Last time: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, Solve $Ax=b$

$$\Longleftrightarrow \min_{x\in\mathbb{R}^n} f(x), \quad f(x) = \tfrac{1}{2}x^TAx - b^Tx$$

For qudratic $f(x)$, both GD and CG work

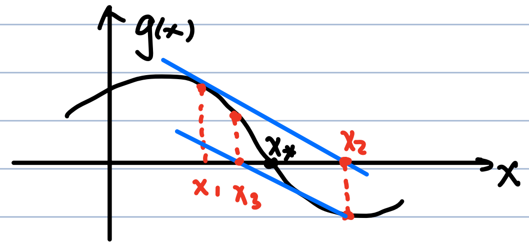
For generic $f(x)$, how to efficiently find minimizer?

$$x_* \in \operatorname{argmin}_{x\in\mathbb{R}^n} f(x) \Longleftrightarrow \nabla f(x_*) = 0 \longleftarrow \text{root finding problem}$$

---

Nonlinear root finder: Newton method

1D:




Idea:

1) Guess $x_k$
2) Find root of tangent line at $x_k$
3) Set $x_{k+1}$ = root

equation of tangent line: $l(x) = g(x_k) + g'(x_k)(x - x_k)$

root $l(x_{k+1}) = 0 \Longleftrightarrow x_{k+1} = \varphi(x_k) := x_k - \dfrac{g(x_k)}{g'(x_k)}$

Newton's method typically converges quadratically,

$e_k := x_k - x_*$, then $|e_{k+1}| \le M|e_k|^2 = O(|e_0|^{2^k})$

for smooth $g$ and $x_0$ sufficiently close to root

If $g'(x_*) = 0$, then convergence is usually linear.

(exercise: if we know $x_*$ is an $m$-fold root of $g(x)$

we can modify $\varphi(x)$ so that $x_{k+1} = x_k - m\dfrac{g(x_k)}{g'(x_k)}$ )

---

Newton's method in $n \geq 1$

Goal: $h: \mathbb{R}^n \to \mathbb{R}^n$, find $h(x) = 0$

$$h(x+\Delta x) = h(x) + J_h(x)\,\Delta x + o(\|\Delta x\|)$$

$$\Rightarrow \quad \Delta x \approx J_h^{-1}(x)\,[\cancel{h(x+\Delta x)} - h(x)]$$

= 0 when
$x+\Delta x$ is root of linear approximation

Newton step $\Rightarrow$ $x_{k+1} = x_k + J_h^{-1}(x_k)\,h(x_k)$

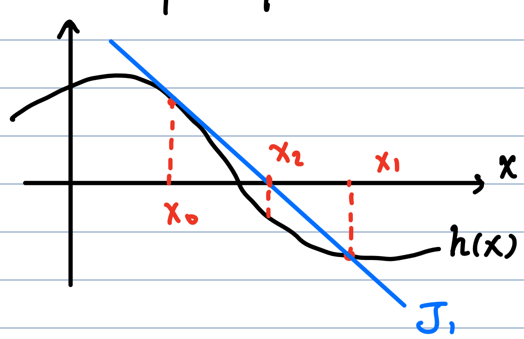
Note: adapts to non-square systems via pseudoinverse or least-squares solve.

---

## Quasi-Newton Methods

Computing the Jacobian and solving $J_h(x)\,\Delta x_k = h(x_k)$ at each step is not always computationally tractable, e.g. when $n$ is very large or computing $h(x)$ is very expensive and $J_h$ is not known analytically.

Idea: approximate $J_k \approx J_h(x_k)$ on the fly, using cheap updates at each iteration




If $x_k \to x_*$, secant line becomes better and better approximation to tangent line

- Secant line requires no explicit knowledge of derivatives of $h$

Secant line equation: $$\underbrace{h(x_{k+1}) - h(x_k)}_{=:\Delta h_k} = J_{k+1}\underbrace{(x_{k+1} - x_k)}_{=:\Delta x_k} \quad (*)$$

for $n > 1$, this does not uniquely determine $J_{k+1}$, so we need to impose more constraints.

<u>No-change condition</u>: A quasi-Newton update must not alter the curvature estimate in directions where the gradient provides no new information.

Formally, $J_{k+1} q = J_k q$ when $q^T \Delta x_k = 0$ (**)

(*) and (**) uniquely determine $J_{k+1}$ from $J_k$:

From (**): $(J_{k+1} - J_k) q = u (\Delta x_k)^T q$ for some $u$, $\forall q$

From (*): $\Delta h_k = (J_k + u (\Delta x_k)^T) \Delta x_k$

$$\Rightarrow \quad u = \frac{\Delta h_k - J_k \Delta x_k}{(\Delta x_k)^T \Delta x_k}$$

So we have

$$J_{k+1} = J_k + \underbrace{(\Delta h_k - J_k \Delta x_k) \frac{\Delta x_k^T}{(\Delta x_k)^T \Delta x_k}}_{\text{rank-1 update to } J_k}$$

equivalently, $J_{k+1} = \arg\min \| J - J_k \|_F$

s.t. (*) holds true

i.e. minimal information update.

Algorithm:

Given $x_0 \in \mathbb{R}^n$, $J_0 \in \mathbb{R}^{n \times n}$, $h_0 = h(x_0) \in \mathbb{R}^n$

for $k = 1, 2, 3, \ldots$

$x_{k+1} = x_k - J_k^{-1} h_k$

$$J_{k+1} = J_k + (\Delta h_k - J_k \Delta x_k) \frac{\Delta x_k^T}{(\Delta x_k)^T \Delta x_k}$$

end

Since we actually need $J_k^{-1}$ at each step, we can use the Sherman–Morrison formula to update $J_{k+1}^{-1}$ from $J_k^{-1}$

$$J_{k+1}^{-1} = J_k^{-1} + (\Delta x_k - J_k^{-1} \Delta h_k) \frac{(\Delta x_k)^T J_k^{-1}}{(\Delta x_k)^T J_k^{-1} \Delta h_k}$$

this is called Broyden's first update

Broyden's second update comes from applying secant and no-change conditions directly to $G_{k+1} = J_{k+1}^{-1}$

Secant condition: $\Delta x_k = G_{k+1} \Delta h_k$ $\quad (*)$

No-change: $G_{k+1} q = G_k q$ when $q^T \Delta h_k = 0$ $\quad (**)$

$$\Rightarrow \quad G_{k+1} = G_k + (\Delta x_k - G_k \Delta h_k) \frac{\Delta h_k^T}{(\Delta h_k)^T \Delta h_k}$$

or equivalently, $G_{k+1} = \arg\min \| G - G_k \|_F$

s.t. $(*)$ holds true

---

BFGS update

Recall in optimization problem $f: \mathbb{R}^n \to \mathbb{R}$, $\min\limits_{x \in \mathbb{R}^n} f(x)$

- Since $x_* \in \arg\min\limits_{x \in \mathbb{R}^n} f(x)$ necessarily requires $\nabla f(x_*) = 0$

Run Newton on $\nabla f$ using Hessian $H = \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{1 \le i, j \le n}$

- Note that $H$ is always symmetric, and at local minima,

$H$ is symmetric positive definite (PSD).

- Notice that Broyden updates are not symmetric. Since we want approximation $H_k \to H$ when $x_k \to x_*$, can we adapt Broyden updates to keep $H_k$ PSD?
- Idea: Use symmetric low-rank update to enforce symmetry while satisfying secant condition.

$$\Delta g_k = \nabla f(x_{k+1}) - \nabla f(x_k), \qquad \Delta x_k = x_{k+1} - x_k$$

Secant equation: $H_{k+1} \Delta x_k = \Delta g_k$ $\qquad (\tilde{*})$

Symm. Rank-2 update: $H_{k+1} = H_k + \alpha u u^T + \beta v v^T$ $\qquad (\widetilde{**})$

with $u = \Delta g_k$ and $v = H_k \Delta x_k$, i.e.

$$\Delta H_k = H_{k+1} - H_k = \alpha \underbrace{(\Delta g_k)(\Delta g_k)^T}_{\substack{\text{symm. type} \\ \text{2 update}}} + \beta \underbrace{(H_k \Delta x_k)(H_k \Delta x_k)^T}_{\substack{\text{symm. type} \\ \text{1 update}}}$$

choose $\alpha, \beta$ to satisfy secant equation $(\tilde{*})$:

$$\Delta g_k = H_k \Delta x_k + \alpha (\Delta g_k)(\Delta g_k)^T \Delta x_k + \beta \underbrace{(H_k \Delta x_k)(H_k \Delta x_k)^T \Delta x_k}_{H_k \Delta x_k (\Delta x_k)^T \cdot H_k \Delta x_k}$$

simply take $\alpha = [(\Delta g_k)^T \Delta x_k]^{-1}$, $\beta = -[(\Delta x_k)^T H_k \Delta x_k]^{-1}$

then

$$\beta (H_k \Delta x_k)(H_k \Delta x_k)^T \Delta x_k = -\frac{H_k \Delta x_k (\Delta x_k)^T H_k \Delta x_k}{(\Delta x_k)^T H_k \Delta x_k} = -H_k \Delta x_k$$

and $(\tilde{*})$ holds true

$$\Rightarrow \quad H_{k+1} = H_k + \frac{\Delta g_k (\Delta g_k)^T}{(\Delta g_k)^T \Delta g_k} - \frac{H_k \Delta x_k (\Delta x_k)^T H_k}{(\Delta x_k)^T H_k \Delta x_k}$$

we can use the Sherman–Morrison–Woodbury formula to get a fast update formula directly for $H_k^{-1} \to H_{k+1}^{-1}$

One can also show that BFGS update is PSD under appropriate restriction on $\Delta x_k$

Storing $H_k$ could be memory intense, instead, store the most recent few $(\Delta g_k, \Delta x_k)$ and use them to compute $H_k$ is call L-BFGS (Limited memory BFGS).

---

Lemma (Sherman-Morrison-Woodbury)

Let $A \in \mathbb{R}^{n\times n}$, $A$ invertible, $u, v \in \mathbb{R}^n$,

then $A + uv^T$ invertible iif $1 + v^T A u \neq 0$

and $$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u}$$