Gradiente Conjugado

+ Método Iterativo para resolver el sistema lineal $Ax = b$
+ $$\min_{x} \phi(x) = \frac{1}{2} x^T A x - b^T x.$$

1950. ⇒ Eliminación Gaussiana.

$$\nabla \phi(x) = Ax - b = r(x)$$ → residual del sistema.

$$x_k \Rightarrow Ax_k - b = r_k$$

Métodos de direcciones Conjugadas

ϕ Barato $\{p_0, p_1, \dots\}$ son conjugados

con respecto a la matriz A - simétrica pos def.

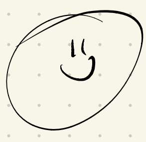
Un par de vectores p_i, p_j son conjugados con respecto a A si

$$p_i^T A p_j = 0 \quad \forall i \neq j.$$

$\Rightarrow \{p_0, \dots, p_{n-1}\}$ es un conjunto Linealmente independiente

Dado un $x_0 \in \mathbb{R}^n$ y un conjunto de direcciones conjugadas

$$\{p_0, \dots, p_{n-1}\}$$



$$x_{k+1} = x_k + \alpha_k p_k$$ donde elegimos α_k

→ minimice $\phi(x)$ sobre la línea $x_k + \alpha p_k$.

$$\hat{\phi}(\alpha) = \phi(x_k + \alpha p_k)$$

$$\boxed{\alpha_k = \frac{-r_k^T p_k}{p_k^T A p_k}}$$

Teorema.

Para cualquier $x_0 \in \mathbb{R}^n$ la secuencia generada por el algoritmo converge a la solución x^* del sistema lineal $Ax=b$ en a lo más n pasos

Demostración.

$\{p_i\}$ es linealmente Independiente. $\Rightarrow \operatorname{Span}(\{p_i\}) = \mathbb{R}^n$

$$\Rightarrow \quad x^* - x_0 = \sigma_0 p_0 + \dots + \sigma_{n-1} p_{n-1}$$

Premultiplicamos por... $p_k^T A$.

$$\sigma_k = \frac{p_k^T A (x^* - x_0)}{p_k^T A p_k}$$

Si $\{x_k\}$ es generado por

tenemos que:

$$x_k = x_0 + \alpha_0 p_0 + \dots + \alpha_{k-1} p_{k-1}$$

$$\Rightarrow p_k^T A (x_k - x_0) = 0 \quad \Rightarrow p_k^T A (x^* - x_0 + x_k - x^*) = 0.$$

$$\Rightarrow p_k^T A (x^* - x_0) = p_k^T A (x^* - x_k)$$

$$= p_k^T (b - A x_k)$$

$$= -p_k^T r_k$$

$$\Rightarrow \sigma_k = \alpha_k$$

$$r_{k+1} = r_k + \alpha_k A p_k.$$

Teorema:

Sea $x_0 \in \mathbb{R}^n$ sea cualquier punto inicial y supongamos que tenemos la secuencia $\{x_k\}$ generada por (¨) Entonces:

$$r_k^T p_i = 0 \quad \text{para } i = 0, \dots, k-1.$$

y x_k es el minimizador de $\phi(x) = \frac{1}{2} x^T A x - b^T x$ sobre el subespacio:

$$\{x \mid x = x_0 + \mathrm{Span}\{p_0, \dots, p_{i-1}\}\} \rightarrow (¨)$$

Demostración:

$\tilde{x}$ minimiza ϕ sobre (¨) $\iff r(\tilde{x})^T p_i = 0$ para cada $i = 0, \dots, k-1$

$$h(\sigma) = \phi(x_0 + \sigma_0 p_0 + \dots + \sigma_{k-1} p_{k-1}), \text{ donde } \sigma = \begin{pmatrix} \sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_{k-1} \end{pmatrix}$$

h es cuadrática convexa. σ^*

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = 0 \qquad i = 0, \dots, k-1.$$

$$\nabla \phi(x_0 + \dots + \sigma_{k-1}^* p_{k-1})^T p_i = 0 \quad \forall i = 0, \dots, k-1.$$

$$\searrow r^T(\tilde{x})$$

$$\tilde{x} = x_0 + \sigma_0^* p_0 + \dots + \sigma_{k-1}^* p_{k-1}$$

$$\Rightarrow r(\tilde{x})^T p_i = 0 \quad \forall i = 0, \dots, k-1$$

P.D. x_k cumple $r_k^T p_k = 0$.

Inducción:

$k=1 \qquad x_1 = x_0 + \underline{\alpha_0 p_0}$ minimiza ϕ sobre la dirección p_0

$$\underline{r_1^T p_0 = 0} \,//$$

Hipótesis de inducción:

$r_{k-1}^T p_i = 0 \quad i=0,1,\dots,k-2$, $r_k = r_{k-1} + \alpha_{k-1} A p_{k-1}$ *

$$p_{k-1}^T r_k = p_{k-1}^T r_{k-1} + \overset{-\frac{r_{k-1}^T p_{k-1}}{p_{k-1}^T A p_{k-1}}}{\alpha_{k-1}} A p_{k-1}$$

$$= 0 = p_{k-1}^T r_{k-1} - p_{k-1}^T r_{k-1}$$

$p_i \; , i=0,\dots,k-2$

$$p_i^T r_k = \underbrace{\cancel{p_i^T r_{k-1}}}_{\text{inducción}} + \alpha_{k-1} \underbrace{\cancel{p_i^T A p_{k-1}}}_{\text{conjugacía}} = 0$$

$$\Rightarrow r_k^T p_i = 0 \; , i=0,\dots,k-1. \quad \blacksquare$$

* el residual es ortogonal a las direcciones anteriores.

* No necesitamos saber todas las direcciones anteriores Con p_{k-1} es suficiente.

p_k será conjugada al resto de las direcciones

$$p_k = -r_k + \beta_k p_{k-1}$$

β_k lo elegimos tal que p_{k-1} y p_k sean conjugados

$$0 = p_{k-1}^T A p_k = -p_{k-1}^T A r_k + \beta_k p_{k-1}^T A p_{k-1}$$

$$\Rightarrow \beta_k = \frac{p_{k-1}^T A r_k}{p_{k-1}^T A p_{k-1}} = \frac{r_k^T A p_{k-1}}{p_{k-1} A p_{k-1}}.$$

Habemous Algoritmo:

Dado x_0:

$$r_0 = Ax_0 - b \quad , \quad P_0 = -r_0 \quad , \quad k \leftarrow 0.$$

mientras $r_k \neq 0$.

$$\alpha_k \leftarrow \frac{-r_k^T P_k}{P_k^T A P_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k P_k$$

$$r_{k+1} \leftarrow Ax_{k+1} - b$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A P_k}{P_k^T A P_k}.$$

$$P_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} P_k$$

$$k \leftarrow k+1$$

fin while.

$r_i^T r_j = 0 \ \forall i \neq j$ $\forall p_k, r_k$ $\begin{matrix} p_k \\ r_k \end{matrix} \in K(r_{0;k}) = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}$

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

Teorema:

Suponga que el kesimo iterando no es x^* (solución del problema)
Entonces se cumple lo siguiente:

a) $r_i^T r_j = 0$ Para $i = 0, 1, \ldots, k-1$

1 $\subset$ 2 $\supset$ b) $\text{Span}\{r_0, r_1, \ldots, r_k\} = \text{Span}\{r_0, Ar_0, \ldots, A^k r_0\}$

3 $=$ c) $\text{Span}\{p_0, p_1, \ldots, p_k\} = \text{Span}\{r_0, Ar_0, \ldots, A^k r_0\}$

4 d) $p_k^T A p_i = 0 \ \forall i = 0, \ldots, k-1$

Por lo tanto $\{x_k\}$ converge a x^* en a lo más n pasos

Demostración.

Se cumple para k, P.d. para $k+1$.

Hip.

$r_k \in \text{Span}\{r_0, Ar_0, \ldots, A^k r_0\}$

$p_k \in \text{Span}\{r_0, Ar_0, \ldots, A^k r_0\}$

1

Sabemos que

$Ap_k \in \text{Span}\{Ar_0, \ldots, A^{k+1} r_0\}$

$r_{k+1} \in \text{Span}\{r_0, \ldots, A^{k+1} r_0\}$.

$\text{Span}\{r_0, \ldots, r_{k+1}\} \subset \text{Span}\{r_0, Ar_0, \ldots, A^{k+1} r_0\}$

$$A^{k+1} r_0 = A(A^k r_0) \in \text{Span}\{Ap_0, \dots, Ap_k\}$$ (2)

$$r_{i+1} = r_i + \alpha_i A p_i \Rightarrow A p_i = \frac{r_{i+1} - r_i}{\alpha_i}$$

$$\Rightarrow A^{k+1} r_0 \in \text{Span}\{r_0, \dots, r_{k+1}\}$$

---

(3)

$$\text{Span}\{p_0, \dots, p_{k+1}\} = \text{Span}\{p_0, \dots, p_k, r_{k+1}\} \quad \text{Algoritmo.}$$

$$= \text{Span}\{r_0, \dots, A^k r_0, r_{k+1}\} \quad \text{Hipótesis de Inducción.}$$

$$= \text{Span}\{r_0, r_1, \dots, r_k, r_{k+1}\} \quad (b)$$

$$= \text{Span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}$$

---

(4) $$p_{k+1}^T A p_i = -r_{k+1}^T A p_i + \beta_k p_k^T A p_i = 0 \quad \text{si } i = k.$$

$$\beta_k = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

$i \leq k-1$ $\quad p_0, \dots, p_k$ son conjugados ✓

Teo clase pasada $\rightarrow r_{k+1}^T p_i = 0$ para $i = 0, \dots, k$.

$$A p_i \in A\,\text{Span}\{r_0, Ar_0, \dots, A^i r_0\} = \text{Span}\{Ar_0, A^2 r_0, \dots, A^{i+1} r_0\}$$
$$\subset \text{Span}\{p_0, \dots, p_{i+1}\}$$

$$r_{k+1} A p_i = 0 \quad \forall i = 0, \dots, k-1.$$

$$p_k^T A p_i = 0, \quad \text{con } i = 0, \dots, k-1.$$

$$\therefore p_{k+1}^T A p_k = 0$$

$r_k^T r_i = 0 \quad \forall i = 0, \dots, k-1.$

Gradiente Conjugado ~~X~~

$r_i, r_j \rightarrow$ ortogonales

$r_0 = Ax_0 - b$

Sabemos $r_k^T p_i = 0 \quad \forall i = 0, \dots, k-1 \quad \forall k = 1, \dots, n-1.$

$$p_i = -r_i + \beta_i p_{i-1}.$$

$$\Rightarrow r_i \in \text{Span}\{p_i, p_{i-1}\} \quad \forall i = 1, \dots, k-1.$$

$$\Rightarrow r_k^T r_i = 0 \quad \forall i = 1, \dots, k-1.$$

$$r_k^T r_0 = -r_k^T p_0 = 0$$

$$\therefore r_k^T r_i = 0 \quad \forall i = 0, \dots, k-1.$$

Usando que $r_k^T p_i = 0 \quad i = 0, \dots, k-1$ y $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$

$p_k = -r_k + \beta_k p_{k-1}$

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}.$$

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

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

$r_k^T p_i = 0 \quad \forall i = 0, \dots, k-1.$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

$$\beta_{k+1} = \frac{r_{k+1}^T \left[\frac{r_{k+1} - r_k}{\alpha_k}\right]}{p_k^T A p_k} = \frac{\left[r_{k+1}^T r_{k+1} + 0\right]}{p_k^T A p_k \, \alpha_k}$$

$$= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

GC.

Dado x_0;

$r_0 \leftarrow A x_0 - b$

$p_0 \leftarrow -r_0$

$k \leftarrow 0$.

Mientras $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

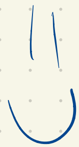
$$r_{k+1} \leftarrow r_k + \alpha_k A p_k$$

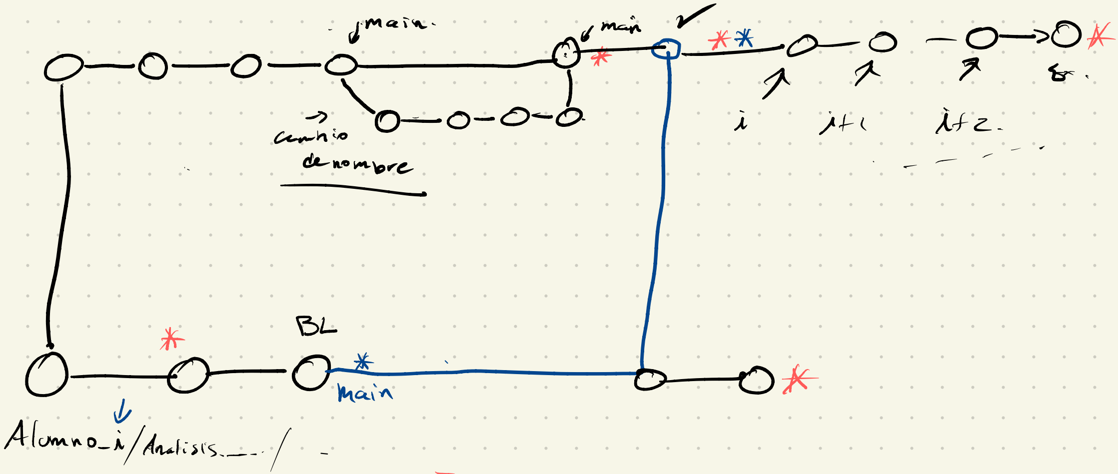
$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$$

$$k \leftarrow k+1$$

Fin.





git pull repo-clase main	Actualizar cambios Miguel
work	work
git add	mete al "loby" los cambios
git commit -m "mensaje"	guarda los cambios
git push origin main	sube los cambios al repositorio Alumno_i

* en github *

↳ Pull Request

Convergencia GC.

Los eigenvalores de la matriz A, determinan el desempeño del GC.

$$x_{k+1} = x_0 + \alpha_0 p_0 + \dots + \alpha_k p_k$$
$$= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \dots + \gamma_k A^k r_0 \quad \text{para constantes } \gamma_i$$

Polinomio $P_k^*(\cdot)$. Polinomio de grado k. $\begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix}$

$$P_k^*(A) = \gamma_0 I + \gamma_1 A + \dots + \gamma_k A^k$$

$$x_{k+1} = x_0 + P_k^*(A) \cdot r_0$$

$$\|z\|_A^2 = z^T A z.$$

$$\phi(x) = \frac{1}{2} x^T A x - b^T x.$$

$$\frac{1}{2}\|x - x^*\|_A^2 = \frac{1}{2}(x-x^*)^T A (x - x^*) = \phi(x) - \phi(x^*)$$

$$\left[\frac{1}{2} x^T A x - b^T x\right] - \left[\frac{1}{2} x^{*T} A x^* - b^T x^*\right]$$

Sabemos que x_{k+1} minimiza ϕ (y $\|x - x^*\|_A^2$) sobre el conjunto $x_0 + \text{Span}\{p_0 + p_1 + \dots + p_k\} = x_0 + \text{Span}\{r_0 + A r_0 + \dots + A^k r_0\}$

P_k^* resuelve.

$$\min_{P_k} \|x_0 + P_k(A) r_0 - x^*\|_A.$$

y como $r_0 = A x_0 - b = A(x_0) - A x^* = A(x_0 - x^*)$.

$$x_{k+1} - x^* = x_0 - P_k^*(A) r_0 - x^*$$
$$= (I + P_k^*(A) A)(x_0 - x^*).$$

Sean $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ eigen valores de A. y $\{v_i\}$ sus eigen vectores ortogonales

$$\Rightarrow \boxed{A = \sum_{i=1}^{n} \lambda_i v_i v_i^T}$$

$\{v_i\}$ genera $\mathbb{R}^n$ $\therefore$

$$x_0 - x^* = \sum_{i=1}^{n} \xi_i v_i \text{ para coeficientes } \xi_i$$

$$P_k(A) v_i = P_k(\lambda_i) v_i \quad , i = 1, \dots, n.$$

$$x_{k+1} - x^* = \sum_{i=1}^{n} [1 + \lambda_i P_k^*(\lambda_i)] \xi_i v_i$$

$$\|x_{k+1} - x^*\|_A^2 = \min_{P_k} \sum_{i=1}^{n} \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2$$

$$\underline{\|x_{k+1} - x^*\|_A^2} \leq \min_{P_k} \underline{\max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2} \left(\sum_{j=1}^{n} \lambda_j \xi_j \right)$$

$$= \min_{P_k} \max_{1 \leq i \leq n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2 \overline{\|x_0 - x^*\|_A^2}$$

$$\|x_0 - x^*\|_A^2 = \sum_{j=1}^{n} \lambda_j \xi_j^2$$

Tendremos una velocidad de convergencia al estimar

$$\min_{P_k} \max_{1 \leq i \leq n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2$$

Teo: Si A tiene r eigenvalores diferentes, entonces GC terminará en a lo más r-iteraciones.

Prueba: $\lambda_1, \dots, \lambda_n$ toman r valores diferentes $z_1 < z_2 < \dots < z_r$. definimos el polinomio

$$Q_r(\lambda) = \frac{(-1)^r}{z_1 \cdots z_r}(\lambda - z_1)(\lambda - z_2)\cdots(\lambda - z_r), \quad Q(\lambda_i) = 0, \; Q(0) = 1$$

$\forall i = 1, \dots, n$

$Q_r(\lambda) - 1$ es un polinomio grado r con raíz en $\lambda = 0$

Polinomio grado $r-1$. $\leftarrow \tilde{P}_{r-1}(\lambda) = \frac{(Q_r(\lambda) - 1)}{\lambda}$

$k = r-1$

$$0 \leq \min_{P_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} [1 + \lambda_i \tilde{P}_{r-1}(\lambda_i)]^2$$

$$= \max_{1 \leq i \leq n} Q_r^2(\lambda_i) = 0$$

$\|x_r - x^*\|_A^2 = 0 \Rightarrow x_r = x^*$ //

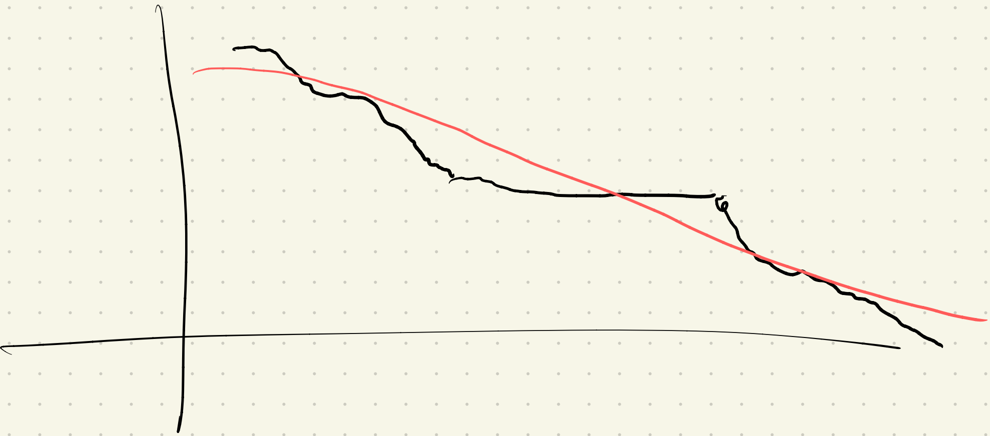
Si A tiene eigenvalores $\lambda_1 \leq \dots \leq \lambda_n$ tenemos que:

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right) \|x_0 - x^*\|_A^2$$

Además si tomamos el número de cond $K(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_n}{\lambda_1}$

se puede mostrar que:

$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1}\right) \|x_0 - x^*\|_A$$



Las distribuciones de los eigenvalores nos dicen cómo se comportará el algoritmo

Precondicionamiento

Podemos acelerar el GC. Ajustando la distribución delos eigenvalores. El precondicionamiento es un cambio de variable vía una matriz no singular C.

$$\hat{x} = Cx.$$

$$\hat{\phi}(x) = \frac{1}{2}\hat{x}^T (C^{-T} A C^{-1})\hat{x} - (C^{-T} b)^T \hat{x}$$

$$\Rightarrow \quad C^{-T} A C^{-1} \hat{x} = C^{-T} b.$$

Ahora nos importan los eigenvalores de $\underline{C^{-T} A C^{-1}}$

$\therefore$ La elección de C es crucial.

Elegir C tal que el numero de condición se achique. o tal que se agrupen eigenvalores.

$$M = \underline{C^T C}.$$

Algoritmo Precondicionado Gradiente conjugado

dado x_0, Precondicionador M, A, b.

$r_0 \leftarrow Ax_0 - b$

Resolver $My_0 = r_0$ para y_0.

$p_0 \leftarrow -y_0$, $k \leftarrow 0$.

mientras $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k$$

Resuelve $My_{k+1} = r_{k+1}$ para y_{k+1}

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k$$

$$k \leftarrow k+1$$

fin.

En lugar que $\cancel{r_i^T r_j = 0}$

tenemos $r_i^T M r_j = 0. \ \forall i \neq j.$

$$My = r.$$