

Centro de investigación en Matemáticas



**A family of hybrid conjugate gradient
method with restart procedure for
unconstrained optimizations and image
restorations**

Ezau Faridh Torres Torres

Maestría en Ciencias con Orientación en:
Matemáticas Aplicadas

Curso :
Optimización

Fecha de entrega: **30 de Mayo del 2024**

Abstract

Se hará uso del artículo: [2]. El método de gradiente conjugado es uno de los métodos más efectivos para resolver problemas de optimización a gran escala. Basado en el **parámetro conjugado CD** y en un **parámetro conjugado PRP mejorado**, un parámetro conjugado híbrido con un solo parámetro se diseña mediante el uso de dos técnicas híbridas, y luego se establece un procedimiento de reinicio en su dirección de búsqueda para mejorar su propiedad de descenso y eficiencia computacional. En consecuencia, se establece **una familia de método de gradiente conjugado híbrido con procedimiento de reinicio**, que es descenso suficiente en cada iteración sin depender de ninguna selección de criterios de búsqueda en línea. Bajo las suposiciones habituales y utilizando el criterio débil de búsqueda en línea de Wolfe para generar los tamaños de paso, se demuestra la convergencia global de la familia propuesta.

Finalmente, al elegir un algoritmo específico de esta familia para resolver problemas de optimización sin restricciones a gran escala y restauraciones de imágenes, todos los resultados numéricos muestran que el algoritmo propuesto es efectivo.

1. Introducción

Se considera el problema de optimización sin restricciones:

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1)$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es una función suave. Sabemos que los métodos comunes para resolver este problema incluyen al Método de Newton, el Método de Cuasi-Newton, al Método de Gradiente Conjugado (CGM), etc. Sin embargo, para problemas de gran escala, el CGM sigue siendo el método más popular debido a su simplicidad y menores requisitos de almacenamiento. El CGM clásico siempre se itera en la siguiente forma

$$x_{x+1} = x_k + \alpha_k d_k \quad (2)$$

donde α_k es el **tamaño de paso** y generalmente se calcula mediante una búsqueda en línea exacta adecuada, por ejemplo, la búsqueda en línea con las condiciones fuertes o débiles de Wolfe. d_k es la **dirección de búsqueda** que se genera por

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2. \end{cases} \quad (3)$$

Y aquí, $g_k = \nabla f(x_k)$ representa el gradiente de la función objetivo en el punto x_k y β_k es el **parámetro conjugado**, el cual determina la dirección de búsqueda. Existen diferentes tipos de

parámetros conjugados:

- **Hestenes-Stiefel (HS)** : $\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}$.
- **Fletcher-Reeves (FR)** : $\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$.
- **Polak-Ribière-Polak (PRP)** : $\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}$.
- **Fletcher (CD)** : $\beta_k^{CD} = \frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}}$.
- **Liu-Storey (LS)** : $\beta_k^{LS} = \frac{g_k^T y_{k-1}}{-g_{k-1}^T d_{k-1}}$.
- **Dai-Yuan (DY)** : $\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}$.

donde $\|\cdot\|$ denota la norma Euclideana y $y_{k-1} = g_k - g_{k-1}$. Por lo general, diferentes parámetros conjugados corresponden a diferentes CGMs, y estos conducen a diferentes propiedades teóricas y numéricas. Por ejemplo, entre los seis CGM mencionados anteriormente, los métodos HS, PRP y LS poseen un rendimiento numérico excelente, pero su convergencia global es difícil de lograr bajo la búsqueda en línea inexacta. Los métodos FR, CD y DY tienen fuertes propiedades de convergencia; sin embargo, su rendimiento práctico es modesto.

Para diseñar un CGM con excelentes resultados numéricos y buenas propiedades de convergencia, los enfoques factibles habituales incluyen mejorar las propiedades de convergencia de los métodos HS, PRP y LS o mejorar la eficiencia computacional de los métodos FR, CD y DY, o hibridar el método con un rendimiento numérico fino y con una buena propiedad de convergencia o extendiendo la forma de las direcciones de búsqueda clásicas.

En este trabajo, se estudia la mejora de un MCG con doble parámetro mediante el uso de la forma de hibridación y el procedimiento de reinicio. El objetivo es establecer una familia eficiente de CGM híbridos para resolver problemas de optimización sin restricciones y restauración de imágenes. Las principales contribuciones de este trabajo son:

- Basado en un parámetro conjugado con parámetro doble, se proporciona un nuevo parámetro conjugado híbrido con parámetro único (θ_k). Aquí se proporcionan varias opciones para θ_k y luego se propone una familia de MCGs híbridos con procedimientos de reinicio. La condición de reinicio está diseñada por el parámetro conjugado presentado, y su dirección de reinicio está diseñada para ser un descenso suficiente sin depender de ninguna búsqueda en línea.
- La dirección de búsqueda generada por la familia propuesta satisface la condición de descenso suficiente sin depender de las elecciones del parámetro θ_k , los vectores p_{k-1} y q_k , y las búsquedas en línea. **Bajo los supuestos usuales y la búsqueda en línea con condiciones débiles de Wolfe, se demuestra la convergencia global de la familia propuesta.**
- Se aplica un algoritmo específico producido por la familia propuesta para resolver problemas de optimización sin restricciones y restauración de imágenes a gran escala, sus resultados

numéricos muestran que el nuevo método es muy prometedor para resolver estos dos tipos de problemas.

2. Motivación y propiedad de descenso suficiente del algoritmo IJYHL.

Se establece una familia de algoritmos mediante el diseño de un nuevo parámetro conjugado híbrido y el uso del procedimiento de reinicio para acelerar el descenso de su dirección de búsqueda.

Diversos autores se han basado en los métodos FR, PRP, HS y DY para proponer familias de métodos de gradiente conjugado con doble parámetro con diferentes parámetros conjugados β_k con diferentes variantes y subparámetros según sus finalidades. Tales β_k absorben la información de cada uno de los cuatro métodos en cada iteración, sin embargo, algunos no son fáciles de escoger y se proponen opciones para lograr la convergencia global.

Motivados por la idea de construir el parámetro conjugado híbrido [1] y por las formas de mejorarlo [3], [4], para que tal parámetro diseñado tenga mejores propiedades teóricas y numéricas, se propone una nueva fórmula de parámetro conjugado con un solo parámetro de la siguiente manera:

$$\beta_k^{\text{JYHL}} = \frac{2 \left(\|g_k\|^2 - \theta_k \frac{\|g_k\|}{\|p_{k-1}\|} |g_k^T p_{k-1}| \right)}{g_{k-1}^T (g_{k-1} - d_{k-1})} \quad (4)$$

donde p_{k-1} es un vector arbitrario no nulo y θ_k es el **parámetro de peso combinado** que afecta a las propiedades teóricas y numéricas de un algoritmo, y generalmente debe satisfacer $0 \leq \theta_k \leq 1$. Además, tal selección no solo es importante, sino también difícil, se considera la selección de parámetros de peso combinado como uno de los problemas abiertos de las CGMs no lineales. Por lo tanto, para evitar los problemas causados por la selección de estos parámetros, se ofrecen varias opciones de la siguiente manera:

$$\begin{aligned} & \frac{|g_k^T g_{k-1}|}{\|g_k\| \|g_{k-1}\|}, & \frac{|g_k^T d_{k-1}|}{\|g_k\| \|d_{k-1}\|}, & \frac{|g_k^T y_{k-1}|}{\|g_k\| \|y_{k-1}\|}, \\ & \frac{|g_{k-1}^T y_{k-1}|}{\|g_{k-1}\| \|y_{k-1}\|}, & \frac{|d_{k-1}^T y_{k-1}|}{\|d_{k-1}\| \|y_{k-1}\|}, & \frac{|g_{k-1}^T d_{k-1}|}{\|g_{k-1}\| \|d_{k-1}\|}. \end{aligned} \quad (5)$$

Sin embargo, en el algoritmo generado por la JYHL, es difícil de obtener la convergencia global bajo las suposiciones habituales y los criterios de búsqueda en línea inexactos. Por lo tanto, se mejora (4) aún más, adoptando la forma de hibridación de la siguiente manera:

$$\beta_k^{\text{IJYHL}} = \frac{2 \left(\|g_k\|^2 - \theta_k \frac{\|g_k\|}{\|p_{k-1}\|} |g_k^T p_{k-1}| \right)}{\max\{g_{k-1}^T (g_{k-1} - d_{k-1}), 2\|g_{k-1}\|^2\}} \quad (6)$$

con esto, β_k^{IYHL} satisface:

$$0 \leq \beta_k^{\text{IYHL}} \leq \frac{2 \left(\|g_k\|^2 - \theta_k \frac{\|g_k\|}{\|p_{k-1}\|} |g_k^T p_{k-1}| \right)}{2\|g_{k-1}\|^2} \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2}. \quad (7)$$

Por otro lado, el procedimiento de reinicio ayuda a mejorar la eficiencia computacional del CGM. Diversos autores propusieron diferentes variantes de CGM con procedimiento de reinicio que tuvieron buenas propiedades teóricas y desempeño numérico [4], [5]. Inspirado por las ideas de elegir el procedimiento de reinicio para hacer que la dirección diseñada cumpla con la condición de descenso suficiente independientemente de cualquier búsqueda en línea y para mejorar su eficiencia computacional, se introduce un procedimiento de reinicio en el próximo esquema, y se propone la dirección de búsqueda de la IYHL de la siguiente manera:

$$d_k = \begin{cases} -g_k, & (k = 1), \\ -g_k + \beta_k^{\text{IYHL}} d_{k-1}, & (k \geq 2) \text{ si } -\mu \|g_{k-1}\|^2 \leq g_k^T d_{k-1} \leq \nu \|g_{k-1}\|^2, \\ -g_k + \zeta \frac{g_k^T q_k}{\|q_k\|^2} q_k, & (k \geq 2) \text{ otro caso.} \end{cases} \quad (8)$$

donde $|\zeta| < 1$, $\mu > 0$ y $0 \leq \nu < 1$ son constantes, q_k es un vector arbitrario no nulo, el cual usualmente se escoge como $g_k, g_{k-1}, d_{k-1}, y_{k-1}$, etc., $-\mu \|g_{k-1}\|^2 \leq g_k^T d_{k-1} \leq \nu \|g_{k-1}\|^2$ es la **condición de reinicio** y $d_k = -g_k + \zeta \frac{g_k^T q_k}{\|q_k\|^2} q_k$ es la **dirección de reinicio**. Específicamente, si $\zeta = 0$, entonces la iteración se reiniciará a lo largo de la dirección de descenso más inclinada $d_k = -g_k$. De hecho, además de la condición de reinicio utilizada en (8), si los parámetros ν, μ se seleccionan apropiadamente, entonces se pueden elegir otras condiciones de reinicio.

Basado en las direcciones de búsqueda (8) y en la condición débil de Wolf, se genera el algoritmo (1) y se enuncia el siguiente lema en donde se prueba que la dirección de búsqueda generada por el algoritmo IYHL satisface la condición de descenso suficiente sin depender de ningún criterio de búsqueda de línea (la prueba se encuentra en el Lema 1 en [2]).

Algoritmo 1 IYHL.

- 1: Escoger punto inicial $x_1 \in \mathbb{R}^n$, parámetros apropiados: $0 < \delta < \sigma < 1, |\zeta| < 1, 0 < \mu, 0 \leq \nu < 1$ y tolerancia $\epsilon > 0$ y hacer $d_1 := -g_1$ y $k := 1$.
- 2: **Si** $\|g_k\| < \epsilon$ **Entonces**
- 3: **Termina.**
- 4: Calcular el tamaño de paso α_k satisfaciendo

$$\begin{cases} f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \end{cases} \quad (9)$$

- 5: Generar la siguiente iteración con $x_{k+1} := x_k + \alpha_k d_k$.
 - 6: Hacer $g_{k+1} := g(x_{k+1})$.
 - 7: Calcular β_k^{IYHL} usando (6).
-

-
- 8: **Si** $-\mu\|g_{k-1}\|^2 \leq g_k^T d_{k-1} \leq \nu\|g_{k-1}\|^2$ **Entonces**
 9: $d_{k+1} := -g_k + \beta_k^{\text{IYHL}} d_{k-1}$.
 10: **Sino**
 11: $d_{k+1} := -g_k + \zeta \frac{g_k^T q_k}{\|q_k\|^2} q_k$.
 12: Hacer $k := k + 1$.
-

Lema 1

Suponiendo que la dirección de búsqueda d_k es generada por el algoritmo IYHL, entonces se cumple la relación $g_k^T d_k \leq -c\|g_k\|^2$. Es decir, la dirección de búsqueda satisface la condición de descenso suficiente.

3. Convergencia global del algoritmo IYHL.

Para demostrar las propiedades de convergencia del algoritmo IYHL, son necesarias las siguientes suposiciones para las funciones objetivo:

- **(H1):** La función objetivo $f(x)$ está acotada por debajo en el conjunto de nivel $\Lambda = \{x \in \mathbb{R}^n | f(x) \leq f(x_1)\}$.
- **(H2):** En un vecindario U de Λ , $f(x)$ es diferenciable y su gradiente $g(x)$ es Lipschitz continuo, es decir, existe una constante $L > 0$ tal que $\|g(x) - g(y)\| \leq L\|x - y\| \forall x, y \in U$.

A continuación se tiene el Lema 2, en el cual se dan **las condiciones de Zoutendijk** [6], las cuales desempeñan un papel importante en el análisis de la convergencia global para el CGM.

Lema 2

Asumiendo válidas las hipótesis anteriores, considérese la iteración $x_{k+1} = x_k + \alpha_k d_k$ si d_k satisface $g_k^T d_k < 0$ y el tamaño de paso α_k satisface la condición de Wolf (9), entonces,

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (10)$$

Además, si la condición de descenso suficiente $g_k^T d_k \leq -c\|g_k\|^2$ (para $c > 0$) se satisface, entonces

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty \quad (11)$$

Con ayuda de los lemas 1 y 2, se demuestra la convergencia global del algoritmo IYHL:

Teorema 1

Suponga válidas **(H1)** y **(H1)**. y sea $\{x_k\}$ la secuencia iterativa generada por el algoritmo IJYHL, entonces $\lim_{k \rightarrow \infty} \|g_k\| = 0$.

4. Experimentos Numéricos y resultados.

Para probar el efecto del experimento numérico del algoritmo IJYHL, los parámetros se seleccionan de la siguiente manera:

$$\theta_k = \frac{|g_k^T d_{k-1}|}{\|g_k\| \|d_{k-1}\|}, \quad p_{k-1} = g_{k-1}, \quad q_k = d_k - 1. \quad (12)$$

Existe en código público en [7] y en la página https://github.com/jhyin-optim/FHTTCGMs_with_applications, descrito en Matlab.

4.1. Problema de optimización sin restricciones

Aquí se prueba la efectividad numérica del algoritmo IJYHL para resolver problemas de optimización sin restricciones. Se utiliza la búsqueda en líneas con la condición débil de Wolfe (9) para calcular los tamaños de paso. Se toman los parámetros como los recomiendan los autores:

- Número máximo de iteraciones: $N = 2000$.
- Tolerancia: $tol = 10^{-6}$.
- Factor de reducción para algoritmo de Backtracking $\rho = 0.5$.
- Parámetro para la condición de descenso suficiente $\delta = 0.01$.
- Parámetro para la condición de curvatura $\sigma = 0.1$.
- Parámetro para la condición de reinicio $\mu = 0.9$.
- Parámetro para la condición de reinicio $\nu = 0.4$.
- Parámetro para la dirección de reinicio $\eta = 0.04$.

Usando estos parámetros, se hace una comparación con los demás métodos: Hestenes-Stiefel (HS), Fletcher-Reeves (FR), Polak-Ribière-Polak (PRP), Fletcher (CD), Liu-Storey (LS) y Dai-Yuan (DY), en puntos específicos para las funciones:

- Función de cuadrática 1: Para $\mathbf{x} = (x_1, x_2, \dots, x_n)$

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A}_1 \mathbf{x} - \mathbf{b}_1^T \mathbf{x}.$$

- Función de Himmelblau para $\mathbf{x} = (x_1, x_2)$

$$f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

- Función de Beale para $\mathbf{x} = (x_1, x_2)$

$$f(\mathbf{x}) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2.$$

- Función de Rosenbrock para $\mathbf{x} = (x_1, x_2, \dots, x_n)$

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad n \geq 2.$$

Resultando en que el comportamiento del método IJYHL es mejor en la mayoría de los casos para los ejemplos tratados, mostrando un mejor desempeño tanto en el número de iteraciones como generando convergencia donde los demás métodos no pudieron, como en el caso de la función Rosenbrock (en el archivo [PROY_OP_EFTT.ipynb](#) se muestran más detalles de este análisis como los puntos mínimos encontrados y el número de reinicios). Además, para las funciones Himmelblau, Beale y Rosenbrock, se generaron 20 puntos aleatorios, en los cuales se revisó el comportamiento de los distintos métodos, generando los resultados siguientes:

Para la función Himmelblau se generaron 20 puntos aleatorios en el intervalo $[-4, 4] \times [-4, 4]$:

MÉTODO	CONVERGENCIA (%)	ITERACIONES
Fletcher-Reeves	0.8	420.0
Hestenes-Stiefel	0.85	320.95
Polak-Ribière-Polak	0.8	419.95
Dai-Yuan	0.85	320.95
Fletcher	0.8	420.0
Liu-Storey	0.8	419.95
IJYHL	0.85	319.0

Para la función Beale se generaron 20 puntos aleatorios en el intervalo $[-2, 2] \times [-2, 2]$:

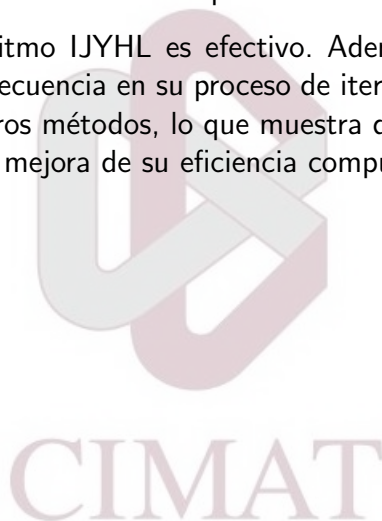
MÉTODO	CONVERGENCIA (%)	ITERACIONES
Fletcher-Reeves	0.65	767.05
Hestenes-Stiefel	0.65	748.05
Polak-Ribière-Polak	0.7	701.15
Dai-Yuan	0.65	748.05
Fletcher	0.7	701.85
Liu-Storey	0.7	699.35
IJYHL	0.75	573.75

Para la función Rosenbrock se generaron 20 puntos aleatorios en el intervalo $[-1, 2] \times [-1, 2]$:

MÉTODO	CONVERGENCIA (%)	ITERACIONES
Fletcher-Reeves	0.0	2000.0
Hestenes-Stiefel	0.95	931.75
Polak-Ribière-Polak	0.95	883.25
Dai-Yuan	0.95	931.75
Fletcher	0.0	2000.0
Liu-Storey	0.95	1025.2
IJYHL	1.0	420.5

En las tres tablas anteriores podemos ver que, como los autores proponen, el desempeño del método IJYHL es mejor que el resto, por lo menos en los ejemplos revisados en este trabajo. En las diversas pruebas hechas por los autores se hacen comparaciones con una gran cantidad de funciones prueba poniendo principal atención en tres índices de comparación: El número de iteraciones, el tiempo de cálculo de la CPU y el valor de gradiente g_k al final de la iteración, y particularmente para el algoritmo IJYHL también probaron el número de reinicios.

Así que se concluye que el algoritmo IJYHL es efectivo. Además, vale la pena señalar que el algoritmo IJYHL se reinicia con frecuencia en su proceso de iteración, y su rendimiento numérico es obviamente mejor que el de otros métodos, lo que muestra que los procedimientos de reinicio juegan un papel importante en la mejora de su eficiencia computacional.

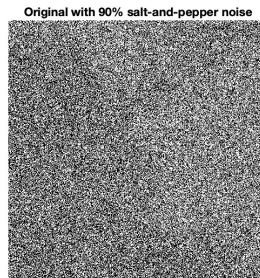


4.2. Problemas de restauración de imagen

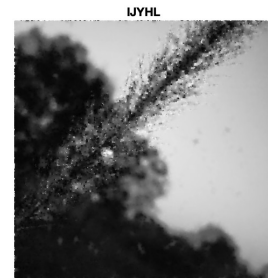
Usando la metodología propuesta por los autores, se adaptó su algoritmo para implementar el método IJYHL para restaurar imágenes que habían sido contaminadas por ruido *salt & pepper* minimizando la función de error generado por el ruido, generando los resultados:



(a)



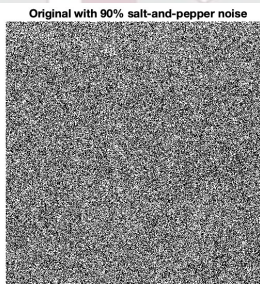
(b)



(c)



(d)



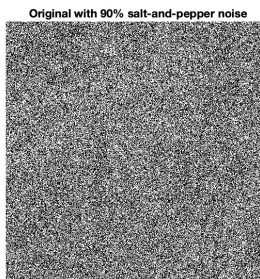
(e)



(f)



(g)



(h)



(i)

Referencias

- [1] Nazareth J.L. "Conjugate-gradient methods". En: *Encyclopedia of Optimization*. Kluwer Academic Publishers, Boston. (1999).
- [2] Xiaomin Y. Xianzhen J. y Zefeng H. *A family of hybrid conjugate gradient method with restart procedure for unconstrained optimizations and image restorations*. 2023.
- [3] L. Zhang. "An improved Wei-Yao-Liu nonlinear conjugate gradient method for optimization computation". En: *Appl. Math. Comput.* 215 (6), 2269–2274 (2009).
- [4] X.Z. Jiang y J.B. Jian. "Wolfe line search for the next new globally convergent conjugate gradient method". En: *Chinese J. Engrg. Math.* 28 (6), 779–786 (2011).
- [5] C.X. Kou e Y.H. Dai. "A modified self-scaling memoryless Broyden–Fletcher–Goldfarb–Shanno method for unconstrained optimization". En: *J. Optim. Theory Appl.* 165 (1), 209–224 (2015).
- [6] G. Zoutendijk. *Nonlinear programming computational methods*. 1970.
- [7] X.Z. Jiang y J.H. Yin. "A new family of hybrid three-term conjugate gradient methods with applications in image restoration". En: *Numer. Algorithms* 91, 161–191 (2022).

