Application of Algs. designed for quadratics to general nonlinear V

Even for nasty, differentiable, V :
s.d. algs. give

$$V(x_{j+1}) = V(x_j) \quad \text{if } x_j \neq \text{a l.m. } \hat{x}$$

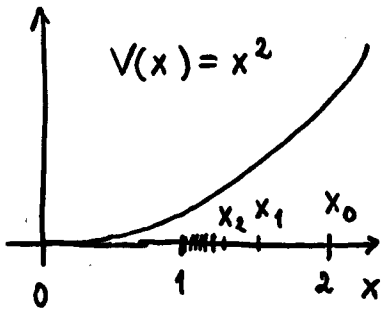
Makes one think that #

always $x_j \xrightarrow[j\to\infty]{} [\text{a l.m. } \hat{x}]$

WRONG !

e.g. $F = \mathbb{R}$, $\hat{x} = 0$

Alg : $x_j = 1 + \frac{1}{j+1}$, $\forall j \geq 0$



$x_j \to 1 \neq \hat{x}$
even though
$V(x_{j+1}) < V(x_j)$, $\forall j$

because the cost decreases at each iteration whenever $x_j \neq \hat{x}$

For a general function V :

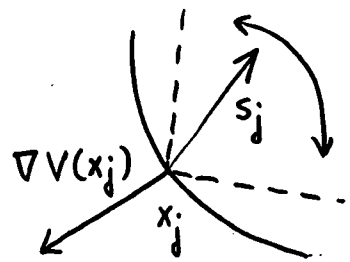
- impossible to guarantee convergence of x_j to a l. m. $\hat{x}$

BUT

- convergence to a l. m. $\hat{x}$ is encouraged by using , $\forall j$:

① w_j = a fairly good approximation to $\hat{w}_j$ which minimizes V exactly along s_j

② an s_j satisfying an angle condition with respect to $\nabla V(x_j)$, i.e. a cone-condition : ↓



① Determination of suitable ω_j

- For standard quadratic V:

$$\hat{\omega}_j := \arg\min_{\omega \in \mathbb{R}^n} V(x_j + \omega s_j)$$

$$= -\frac{[\nabla V(x_j)^T s_j]}{s_j^T C s_j}$$

- For more general V:

$\nexists$ nice formula for $\hat{\omega}_j$

One practical way to estimate $\hat{\omega}_j$ is the Armijo Alg.

First note that:

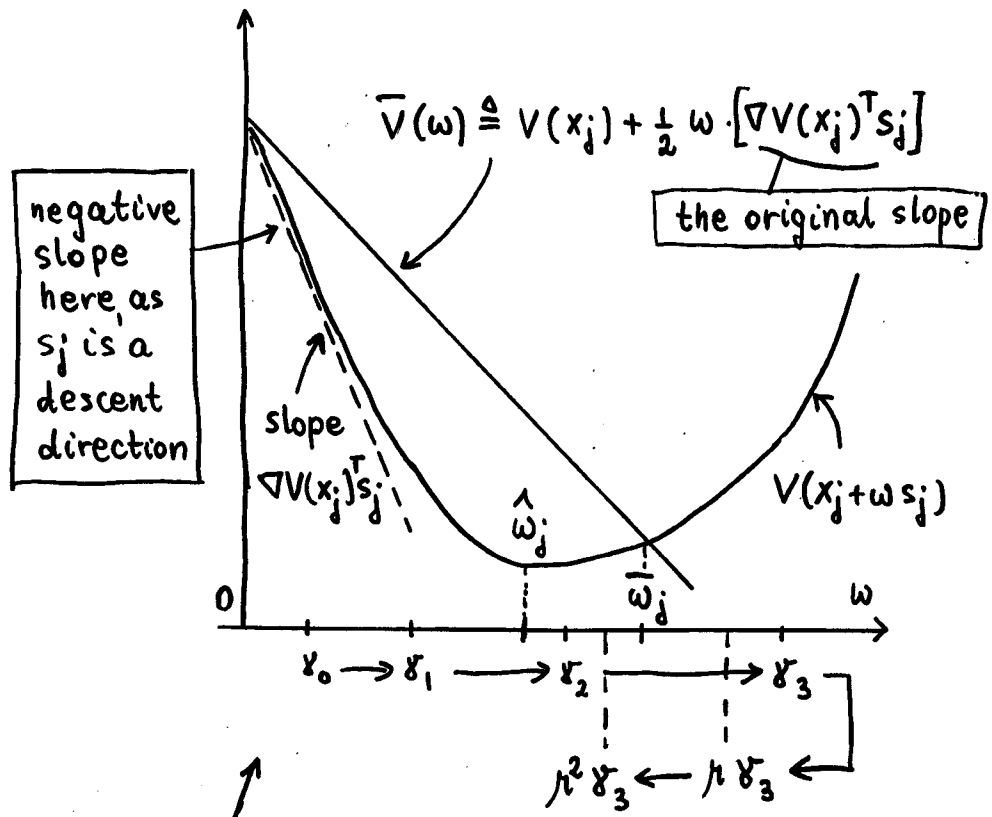
the slope of $V(x_j + \omega s_j)$ at $\omega = 0$ is:

$$\frac{d}{d\omega} V(x_j + \omega s_j)\Big|_{\omega=0} = \nabla V(x_j + \omega s_j)^T s_j\Big|_{\omega=0}$$

$$= \nabla V(x_j)^T s_j$$

A typical situation with s_j = a descent direction

- a typical plot of $V(x_j + \omega s_j)$ v.s. ω :



Armijo Alg.

(a) Find a ω on right of $\bar{\omega}_j$

Choose $\gamma > 1$ (§) (e.g. $\gamma = 1.5$).

Consider $\omega = \gamma^p$.

Increase p from 0 in unit steps until

$$V(x_j + \gamma^p s_j) \geq \bar{V}(\gamma^p)$$

for the first time.

Then: $\bar{\omega}_j \leq \gamma^p$ [$p = 3$ in example]

(b) Find a ω_j on left of $\bar{\omega}_j$, near $\bar{\omega}_j$

Choose $\mu \in (0,1)$, e.g. $\mu = 0.8$ (£)

Consider $\omega = \mu^q \gamma^p$

Increase q from 0 in unit steps until ...

(§) choose γ reasonably big so that one finds a point on the right of $\bar{\omega}_j$ for a small p i.e. using little work

(£) choose μ reasonably near 1, so $\bar{\omega}_j$ is approximated reasonably accurately.

... until

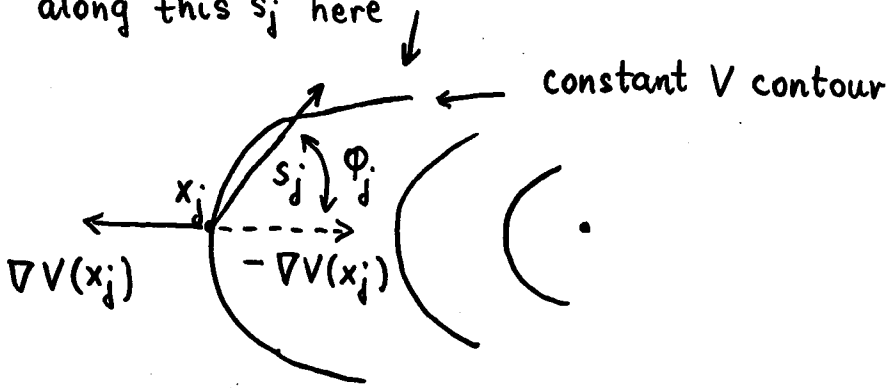
$$V(x_j + \mu^q \gamma^p s_j) \leq \bar{V}(\mu^q \gamma^p)$$

for the first time [$q = 2$ in example]

(c) Set $\omega_j = \mu^q \gamma^p$

(2) The cone condition

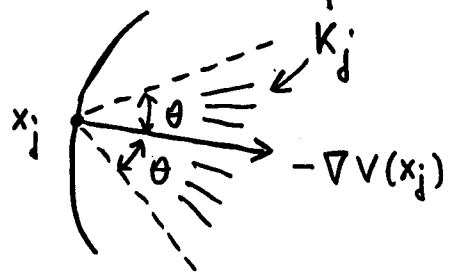
Not likely to get a big cost decrease along this s_j here



this is because s_j passes too near the contour (the tangent to the contour) through x_j,
i.e. because Φ_j = too near 90°

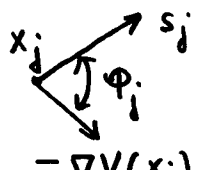
So: use a given s_j iff

$s_j \in$ cone of acceptable search directions



i.e.
$s_j \in K_j$ iff
$\varphi_j < \theta < 90°$
#

Test for $s_j \in K_j$:



(I) $\cos\varphi_j = \dfrac{s_j^T[-\nabla V(x_j)]}{\|s_j\|\,\|\nabla V(x_j)\|}$

(II) $\varphi_j < \theta$
iff $\cos\varphi_j > \cos\theta$

Hence : $s_j \in K_j$ iff :

$s_j^T[-\nabla V(x_j)] > \|s_j\|\,\|\nabla V(x_j)\|\cos\theta$

easy to test

which eliminates search directions passing too near the contour through x_j but not unduly restrictive

Action:
If an algorithm finds an $s_j \notin K_j$
$\Rightarrow$ reset s_j to $-\nabla V(x_j) \in K_j$

$\Rightarrow s_j \in K_j$, $\forall_j$ – encouraging $x_j \to$ l.m. $\hat{x}$

C.G. for non-quadratic V

- C.G. designed for standard quadratics
  – for which $\hat{x} = -C^{-1}b$

⇓

- C.G. not usually used for quadratics

BUT

- C.G. often works well for general V when applied sensibly

see next

Eventually this hope will be realized because functions V usually look like quadratics near a local minimizer

START C.G

C.G. stops at x_j if $\nabla V(x_j) = 0$

If V acting like a quadratic: $\Rightarrow$ $\nabla V(x_j) = 0$ for some $j \leq n$ $\leftarrow$ unlikely for $j < n$

x_0
x_1
x_2
$\vdots$
x_n

If $\nabla V(x_n) \neq 0$, decide that V did not act like a quadratic, so $\rightarrow$ RESTART C.G. with new x_0 = current x_n

AND HOPE # V will act like a quadratic this time

Also: at each j,

IF $s_j \notin K_j$

THEN: restart with new x_0 = current x_j

$\Rightarrow$ s_j used $\in K_j$, $\forall j$ encourages $x_j \rightarrow \ell.m.\hat{x}$

For a general V:

quadratic theory NEVER exactly valid

$\Downarrow$

no need to use exact minimization along each s_j – which is required by CG. alg. for quadratics

$\Downarrow$

Use Armijo to approximate $\hat{w}_j$

---

because then the search direction used is the steepest descent search direction

Use of Secant Alg. with general V.

START with $H_0 = I$, x_0

If V acting like a quadratic: H_1, x_1

$V_{xx}(x_j) = C \ , \ \forall j$ H_2, x_2

$\Downarrow$ $\vdots$

$H_n = C^{-1}$

computed using update formula designed for quadratic V

$\Downarrow$ x_{n+1} should $= \hat{x}$ $\downarrow$

H_{n+1}, x_{n+1}

If $\nabla V(x_{n+1}) \neq 0$, assume it is because V has not so far been acting like a quadratic → RESTART with $H_0 = I$ (**)

new x_0 = current x_{n+1}

Also: if $s_j \notin K_j$ then RESTART with $H_0 = I$

new x_0 = current x_{n+1}

(**) Discard H_{n+1} as it has been computed using an update formula designed for quadratics, and we have just decided that V has not been acting like a quadratic

Remark: Basic ideas involved in applying Algs. designed for quadratics to general V

(I) approximate $\hat{\omega}_j$ using Armijo Alg.

⇒ for Secant Alg. use H_j update scheme valid for approx. minimization along s_j

(II) restart (so as to use S.D. search dir.) whenever:

- alg. performance suggests V has not been acting like a quadratic
- $s_j \notin K_j$

Remark: Performance of algs. for general V

roughly:

- CG better than SD
  (as it is much better on quadratics)
- Secant algs. better than CG
  (as they approximate Newton)
- Newton - good in theory but impractical
  (as finding V_{xx} impractical)
- CG, Secant converge fast to l.m. $\hat{x}$ once they get near $\hat{x}$
  (as usually V looks like a quadratic near $\hat{x}$)

Remarks on calculating $\nabla V(x)$

Sometimes V is so complicated that finding formulae for the first-order p.ds. is too much like hard work

$\Rightarrow$ you may estimate each $\frac{\partial V(x)}{\partial x_i}$ using a finite difference method:

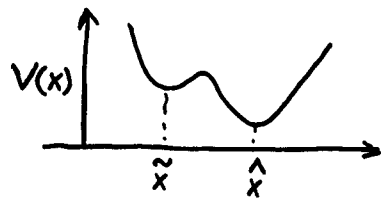
e.g. $\frac{\partial V}{\partial x_1}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) \cong \frac{V\left(\begin{bmatrix} x_1+h \\ x_2 \end{bmatrix}\right) - V\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)}{h}$

where h is chosen:

- small enough — so that a good approximation to the p.d. is obtained
- big enough — so that numerical noise does not give big errors

Choice of h is problem-dependent

(5.1) A sufficient condition for unconstrained local optimality



Th (2.6) says:

$$[\tilde{x} = \text{a local min'zer}] \Rightarrow [\nabla V(\tilde{x}) = 0]$$

↑ a necessary condition for local optimality

"Stronger" result is:

(5.2) Th

$$\left\{ \begin{array}{l} \nabla V(\tilde{x}) = 0 \\ \text{and} \\ V_{xx}(\hat{x}) > 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \tilde{x} \text{ is a local} \\ \text{minimizer of } V \end{array} \right\}$$

a sufficient condition for local optimality

Plausibilification

For x near $\tilde{x}$:

$$\nabla V(x) \cong \nabla V(\tilde{x}) + \underbrace{\nabla V(\tilde{x})^T}_{0}(x-\tilde{x}) +$$

$$+ \tfrac{1}{2}(x-\tilde{x})^T V_{xx}(\tilde{x})(x-\tilde{x})$$

$$\geq V(\tilde{x}), \ \forall \ x \text{ near } \tilde{x}$$

($V_{xx}(\tilde{x}) > 0$)

i.e. $V(x) \geq V(\tilde{x}), \forall x$ near $\tilde{x}$

$\Rightarrow \tilde{x}$ = a local minimizer of V on $\mathbb{R}^n$

Example $V : \mathbb{R}^2 \to \mathbb{R}$

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

Might guess $\tilde{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is a local minimizer

Is it? Calculate:

$$\nabla V(\tilde{x}) = \begin{bmatrix} \partial V(\tilde{x})/\partial x_1 \\ \partial V(\tilde{x})/\partial x_2 \end{bmatrix}$$

$$\frac{\partial V}{\partial x_1}(x) = -400\,(x_2 - x_1^2)\,x_1 - 2(1 - x_1) = 0 \quad @\ \tilde{x}$$

$$\frac{\partial V}{\partial x_2}(x) = 200\,(x_2 - x_1^2) = 0 \quad @\ \tilde{x} \quad \Rightarrow \nabla V(\tilde{x}) = 0$$

To check if $\tilde{x}$ = a loc. minimizer evaluate $V_{xx}(\tilde{x})$:

$$\frac{\partial^2 V(x)}{\partial x_1 \partial x_1} = -400\,x_2 + 1200\,x_1^2 + 2 = 802 \ @\ \tilde{x}$$

$$\frac{\partial^2 V(x)}{\partial x_1 \partial x_2} = \frac{\partial^2 V(x)}{\partial x_2 \partial x_1} = -400\,x_1 = -400 \ @\ \tilde{x}$$

$$\frac{\partial^2 V(x)}{\partial x_2^2} = 200.$$

So, $V_{xx}(\tilde{x}) = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix} > 0$

since $\det(802) = 802 > 0$

$$\det \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix} > 0$$

Hence: $\tilde{x}$ is a local minimizer of V on $\mathbb{R}^n$