OPTIMIZACIÓN NO LINEAL

Enero 2021

De la Torre Ortíz Bibiana
García de la Cruz Semiramís

Índice general

Primera Parte

1. Introducción

En optimización no lineal, estudiamos métodos que encuentran el conjunto de valores para obtener la mejor solución o la más aproximada, la cual puede estar sujeta a una serie de restricciones, donde una o más variables incluidas es no lineal.

En el presente proyecto, nos centraremos en aquellos problemas que no tienen restricciones y compararemos la eficacia de los métodos vistos, que son los siguientes

1. Método de Máxima Pendiente *(Cauchy)*
2. Condición de Armijo
3. Método de Newton
4. Método de Cuasi-Newton
5. Método de Gradientes Conjugados

Para el análisis, tomaremos cuatro funciones de prueba, de las cuales tenemos resultados esperados para luego compararlos con los resultados obtenidos en cada programa. Además, tendremos en cuenta el tiempo y número de evaluaciones de la función objetivo, puesto que no sólo se requiere que los métodos sean exactos sino también eficientes, es decir, preferiremos un método que disminuya el costo computacional.

Por ser un problema de optimización, es evidente que la finalidad es que encontremos el óptimo de la función, dicho de otro modo, el mínimo global si es que existe. Cabe resaltar, que cada algoritmo arrojará la mejor aproximación a dicho punto.

2. Marco teórico

2.1 Marco conceptual

Antes de comenzar, debemos mencionar conceptos relevantes que usaremos de aquí en adelante en el presente trabajo.

Optimización Acción y efecto de optimizar. Buscar la mejor manera de realizar una actividad.[1]

Función objetivo Es la expresión matemática del cuya meta debe ser maximizar o minimizar esa expresión. La función objetivo lineal se puede representar de como una ecuación o en sumatorias.[1]

Variables Como variables independientes se suelen elegir aquellas que tienen un impacto significativo sobre la función objetivo.[1]

Restricciones Son ecuaciones o inecuaciones con relaciones existentes entre las variables de decisión. Limitan el sistema, a leyes naturales o a limitaciones tecnológica.[3]

Entonces a partir de la función objetivo, planteada para cuantificar el rendimiento y medir la calidad de una decisión, se obtendrán los valores para cierto número de variables, relacionadas entre sí, de manera que minimicen dicha función, teniendo en cuanta por lo general una serie de restricciones.

Definition 2.1.1 — Problema de optimización no lineal. La formulación matemática de un problema de optimización no lineal es

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g_i}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{h_j}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{2.1}$$

Entonces, si $\mathbf{x}^*$ es un punto óptimo de f, al valor $f(\mathbf{x}^*)$ se le llama valor óptimo de la función. En las fases anteriormente mencionadas, se encuentra la verificación de la solución que implica el cumplimiento de las condiciones de primer y segundo orden. Análogamente, en funciones multivariables tendremos condiciones de optimalidad. En este caso, un mínimo se presentará

cuando

$$\Delta f(\mathbf{x}) = f(\mathbf{x}) - f(\mathbf{x}^*) \leq 0$$

Equivalentemente, la derivada direccional debe ser **no negativa**, puesto que eso significaría que la función podría hacer un movimiento en d con el que la f sufriría un decremento, contradiciendo que es un mínimo.

Theorem 2.1.1 — Condición necesaria. Si $\mathbf{x}^*$ es un punto mínimo local, entonces necesariamente se debe cumplir que

$$\Delta f(\mathbf{x}^*) = 0 \tag{2.2}$$

es decir, $\mathbf{x}^*$ es un **punto crítico o estacionario**.

Theorem 2.1.2 — Condición suficiente. Si $\mathbf{x}^*$ es un punto estacionario, entonces $\mathbf{x}^*$ es un punto

- **Mínimo** si $\Delta f(\mathbf{x}^*)^2$ es definida positiva.
- **Máximo** si $\Delta f(\mathbf{x}^*)^2$ es definida negativa.
- **Silla**, o de inflexión, si $\Delta f(\mathbf{x}^*)^2$ es indefinida. positiva.

2.2 Métodos

La optimización numérica en funciones no lineales, requiere la utilización de algoritmos eficientes y robustos. Para conocer las posibilidades y limitaciones de cada método, recurrimos a la experimentación.

2.2.1 Método de Máxima Pendiente

También conocido como *Método de Cauchy* o *Steepest Descent*, nos lleva a encontrar mínimos locales en funciones en $\mathbb{R}^n$, de manera iterativa con el objetivo de encontrar el punto óptimo. Este método es extremadamente importante desde el punto de vista teórico, por ser uno de los más sencillos con resultados satisfactorios. Muchos de los algoritmos más avanzados usualmente se basan en modificar esta técnica. El método de Cauchy, es definido por el algoritmo iterativo

$$x_{k+1} = x_k - \alpha_k g_k$$

que quiere decir que para cada iteración realizaremos dos cálculos. Primero d_k, que es la dirección de búsqueda de la cual se generarán los puntos x_{k+1} y también α_k el tamaño de paso (avance) que se hará en la dirección definida. [4]

Para el cálculo de d_k, se requiere información de primer orden (e.i. primeras derivadas), mientras que para α_k cualquier dirección cuya derivada direccional sea negativa (e.i. $\langle \Delta f, d \rangle < 0$), que es la máxima dirección de decremento.

Notation 2.1. *El cálculo de el α_k como la máxima dirección de descenso, se puede expresar como*

$$\alpha_k = arg\ mín_{\alpha_k \leq 0}\{f(x_k + \alpha_k d_k)\}$$

Notemos que, los movimientos que esta técnica se mueve a lo largo del cálculo de x_k, aparente-

mente en pasos ortogonales. En resumen, el algoritmo del método sería el siguiente

Algorithm 1: Descenso con pendiente máxima

Require: Punto inicial x_0, función derivable $f(x) : \mathbb{R}^2 \to \mathbb{R}$
Result: Punto óptimo local x^*.
k = 0;
while $\nabla f(\mathbf{x}) \neq 0$ **do**
  $d_k = -\nabla f(\mathbf{x}^k)$;
  Calcular $\alpha_k = \arg\min_{\alpha_k \leq 0}\{f(x_k + \alpha_k d_k)\}$;
  $x_{k+1} = x_k + \alpha_k d_k$;
  k = k + 1;
end
return $x^* = x_k$;

2.2.2 Condición de Armijo

Este es un criterio, práctico y popular para aplicar una búsqueda lineal, de la función $f(x_k + \alpha d_k)$, así se determina un intervalo donde se encuentran los valores de α que son buenas estimaciones de α_k. A este intervalo le llamaremos intervalo de valores admisibles de α_k.

Definition 2.2.1 — Condición de Armijo. Para determinar si el decremento en f es suficiente, se evalúa la siguiente desigualdades y si ésta se cumple, se acepta como α_k al correspondiente valor de α tal que:

$$f(x_k + \alpha v_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T v_k \tag{2.3}$$

para alguna constante $c_1 \in (0, 1)$.

2.2.3 Método de Newton

El método de Newton (a veces llamado Newton-Raphson) utiliza primeras y segundas derivadas (información de primer y segundo orden) y de hecho funciona mejor que el método de Cauchy si el punto inicial está cerca del vector mínimo.

La idea detrás de este método es la siguiente: Dado un punto de partida x_k , construimos una aproximación cuadrática q de la función objetivo f, haciendo coincidir la primera y segunda derivada de ambas funciones, f y q valuadas en el punto x_k. [4] Luego minimizamos la función aproximada q (cuadrática) en vez de la función objetivo original f; este paso es simple dado que la función q fue propuesta y su óptimo es conocido. Luego, utilizamos el vector x_{k+1} mínimo de la función aproximada como punto de partida en el siguiente paso y repetimos el procedimiento iterativamente.

Lo podemos expresar de mejor manera en el siguiente algoritmo. [2]

Algorithm 2: Método de Newton Multivariable

Require: Punto inicial x_0, función dos veces diferenciable $f(x) : \mathbb{R}^2 \to \mathbb{R}$
Result: Punto óptimo local x^*.
k = 0;
while $\nabla f(\mathbf{x}) \neq 0$ **do**
  $d_k = -H_k^{-1} g_k$;
  $x_{k+1} = x_k + d_k$;
  k = k + 1;
end
return $x^* = x_k$;

2.2.4 Método de Cuasi-Newton

Hemos visto que el método de Newton tiene dos elementos que implican cierto costo computacional y que pueden acarrear imprecisiones y/o errores.

- El cálculo de la matriz Hessiana.
- La solución de un sistema (posiblemente grande) de ecuaciones lineales.

Rango 1

Es por eso que se recurre al método de Cuasi-Newton ya que en cada iteración se crea una aproximación a la inversa del hessiano la cual se obtiene sumando a la anterior una matriz de actualización H_k simétrica, semidefinida positiva y de rango uno, que puede escribirse en la forma:

$$H_{k+1} = H_k + \frac{(\Delta x^{(k)} - H_k \Delta g^{(k)})(\Delta x^{(k)} - H_k \Delta g^{(k)^T})}{\Delta g^{(k)^T}(\Delta x^{(k)} - H_k \Delta g^{(k)})}$$

En la práctica, tomaremos como $H_0 = \mathbb{I}$. Ahora bien, la actualización de la matriz H puede realizarse de diversas formas, con la única restricción de que se satisfaga:

$$d_k = x_{k+a} - x_k = -H\left(x^k\right)^{-1} g_k$$

Por lo tanto, si la aproximación inicial es simétrica, se garantiza la simetría, aunque no la definición de positiva, de las aproximaciones siguientes.

Rango 2

También conocido como *método de métrica variable* o de *Davidson-Fletcher-Powell* por sus autores. Partiendo de una aproximación inicial simétrica y definida positiva, nos aseguramos de la simetría y la definición positiva de las aproximaciones siguientes, cuya forma es:

$$H_{k+1} = H_k + \left(1 + \frac{\Delta g^{(k)^T} H_k \Delta g^{(k)}}{\Delta g^{(k)^T} \Delta x^{(k)}}\right)\left(\frac{\Delta x^{(k)} \Delta x^{(k)^T}}{\Delta x^{(k)^T} \Delta g^{(k)}}\right) - \frac{H_k \Delta g^{(k)}\left(H_k \Delta g^{(k)} \Delta x^{(k)^T}\right)^T}{\Delta g^{(k)^T} \Delta x^{(k)}}$$

Es posible que la expresión H_{k+1}, nos lleve a calcular iterativamente direcciones conjugadas, por lo que el método asociado converge en n (número de variables) iteraciones para funciones cuadráticas. Aunque no podemos asegurar que la matriz hessiana no se vuelva singular.

Algorithm 3: Método de Cuasi-Newton Rango 1

Require: Punto inicial x_0, función derivable $f(x) : \mathbb{R}^2 \to \mathbb{R}$, matriz H_0 inicial
Result: Punto óptimo local x^*.
k = 0 ; `// Puede tomarse` $H_0 = I_n$
while $\nabla f(\mathbf{x}^k) \neq 0$ **do**
  $g_k = -\nabla f(\mathbf{x}^k)$;
  $d_k = -H_k g^k$;
  Calcular $\alpha_k = \arg\min_{\alpha_k \leq 0}\{f(x_k + \alpha_k d_k)\}$;
  $x_{k+1} = x_k + \alpha_k d_k$;
  $g_{k+1} = \nabla f(\mathbf{x}_{k+1})$;
  if $g_{k+1} \neq 0$ **then**
    $\Delta \mathbf{x}^k = \alpha_k d_k$;
    $\Delta g_k = g^{k+1} - g^k$;
    $H_{k+1} = H_k + \frac{(\Delta x^{(k)} - H_k \Delta g^{(k)})(\Delta x^{(k)} - H_k \Delta g^{(k)^T})}{\Delta g^{(k)^T}(\Delta x^{(k)} - H_k \Delta g^{(k)})}$;
  end
  k = k + 1
end
return $x^* = x_k$;

El algoritmo de Rango 2, solo difiere en el cálculo de de H.

2.2.5 Método de Gradientes Conjugados

Es un método de descenso del gradiente para minimizar funciones cuadráticas convexas $f : \mathbb{R}^n \to \mathbb{R}$, también es en sí mismo el algoritmo de direcciones conjugadas.

Durante la ejecución del algoritmo, es posible construir las direcciones conjugadas (tomando a cada iteración k una combinación lineal de la dirección previa d^{k-1} y el gradiente g^k en la iteración actual), ya que se usa la información del gradiente de f a cada iteración.

También se puede aplicar este método en funciones no cuadráticas, usando la serie de Taylor para la aproximación de la función objetivo, en el punto en cuestión, sin embargo la matriz Hessiana Q debe reevaluarse en cada iteración del algoritmo y eso implica un costo computacional considerable, por ello el tamaño de paso se puede obtener con una búsqueda lineal. Para el calculo de las direcciones conjugadas $d^{(k)}$ sin necesidad de Hessianas, se ocupa la fórmula de Fletcher-Reevers:

$$\beta_k = \frac{g^{(k+1)^T} g^{(k+1)}}{g^{(k)^T} g^{(k)}}$$

El algoritmo a este método se puede observar a continuación.

Algorithm 4: Método de Gradientes Conjugados

Require: Punto inicial x_0, $f : \mathbb{R}^2 \to \mathbb{R}$ derivable y cuadrática
Result: Punto óptimo local x^*.
k := 0 , $g^0 = \nabla f(\mathbf{x}^{(0)})$; $d^0 = -g^0$;
if $g^{(k)} \neq 0$ **then**
  $\alpha_k = -\frac{g^{(k)^T} d^{(k)}}{d^{(k)^T} Q d^{(k)}}$;
  $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$;
  $g^{(k+1)} = \nabla f(x^{(k+1)})$;
  if $g^{k+1} = 0$ **then**
    Terminar
  else
    $\beta_k = -\frac{g^{(k+1)^T} Q d^{(k)}}{d^{(k)^T} Q d^{(k)}}$;
    $d^{(k+1)} = -g^{(k+1)} + \beta_k d^{(k)}$;
    $k := k+1$ ir al paso 3;
  end
end

3. Metodología

En este apartado, describiremos el conjunto de procedimientos y técnicas empleadas para llevar a cabo el análisis de cada función, enfatizando en la selección de dichas funciones y los métodos usados en cada uno. En este apartado se describe la metodología que se utilizo para llevar acabo el análisis de cada función.

3.1 Funciones

Las funciones seleccionadas[1] cumplen con ciertas características que nos sirven como punto de partida para decidir si se va a hacer uso de la técnica **Multistart** para hallar mínimos locales y globales; tener información de la función objetivo nos ayuda a intuir en que métodos puede fallar algún programa.

3.1.1 Función Six Hump Camel

$$f(x) = \left(4 - 2{,}1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \tag{3.1}$$

Clasificación: Pertenece al conjunto de funciones en forma de valle.
Dominio: La función generalmente se evalúa en el rectángulo $x_1 \in [-3,3], x_2 \in [-2,2]$.
Descripción y características:

- La función es continua.
- La función no es convexa.
- La función se define en un espacio bidimensional.
- La función es multimodal.
- La función es diferenciable.
- La función no es separable.

[1] De la librería `https://www.sfu.ca/~ssurjano/index.html`

Mínimo global: $x^* = (0{,}0898, -0{,}7126)$, $x^* = (-0{,}0898, 0{,}7126)$, evaluada en la función objetivo obtenemos $f(x^*) = -1{,}0316$
Por la naturaleza de la función, hará uso de un multistart debido a que a función tiene más de un mínimo global.

3.1.2 Función Easom

$$f(x) = -\cos(x_1)\cos(x_2)exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right) \tag{3.2}$$

Clasificación: Pertenece al conjunto de funciones con crestas empinadas (o caídas).
Dominio: La función generalmente se evalúa en el cuadrado $x_i \in [-100, 100]$, para todo $i = 1, 2$.
Descripción y características:

- La función es continua.
- La función no es convexa.
- La función se define en un espacio bidimensional.
- La función es unimodal.
- La función es diferenciable.
- La función no es separable.
- La función no es escalable

Mínimo global: $x^* = (\pi, \pi)$, evaluada en la función objetivo obtenemos $f(x^*) = -1$.
Ya que la función es diferenciable, se pueden emplear todos los métodos sin temor a que se generen errores como alguna matriz singular en el método de newton, esta función también es multimodal sin embargo se decidió no aplicar un multistart.

3.1.3 Función Coss in Tray

$$f(x) = -0{,}001\left(\left|sin(x_1)sin(x_2)exp\left(\left|100-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right|\right)\right|+1\right)^{0{,}1} \tag{3.3}$$

Clasificación: Pertenece al conjunto de funciones con muchos mínimos locales.
Dominio: generalmente se evalúa dentro del dominio: $x_i \in [-10, 10]$, para todo $i = 1, 2$.
Descripción y características:

- La función es continua.
- La función no es convexa.
- La función se define en un espacio bidimensional.
- La función es multimodal.
- La función no es diferenciable.

Mínimo global: $x^* = (\pm 1{,}3491, \pm 1{,}3491)$, donde los cuatro puntos se evalúan en la función objetivo $f(x^*) = -2{,}06261$.
Por ser la única función que no es diferenciable, no se podrá implementar el método de Newton.

3.1.4 Función Griewank

$$f(x) = \frac{x_1^2}{4000}+\frac{x_2^2}{4000}-\cos(x_1)\cos\left(\frac{x_2}{\sqrt{2}}\right)+1 \tag{3.4}$$

Clasificación: Esta función también entra dentro de la clasificación de funciones con muchos mínimos globales.
Dominio: Se evalúa en el hipercubo $x_i \in [-600, 600]$, para todo $i = 1, 2$. **Descripción y características:**

- La función es continua.
- La función no es convexa.
- La función se puede definir en un espacio n-dimensional.
- La función es multimodal.

Mínimo global: $x^* = (0, 0)$, que evaluado nos da $f(x^*) = 0$.
Estos mínimos locales generalizados se distribuyen regularmente. La función es de dos dimensiones; pero para comodidad del proyecto utilizaremos un dominio más pequeño.

3.1.5 Función Trid

$$f(x) = (x_1 - 1)^2 + (x_2 - 1)^2 - x_1 x_2 \tag{3.5}$$

Clasificación: La función Trid pertenece al conjunto de funciones en forma de cuenco.
Dominio: La función generalmente se evalúa en el hipercubo $x_i \in [-2^2, 2^2]$, para todo $i = 1, 2$.
Descripción y características:

- La función es continua.
- La función es bidimensional.
- La función es convexa.
- La función se puede definir en un espacio n-dimensional.
- La función es unimodal.

Mínimo global: $x^* = (2, 2)$ y evaluada en la función objetivo obtenemos $f(x^*) = -2$.

3.1.6 Función McCormick

$$f(x) = sen(x_1 + x_2) + (x_1 - x_2)^2 - 1{,}5x_1 + 2{,}5x_2 + 1 \tag{3.6}$$

Clasificación: Pertenece al conjunto de funciones en forma de placa, es de dos dimensiones y la función.
Dominio: Generalmente se evalúa en el rectángulo $x_1 \in [-1{,}5, 4], x_2 \in [-3, 4]$.
Descripción y características:

- La función es continua.
- La función es convexa.
- La función se define en un espacio bidimensional.
- La función es unimodal en el dominio.
- La función es diferenciable.

Mínimo global: $x^* = (-0{,}54719, -1{,}54719)$, que evaluada en la función nos da $f(x^*) = -1{,}9133$.

3.2 Secuencia de Análisis

Para un mejor control del análisis, decidimos optar por una estructura

1. Descripción de la función. Se presenta la función son su dominio, su mínimo local o global, según sea el caso en cada una y con su gráfica correspondiente.
2. Hipótesis. Realizamos un análisis previo, comparando la función con los métodos elegidos y cuáles son los problemas que se podrían presentar.

3. Ejecución del programa. Ejecutamos los métodos ya programados con los parámetros requeridos, así obteniendo una tabla de resultados con todos los métodos y los parámetros de salida. En este punto puede ser ejecutados para un solo punto o para varios, todo de acuerdo a las características de la función, ya que algunas necesitaran el proceso multistart.
4. Análisis de los resultados. La tabla de resultados que se obtiene en el punto anterior viene dada con los 4 métodos utilizados y se obtienen 5 criterios. La comparación será bajo los 5 criterios de cada método y se interpretará.
5. Conclusión.

3.3 Experimentos

Los parámetros usados en cada experimento, es decir, por función serán los mismos, esto con el objetivo de comparar el desempeño de los algoritmos para un mismo problema. Cabe mencionar que por la naturaleza de la función el dominio de entrada es diferente.

3.3.1 Parámetros de las funciones

Para las funciones unidmodales ocupamos los siguientes parámetros.

- Función Eason
  Parámetros entrada
  x0 = [3,3]
  eps = 0.001
- Función Trid
  Parámetros entrada
  x0 = [1,1]
  eps = 0.001
- Función McCormick
  Parámetros entrada
  x0 = [0,-1]
  eps = 0.001

Por otro lado, para las funciones multimodales tuvimos las siguientes consideraciones:

- Función Six Hump Camel
  Parámetros entrada
  Generación de 100 puntos uniformemente distribuidos en $[-3,3] \times [-2,2]$. Tomando cada uno de estos puntos como punto inicial.
  eps2 = 0.001
- Función Griewank
  Parámetros entrada
  Generación de 100 puntos uniformemente distribuidos en $[-10,10] \times [-10,10]$. Tomando cada uno de estos puntos como punto inicial.
  eps2 = 0.001
- Función Cross in Tray
  Parámetros entrada
  Generación de 100 puntos uniformemente distribuidos en $[-10,10] \times [-10,10]$. Tomando cada uno de estos puntos como punto inicial.
  eps2 = 0.001

3.3.2 Parámetros de los Métodos

Para el análisis de las funciones, consideramos los siguientes métodos usando los parámetros mostrados a continuación.

- Parámetros de entrada:
  x0 = punto inicial
  eps = tolerancia para la condición de paro, $|(g(x)| < eps$
- Parámetros durante la ejecución:
  α_k : Tamaño de paso (dependiendo del método)
  d^k : Dirección de descenso (dependiendo del método)
- Parámetros de salida:
  opt = punto óptimo
  fopt = Evaluación del óptimo en la función
  llamadas_f = número de invocaciones a la función f
  Tiempo de ejecución del algoritmo.

En general, todos los parámetros son iguales en cada algoritmo; sin embargo, hay variación en el cálculo del tamaño de paso y la dirección de descenso, ya que no siempre se ocupan ambos, por ello especificamos cuál usamos dependiendo del método.
Además los programas: `funcion.m`, `grad.m`, `hessiana.m`, `newton_univar.m` y `acotamiento.m`, (elaborados en MATLAB y anexados en el proyecto) están definidos como funciones que se mandan a llamar para los diferentes métodos, evitando así funciones anónimas.

3.3.3 Método de Máxima Pendiente

El uso de este método es porque aproxima el mínimo de la función de manera iterativa haciendo pasos ortogonales a lo largo del cálculo de los puntos x^k. Además, cuenta con una convergencia lineal, por lo que se aproxima lentamente, siendo un dato de gran importancia al comparar los métodos. Aplicamos el programa `max_pend_newton.m` elaborado en MATLAB y anexado en el proyecto, para el análisis del experimento.
Donde, la función objetivo debe ser derivable $f(x) : \mathbb{R}^n \to \mathbb{R}$.
Definición:

[opt,fopt,it_pend,it_alpha,llamadas_f] = max_pend_newton(x0,eps)

Parámetro durante la ejecución

α_k : Tamaño de paso usando el método de Newton univariable.

Ejemplo de uso

Desde la línea de comandos introducimos la definición de la función objetivo f al método:
[opt,fopt,iter_pend,it_aplha,llamadas] = max_pend_newton([0,0],0.001)

3.3.4 Condición de Armijo

Usamos este método por que determina un decremento en f suficiente
Aplicamos el programa `max_pend_armijo.m` elaborado en MATLAB y anexado en el proyecto, para el análisis del experimento.
Donde, la función debe ser continúa y derivable $f(x) : \mathbb{R}^n \to \mathbb{R}$
Definición:

[opt,fopt,it_pend,it_alpha,llamadas_f2,llamadas_g2] = max_pend_armijo(x0,eps)

Parámetros durante la ejecución

$\alpha_k = 1$: Tamaño de paso inicial.
α_k : Tamaño de paso usando la condición de Armijo.

Ejemplo de uso

Desde la línea de comandos introducimos la definición de la función objetivo f al método:
[opt,fopt,it_pend,it_alpha,llamadas_f2,llamadas_g2] = max_pend_armijo([0,0],0.001)

3.3.5 Método de Newton

Este método presenta propiedades de convergencia superiores a cualquier otro método, siempre y cuando el punto de partida este cerca de la solución. Además la forma recursiva del algoritmo aplica la condición de optimalidad de primer orden, por lo que se alcanza un mínimo en x^*.
Aplicamos el programa `newton_mult.m` elaborado en MATLAB y anexado en el proyecto, para el análisis del experimento.
Donde, la función debe admitir dos derivadas continúas $f(x) : \mathbb{R}^n \to \mathbb{R}$
Definición:

[opt,fopt,llamadas_f1] = newton_mult(x0,eps1)

Parámetro durante la ejecución

$d^k =$: dirección de descenso

Nota: Dado que para obtener d^k se resuelve un sistema de $n \times n$ ecuaciones lineales, un método eficiente para resolverlo es esencial.
Ejemplo de uso

Desde la línea de comandos introducimos la definición de la función objetivo f al método:
[opt,fopt,cont_met,llamadas_f] = newton_mult([0,0],0.001)

3.3.6 Método de Cuasi Newton

Dado que el algoritmo es una mejorá del método de Newton, al no cálcular el Hessiano, presenta una covergencia superlineal.
Aplicamos el programa `cuasi_newton.m` elaborado en MATLAB y anexado en el proyecto, para el análisis del experimento.
Donde, la función es derivable $f(x) : \mathbb{R}^n \to \mathbb{R}$
Definición:

[xm,fxm, iter,llamadas_f]=cuasi_newton(xm,epsilon)

Parámetro durante la ejecución

$H_0 = H_0^T > 0$: Matriz Hessiana
$\alpha_0 = 1$: Tamaño de paso inicial
α_k : Tamaño de paso usando el método de Sección dorada

Ejemplo de uso

Desde la línea de comandos introducimos la definición de la función objetivo f al método:
[xm,fx0,iter,llamadas_f]=cuasi_newton([1,1],0.001)

3.3.7 Método de Gradientes Conjugados

Este método se aplica a funciones convexas y cuadráticas donde en único mínimo global es el punto x, que es solución de $Qb = x$, pero también lo podemos aplicar si la función cuadrática se interpreta como la aproximación de la función objetivo, mediante la serie de serie de Taylor. Por ello es buen algoritmo a usar.
Aplicamos el programa `grad_conjugados.m` elaborado en MATLAB y anexado en el proyecto, para el análisis del experimento.
Donde, la función es derivable y cuadrática $f(x) : \mathbb{R}^n \to \mathbb{R}$
Definición:

[xm,fxm, iter,llamadas_f]=cuasi_newton(xm,epsilon)

Parámetro durante la ejecución

α_k : Tamaño de paso usando el método de Armijo
d^k : Dirección de descenso

Ejemplo de uso

Desde la línea de comandos introducimos la definición de la función objetivo f al método:
[x0,fx0,llamadas_f] = grad_conjugados([1,1],0.01)

3.3.8 Método Multistart

Este algoritmo posee un enfoque iterativo en el cual, a través de cada iteración, se lleva a cabo una búsqueda de óptimos del problema. Su propósito es inspeccionar otras soluciones en otras regiones dentro del espacio solución y hallar aquellas que proporcionen un mejor valor de la función objetivo.
Aplicamos el programa `Multistart.m` elaborado en MATLAB y anexado en el proyecto, para el análisis del experimento.
Donde, la función es multimodal
Parámetros de salida:

opt(1) = Entrada uno del vector del punto óptimo
opt(2) = Entrada dos del vector del punto óptimo
fopt = Evaluación del óptimo en la función
llamadas_f = número de invocaciones a la función f
Tiempo de ejecución del algoritmo.
Documento en excel con el nombre 'test.xlxs' , donde se encuentra la matriz $[101,4] \times [101,4]$ con todos los parámetros de salida.

Para el uso del método, solo es necesario ejecutarlo, variando el dominio de entrada.

II

Segunda Parte

4. Hipótesis

Cuando tomamos cursos de Cálculo vectorial y buscamos aproximar el mínimo de una función usamos métodos directos; sin embargo, para este trabajo implementamos los métodos de optimización ya mencionados y analizamos el comportamiento de cada uno.

La función **Six Hump Camel** presenta cierta dificultad debido a que cuenta con varios mínimos locales, por lo que emplearemos un multistart. Debido a sus características, la hipótesis es no tener alguna complicación en los métodos ya que su dominio es considerablemente menor a la mayoría de las funciones presentadas.

La función **Eason** puede parecer sencilla, dado que presenta un único mínimo global en el medio una gran planicie. El reto que ofrece esta función se relaciona con el hecho de poder encontrar verdaderamente el mínimo global en el menor número de iteraciones, es decir buscamos que el método empleado sea efectivo.

La función **Cross in Tray**, al igual que la función Griewank, presenta su complejidad en el hecho de que es una función multimodal. El riego está en que al evaluar en un punto aleatorio podríamos llegar a un mínimo local solamente. Además, por ser no diferenciable dificulta el funcionamiento del método de Newton, sin embargo, una ventaja si la comparamos con la función Griewank es que el dominio de búsqueda es más pequeño.

La función **Griewank** es considerada una función con un nivel alto de complejidad, ya que cuenta con muchos máximos y mínimos distribuidos. Esta característica, suponen un desafío para puesto que presenta un mínimo global y muchos locales. Dicho esto, con el uso de la técnica multistart, esperamos encontrar los puntos que se presenten en el intervalo dado.

La función **Trid** es relativamente sencilla de analizar. No presenta ningún tipo de dificultad debido a su naturaleza,pero aun así existe el riesgo (aunque sea pequeño) de que el algoritmo utilizado arroje un punto que sea tomado como mínimo sin que este lo sea estrictamente.

La función **McCormick** cuenta con un solo mínimo global que la hace un poco más fácil de manejar, aunque la forma de placa de la función nos podría anticipar un gran riesgo de que el programa arroje un punto que sea tomado como mínimo sin que este lo sea estrictamente. Echaremos mano de un multistart para analizar todos los puntos.

Para respaldar las las ideas, mencionadas en esta sección, también nos apoyaremos en las

gráficas de las funciones.

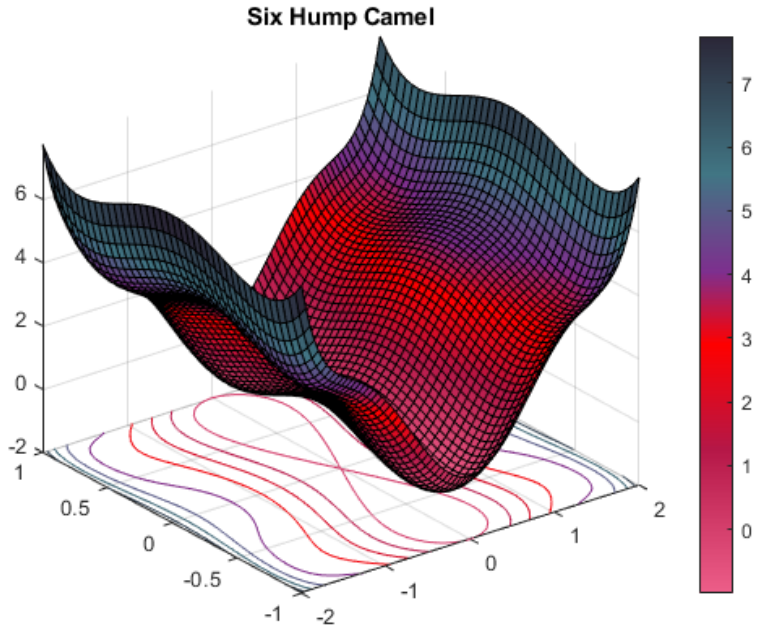

Figura 4.1: Six-Hump Camel function

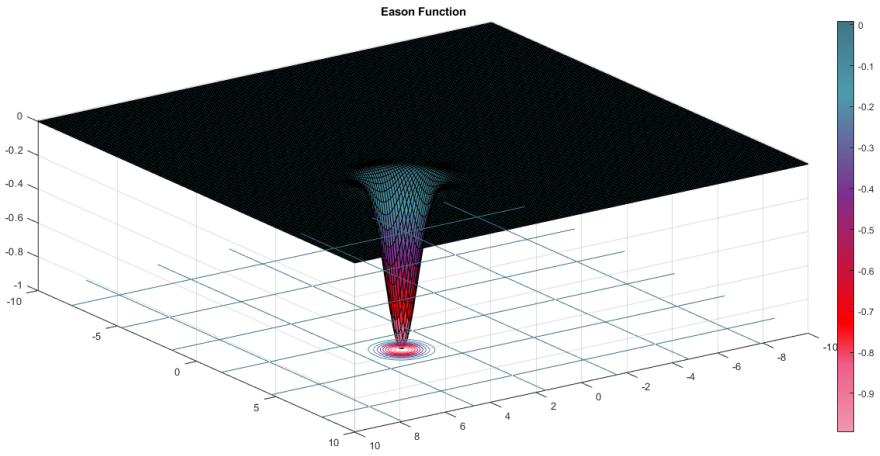

Figura 4.2: Eason function

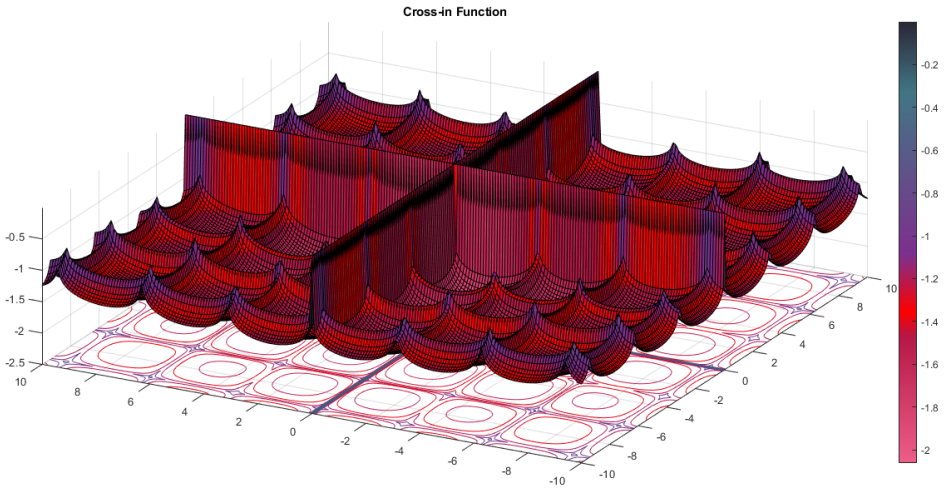

Figura 4.3: Cross in Tray function

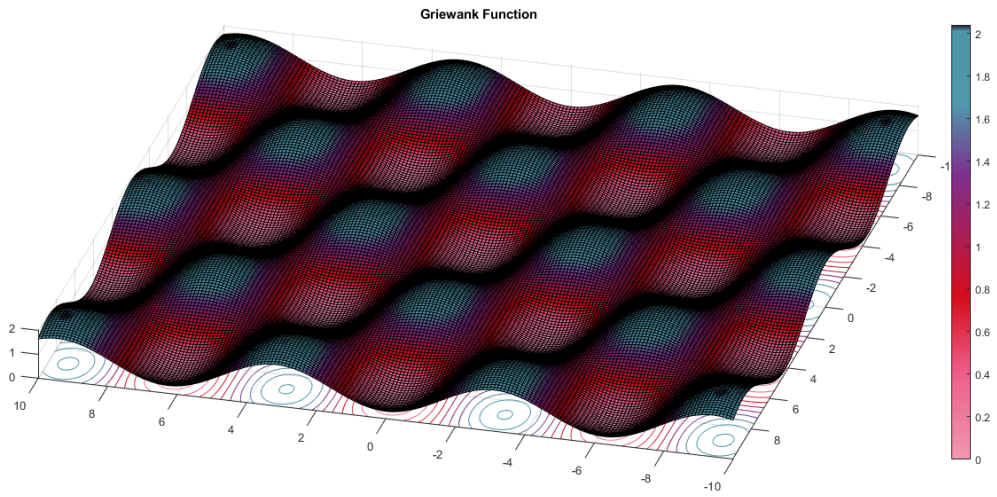

Figura 4.4: Griewank function

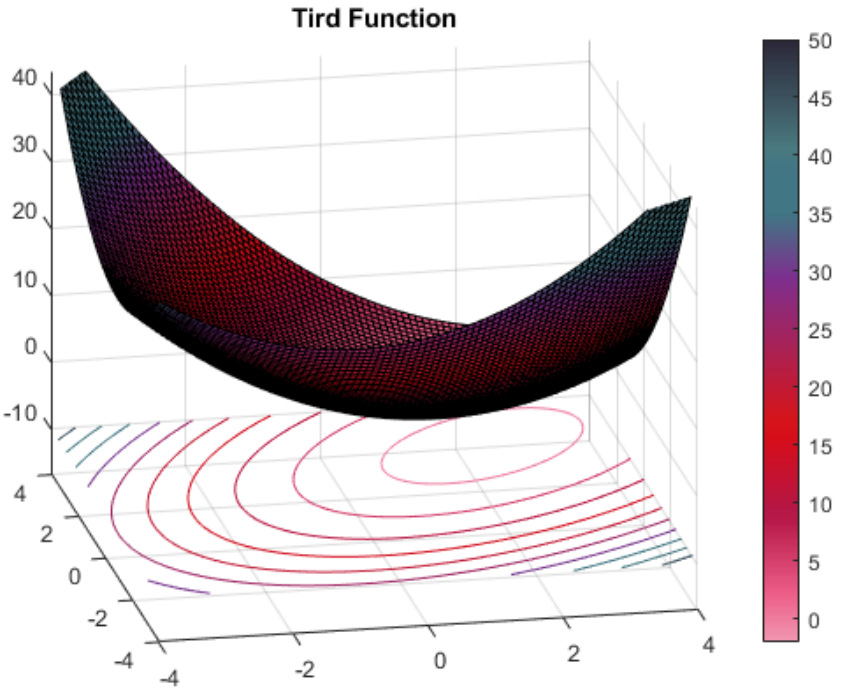

Figura 4.5: Trid function

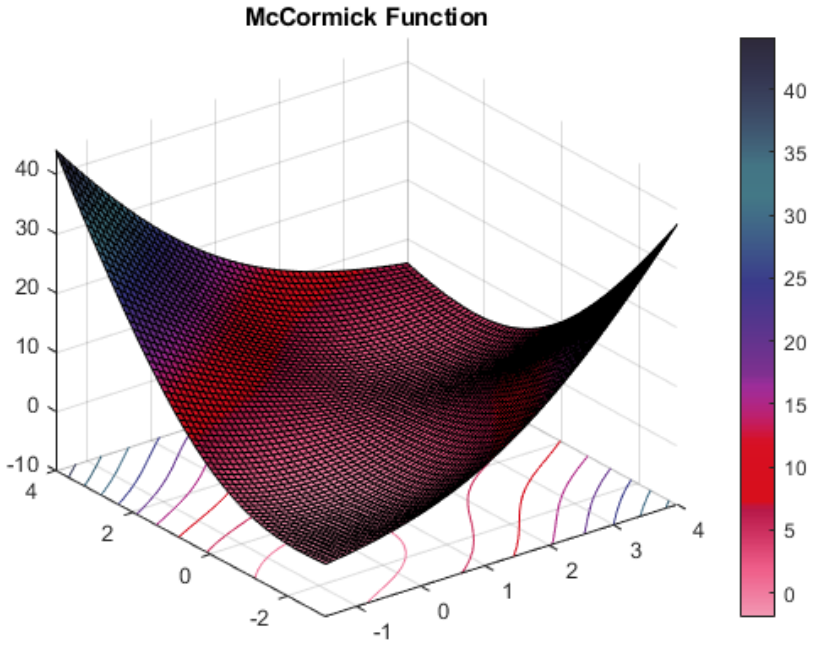

Figura 4.6: McCormick function

5. Resultados y conclusiones

5.1 Análisis por función

Luego de programar los métodos para optimización sin restricciones en el lenguaje MATLAB, con el fin de minimizar las funciones propuestas, analizamos los datos que arrojan. De acuerdo a la metodología antes mencionada, al evaluar en un mismo punto x_0 y una misma tolerancia, compararemos el costo computacional de cada algoritmo por función. A continuación, se presentan los resultados de las funciones **Six Hump Camel, Eason, Cross in Tray, Griewank, Trid y McCormick** en las tablas (5.1), (5.3), (5.4), (5.6), (5.8) y (5.9): respectivamente.

5.1.1 Resultados Six Hump Camel

En primer lugar, la función **Six Hump Camel** (4.1), por sus característica requirió un análisis con la técnica multistart, de este procedimiento se obtuvo la tabla (5.2).

- Máxima Pendiente: De los puntos hallados el 20 % de los puntos alcanzaron el óptimo global, mientras que el porcentaje restante es de los que alcanzaron un valor óptimo con precisión $O(h)^5$
- Condición de Armijo: De los puntos hallados el 16 % de los puntos alcanzaron el óptimo global, mientras que el 84 % alcanzaron óptimos locales con precisión $O(h)^5$
- Newton: De los puntos hallados el 18 % de los puntos alcanzaron el óptimo global, mientras que el 82 % alcanzaron óptimos locales con precisión $O(h)^5$.
- Cuasi-Newton: De los puntos hallados el 15 % de los puntos alcanzaron el óptimo global, mientras que el 38 % restante alcanzaron óptimos locales. También, el 47 % de los puntos restantes no convergen debido a que la matriz es singular.
- Gradientes conjugados: De los puntos hallados el 60 % de los puntos alcanzaron el óptimo global, mientras que el 40 % alcanzaron óptimos locales con precisión $O(h)^5$

La respuesta del multistart, en general, es mejor de lo que se esperaba ya que en la mayoría de los métodos arrojo los cien puntos y todos estos cumples con la segunda condición de optimalidad que afirma que estos puntos son mínimos; sin embargo, hubo métodos que trabajaron mejor con esta función.

A pesar de que el método de gradiente conjugado arrojo el mayor porcentaje de puntos mínimos

globales, en este proyecto se busca minimizar el número de evaluaciones de la función, ya que en la vida real no se cuenta con una como en los ejemplos propuestos en este trabajo, basándonos en esto, el método que respondió mejor fue el método de Newton.

5.1.2 Resultados Eason

En el caso de la función **Eason** (4.2) los resultados nos dicen que las mejores aproximaciones son las que arrojan los métodos de Máxima Pendiente, Newton y Cuasi-Newton; si, además ponderamos el costo computacional, el más rápido y que menos evaluaciones requiere es el de **Cuasi-Newton** que le toma 0.111268 segundos y 6 evaluaciones converger al óptimo global. Esto se debe a que el Máxima Pendiente, necesita más evaluaciones en la función por el cálculo del gradiente si el punto x0 inicial está lejos del óptimo. De igual forma, el método de Newton, que de hecho puede fallar, es muy costoso al calcular la Hessiana y su inversa por cada iteración.

5.1.3 Resultados Cross in Tray

En la tabla (5.4) podemos observar como se comportó la función con los algoritmos antes planteados. La gráfica de está figura (4.3), nos ayuda a darnos una idea en la dificultad.

- Máxima Pendiente: Todos los puntos aproximan al valor mínimo global.
- Condición de Armijo: De los puntos hallados el 20 % de los puntos alcanzaron el óptimo global, el resto de los son aproximaciones a los óptimos.
- Newton: De los puntos hallados, todos los valores son son aproximaciones al óptimo global.
- Cuasi-Newton: De los puntos hallados el 29 % de los puntos alcanzaron el óptimo global, mientras que el 55 % restante alcanzaron óptimos locales, 12 % se salen del rango del dominio y el 4 % de los puntos restantes no convergen debido a que la matriz es singular.
- Gradientes Conjugados: De los puntos hallados el 15 % de los puntos alcanzaron el óptimo global, mientras que el 85 % son aproximaciones del óptimo.

Estas observaciones, son gracias a la técnica multistart que se empleó en este programa, algunos resultados importantes de este proceso se aprecian en la tabla (5.5).

El método que mejor converge a los óptimos globales es Cuasi-Newton, puesto que se hallaron más puntos globales que en los otros métodos; sin embargo, el costo computacional es mayor al tener un promedio de 82 llamadas de la función y tiene menor cantidad de óptimos locales. A pesar de no calcular derivadas, no es suficiente para disminuir el costo computacional. Por lo tanto, a pesar de los buenos resultados de otros algoritmos, el método más eficiente es Máxima pendiente con la condición de Armijo, ya que el costo computacional (llamadas a la función) es menor, y es precisamente lo que buscamos. Esto se debe a que la condición consigue reducciones adecuadas en f a un costo mucho menor. Aunque el algoritmo no tiene el menor tiempo de ejecución, posiblemente el punto de partida se ubicó lejos del óptimo, tardando más en encontrarlo, pero asegurando la propiedad de descenso.

5.1.4 Resultados Griewank

Para hacer la comparación, se realizó una aproximación con un punto inicial. Los resultados fueron escritos en la tabla (5.6). Como la gráfica (4.4) lo muestra, esta función tiene múltiples puntos mínimos por lo que se utilizó la técnica multistart cuyas observaciones se mencionan en la tabla (5.7).

- Máxima Pendiente: De los puntos hallados el 2 % de los puntos alcanzaron el óptimo global, mientras que el resto alcanzaron óptimos locales con precisión $O(h)^5$.
- Condición de Armijo: Tuvo un comportamiento similar al caso de máxima pendiente.
- Newton: De nuevo el 2 % alcanzó al valor mínimo global.
- Cuasi-Newton: De los puntos hallados el 14 % de los puntos alcanzaron el óptimo global, mientras que el 65 % restante alcanzaron óptimos locales, y el 22 % de los puntos restantes

no convergen debido a que la matriz es singular.

- Gradientes Conjugados: De los puntos hallados el 2 % de los puntos alcanzaron el óptimo global, mientras que el 98 % alcanzaron óptimos locales con precisión $O(h)^5$

De nuevo la mejor convergencia está en Cuasi-Newton, pues se hallaron más puntos globales que en los otros métodos; sin embargo, el costo computacional es mayor al tener un promedio de 107 llamadas de la función y tiene menor cantidad de óptimos locales. Por su parte, el método de newton tiene el menor promedio de llamadas de la función y alcanza más puntos óptimos locales pero solo dos globales (2 % de los puntos alcanzaron el óptimo global).

El comportamiento de la función nos indica una dificultad de convergencia para cualquier método en general debido a que cuenta "valles en cazuela "que provoca encontrar distintos puntos óptimos locales. En métodos como en Newton entre más próximo sea el punto inicial al óptimo global va a converger con mayor rapidez.

5.1.5 Resultados Trid

Desde su gráfica, la función **Trid** (4.5), muestra en sus curvas de nivel un valor al que se converge. Luego de realizar los experimentos, presentados en la tabla (5.8), notamos que todos los métodos alcanzan o se aproximan bastante a ese resultado. Sin embargo, si analizamos el costo computacional, es evidente que el algoritmo de **Máxima Pendiente con la Condición de Armijo** arroja los resultados esperados en el menor tiempo y con sólo once iteraciones. Esto se debe a que la función es convexa y el método de descenso funciona muy bien para estas funciones, puesto que el algoritmo asegura la propiedad de descenso dando pasos ortogonales hacia el valor óptimo.[1]

5.1.6 Resultados McCormick

Finalmente, en el caso de la función **McCormick** (4.6) veremos como todos los métodos llegan a el mínimo global, excepto Cuasi-Newton que inclusive se sale del dominio en el que se define la función. El algoritmo más rápido es el de Máxima Pendiente, pero requiere 153 evaluaciones; por otra parte el que menos evaluaciones requiere es Gradientes Conjugados. Al analizar tanto tiempo como número de evaluaciones, podemos decir que el método más eficiente para esta función es **Newton Multivariable** puesto que la relación tiempo-evaluaciones nos indica que es la de menor costo computacional. Esto es gracias a una ventaja del Método Newton, su rápida convergencia al óptimo, siempre y cuando se tenga una buena aproximación inicial. [2]

5.2 Conclusión final

En este proyecto, analizamos diversas funciones de prueba (unimodales y multimodales) que corresponden a problemas de optimización sin restricciones. Aplicamos los métodos correspondientes, además de la técnica multistart con el fin de determinar los puntos mínimos globales, o la mejor aproximación a ellos Con base en la teoría investigada y los resultados obtenidos en los experimentos, concluimos el trabajo realizado haciendo las siguientes observaciones:

- El método de Cauchy presenta una convergencia lineal, pero asegura la propiedad de descenso, si a esto le sumamos la condición de Armijo tenemos un mejor algoritmo que cuenta con un menor costo computacional.
- El método de Newton goza de una tasa de convergencia cuadrática siempre y cuando el punto inicial se encuentre en una vecindad cercana al punto óptimo; si esto no se cumple el método podría fallar y no converger a la solución. Para resolver este problema existen las modificaciones de Cuasi-Newton donde se emula la eficiencia de Newton; sin embargo, aún existen complicaciones como que la matriz Hessiana se haga singular.

[1] El punto x_0 inicial se tomó cercano al óptimo, generando una disminución suficiente en el tamaño de paso.
[2] Nuestro x_0 inicial está muy cerca del punto óptimo, generando una rápida convergencia en el método.

- En un inicio, pensamos que Gradientes conjugados sería el más eficaz; no obstante, si bien llegamos a la solución tenemos una convergencia muy lenta, más que la de Cuasi-Newton y de hecho el tamaño de paso es muy pequeño provocando más iteraciones para encontrar un tamaño de paso aceptable.

Todo lo anterior hace de los métodos presentados en este trabajo herramientas importantes en la resolución de problemas de optimización no lineal; pero, para decidir cuál implementar se necesita un análisis previo de la naturaleza de la función para así tomar la mejor decisión.

Método	x^*	$f(x^*)$	**Tiempo**	**Eval.**
Máx. Pend.	$(-0{,}0897, 0{,}71276)$	-1.0316	0.324932	69
Cond. Armijo	$(-0{,}0898, 0{,}7127)$	-1.0316	4.592014	3
Newton	$(-0{,}0898, 0{,}7127)$	-1.0316	0.140852	104
Cuasi-Newton	$(0{,}0899, -0{,}7126)$	-1.0316	0.547683	527
Grad. Conj.	$(0{,}0890, 0{,}7125)$	-1.0316	1.573032	86

Cuadro 5.1: Resultados de la función Six Hump Camel

Método (Multistart)	**Total de puntos**	**Óptimos Obtenidos**	**Tiempo Prom.**	**Eval. Prom.**
Máx. Pend.	100	(1.6071,0.5687) (- 1.6071,0.5687) (-1.7036, -0.7961)	0.0324932	143
Cond. Armijo	100	(1.6071,0.5687) (- 1.6071,0.5687) (-1.7036, -0.7961)	4.592014	278
Newton	100	(1.6071,0.5687) (- 1.6071,0.5687) (-1.7036, -0.7961) (0,0)	0.140852	413
Cuasi-Newton	53	(1.6071,0.5687) (- 1.6071,0.5687) (-1.7036, -0.7961) (0,0)	0.547683	131
Grad. Conj.	100	(1.6071,0.5687) (- 1.6071,0.5687) (-1.7036, -0.7961) (0,0)	1.573032	210

Cuadro 5.2: Multistart Six Hump Camel

Método	x^*	$f(x^*)$	**Tiempo (s)**	**Eval.**
Máx. Pend.	$(3{,}1416, 3{,}1416)$	-1	0,448239	25
Cond. Armijo	$(3{,}1418, 3{,}1418)$	-1	4,592014	3
Newton	$(3{,}1416, 3{,}1416)$	-1	4,112764	62
Cuasi-Newton	$(3{,}1416, 3{,}1416)$	-1	0,111268	6
Grad. Conj.	$(3{,}1418, 3{,}1418)$	-1	0,048524	105

Cuadro 5.3: Resultados de la función Eason

Método	x^*	$f(x^*)$	**Tiempo**	**Eval.**
Máx. Pend.	$(-1{,}34924, -1{,}34935)$	-2.0626	0.011643	38
Cond. Armijo	$(-1{,}3491, -1{,}3491)$	-2.0626	0.030000	30
Newton	$(-1{,}3492, -1{,}34919)$	-2.0626	0.041411	83
Cuasi-Newton	$(-1{,}3491, -1{,}3491)$	-2.0626	0.007898	82
Grad. Conj.	$(-1{,}3491, -1{,}3491)$	-2.0626	0.058681	323

Cuadro 5.4: Resultados de la función Cross in Tray

Método (Multistart)	**Total de puntos**	**Óptimos obtenidos**	**Tiempo Prom.**	**Eval. Prom**
Máx. Pend.	100	$(-1{,}3492, -1{,}34919)$ $(1{,}34922, -1{,}3494)$ $(1{,}34922, 1{,}3494)$ $(-1{,}3494, 1{,}34928)$	0.011643	65.6
Cond. Armijo	100	$(1{,}3491, -1{,}3491)$ $(-1{,}3491, -1{,}3491)$ $(1{,}3491, 1{,}3491)$ $(-1{,}3491, 1{,}3491)$	0.04422	33
Newton	100	$(-1{,}34924, -1{,}34935)$ $(-1{,}3492, 1{,}34919)$ $(1{,}34923, -134919)$ $(1{,}34923, 1{,}34919)$	0.051298	98.96
Cuasi-Newton	100	$(1{,}3491, -1{,}3491)$ $(-1{,}3491, -1{,}3491)$ $(1{,}3491, 1{,}3491)$ $(-1{,}3491, 1{,}3491)$	0.00780822	87
Grad. Conj.	100	$(1{,}3491, -1{,}3491)$ $(-1{,}3491, -1{,}3491)$ $(1{,}3491, 1{,}3491)$ $(-1{,}3491, 1{,}3491)$	0.0699988	320.8

Cuadro 5.5: Multistart Cross in Tray

Método	x^*	$f(x^*)$	**Tiempo**	**Eval.**
Máx. Pend.	$(0,-0{,}0015)$	5.74e-07	0.010729	25
Cond. Armijo	$(0,0{,}0011)$	2.91e-07	0.005898	65
Newton	$(0,-0{,}0015)$	5.74e-07	0.017530	62
Cuasi-Newton	$(0,1e-07)$	0	0.008443	71
Grad. Conj.	$(0,0{,}0011)$	2.92e-07	0.029152	114

Cuadro 5.6: Resultados de la función Griewank

Método (Multistart)	**Total de puntos**	**Óptimos Obtenidos**	**Tiempo Prom.**	**Eval. Prom.**
Máx. Pend.	100	(-3.1,4.4)	0.010729	107
Cond. Armijo	100	(-6.2,-8.7) (-3.1,-4.5) (0,0.05)	0.005898	90
Newton	100	(0,0) (0,-8.7) (-3.1,4.4)	0.017530	89
Cuasi-Newton	79	(-6.2,-8.9) (0,0) (3.1,4.4) (6.2,8.8)	0.008443	83
Grad. Conj.	100	(3.1,4,4)	0.029152	89

Cuadro 5.7: Multistart en Griewank

Método	x^*	$f(x^*)$	**Tiempo**	**Eval.**
Máx. Pend.	$(2,2)$	-2	0.117920	17
Cond. Armijo	$(2,2)$	-2	0.019105	11
Newton	$(2,2)$	-2	0.065114	41
Cuasi-Newton	$(2,2)$	-2	0.021671	78
Grad. Conj.	$(2,2)$	-2	0.037204	15

Cuadro 5.8: Resultados de la función Trid

Método	x^*	$f(x^*)$	**Tiempo**	**Eval.**
Máx. Pend.	$(-0{,}5469,-1{,}5471)$	-1.9132	0.002971	153
Cond. Armijo	$(-0{,}5469,-1{,}5471)$	-1.9132	0.006203	113
Newton	$(-0{,}5469,-1{,}5471)$	-1.9132	0.733146	62
Cuasi-Newton	$(-9{,}9717,-10{,}9717)$	-11.3380	1.61782	395
Grad. Conj.	$(-0{,}5469,-1{,}5471)$	-1.9132	2.688147	37

Cuadro 5.9: Resultados de la función McCormick

Bibliography

Books

[Ari15] Juan Arias. *Uso de técnicas de optimización global resolver problemas de inversión sísmica: Primer grupo de exploración*. Bucaramanga: Universidad industrial de Santander, abril de 2015, páginas 2-4 (véase página 9).

[B18] Lopez T. B. *Algoritmos*. `http://www.itnuevolaredo.edu.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/IA/Hill\protect\discretionary{\char\hyphenchar\font}{}{}Climbing.pdf`, jun. de 2018, página 1 (véase página 11).

[Sal09] Humberto Salas. *Programación lineal aplicada*. 21.ª edición. Volumen 1. Bogotá: Ecoe Ediciones, Agosto de 2009, páginas 1-30 (véase página 9).

Articles

[Góm84] Miguel Gómez. «Utilización de los algoritmos genéticos para la resolución numérica de los problemas de optimización en mecánica de vuelo». En: (sep. de 1984), páginas 117-119 (véanse páginas 10, 11).

Índice alfabético