Tarea 2:

Juan Gerardo Fuentes Almeida

Abstract—Se crea un programa que realiza una búsqueda en linea a partir de diferentes métodos de cálculo de dirección de descenso y tamaño de paso.

1 Introducción

En esta práctica se pretende encontrar el mínimo global de una función utilizando métodos para calcular la dirección de descenso p_k, tal como el Descenso del Gradiente y el método de Newton; así como la correcta estimación del tamaño de paso α_k que nos permita llegar al óptimo en un tiempo mínimo. Se agregan también los distintos criterios para la valoración de nuestra estimación de α_k, tales como las condiciones de Wolfe y la condición de Goldstein.

La función que minimizamos en esta práctica es la función de Rosenbrock, en la cual el mínimo global se encuentra en un largo y plano valle de forma parabólica (Figura 1):

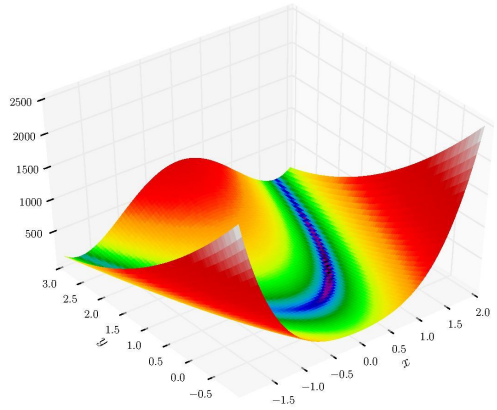

Fig. 1. Funcion Rosenbrock

Trataremos de encontrar el mínimo global de esta función en su forma bidimensional, el cual se conoce que se encuentra en (1,1) con $f(1,1) = 0$:

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

el gradiente de esta función está dado por

$$\nabla f(x_1, x_2) = \begin{pmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}$$

2 Teoría

2.1 Método de Descenso de Gradiente

Es un método de optimización de primer orden, en el cual se determina un mínimo local de una función tomando pasos proporcionales al negativo de su gradiente, y partiendo de que para una $\epsilon > 0$ suficientemente pequeña y una direccion de descenso p_k:

$$f(x_k) + \epsilon \nabla f(x_k)^T p_k < f(x_k)$$

Lo anterior implica que $\nabla f(x)^T p < 0$. El algoritmo de Descenso del Gradiente consiste en tomar $p = -\nabla f(x)$ y recalcular las coordenadas del punto actual utilizando la expresión general $x_{k+1} = x_k + \alpha_k p_k$. Así, en cada iteración nos acercamos al mínimo local de la función siguiendo la dirección de descenso.

2.2 Metodo de Newton

En el método de Newton, la dirección de descenso se calcula a partir de la diferenciación de la aproximación de segundo orden de la serie de Taylor para calcular el mínimo:

$$f(x + p_k) = f(x_k) + \nabla f(x_k)^T p_k + \tfrac{1}{2} p_k^T \nabla^2 f(x_k) p_k$$

$$\Rightarrow \nabla f(x + p_k) = \nabla f(x_k) + \nabla^2 f(x_k) p_k = 0$$

por tanto, la dirección descenso esta dada por:

$$p_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

donde $\nabla^2 f(x_k)$ denota el *Hessiano* de la función $f(x_k)$

$$\nabla^2 f(x_k) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

el cual es una matriz simétrica que para este método debe ser definida positiva.

Computacionalmente, el inverso de esta matriz se obtiene aplicando la factorización de Cholesky para el sistema de la forma **Ax=b** en la expresión:

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

y resolver el sistema de ecuaciones para p_k

La Figura 2 muestra una comparativa entre la dirección que toman los vectores p_k para cada método, el método

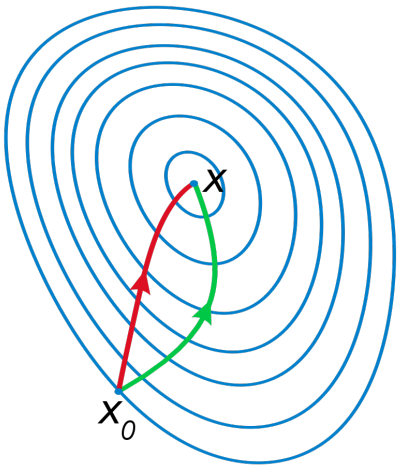

Fig. 2. Comparacion entre el metodo de Descenso del Gradiente (verde) y el metodo de Newton (rojo), el cual toma una ruta mas directa al minimo siguiendo la curvatura de la funcion

de Newton en general llegará al óptimo local en menos iteraciones, ya que utiliza la información del Hessiano para seguir la curvatura de la función mientras desciende.

2.3 Condiciones de Wolfe

2.3.1 *Suficiente descenso*

La condición de suficiente descenso en la función objetivo para α_k esta dada por la siguiente desigualdad:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T p_k$$

para alguna constante $c_1 \in (0, 1)$. En otras palabras, la redución de f debe ser proporcional tanto al tamaño de paso α como a la derivada direccional $\nabla f(x_k)^T p_k$

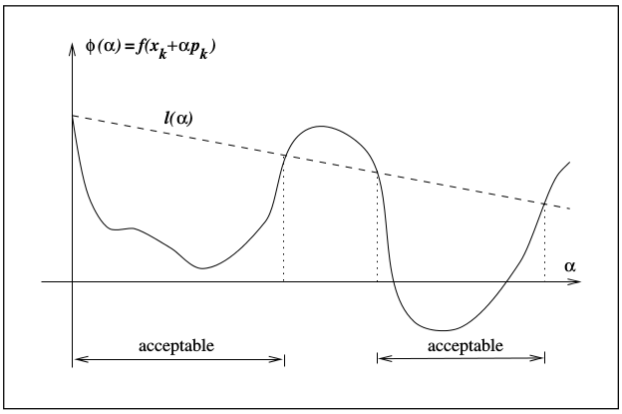

Fig. 3. Condición de Suficiente Descenso. El lado derecho de la desigualdad se denota como $l(\alpha)$.

2.3.2 *Curvatura*

La condición de suficiente descenso por sí sola no es suficiente para garantizar que el algoritmo haga un progreso razonable hacia el mínimo, porque como podemos ver en la Figura 3, esta condición es satisfecha para valores suficientemente pequeños de α. Para descartar pasos inaceptablemente pequeños se introduce un segundo requerimiento, denominado condición de curvatura, el cual requiere que α satisfaga

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$$

para alguna constante $c_2 \in (c_1, 1)$. Al ser el término de la iquierda la derivada $\phi^{'}(\alpha_k)$, se garantiza que la pendiente de ϕ no caiga por debajo de un límite establecido por la constante c_1 y la dirección de descenso.

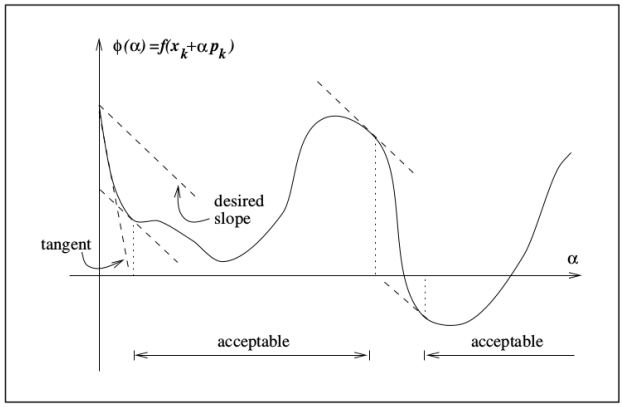

Fig. 4. Condición de curvatura

Las condiciones de suficiente descenso y de curvatura se conocen como condiciones débiles de Wolfe, mientras que las condiciones fuertes de Wolfe están dadas por la condición de suficiente descenso y una condición determinada de la siguiente desigualdad:

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k|$$

Aquí la diferencia radica en que ya no se permite que la derivada $\phi^{'}(\alpha_k)$ sea demasiado positiva, excluyendo puntos que se encuentran lejos de los puntos estacionarios de ϕ.

2.4 Condiciones de Goldstein

Al igual que las condiciones de Wolfe, las condiciones de Goldstein garantizan que el tamano de paso α alcance suficiente descenso pero sin disminuir demasiado su magnitud:

$$f(x_k) + (1 - c)\alpha \nabla f(x_k)^T p_k \leq f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f(x_k)^T p_k$$

con $0 < c < 1/2$. La segunda desigualdad es la condición de suficiente descenso, mientras que la primera desigualdad se introduce para controlar el tamaño de paso (ver Figura 5).

2.5 Estimación del tamaño de paso

2.5.1 *Modelos de Estimación*

Se obtienen a partir de la formulación de un modelo para $\phi_k(\alpha) = f(x_k + \alpha p_k) = f(x_{k+1})$.

Utilizamos el teorema de Taylor para ajustar la función a un modelo cuadrático:

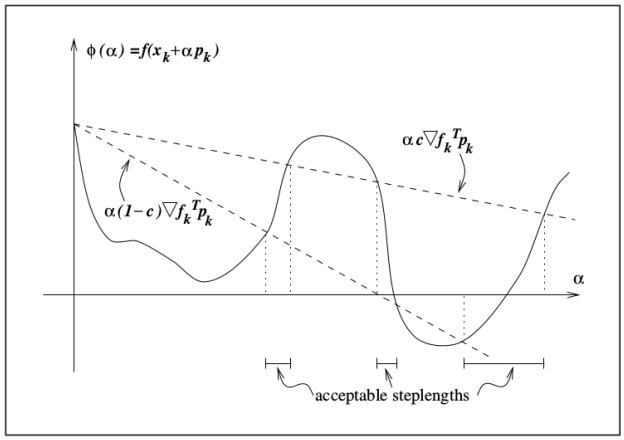

Fig. 5. Condiciones de Goldstein

$$m_k(\alpha) = f(x_k) + \alpha \nabla f(x_k)^T p_k + \tfrac{1}{2}\alpha^2 p_k^T \nabla^2 f(x_k) p_k$$

y se obtiene la siguiente estimación cuadrática para α:
$\alpha = \frac{\alpha_0^2 \phi'(0)}{2(\phi(0) + \alpha_0 \phi'(0) - \phi(\alpha_0))}$

con $\phi(0) = f_k$ y $\phi'(0) = \nabla f(x_k)^T p_k$.

Se puede proponer un modelo cúbico agregando el término correspondiente al modelo ya establecido:

$$m_3(\alpha) = a + b\alpha + c\alpha^2 + d\alpha^3 = \phi(\alpha)$$

donde:
$m_3(0) = a \Rightarrow \phi(0) = a$
$m_3'(0) = b \Rightarrow \phi'(0) = b$
$m_3"(0) = 2c \Rightarrow c = \frac{1}{2}\phi"(0)$

y para d:
$m_3(\alpha_0) = a + b\alpha_0 + c\alpha_0^2 + d\alpha_0^3 = \phi(\alpha_0)$

$\Rightarrow d = \frac{1}{\alpha_0^3}[\phi(\alpha_0) - \phi(0) - \alpha_0 \phi 0(0) - \frac{1}{2}\alpha_0^2 phi"(0)]$

y definimos $\alpha_3 = argmin_\alpha m_3(\alpha) \Rightarrow \frac{\partial}{\partial \alpha} m_3(\alpha) = 0$

$\Rightarrow b + c\alpha_3 + 3d\alpha_3^2 = 0$

Utilizando la formula general para calcular las raices de esta ecuación:

$$\alpha_3 = \frac{-c \pm \sqrt{c^2 - 3bd}}{3d}$$

2.5.2 Algoritmo de Backtracking con Inercia

Otra forma de estimar α es con un algoritmo que dado un valor inicial, realice una actualización hasta que alguna de las condiciones mencionadas anteriormente se cumpla. Por ejemplo, en el Algoritmo 1 se muestra el método de Newton utilizando este método de estimación.

Algorithm 1 Backtracking con Inercia

Require: Punto inicial x, α_0 $maxIter$, ϵ,$k \leftarrow 0$
  while $k < maxIter$ **and** $error > \epsilon$ **do**
    $G \leftarrow \nabla f_k$
    $H \leftarrow \nabla^2 f(x_k)$
    $p_k \leftarrow -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$
    $\alpha_k \leftarrow \alpha_2$ (modelo cuadratico)
    $tracking \leftarrow 0$,$count \leftarrow 0$,$maxcount$
    while !Condition(Wolfe) **and** $count < maxcount$ **do**
      $tracking \leftarrow tracking + 1$
      if $tracking > 2$ **then**
        $\alpha_k \leftarrow 3\alpha_k$
        $tracking \leftarrow 0$
      else
        $\alpha_k \leftarrow \alpha_k / 2$
      $count \leftarrow count + 1$
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    $error \leftarrow |x_{k+1} - x_k|$
    $x_{k+1} \leftarrow x_k$
    $k \leftarrow k + 1$
  return x

3 RESULTADOS

Como parte de la práctica, se generaron 30 resultados para cada combinación de los métodos descritos. A continuación se muestra una tabla comparativa entre cada método; se documenta el número de iteraciones a las que convergió el algoritmo, el tiempo de ejecución y la función evaluada en el punto mínimo obtenido. El algoritmo inicia con un valor aleatorio para x_1 y x_2 entre -5 y 15, y un α inicial estático de 0.2:

Metodo de Descenso de Gradiente					
Condicion	α	Exitos	Iteraciones	Tiempo	f_{min}
Wolfe D	cuad	25	16489	17.54042	0
Wolfe D	cub	29	5308	6.9045862	0
Wolfe D	BT	20	16773	111.0014	0
Wolfe F	cuad	24	13895	14.9569	0
Wolfe F	cub	30	5981	7.46247	0
Wolfe F	BT	21	18560	124.7305	0
Goldstein	cuad	22	1077	1.697818	0
Goldstein	cub	30	1889	2.108	0
Goldstein	BT	23	1064	1.2829	0

La columna *Éxitos* indica cuántas de las 30 ejecuciones resultaron en el cálculo exitoso del mínimo, en los casos no exitosos se observó que el algoritmo no pudo obtener el mínimo en el número de iteraciones máximas dado ($1e6$), o que el valor de α se volvía nulo y por consiguiente se originaba una falsa convergencia del algoritmo. El tiempo en ambas tablas esta dado en milisegundos.

Metodo de Newton					
Condicion	α	Exitos	Iteraciones	Tiempo	f_{min}
Wolfe D	cuad	30	12	0.0157	0
Wolfe D	cub	30	38	0.0573	0
Wolfe D	BT	30	13	0.0613	0
Wolfe F	cuad	30	12	0.0159	0
Wolfe F	cub	30	38	0.0589	0
Wolfe F	BT	30	12	0.0612	0
Goldstein	cuad	30	133	0.1108	0
Goldstein	cub	30	138	0.1192	0
Goldstein	BT	30	151	0.1287	0

4 CONCLUSIONES

En general se observa que el método de Descenso de Gradiente es mucho menos eficiente; aunque ya se tenía previsto que el método de Newton estaba formulado para llegar al punto mínimo mas rápido, no se había estimado la magnitud de la diferencia en tiempos e iteraciones entre estos dos métodos, también se observa que no siempre el método del Gradiente es capaz de llegar a un resultado, en cambio el método de Newton siempre logra calcular el mínimo con exito.

En cuanto a la tabla que muestra el desempeño del método de Newton, se observa que trabaja mejor con una de las condiciones de Wolfe, aunque bien la diferencia en número de iteraciones puede deberse a la elección de la constante c para la condición de Goldstein. Por otra parte, el método funciona de manera excelente cuando se combina también con el modelo cuadrático para estimar α, o bien con el algoritmo de backtracking con inercia, sin olvidar que en esta implementación en específico esta combinado con el modelo cuadrático.

REFERENCES

[1] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006.

APPENDIX

El programa está implementado tomando en cuenta todas estandarizaciones indicadas en el curso.

Un *makefile* ha sido generado, el cual soporta los comandos *make*, *run* and *clean*. Para el tercer programa se incluye un script para correr las 30 ejecuciones de cada método, los resultados se guardan en un archivo diferente para cada combinación, los argumentos del programa tienen el siguiente formato (todos son requeridos):

$arg1$ = 0(Descenso de Gradiente), 1(Newton)
$arg2$ = 0(Wolfe Debil), 1(Wolfe Fuerte), 2(Goldstein)
$arg3$ = 0(cuadratico), 1(cubico), 2(backtracking)