NLP III : Some algorithms

a) Unconstrained optimization.

$$\begin{cases} \min \theta(x) \\ x \in \mathbb{R}^n \end{cases}$$

<u>idea</u>:

Compute a sequence of vectors

$x_1, x_2, \ldots$

so that

$$\theta(x_i) > \theta(x_{i+1})$$

Typically:

$$x_{i+1} = x_i + \alpha \cdot \underbrace{y_i}_{\text{descent direction}}$$

descent direction:

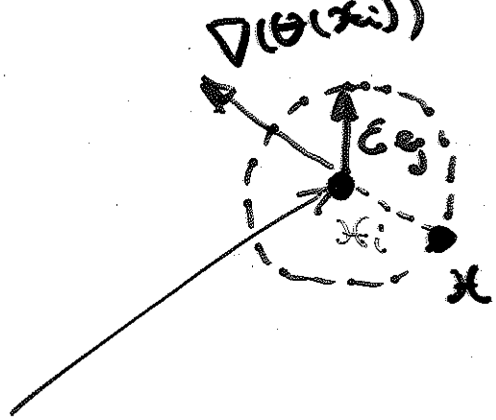
(if $\theta(x_i + \varepsilon y_i) < \theta(x_i)$,

ε - small)

HLP III (c'ed)

Descent direction (c'ed)

If $\theta(x)$ differentiable, y_i : descent direction iff $y_i^T \nabla\theta(x_i) < 0$.



x : point minimizing $\theta(x)$, $x \in$ circle with radius $\varepsilon \to 0$.

Alg

- pick x_1 ; $i \leftarrow 1$.
- while x_i doesn't satisfy stopping condition
  - compute y_i (eg: $-\nabla(\theta(x_i))$)
  - compute $\alpha_i \in \mathbb{R}$.
  - $x_{i+1} \leftarrow x_i + \alpha y_i$; $i \leftarrow i+1$.
- return x_i

HLP III (c'ed)

Stopping condition

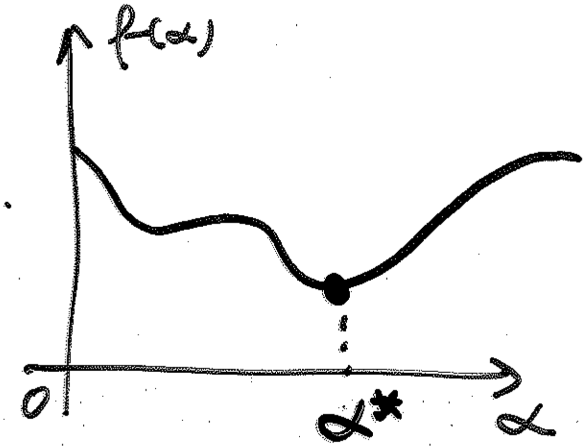
eg $\|\nabla \theta(x_i)\| \to 0$.

or $|\theta(x_i) - \theta(x_{i+1})| \to 0$.

Computing α_i (line search)

Given x_i, y_i define

$$f(\alpha) = \theta(x_i + \alpha y_i).$$



(f decreasing near 0 since y_i is decrease direction)

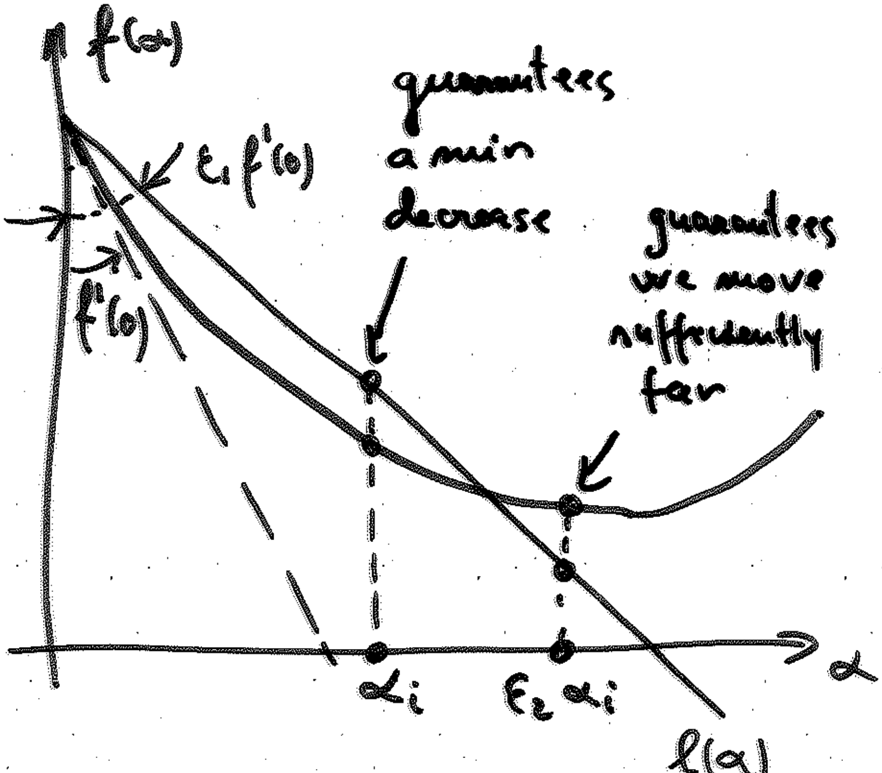
HLP III (c'ed)

Choices for α_i (line search):

- $\alpha_i = h = \text{const.}$
- $\alpha_i = \dfrac{h}{\sqrt{i+1}}$

$\left.\right\}$ independent of $\theta(x)$

- $\alpha_i = \alpha^* = \min f(\alpha)$ (exact line search)
- Armijo's inexact line search:



Chosen α_i is such that:

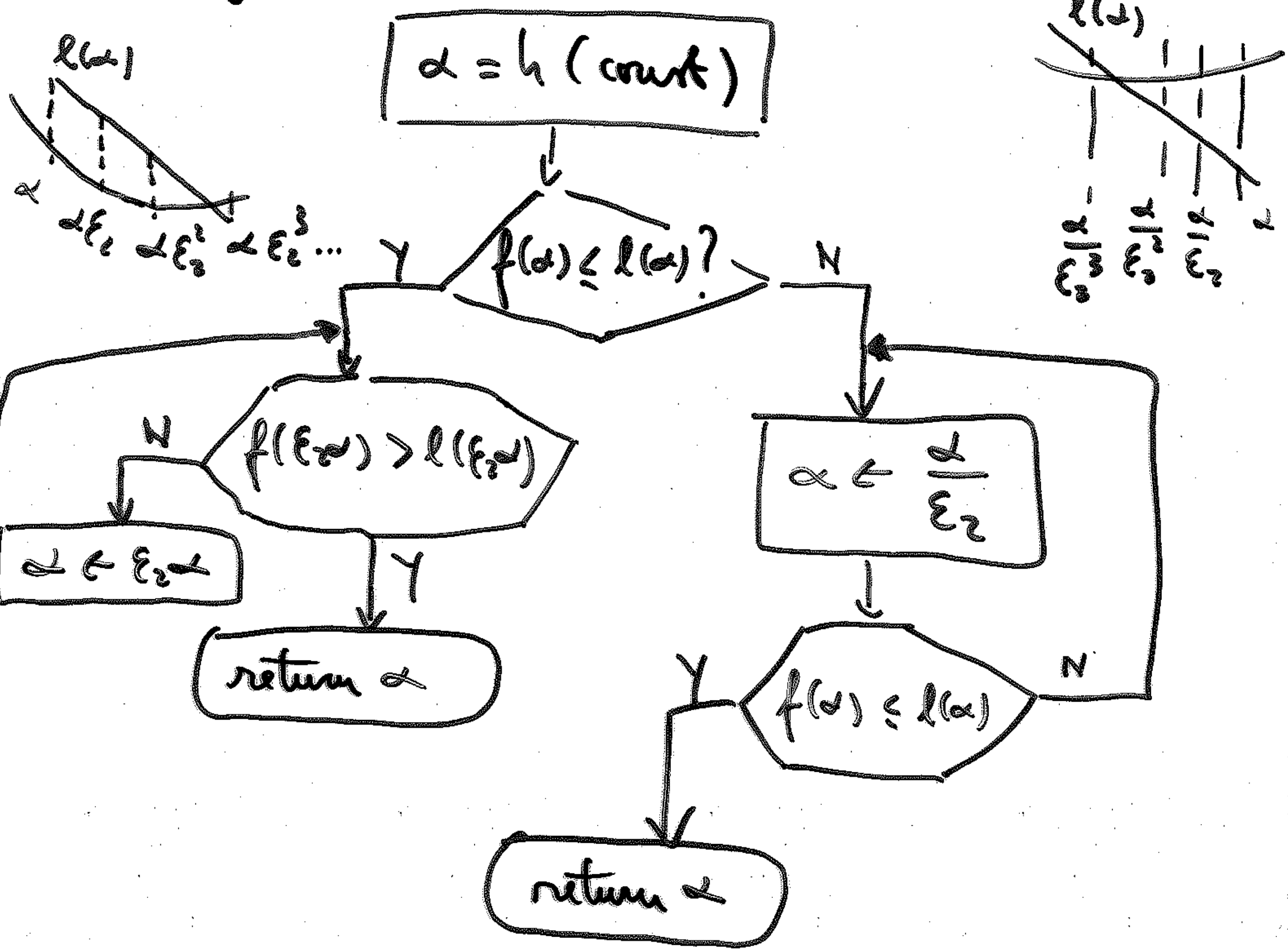
$$\begin{cases} f(\alpha_i) < \ell(\alpha_i) \\ f(\varepsilon_2 \alpha_i) > \ell(\varepsilon_2 \alpha_i) \end{cases}$$

where

$\ell(\alpha) = f(0) + \varepsilon_1 f'(0)$.

NLP III (c'ed)

Armijo's inexact line search (c'ed):



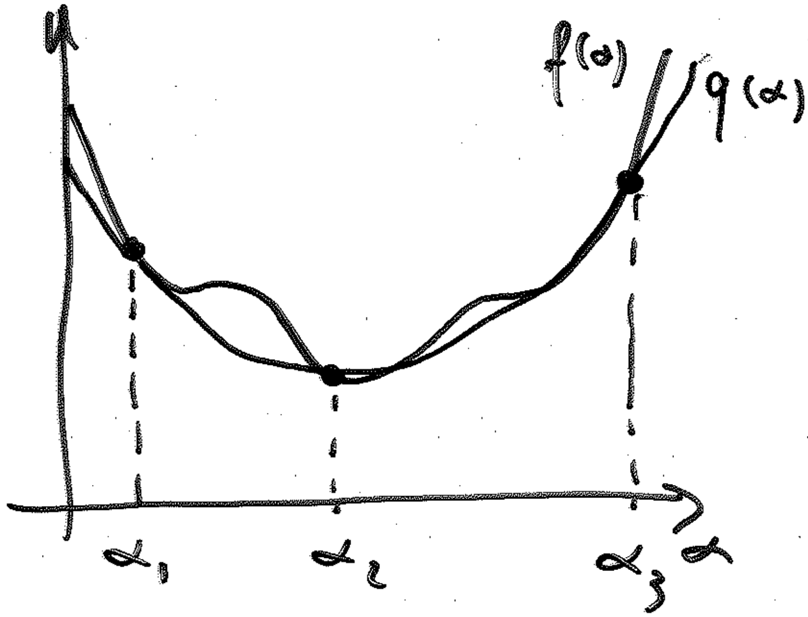
OBS: Typical value for $\varepsilon_2 = 2$ because

$$\underbrace{i}_{\#\text{ of iterations}} = O\left(\max\left\{\log h, \log \frac{\mu}{h}\right\}\right)$$

μ: max stepsize.

NLP III (c'ed)

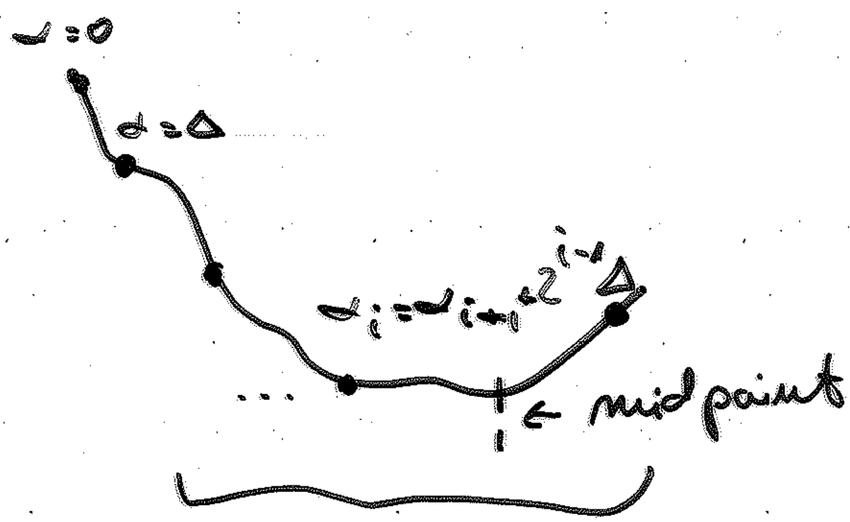
- TPB line search (three point bracket)



a) Find $\alpha_1 < \alpha_2 < \alpha_3$, $f(\alpha_2) < \min\{f(\alpha_1), f(\alpha_3)\}$

b) Compute $q(\alpha) = a\alpha^2 + b\alpha + c$ as interpolation of $f(\alpha_1), f(\alpha_2), f(\alpha_3)$

c) Return $\alpha^* = \min_\alpha q(\alpha)$.



choose $\alpha_1, \alpha_2, \alpha_3$ from last 3 candidates + midpoint.

NLP (III) (cont'd)

Previous algorithm(s) were gradient descent type:

$$y = -\nabla\theta(x_i)$$

(using first derivative information)

Recall Taylor series in one dimension:

$$g(x) = g(a) + g^{(1)}(a)\,(x-a) + \ldots + g^{(k)}(a)\,\frac{(x-a)^k}{k!} + \ldots$$

1-st order approximation:

$$g(x) = g(a) + g^{(1)}(a)\,(x-a) + O(x^2)$$

2-nd order approximation:

$$g(x) = g(a) + g^{(1)}(a)\,(x-a) + \frac{g^{(2)}(a)}{2!}\,(x-a)^2 + O(x^3)$$

NLP III (c'ed)

Approximations with Taylor series in n dimension.

Order 1:

$$f(x+\varepsilon y) = f(x) + \varepsilon \underbrace{y^T \nabla f(x)} + O(\varepsilon^2)$$

if $y^T \nabla f(x) < 0$ then y is a descent direction (gradient methods $y = -\nabla f(x)$)

Order 2:

$$f(x+y) \cong f(x) + y^T \nabla f(x) + \frac{1}{2} y^T H(x)\, y = m(y)$$

$$H(x) = \left(\frac{\partial^2 f(x)}{\partial \xi_i \partial \xi_j} \right)_{ij}, \text{ the Hessian of } f$$

<u>Obs</u>: If $H(x)$ is positive definite (recall NLP I) then $m(y)$ is convex in y and we can compute $y^* = \min_y m(y)$ analytically.

NLP III (cted)

$$\nabla m(y) = \nabla f(x) + H(x)\, y = 0$$

$$\Rightarrow y^{*} = - H(x)^{-1} \cdot \nabla f(x)$$

This is the direction of improvement chosen by the so called Newton methods.

OBS 1 : In Newton type methods, the solution in next iteration is

$$x_{k+1} = x_k - H^{-1}(x_k) \cdot \nabla \theta(x_k).$$

(no step length calculation).

OBS 2 : $H(x_k)$ is tedious to compute.

In quasi-Newton methods, we approximate the Hessian $D_k \approx H(x_k)$ & we update $D_{k+1} \approx H(x_{k+1})$ from D_k.

HLP III (cont'd)

Q: Is y^* a descent direction?

$$y^{*T} \nabla \theta(x) = y^{*T}\left(-H(x)\cdot y^*\right) =$$

$$= -y^{*T} H(x)\, y^* < 0$$

if $H(x)$ is positive definite.

idea behind quasi-Newton methods:

1-D: $\quad f(x) = f(a) + \int_a^x f'(t)\,dt$

by extension to n-D:

$$\nabla\theta(x+y) = \nabla\theta(x) + \underbrace{\int_0^1 H(x+t\cdot y)\, y\, dt}_{+H(x)y - H(x)y}$$

(t is scalar)

$$\Rightarrow \boxed{\nabla\theta(x+y) \cong \nabla\theta(x) + H(x)\cdot y}$$

NLP III (c'nd)

Quasi-Newton (c'nd):

Choose matrices D_k to approximate $H(x_k)$ so that D_k satisfies relation:

$$\boxed{\nabla\theta(x_{k+1}) - \nabla\theta(x_k) = D_k\,(x_{k+1}-x_k)}$$

Start with $D_0 = I_n$, apply certain formulae attempting

(BFGS formula, see text; symmetric rank one formula, etc).

NLP III (c'td)

Constrained optimization:

$$\begin{cases} \min \theta(x) \\ h_i(x) = 0 \quad i = 1 \dots m \\ g_p(x) \geq 0 \quad p = 1 \dots t \end{cases}$$

Penalty function

$$f(x, \lambda) = \theta(x) + \lambda \Big(\sum_{i=1}^{m} h_i^2(x) + \sum_{p=1}^{t} \max\{0, -g_p(x)\} \Big) = \theta(x) + \lambda P(x)$$

- given λ, solve unconstrained problem $\min_{x \in \mathbb{R}^n} f(x, \lambda)$

- if $P(x) > \varepsilon$, increase λ, repeat otherwise STOP.