Optimización Numérica sin restricciones

Tema 3: Gradiente Conjugado

Oscar Dalmau

Centro de Investigación en Matemáticas
CIMAT

23 de abril de 2021

Orden del Tema

1. Gradiente Conjugado Lineal
2. Gradiente Conjugado No Lineal
3. Gradiente Conjugado Precondicionado
    - Precondicionamiento
    - Precondicionadores

Gradiente Conjugado

Se quiere resolver el problema de optimización sin restricciones

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \tag{1}$$

donde $\mathbf{Q}$ es una matriz simétrica y definida positiva.

Direcciones Conjugadas

- Sea $\mathbf{Q}$ una matriz simétrica y definida positiva. Dos vectores $\boldsymbol{d}_1, \boldsymbol{d}_2$ se dicen *conjugados con respecto a* $\mathbf{Q}$ o simplemente $\mathbf{Q}$-ortogonales si $\boldsymbol{d}_1^T \mathbf{Q} \boldsymbol{d}_2 = 0$.
- Un conjunto de vectores $\boldsymbol{d}_0, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_k$ son mutuamente $\mathbf{Q}$-ortogonales si $\boldsymbol{d}_i^T \mathbf{Q} \boldsymbol{d}_j = 0$ para $i \neq j$.

Algoritmo Gradiente Conjugado

Algorithm 1 GC-Versión Preliminar

Require: x_0

Ensure: x^*

Hacer $\boldsymbol{g}_0 = \mathbf{Q}\boldsymbol{x}_0 - \boldsymbol{b}$,$\boldsymbol{d}_0 = -\boldsymbol{g}_0$, $k = 0$

while $\|\boldsymbol{g}_k\| \neq 0$ (No conveja) **do**

$\quad \alpha_k = -\dfrac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k}$

$\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$

$\quad \boldsymbol{g}_{k+1} = \mathbf{Q}\boldsymbol{x}_{k+1} - \boldsymbol{b} = \nabla f(\boldsymbol{x}_{k+1})$

$\quad \beta_{k+1} = \dfrac{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{g}_{k+1}}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k}$

$\quad \boldsymbol{d}_{k+1} = -\boldsymbol{g}_{k+1} + \beta_{k+1} \boldsymbol{d}_k$

$\quad k = k + 1$

end while

Forma práctica de GC

Algunas relaciones

- $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
- $\boldsymbol{g}_{k+1} = \mathbf{Q}\boldsymbol{x}_{k+1} - \boldsymbol{b}$
- $\boldsymbol{g}_{k+1} = \boldsymbol{g}_k + \alpha_k \mathbf{Q}\boldsymbol{d}_k$

Forma práctica de GC

Otras relaciones

- $\boldsymbol{g}_{k+1}^T \boldsymbol{d}_k = 0$
- $\alpha_k = -\dfrac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k}$
- $\beta_{k+1} = \dfrac{\boldsymbol{g}_{k+1}^T \mathbf{Q} \boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k}$
- $\boldsymbol{d}_{k+1} = -\boldsymbol{g}_{k+1} + \beta_{k+1} \boldsymbol{d}_k$

Forma práctica de GC

Proposicion

Si $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_k\}$ son $\mathbf{Q}$-ortogonales entonces,

$$\begin{aligned} \boldsymbol{g}_{k+1}^T \boldsymbol{d}_i &= 0 \\ \boldsymbol{g}_{k+1}^T \boldsymbol{g}_i &= 0 \end{aligned}$$

para $i = 0, 1, \cdots, k$

Algoritmo Gradiente Conjugado

$$\alpha_k = \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k} = -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k}$$
$$\beta_{k+1} = \frac{\boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k^T \boldsymbol{g}_k} = \frac{\boldsymbol{g}_{k+1}^T \mathbf{Q} \boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q} \boldsymbol{d}_k}$$

Nos basamos en:

$$\begin{aligned} \boldsymbol{g}_k^T \boldsymbol{g}_k &= -\boldsymbol{g}_k^T \boldsymbol{d}_k \\ \alpha_k \mathbf{Q} \boldsymbol{d}_k &= \boldsymbol{g}_{k+1} - \boldsymbol{g}_k \\ \boldsymbol{g}_{k+1}^T \boldsymbol{g}_k &= 0 \\ \boldsymbol{g}_{k+1}^T \boldsymbol{d}_k &= 0 \end{aligned}$$

Algoritmo Gradiente Conjugado

Algorithm 2 GC (Estandar)

Require: $\boldsymbol{x}_0$

Ensure: x^*

Hacer $\boldsymbol{g}_0 = \mathbf{Q}\boldsymbol{x}_0 - \boldsymbol{b}, \boldsymbol{d}_0 = -\boldsymbol{g}_0$, $k = 0$

while $\|\boldsymbol{g}_k\| \neq 0$ (No conveja) **do**

$\quad \boldsymbol{q}_k = \mathbf{Q}\boldsymbol{d}_k$

$\quad \alpha_k = \dfrac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{d}_k^T \boldsymbol{q}_k} = -\dfrac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q}\boldsymbol{d}_k}$

$\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$

$\quad \boldsymbol{g}_{k+1} = \boldsymbol{g}_k + \alpha_k \boldsymbol{q}_k = \nabla f(\boldsymbol{x}_{k+1})$

$\quad \beta_{k+1} = \dfrac{\boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k^T \boldsymbol{g}_k} = \dfrac{\boldsymbol{g}_{k+1}^T \mathbf{Q}\boldsymbol{d}_k}{\boldsymbol{d}_k^T \mathbf{Q}\boldsymbol{d}_k}$

$\quad \boldsymbol{d}_{k+1} = -\boldsymbol{g}_{k+1} + \beta_{k+1}\boldsymbol{d}_k$

$\quad k = k + 1$

end while

Algoritmo GC no lineal

El Algoritmo GC se usa para resolver el problema de optimización

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \tag{2}$$

donde $\mathbf{Q}$ es una matriz simétrica y definida positiva, o equivalentemente, resolver el sistema de ecuaciones

$$\mathbf{Q}\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

Será posible usarlo cuando $f(\cdot)$ es una función convexa cualquiera?

Algoritmo GC no lineal

El Algoritmo GC se usa para resolver el problema de optimización

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \tag{2}$$

donde $\mathbf{Q}$ es una matriz simétrica y definida positiva, o equivalentemente, resolver el sistema de ecuaciones

$$\mathbf{Q}\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

Será posible usarlo cuando $f(\cdot)$ es una función convexa cualquiera?

Algoritmo GC no lineal Fletcher-Reeves

Algorithm 3 GC Fletcher-Reeves

Require: $\boldsymbol{x}_0$
Ensure: x^*

$\boldsymbol{d}_0 = -\nabla f(\boldsymbol{x}_0)$, $k = 0$
while $\|\nabla f(\boldsymbol{x}_k)\| \neq 0$ (No conveja) **do**
    Calcular α_k usando un método de búsqueda en línea
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
    Calcular $\nabla f(\boldsymbol{x}_{k+1})$
    $\beta_{k+1}^{FR} = \frac{\nabla^T f(\boldsymbol{x}_{k+1}) \nabla f(\boldsymbol{x}_{k+1})}{\nabla^T f(\boldsymbol{x}_k) \nabla f(\boldsymbol{x}_k)}$
    $\boldsymbol{d}_{k+1} = -\nabla f(\boldsymbol{x}_{k+1}) + \beta_{k+1}^{FR} \boldsymbol{d}_k$
    $k = k + 1$
end while

Sugerencia: scipy.optimize.line_search

Algoritmo GC no lineal

Otras variantes

El método de Fletcher-Reeves (FR) converge si el punto inicial está suficientemente cerca del óptimo. Existen variantes del método FR. La diferencia fundamental es la forma de calcular el parámetro β_k

- Polak-Ribiere

$$\beta_{k+1}^{PR} = \frac{\nabla^T f(\boldsymbol{x}_{k+1})(\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k))}{\nabla^T f(\boldsymbol{x}_k) \nabla f(\boldsymbol{x}_k)}$$

- En algunos casos (raros) el algoritmo de PR se puede ciclar infinitamente. Sin embargo, se puede garantizar la convergencia tomando $\beta_{k+1}^{+} = \max(0, \beta_{k+1}^{PR})$.

Algoritmo GC no lineal

Otras variantes

El método de Fletcher-Reeves (FR) converge si el punto inicial está suficientemente cerca del óptimo. Existen variantes del método FR. La diferencia fundamental es la forma de calcular el parámetro β_k

- Polak-Ribiere

$$\beta_{k+1}^{PR} = \frac{\nabla^T f(\boldsymbol{x}_{k+1})(\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k))}{\nabla^T f(\boldsymbol{x}_k)\nabla f(\boldsymbol{x}_k)}$$

- En algunos casos (raros) el algoritmo de PR se puede ciclar infinitamente. Sin embargo, se puede garantizar la convergencia tomando $\beta_{k+1}^{+} = \max(0, \beta_{k+1}^{PR})$.

Algoritmo GC no lineal

Otras variantes

El método de Fletcher-Reeves (FR) converge si el punto inicial está suficientemente cerca del óptimo. Existen variantes del método FR. La diferencia fundamental es la forma de calcular el parámetro β_k

- Hestenes-Stiefel

$$\beta_{k+1}^{HS} = \frac{\nabla^T f(\boldsymbol{x}_{k+1})(\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k))}{(\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k))^T \boldsymbol{d}_k}$$

Algoritmo GC no lineal

Fletcher-Reeves Polak-Ribiere

- Es posible garantizar la convergencia si $|\beta_k| \leq \beta_k^{FR}$.
- Lor lo anterior, una variante adecuada sería: FR-PR para $k \geq 2$

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{si } \beta_k^{PR} < -\beta_k^{FR} \\ \beta_k^{PR} & \text{si } |\beta_k^{PR}| \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{si } \beta_k^{PR} > \beta_k^{FR} \end{cases}$$

Nota: La idea anterior se basa en que para cualquier $|\beta_k| \leq \beta_k^{FR}$ se puede demostrar convergencia global para $k \geq 2$.

Descenso de Fletcher-Reeves

$$\boldsymbol{d}_{k+1} \quad = \quad -\boldsymbol{g}_{k+1} + \beta_{k+1}^{FR}\boldsymbol{d}_k$$

- Si α_k es un tamaño de paso exacto entonces $\boldsymbol{g}_{k+1}\boldsymbol{d}_k = 0$, luego

$$\boldsymbol{g}_{k+1}^T\boldsymbol{d}_{k+1} \quad = \quad -\|\boldsymbol{g}_{k+1}\|^2 + \beta_{k+1}^{FR}\boldsymbol{g}_{k+1}^T\boldsymbol{d}_k = -\|\boldsymbol{g}_{k+1}\|^2 < 0$$

  es decir, $\boldsymbol{d}_{k+1}$ es de descenso.
- Si α_k no es un tamaño de paso exacto, el termino $\beta_{k+1}^{FR}\boldsymbol{g}_{k+1}^T\boldsymbol{d}_k$ podria dominar al primero, y no se garantiza descenso.

Descenso de Fletcher-Reeves

Para el caso de tamaño de paso no exacto, se puede garantizar descenso si se usan las condiciones fuertes de Wolfe

$$\begin{aligned} f_{k+1} &\leq f_k + c_1 \alpha_k \boldsymbol{g}_k^T \boldsymbol{d}_k && (4) \\ |\boldsymbol{g}_{k+1}^T \boldsymbol{d}_k| &\leq -c_2 \boldsymbol{g}_k^T \boldsymbol{d}_k && (5) \end{aligned}$$

con $0 < c_2 < c_1 < 1$. La condiciones anteriores garantiza la convergencia, seleccionando $0 < c_2 < \frac{1}{2}$

Descenso de Fletcher-Reeves

Proposicion

Si el algoritmo de GC no lineal usa el tamaño de paso inexacto satisfaciendo las condiciones fuertes de Wolfe con $0 < c_2 < \frac{1}{2}$. Entonces el algoritmo produce una direccion de descenso $\boldsymbol{d}_k$, que satisface

$$-\frac{1}{1-c_2} \leq \frac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\|\boldsymbol{g}_k\|^2} \leq \frac{2c_2 - 1}{1 - c_2}$$

para k = 0,1,..

Ver la demostracion en el Nocedal (por induccion)

El problema de optimizacion

$$\min_{\boldsymbol{x}} \qquad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$$

Puede ser resuelto en a lo mas ‘n’ iteraciones usando Gradiente Conjugado.

Sin embargo, la convergencia de GC depende del numero de condicion

$$\kappa(\mathbf{Q}) \quad = \quad \|\mathbf{Q}\|_2 \|\mathbf{Q}^{-1}\|_2 = \frac{\lambda_n}{\lambda_1}$$

de la matriz $\mathbf{Q}$, es decir,

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\mathbf{Q}} \quad \leq \quad 2 \left(\frac{\sqrt{\kappa(\mathbf{Q})} - 1}{\sqrt{\kappa(\mathbf{Q})} + 1} \right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_{\mathbf{Q}}$$

aunque la cota anterior esta sobreestimada, nos da una idea de como acelerar el GC. Se puede ver que si $\kappa(\mathbf{Q}) \approx 1$ entonces la convergencia podría ser mas rapida, en particular, si $\kappa(\mathbf{Q}) = 1$, converge en una iteracion.

Orden del Tema

1. Gradiente Conjugado Lineal
2. Gradiente Conjugado No Lineal
3. Gradiente Conjugado Precondicionado
   - Precondicionamiento
   - Precondicionadores

La idea general del precondicionamiento consiste en transformar el sistema

$$\mathbf{Q}\boldsymbol{x} \;=\; \boldsymbol{b}$$

de modo que la matriz del sistema $\mathbf{Q}$ tenga un mejor numero de condicion. Para ello podriamos premultiplicar el sistema por una matriz $\mathbf{M}^{-1}$ de tal forma que $\mathbf{M}^{-1} \approx \mathbf{Q}^{-1}$.

Luego, obtendriamos el sistema equivalente

$$\mathbf{M}^{-1}\mathbf{Q}\boldsymbol{x} \;=\; \mathbf{M}^{-1}\boldsymbol{b}$$

donde suponemos que $\mathbf{M}$ es positiva definida, por ejemplo podriamos seleccionar una matriz diagonal. Si $\mathbf{M}^{-1} \approx \mathbf{Q}^{-1}$ entonces

$$\mathbf{M}^{-1}\mathbf{Q} \;\approx\; \mathbf{I}$$

por lo que $\kappa(\mathbf{M}^{-1}\mathbf{Q}) \approx 1$.

Definamos

$$\mathbf{M} \stackrel{def}{=} \mathbf{C}^T\mathbf{C}$$

El sistema anterior, puede ser reescrito de la siguiente forma

$$\mathbf{M}^{-1}\mathbf{Q}\boldsymbol{x} = \mathbf{M}^{-1}\boldsymbol{b} \tag{6}$$

$$\mathbf{C}^{-1}\mathbf{C}^{-T}\mathbf{Q}\mathbf{C}^{-1}\hat{\boldsymbol{x}} = \mathbf{C}^{-1}\mathbf{C}^{-T}\boldsymbol{b} \tag{7}$$

$$\mathbf{C}^{-T}\mathbf{Q}\mathbf{C}^{-1}\hat{\boldsymbol{x}} = \mathbf{C}^{-T}\boldsymbol{b} \tag{8}$$

con

$$\mathbf{C}^{-1}\hat{\boldsymbol{x}} = \boldsymbol{x} \tag{9}$$

$$\hat{\boldsymbol{x}} = \mathbf{C}\boldsymbol{x} \tag{10}$$

- El nuevo sistema de ecuaciones (8), es equivalente a resolver un problema de optimization cuadratico.
- El siguiente problema se hubiera obtenido tambien sustituyendo (9), $\boldsymbol{x} = \mathbf{C}^{-1}\hat{\boldsymbol{x}}$, en funcion $f(\cdot)$, es decir $g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x}) = f(\mathbf{C}^{-1}\hat{\boldsymbol{x}})$

$$
\begin{aligned}
\min_{\hat{\boldsymbol{x}}} \; g(\hat{\boldsymbol{x}}) &= \frac{1}{2}\hat{\boldsymbol{x}}^T \mathbf{C}^{-T}\mathbf{Q}\mathbf{C}^{-1}\hat{\boldsymbol{x}} - (\mathbf{C}^{-T}\boldsymbol{b})^T\hat{\boldsymbol{x}} && (11)\\
&= \frac{1}{2}\hat{\boldsymbol{x}}^T \hat{\mathbf{Q}}\hat{\boldsymbol{x}} - \hat{\boldsymbol{b}}^T\hat{\boldsymbol{x}} && (12)
\end{aligned}
$$

Donde

$$
\begin{aligned}
\hat{\mathbf{Q}} &= \mathbf{C}^{-T}\mathbf{Q}\mathbf{C}^{-1} && (13)\\
\hat{\boldsymbol{b}} &= \mathbf{C}^{-T}\boldsymbol{b} && (14)\\
\hat{\boldsymbol{x}} &= \mathbf{C}\boldsymbol{x} && (15)\\
\mathbf{C}^{-1}\hat{\boldsymbol{x}} &= \boldsymbol{x} && (16)
\end{aligned}
$$

- La idea ahora es aplicar al algoritmo GC al problema de minimizacion de $g(\hat{\boldsymbol{x}})$, Ecuacion (12).
- Luego, hacer sustitución de variables para llevar las ecuaciones a los terminos equivalentes que corresponden a la minimizacion de $f(\boldsymbol{x})$.
- Es decir, convertir de la notacion nueva a la original, i.e., $\hat{\mathbf{Q}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{x}} \rightarrow \mathbf{Q}, \boldsymbol{b}, \boldsymbol{x}$ mediante las ecuaciones (13)-(15).

Require: $\hat{\boldsymbol{x}}_0$
Ensure: $\hat{\boldsymbol{x}}^*$

Hacer $\hat{\boldsymbol{g}}_0 = \hat{\mathbf{Q}}\hat{\boldsymbol{x}}_0 - \hat{\boldsymbol{b}}$, $\hat{\boldsymbol{d}}_0 = -\hat{\boldsymbol{g}}_0$, $k = 0$
while $\|\hat{\boldsymbol{g}}_k\| \neq 0$ (No conveja) **do**
    $\hat{\alpha}_k = \frac{\hat{\boldsymbol{g}}_k^T \hat{\boldsymbol{g}}_k}{\hat{\boldsymbol{d}}_k^T \hat{\mathbf{Q}} \hat{\boldsymbol{d}}_k}$
    $\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \hat{\alpha}_k \hat{\boldsymbol{d}}_k$
    $\hat{\boldsymbol{g}}_{k+1} = \hat{\boldsymbol{g}}_k + \hat{\alpha}_k \hat{\mathbf{Q}} \hat{\boldsymbol{d}}_k$
    $\hat{\beta}_{k+1} = \frac{\hat{\boldsymbol{g}}_{k+1} \hat{\boldsymbol{g}}_{k+1}}{\hat{\boldsymbol{g}}_k^T \hat{\boldsymbol{g}}_k}$
    $\hat{\boldsymbol{d}}_{k+1} = -\hat{\boldsymbol{g}}_{k+1} + \hat{\beta}_{k+1} \hat{\boldsymbol{d}}_k$
    $k = k + 1$
end while

Por ejemplo para $\hat{\alpha}_k$ tenemos

$$\hat{\alpha}_k = \frac{\hat{\boldsymbol{g}}_k^T \hat{\boldsymbol{g}}_k}{\hat{\boldsymbol{d}}_k^T \hat{\mathbf{Q}} \hat{\boldsymbol{d}}_k} = \frac{\boldsymbol{g}_k^T \mathbf{M}^{-1} \boldsymbol{g}_k}{\boldsymbol{d}_k \mathbf{Q} \boldsymbol{d}_k}$$

PAra $\hat{\beta}_{k+1}$ se obtiene

$$\hat{\beta}_{k+1} = \frac{\hat{\boldsymbol{g}}_{k+1}^T \hat{\boldsymbol{g}}_{k+1}}{\hat{\boldsymbol{g}}_k^T \hat{\boldsymbol{g}}_k} = \frac{\boldsymbol{g}_{k+1}^T \mathbf{M}^{-1} \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k^T \mathbf{M}^{-1} \boldsymbol{g}_k}$$

Defininiendo $\boldsymbol{z}_k$ como la solucion del siguiente sistema de ecuaciones,

$$\mathbf{M} \boldsymbol{z}_k \stackrel{def}{=} \boldsymbol{g}_k$$

entonces

$$\boldsymbol{z}_k = \mathbf{M}^{-1} \boldsymbol{g}_k$$

Hacer $\boldsymbol{g}_0 = \mathbf{Q}\boldsymbol{x}_0 - \boldsymbol{b}$,
Resolver $\mathbf{M}\boldsymbol{z}_0 = \boldsymbol{g}_0$ para $\boldsymbol{z}_0$
$\boldsymbol{d}_0 = -\boldsymbol{z}_0$, $k = 0$
while $\|\boldsymbol{g}_k\| \neq 0$ (No conveja) **do**
    $\alpha_k = \frac{\boldsymbol{g}_k^T \boldsymbol{z}_k}{\boldsymbol{d}_k \mathbf{Q} \boldsymbol{d}_k}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
    $\boldsymbol{g}_{k+1} = \boldsymbol{g}_k + \alpha_k \mathbf{Q} \boldsymbol{d}_k$.
    Resolver el sistema $\mathbf{M}\boldsymbol{z}_{k+1} = \boldsymbol{g}_{k+1}$ para $\boldsymbol{z}_{k+1}$,
    $\beta_{k+1} = \frac{\boldsymbol{g}_{k+1}^T \boldsymbol{z}_{k+1}}{\boldsymbol{g}_k^T \boldsymbol{z}_k}$
    $\boldsymbol{d}_{k+1} = -\boldsymbol{z}_{k+1} + \beta_{k+1} \boldsymbol{d}_k$.
    $k = k + 1$
end while

Notar que si en el algoritmo *Gradiente Conjugado Precondicionado* se hace $\mathbf{M} = \mathbf{I}$ entonces se obtiene el *Algoritmo de Gradiente Conjugado* estandar, pues en este caso se tiene $\boldsymbol{g}_k = \boldsymbol{z}_k$.

Orden del Tema

1. Gradiente Conjugado Lineal
2. Gradiente Conjugado No Lineal
3. Gradiente Conjugado Precondicionado
   - Precondicionamiento
   - Precondicionadores

1. $\mathbf{M}$ debe ser simetrica y positiva definida .
2. $\mathbf{M}$ debe ser tal que el sistema $\mathbf{M}\boldsymbol{z}_k = \boldsymbol{g}_k$ se pueda resolver eficientemente.
3. $\mathbf{M}^{-1}$ debe aproximar $\mathbf{Q}^{-1}$ en el sentido que $\|\mathbf{I} - \mathbf{M}^{-1}\mathbf{Q}\| < 1$

Usando la descomposicion $\mathbf{Q} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$ de la matriz positiva definida $\mathbf{Q}$, donde $\mathbf{L}$ es la matriz triangular inferior y $\mathbf{D}$ es una matriz diagonal, entonces podemos tener opciones para $\mathbf{M}$ dadas

1. $\mathbf{M} = \mathbf{D}$: Precondicionamiento Jacobi ,
2. $\mathbf{M} = \mathbf{L} + \mathbf{D}$: Precondicionamiento Gauss-Seidel,
3. $\mathbf{M} = \mathbf{L} + \frac{1}{\omega}\mathbf{D}$: Precondicionamiento SOR, con $\omega \in (0, 2)$

Tarea

Implementar los Algoritmos de GC lineal y no lineal. Probar los algoritmos anteriores con las siguientes funciones: $f(x_1,x_2,x_3) = x_1^2 + x_2^2 + x_3^2 + x_1x_2$, $f(x,y) = xe^{-x^2-y^2}$ y $f(x_1,x_2) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$ (La Función de Rosembrock)

1. Ejecute los algoritmos, adecuadamente, para cada una de las funciones desde 2 puntos iniciales.
2. Reporte la siguiente información para algunas de las iteraciones (incluya algunas de las iteraciones finales): k (Iteración), $|\nabla f(\boldsymbol{x}_k)|$ (Norma del Gradiente), , $|\nabla \boldsymbol{d}_k|$ (Norma de la dirección de descenso)$f(\boldsymbol{x}_k)$ (Valor de la función), $\boldsymbol{x}_k$ (Punto encontrado).
3. Grafique $(\boldsymbol{x}_k, f(\boldsymbol{x}_k))$ y $(\boldsymbol{x}_k, |\nabla f(\boldsymbol{x}_k)|)$