &\leftrightarrow \begin{bmatrix} A \\ \sqrt{\tau}I \end{bmatrix}^T \left(\begin{bmatrix} A \\ \sqrt{\tau}I \end{bmatrix} f - \begin{bmatrix} g \\ 0 \end{bmatrix} \right) = 0 \end{aligned}$$

When Lanczos is applied to (1), structure appears which gives raise to G-K bidiagonalization.

Golub-Kahan bidiagonalization

G-K bidiagonalization with a starting vector g for $\min_f \|g - Af\|$
(another variant starting with $A^T g$)

$$U_{i+1}(\beta_1 e_1) = g, \quad (\alpha_1 v_1 = A^T u_1)$$

$$AV_i = U_{i+1}B_i$$

$$A^T U_{i+1} = V_i B_i^T + \alpha_{i+1} v_{i+1} e_{i+1}^T,$$

e_i : i th canonical basis vector, $\alpha_i \geq 0$ and $\beta_i \geq 0$ chosen such that
 $\|u_i\| = \|v_i\| = 1$ and $U_i^T U = I$, $V_i^T V = I$,

$$B_i = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \ddots & \alpha_i \\ & & & \beta_{i+1} \end{bmatrix}, \quad U_i = [u_1, u_2, \dots, u_i], \quad V_i = [v_1, v_2, \dots, v_i].$$

Preconditioned LSQR

$$\hat{f} = \operatorname{argmin} \|g - AL^{-1}\hat{f}\|.$$

The corresponding normal equation is exactly the split preconditioned normal equation

$$L^{-T} A^T A L^{-1} \hat{f} = L^{-T} A^T g, \quad f = L^{-1} \hat{f}. \quad (2)$$

Similarly as for CG, the LSQR algorithm can be formulated without the need to provide a factorization $L^T L$ of M , the MLSQR algorithm [Arridge, B, Harhanen '14].

1: **Initialization:**

2: $\beta_1 u_1 = g$

3: $\tilde{p} = A^T u_1$

4: $\tilde{v}_1 = M^{-1} \tilde{p}, \alpha_1 = (\tilde{v}_1, \tilde{p})^{1/2}, \tilde{v}_1 = \tilde{v}_1 / \alpha_1$

5: $\tilde{w}_1 = \tilde{v}_1, f_0 = 0, \phi_1 = \beta_1, \bar{\rho}_1 = \alpha_1$

6: **for** $i = 1, 2, \dots$ **do**

7: **Bidiagonalization:**

8: $\beta_{i+1} u_{i+1} = A \tilde{v}_i - \alpha_i u_i$

9: $\tilde{p} = A^T u_{i+1} - \beta_{i+1} \tilde{p}$

10: $\tilde{v}_{i+1} = M^{-1} \tilde{p}, \alpha_{i+1} = (\tilde{v}_{i+1}, \tilde{p})^{1/2},$

11: $\tilde{p} = \tilde{p} / \alpha_{i+1}, \tilde{v}_{i+1} = \tilde{v}_{i+1} / \alpha_{i+1}$

12: **Orthogonal transformation:** smart QR, see [Paige Sanders '82]

13: **Update:**

14: $f_i = f_{i-1} + (\phi_i / \rho_i) \tilde{w}_i$

15: $\tilde{w}_{i+1} = \tilde{v}_{i+1} - (\theta_{i+1} / \rho_i) \tilde{w}_i$

16: **Break if stopping criterion satisfied**

17: **end for**

LSQR with explicit regularization ($\tau \neq 0$)

- In preconditioned formulation, Tikhonov (explicit) regularization amounts to damping. For a fixed value of τ , damping can be easily incorporated in LSQR at the cost of doubling the number of Givens rotations [Paige, Saunders '82].
- Due to the shift invariance of Krylov spaces, V_i are the same for any τ . If V_i are stored (expensive for many/long vectors), the projected least squares problem (P-LS) can be efficiently solved for multiple values of τ .
- Solving (P-LS) with a variable τ is discussed in [Bjorck '88] using singular value decomposition of the bidiagonal matrix B_i (no efficient SVD update even though B_i simply expands by a row and a column in each iteration). Those quantities can be obtained at the cost $\mathcal{O}(i^2)$ at the i^{th} iteration.
- For larger i , the algorithm described in [Elden '77] for the least squares solution of (P-LS) at the cost of $\mathcal{O}(i)$ for each value of τ is the preferable option.

Stopping LSQR / MLSQR

[Saunders Paige '82] discuss three stopping criteria:

S1: $\|\bar{r}_i\| \leq \text{BTOL}\|g\| + \text{ATOL}\|\bar{A}\|\|f_i\|$ (consistent systems),

S2: $\frac{\|\bar{A}^T \bar{r}_i\|}{\|\bar{A}\|\|\bar{r}_i\|} \leq \text{ATOL}$ (inconsistent systems),

S3: $\text{cond}(\bar{A}) \geq \text{CONLIM}$ (both),

where $\bar{r}_i := \bar{g} - \bar{A}f_i$ with $\bar{A} = \begin{bmatrix} A \\ \sqrt{\tau}L \end{bmatrix}, \quad \bar{g} = \begin{bmatrix} g \\ 0 \end{bmatrix}$.

[Arridge, B, Harhanen '14] uses Morozov discrepancy principle
(suitable for ill-posed problems)

S4: $\|r_i\| \leq \eta\delta, \quad \eta > 1,$

where $r_i := g - Af_i$, δ is the (estimated) noise level, $\eta > 1$ prevents overregularization

- + if $\tau = 0$, $r_i = \bar{r}_i$ and the sequence $\|r_i\| = \|\bar{r}_i\|$ is monotonically decreasing. Moreover if initialised with f_0 , $\|f_i\|$ is strictly monotonically growing (relevant for damped problem),
- + priorconditioning does not alter the residual.

Gauss-Newton (GN) method

Gauss-Newton (GN) can be viewed as a modified Newton method with line search.

Recall the specific form of gradient and Hessian of least squares problems

$$\nabla f(x) = J(x)^T r(x), \quad \nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x).$$

Substituting into the Newton equation

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

and using the approximation $\nabla^2 f(x) \approx J(x)^T J(x)$ we obtain

$$\underbrace{J(x_k)^T}_{=:J_k^T} J(x_k) p_k^{\text{GN}} = -J(x_k)^T \underbrace{r(x_k)}_{=:r_k}. \quad (\text{GN})$$

Implementations of GN usually perform a line search along p_k^{GN} requiring the step length to satisfy e.g. Armijo or Wolfe conditions.

- Does not require computation of the individual Hessians $\nabla^2 r_j, j = 1, \dots, m$. If the Jacobian J_k has been computed when evaluating the gradient no other derivatives are needed.
- Frequently the first term $J_k^T J_k$ dominates the second term in the Hessian i.e. $|r_j(x)|\|\nabla^2 r_j(x)\|$ are much smaller than the eigenvalues $J^T J$. This happens if either the residual r_j or its curvature $\|\nabla^2 r_j\|$ are small. Thus $J_k^T J_k$ is a good approximation to $\nabla^2 f_k$ and the convergence rate of GN is close to Newton.
- If J_k has full rank and $\nabla f_k \neq 0$, p_k^{GN} is a descent direction for f and hence suitable for line search

$$\begin{aligned}(p_k^{\text{GN}})^T \nabla f_k &= (p_k^{\text{GN}})^T J_k^T r_k = -(p_k^{\text{GN}})^T J_k^T J_k p_k^{\text{GN}} \\ &= -\|J_k p_k^{\text{GN}}\|^2 \leq 0.\end{aligned}$$

The final inequality is strict unless $J_k p_k^{\text{GN}} = 0$ in which case by J_k being full-rank we have from (GN) that $0 = J_k^T r_k = \nabla f_k$ and thus x_k is a stationary point.

Interpretation of GN step

The GN equation

$$J_k^T J_k p_k^{\text{GN}} = -J_k^T r_k$$

is exactly the normal equation for the linear least squares problem

$$\min_p \frac{1}{2} \|J_k p + r_k\|^2. \quad (\text{GN-LS})$$

Hence, we can find the GN direction p_k^{GN} solving this linear least squares problem using any of the techniques discussed before.

We can view GN equation as obtained from a linear model for the vector function $r(x_k + p) = r_k + (\nabla r)p + o(\|p\|) \approx r_k + J_k p$,

$$f(x_k + p) = \frac{1}{2} \|r_k(x_k + p)\|^2 \approx \frac{1}{2} \|J_k p + r_k\|^2$$

and $p_k^{\text{GN}} = \arg \min_p \frac{1}{2} \|J_k p + r_k\|^2$.

Global convergence of GN

The global convergence is a consequence of the convergence theorem for line search methods [Zoutendijk].

To satisfy the conditions of Zoutendijk's theorem, we need to make following assumptions:

- r_j is Lipschitz continuously differentiable in a neighbourhood \mathcal{M} of the bounded level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$,
- $J(x)$ satisfies the uniform full-rank condition, $\gamma > 0$

$$\|J(x)z\| \geq \gamma \|z\|, \quad \forall x \in \mathcal{M}.$$

Then for the iterates x_k generated by the GN method with step length satisfying Wolfe conditions, we have

$$\lim_{k \rightarrow \infty} J_k^T r_k = \nabla f(x_k) = 0.$$

Similarly as for the line search, we check that the angle $\theta_k = \angle(p_k^{\text{GN}}, -\nabla f_k)$ is uniformly bounded away from $\pi/2$

$$\cos \theta_k = -\frac{(p_k^{\text{GN}})^T \nabla f_k}{\|p_k^{\text{GN}}\| \|\nabla f_k\|} = \frac{\|J_k p_k^{\text{GN}}\|^2}{\|p_k^{\text{GN}}\| \|J_k^T J_k p_k^{\text{GN}}\|} \geq \frac{\gamma^2 \|p_k^{\text{GN}}\|^2}{\beta^2 \|p_k^{\text{GN}}\|^2} = \frac{\gamma^2}{\beta^2} > 0,$$

where the boundedness of $\|J_k(x)\| \leq \beta < \infty$, $\forall x \in \mathcal{L}$ is a consequence Lipschitz continuous differentiability of $r_j, j = 1, \dots, m$ on a closed bounded level set \mathcal{L} .

Then from $\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$ in Zoutendijk's theorem it follows $\nabla f_k \rightarrow \bar{0}$.

If J_k for some k is rank deficient, the matrix $J_k^T J_k$ in equation (GN) is singular and the system has infinitely many solutions, however $\cos \theta_k$ is not necessarily bounded away from 0.

Convergence rate GN

The convergence of GN can be rapid if $J_k^T J_k$ dominates the second term in the Hessian. Similarly as showing the convergence rate of Newton iteration, if x_k is sufficiently close to x^* , $J(x)$ satisfies the uniform full-rank condition, we have for a unit step in GN direction

$$\begin{aligned} x_k + p_k^{\text{GN}} - x^* &= x_k - x^* - [\underbrace{J^T J(x_k)}_{:= J(x_k)^T J(x_k)}]^{-1} \nabla f(x_k) \\ &= [J^T J(x_k)]^{-1} \left[J^T J(x_k)(x_k - x^*) + \underbrace{\nabla f(x^*) - \nabla f(x_k)}_{=0} \right]. \end{aligned}$$

Using $H(x)$ to denote the second term in the Hessian, it follows from Taylor theorem that

$$\begin{aligned} \nabla f(x_k) - \nabla f(x^*) &= \int_0^1 J^T J(x^* + t(x_k - x^*))(x_k - x^*) dt \\ &\quad + \int_0^1 H(x^* + t(x_k - x^*))(x_k - x^*) dt, \end{aligned}$$

Using L.c.d. of $r_j \Rightarrow$ L.c. of J and hence $J^T J$

$$\begin{aligned} \|x_k + p_k^{\text{GN}} - x^*\| &\leq \int_0^1 \|[J^T J(x_k)]^{-1} H(x^* + t(x_k - x^*))\| \|x_k - x^*\| dt \\ &+ \int_0^1 \underbrace{\|[J^T J(x_k)]^{-1}\|}_{\|\cdot\| \leq \gamma^{-2}} \underbrace{(J^T J(x^* + t(x_k - x^*)) - J^T J(x_k))\|}_{\|\cdot\| = \mathcal{O}(\|x_k - x^*\|) \text{ by L.c. of } J^T J(x)} \|x_k - x^*\| dt \\ &\approx \|[J^T J(x^*)]^{-1} H(x^*)\| \|x_k - x^*\| + \mathcal{O}(\|x_k - x^*\|^2). \end{aligned}$$

where the last step follows assuming Lipschitz continuity of $H(\cdot)$ near x^* .

Hence, if $\|[J^T J(x^*)]^{-1} H(x^*)\| \ll 1$, we can expect GN to converge quickly towards the solution x^* . When $H(x^*) = 0$, the convergence is quadratic (Newton).

When the Jacobian $J(x)$ is large and sparse, the exact solve of GN equation can be replaced by an *inexact* solve as in inexact Newton methods but with the true Hessian $\nabla^2 f(x_k)$ replaced with $J(x_k)^T J(x_k)$. The positive semidefiniteness of $J(x_k)^T J(x_k)$ simplifies the algorithms. CG on (GN) using J or $J^T J$ if J too large or LSQR on (GN-LS).

Levenberg-Marquardt (LM) method

Levenberg-Marquardt (LM) makes use of the same Hessian approximation as GN but within the framework of trust region methods. Trust region methods can cope with (nearly) rank-deficient Hessian, which is a weakness of GN.

The constraint model problem to be solved at each iteration

$$\min_p \frac{1}{2} \|J_k p + r_k\|^2, \quad \text{subject to } \|p\| \leq \Delta_k, \quad (\text{CM-LM})$$

where $\Delta_k > 0$ is the trust region radius.

Note: The least squares term corresponds to quadratic model

$$m_k(p) = \frac{1}{2} \|r_k\|^2 + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p.$$

Solution of the constraint model problem

The solution of the constraint model problem (CM-LM) is an immediate consequence of the general result for trust region methods [More, Sorensen]:

- If the solution p_k^{GN} of the GN equation lies strictly inside the trust region i.e. $\|p_k^{\text{GN}}\| < \Delta_k$, then $p_k^{\text{LM}} = p_k^{\text{GN}}$ solves (CM-LM).
- Otherwise, there is a $\lambda > 0$ such that the solution p_k^{LM} of (CM-LM) satisfies $\|p_k^{\text{LM}}\| = \Delta_k$ and

$$(J_k^T J_k + \lambda I) p_k^{\text{LM}} = -J_k^T r_k.$$

Note: The last equation is the normal equation to the linear least squares problem

$$\min_p \frac{1}{2} \left\| \begin{bmatrix} J_k \\ \sqrt{\lambda} I \end{bmatrix} p + \begin{bmatrix} r \\ 0 \end{bmatrix} \right\|^2,$$

which gives us a way of solving (CM-LM) without computing $J_k^T J_k$.

Global convergence LM

Global convergence is a consequence of the corresponding trust region global convergence theorem.

To satisfy the conditions of that theorem we make the following assumptions:

- $\eta \in (0, \frac{1}{4})$ (for strong convergence)
- $f(x) = \frac{1}{2} \|r(x)\|^2 \geq 0$ is bounded below
- r_j is Lipschitz continuously differentiable in a neighbourhood \mathcal{M} of the bounded level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$. This implies that $\exists M < \infty : \|J_k^T J_k\| \leq M$ for all k
- The approximate solution p_k of (CM-LM) satisfies

$$m_k(0) - m_k(p_k) \geq c_1 \|J_k^T r_k\| \min \left(\Delta_k, \frac{\|J_k^T r_k\|}{\|J_k^T J_k\|} \right),$$

for some constant $c_1 > 0$, and in addition $\|p_k\| \leq \gamma \Delta_k$ for some constant $\gamma \geq 1$.

We then have that

$$\lim_{k \rightarrow \infty} \nabla f_k = \lim_{k \rightarrow \infty} J_k^T r_k = 0.$$

- As for trust region methods, there is no need to evaluate the right hand side of the decrease condition, but it is sufficient to ensure reduction of at least the Cauchy point, which can be calculated inexpensively. If the iterative CG-Steighaus approach is used, this is guaranteed.
- The local convergence of LM is similar to GN. Near the solution x^* , at which the first term $J(x^*)^T J(x^*)$ of the Hessian $\nabla^2 f(x^*)$ dominates the second term, the trust region becomes inactive and the algorithm takes GN steps giving rapid local convergence.
- In large residual problems, the GN approximation to the Hessian is not adequate and convergence is slowed. Hybrid methods: GN + Newton / Quasi Newton.

References

- [Paige, Saunders '82] Link to website with papers and codes
<https://web.stanford.edu/group/SOL/software/lqr/>
- [Bjorck '96] A. Bjorck, "Numerical Methods for Least Squares Problems", SIAM, 1996
- [Golub, Kahan '65] G.H. Golub and W. Kahan, "Calculating the singular values and pseudoinverse of a matrix", *SIAM J.Numer.Anal.* 2 , 1965
- [Elden '77] L. Elden, "Algorithms for the regularization of ill-conditioned least squares problems", *BIT*, 17, 1977
- [Arridge, B, Harhanen '14] S R Arridge, M M Betcke and L Harhanen, "Iterated preconditioned LSQR method for inverse problems on unstructured grids", *Inverse Problems*, 30(7) 2014
- [Hansen '98] P. C. Hansen "Rank-Deficient and Discrete Ill-Posed Problems", SIAM, 1998

Numerical Optimisation: Constraint Optimisation

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 12

Constraint optimisation problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, & i \in \mathcal{E} \\ c_i(x) \geq 0, & i \in \mathcal{I} \end{cases} \quad (\text{COP})$$

- $f : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}$: objective function, **assume smooth**
- $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$: constraint function, **assume smooth**
 - $i \in \mathcal{E}$ equality constraints,
 - $i \in \mathcal{I}$ inequality constraints.
- $x \in \mathbb{R}^n$: optimisation variable

Feasible set Ω is a set of all points satisfying the constraints

$$\Omega = \{x \in \mathcal{D} : c_i(x) = 0, i \in \mathcal{E}; c_i(x) \geq 0, i \in \mathcal{I}\}.$$

Optimal value: $x^* = \inf_{x \in \Omega} f(x)$

- $x^* = \infty$ if (COP) is infeasible i.e. $\Omega = \emptyset$
- $x^* = -\infty$ if (COP) is unbounded below

Examples: smooth constraints

Smooth constraints can describe regions with *kinks*.

Example: 1-norm:

$$\|x\|_1 = |x_1| + |x_2| \leq 1$$

can be described as

$$x_1 + x_2 \leq 1, \quad x_1 - x_2 \leq 1, \quad -x_1 + x_2 \leq 1, \quad -x_1 - x_2 \leq 1.$$

Example: pointwise max

$$\min f(x) = \max(x^2, x)$$

can be reformulated as

$$\min t, \quad \text{s.t.} \quad t \geq x, \quad t \geq x^2.$$

Types of minimisers of constraint problems

A point $x^* \in \Omega$ is a **global minimiser** if

$$f(x^*) \leq f(x), \quad \forall x \in \Omega.$$

A point $x^* \in \Omega$ is a **local minimiser** if

$$\exists \mathcal{N}(x^*) : f(x^*) \leq f(x), \quad \forall x \in \mathcal{N}(x^*) \cap \Omega.$$

A point $x^* \in \Omega$ is a **strict (or strong) local minimiser** if

$$\exists \mathcal{N}(x^*) : f(x^*) < f(x), \quad \forall x \in \mathcal{N}(x^*) \cap \Omega, \quad x \neq x^*.$$

A point $x^* \in \Omega$ is an **isolated local minimiser** if

$$\exists \mathcal{N}(x^*) : x^* \text{ is the only local minimiser in } \mathcal{N}(x^*) \cap \Omega.$$

Feasibility problem: Find x such that all constraints are satisfied at x .

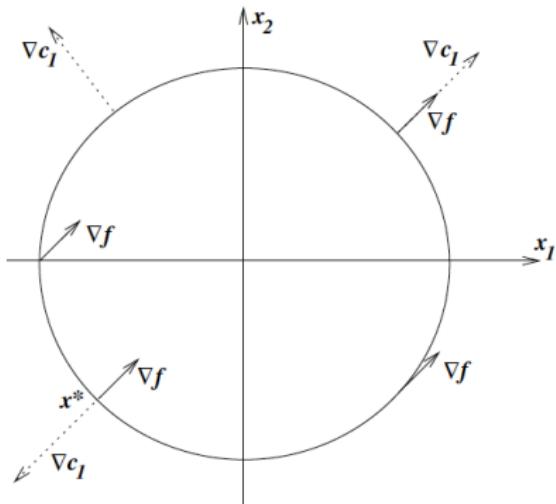
Active set $\mathcal{A}(x)$ at any feasible $x \in \Omega$ consists of the equality constraint indices set \mathcal{E} and the inequality constraints $i \in \mathcal{I}$ for which $c_i(x) = 0$

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} : c_i(x) = 0\}.$$

At a feasible point $x \in \Omega$, the inequality constraint $i \in \mathcal{I}$ is said to be *active* if $c_i(x) = 0$ and *inactive* if the strict inequality holds $c_i(x) > 0$.

Single equality constraint

$$\min x_1 + x_2 \quad \text{s.t.} \quad x_1^2 + x_2^2 - 2 = 0.$$



Feasibility:
(Taylor expansion of c_1)

$$0 = c_1(x+s) \approx \underbrace{c_1(x) + \nabla c_1(x)^T s}_{=0}$$

Decrease direction:
(Taylor expansion of f)

$$0 > f(x+s) - f(x) \approx \nabla f(x)^T s$$

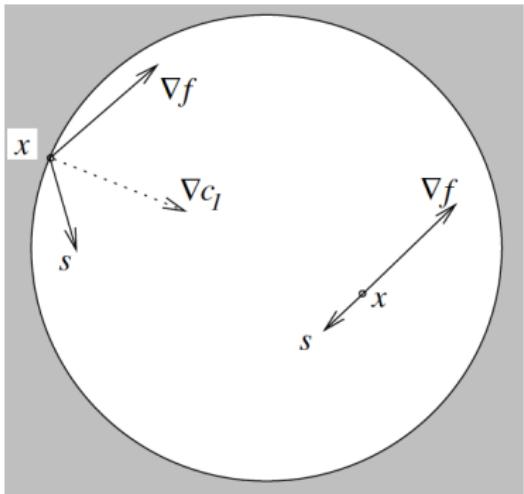
The only situation that such s does not exist is if for some scalar λ_1

$$\nabla f(x) = \lambda_1 \nabla c_1(x).$$

Note: sign of λ_1 cannot be specified: $c_1(x) = 0 \Rightarrow -c_1(x) = 0$.

Single inequality constraint

$$\min x_1 + x_2 \quad \text{s.t.} \quad 2 - x_1^2 - x_2^2 \geq 0.$$



Feasibility:
(Taylor expansion of c_1)

$$0 \leq c_1(x+s) \approx c_1(x) + \nabla c_1(x)^T s$$

Decrease direction:
(Taylor expansion of f)

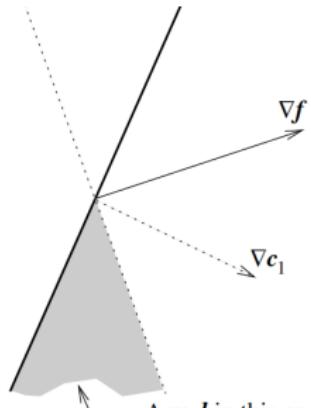
$$0 > f(x+s) - f(x) \approx \nabla f(x)^T s$$

Case: x inside the circle, i.e. $c_1(x) > 0$

$$s = -\alpha \nabla f(x), \quad \alpha > 0.$$

Single inequality constraint

$$\min x_1 + x_2 \quad \text{s.t.} \quad 2 - x_1^2 - x_2^2 \geq 0.$$



Feasibility:
(Taylor expansion of c_1)

$$0 \leq c_1(x+s) \approx \underbrace{c_1(x)}_{=0} + \nabla c_1(x)^T s$$

Decrease direction:
(Taylor expansion of f)

$$0 > f(x+s) - f(x) \approx \nabla f(x)^T s$$

Case: x on the boundary of the circle, i.e. $c_1(x) = 0$

$$\nabla f(x)^T s < 0, \quad \nabla c_1(x)^T s \geq 0$$

Empty only if $\nabla f(x) = \lambda_1 \nabla c_1(x)$ for some $\lambda_1 \geq 0$.

Given the point x in the active set $\mathcal{A}(x)$, the **linear independent constraint qualification (LICQ)** holds if the set of active constraint gradients $\{\nabla c_i(x), i \in \mathcal{A}(x)\}$ is linearly independent.

Note that for LICQ to be satisfied, none of the active constraint gradients can be 0.

Example:

LICQ is not satisfied if we define the equality constraint $c_1(x) = (x_1^2 + x_2^2 - 2)^2 = 0$ (same feasibility region, different constraint).

There are other constraint qualifications e.g. Slater's conditions for convex problems.

Theorem: 1st order necessary conditions

Lagrangian function

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x).$$

Let x^* be a local solution of (COP) and f and c_i be continuously differentiable and LICQ hold at x^* . Then there exists a **Lagrange multiplier** λ^* with components $\lambda_i^*, i \in \mathcal{E} \cup \mathcal{I}$ such that the following **Karush-Kuhn-Tucker conditions** are satisfied at (x^*, λ^*)

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \tag{KKT-\nabla L}$$

$$c_i(x^*) = 0, \quad \forall i \in \mathcal{E}, \tag{KKT-PE}$$

$$c_i(x^*) \geq 0, \quad \forall i \in \mathcal{I}, \tag{KKT-PI}$$

$$\lambda^* \geq 0, \quad \forall i \in \mathcal{I}, \tag{KKT-DI}$$

$$\lambda_i^* c_i(x^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}. \tag{KKT-CS}$$

Strong complementarity condition

The *complementarity condition* (KKT-CS) can be made stronger.

Given x^* a local solution of (COP) and a vector λ^* satisfying the KKT conditions, the **strict complementarity condition** holds if exactly one of λ_i^* and $c_i(x^*)$ is zero for each $i \in \mathcal{I}$. In other words, $\lambda^* > 0$ for each $i \in \mathcal{I} \cap \mathcal{A}(x^*)$.

Strict complementarity makes it easier for the algorithms to identify the active set and converge quickly to the solution.

For a given solution x^* of (COP), there may be many vectors λ^* which satisfy the KKT condition (2). However, if LICQ holds, the optimal λ^* is unique.

Lagrangian: primal problem

For convenience we change (and refine) our notation for the constraint optimisation problem. The following slides are based on Boyd (Convex Optimization I).

Let p^* be the optimal value of the **primal problem**

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{aligned} \tag{COP:P}$$

The **Lagrangian** $\mathcal{L} : \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$

$$\mathcal{L}(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

- λ_i are Lagrange multipliers associated with $f_i(x) \leq 0$
- ν_i are Lagrange multipliers associated with $h_i(x) = 0$

Lagrange dual function

Lagrange dual function: $g : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) && (\text{LD}) \\ &= \inf_{x \in \mathcal{D}} \left(f(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right). \end{aligned}$$

g : is concave, can be $-\infty$ for some λ, ν (inf of affine functions).

Lower bound property: If $\lambda \geq 0$, then $g(\lambda, \nu) \leq p^*$.

Proof: For any feasible \tilde{x} and $\lambda \geq 0$ we have

$$f(\tilde{x}) \geq \mathcal{L}(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) = g(\lambda, \nu).$$

Minimising over all feasible \tilde{x} gives $p^* \geq g(\lambda, \nu)$.

Convex problem in standard form

$$\min_{x \in \mathcal{D} \subset \mathbb{R}^n} f(x)$$

$$\begin{aligned} \text{subject to } & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{aligned}$$

- f is convex and \mathcal{D} is convex
- f_i are convex
- h_i are affine i.e. $a_i^T x = b_i$

Feasibility set Ω of a convex problem is a convex set.

Example: least norm solution of linear equations

$$\min_{x \in \mathbb{R}^n} x^T x$$

subject to $Ax = b$

- Lagrangian: $\mathcal{L}(x, \nu) = x^T x + \nu^T (Ax - b)$

- Dual function:

$$g(\nu) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \nu) = \inf_{x \in \mathbb{R}^n} (x^T x + \nu^T (Ax - b))$$

- $\mathcal{L}(x, \nu)$ is strictly convex in x , to minimise over x set gradient equal zero:

$$\nabla_x \mathcal{L}(x, \nu) = 2x + A^T \nu = 0 \quad \Rightarrow \quad x_{\min} = -1/2A^T \nu$$

- Plug x_{\min} into g

$$g(\nu) = \mathcal{L}(x_{\min}, \nu) = -\frac{1}{4}\nu A^T A \nu - b^T \nu.$$

g is a concave function of ν .

Lower bound property: $p^* \geq -1/4\nu A^T A \nu - b^T \nu$ for all ν .

Example: standard form LP

$$\min_{x \in \mathbb{R}^n} c^T x$$

$$\text{subject to } Ax = b, \quad x \geq 0$$

- Lagrangian:

$$\mathcal{L}(x, \nu) = c^T x + \nu^T (Ax - b) - \lambda^T x = -b^T \nu + (c + A^T \nu - \lambda)^T x$$

- Dual function:

$$g(\lambda, \nu) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) = \inf_{x \in \mathbb{R}^n} (-b^T \nu + (c + A^T \nu - \lambda)^T x)$$

- $\mathcal{L}(x, \nu)$ is affine in x , hence

$$g(\lambda, \nu) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \nu) = \begin{cases} -b^T \nu, & A^T \nu - \lambda + c = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

g is linear on affine domain $\{(\lambda, \nu) : A^T \nu - \lambda + c = 0\}$,
hence concave.

Lower bound property: $p^* \geq -b^T \nu$ if $A^T \nu + c \geq 0$.

Example: equality constraint norm minimisation

$$\min_{x \in \mathbb{R}^n} \|x\|$$

$$\text{subject to } Ax = b$$

- Lagrangian:

$$\mathcal{L}(x, \nu) = \|x\| + \nu^T(b - Ax) = \|x\| - \nu^T Ax + b^T \nu$$

- $\inf_{x \in \mathbb{R}^n} (\|x\| - y^T x) = 0$ if $\|y\|_* \leq 1$, $-\infty$ otherwise, where $\|\nu\|_* = \sup_{\|u\| \leq 1} u^T \nu$ is dual norm of $\|\cdot\|$.

Proof:

If $\|y\|_* \leq 1$: let $x = tu$, $\|u\| = 1$, $t \geq 0$ then

$$\|x\| - y^T x = t\|u\| - t(y^T u) \geq t\|u\| - t \sup_{\|u\| \leq 1} (y^T u) = t(1 - \|y\|_*) \geq 0.$$

If $\|y\|_* > 1$: choose $\tilde{x} = t\tilde{u}$, $t \geq 0$, $\|\tilde{u}\| \leq 1$ such that $\tilde{u}^T y \leq \sup_{\|u\| \leq 1} u^T y = \|y\|_* > 1$ then

$$\inf_{x \in \mathbb{R}^n} (\|x\| - y^T x) \leq t(\|\tilde{u}\| - y^T \tilde{u}) \leq t(\|\tilde{u}\| - \|y\|_*) \rightarrow -\infty \text{ as } t \rightarrow \infty$$

- Dual function:

$$g(\nu) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \nu) = \begin{cases} b^T \nu, & \|A^T \nu\|_* \leq 1 \\ -\infty, & \text{otherwise} \end{cases}$$

Lower bound property: $p^* \geq b^T \nu$ if $\|A^T \nu\|_* \leq 1$.

Conjugate function

The **conjugate** of function f is

$$f^*(y) = \sup_{x \in \mathcal{D}} (y^T x - f(x))$$

The conjugate f^* is convex (even if f is not)

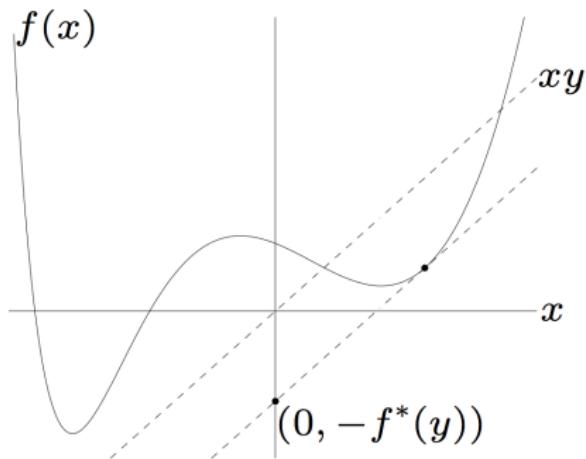


Figure: Boyd, Convex Optimization I

Lagrange dual and conjugate function

$$\min_{x \in \mathbb{R}^n} f(x)$$

subject to $Ax \leq b, \quad Cx = d$

- Lagrangian:

$$\begin{aligned}\mathcal{L}(x, \lambda, \nu) &= f(x) + \lambda^T(Ax - b) + \nu^T(Cx - d) \\ &= f(x) + (A^T\lambda + C^T\nu)^T x - b^T\lambda - d^T\nu.\end{aligned}$$

- Dual function:

$$\begin{aligned}g(\lambda, \nu) &= \inf_{x \in \mathcal{D}} (f(x) + (A^T\lambda + C^T\nu)^T x - b^T\lambda - d^T\nu) \\ &= -\sup_{x \in \mathcal{D}} (-f(x) - (A^T\lambda + C^T\nu)^T x) - b^T\lambda - d^T\nu \\ &= -f^*(-A^T\lambda - C^T\nu) - b^T\lambda - d^T\nu\end{aligned}$$

Lagrange dual problem

$$\begin{aligned} & \max \quad g(\lambda, \nu) \\ & \text{subject to} \quad \lambda \geq 0 \end{aligned}$$

- provides the best lower bound on p^* obtained from Lagrange dual function
- is a convex optimization problem, we denote its optimal value with d^*
- λ, ν are dual feasible if $\lambda \geq 0$, $(\lambda, \nu) \in \text{dom } g$
- often simplified by making implicit constraint $(\lambda, \nu) \in \text{dom } g$, explicit

Weak and strong duality

Weak duality: $d^* \leq p^*$

- always holds (for convex and nonconvex problems)
- can be used to find nontrivial lower bounds for difficult problems

Strong duality: $d^* = p^*$

- does not hold in general
- holds for convex problems under **constraint qualifications**

Slater's constraint qualification

Strong duality holds for a convex problem

$$\begin{aligned} & \min_{x \in \mathcal{D}} f(x) \\ \text{subject to } & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & Ax = b, \end{aligned}$$

if it is strictly feasible i.e.

$$\exists x \in \text{int}\mathcal{D} : \quad f_i(x) < 0, \quad i = 1, \dots, m, \quad Ax = b$$

- also guarantees that the dual optimum is attained (if $p^* > -\infty$)
- can be sharpened: e.g. can replace $\text{int}\mathcal{D}$ with $\text{relint}\mathcal{D}$ (interior of the affine hull); linear inequalities do not need to hold with strict inequality, ...
- other constraint qualifications exist e.g. LICQ

Example: inequality form LP

Primal problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \end{aligned}$$

Dual function

$$g(\lambda) = \inf_{x \in \mathbb{R}^n} ((c + A^T \lambda)^T x - b^T \lambda) = \begin{cases} -b^T \lambda, & A^T \lambda + c = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

Dual problem

$$\begin{aligned} \max \quad & -b^T \lambda \\ \text{subject to} \quad & A^T \lambda + c = 0, \quad \lambda \geq 0 \end{aligned}$$

- From Slater's condition: $p^* = d^*$ if $\exists \tilde{x} : A\tilde{x} < b$
- In fact, $p^* = d^*$ except when primal and dual are infeasible

Example: Quadratic program

Primal problem (assume P symmetric positive definite)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x^T P x \\ \text{subject to} \quad & Ax \leq b \end{aligned}$$

Dual function

$$g(\lambda) = \inf_{x \in \mathbb{R}^n} (x^T P x + \lambda^T (Ax - b)) = -\frac{1}{4} \lambda^T A P^{-1} A^T \lambda - b^T \lambda$$

Dual problem

$$\begin{aligned} \max \quad & -\frac{1}{4} \lambda^T A P^{-1} A^T \lambda - b^T \lambda \\ \text{subject to} \quad & \lambda \geq 0 \end{aligned}$$

- From Slater's condition: $p^* = d^*$ if $\exists \tilde{x} : A\tilde{x} < b$
- In fact, $p^* = d^*$ always

Example: nonconvex problem with strong duality

Primal problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x^T A x + 2b^T x \\ \text{subject to} \quad & x^T x \leq 1 \end{aligned}$$

$A \not\succeq 0$ is not positive definite.

Dual function

$$g(\lambda) = \inf_{x \in \mathbb{R}^n} (x^T (A + \lambda I)x + 2b^T x - \lambda)$$

- unbounded below if $A + \lambda I \not\succeq 0$ or if $A + \lambda I \succeq 0$ and $b \notin \mathcal{R}(A + \lambda I)$
- otherwise minimised by $x = -(A + \lambda I)^\dagger b$:
$$g(\lambda) = -b^T (A + \lambda I)^\dagger b - \lambda$$

Dual problem

$$\begin{aligned} \max \quad & -b^T(A + \lambda I)^\dagger b - \lambda \\ \text{subject to} \quad & A + \lambda I \succeq 0 \\ & b \in \mathcal{R}(A + \lambda I) \end{aligned}$$

and equivalent semidefinite program:

$$\begin{aligned} \max \quad & -t - \lambda \\ \text{subject to} \quad & \begin{bmatrix} A + \lambda I & b \\ b^T & t \end{bmatrix} \succeq 0 \end{aligned}$$

Strong duality although primal problem is not convex (not easy to show).

KKT conditions: necessity under strong duality

Karush-Kuhn-Tucker conditions are satisfied at x^*, ν^*, λ^* i.e.

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0, \quad (\text{KKT-}\nabla\text{L})$$

$$h_i(x^*) = 0, \quad i = 1, \dots, p \quad [\text{primary constraints}] \quad (\text{KKT-PE})$$

$$f_i(x^*) \leq 0, \quad i = 1, \dots, m \quad [\text{primary constraints}] \quad (\text{KKT-PI})$$

$$\lambda^* \geq 0, \quad i = 1, \dots, m \quad [\text{dual constraints}] \quad (\text{KKT-DI})$$

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, \dots, m \quad [\text{complementary slackness}] \quad (\text{KKT-CS})$$

If $p^* = f(x^*)$, $d^* = g(\lambda^*, \nu^*)$ are attained and $p^* = d^*$ i.e. strong duality holds, KKT are **necessary conditions**

$$\begin{aligned} f(x^*) &= g(\lambda^*, \nu^*) = \inf_{x \in \mathbb{R}^n} f(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \\ &\leq f(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*) \underbrace{\leq}_{x^* \in \Omega} f(x^*) \end{aligned}$$

Thus " \leq " is in fact " $=$ " and " \inf " is " \min " i.e. attained by x^* .

KKT conditions: sufficiency for convex problems

KKT are sufficient conditions for convex problems: If there exist x^* and ν^*, λ^* that satisfy KKT conditions, then x^* and ν^*, λ^* are primal and dual optimal with zero duality gap.

- from primary constraints and complementary slackness

$$f(x^*) = f(x^*) + \sum_{i=1}^m \underbrace{\lambda_i^* f_i(x^*)}_{=0, \text{ (KKT-CS)}} + \sum_{i=1}^p \nu_i^* \underbrace{h_i(x^*)}_{=0, \text{ (KKT-PE)}} = \mathcal{L}(x^*, \lambda^*, \nu^*)$$

- by definition $g(\lambda^*, \nu^*) = \inf_x \mathcal{L}(x, \lambda^*, \nu^*)$. From (KKT- ∇L) (1st oder necessary condition for Lagrangian wrt. x) and convexity we have that the minimum is attained at x^* , hence $g(\lambda^*, \nu^*) = \mathcal{L}(x^*, \lambda^*, \nu^*)$

Thus it follows that $f(x^*) = g(\lambda^*, \nu^*)$.

Nec. & suff. : If Slater's conditions are satisfied, then strong duality holds and the dual optimum is attained i.e. x^* is optimal iff there exists λ^*, ν^* such that x^* and λ^*, ν^* satisfy KKT conditions.

Numerical Optimisation: Solution with equality constraints

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 13 (based on Boyd, Vanderberghe)

Equality constraint optimisation

$$\begin{aligned} & \min f(x) \\ \text{subject to } & Ax = b \end{aligned} \tag{CCOP:E}$$

where $f : \mathcal{D} \rightarrow \mathbb{R}$ is convex and twice continuously differentiable,
 $A \in \mathbb{R}^{p \times n}$ with $\text{rank } A = p < n$.

$x^* \in \mathcal{D}$ is optimal for (CCOP:E) iff $\exists \nu^* \in \mathbb{R}^p$ such that

$$Ax^* = b \quad \text{primal feasibility (linear)} \tag{CCOP:E:PF}$$

$$\nabla f(x^*) + A^T \nu^* = 0 \quad \text{dual feasibility (in general nonlinear)} \tag{CCOP:E:DF}$$

Thus solving the equality constraint optimisation problem
(CCOP:E) is equivalent to solving the KKT equations.

Note: if x^* is feasible, we have strong duality by Slater conditions.

Quadratic problem with equality constraints

$$\max \quad \frac{1}{2} x^T P x + q^T x + r,$$

subject to $Ax = b$

where P is positive semidefinite, $A \in \mathbb{R}^{p \times n}$.

$x^* \in \mathbb{R}^n$ is optimal iff $\exists \nu^* \in \mathbb{R}^p : Ax^* = b, Px^* + q + A^T \nu^* = 0$.

KKT system:
$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \nu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

- if KKT matrix is non-singular \rightarrow unique solution
- if KKT matrix is singular, either
 - infinitely many solutions (each yields an optimal pair) or
 - not solvable (unbounded or infeasible)

Equivalent conditions to nonsingularity of KKT matrix:

- $\text{rank } A = p < n$
- $\text{Null}(P) \cap \text{Null}(A) = \{0\}$
- $Ax = 0, x \neq 0 \Rightarrow x^T P x > 0$

Eliminating equality constraints

Since $A \in \mathbb{R}^{p \times n}$, it has a null space of dimension $n \times (n - p)$. Find a basis for this null space, $N \in \mathbb{R}^{n \times (n-p)}$ (e.g. swapping columns) and rewrite $x = Nz + \hat{x}$, where $z \in \mathbb{R}^{n-p}$ and $\hat{x} \in \mathbb{R}^n$ is any particular solution $\hat{x} \in \mathbb{R}^n : A\hat{x} = b$.

Solve the resulting unconstraint problem

$$\min_{z \in \mathbb{R}^{n-p}} f(Nz + \hat{x}).$$

From solution z^* recover $x^* = Nz^* + \hat{x}$.

You can construct the optimal dual $\nu^* \in \mathbb{R}^p$, $p < n$ solving the overdetermined system (CCOP:E:DF) (dual feasibility)

$$\nu^* = -(AA^T)^{-1}A\nabla f(x^*).$$

Note, that the solvability of this system i.e. $\nabla f(x^*) \in \mathcal{R}(A^T)$ is one of the KKT conditions. Hence, (CCOP:E:DF) can be solved projecting on range of $\mathcal{R}(A^T)$.

Solve via dual

We assume that x^* is primal optimal. Hence Slater conditions hold which in turn imply that strong duality holds and the dual optimum is attained i.e.

$$\exists \nu^* \in \mathbb{R}^P : g(\nu^*) = \max_{\nu} g(\nu) = p^* = f(x^*).$$

Solve the dual problem (if easy, potentially unconstraint but for boundedness of Lagrangian function in x)

$$\nu^* = \max_{\nu} g(\nu) = \max_{\nu} -b^T \nu - f^*(-A^T \nu).$$

Let the minimiser of the Lagrangian wrt. x be unique

$$\min_x \mathcal{L}(x, \nu^*) = \min_x f(x) + (Ax - b)^T \nu^*.$$

Then if it is feasible then it is primal optimal, otherwise no primal solution exists. Usage: if $\min_x \mathcal{L}(x, \nu^*)$ is simple to compute e.g. $\mathcal{L}(x, \nu^*)$ is strictly convex function of x .

Feasible Newton method

Newton method which starts at a feasible point and subsequently enforces the equality constraints on the step maintaining feasibility.

Interpretations:

- Δx_n minimises the second order Taylor polynomial:

$$f(x+v) \approx f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v \quad \text{s.t. } A(x+v) = b.$$

- Linearise optimality conditions:

$$A(x + \Delta x_n) = b,$$

$$\nabla f(x + \Delta x_n) + A^T w \approx \nabla f(x) + \nabla^2 f(x) \Delta x_n + A^T w = 0.$$

Using $Ax = b$ both simplify to

$$A\Delta x_n = 0, \quad \nabla^2 f(x)\Delta x_n + A^T w = -\nabla f(x),$$

compactly written as a quadratic constraint problem (solution defined if KKT matrix is non-singular)

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_n \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix} \quad (\text{KKT:FN})$$

Newton decrement

Newton decrement for the step Δx_n (obtained from (KKT:FN))

$$\lambda(x) = (\Delta x_n^T \nabla^2 f(x) \Delta x_n)^{1/2} = (-\nabla f(x)^T \Delta x_n)^{1/2}.$$

Without constraints, Δx_n solves Newton equation (up to the line search). With constraints it is the difference between $f(x)$ and the minimum of the second order model at x

$$\begin{aligned} f(x) - \inf_{A(x+v)=b} & \left\{ f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v \right\} \\ &= -\nabla f(x)^T \Delta x_n - \frac{1}{2} \Delta x_n^T \nabla^2 f(x) \Delta x_n = \lambda(x)^2 / 2, \end{aligned}$$

i.e. $\lambda(x)^2 / 2$ is an estimate for $f(x) - p^*$ (based on a quadratic model) and hence a good stopping criterion.

Furthermore it holds,

$$\left. \frac{d}{dt} f(x + t \Delta x_n) \right|_{t=0} = \nabla f(x)^T \Delta x_n = -\Delta x_n^T \nabla^2 f(x) \Delta x_n = -\lambda(x)^2.$$

One of the consequences is that Δx_n is a descent direction.

Convergence

It can be shown that Newton with equality constraints is equivalent to applying Newton to reduced problem obtained by eliminating the equality constraints.

Hence the convergence theory for unconstrained problems applies.

The assumption on the eigenvalues of the Hessian being bounded away from 0, needs to be replaced by the requirement that the absolute values of the eigenvalues of the indefinite KKT matrix are bounded away from 0.

Infeasible Newton

Starts at any point $x \in \mathcal{D}$ (not necessarily feasible i.e. in general $Ax \neq b$). Compute step approximately satisfying the optimality conditions $x + \Delta x \approx x^*$.

Of interest if $\mathcal{D} \neq \mathbb{R}^n$. If $\mathcal{D} = \mathbb{R}^n$ then the feasible point can be simply computed solving $Ax = b$, otherwise it may be easier to start with infeasible method.

For inequality constraints (after reformulating into equality constraint problems through e.g. implicit constraints): it is an alternative to phase I methods, but in contrast to phase I methods it will not detect that no strictly feasible point exists.

Substituting into optimality conditions we obtain

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_n \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ b - Ax \end{bmatrix} \quad (\text{KKT:IFN})$$

$Ax - b$ is the residual, which reduces to 0 when x is feasible.

Interpretation as a primal-dual method

Define the residual

$$r(y) = r(\underbrace{x, \nu}_y) = (r_d(x, \nu), r_p(x, \nu)) = (\underbrace{\nabla f(x) + A^T \nu}_{=r_d}, \underbrace{Ax - b}_{=r_p})$$

First order Taylor approximation of r

$$r(y + z) \approx r(y) + Dr(y)z,$$

where $Dr(y) \in \mathbb{R}^{n+p \times n+p}$ is the derivative of r .

Let the primal-dual Newton step Δy_{pd} be the step z for which the Taylor approximation vanishes (i.e. accurate for the linear model)

$$Dr(y)\Delta y_{pd} = -r(y). \quad (\text{PD})$$

Written out this reads

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{pd} \\ \Delta \nu_{pd} \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + A^T \nu \\ Ax - b \end{bmatrix} \quad (\text{KKT:PD})$$

and substituting $\nu^+ = \nu + \Delta \nu_{pd}$ we obtain

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{pd} \\ \nu^+ \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ Ax - b \end{bmatrix},$$

which is the “infeasible Newton system” with

$$\Delta x_n = \Delta x_{pd}, \quad w = \nu^+ = \nu + \Delta \nu_{pd}.$$

The Newton direction at an infeasible point is not necessarily a descent direction

$$\begin{aligned}\frac{d}{dt} f(x + t\Delta x) \Big|_{t=0} &= \nabla f(x)^T \Delta x \\ &= -\Delta x^T (\nabla^2 f(x) \Delta x + A^T w) \\ &= -\Delta x^T \nabla^2 f(x) \Delta x + (Ax - b)^T w.\end{aligned}$$

The last equation is not necessarily negative (unless x is feasible and $Ax = b$). Thus value of f not a progress indicator.

In contrast, the primal-dual residual norm in (PD) decreases in Newton direction

$$\frac{d}{dt} \|r(y + t\Delta y_{pd})\|^2 \Big|_{t=0} = 2r(y)^T Dr(y) \Delta y_{pd} = -2r(y)^T r(y) = -2\|r(y)\|^2.$$

This is equivalent to taking the derivative of $(\cdot)^2$ and multiplying with the interior derivative, hence the latter is

$$\frac{d}{dt} \|r(y + t\Delta y_{pd})\| \Big|_{t=0} = -\|r(y)\|.$$

$\|r\|$ can be used to measure progress of the infeasible Newton method e.g. in line search (instead of f in standard Newton).

By construction the Newton step has the property

$$A(x + \Delta x_n) = b.$$

Thus once a step of length 1 has been taken in the Newton direction, $x_{n+1} = x + \Delta x_n$ and all the following iterates will be feasible (becomes feasible Newton).

Effect of the damped step on the residual r_p . For the next iterate $x^+ = x + t\Delta x_n$, $t \in [0, 1]$, the primary residual

$$r_p^+ = A(x + \Delta x_n t) - b = (1 - t)(Ax - b) = (1 - t)r_p$$

is reduced by a factor $(1 - t)$. After k iterations we have $r^{(k)} = \prod_{i=0}^{k-1} (1 - t^{(i)}) r^{(0)}$, $t^{(i)} \in [0, 1]$, $i = 0, \dots, k - 1$. Thus the primal residual is in the direction $r^{(0)}$ and scaled down at each step. After a full step has been taken, $t = 1$, all future iterates are primal feasible.

Convergence very similar as for feasible Newton (in a finite number of steps the residual is reduced enough and feasibility is achieved, full steps are taken and the convergence becomes quadratic).

Numerical Optimisation

Constraint optimisation:

Penalty and augmented Lagrangian methods

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 14 (based on Nocedal, Wright)

Lagrangian: primal problem

Constraint optimization problem

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{aligned} \tag{COP}$$

Possibly conflicting goals: minimise the function and satisfy the constraints.

Idea: Minimise a merit function $Q(x; \mu)$ with a parameter vector μ . Some minimisers of $Q(x; \mu)$ approach those of f subject to the constraints as μ approach some set \mathcal{M} .

Benefit: reformulation as an unconstraint problem.

Quadratic penalty

Consider a problem with equality constraints

$$\begin{aligned} \min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned} \tag{COP:E}$$

The merit function (*quadratic penalty function*)

$$Q(x; \mu) := f(x) + \frac{\mu}{2} \sum_{i=1}^p h_i^2(x), \tag{Q}$$

where $\mu > 0$ is the *penalty parameter*.

Idea: choose a sequence $\{\mu_k\}$: $\mu_k \rightarrow \infty$ as $k \rightarrow \infty$,
i.e. increasingly penalise the constraint, and compute the sequence
 $\{x_k\}$ of (approximate) minimisers of $Q(x; \mu_k)$.

Convergence for the quadratic penalty

Let $\{x_k\}$ be the sequence of approximate minimisers of $Q(x; \mu_k)$, such that $\|\nabla_x Q(x_k; \mu_k)\| \leq \tau_k$, x^* be the limit point of $\{x_k\}$ as the sequences of the penalty parameters $\mu_k \rightarrow \infty$ and tolerances, $\tau_k \rightarrow 0$.

- If a limit point x^* is infeasible, it is a stationary point of $\|h(x)\|^2$.
- If a limit point x^* is feasible and the constraint gradients $\nabla h_i(x^*)$ are linearly independent, then x^* is a KKT point for (COP:E), and we have that

$$\lim_{k \rightarrow \infty} \mu_k h_i(x_k) = \nu_i^*, \quad i = 1, \dots, p,$$

where ν^* is the Lagrange multiplier vector that satisfies the KKT conditions for (COP:E).

Proof:

$$\nabla_x Q(x_k; \mu_k) = \nabla f(x_k) + \sum_{i=1}^p \mu_k h_i(x_k) \nabla h_i(x_k) \quad (\text{dQ})$$

From the convergence criterium $\|\nabla_x Q(x_k; \mu_k)\| \leq \tau_k$ (using the inequality $\|a\| - \|b\| \leq \|a + b\|$) we obtain

$$\left\| \sum_{i=1}^p h_i(x_k) \nabla h_i(x_k) \right\| \leq \frac{1}{\mu_k} (\tau_k + \|\nabla f(x_k)\|).$$

As $k \rightarrow \infty$: $\tau_k \rightarrow 0$, $\|\nabla f(x_k)\| \rightarrow \|\nabla f(x^*)\|$ and $\mu_k \rightarrow \infty$ thus the limit of the sequence on the l.h.s. is

$$\sum_{i=1}^p h_i(x^*) \nabla h_i(x^*) = 0.$$

- i) If $\exists i \in \{1, \dots, p\} : h_i(x^*) \neq 0$ then $\nabla h_i(x^*)$ are linearly dependent which implies that x^* is a stationary point of $\|h(x)\|^2$.
- ii) If $\nabla h_i(x^*)$, $i = 1, \dots, p$ are linearly independent, $h_i(x^*) = 0$, $i = 1, \dots, p$ and x^* is primarily feasible i.e. satisfies the second KKT condition. It remains to show that the “dual feasibility” (the first KKT condition) is satisfied.

Case ii):

Intuition:

As $k \rightarrow \infty$, $Q(x^k)$ should approach the Lagrangian

$$\mathcal{L}(x^*; \nu^*) = f(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*). \quad (\text{L})$$

and $\nabla_x Q(x^k)$ its derivative i.e. the “dual feasibility” condition

$$\nabla_x \mathcal{L}(x^*; \nu^*) = \nabla f(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*). \quad (\text{dL})$$

Rearranging (dQ) and denoting $A(x)^T := \nabla h_i(x_k)$, $i = 1, \dots, p$ and $\nu^k := \mu_k h(x_k)$ we obtain

$$A(x_k)^T \nu^k = -\nabla f(x_k) + \nabla_x Q(x_k; \mu_k), \quad \|\nabla_x Q(x_k; \mu_k)\| \leq \tau_k.$$

For large enough k the matrix $A(x_k)$ has full row rank and hence the above overdetermined system has the unique solution

$$\nu^k = (A(x_k) A(x_k)^T)^{-1} A(x_k) [-\nabla f(x_k) + \nabla_x Q(x_k; \mu_k)].$$

Taking the limit as $k \rightarrow \infty$

$$\lim_{k \rightarrow \infty} \nu^k = \nu^* = - (A(x^*) A(x^*)^T)^{-1} A(x^*) \nabla f(x^*)$$

and the same in (dQ) yields the “dual feasibility” condition

$$\nabla f(x^*) + A(x^*)^T \nu^* = 0.$$

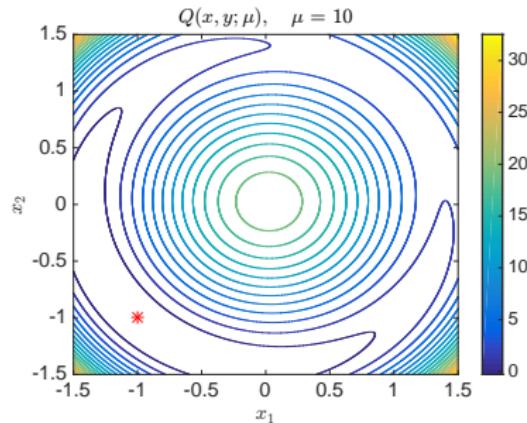
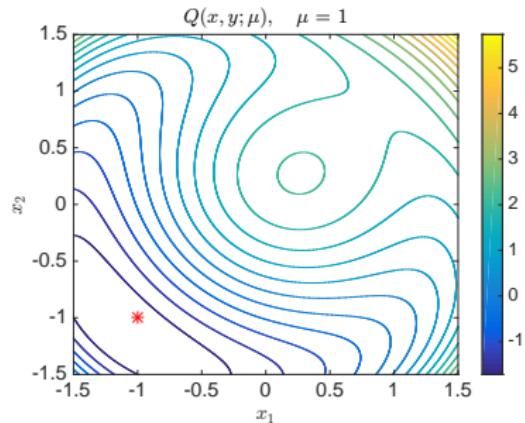
Hence, x^* is the KKT point with unique Lagrange multiplier ν^* .

Example

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 - 2 = 0. \end{aligned}$$

Solution: $(-1, -1)^T$.

Quadratic penalty function: $Q(x; \mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2$.



Problems

For equality constraints, $Q(x; \mu)$ is smooth and can be solved with methods for unconstraint optimisation.

- Hessian ill-conditioning (see next slide) poses problems for methods like CG or quasi Newton, Newton's convergence does not depend on it, but still poses numerical problems which can be remedied by reformulation.
- For larger μ the quadratic model underlying most solvers is a poor approximation to $Q(x; \mu)$.
- Example:

$$\begin{aligned} \min \quad & -5x_1^2 + x_2^2 \\ \text{subject to} \quad & x_1 = 1. \end{aligned}$$

has a solution $(1, 0)^T$. The quadratic penalty function

$$Q(x; \mu) = -5x_1^2 + x_2^2 + \frac{\mu}{2}(x_1 - 1)^2$$

is unbounded for $\mu < 10$. The iterates would diverge.
Unfortunately, a common problem.

III-conditioning of Hessian

Newton step: $\nabla_{xx}^2 Q(x; \mu_k) p_n = -\nabla_x Q(x; \mu_k)$

$$\nabla_{xx}^2 Q(x; \mu_k) = \nabla^2 f(x) + \sum_{i=1}^p \underbrace{\mu_k h_i(x)}_{\approx \nu_i^*} \nabla^2 h_i(x) + \mu_k \underbrace{\nabla h(x)}_{=: A(x)^T} \nabla h(x)^T.$$

If x is sufficiently close to the minimiser of $Q(\cdot; \mu_k)$

$$\nabla_{xx}^2 Q(x; \mu_k) \approx \nabla_{xx}^2 \mathcal{L}(x; \nu^*) + \mu_k A(x)^T A(x).$$

As $\mu_k \rightarrow \infty$ the Hessian is dominated by the second term and hence increasingly ill-conditioned.

Alternative formulation avoids ill-conditioning, $\zeta = \mu_k A(x) p_n$

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i=1}^p \mu_k h_i(x) \nabla^2 h_i(x) & A(x)^T \\ A(x) & \mu_k^{-1} I \end{bmatrix} \begin{bmatrix} p_n \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x; \mu_k) \\ 0 \end{bmatrix}.$$

Still, if $\mu_k h_i(x)$ is not a good enough approximation to ν^* , inadequate quadratic model yields inadequate search direction p_n .

General constraint problem

For general constraint problems including equality and inequality constraints, the quadratic penalty function can be defined as

$$Q(x; \mu) := f(x) + \frac{\mu}{2} \sum_{i=1}^p h_i^2(x) + \frac{\mu}{2} \sum_{i=1}^m ([f_i(x)]^+)^2,$$

where $[y]^+ := \max\{y, 0\}$

Note: Q may be less smooth than the objective and constraint functions e.g. $f_1(x) = x_1 \geq 0$, then $\max\{y, 0\}^2$ has discontinuous second derivate and so does Q .

Practical penalty methods

- μ_k can be chosen adaptively based on the difficulty of minimising the penalty function in each iteration i.e. when minimising $Q(x; \mu_k)$ is expensive, choose μ_{k+1} moderately larger than μ_k e.g. $\mu_{k+1} = 1.5\mu_k$, when minimising $Q(x; \mu_k)$ is cheap, choose μ_{k+1} larger e.g. $\mu_{k+1} = 10\mu_k$.
- There is no guarantee that $\|\nabla_x Q(x; \mu_k)\| \leq \tau_k$ will be satisfied. Practical implementations need safe guards to increase μ (and possibly restore the initial point) when constraint violation is not decreasing fast enough or when the iterates appear diverging.
- Choice of initial point e.g. warm start $x_{k+1}^s = x_k$ can improve performance of Newton.

Nonsmooth penalty functions

Some penalty functions are *exact* i.e. for certain choices of penalty parameters, minimisation w.r.t. x yields the exact minimiser of f .
To be exact the penalty function has to be nonsmooth.

$$Q_1(x; \mu) := f(x) + \mu \sum_{i=1}^p |h_i(x)| + \mu \sum_{i=1}^m [f_i(x)]^+,$$

where $[y]^+ := \max\{y, 0\}$.

Let x^* be a strict local minimiser of (COP), which satisfies the 1st order necessary conditions with Lagrange multipliers ν^*, λ^* . Then x^* is a local minimiser of $Q_1(x; \mu)$ for all $\mu > \mu^* = \|(\nu^*, \lambda^*)^T\|_\infty$. If moreover, the 2nd order sufficient conditions hold at $\mu > \mu^*$, then x^* is a strict local minimiser of $Q_1(x; \mu)$.

Let \hat{x} be a stationary point of the penalty function $Q_1(x; \mu)$ for all $\mu > \hat{\mu} > 0$. Then, if \hat{x} is feasible for (COP), it satisfies KKT conditions. If \hat{x} is not feasible for (COP), it is an infeasible stationary point.

1d example of general constraints (threshold μ^*)

$$\begin{aligned} & \min x \\ \text{s.t. } & x \geq 1 \end{aligned}$$

with solution $x^* = 1$.

$$Q_1(x; \mu) = x + \mu[1 - x]^+ = \begin{cases} x & x \geq 1 \\ x + \mu(1 - x) & x < 1 \end{cases}$$

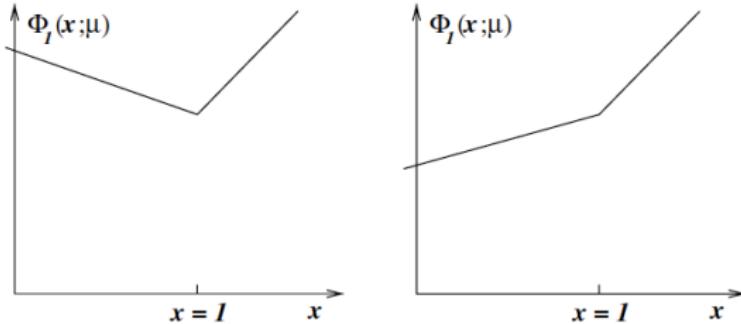


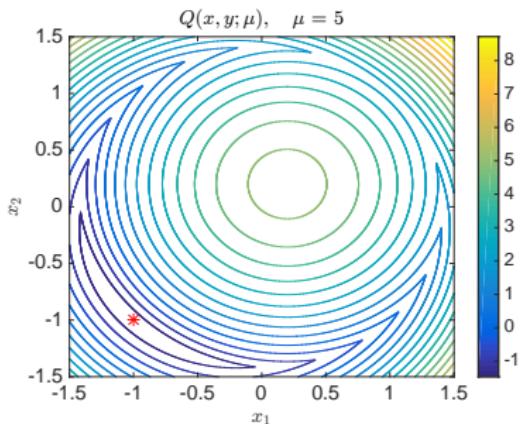
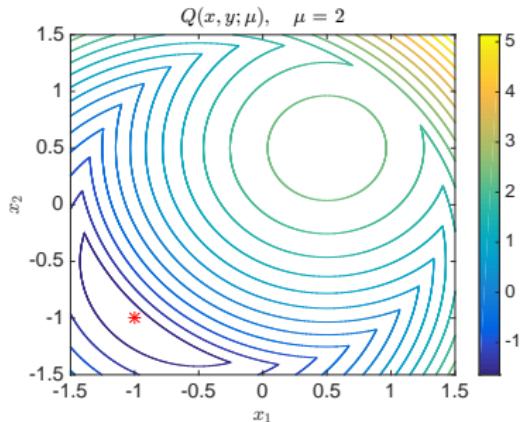
Figure: Fig. 17.3 from Nocedal, Wright: (left) $\mu > 1$, x^* minimised Q_1 , (right) $\mu < 1$, Q_1 is unbounded.

Example revisited

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 - 2 = 0. \end{aligned}$$

Solution: $(-1, -1)^T$.

ℓ_1 penalty function: $Q_1(x; \mu) = x_1 + x_2 + \mu|x_1^2 + x_2^2 - 2|$.



Augmented Lagrangian

Reduces ill-conditioning by introducing explicit Lagrange multiplier estimates into the function to be minimised.

Can preserve smoothness. Can be implemented using standard unconstrained (or bound constrained) optimization.

Motivation: The minimisers x_k of $Q(x; \mu_k)$ do not quite satisfy the feasibility condition $h_i(x) = 0$

$$h_i(x_k) \approx \nu^*/\mu_k, \quad i = 1, \dots, p.$$

Obviously, in the limit $\mu_k \rightarrow \infty$, $h_i(x) \rightarrow 0$ but can we avoid this systematic perturbation for moderate values of μ_k ?

Augmented Lagrangian:

$$\mathcal{L}_A(x, \nu; \mu) := f(x) + \sum_{i=1}^p \nu h_i(x) + \frac{\mu}{2} \sum_{i=1}^p h_i^2(x).$$

Update of Lagrange multiplier estimate

Optimality condition for the unconstraint minimiser of
 $\mathcal{L}_A(x, \nu^k; \mu_k)$

$$0 \approx \nabla_x \mathcal{L}_A(x_k, \nu^k; \mu_k) = \nabla f(x_k) + \sum_{i=1}^p [\nu_i^k + \mu_k h_i(x_k)] \nabla h_i(x_k).$$

Optimality condition for the Lagrangian of (COP:E)

$$0 \approx \nabla_x \mathcal{L}(x_k, \nu^*) = \nabla f(x_k) + \sum_{i=1}^p \nu_i^* \nabla h_i(x_k).$$

Comparison yields (an update scheme for ν):

$$\nu_i^* \approx \nu_i^k + \mu_k h_i(x_k), \quad i = 1, \dots, p$$

as from $h_i(x_k) = \frac{1}{\mu_k} (\nu_i^* - \nu_i^k)$, $i = 1, \dots, p$ we see that if ν^k is close to ν^* the infeasibility goes to 0 faster than $1/\mu_k$.

Example revisited

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 - 2 = 0. \end{aligned}$$

Solution: $(-1, -1)^T$.

Augmented Lagrangian:

$$\mathcal{L}(x, \nu; \mu) = x_1 + x_2 + \nu(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$

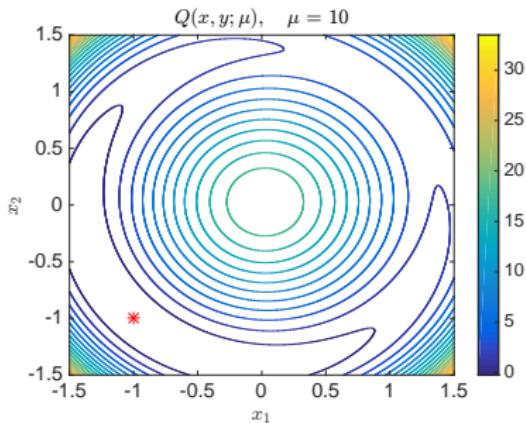
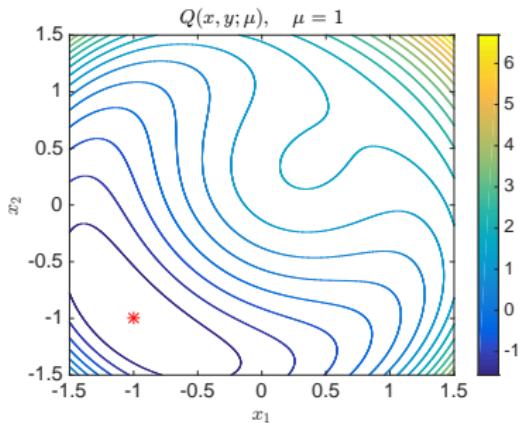


Figure: $\nu = 0.4$

Convergence

Let x^* be a local minimiser of (COP:E) at which the constraint gradients are linearly independent and which satisfies the 2nd order sufficient conditions with Lagrange multipliers ν^* . Then for all $\mu \geq \bar{\mu} > 0$, x^* is a strict local minimiser of $\mathcal{L}_A(x, \nu^*; \mu)$. Furthermore, there exist $\delta, \epsilon, M > 0$ such that for all ν^k, μ_k satisfying

$$\|\nu^k - \nu^*\| \leq \mu_k \delta, \quad \mu_k \geq \bar{\mu},$$

- the problem $\min \mathcal{L}_A(x, \nu^k; \mu_k)$, subject to $\|x - x^*\| \leq \epsilon$, has a unique solution x_k and it holds

$$\|x_k - x^*\| \leq M \|\nu^k - \nu^*\| / \mu_k$$

- it holds

$$\|\nu^{k+1} - \nu^*\| \leq M \|\nu^k - \nu^*\| / \mu_k,$$

where $\nu^{k+1} = \nu^k + \mu_k h(x_k)$.

- the matrix $\nabla_{xx}^2 \mathcal{L}_A(x_k, \nu^k; \mu_k)$ is positive definite and the constraint gradients $\nabla h_i(x_k), i = 1, \dots, p$ are linearly independent.

- **Bound constraint formulation:** convert inequality constraints into equality constraints using slack variables

$$f_i(x) - s_i = 0, \quad s_i \geq 0, \quad i \in \{1, \dots, m\}.$$

Bound constraints are not transformed. Solve by projected gradient algorithm

$$x_{k+1} = P(x_k - \nabla_x \mathcal{L}_A(x, \nu; \mu)|_{x_k}; l, u) = 0,$$

where $P(\cdot; l, u)$ projects on the box $[l, u]$.

- **Linearly constraint formulation:** transform into equality constraint problem with linearised constraints

$\min F_k(x)$, subject to $f_i(x_k) + \nabla f_i^T(x_k)(x - x_k) = 0$, $l \leq x \leq u$.

Choose F_k as

$$F_k(x) = f(x) + \sum_{i=1}^m \nu_i^k \bar{f}_i^k(x),$$

explicitly including the higher order constraint violation

$$\bar{f}_i^k(x) = f_i(x) - f_i(x_k) - \nabla f_i(x_k)^T(x - x_k).$$

Preferred choice (larger convergence radius in practise)

$$F_k(x) = f(x) + \sum_{i=1}^m \nu_i^k \bar{f}_i^k(x) + \frac{\mu}{2} \sum_{i=1}^m (\bar{f}_i^k(x))^2$$

- **Unconstraint formulation:** obtain unconstraint formulation using smooth approximation to feasibility set indicator function.

Numerical Optimisation

Constraint optimisation:

Interior point methods

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 15 (based on Boyd, Vanderbergher)

Convex constraint optimisation problem

Convex constraint optimization problem

$$\min_{x \in \mathcal{D} \subset \mathbb{R}^n} f(x) \quad (\text{CCOP})$$

$$\begin{aligned} \text{subject to } & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & Ax = b, \end{aligned}$$

where

- $f : \mathcal{D} \rightarrow \mathbb{R}$ is convex, twice continuously differentiable function, $\mathcal{D} \subset \mathbb{R}^n$ is convex
- $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ are convex, twice continuously differentiable functions
- $A \in \mathbb{R}^{p \times n}$ with $\text{rank } A = p < n$.

We assume that

- (CCOP) is solvable i.e. an optimal x^* exists, and we denote the optimal value as $p^* = f(x^*)$.
- (CCOP) is strictly feasible i.e. there exists $x \in \mathcal{D}$ that satisfies $Ax = b$ and $f_i(x) < 0$ for $i = 1, \dots, m$. This means that Slater's constraint qualification holds, thus there exists dual optimal $\lambda^* \in \mathbb{R}^m$, $\nu^* \in \mathbb{R}^P$, which together with x^* satisfy the KKT conditions

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + A^T \nu^* = 0, \quad (\text{KKT})$$

$$Ax^* = b,$$

$$f_i(x^*) \leq 0, \quad i = 1, \dots, m,$$

$$\lambda^* \geq 0,$$

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, \dots, m.$$

Interior point methods

Interior point methods solve either

- the problem (CCOP) by applying Newton's method to a sequence of equality constraint problems.
- the conditions (KKT) by applying Newton's method to a sequence of modified versions of the KKT conditions.

We will consider the *barrier method* and the *primal-dual interior-point method*.

Logarithmic barrier function

Rewrite (CCOP) making the inequality constraints implicit

$$\min_{x \in \mathcal{D} \subset \mathbb{R}^n} f(x) + \sum_{i=1}^m I_-(f_i(x))$$

subject to $Ax = b,$

where $I_- : \mathbb{R} \rightarrow \mathbb{R}$ is the indication function for the nonpositive reals

$$I_-(u) = \begin{cases} 0 & u \leq 0, \\ \infty & u > 0. \end{cases}$$

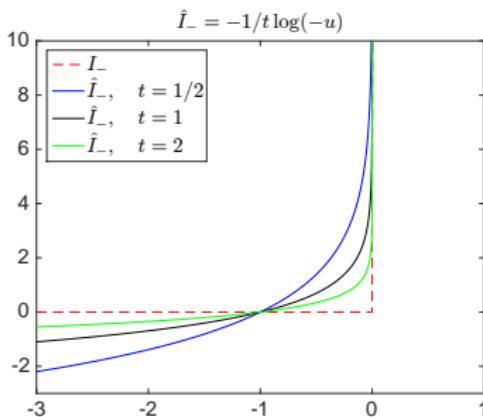
I_- is non-differentiable thus we need a smooth approximation before Newton method can be applied.

Approximate I_- with a smooth *logarithmic barrier*

$$\hat{I}_-(u) = -1/t \log(-u), \quad \text{dom } \hat{I}_- = [-\infty, 0),$$

where $t > 0$ is a parameter that sets the accuracy of the approximation.

- Like I_- , \hat{I}_- is convex, nondecreasing and by convention ∞ for $u > 0$.
- Unlike I_- , \hat{I}_- is differentiable and closed i.e. it increases to ∞ as u increases to 0.



Substituting \hat{I}_- for I_- yields an approximation

$$\min_{x \in \mathcal{D} \subset \mathbb{R}^n} f(x) + \sum_{i=1}^m -1/t \log(-f_i(x))$$

subject to $Ax = b$.

The objective function is convex since $-1/t \log(-u)$ is convex, increasing in u , and differentiable, thus Newton's method can be applied.

Logarithmic barrier

$$\phi(x) = -\sum_{i=1}^m \log(-f_i(x)), \quad \text{dom } \phi = \{x \in \mathbb{R}^n : f_i(x) < 0, i = 1, \dots, m\}$$

Gradient and Hessian of ϕ

$$\nabla \phi(x) = \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x),$$

$$\nabla^2 \phi(x) = \sum_{i=1}^m \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla^2 f_i(x)$$

Central path

Consider the equivalent problem

$$\min_{x \in \mathcal{D} \subset \mathbb{R}^n} \quad tf(x) + \phi(x) \quad (\text{CENT})$$

subject to $Ax = b.$

We assume that (CENT) has a unique solution for each $t > 0$, and denote this solution with $x^*(t)$.

The set of points $x^*(t)$, $t > 0$ is called the **central path**. The points on central path are characterised by the following necessary and sufficient *centrality conditions*:

$x^*(t)$ is strictly feasible i.e. satisfies

$$Ax^*(t) = b, \quad f_i(x^*(t)) < 0, \quad i = 1, \dots, m,$$

and there exists a $\hat{\nu} \in \mathbb{R}^p$ such that

$$0 = t\nabla f(x^*(t)) + \nabla\phi(x^*(t)) + A^T\hat{\nu} \quad (\text{CENT:COND})$$

$$= t\nabla f(x^*(t)) + \sum_{i=1}^m \frac{1}{-f_i(x^*(t))} \nabla f_i(x^*(t)) + A^T\hat{\nu}$$

Example: LP with inequality constraints

$$\min_{x \in \mathbb{R}^n} c^T x$$

$$\text{subject to } Ax \leq b.$$

The logarithmic barrier:

$$\phi(x) = -\sum_{i=1}^m \log(b_i - \underbrace{a_i^T x}_{=: A_{i,:}}), \quad \text{dom } \phi = \{x : Ax < b\}.$$

The gradient and Hessian:

$$\nabla \phi(x) = \sum_{i=1}^m \frac{1}{b_i - a_i^T x} a_i, \quad \nabla^2 \phi(x) = \sum_{i=1}^m \frac{1}{(b_i - a_i^T x)^2} a_i a_i^T.$$

Since x is strictly feasible, we have $b_i - a_i^T x > 0$ and the Hessian is nonsingular iff A has rank n .

The centrality condition (CENT:COND): $(\nabla \phi(x^*(t)) \parallel -c)$

$$tc + \sum_{i=1}^m \frac{1}{b_i - a_i^T x} a_i = 0.$$

Example: central path for LP

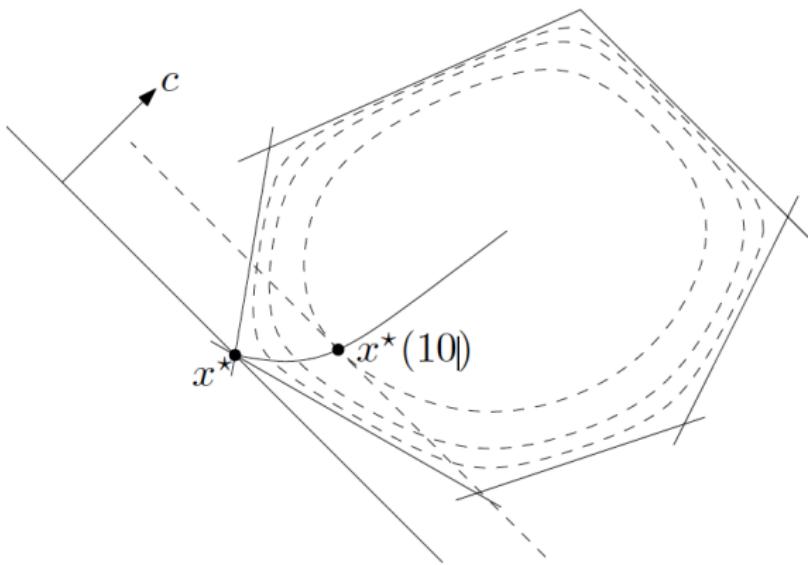


Figure: Boyd Vandenberghe Fig. 11.2

Dual points from central path

Claim: Every point on central path yields a dual feasible point and hence a lower bound on p^* . More precisely, the pair

$$\lambda^*(t) = \frac{1}{-tf_i(x^*(t))}, \quad i = 1, \dots, m, \quad \nu^*(t) = \hat{\nu}/t$$

is dual feasible.

Proof: $\lambda^*(t) > 0$ because $x^*(t)$ is strictly feasible $f_i(x^*(t)) < 0$.
The optimality conditions (CENT:COND)

$$0 = \nabla f(x^*(t)) + \sum_{i=1}^m \frac{1}{-tf_i(x^*(t))} \nabla f_i(x^*(t)) + \frac{1}{t} A^T \hat{\nu}$$

implies that $x^*(t)$ minimises the Lagrangian of (CCOP)

$$\mathcal{L}(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i f_i(x) + \nu^T (Ax - b)$$

for the fixed $\lambda = \lambda^*(t), \nu = \nu^*(t)$.

This means that $\lambda^*(t), \nu^*(t)$ are dual feasible, the dual function is finite (recall: $g \leq p^*$ so whenever $p^* < \infty$ i.e. the primal problem is feasible, $g < \infty$) and

$$\begin{aligned}
g(\lambda^*(t), \nu^*(t)) &= \inf_x \mathcal{L}(x, \lambda^*(t), \nu^*(t)) = \mathcal{L}(x^*(t), \lambda^*(t), \nu^*(t)) \\
&= f(x^*(t)) + \sum_{i=1}^m \underbrace{\lambda_i^*(t)}_{= -\frac{1}{tf_i(x^*(t))}} f_i(x^*(t)) \\
&\quad + \nu^*(t)^T \underbrace{(Ax^*(t) - b)}_{= 0 \text{ (primal feasibility)}} \\
&= f(x^*(t)) - \cancel{m/t} \rightarrow \text{duality gap}
\end{aligned}$$

thus as $g(\lambda^*(t), \nu^*(t)) \leq p^*$, $x^*(t)$ is not more than m/t suboptimal

$$f(x^*(t)) - p^* \leq m/t$$

and $x^*(t)$ converges to an optimal point as $t \rightarrow \infty$.

Interpretation via KKT conditions

We can interpret the central path conditions as a continuous deformation of (KKT). A point x is equal to $x^*(t)$ iff there exists λ, ν such that

$$\nabla f(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + A^T \nu = 0, \quad (\text{KKT:CENT})$$

$$Ax = b,$$

$$f_i(x) \leq 0, \quad i = 1, \dots, m,$$

$$\lambda \geq 0,$$

$$-\lambda_i f_i(x) = 1/t, \quad i = 1, \dots, m.$$

The only difference to (KKT) is the complementarity slackness condition being replaced by $-\lambda_i f_i(x) = 1/t$. Consequently, for large t , $x^*(t), \lambda^*(t), \nu^*(t)$ almost satisfy the KKT conditions.

Newton for centering problem (CENT)

The Newton step for the centering problem (CENT) (linear equality constraint problem) reads

$$\begin{bmatrix} t\nabla^2 f(x) + \nabla^2 \phi(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_n \\ \nu_n \end{bmatrix} = - \begin{bmatrix} t\nabla f(x) + \nabla \phi(x) \\ 0 \end{bmatrix}.$$

and more compactly

$$\begin{bmatrix} tH & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_n \\ \nu_n \end{bmatrix} = - \begin{bmatrix} tg \\ 0 \end{bmatrix},$$

where

$$H = \nabla^2 f(x) + \frac{1}{t} \nabla^2 \phi(x),$$

$$g = \nabla f(x) + \frac{1}{t} \nabla \phi(x).$$

Here we assumed feasibility.

We can interpret this Newton step for (CENT) as Newton for directly solving the modified (KKT:CENT) in a particular way.

Newton for modified KKT (KKT:CENT)

First, eliminate λ using $\lambda_i = -1/(tf_i(x))$ from the (KKT:CENT) system

$$\nabla f(x) + \underbrace{\sum_{i=1}^m \frac{1}{-tf_i(x)} \nabla f_i(x)}_{= \frac{1}{t} \nabla \phi(x)} + A^T \nu = 0, \quad Ax = b.$$

To find the Newton step for the solution of the nonlinear equations above, we form the Taylor expansion for the [nonlinear term](#)

$$\begin{aligned} & \nabla f(x + v) + \frac{1}{t} \nabla \phi(x + v) \\ & \approx \underbrace{\nabla f(x) + \frac{1}{t} \nabla \phi(x)}_{=:g} + \underbrace{\left(\nabla^2 f(x) + \frac{1}{t} \nabla^2 \phi(x) \right) v}_{=:Hv}. \end{aligned}$$

Replace the nonlinear term with this linear approximation

$$H\nu + A^T \nu = -g, \quad A\nu = 0$$

and observe that the Newton step $\Delta x_n, \nu_n$ for (CENT) satisfies

$$tH\Delta x_n + A^T \nu_n = -tg, \quad A\Delta x_n = 0.$$

Comparing to the Newton step for (KKT:CENT) yields

$$\nu = \Delta x_n, \quad \nu = (1/t)\nu_n.$$

This shows that the Newton step for the centring problem (CENT) can be interpreted (after scaling of the dual variable) as the Newton step for solving the modified (KKT:CENT) system.

The barrier method

Require: Strictly feasible $x^s := x^{(0)}$, $t := t^{(0)} > 0$, $\mu > 1$

Require: Tolerance $\epsilon > 0$

- 1: **loop**
- 2: Centering step:
 obtain $x^*(t)$ by minimising (CENT) starting from x^s
- 3: Update $x^s = x^*(t)$
- 4: **if** $m/t < \epsilon$ **then**
- 5: break {stopping criterium ϵ -sub optimal point}
- 6: **end if**
- 7: Increase $t = \mu t$
- 8: **end loop**

- **Centring step:** can be solved by any methods for linearly constraint minimisation, in particular Newton method.
 - **Exact centring** is not necessary since the central path has no significance beyond that it leads to the solution of the original problem (CCOP) as $t \rightarrow \infty$.
 - **Inexact centring** will still produce a convergent sequence, however the $\lambda^*(t), \nu^*(t)$ are not exactly dual feasible (can be corrected).
 - The difference in cost between the exact and good approximate centring is marginal (few Newton steps), so centring is usually assumed exact.

- **Choice of μ :** trade off between the number of outer (centering) and inner (Newton) iterations.
 - **Small $\mu \approx 1$:** good initial guess for Newton i.e. small number of inner iterations closely following the central path but a large number of outer iterations to reach desired accuracy ϵ .
 - **Larger μ :** only a few outer iterations but with a large number of inner iterations straying from the central path.
 - In practice for a large range of $\mu \in (3, 100)$ these effects balance each other yielding approximately same total number of Newton iterations. Values around 10-20 seem to work well. For best worst-case bound on total Newton steps set μ close to 1.

- **Choice of $t^{(0)}$:** Trade off between the number of inner iterations in the first step and number of outer iterations.
 - Choose so that $m/t^{(0)} \approx f(x^{(0)}) - p^*$. For instance if a dual feasible point λ, ν is known with the duality gap $\eta = f(x^{(0)}) - g(\lambda, \nu)$, then we can set $t^{(0)} = m/\eta$ (the first centring step will compute a pair with the same duality gap as the initial primal and dual feasible points).
 - Choose $t^{(0)}$ as a minimiser of

$$\inf_{t, \nu} \|t \nabla f(x^{(0)}) + \nabla \phi(x^{(0)}) + A^T \nu\|,$$

a measure of deviation of $x^{(0)}$ from $x^*(t)$ (least squares problem for t, ν).

- **Infeasible Newton method** for centring step. Choose $x^{(0)} \in \mathcal{D}$, $f_i(x^{(0)}) < 0, i = 1, \dots, m$ but not necessarily $Ax^{(0)} = b$. Assuming the centring problem is strictly feasible, a full Newton step is taken at some point during the first centring step and thereafter the iterates are primal feasible and the algorithm coincides with the standard barrier method.

Computing a strictly feasible point

The barrier method requires a strictly feasible point $x^{(0)}$. When such a point is not known, the barrier method is preceded by a preliminary stage called *phase I* to compute a strictly feasible point (or to find that the constraints are infeasible).

Consider the set of inequalities and equalities

$$f_i(x) \leq 0, \quad i = 1, \dots, m, \quad Ax = b \quad (\text{FEAS})$$

Assume we have a point $x^{(0)} \in \prod_{i=1}^m \text{dom } f_i$ and $Ax^{(0)} = b$ i.e. the inequalities are possibly not satisfied at $x^{(0)}$.

Phase I: max

Goal: find a strictly feasible solution of equalities and inequalities:

$$\begin{aligned} \min \quad & s \\ \text{subject to} \quad & f_i(x) \leq s, \quad i = 1, \dots, m \\ & Ax = b \end{aligned} \tag{PH1:MAX}$$

s : bound on the maximum infeasibility of the inequalities. The goal is to drive this maximum below 0.

The problem (PH1:MAX) is always strictly feasible. Thus we can initialise with $x = x^{(0)}$ and for s with any number larger than $\max_{i=1, \dots, m} f_i(x^{(0)})$ and apply the barrier method.

Let p_I^* denote the optimal value for (PH1:MAX).

- $p_I^* < 0$: (FEAS) has a strictly feasible solution.
If (x, s) is feasible for (PH1:MAX) with $s < 0$, then x satisfies $f_i(x) < 0$.
We do not need to solve (PH1:MAX) with high accuracy, we can terminate when $s < 0$.
- $p_I^* > 0$: (FEAS) are infeasible.
We do not need to solve (PH1:MAX) with high accuracy, we can terminate when a dual feasible point is found with positive objective, which proves that $p_I^* > 0$.
- If $p_I^* = 0$ and the minimum is attained at x^* and $s^* = 0$, then the set of inequalities is feasible, but not strictly feasible. If $p^* = 0$ and the minimum is not attained, the inequalities are infeasible.

Phase I: sum

$$\begin{aligned} \min \quad & \mathbf{1}^T s \\ \text{subject to} \quad & f_i(x) \leq s_i, \quad i = 1, \dots, m \\ & Ax = b \\ & s \geq 0 \end{aligned} \tag{PH1:SUM}$$

- For a fixed x , the optimal value of s_i is $\max\{f_i(x), 0\}$. Thus we are minimising a sum of infeasibilities.
- The optimal value is 0 and achieved iff the original set of equalities and inequalities is feasible.
- When the system of equalities and inequalities is infeasible, often the solution violates only a small number of constraints i.e. we identified a large feasible subset. This is more informative than finding that m inequalities together are mutually infeasible.

Termination near phase II central path

Assume $x^{(0)} \in \mathcal{D} \cap \prod_{i=1}^m \text{dom} f_i$ with $Ax^{(0)} = b$.

Modified phase I optimisation problem

$$\begin{aligned} & \min \quad s \\ \text{subject to} \quad & f_i(x) \leq s, \quad i = 1, \dots, m \\ & f(x) \leq M \\ & Ax = b \end{aligned}$$

with $M > \max\{f(x^{(0)}), p^*\}$.

Central path for this modified problem $(x^*(\bar{t}), s^*(\bar{t}))$, $\bar{t} > 0$

$$\sum_{i=1}^m \frac{1}{s - f_i(x)} = \bar{t}, \quad \frac{1}{M - f(x)} \nabla f(x) + \sum_{i=1}^m \frac{1}{s - f_i(x)} \nabla f_i(x) + A^T \nu = 0.$$

If (x, s) with $s = 0$ is on this central path, it is also on the central path for (CCOP) if the latter is strictly feasible i.e. $p^* < 0$ as

$$t \nabla f(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) + A^T \nu = 0$$

with $t = 1/(M - f(x))$ and dual. gap $m(M - f(x)) \leq m(M - p^*)$.

Phase I via infeasible Newton

We express (CCOP) in an equivalent form

$$\begin{aligned} & \min f(x) \\ \text{subject to } & f_i(x) \leq s, \quad i = 1, \dots, m \\ & Ax = b, \quad s = 0. \end{aligned}$$

Start the barrier method using infeasible Newton method so solve

$$\begin{aligned} & \min t f(x) - \sum_{i=1}^m \log(s - f_i(x)) \\ \text{subject to } & Ax = b, \quad s = 0, \end{aligned}$$

which can be initialised with any $x \in \mathcal{D}$ and any $s > \underbrace{\max_i f_i(x)}_{\text{infeasibility}}$.

Provided the problem is strictly feasible, the infeasible Newton will eventually take an undamped step and thereafter we will have $s = 0$ i.e. x strictly feasible.

Finding a point in the domain \mathcal{D}

The same trick can be applied if a point in $\mathcal{D} \cap \prod_{i=1}^m \text{dom } f_i$ (domain of the function and inequality constraints) is not known.

Apply infeasible Newton to

$$\min \quad t f(x + z_0) - \sum_{i=1}^m \log(s - f_i(x + z_i))$$

subject to $Ax = b, s = 0, z_0 = 0, z_1 = 0, \dots, z_m = 0,$

with initialisation $z_i : x + z_i \in \text{dom } f_i$.

Disadvantage: no good stopping criterion for infeasible problems; the residual simply fails to converge to 0.

Characteristic performance

- Typically the cost of solving a set of convex inequalities and linear equalities using the barrier method is modest, and approximately constant, as long as the problem is not very close to the boundary between feasibility and infeasibility.
- When the problem is very close to the boundary, the number of Newton steps required to find a strictly feasible point or produce a certificate of infeasibility grows.
- When the problem is exactly on the boundary between strictly feasible and infeasible, for example, feasible but not strictly feasible, the cost becomes infinite.
- Typically the infeasible start Newton method works very well provided the inequalities are feasible, and not very close to the boundary between feasible and infeasible.
- When the feasible set is just barely nonempty, a phase I method is far better choice. Phase I method gracefully handles the infeasible case; the infeasible start Newton method, in contrast, simply fails to converge.

Primal-dual interior point method

Primal-dual interior point method is similar to barrier method with key differences:

- There is only one loop or iteration, i.e., there is no distinction between inner and outer iterations as in the barrier method. At each iteration, both the primal and dual variables are updated.
- The search directions in a primal-dual interior-point method are obtained from Newton's method, applied to modified KKT equations (i.e. the optimality conditions for the logarithmic barrier centring problem). The primal-dual search directions are similar to, but not quite the same as, the search directions that arise in the barrier method.
- In a primal-dual interior-point method, the primal and dual iterates are not necessarily feasible.
- Usually more efficient than barrier methods, and do not require strict feasibility.

Primal-dual search direction

As in barrier method we start from (KKT:CENT) which we rewrite in the form

$$0 = r_t(x, \lambda, \nu) = \begin{bmatrix} \nabla f(x) + J(x)^T \lambda + A^T \nu \\ -\text{diag}(\lambda) F(x) - (1/t) \mathbf{1} \\ Ax - b \end{bmatrix} =: \begin{bmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{prim}} \end{bmatrix},$$

with $t > 0$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and its Jacobian

$$F(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}, \quad J(x) = DF(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}.$$

If x, λ, ν satisfy $r_t(x, \lambda, \nu) = 0$ (and $f_i(x) < 0$), then

$x = x^*(t)$, $\lambda = \lambda^*(t)$, $\nu = \nu^*(t)$. In particular, x is primal feasible, and λ, ν are dual feasible, with duality gap m/t .

Newton step for solution of $r_t(x, \lambda, \nu) = 0$ at $y = (x, \lambda, \nu)$ a primal-dual strictly feasible point $F(x) < 0, \lambda > 0$.

Difference to barrier method: we do not eliminate λ before taking the Newton step

$$r_t(y + \Delta y) \approx r_t(y) + Dr_t(y)\Delta y = 0,$$

where $\Delta y = (\Delta x, \Delta \lambda, \Delta \nu)$ is the *primal-dual search direction*.

Written in terms of x, λ, ν :

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & J(x)^T & A^T \\ -\text{diag}(\lambda)J(x) & -\text{diag}(F(x)) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{prim}} \end{bmatrix}. \quad (\text{PD:N})$$

Comparison of primal-dual and barrier search directions

Eliminate $\Delta\lambda_{\text{pd}}$ from (PD:N):

From the second block

$$\Delta\lambda_{\text{pd}} = -\text{diag}(F(x))^{-1} \text{diag}(\lambda) J(x) \Delta x_{\text{pd}} + \text{diag}(F(x))^{-1} r_{\text{cent}}$$

and substitute into the first block

$$\begin{bmatrix} H_{\text{pd}} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{pd}} \\ \Delta \nu_{\text{pd}} \end{bmatrix} = - \begin{bmatrix} r_{\text{dual}} + J(x)^T \text{diag}(F(x))^{-1} r_{\text{cent}} \\ r_{\text{pri}} \end{bmatrix}$$
$$= - \begin{bmatrix} \nabla f(x) + \frac{1}{t} \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) + A^T \nu \\ r_{\text{pri}} \end{bmatrix},$$

where

$$H_{\text{pd}} = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) + \sum_{i=1}^m \frac{\lambda_i}{-f_i(x)} \nabla f_i(x) \nabla f_i(x)^T.$$

Compare to the Newton step in the barrier method (in the infeasible form)

$$\begin{bmatrix} H_{\text{bar}} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{bar}} \\ \nu_{\text{bar}} \end{bmatrix} = - \begin{bmatrix} t \nabla f(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) \\ r_{\text{pri}} \end{bmatrix},$$

where

$$H_{\text{bar}} = t \nabla^2 f(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla^2 f_i(x) + \sum_{i=1}^m \frac{1}{f_i^2(x)} \nabla f_i(x) \nabla f_i(x)^T.$$

Multiplying first block by $1/t$ and changing variables

$$\Delta \nu_{\text{bar}} = (1/t) \nu_{\text{bar}} - \nu$$

$$\begin{bmatrix} \frac{1}{t} H_{\text{bar}} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{bar}} \\ \Delta \nu_{\text{bar}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + \frac{1}{t} \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) + A^T \nu \\ r_{\text{pri}} \end{bmatrix},$$

The right hand sides are identical.

The only difference are

$$H_{\text{pd}} = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) + \sum_{i=1}^m \frac{\lambda_i}{-f_i(x)} \nabla f_i(x) \nabla f_i(x)^T.$$

$$\frac{1}{t} H_{\text{bar}} = \nabla^2 f(x) + \sum_{i=1}^m \frac{1}{-tf_i(x)} \nabla^2 f_i(x) + \sum_{i=1}^m \frac{1}{tf_i^2(x)} \nabla f_i(x) \nabla f_i(x)^T.$$

When x, λ, ν satisfy $-f_i(x)\lambda_i = 1/t$, the coefficient matrices (and hence directions) coincide.

The surrogate duality gap

- In the primal-dual interior point methods, the iterates $x^{(k)}, \lambda^{(k)}, \nu^{(k)}$ are not necessarily feasible, except in the limit as the algorithm converges.
- Hence, cannot easily evaluate duality gap $\eta^{(k)}$ in the k th step, as we do in the outer loop of the barrier method.
- Instead, we define the **surrogate duality gap**, for any x that satisfied $F(x) < 0$ and $\lambda \geq 0$ as

$$\eta(x, \lambda) = -F(x)^T \lambda.$$

- The surrogate gap is the duality gap if x were primal feasible and λ, μ were dual feasible i.e. if $r_{\text{prim}} = 0$, $r_{\text{dual}} = 0$. Note that value of t corresponds to the surrogate duality gap $\eta \approx m/t \rightarrow t = m/\eta$.

Primal-dual interior point

Require: x that satisfies $F(x) < 0$, $\lambda > 0$

Require: $\mu > 1$

Require: Tolerances $\epsilon_{\text{feas}} > 0$, $\epsilon > 0$

1: **repeat**

2: Determine t : $t := \mu m / \eta$

3: Compute primal-dual search direction Δy_{pd}

4: Line search: determine step length $s > 0$ and set

$y := y + s \Delta y_{\text{pd}}$

5: **until** $\|r_{\text{prim}}\| \leq \epsilon_{\text{feas}}$, $\|r_{\text{dual}}\| \leq \epsilon_{\text{feas}}$ and $\eta \leq \epsilon$

Remarks

- The parameter t is set to a factor $\mu m/\eta$, which is the value of t associated with the current surrogate duality gap η . If x, λ, ν were central, with parameter t (and therefore with duality gap m/t), then we would increase t by the factor μ (as in the barrier method).
- Values of the parameter μ on the order of 10 appear to work well.
- The primal-dual interior-point algorithm terminates when x is primal feasible and λ, ν are dual feasible (within the tolerance ϵ_{feas}) and the surrogate gap is smaller than the tolerance ϵ . Since the primal-dual interior-point method often has faster than linear convergence, it is common to choose $\epsilon_{\text{feas}}, \epsilon$ small.

- The line search in the primal-dual interior point method is a standard backtracking line search, based on the norm of the residual, and modified to ensure that $\lambda > 0$ and $F(x) < 0$.
- Start with $s_{\max} = \sup\{s \in [0, 1] : \lambda + s\Delta\lambda \geq 0\}$, multiply by $\rho \in (0, 1)$ until $F(x + s\Delta x_{\text{pd}}) < 0$. Continue multiplying until we have

$$\|r_t(x + s\Delta x_{\text{pd}}, \lambda + s\Delta\lambda_{\text{pd}}, x + s\Delta\nu_{\text{pd}})\| \leq (1 - \alpha s)\|r_t(x, \lambda, \nu)\|.$$

Common choices for backtracking parameters are same as for Newton method α in the range 0.01 to 0.1 and ρ 0.3 to 0.8.

- One iteration of the primal-dual interior-point algorithm is the same as one step of the infeasible Newton method, applied to solving $r_t(x, \lambda, \nu) = 0$, but modified to ensure $\lambda > 0$ and $F(x) < 0$ (or, equivalently, with $\text{dom } r_t$ restricted to $\lambda > 0$ and $F(x) < 0$). The same arguments used in the proof of convergence of the infeasible start Newton method show that the line search for the primal-dual method always terminates in a finite number of steps.

Numerical Optimisation

Nonsmooth optimisation

Marta M. Betcke

m.betcke@ucl.ac.uk,

Kiko Rullan, Bolin Pan

f.rullan@cs.ucl.ac.uk, bolin.pan.15@ucl.ac.uk

Department of Computer Science,
Centre for Medical Image Computing,
Centre for Inverse Problems
University College London

Lecture 16

Subgradient

For convex differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ it holds

$$f(y) \geq f(x) + \nabla f(x)^T(y - x).$$

A vector $g \in \mathbb{R}^n$ is a **subgradient** of a function f at $x \in \text{dom } f$ if

$$f(y) \geq f(x) + g^T(y - x) \quad \forall y \in \text{dom } f.$$

- $f(x) + g^T(y - x)$ is affine global underestimator
- g is a subgradient of f at x if $(g, -1)$ supports the epigraph of f at $(x, f(x))$

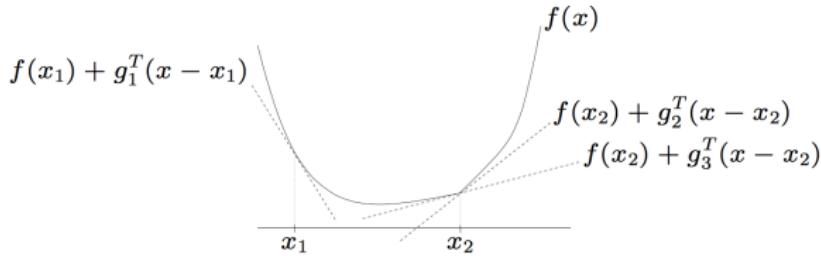


Figure: $\partial f(x_1) = \{\nabla f(x_1)\} = \{g_1\}$, $\partial f(x_2) = [g_3, g_2]$. Fig. from S. Boyd, EE364b, Stanford University.

Subdifferential

A function f is called **subdifferentiable** at x if there exists at least one subgradient at x . Note that subgradient of a non-convex function will be empty in the “non-convexity region”.

Subdifferential of f at x , $\partial f(x)$, is the set of all subgradients of f at x .

$\partial f(x)$ is a closed convex set (can be empty) even if f is not convex.

Proof: It follows from it being intersection of infinite set of halfspaces

$$\partial f(x) = \bigcap_{z \in \text{dom } f} \{g : f(z) \geq f(x) + g^T(z - x)\}.$$

If f is continuous at x , then the $\partial f(x)$ is bounded.

If $f(x)$ is **convex** and $x \in \text{relint dom } f$

- $\partial f(x)$ is nonempty and bounded
- $\partial f(x) = \{\nabla f(x)\}$ iff f differentiable at x (abuse of notation!)

Minimum of nondifferentiable function (unconstraint)

A point x^* is a minimiser of a convex function¹ f iff f is subdifferentiable at x^* and

$$0 \in \partial f(x^*),$$

i.e. $g = 0$ is a subgradient of f at x^* .

Proof: This follows directly from $f(x) \geq f(x^*)$ for all $x \in \text{dom } f$. f is subdifferentiable at x^* with $0 \in \partial f(x^*)$ is equivalent to $f(x) \geq f(x^*) + 0^T(x - x^*)$ for all $x \in \text{dom } f$.

The condition $0 \in \partial f(x^*)$ reduces to $\nabla f(x^*) = 0$ when f is convex and differentiable at x^* . Also in that case it is a necessary and sufficient condition.

¹The definition holds for non-convex functions but the subdifferential will be mostly empty.

Minimum of nondifferentiable function (constraint)

Convex constraint optimisation problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & Ax = b, \end{aligned} \tag{COP}$$

where

- $f, f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convex hence subdifferentiable
- strict feasibility holds (Slater's conditions)

Generalised KKT conditions:

x^* is primal optimal and λ^* dual optimal iff

$$\begin{aligned} & Ax^* = b \\ & f_i(x^*) \leq 0, \\ & \lambda_i^* \geq 0, \\ & 0 \in \partial f(x^*) + \sum_{i=1}^m \lambda_i^* \partial f_i(x^*), \\ & \lambda_i^* f_i(x^*) = 0 \end{aligned} \tag{KKT}$$

Directional derivatives and subdifferential

The **directional derivative** of a function f at x in the direction v

$$f'(x; v) := \lim_{t \rightarrow 0} \frac{f(x + tv) - f(x)}{t}.$$

The limit always exists for convex f but can be $\pm\infty$.

f is differentiable at x iff for some g ($= \nabla f(x)$) and all $v \in \mathbb{R}^n$ we have $f'(x; v) = g^T v$ (i.e. $f'(x; v)$ is a linear function of v).

The directional derivative $f'(x; v)$ of a convex function f satisfies

$$f'(x; v) = \sup_{g \in \partial f(x)} g^T v.$$

Proof idea: Note that $f'(x; v) \geq \sup_{g \in \partial f(x)} g^T v$ by the definition of the subgradient $f(x + tv) - f(x) \geq tg^T v$ for any $t \in \mathbb{R}$ and $g \in \partial f(x)$. Other direction $f'(x; v) \leq \sup_{g \in \partial f(x)} g^T v$ more involved.

Subgradient calculus

Weak subgradient calculus: formulas for finding *one* $g \in \partial f(x)$.

If you can compute f , you can usually compute one subgradient.
Many algorithms require only one subgradient.

Strong subgradient calculus: formula for finding *the whole* subdifferential $\partial f(x)$

Optimality conditions and some algorithms require the whole differential.

Basic rules:

- non-negative scaling: for $\alpha > 0$, $\partial(\alpha f) = \alpha \partial f$
- addition: $\partial(f_1 + f_2) = \partial(f_1) + \partial(f_2)$
- affine transformation: $g(x) = f(Ax + b)$,
 $\partial g(x) = A^T \partial f(Ax + b)$
- finite point wise maximum: $f = \max_{i=1,\dots,m} f_i$,
 $\partial f(x) = \text{Co} \bigcup \{\partial f_i(x) : f_i(x) = f(x)\}$ (convex hull of a union of subdifferentials of active functions at x)

Subgradient and descent direction

p is a descent direction for f at x if $f'(x; p) < 0$.

If f is differentiable, $-\nabla f$ is always a descent direction (except when it is 0).

For a nondifferentiable convex function f , $p = -g$, $g \in \partial f(x)$ need not to be a descent direction.

Example: $f(x) = |x_1| + 2|x_2|$

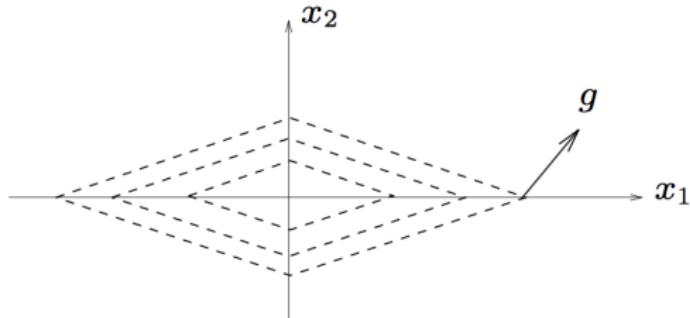


Figure: Fig. from S. Boyd, EE364b, Stanford University.

Subgradient and distance to sublevel set

For a convex f , if $f(z) < f(x)$, $g \in \partial f(x)$, then for small $t > 0$

$$\|x - tg - z\|_2 < \|x - z\|_2.$$

Thus $-g$ is descent direction for $\|x - z\|_2$, for any z with $f(z) < f(x)$.

Proof:

$$\begin{aligned}\|x - tg - z\|_2^2 &= \|x - z\|_2^2 - 2tg^T(x - z) + t^2\|g\|_2^2 \\ &\leq \|x - z\|_2^2 - 2t(\underbrace{f(x) - f(z)}_{>0}) + \underbrace{t^2\|g\|_2^2}_{t: \frac{t}{2}\|g\|_2^2 < f(x) - f(z)}\end{aligned}$$

In particular, choosing $z = x^*$, we obtain that the negative subgradient is a descent direction for the distance to the optimal point x^* .

Proximal operator

Proximal operator of $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$

$$\text{prox}_{\lambda f}(v) := \arg \min_x (f(x) + 1/(2\lambda) \|x - v\|_2^2), \quad \lambda > 0 \quad (\text{PROX})$$

Evaluating prox_f involves minimising closed strongly convex function (unique minimum).

Can evaluate numerically via e.g. BFGS, but often the convex problem (PROX) has an analytical solution or at least a specialised linear-time algorithm.

Indicator function of a closed convex set, $\mathcal{C} \neq \emptyset$

$$I_{\mathcal{C}}(x) = \begin{cases} 0 & x \in \mathcal{C} \\ \infty & \text{otherwise} \end{cases}$$

Proximal operator of $I_{\mathcal{C}}$ is the Euclidean projection

$$\text{prox}_{\lambda I_{\mathcal{C}}}(v) = \arg \min_{x \in \mathcal{C}} \|x - v\|_2 = \Pi_{\mathcal{C}}(v).$$

Many properties of projection carry over to proximal operator.

Examples of proximal operators

Important special choices of f , for which $\text{prox}_{\lambda f}$ has a closed form:

- $f(x) = \frac{1}{2}\|Px - q\|_2^2,$

$$\text{prox}_{\lambda f}(v) = (P^T P + \lambda^{-1} I)^{-1} (P^T q + \lambda^{-1} v).$$

$$(P^T P + Q)^{-1} = Q^{-1} - Q^{-1} P^T (I + P Q^{-1} P^T)^{-1} P Q^{-1}.$$

- f is separable i.e. $f(x) = \sum_i^n f_i(x_i)$, proximal operator acts componentwise

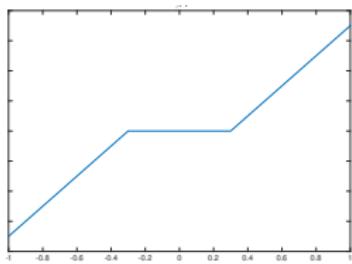
$$(\text{prox}_{\lambda f}(v))_i = \text{prox}_{\lambda f_i}(v_i), \quad i = 1, \dots, n$$

- $f(x) = \|x\|_1$

$$\text{prox}_{\lambda f}(v) = S_\lambda(v),$$

with elementwise soft thresholding

$$S_\delta(x) = \begin{cases} x - \delta & x > \delta \\ 0 & x \in [-\delta, \delta] \\ x + \delta & x < -\delta \end{cases}$$



Examples of proximal operators

Another important example which does not admit close form is Total Variation, $f(x) = TV(x)$, defined as follows

$$\begin{aligned} TV(x) := & \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sqrt{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2} \\ & + \sum_{i=1}^{m-1} |x_{i,n} - x_{i+1,n}| + \sum_{i=1}^{n-1} |x_{m,j} - x_{m,j+1}| \end{aligned}$$

assuming standard reflexive boundary conditions

$$x_{m+1,j} = x_{m,j}, \quad x_{i,n+1} = x_{i,n}.$$

The proximal operator has to be computed iteratively using e.g. Chambolle-Pock algorithm (primal dual proximal gradient).

Resolvent of subdifferential operator

Proximal operator

$$\text{prox}_{\lambda f}(v) = \arg \min_x (f(x) + 1/(2\lambda) \|x - v\|_2^2).$$

The first order condition for the minimiser reads

$$\begin{aligned} 0 &\in \partial f(x) + 1/\lambda(x - v) \\ v - x &\in \lambda \partial f(x) \\ v &\in \lambda \partial f(x) + x \\ \text{prox}_{\lambda f}(v) &= (I + \lambda \partial f)^{-1} v \end{aligned}$$

Mapping $(I + \lambda \partial f)^{-1}$ is called **resolvent** of operator ∂f .

x^* minimises f iff x^* is a fixed point

$$x^* = \text{prox}_f(x^*)$$

Moreau-Yosida regularisation

Moreau envelope or Moreau-Yosida regularisation of f

$$M_{\lambda f}(v) = \inf_x (f(x) + 1/(2\lambda) \|x - v\|_2^2).$$

$M_{\lambda f}$ is a smoothed (regularised) version of f :

- always defined on entire domain
- always continuously differentiable
- has the same minimisers as f

Can show that $M_f = (f^* + 1/2\|\cdot\|_2^2)^*$.

Example: Moreau envelope of $|\cdot|$ is the Huber function

$$M_{|\cdot|}(x) = \begin{cases} \frac{1}{2}x^2 & |x| \leq 1 \\ |x| - \frac{1}{2} & |x| > 1 \end{cases}$$

Moreau decomposition: $v = \text{prox}_f(v) + \text{prox}_{f^*}(v)$ is generalisation of orthogonal decomposition $v = \Pi_W(v) + \Pi_{W^\perp}(v)$.

Forward Backward splitting

$$\min_x f(x) + g(x) \quad \text{subject to } x \in \mathbb{E} \quad (1)$$

- \mathbb{E} : finite dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and a self dual norm $\|\cdot\| = \langle \cdot, \cdot \rangle^{1/2} = \|\cdot\|_*$, e.g. space of $n \times m$ images, $\mathbb{R}^{n \times m}$
- $f : \mathbb{E} \rightarrow \mathbb{R}$ continuously differentiable with Lipschitz continuous gradient,
$$\|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\|, \quad \forall x, y \in \mathbb{E}.$$
- $g : \mathbb{E} \rightarrow (-\infty, \infty]$ proper closed convex.

From first order optimality condition we have

$$\begin{aligned} 0 &\in \nabla f(x^*) + \partial g(x^*) \\ 0 &\in \tau \nabla f(x^*) + \tau \partial g(x^*) - x^* + x^* \\ (I + \tau \partial g)(x^*) &\ni (I - \tau \nabla f)(x^*) \\ x^* &= (I + \tau \partial g)^{-1}(I - \tau \nabla f)(x^*) \end{aligned} \quad (2)$$

Iterative scheme:

$$\begin{aligned}x_k &= \text{prox}_{\tau_k g}(x_{k-1} - \tau_k \nabla f(x_{k-1})) \\&= \arg \min_x \left\{ g(x) + \frac{1}{2\tau_k} \|x - (x_{k-1} - \tau_k \nabla f(x_{k-1}))\|^2 \right\}.\end{aligned}$$

- **Gradient Projection:** $g(x) = l_{\mathcal{C}}(x)$: smooth constrained minimisation, $\tau_k \in (0, 2/L(f))$

$$x_k = \Pi_{\mathcal{C}}(x_{k-1} - \tau_k \nabla f(x_{k-1})).$$

- **Proximal Minimization:** $f(x) = 0$: non-smooth convex minimisation

$$x_k = \arg \min_x \left\{ g(x) + \frac{1}{2\tau_k} \|x - x_{k-1}\|^2 \right\}.$$

- **Iterative Shrinkage Thresholding Algorithm (ISTA):** $g(x) = \|x\|_1$, $f(x) = \|Ax - b\|^2$, $\tau_k \in (0, 2/L(f))$

$$x_k = S_{\tau_k}(x_{k-1} - \tau_k \nabla f(x_{k-1})).$$

Proximal gradient:

$$\begin{aligned} x_k &= \text{prox}_{\tau_k g}(x_{k-1} - \tau_k \nabla f(x_{k-1})) \\ &= \arg \min_x \left\{ g(x) + \frac{1}{2\tau_k} \|x - (x_{k-1} - \tau_k \nabla f(x_{k-1}))\|^2 \right\}. \end{aligned}$$

- **Gradient Projection:** $g(x) = l_{\mathcal{C}}(x)$: smooth constrained minimisation, $\tau_k \in (0, 2/L(f))$

$$x_k = \Pi_{\mathcal{C}}(x_{k-1} - \tau_k \nabla f(x_{k-1})).$$

- **Iterative Shrinkage Thresholding Algorithm (ISTA):**
 $g(x) = \|x\|_1$, $f(x) = \|Ax - b\|^2$, $\tau_k \in (0, 2/L(f))$

$$x_k = S_{\tau_k}(x_{k-1} - \tau_k \nabla f(x_{k-1})).$$

Slow convergence, if $\tau_k = \tau = 1/L$, $L \geq L(f)$, $F(x) := f(x) + g(x)$

$$F(x_k) - F(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2k}.$$

Fast Iterative Shrinkage Thresholding Algorithm (FISTA):

Initialize: $y_1 := x_0 \in \mathbb{E}$, $\tau_1 = 1$.

$$\begin{aligned} \text{Step } k : \quad x_k &= \text{prox}_{1/L}(g) \left(\color{blue}{y_k} - \frac{1}{L} \nabla f(\color{blue}{y_k}) \right) \\ \tau_{k+1} &= \frac{1 + \sqrt{1 + 4\tau_k^2}}{2} \\ \color{blue}{y_{k+1}} &= x_k + \frac{\tau_k - 1}{\tau_{k+1}} (x_k - x_{k-1}). \end{aligned}$$

Convergence, if $\tau_k = \tau = 1/L$, $L \geq L(f)$, $F(x) := f(x) + g(x)$

$$F(x_k) - F(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{(k+1)^2}.$$

Fast Projection Gradient [Nesterov'83]: $g(x) = l_{\mathcal{C}}(x)$

$$x_k = \Pi_{\mathcal{C}} \left(y_k - \frac{1}{L} \nabla f(y_k) \right).$$

More details on Nesterov algorithm see e.g. <http://www.seas.ucla.edu/~vandenbe/236C/lectures/fgrad.pdf>

Review: Optimisation with equality constraints

Let $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, closed, proper and convex.

Primal problem

$$\min_x f(x) \quad \text{subject to } Ax = b$$

Lagrangian

$$\mathcal{L}(x, y) = f(x) + y^T(Ax - b)$$

Dual function

$$g(y) = \inf_x \mathcal{L}(x, y) = -f^*(-A^T y) - b^T y$$

y : dual variable (Lagrange multiplier),

f^* : convex conjugate of f (f^* is convex and closed even if f is not).

Dual problem (always concave, $g(y^*) \leq f(x^*)$, $g(y^*) = f(x^*)$ if strong duality holds)

$$\max_y g(y). \tag{3}$$

Gradient methods

Gradient descent for primal problem (assuming f continuously differentiable)

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

Gradient ascent for dual problem (assuming g continuously differentiable)

$$\begin{aligned}x_{k+1} &= \arg \min_x L(x, y_k) \\y_{k+1} &= y_k + \tau_k \underbrace{(Ax_{k+1} - b)}_{=\nabla g(y_k)}\end{aligned}$$

Remark: the primal update is part of evaluation of $\nabla g(y_k)$

- + for separable f it leads to a parallel algorithm.
- various conditions necessary for convergence e.g. strict convexity of f , $f(x) < \infty, \forall x$.

Augmented Lagrangian

Augmented Lagrangian

$$\mathcal{L}_\rho(x, y) = f(x) + y^T(Ax - b) + \rho/2\|Ax - b\|_2^2, \quad \rho > 0 \quad (\text{AL})$$

Equivalent to Lagrangian of an equivalent problem (for all feasible x the quadratic term equals 0)

$$\min_x f(x) + \rho/2\|Ax - b\|_2^2, \quad \text{subject to } Ax = b.$$

Method of multipliers (MM): dual ascent applied to (AL)

$$\begin{aligned} x_{k+1} &= \arg \min \mathcal{L}_\rho(x, y_k) \\ y_{k+1} &= y_k + \underbrace{\rho(Ax_{k+1} - b)}_{=\nabla g(y_k)} \end{aligned}$$

Using ρ as the step size guarantees **dual feasibility of (x_{k+1}, y_{k+1})** :

$$0 = \nabla_x \mathcal{L}_\rho(x_{k+1}, y_k) = \nabla f(x) + A^T y_k + \rho A^T (Ax - b) \Big|_{x=x_{k+1}} = \nabla f(x_{k+1}) + A^T y_{k+1} =: s_{k+1} = 0.$$

- + converges under more general conditions
- augmented Lagrangian is non-separable.

Alternating Directions Methods of Multipliers (ADMM)

Blend separability of dual ascent with superior convergence of MM:

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x) + g(z) \quad \text{subject to } Ax + Bz = c \quad (4)$$

with $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, $c \in \mathbb{R}^p$ and $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ closed, proper and convex.

The equality constraint comes from the split of the variable into x and z with the objective function separable across the splitting.

Augmented Lagrangian

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \rho/2 \|Ax + Bz - c\|_2^2,$$

ADMM

$$x_{k+1} = \arg \min_x L_\rho(x, z_k, y_k)$$

$$z_{k+1} = \arg \min_z L_\rho(x_{k+1}, z, y_k)$$

$$y_{k+1} = y_k + \rho(Ax_{k+1} + Bz_{k+1} - c).$$

Alternating Directions Methods of Multipliers (ADMM)

Blend separability of dual ascent with superior convergence of MM:

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x) + g(z) \quad \text{subject to } Ax + Bz = c \quad (4)$$

with $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, $c \in \mathbb{R}^p$ and $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ closed, proper and convex.

The equality constraint comes from the split of the variable into x and z with the objective function separable across the splitting.

Augmented Lagrangian

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \rho/2 \|Ax + Bz - c\|_2^2,$$

Dual ascent on \mathcal{L}_ρ (joint minimisation)

$$(x_{k+1}, z_{k+1}) = \arg \min_{x, z} L_\rho(x, z, y_k)$$

$$y_{k+1} = y_k + \rho(Ax_{k+1} + Bz_{k+1} - c).$$

ADMM: scaled form

Augmented Lagrangian

$$\begin{aligned} L_\rho(x, z, y) &= f(x) + g(z) + y^T \underbrace{(Ax + Bz - c)}_{=:r} + \rho/2 \| \underbrace{Ax + Bz - c}_r \|_2^2, \\ &= f(x) + g(z) + y^T r + \rho/2 \| r \|_2^2. \\ &= f(x) + g(z) + \rho/2 \| r + u \|_2^2 - \rho/2 \| \underbrace{u}_{:= (1/\rho)y} \|_2^2, \end{aligned}$$

with $u = (1/\rho)y$ the *scaled dual variable*.

ADMM: scaled form

$$\begin{aligned} x_{k+1} &= \arg \min_x f(x) + \rho/2 \| Ax + Bz_k - c + u_k \|_2^2 \\ z_{k+1} &= \arg \min_z g(z) + \rho/2 \| A\cancel{x_{k+1}} + Bz - c + u_k \|_2^2 \\ u_{k+1} &= u_k + Ax_{k+1} + Bz_{k+1} - c. \end{aligned}$$

ADMM convergence

Assume in addition that the unaugmented Lagrangian \mathcal{L} has a saddle point.

For f, g proper, closed, convex, it follows that strong duality holds (no explicit assumptions on A, B, c).

Under these assumptions the ADMM iterates satisfy

- Residual convergence: $r^k \rightarrow 0$ as $k \rightarrow \infty$ i.e. the iterates approach feasibility.
- Objective convergence: $f(x^k) + g(z^k) \rightarrow p^*$ as $k \rightarrow \infty$ i.e. the objective function of the iterates approach the optimal value
- Dual variable convergence: $y^k \rightarrow y^*$ as $k \rightarrow \infty$, where y^* is a dual optimal point.

Note, that x^k, z^k need not converge to optimal points, although such a result can be shown under additional assumptions.

Optimality conditions

Necessary and sufficient optimality conditions for ADMM

$$Ax^* + Bz^* - c = 0 \quad \text{primal feasibility}$$

$$0 \in \partial f(x^*) + A^T y^* \quad \text{dual feasibility}$$

$$0 \in \partial g(z^*) + B^T y^* \quad \text{dual feasibility}$$

As for MM, it follows from $z_{k+1} = \arg \min_z \mathcal{L}_\rho(x_{k+1}, z, y_k)$ that z_{k+1} and y_{k+1} always satisfy the last equation.

From $x_{k+1} = \arg \min_x \mathcal{L}_\rho(x, z_k, y_k)$ we have

$$\begin{aligned} 0 &\in \partial f(x_{k+1}) + A^T y_k + \rho A^T (Ax_{k+1} + Bz_k - c) \\ &= \partial f(x_{k+1}) + A^T (y_k + \rho r_{k+1} + \rho B(z_k - z_{k+1})) \\ &= \partial f(x_{k+1}) + A^T y_{k+1} + \color{blue}{\rho A^T B(z_k - z_{k+1})} \end{aligned}$$

or equivalently

$$s_{k+1} := \color{blue}{\rho A^T B(z_{k+1} - z_k)} \in \partial f(x_{k+1}) + A^T y_{k+1},$$

which can be interpreted as dual feasibility condition and s_{k+1} is the *dual residual* at iteration $k + 1$.

Literature

- S.Boyd, Stanford EE364b
<http://stanford.edu/class/ee364b/lectures.html>
- L. Vandenberghe, UCLA EE236C
<http://www.seas.ucla.edu/~vandenbe/ee236c.html>
- Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, S.Boyd at all, 2010
- A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, A. Beck, M. Teboulle, 2009
- Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems, A. Beck, M. Teboulle, 2009
- A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging, A. Chambolle, T. Pock, 2011