Conjugate Gradient Method :-

- iterative method for solving system of linear equation $(Ax = b)$.

- A is large, ~~syst~~ symmetric, positive definite matrix.

Definition :-

<u>Inner product</u> : The ~~#~~ inner product of two column vector $\bar{x}, \bar{y} \in \mathbb{R}^n$ is defined as:

$$\langle \bar{x}, \bar{y} \rangle = \bar{x}^T \bar{y}.$$

$\Rightarrow$ $\bar{x}, \bar{y}$ are called orthogonal if

$$\langle \bar{x}, \bar{y} \rangle = 0.$$

<u>Positive definite</u> : A symmetric matrix A is called positive-definite if for every nonzero $\bar{x}$,

$$\bar{x}^T A \bar{x} = \langle \bar{x}, A\bar{x} \rangle > 0$$

<u>Result</u> :- A is +ve definite $\Leftrightarrow$ all its eigenvalues are +ve.

EXM. $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$ $\quad \lambda = 1, 3.$

<u>EXM</u>. $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ $\quad \lambda = -1, 3,$

$$x = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \Rightarrow x^T A x = -2 < 0.$$

1. $f(x) = x^T A x \qquad f : \mathbb{R}^n \to \mathbb{R}$

$\nabla f = [(A^T + A)x]^T_{1\times n}$

2. $g(x) = b^T x \qquad g : \mathbb{R}^n \to \mathbb{R}$

$= \sum_{i=1}^{n} b_i x_i \qquad (b_1 \quad b_n)\begin{pmatrix} x_1 \\ \\ x_n \end{pmatrix}$

$= b_j x_j + \sum_{i \neq j} b_i x_i$

$\nabla g = (b_1, \ldots, b_n) = b^T_{1\times n}$

3. If $F(x) = \frac{1}{2} x^T A x - b^T x + c$

then $\nabla F = \frac{1}{2} x^T (A + A^T) - b^T$.

For A symmetric,

$\nabla F = x^T A - b^T$

i.e. $(\nabla F)^T = Ax - b$.

1* $f(x) = x^T A x = x^T \begin{pmatrix} \sum a_{1i} x_i \\ \sum a_{2i} x_i \\ \vdots \\ \sum a_{ni} x_i \end{pmatrix}_{n\times 1}$

$= \sum_{k=1}^{n} x_k \left(\sum_{i=1}^{n} a_{ki} x_i \right)$

$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots \frac{\partial f}{\partial x_n} \right)$

To find $\frac{\partial f}{\partial x_j}$, we rewrite,

$\sum_{k=1}^{n} x_k \left(\sum_{i=1}^{n} a_{ki} x_i \right) = x_j \sum_{i=1}^{n} a_{ji} x_i + \sum_{k \neq j} x_k \left(\sum_i a_{ki} x_i \right)$

$= x_j \left(a_{jj} x_j + \sum_{i \neq j} a_{ji} x_i \right)$

$+ \sum_{k \neq j} x_k \left(a_{kj} x_j + \sum_{i \neq j} a_{ki} x_i \right)$

$$= a_{JJ} x_J^2 + x_J \left(\sum_{i \neq J} a_{Ji} x_i \right)$$

$$+ \left(\sum_{K \neq J} x_K a_{KJ} \right) x_J + \sum_{K \neq J} x_K \sum_{i \neq J} a_{Ki} x_i$$

So, $$\frac{\partial f}{\partial x_J} = 2 a_{JJ} x_J + \sum_{K \neq J} a_{KJ} x_K + \sum_{i \neq J} a_{Ji} x_i$$

$$= \sum_{i=1}^{n} a_{Ji} x_i + \sum_{K=1}^{n} a_{KJ} x_K$$

This is the Jth component of $(A + A^T) x$.

So, the full gradient ∇f is $\left[(A + A^T) x \right]^T$.

$$P = \begin{pmatrix} 3 & 1 \\ 3 & 6 \end{pmatrix}$$

Consider $\begin{pmatrix} x & y \end{pmatrix} P \begin{pmatrix} x \\ y \end{pmatrix}$

$$= X^T P X$$

$$= 3x^2 + 4xy + 6y^2$$

Clearly P is not symmetric.

but, $$X^T P X = X^T \underbrace{\tfrac{1}{2} (P + P^T)}_{= A} X$$

$$= X^T A X$$

A is symmetric. , $A = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}$

$\Rightarrow$ $Q(x) = \frac{1}{2} x^T A x \mp b^T x + c$, $Q: \mathbb{R}^n \to \mathbb{R}$ is strictly convex, and has a unique finite global minimum iff A is +ve definite.

CG: To minimize $Q(x)$ we solve

$$\nabla Q = 0$$

i.e. $x^T A - b^T = 0$

i.e. $\underline{Ax = b}$.

$\Rightarrow$ Two vectors u, v are A-orthogonal or conjugate if

$$u^T A v = 0$$

$$\Rightarrow \langle u, Av \rangle = 0$$

Underlying Mathematics of CG:

Let, $D = \{ d_i, \dots, d_n \}$ is a set of mutually conjugate vectors.

H.W: D is linearly independent.

So, D forms a basis for $\mathbb{R}^n$, and we can write the solution of $Ax = b$ in this basis:

$$x^* = \sum_{i=1}^{n} \alpha_i d_i$$

Pre-multiplying by $d_K^T A$ gives:

$$d_K^T A x^* = \alpha_K d_K^T A d_K \quad [d_K, d_i \text{ are conjugate}]$$

$$\Rightarrow \alpha_K = \frac{d_K^T b}{d_K^T A d_K} = \frac{\langle d_K, b \rangle}{\langle d_K, A d_K \rangle}, \quad K = 1, 2, \dots n$$

Compute the n conjugate directions and α_K's to obtain x^*.

But, if n is large, the above method would take too much time to compute all n conjugate directions.

<u>Remedy</u>: To compute few d_k's to get a good approximation to x^*.

<u>CG Method</u>: (to solve $Ax = b$)

Starting with any $x_0 \in \mathbb{R}^n$,

1. Set $d_0 = r_0 = b - Ax_0 = -\nabla Q(x_0)$ [Taking ∇Q as column vector]

2. for $k = 0, \dots, n-1$ do

   a) $\alpha_k = \dfrac{r_k^T d_k}{d_k^T A d_k} \left(= \dfrac{r_k^T r_k}{d_k^T A d_k} \right)$ ... Step length

   b) $x_{k+1} = x_k + \alpha_k d_k$ ... approx soln.

   c) $r_{k+1} = b - Ax_{k+1} = r_k - \alpha_k A d_k$. ... residual

   d) $\beta_k = -\dfrac{r_{k+1}^T A d_k}{d_k^T A d_k} \left(= \dfrac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \right)$ ... improvement

   e) $d_{k+1} = r_{k+1} + \beta_k d_k$. ... search direction

until convergence.

<u>Note</u>: 1. The residual set is orthogonal, i.e.

$$r_i^T r_j = 0, \quad i \neq j$$

2. d_i is A-orthogonal to d_j, i.e.

$$d_i^T A d_j = 0, \quad i \neq j$$

3. Step (a) & (d) can be further rewritten.

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

$$\Rightarrow \quad A d_k = \frac{1}{\alpha_k}(r_k - r_{k+1})$$

$$\therefore \quad r_{k+1}^T A d_k = -\frac{1}{\alpha_k} r_{k+1}^T r_{k+1}$$

Also, $d_k^T A d_k = (r_k - \beta_{k-1} d_{k-1})^T A d_k$

$$= r_k^T A d_k = \frac{1}{\alpha_k} r_k^T (r_k - r_{k+1})$$

$$= \frac{1}{\alpha_k} r_k^T r_k \quad \ldots (*)$$

So, $\boxed{\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}}$ .. (I)

Also, $(*) \Rightarrow$ $\boxed{\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}}$ .. (II)

Also, $d_k^T A d_k = r_k^T A d_k$ from $(*)$ ..(III)

We have, $r_{k+1} = r_k - \alpha_k A d_k$

So, $r_k^T r_{k+1} = r_k^T r_k - \alpha_k r_k^T A d_k$

$$= r_k^T r_k - \frac{r_k^T r_k}{d_k^T A d_k} r_k^T A d_k$$

$$= 0.$$

$\boxed{\text{So } r_k \perp r_{k+1}}$

<u>Thm</u>:- Let $\{d_i\}_{i=0}^{n-1}$ be a set of non-zero A-orthogonal vectors. For any $x_0 \in \mathbb{R}^n$, the seq^n generated by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \geq 0$$

with $\alpha_k = \frac{r_k^T d_k}{d_k^T A d_k}$ and $r_k = b - A x_k$.

converges to the unique soln x^* of $Ax = b$ after n steps.

Proof:- Since d_i's are L.I, we can write

$$x^* - x_0 = c_0 d_0 + c_1 d_1 + \cdots + c_{n-1} d_{n-1} \quad \cdots (*)$$

$$\therefore \quad c_k = \frac{d_k^T A (x^* - x_0)}{d_k^T A d_k}$$

Now, we get from the iterative process,

$$x_k - x_0 = \alpha_0 d_0 + \alpha_1 d_1 + \cdots + \alpha_{k-1} d_{k-1} \quad \cdots (**)$$

So, $d_k^T A (x_k - x_0) = 0$

$$\therefore \quad c_k = \frac{d_k^T A (x^* - x_k + x_k - x_0)}{d_k^T A d_k}$$

$$= \frac{d_k^T (b - A x_k)}{d_k^T A d_k} = \frac{d_k^T r_k}{d_k^T A d_k}.$$

Now, if $\alpha_k := c_k \ \forall k$, then after n steps (*) & (**) coincides, i.e. $x_k \to x^*$ after n steps.

Thm 2:- α_k minimizes $Q(x_k + \alpha_k d_k)$. (Method of line search)

Proof: To minimize: $Q(x_k + \alpha d_k)$, $\alpha \in \mathbb{R}$.

Then $\left.\frac{\partial}{\partial \alpha} Q \right|_{\alpha = \alpha_k} = 0 \longrightarrow D_{d_k} Q(x_k + \alpha d_k) = 0$

$$\cancel{\Rightarrow} \quad \cancel{\nabla Q \cdot d_k = 0} \Rightarrow (\nabla Q)^T \cdot \frac{\partial}{\partial \alpha}(x_k + \alpha_k d_k) = 0$$

$$\Rightarrow (A x_{k+1} - b)^T \cdot d_k = 0$$

$$\Rightarrow (A x_k + \alpha_k A d_k - b)^T d_k = 0$$

$$\Rightarrow (\alpha_k A d_k - r_k)^T d_k = 0$$

$$\Rightarrow \boxed{\alpha_k = \frac{r_k^T d_k}{d_k^T A d_k}}$$

Thm 1 and Thm 2 produce the same step length α_k.

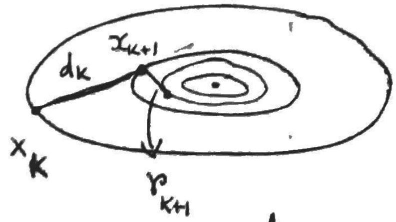
<u>Property :-</u> $r_{k+1} \perp d_k$.

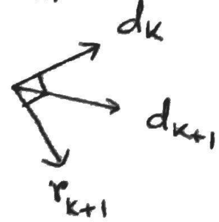
We have
$$r_{k+1} = r_k - \alpha_k A d_k .$$
$$\therefore \quad d_k^T r_{k+1} = d_k^T r_k - \alpha_k d_k^T A d_k$$
$$= d_k^T r_k - r_k^T d_k \qquad \left[\because \alpha_k = \frac{r_k^T d_k}{d_k^T A d_k} \right]$$
$$= 0 .$$
So, $\underline{d_k \perp r_{k+1}}$.

<u>New Search direction, β_k :-</u>

Let, $d_{k+1} = r_{k+1} + \beta d_k$, a linear combination of r_{k+1} & d_k.

$$\therefore \quad d_k^T A d_{k+1} = d_k^T A r_{k+1} + \beta d_k^T A d_k$$
$$\Rightarrow \quad d_k^T A r_{k+1} = - \beta d_k^T A d_k$$
$$\therefore \quad \beta_k := - \frac{d_k^T A r_{k+1}}{d_k^T A d_k}$$





Extension to Non-quadratic problems :-

To minimize an arbitrary function f.

Start with any $x_0 \in \mathbb{R}^n$.

1. Set $d_0 = -\nabla f(x_0)$

2. for $k = 0, 1, \ldots (n-1)$ do

   a) obtain α_k that minimizes
   $$g(\alpha) = f(x_k + \alpha d_k).$$

   b) $x_{k+1} = x_k + \alpha_k d_k$

   c) $$\beta_k = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})^T}{\nabla f(x_k)^T \nabla f(x_k)^T}.$$

   d) $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k.$

until conv.

Exm :- Solve $\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

CG: $x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

1. $d_0 = r_0 = b - Ax_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

2. a) $\alpha_0 = \dfrac{r_0^T r_0}{d_0^T A d_0} = \dfrac{1}{2}$ Iter 1

   b) $x_1 = x_0 + \alpha_0 d_0 = \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}$

   c) $r_1 = b - Ax_1 = \begin{pmatrix} 0 \\ 1/2 \end{pmatrix}$

   d) $\beta_0 = \dfrac{r_1^T r_1}{r_0^T r_0} = \dfrac{1}{4}$

   e) $d_1 = r_1 + \beta_0 d_0 = \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 1/2 \end{pmatrix}$

Iter 2

   a) $\alpha_1 = \dfrac{r_1^T r_1}{d_1^T A d_1} = 2/3$

   b) $x_2 = x_1 + \alpha_1 d_1 = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}$

   c) $r_2 = b - Ax_2 = 0$

So, $x_2 = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}$ is the exact soln.