Lecture 7

Hinderance of using $\mathbf{H}^{-1}$, Newton-Raphson, Conjugate Gradients

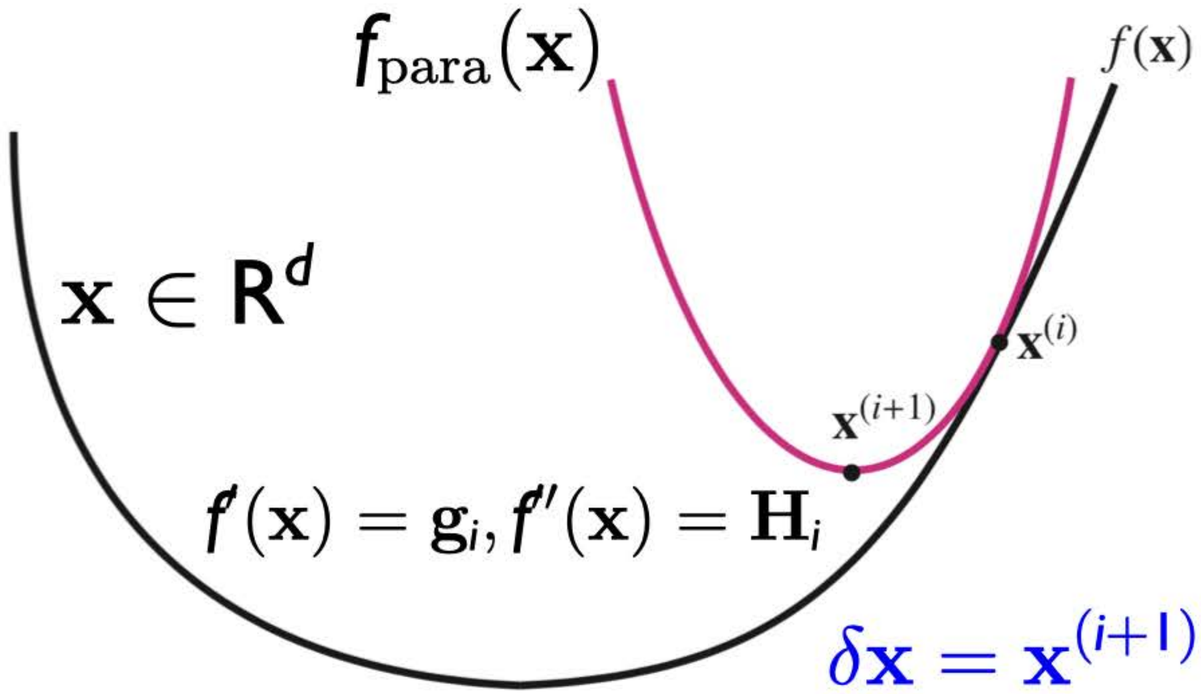
Bao-Jun Cai, 4/15/2026

Introduction to Algorithms for Data Science and Physics IMP@Fudan, 2026

Topics of this lecture:

- Newton's algorithm, initial value issue
- convergence order analysis $e_{i+1} = \phi e_i^{\delta}$
- Gauss-Newton, Levenberg-Marquardt $\mathbf{H} + \zeta \vec{\mathbf{I}} \to \mathbf{H}$
- approximations for the inverse of Hessian $\mathbf{H}_{\text{app}}$
- conditional number kappa $\kappa = \kappa(\mathbf{A}) = |\lambda_{\max}|/|\lambda_{\min}|$
- conjugate gradients search $\kappa \to \sqrt{\kappa}$

Using Hessian to design the optimization



$$f(\mathbf{x}) \approx f_{\text{para}}(\mathbf{x}) = f_i + \mathbf{g}_i^\top \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^\top \mathbf{H}_i \delta\mathbf{x}$$

$$\delta\mathbf{x} = \mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}$$

an approximation of $\mathbf{f}(\mathbf{x})$ around the point $\mathbf{x}^{(i)}$, the same gradient and the same Hessian matrix

GD with exact-line search:

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \epsilon_i \mathbf{g}_i, \ \epsilon_i = \frac{\mathbf{g}_i^\top \mathbf{g}_i}{\mathbf{g}_i^\top \mathbf{H}_i \mathbf{g}_i}$$

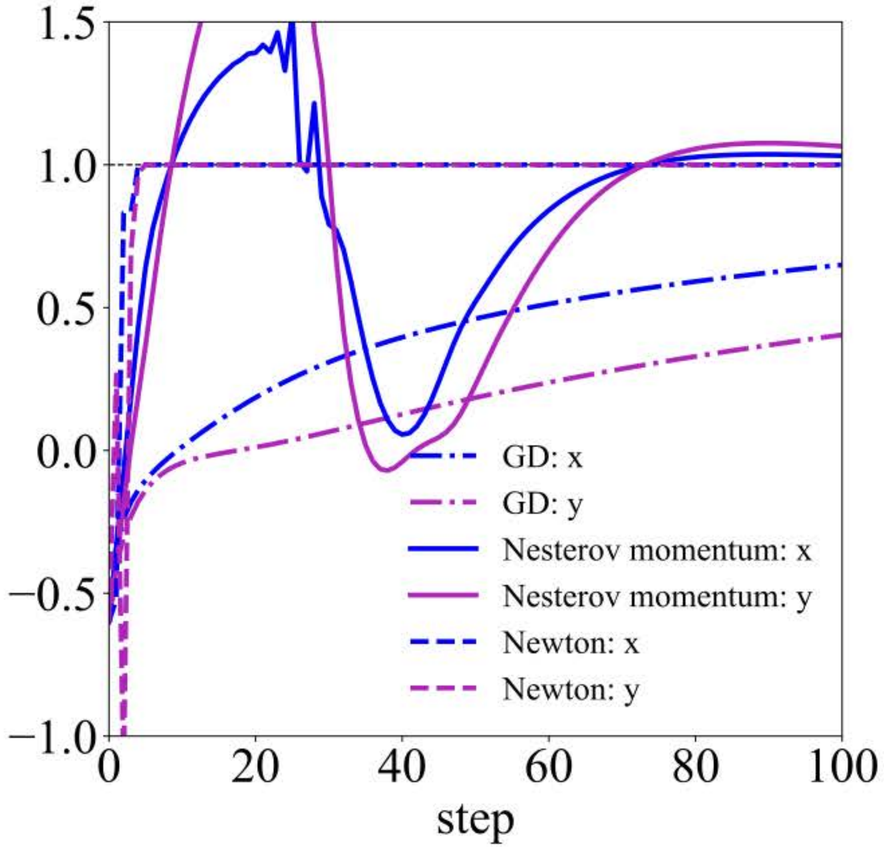
1. first-order in nature
2. $\mathbf{H}_i\mathbf{g}_i$ could be calculated
3. 1D: $x^{(i+1)} = x^{(i)} - g_i/H_i$

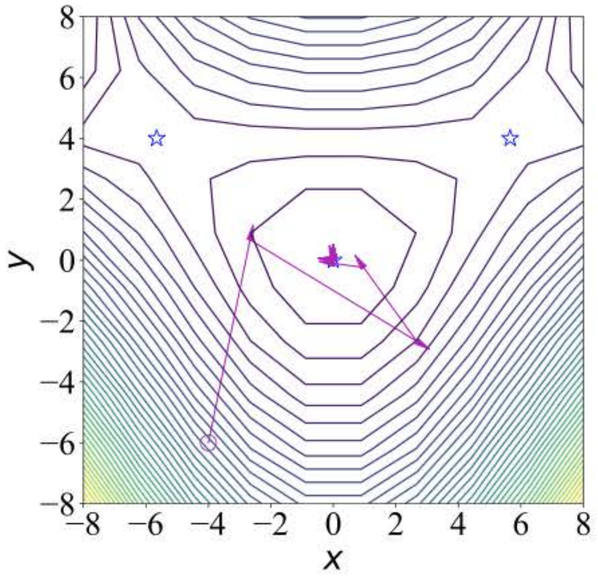
minimum of $f_{\text{para}}(\mathbf{x})$

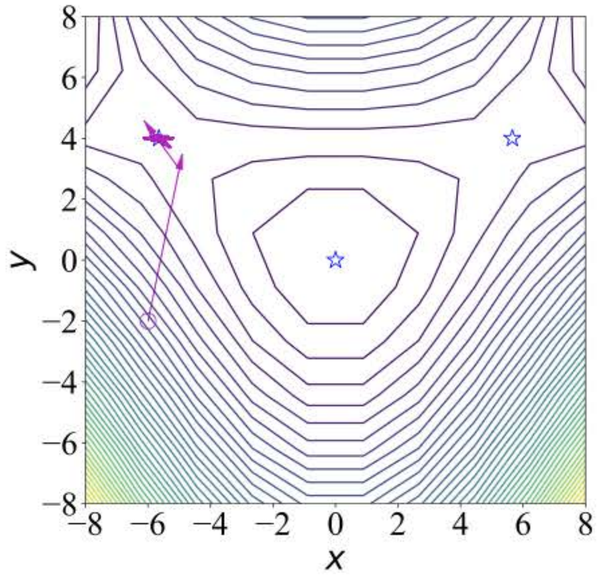
$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \mathbf{H}_i^{-1}\mathbf{g}_i$$

$\mathbf{H}_i^{-1} \sim \mathcal{O}(d^3)$; e.g., $d = 10^6$

Examples







$$f(\mathbf{x}) = 4x^2 + 4y^2 - x^2y$$

$$\mathbf{g} = \begin{pmatrix} 8x - 2xy \\ 8y - x^2 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 8 - 2y & -2x \\ -2x & 8 \end{pmatrix}$$

$$\mathbf{g} = \mathbf{0} \rightarrow (0,0)^\top, (\pm 4\sqrt{2}, 4)^\top$$

$$\mathbf{H} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

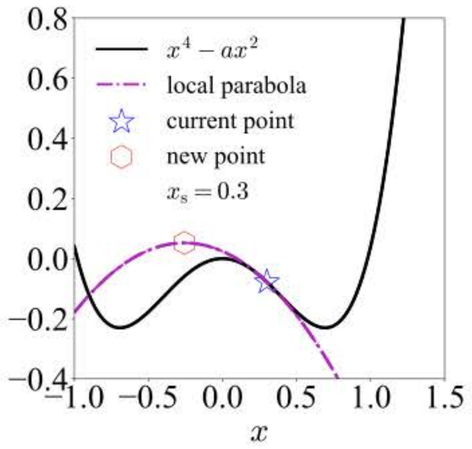
Rosenbrock function

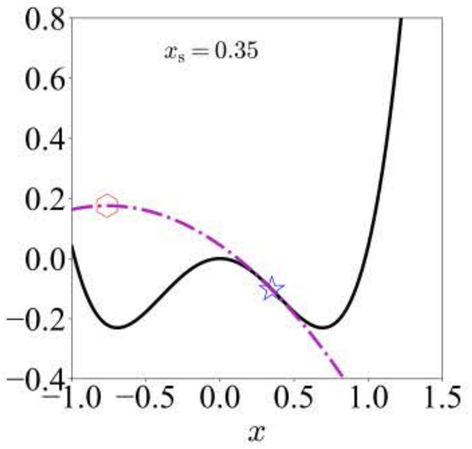
$$\mathbf{H} = \begin{pmatrix} 2 + 12bx^2 - 4by & -4bx \\ -4bx & 2b \end{pmatrix}$$

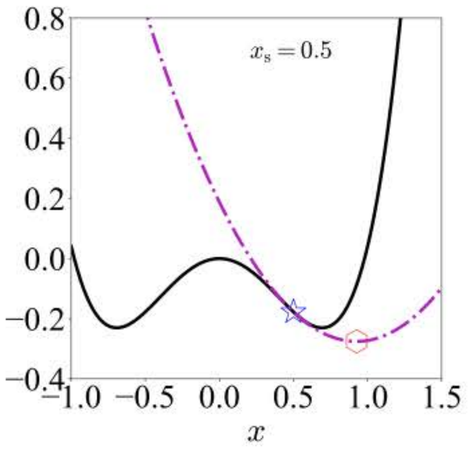
Ex.: write down the general expression for H in 2D

Issue of initial value for Newton algorithm

$$f(x) = x^4 - 0.96x^2$$







theoretical formulae:

$$f_{\text{para}}(x) \approx \frac{1}{2} f_s''(x - x_s)^2 + f_s'(x - x_s) + f_s$$

$$x_{\text{loc}}^{\text{ex}} = \frac{f_s' x_s - f_s'}{f_s''} = x_s - \frac{f_s'}{f_s''} \quad \text{(axis of symmetry)}$$

$$f_{\text{loc}}^{\text{ex}} \equiv f_{\text{para}}(x_{\text{loc}}^{\text{ex}}) = f_s - \frac{f_s'^2}{2f_s''}$$

Ex.: show it

negative-Hessian region:

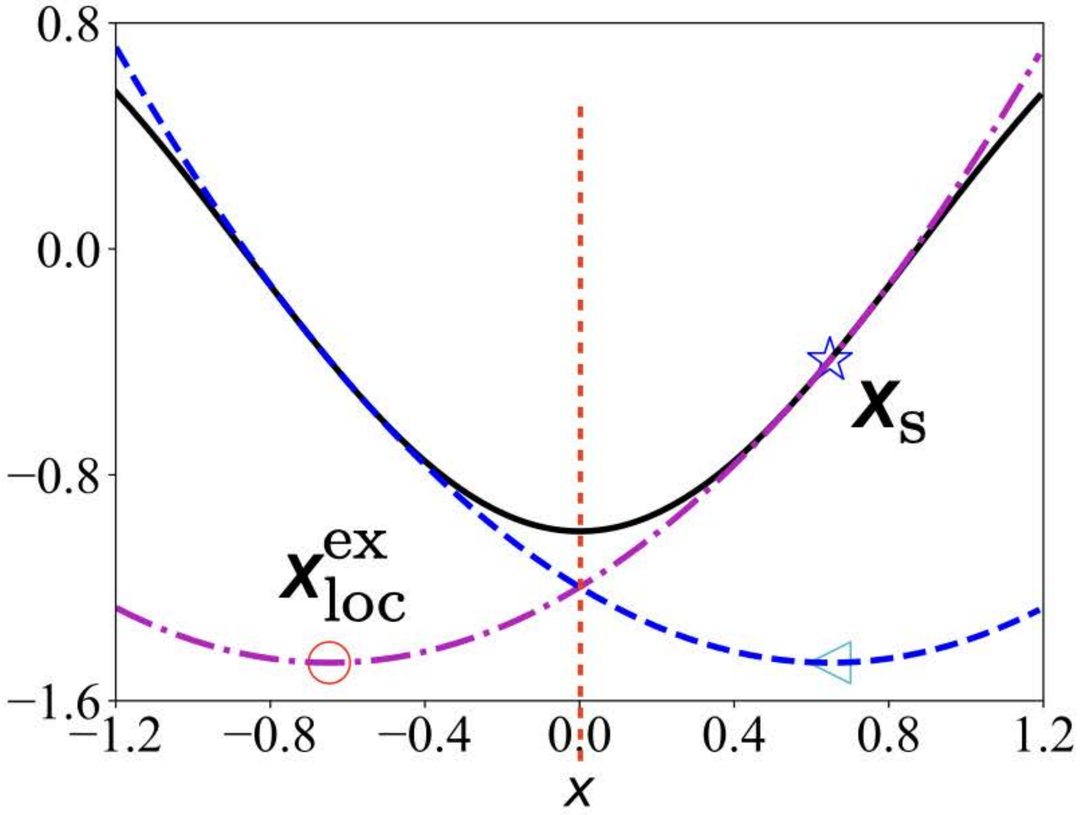
$$-\sqrt{a/6} \le x \le \sqrt{a/6}$$

$$f(x) = x^4 - ax^2$$

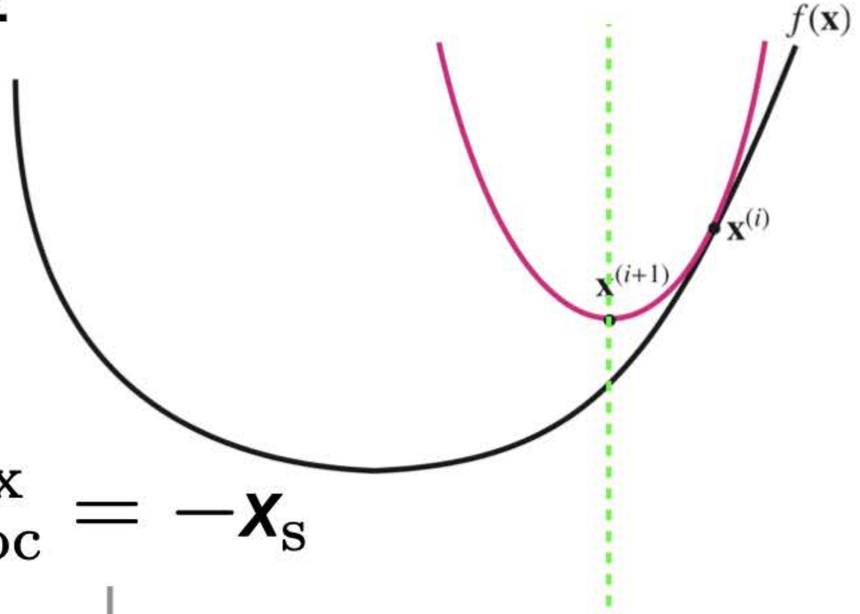
lessons:

1. f_s'' should be positive
2. f_s'/f_s'' should be reasonable
   (selecting a reasonable starting point)

Even more serious case: oscillation



$$f_{para}(x) \approx \frac{1}{2} f_s''(x - x_s)^2 + f_s'(x - x_s) + f_s$$



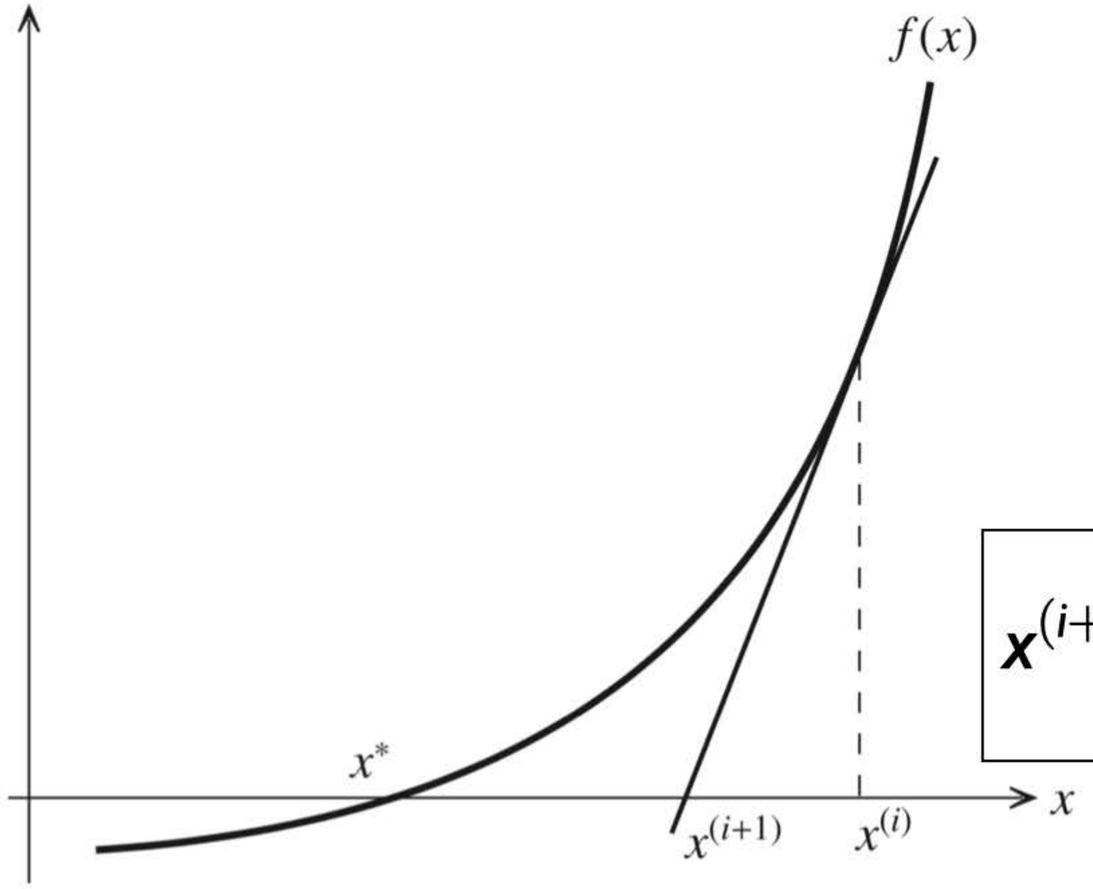
$$x_{loc}^{ex} = -x_s$$

$$f(x) = -\cos ax$$

$$2x_s = f_s'/f_s'' \rightarrow 2ax = \tan ax$$

Another version of Newton method: root solver

finding $x = x^*$ such that $f(x) = 0$



approximate $f(x)$ as

$$f(x) \approx f(x^{(i)}) + f'(x^{(i)})(x - x^{(i)})$$

$$x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$$

example: $x^2 - 5 = 0$

$f'(x) = 2x$

$x^{(i+1)} = x^{(i)}/2 + 5/2x^{(i)}$

$x^{(0)} = 1$

$x^{(1)} = 3$

$x^{(2)} = 2.3333$

$x^{(3)} = 2.2381$

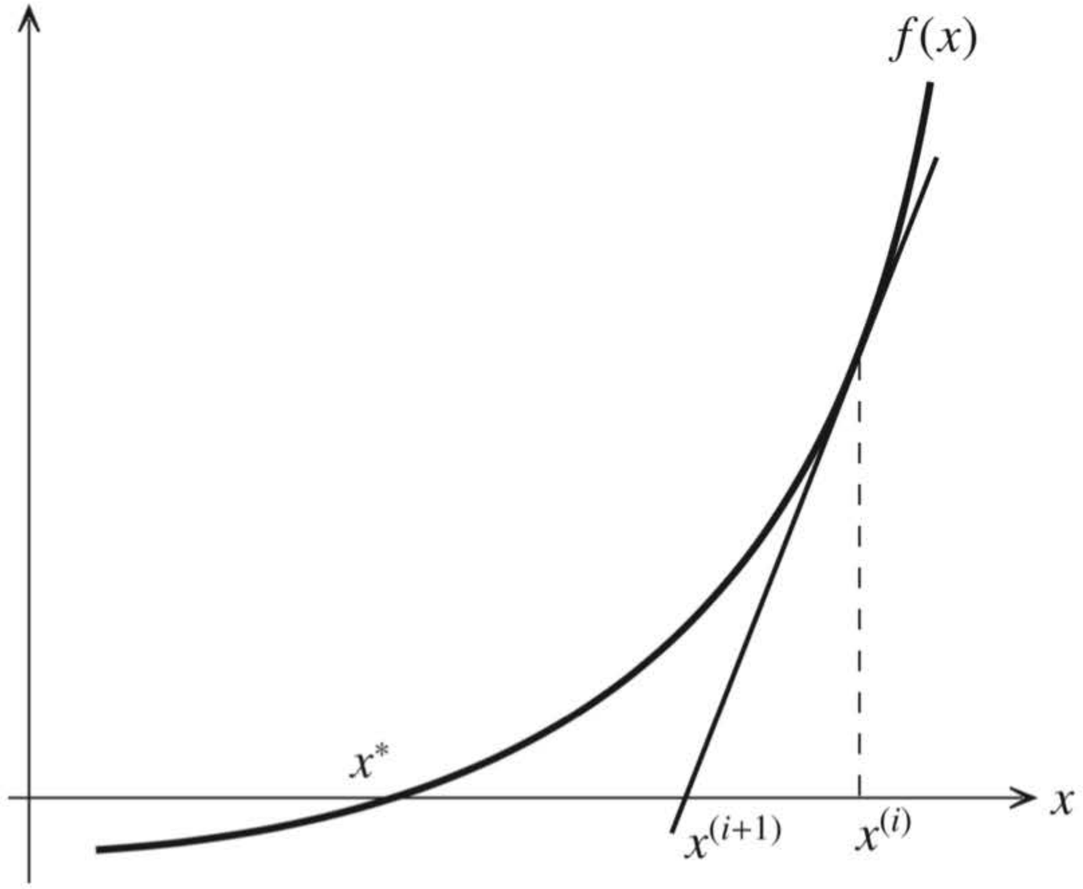
$x^{(4)} = 2.2383$

$x^{(5)} = 2.2361$

...

If the derivative itself is hard to obtain

finding $x = x^*$ such that $f(x) = 0$



$$x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$$

$$f'(x^{(i)}) \approx \frac{f(x^{(i)}) - f(x^{(i-1)})}{x^{(i)} - x^{(i-1)}}$$

$$x^{(i+1)} = x^{(i)} - \frac{(x^{(i)} - x^{(i-1)})f(x^{(i)})}{f(x^{(i)}) - f(x^{(i-1)})}$$

secant method

Error analysis: first glimpse

error at step i: $e_i = x^{(i)} - x^*$

error at step $i+1$: $e_{i+1} = \phi e_i^{\delta}$

index δ effectively characterizes the convergence of the algorithm

1. if δ is large, the convergence is fast
2. if δ is small, the convergence is slow

gradient descent:

$$\begin{aligned} x^{(i+1)} &= x^{(i)} - \epsilon_i f'(x^{(i)}) = x^{(i)} - \epsilon_i f'(x^* + e_i) \\ &\approx x^* + e_i - \epsilon_i [f'(x^*) + e_i f''(x^*)] = x^* + e_i - \epsilon_i f'(x^*) - \epsilon_i e_i f''(x^*) \\ &= x^* + [1 - \epsilon_i f''(x^*)] e_i - \epsilon_i f'(x^*) \end{aligned}$$

$$\boxed{e_{i+1} = [1 - \epsilon_i f''(x^*)] e_i - \epsilon_i f'(x^*)}$$

adopt Taylor's expansion as an effective tool

convergence order: $\delta = 1$

Error in Newton's algorithm

error at step i: $e_i = x^{(i)} - x^*$

error at step $i+1$: $e_{i+1} = \phi e_i^{\delta}$

$$
\begin{aligned}
x^{(i+1)} &= x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})} \approx x^* + e_i - \left(\frac{e_i f'(x^*) + 2^{-1} e_i^2 f''(x^*)}{f'(x^*) + e_i f''(x^*)} \right) \\
&\approx x^* + e_i - \left(e_i f'(x^*) + \frac{1}{2} e_i^2 f''(x^*) \right) \frac{1}{f'(x^*)} \left(1 - e_i \frac{f''(x^*)}{f'(x^*)} \right) \\
&\approx x^* + e_i - \left(e_i + \frac{1}{2} \frac{f''(x^*)}{f'(x^*)} e_i^2 - \frac{f''(x^*)}{f'(x^*)} e_i^2 \right) = x^* + \frac{1}{2} \frac{f''(x^*)}{f'(x^*)} e_i^2
\end{aligned}
$$

$$
\boxed{e_{i+1} = \phi e_i^2, \quad \phi = \frac{f''(x^*)}{2f'(x^*)}}
$$

Ex.: what is the value of convergence index for the secant method?

Quiz 2: 4/15/2026

Quiz 2.1 :

What is the normal equation for $f_{\mathbf{w}}(x) = \mathbf{w}^\top \vec{\phi}(x)$?

(a) $\vec{\Phi}\vec{\Phi}^\top \mathbf{w} = \vec{\Phi}\mathbf{y}$ (b) $\vec{\Phi}^\top \mathbf{w} = \mathbf{y}$ (c) $\vec{\Phi}^\top \vec{\Phi}\mathbf{w} = \vec{\Phi}^\top \mathbf{y}$ (d) $\vec{\Phi}\mathbf{w} = \mathbf{y}$

Write down the form of $\vec{\Phi}$, what is the main advantage of normal equation?

Quiz 2.2 :

Explain the Polyak and Nesterov momentum mechanisms briefly.

Quiz 2.3 :

Write a few algorithms for computing $\mathrm{E}[f] = \int dx f(\mathbf{x}) p(\mathbf{x})$ and give short comments.

Quiz 2.4 :

What is the conditional number κ of a symmetric matrix $\mathbf{A}$?

Gauss-Newton: preparation

error: $e_i(\mathbf{x}) = (\chi(\mathbf{t}^{(i)}, \mathbf{x}) - y^{(i)})^2$

nonlinear model

output

input

parameter

Jacobian

$$\mathbf{J}(\mathbf{x}) = \left(\frac{\partial e_i}{\partial x_j}\right) = \begin{pmatrix} \frac{\partial e_1(\mathbf{x})}{\partial x_1} & \frac{\partial e_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial e_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial e_2(\mathbf{x})}{\partial x_1} & \frac{\partial e_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial e_2(\mathbf{x})}{\partial x_n} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial e_m(\mathbf{x})}{\partial x_1} & \frac{\partial e_m(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial e_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

$\mathbf{e}(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}^m$

total error: $E(\mathbf{x}) = \frac{1}{2}\mathbf{e}^\top(\mathbf{x})\mathbf{e}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m} e_i^2(\mathbf{x})$

gradient

$$\mathbf{g}(\mathbf{x}) \equiv \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}} = \sum_{i=1}^{m} e_i(\mathbf{x})\nabla e_i(\mathbf{x})$$

$$= \mathbf{J}^\top(\mathbf{x})\mathbf{e}(\mathbf{x}) = \begin{pmatrix} e_i(\mathbf{x})\frac{\partial e_i(\mathbf{x})}{\partial x_1} \\ \vdots \\ e_i(\mathbf{x})\frac{\partial e_i(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

Hessian

$$\mathbf{H}(\mathbf{x}) \equiv \frac{\partial^2 E(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}$$

$$= \sum_{i=1}^{m} \left(\nabla e_i(\mathbf{x}) \left(\nabla e_i(\mathbf{x})\right)^\top + e_i(\mathbf{x})\nabla^2 e_i(\mathbf{x}) \right)$$

$$= \mathbf{J}^\top(\mathbf{x})\mathbf{J}(\mathbf{x}) + \mathbf{S}(\mathbf{x})$$

Ex.: how can one approximate $E(\mathbf{x})$ as a quadratic function?

Gauss-Newton: algorithm

$$E(\mathbf{x}) \approx E(\mathbf{x}^{(i)}) + \mathbf{g}^\top(\mathbf{x}^{(i)})\left(\mathbf{x}-\mathbf{x}^{(i)}\right) + \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^{(i)}\right)^\top \mathbf{H}(\mathbf{x}^{(i)})\left(\mathbf{x}-\mathbf{x}^{(i)}\right)$$

$$= \frac{1}{2}\mathbf{e}^\top(\mathbf{x}^{(i)})\mathbf{e}(\mathbf{x}^{(i)}) + \left(\mathbf{J}^\top(\mathbf{x}^{(i)})\mathbf{e}(\mathbf{x}^{(i)})\right)^\top\left(\mathbf{x}-\mathbf{x}^{(i)}\right)$$

$$+ \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^{(i)}\right)^\top\left(\mathbf{J}^\top(\mathbf{x}^{(i)})\mathbf{J}(\mathbf{x}^{(i)}) + \mathbf{S}(\mathbf{x}^{(i)})\right)\left(\mathbf{x}-\mathbf{x}^{(i)}\right)$$

Newton's update

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\mathbf{J}^\top(\mathbf{x}^{(i)})\mathbf{J}(\mathbf{x}^{(i)}) + \mathbf{S}(\mathbf{x}^{(i)})\right)^{-1}\left(\mathbf{J}^\top(\mathbf{x}^{(i)}\mathbf{e}(\mathbf{x}^{(i)})\right)$$

if $\mathbf{S}(\mathbf{x}^{(i)})$ is small

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\mathbf{J}^\top(\mathbf{x}^{(i)})\mathbf{J}(\mathbf{x}^{(i)})\right)^{-1}\left(\mathbf{J}^\top(\mathbf{x}^{(i)}\mathbf{e}(\mathbf{x}^{(i)})\right)$$

Gauss-Newton (GN)

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\mathbf{J}^\top(\mathbf{x}^{(i)})\mathbf{J}(\mathbf{x}^{(i)}) + \zeta_i \mathbf{I}\right)^{-1}\left(\mathbf{J}^\top(\mathbf{x}^{(i)}\mathbf{e}(\mathbf{x}^{(i)})\right)$$

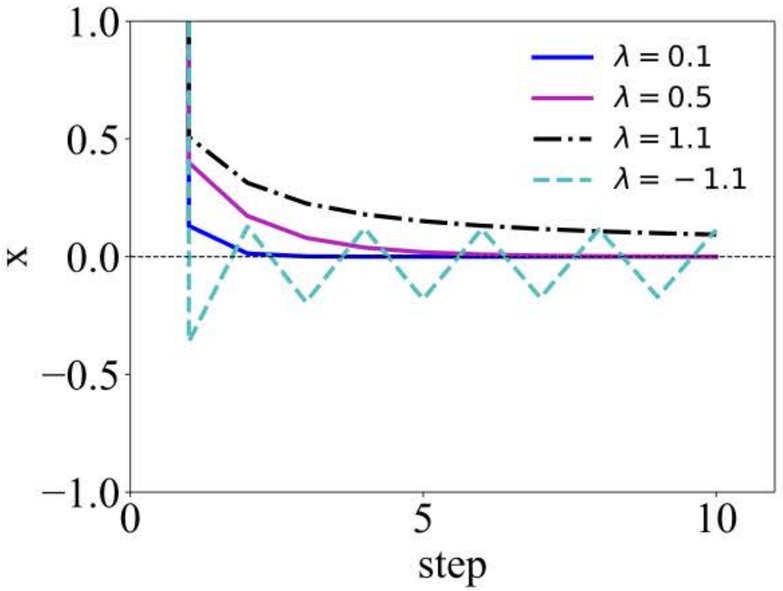
Levenberg-Marquardt (LM)
ζ_i : damping term

example:

$$E(x) = (x+1)^2 + (\lambda x^2 + x - 1)^2$$

$$e_1(x) = \sqrt{2}(x+1),\ e_2(x) = \sqrt{2}(\lambda x^2 + x - 1)$$

$$S(x) = 4\lambda(\lambda x^2 + x - 1) : \lim_{\lambda \to 0} S(x) \to 0$$



Approximations for Hessian (concept)

expand the gradient $\mathbf{g}$ around $\mathbf{x}^{(i+1)}$

$$\delta\mathbf{x} = \mathbf{x} - \mathbf{x}^{(i+1)}$$

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}^{(i+1)} + \delta\mathbf{x}) \approx \mathbf{g}_{i+1} + \mathbf{H}_{i+1}(\mathbf{x} - \mathbf{x}^{(i+1)})$$

$$\downarrow \mathbf{x} = \mathbf{x}^{(i)}$$

$$\mathbf{g}_i \approx \mathbf{g}_{i+1} + \mathbf{H}_{i+1}(\mathbf{x}^{(i)} - \mathbf{x}^{(i+1)})$$

$$\downarrow \mathbf{s}_i = \mathbf{x}^{(i+1)} - \mathbf{x}^{(i)},\ \mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$$

$$\mathbf{H}_{i+1}^{-1}\mathbf{y}_i \approx \mathbf{s}_i$$

(quasi-Newton condition)

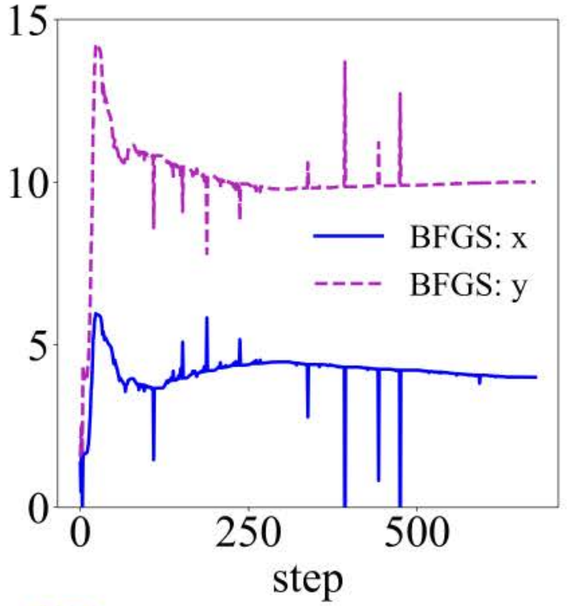
$$\mathbf{W} \approx \mathbf{H}^{-1} \text{ or } \mathbf{B} \approx \mathbf{H}$$

Davidon-Fletcher-Powell (DFP)

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \frac{\mathbf{s}_i\mathbf{s}_i^\top}{\mathbf{s}_i^\top\mathbf{y}_i} - \frac{\mathbf{W}_i\mathbf{y}_i\mathbf{y}_i^\top\mathbf{W}_i}{\mathbf{y}_i^\top\mathbf{W}_i\mathbf{y}_i}$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS)

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \frac{\left(\mathbf{s}_i^\top\mathbf{y}_i + \mathbf{y}_i^\top\mathbf{W}_i\mathbf{y}_i\right)\left(\mathbf{s}_i\mathbf{s}_i^\top\right)}{\left(\mathbf{s}_i^\top\mathbf{y}_i\right)^2} - \frac{\mathbf{W}_i\mathbf{y}_i\mathbf{s}_i^\top + \mathbf{s}_i\mathbf{y}_i^\top\mathbf{W}_i}{\mathbf{s}_i^\top\mathbf{y}_i}$$



Biggs EXP function:

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i^2(\mathbf{x})$$

$$f_i(\mathbf{x}) = ze^{-t_i x} - ue^{-t_i y} + we^{-t_i v} - \phi_i$$

$$\mathbf{x} = (x, y, z, u, v, w)^\top,$$

$$t_i = 0.1i,\ \phi_i = e^{-t_i} - 5e^{-10t_i} + 3e^{-4t_i}$$

$$m \geq 6$$

Conditional number

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x}$$

(gradient-descent)

$$\frac{f(\mathbf{x}^{(i+1)}) - f(\mathbf{x}^*)}{f(\mathbf{x}^{(i)}) - f(\mathbf{x}^*)} \leq \left(\frac{\lambda_1 - \lambda_d}{\lambda_1 + \lambda_d}\right)^2 \equiv \left(\frac{\kappa(\mathbf{A}) - 1}{\kappa(\mathbf{A}) + 1}\right)^2$$

$$\frac{\|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_\mathbf{A}}{\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_\mathbf{A}} \leq \sqrt{\kappa(\mathbf{A})} \cdot \frac{\kappa(\mathbf{A}) - 1}{\kappa(\mathbf{A}) + 1} \leq \frac{\kappa(\mathbf{A}) - 1}{\kappa(\mathbf{A}) + 1}$$

$$\|\mathbf{x}\|_\mathbf{A}^2 = \mathbf{x}^\top \mathbf{A}\mathbf{x}$$

$$\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_\mathbf{A} \leq \Upsilon \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_\mathbf{A} \rightarrow i \leq \frac{1}{2}\kappa(\mathbf{A}) \ln\left(\frac{1}{\Upsilon}\right)$$

conditional number of a matrix

$$\kappa = \kappa(\mathbf{A}) = |\lambda_{\max}|/|\lambda_{\min}|$$

$$\left(\frac{\kappa+1}{\kappa-1}\right)^i > \frac{1}{\Upsilon}, \quad i \geq \ln\left(\frac{1}{\Upsilon}\right) \Big/ \ln\left(\frac{\kappa+1}{\kappa-1}\right)$$

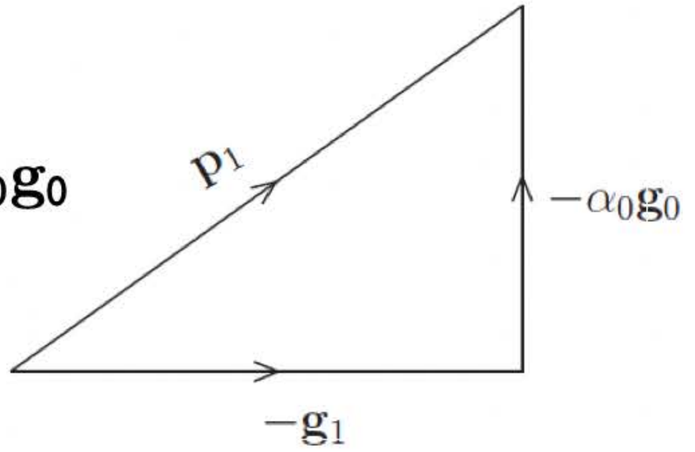
$$\ln\left(\frac{\kappa+1}{\kappa-1}\right) = \ln\left(1 + \frac{2}{\kappa-1}\right) \approx 2\left(\frac{1}{\kappa} + \frac{1}{3\kappa^3} + \cdots\right) \geq \frac{2}{\kappa}$$

Conjugate gradients

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i)} + \epsilon_i \mathbf{p}_i$$

$$\mathbf{p}_1 = -\mathbf{g}_1 - \alpha_0 \mathbf{g}_0$$

$$\mathbf{p}_2 = -\mathbf{g}_2 - \alpha_1 \mathbf{g}_1 - \beta_0 \mathbf{g}_0$$



can we find a better search direction $\mathbf{p}_i \neq \mathbf{g}_i$ (GD)?

conjugate-gradient algorithm:

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \epsilon_i \mathbf{p}_i,\ \mathbf{p}_{i+1} = -\mathbf{g}_{i+1} + \alpha_i \mathbf{p}_i,\ \alpha_i = \frac{\mathbf{g}_{i+1}^\top \mathbf{g}_{i+1}}{\mathbf{g}_i^\top \mathbf{g}_i},\ \mathbf{p}_0 = -\mathbf{g}_0$$

exact-line search:

$$\epsilon_i = -\frac{\mathbf{g}_i^\top \mathbf{p}_i}{\mathbf{p}_i^\top \mathbf{A} \mathbf{p}_i} = \frac{\mathbf{g}_i^\top \mathbf{g}_i}{\mathbf{p}_i^\top \mathbf{A} \mathbf{p}_i}$$

conjugate condition: $\mathbf{p}^\top \mathbf{A} \mathbf{q} = 0$

*Convergence of conjugate-gradient method

$$\{\mathbf{p}_0, \cdots, \mathbf{p}_{n-1}\} \text{ span } \mathbf{R}^n \rightarrow \mathbf{x}^* = \mathbf{x}^{(0)} + \sum_{i=0}^{n-1} \phi_i \mathbf{p}_i$$

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x}$$

$$\mathbf{A} \in \mathbf{R}^{n \times n}$$

on the other hand, $\mathbf{x}^{(i)} = \mathbf{x}^{(0)} + \sum_{j=1}^{i-1} \epsilon_j \mathbf{p}_j,\ 0 \le i \le n$

$$\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)} = \mathbf{A}(\mathbf{x}^* - \mathbf{x}^{(0)}) = \sum_{j=0}^{n-1} \phi_j \mathbf{A}\mathbf{p}_j$$

$$\text{residual } \mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}^{(i)} = \mathbf{r}_0 - \sum_{j=1}^{i-1} \epsilon_j \mathbf{A}\mathbf{p}_j$$

$$\rightarrow \mathbf{p}_i^\top (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}) = \sum_{j=0}^{n-1} \phi_j \overbrace{\mathbf{p}_i^\top \mathbf{A}\mathbf{p}_j}^{\text{force } i=j} = \phi_i \mathbf{p}_i^\top \mathbf{A}\mathbf{p}_i$$

$$\rightarrow \mathbf{p}_i^\top \mathbf{r}_i = \mathbf{p}_i^\top \mathbf{r}_0 - \sum_{j=0}^{i-1} \epsilon_j \mathbf{p}_i^\top \mathbf{A}\mathbf{p}_j = \mathbf{p}_i^\top \mathbf{r}_0 \rightarrow \epsilon_i = \frac{\mathbf{p}_i^\top \mathbf{r}_0}{\mathbf{p}_i^\top \mathbf{A}\mathbf{p}_i}$$

$$\rightarrow \phi_i = \frac{\mathbf{p}_i^\top (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})}{\mathbf{p}_i^\top \mathbf{A}\mathbf{p}_i} = \frac{\mathbf{p}_i^\top \mathbf{r}_0}{\mathbf{p}_i^\top \mathbf{A}\mathbf{p}_i},\ 0 \le i \le n-1$$

$$\boxed{\phi_i = \epsilon_i}$$

CG method terminates at most *n* steps

Role of conditional number in conjugate gradients

$$\frac{\left\|\mathbf{x}^* - \mathbf{x}^{(i)}\right\|_{\mathbf{A}}}{\left\|\mathbf{x}^* - \mathbf{x}^{(0)}\right\|_{\mathbf{A}}} \leq 2\left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1}\right)^i \leq \Upsilon \rightarrow \boxed{i \gtrsim \frac{1}{2}\sqrt{\kappa(\mathbf{A})}\ln\left(\frac{1}{\Upsilon}\right)}$$

1. Initialize $\mathbf{x}^{(0)}$, obtain $\mathbf{r}_0 = -\nabla f(\mathbf{x}^{(0)})$.
2. Obtain $\mathbf{p}_0 = \mathbf{r}_0$, if $\mathbf{p}_0$ is sufficiently small, return $\mathbf{x}^{(0)}$.
3. Calculate the learning rate by line search, $\epsilon_0 = \operatorname{argmin}_\epsilon f(\mathbf{x}^{(0)} + \epsilon\mathbf{r}_0)$.
4. Obtain $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \epsilon_0\mathbf{r}_0$.
5. For general $i \geq 1$, do the following:

   $\mathbf{r}_i = -\nabla f(\mathbf{x}^{(i)})$.

   Search parameter β_{i-1} through certain schemes.

   Update the conjugate direction $\mathbf{p}_i = \mathbf{r}_i + \beta_{i-1}\mathbf{p}_{i-1}$.

   Update the learning rate $\epsilon_i = \operatorname{argmin}_\epsilon f(\mathbf{x}^{(i)} + \epsilon\mathbf{p}_i)$.

   Update $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \epsilon_i\mathbf{p}_i$.
6. Terminate i if $|\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}|$ is sufficiently small.

GD: $i \gtrsim \frac{1}{2}\kappa(\mathbf{A})\ln\left(\frac{1}{\Upsilon}\right)$

CGNE (conjugate-gradient normal equation error)

$\mathbf{A}$ is neither symmetric nor positive-definite

$$\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b},\ \mathbf{A}\mathbf{A}^\top\mathbf{y} = \mathbf{b} \rightarrow \mathbf{x} = \mathbf{A}^\top\mathbf{y}$$

Preconditioning

$\mathbf{B}^{-1}(\mathbf{A}\mathbf{x} - \mathbf{b}) = 0$ where $\kappa(\mathbf{B}^{-1}\mathbf{A})$ is small