

Centro de Investigación en Matemáticas, A.C.

Curso de Optimización I

Impartido por:

Dr. Mariano Rivera Meraz



Un agradecimiento para

Yolanda de Lira Domínguez y Rocky Bizuet García
por su colaboración en la elaboración de este material.

26 de enero de 2005

Índice general

1. Introducción.	6
1.1. Formulación Matemática.	7
1.2. Ejemplo: Un problema de transporte.	8
1.3. Tipos de problemas de Optimización.	9
1.4. Algoritmos de optimización.	11
1.5. Convexidad.	11
2. Fundamentos de Optimización sin restricciones.	13
2.1. Qué es una solución?	14
2.2. Algoritmos (una visión preliminar).	20
2.2.1. Búsqueda en línea.	20
2.2.2. Métodos de región verdadera.	21
3. Métodos de Búsqueda en Línea.	26
3.1. Tamaño de paso.	27
3.2. Algoritmos para selección del tamaño de paso.	32
4. Métodos de Región Verdadera.	36
4.1. Punto de Cauchy.	39
5. Métodos de Gradiente Conjugado.	44
5.1. Método de Gradiente Conjugado lineal.	45

5.2. Gradiente Conjugado No Lineal.	54
6. Introducción al Cálculo Variacional.	58
6.1. Problema sin restricciones.	58
7. Métodos de Newton Prácticos.	62
7.1. Newton con pasos inexactos.	63
7.2. Métodos de Newton con búsqueda en línea.	64
7.3. Técnicas de Modificación del Hessiano.	67
7.4. Métodos de Newton de Región Válida.	71
8. Cálculo Numérico de Derivadas.	73
8.1. Aproximación por diferencia finita.	74
9. Métodos Quasi - Newton.	78
9.1. El método BFGS.	78
10. Mínimos Cuadrados No Lineales.	83

Índice de figuras

1.1.	Formulación matemática. Ejemplo.	8
1.2.	Problema de transporte.	9
1.3.	Función convexa.	10
1.4.	Conjunto convexo.	11
2.1.	Problema de mínimos cuadrados.	14
2.2.	Ejemplos de mínimos.	15
2.3.	Teorema del Valor Medio.	16
2.4.	Teorema Fundamental del Cálculo.	16
2.5.	Demostración del Teorema de Taylor.	17
2.6.	Búsqueda en línea.	20
2.7.	Métodos de Región Verdadera.	21
2.8.	Dirección de descenso acelerado para una función de dos variables.	22
2.9.	Direcciones de descenso.	22
3.1.	Mínimización de Φ .	27
3.2.	Tamaños de paso.	28
3.3.	Condición de curvatura.	28
3.4.	Condiciones “Fuertes” de Wolf.	29
3.5.	Condiciones de Goldstein.	30
3.6.	Métodos de descenso.	31

3.7.	Máximo descenso.	32
3.8.	Aproximación de segunda derivada por diferencias finitas.	33
4.1.	Métodos de Región Válida.	37
4.2.	Punto de Cauchy.	40
4.3.	Punto de Cauchy y solución de Newton.	41
4.4.	Método de Dogleg.	41
5.1.	Actualización de \mathbf{x} en método de descenso.	45
5.2.	Direcciones principales.	46
5.3.	Direcciones conjugadas.	46
5.4.	Propiedades de direcciones conjugadas.	47
5.5.	Convergencia de GC.	53
7.1.	Paso de dogleg.	72
9.1.	Construcción de nuevo modelo.	79

OPTIMIZACIÓN I

Temario tentativo

1. Fundamentos de Optimización
 - 1.1. Fundamentos de Optimización sin restricciones
 - 1.2. Métodos de búsqueda en línea
 - 1.3. Métodos de región verdadera
2. Métodos de Optimización
 - 2.1. Métodos de Gradiente Conjugado
 - 2.2. Métodos de Newton
 - 2.3. Cálculo de Derivadas
 - 2.4. Métodos Quasi-Newton
 - 2.5. Métodos de gran escala
3. Problemas de Optimización no lineales
 - 3.1. Mínimos cuadrados no lineales
 - 3.2. Ecuaciones no lineales
 - 3.3. Optimización con Restricciones
4. Programación Lineal
 - 4.1. Método Simplex
 - 4.2. Métodos de Punto interior
5. Optimización con restricciones no lineales
 - 5.1. Fundamentos
 - 5.2. Programación cuadrática
 - 5.3. Con restricciones
6. Introducción a la Optimización combinatoria
 - 6.1. Búsqueda local (Métodos "greedy")
 - 6.2. Métodos estocásticos

Texto:

Numerical Optimization. Jorge Nocedal & Stephen J. Wright

Capítulo 1

Introducción.

La optimización es una tarea diaria y común. La gente optimiza:

- Compañías de aerolíneas optimizan las asignaciones (calendarios) de las tripulaciones para reducir costos.
- En inversiones se crean portafolios que reducen riesgos y maximizan las ganancias
- Las manufactureras maximizan la eficiencia y tratan de minimizar desperdicios y tiempo.

La naturaleza también optimiza:

- Sistemas físicos tienden a estados de menor energía.
- Reacciones químicas minizan energía potencial.
- La luz sigue trayectorias que consumen menor tiempo.

La optimización es importante en la toma de decisiones y en el análisis de sistemas físicos. Para optimizar requerimos de definir una medida de desempeño, para ello debemos definir o identificar una función objetivo.

Por ejemplo:

- Ganancia,
- Tiempo,
- Energía potencial,
- Espacio, etc.,

las cuales son cantidades representadas por un número.

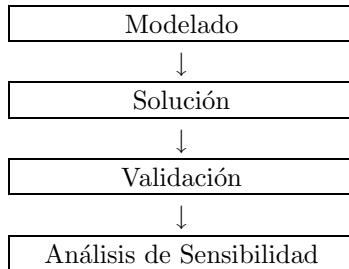
El objetivo depende de características del sistema denominadas variables o incógnitas. La meta es encontrar los valores de las variables que optimicen el sistema.

Es común encontrar que las variables tienen restricciones, por ejemplo el tiempo que requiere realizar una cierta actividad no puede ser negativo (igualmente las distancias de viaje).

Modelado

Modelado es el proceso de identificar el objetivo, variables y las restricciones para un cierto problema.

Los pasos del proceso de modelado son: modelado, en donde se identifican objetivos, variables y restricciones para un problema; solución, en donde se hace uso de álgebra de optimización; validación, valida la solución encontrada en el paso anterior, y finalmente análisis de sensibilidad, en donde se hace un análisis del modelo y variables.



1.1. Formulación Matemática.

Optimizar es minimizar o maximizar una función sujeta a restricciones en sus variables.

Notación:

\mathbf{x} : Vector de variables (incógnitas o parámetros).

f : Función objetivo (función de \mathbf{x}) a maximizar o minimizar.

c : Vector de restricciones en las variables. Función vectorial de \mathbf{x} . Cada componente de c es una restricción en \mathbf{x}

Esto puede ser escrito:

$$\min_{\mathbf{x} \in \mathcal{R}} f(\mathbf{x}) \text{ Sujeto a (S.A.) } \begin{cases} c_i(\mathbf{x}) = 0 & i \in E \\ c_i(\mathbf{x}) \geq 0 & i \in I, \end{cases} \quad (1.1)$$

donde tanto f como las c' s son funciones escalares de la variable \mathbf{x} (c' s y $f : \mathbb{R}^n \rightarrow \mathbb{R}$), E e I son los conjuntos de restricciones de igualdad y desigualdad respectivamente.

Por ejemplo:

$$\min (x_1 - 2)^2 + (x_2 - 1)^2 \text{ S.A. } \begin{cases} x_1^2 - x_2 \leq 0 \\ x_1 + x_2 \leq 0, \end{cases}$$

lo podemos escribir en la forma (1.1):

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

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad c(\mathbf{x}) = \begin{bmatrix} c_1(\mathbf{x}) \\ c_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} -x_1^2 + x_2 \\ -x_1 - x_2 + 2 \end{bmatrix}.$$

En las restricciones note los cambios de signo, y además que $I = \{1, 2\}$ y $E = \emptyset$

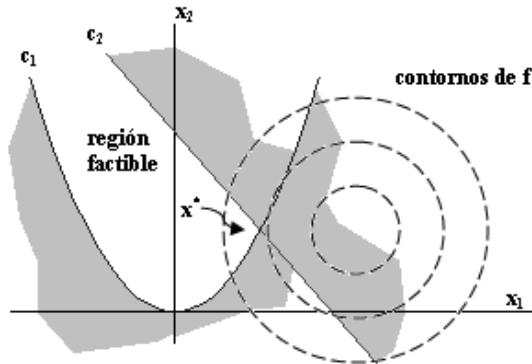


Figura 1.1: Formulación matemática. Ejemplo.

1.2. Ejemplo: Un problema de transporte.

Una compañía química tiene dos plantas: F_1 y F_2 y una docena de puntos de distribución R_1, \dots, R_{12} . Cada planta produce a_i toneladas de cierto producto químico cada semana, donde a_i es la capacidad de la planta i . Cada punto de distribución demanda b_j toneladas del producto. El costo de enviar una tonelada del producto de la planta F_i al punto R_j es c_{ij} , para $i = 1, 2$ y $j = 1, \dots, 12$.

Problema: Determinar cuánto producto debe ser enviado de cada planta a cada punto de distribución de tal forma que se satisfagan los siguientes requerimientos al costo mínimo:

Cantidad de producto de F_i a R_j es x_{ij} donde x_{ij} son las variables del sistema, para $i = 1, 2$ y $j = 1, 2, \dots, 12$.

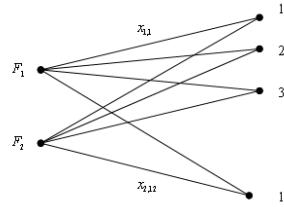


Figura 1.2: Problema de transporte.

Esto puede ser escrito como:

$$\min \sum_{ij} c_{ij} x_{ij}, \text{ para } i = 1, 2 \text{ y } j = 1, 2, \dots, 12,$$

S.A. (sujeto a)

$$\begin{aligned} \sum_j x_{ij} &\leq a_i, && \text{cada planta no puede enviar más de lo que produce,} \\ \sum_i x_{ij} &\geq b_j, && \text{la demanda debe ser satisfecha,} \\ x_{ij} &\geq 0, && \text{ya que son envíos.} \end{aligned}$$

Este es un problema de programación lineal, ya que la función objetivo y las restricciones son lineales.

1.3. Tipos de problemas de Optimización.

$$\text{Optimización} \left\{ \begin{array}{ll} \text{Continua} & x_{ij} \in \mathbb{R} \\ \text{Discreta} & x_{ij} \in Z(\text{restricción adicional.}) \end{array} \right.$$

Si en el ejemplo anterior se producen carros, x_{ij} tiene que ser un número entero, por lo cual se tendría un problema de optimización discreta.

La Optimización continua es generalmente más fácil de resolver dada la suavidad de la función objetivo y de las restricciones.

Es posible encontrar problemas donde:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ tal que } \begin{array}{ll} x_i \in \mathbb{R} & i \in \Omega \\ x_i \in Z & i \in \overline{\Omega} \end{array}$$

Al final veremos estrategias de optimización discreta.

$$\text{Optimización} \left\{ \begin{array}{l} \text{Con restricciones.} \\ \text{Sin restricciones.} \end{array} \right.$$

- **Con restricciones:** restricciones explícitas en las variables.

- cotas: $0 \leq x_i \leq 1$
- restricciones lineales: $\sum_i x_i \leq 1$ ó $= 1$
- restricciones no lineales: $\sum_i x_i^2 - \cos(x_3) < a$

- **Sin restricciones:** a veces son reformulaciones de restricciones que se incorporan como penalizaciones.

$$\text{Optimización} \left\{ \begin{array}{l} \text{Global.} \left\{ \begin{array}{l} \text{Convexa.} \\ \text{No convexa (difícil).} \end{array} \right. \\ \text{Local - no convexa, mínimo o máximo local.} \end{array} \right.$$

Nota: Una función convexa es aquella en la cual si se unen con una línea cualquiera dos puntos de la función, esta línea no cruza la función. Hay funciones estrictamente convexas y convexas límite. En la siguiente figura se muestra un ejemplo de éstas.

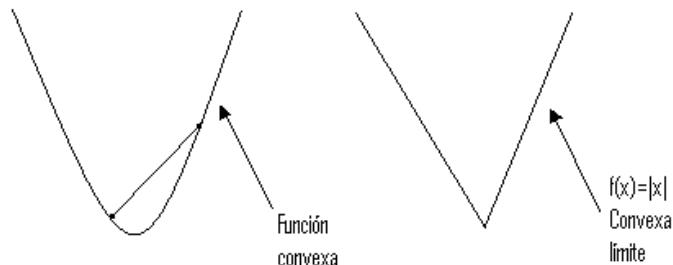


Figura 1.3: Función convexa.

$$\text{Optimización} \left\{ \begin{array}{l} \text{Estocástica.} \\ \text{Determinista.} \end{array} \right.$$

- **Estocástica:** No se conoce con precisión el modelo, sólo posibles escenarios (modelos probabilísticos). Algunas veces es posible predecir escenarios con grados de confianza.
- **Determinista:** Modelo perfectamente conocido (fijo).

1.4. Algoritmos de optimización.

Son algoritmos iterativos:

- parten de una condición inicial y
- generan una secuencia de estimaciones que mejoran la solución.

Características deseables:

- **Robustez:** $\begin{cases} \text{Aplicables a problemas parecidos.} \\ \text{Independencia de la condición inicial.} \end{cases}$
- **Eficiencia:** $\begin{cases} \text{Tiempo de cómputo.} \\ \text{Memoria requerida.} \end{cases}$
- **Precisión:** reducida sensibilidad a errores numéricos.
- existe un compromiso entre las características anteriores.

1.5. Convexidad.

- Aplicable a conjuntos y funciones.

$S \subset \Re^n$ es un conjunto convexo si el segmento de línea que une a cualesquiera dos elementos $s_1, s_2 \in S$ está contenido enteramente en S , esto es, dados dos puntos $x, y \in S$ se tiene: $ax + (1 - a)y \subset S \forall a \in [0, 1]$. (ver figura 1.4).

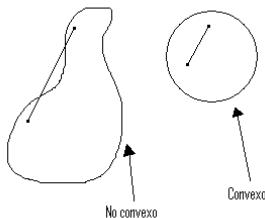


Figura 1.4: Conjunto convexo.

Una función f es convexa si su dominio es un conjunto convexo y para cualesquiera dos puntos x, y de su dominio la gráfica de f cae por debajo de la línea en \mathbb{R}^{n+1} que une a $(x, f(x))$ y $y, f(y)$, esto es:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad \forall \alpha \in [0, 1]$$

Capítulo 2

Fundamentos de Optimización sin restricciones.

Formulación:

$$\begin{aligned} & \min_x f(x) \\ & x \in \mathbb{R}^n, \quad n \geq 1, \end{aligned}$$

esto es: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (es una función suave).

Ejemplo: dadas las medidas $\mathbf{y} = [y_1, y_2, \dots, y_m]^T$ tomadas en tiempos $\mathbf{t} = [t_1, t_2, \dots, t_m]^T$ se desean encontrar los parámetros $\mathbf{x} = [x_1, x_2, \dots, x_6]^T$ del modelo:

$$\phi(t; \mathbf{x}) = x_1 + x_2 e^{-(x_3 - t)^2 / x_4} + x_5 \cos(x_6 t).$$

El error entre el modelo (dados un vector de parámetros x) es dado por:

$$r_j(x) = y_j - \phi(t_j; \mathbf{x}), \quad j = 1, \dots, m,$$

este vector \mathbf{r} es denominado *vector residual* ó simplemente *residual*.

Una estimación de \mathbf{x} puede ser obtenida mediante la solución del problema

$$\min_{\mathbf{x} \in \mathbb{R}^6} f(\mathbf{x}) = r_1^2(\mathbf{x}) + \dots + r_6^2(\mathbf{x}) = \sum_{j=1}^m r_j^2(\mathbf{x}).$$

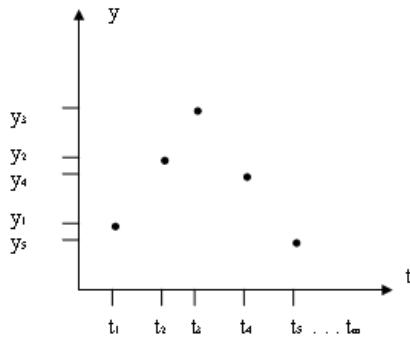


Figura 2.1: Problema de mínimos cuadrados.

Este problema es conocido como problema de mínimos cuadrados no lineales.

Generalmente el modelo propuesto no puede representar exactamente a los datos, es decir: $f(\mathbf{x}^*) \neq 0$, luego, es necesario definir que es una solución \mathbf{x} .

2.1. Qué es una solución?

Mínimo global.

\mathbf{x}^* es el mínimo global de una función f si $f(\mathbf{x})^* \leq f(\mathbf{x}^*) \forall \mathbf{x}$.

Este mínimo global es en general muy “difícil” de encontrar dado que no esperamos realizar una búsqueda exhaustiva.

Mínimo local (o minimizador local débil).

\mathbf{x}^* es un mínimo local de una función f si existe una vecindad $N_{\mathbf{x}^*}$ de \mathbf{x}^* tal que $f(\mathbf{x})^* \leq f(\mathbf{x}^*) \forall \mathbf{x} \in N_{\mathbf{x}^*}$.

Una vecindad $N_{\mathbf{x}^*}$ es un conjunto abierto que contiene a \mathbf{x}^* .

Mínimo local estricto (o minimizador local estricto).

\mathbf{x}^* es un mínimo local estricto de una función f si existe una vecindad $N_{\mathbf{x}^*}$ de \mathbf{x}^* tal que $f(\mathbf{x})^* < f(\mathbf{x}^*) \forall \mathbf{x} \in N_{\mathbf{x}^*}$ con $\mathbf{x} \neq \mathbf{x}^*$.

Para identificar un mínimo local es necesario examinar toda una vecindad alrededor de \mathbf{x} (todos los puntos). Pero si la función es “suave”, es más eficiente examinar el Gradiente $\nabla f(\mathbf{x}^*)$ y el Hessiano $\nabla^2 f(\mathbf{x}^*)$.

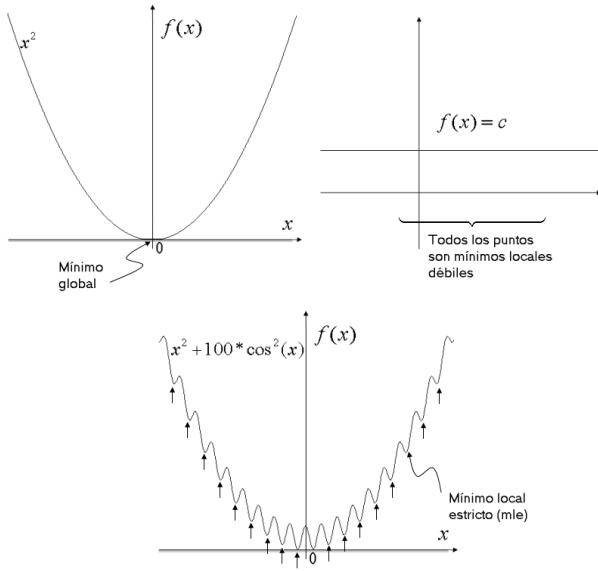


Figura 2.2: Ejemplos de mínimos.

Definición de Gradiente y Hessiano.

Sea $\mathbf{x} = [x_1, \dots, x_n]^T$ y $f(\mathbf{x})$ una función vectorial $f : \mathbb{R}^n \rightarrow \mathbb{R}$. El Gradiente $\nabla f(\mathbf{x})$ se define por:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}.$$

La segunda derivada de f conocida como Hessiano se define por:

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

- Una función f es diferenciable si todas sus derivadas parciales existen.
- Es continuamente diferenciable si todas las segundas derivadas parciales son funciones continuas de \mathbf{x} .
- f es dos veces diferenciable si todas las segundas derivadas parciales existen y es dos veces continuamente diferenciable.

- f es dos veces continuamente diferenciable, si el Hessiano ($\nabla^2 f$) es una matriz simétrica, es decir, si: $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} \forall i, j = 1, 2, \dots, n$.

Teorema del Valor Medio.

Si f es continuamente diferenciable, entonces

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + t\mathbf{p})^T \mathbf{p} \text{ para alguna } t \in (0, 1) \text{ y } \mathbf{p} \in \mathbb{R}^n$$

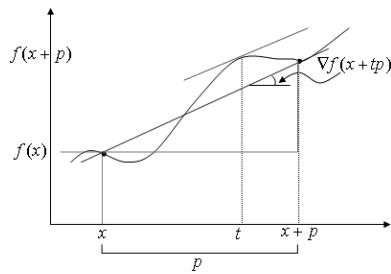


Figura 2.3: Teorema del Valor Medio.

Teorema Fundamental del Cálculo o Teorema de Newton.

Sea f una función $f : \mathbb{R} \rightarrow \mathbb{R}$ diferenciable, entonces:

$$f(1) = f(0) + \int_0^1 f'(t) dt$$

Ejemplo: Sea g una función $g : \mathbb{R} \rightarrow \mathbb{R}$ diferenciable, entonces: $g(1) = g(0) + \int_0^1 g'(t) dt$

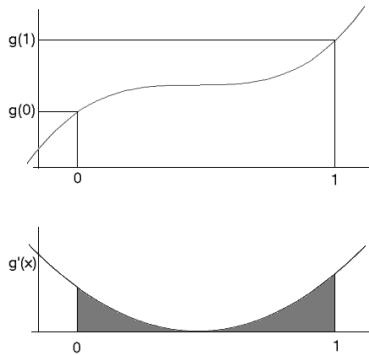


Figura 2.4: Teorema Fundamental del Cálculo.

Teorema 2.1. Teorema de Taylor.

Suponga $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciable, y $\mathbf{p} \in \mathbb{R}^n$, entonces tenemos, por el Teorema del Valor Medio:

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + t\mathbf{p})^T \mathbf{p} \text{ para alguna } t \in (0, 1).$$

Más aún, si es dos veces continuamente diferenciable, tenemos que:

$$\nabla f(\mathbf{x} + \mathbf{p}) = \nabla f(\mathbf{x}) + \int_0^1 \nabla^2 f(\mathbf{x} + t\mathbf{p}) \mathbf{p} dt$$

y que:

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{x} + t\mathbf{p}) \mathbf{p} \text{ para alguna } t \in (0, 1)$$

Prueba:

Por el Teorema de Newton: $g(1) = g(0) + \int_0^1 g'(x) dx$, entonces:

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \int_0^1 \nabla f(\mathbf{x} + \xi \mathbf{p}) \mathbf{p} d\xi. \quad \dots(1)$$

Ahora, por el teorema del valor medio:

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + t\mathbf{p})^T \mathbf{p} \text{ para una } t \in (0, 1). \quad \dots(2)$$

Luego de (1) y (2):

$$\nabla f(\mathbf{x} + t\mathbf{p})^T \mathbf{p} = \int_0^1 \nabla f(\mathbf{x} + \xi \mathbf{p}) \mathbf{p} d\xi \text{ para una } t \in (0, 1).$$

Ahora:

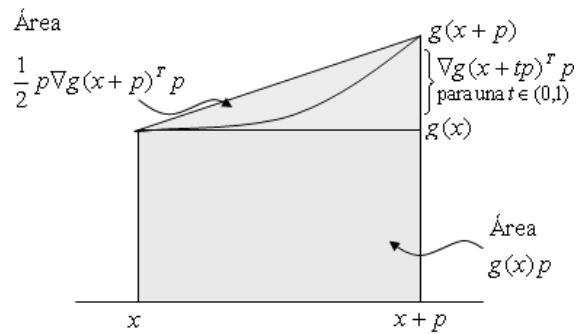


Figura 2.5: Demostración del Teorema de Taylor.

Entonces:

$$\int_0^1 \nabla g(\mathbf{x} + \xi \mathbf{p}) \mathbf{p} d\xi = g(\mathbf{x}) \mathbf{p} + \frac{1}{2} \mathbf{p} g'(\mathbf{x} + t \mathbf{p}) \mathbf{p} \text{ para una } t \in (0, 1).$$

Haciendo $g(\mathbf{x}) = f'(\mathbf{x})$ tenemos:

$$\int_0^1 \nabla f(\mathbf{x} + \xi \mathbf{p}) \mathbf{p} d\xi = f'(\mathbf{x}) \mathbf{p} + \frac{1}{2} \mathbf{p} f''(\mathbf{x} + t \mathbf{p}) \mathbf{p} \text{ para una } t \in (0, 1). \quad \dots(3)$$

Finalmente de (2) y (3) tenemos:

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{x} + t \mathbf{p}) \mathbf{p} \text{ para una } t \in (0, 1). \diamond$$

Las *condiciones necesarias* de optimalidad se derivan asumiendo que \mathbf{x}^* es un minimizador local, y luego probando los hechos para $\nabla f(\mathbf{x})$ y $\nabla^2 f(\mathbf{x})$.

Teorema 2.2. Teorema de primer orden.

Si \mathbf{x}^* es un minimizador local y f es continuamente diferenciable en una vecindad abierta $N_{\mathbf{x}^*}$, entonces: $\nabla f(\mathbf{x}^*) = 0$.

Prueba:

Por contradicción asumimos $f(\mathbf{x}^*) \neq 0$. Ahora elegimos un vector \mathbf{p} tal que $\mathbf{p}^T \nabla f(\mathbf{x}^*) < 0$, por ejemplo $p = -\nabla f(\mathbf{x}^*)$.

Ahora, por continuidad, ∇f es continua en una vecindad $N_{\mathbf{x}^*}$, luego, hay un escalar $T > 0$ tal que $\mathbf{p}^T \nabla f(\mathbf{x}^* + t \mathbf{p}) < 0$ para $t \in [0, T]$. Sea $\bar{t} \in [0, T]$, luego por el teorema de Taylor:

$$f(\mathbf{x}^* + \bar{t} \mathbf{p}) = f(\mathbf{x}^*) + \bar{t} \mathbf{p}^T \nabla f(\mathbf{x}^* + t \mathbf{p}),$$

para alguna $t \in (0, \bar{t})$, pero $\bar{t} \mathbf{p}^T \nabla f(\mathbf{x}^* + t \mathbf{p}) < 0$. De aqui vemos que $f(\mathbf{x}^* + \bar{t} \mathbf{p}) < f(\mathbf{x}^*)$, es decir, hemos encontrado una dirección $(-\nabla f(\mathbf{x}^*))$ a lo largo de la cual $f(\mathbf{x})$ decrece.

Por lo tanto \mathbf{x}^* no es un mínimo local. \diamond

Teorema 2.3. Condición necesaria de segundo orden.

Si \mathbf{x}^* es un minimizador local de f y $\nabla^2 f(x)$ es continuo en una vecindad $N_{\mathbf{x}^*}$, entonces $\nabla f(\mathbf{x}^*) = 0$ y $\nabla^2 f(\mathbf{x}^*)$ es positiva semidefinida.

Nota:

B es una matriz definida positiva si $\mathbf{p}^T B \mathbf{p} > 0 \forall \mathbf{p} \neq 0$.

B es semidefinida positiva si $\mathbf{p}^T B \mathbf{p} \geq 0 \forall \mathbf{p}$.

Prueba:

Sabemos por el Teorema 2.2. que $\nabla f(\mathbf{x}^*) = 0$. Por contradicción asumimos que $\nabla^2 f(\mathbf{x}^*)$ no es positiva semidefinida. Luego podemos elegir un vector \mathbf{p} tal que $\mathbf{p}^T \nabla^2 f(\mathbf{x}^*) \mathbf{p} < 0$, y por continuidad, hay un escalar $T > 0$ tal que:

$$\mathbf{p}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{p}) \mathbf{p} < 0 \quad \forall t \in [0, T].$$

Por series de Taylor alrededor de \mathbf{x}^* , tenemos para toda $\bar{t} \in (0, T]$ y para alguna $t \in (0, \bar{t})$:

$$f(\mathbf{x}^* + \bar{t}\mathbf{p}) = f(\mathbf{x}^*) + \bar{t}\mathbf{p}^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\bar{t}^2 \mathbf{p}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{p}) \mathbf{p} < f(\mathbf{x}^*).$$

para alguna $t \in (0, \bar{t})$. Como en el caso anterior, encontramos una dirección a partir de \mathbf{x}^* a lo largo de la cual f decrece. Por lo tanto \mathbf{x}^* no es un minimizador. \diamond

Nota: Serie de Taylor: $f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2$.

Teorema 2.4. Condiciones suficientes de segundo orden.

Suponga $\nabla^2 f$ continua en un conjunto abierto alrededor de \mathbf{x}^* y que $\nabla f(\mathbf{x}^*) = 0$ y $\nabla^2 f(\mathbf{x}^*)$ es positiva definida. Entonces \mathbf{x}^* es un minimizador local estricto de f .

Prueba:

Dado que el Hessiano es continuo y definido en \mathbf{x}^* , podemos elegir un radio $r > 0$ tal que $\nabla^2 f(\mathbf{x}^*)$ se mantenga definido positivo para toda \mathbf{x} en la bola abierta $D = \{\mathbf{z} \mid \|\mathbf{z} - \mathbf{x}^*\| < r\}$. Tomando algún vector $\mathbf{p} \neq 0$ con $\|\mathbf{p}\| < r$, tenemos $\mathbf{x}^* + \mathbf{p} \in D$ y también:

$$f(\mathbf{x}^* + \mathbf{p}) = f(\mathbf{x}^*) + \mathbf{p}^T \nabla f(\mathbf{x}^*) + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{z}) \mathbf{p} = f(\mathbf{x}^*) + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{z}) \mathbf{p},$$

donde $\mathbf{z} = \mathbf{x}^* + t\mathbf{p}$ para alguna $t \in (0, 1)$. Dado que $\mathbf{z} \in D$, tenemos que $\mathbf{p}^T \nabla^2 f(\mathbf{z}) \mathbf{p} > 0$ y $f(\mathbf{x}^* + \mathbf{p}) > f(\mathbf{x}^*)$. Por lo tanto \mathbf{x}^* es minimizador local estricto de f . \diamond

Teorema 2.5.

Si f es convexa, cualquier minimizador \mathbf{x}^* local es un minimizador global de f . Si en adición f es diferenciable, entonces cualquier punto estacionario es un minimizador global de f .

Prueba:

Suponga \mathbf{x}^* es un minimizador local pero no global. Entonces podemos encontrar un punto $\mathbf{z} \in \Re^n$ tal que $f(\mathbf{z}) < f(\mathbf{x}^*)$. Considere el segmento de línea que une \mathbf{x}^* y \mathbf{z} , esto es:

$$\mathbf{x} = \lambda \mathbf{z} + (1 - \lambda) \mathbf{x}^*$$

para alguna $\lambda \in (0, 1)$. Por propiedad de convexidad de f , tenemos:

$$f(\mathbf{x}) \leq \lambda f(\mathbf{z}) + (1 - \lambda)f(\mathbf{x}^*) < f(\mathbf{x}^*).$$

Cualquier vecindad $N_{\mathbf{x}^*}$ contiene una pieza del segmento descrito, entonces siempre habrá puntos $\mathbf{x} \in N_{\mathbf{x}^*}$ donde $f(\mathbf{x}) < f(\mathbf{x}^*)$. De aquí que \mathbf{x}^* no es un minimizador local. \diamond

2.2. Algoritmos (una visión preliminar).

Dada \mathbf{x}_0 , el algoritmo genera una secuencia de iteraciones $\{\mathbf{x}_k\}_{k=0}^\infty$, tal que termina cuando no hay progreso o cuando ha encontrado una solución con relativa buena resolución.

$$\text{2 tipos de algoritmos} \left\{ \begin{array}{l} \text{Búsqueda en línea,} \\ \text{Región verdadera.} \end{array} \right.$$

2.2.1. Búsqueda en línea.

Dada una dirección \mathbf{p}_k y la posición actual \mathbf{x}_k , la nueva iteración \mathbf{x}_{k+1} busca reducir el valor de la función. El tamaño de paso α a lo largo de la dirección \mathbf{p}_k , se encuentra resolviendo el problema unidimensional:

$$\min_{\alpha > 0} f(\mathbf{x}_k + \alpha \mathbf{p}_k) \quad (2.1)$$

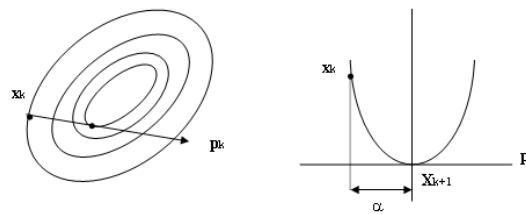


Figura 2.6: Búsqueda en línea.

2.2.2. Métodos de región verdadera.

Se basa en encontrar el mínimo de la función m_k (que es la aproximación simple de f en el punto \mathbf{x}_k) dentro de una “bola” (radio). Esto es:

$$\min_{\mathbf{p}} m_k(\mathbf{x}_k + \mathbf{p}), \quad (2.2)$$

donde $\mathbf{x}_k + \mathbf{p}$ cae en la región verdadera.

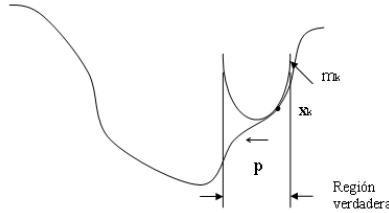


Figura 2.7: Métodos de Región Verdadera.

Generalmente m es una función cuadrática:

$$m(\mathbf{x}_k + \mathbf{p}) = f_k + \mathbf{p}^T \nabla f_k + \frac{1}{2} \mathbf{p}^T \nabla^2 f_k \mathbf{p}. \quad (2.3)$$

donde $f_k = f((x)_k)$, $\nabla f_k = \nabla f((x)_k)$ y $\nabla^2 f_k = \nabla^2 f((x)_k)$

En general, no se busca minimizar (2.1) ni (2.2), esto puede ser muy costoso en términos computacionales, sino encontrar una α y \mathbf{p} respectivamente que reduzcan “suficientemente” el valor de f .

Direcciones de búsqueda para métodos de búsqueda en línea.

La dirección más común es la dirección de máximo descenso $-\nabla f_k$ (es inclusive obvia e intuitiva).

Para verificarlo, por el Teorema de Taylor:

$$f(\mathbf{x}_k + \alpha \mathbf{p}) = f(\mathbf{x}_k) + \underbrace{\alpha \mathbf{p}^T \nabla f_k}_{*} + \underbrace{\frac{1}{2} \alpha^2 \mathbf{p}^T \nabla^2 f(\mathbf{x}_k + t \mathbf{p}) \mathbf{p}}_{>0}$$

para alguna $t \in (0, \alpha)$,

$$\min \mathbf{p}^T \nabla f_k \text{ S.A. } \|\mathbf{p}\| = 1 \text{ (vector unitario)},$$

pero: $\mathbf{p}^T \nabla f_k = \|\mathbf{p}\| \|\nabla f_k\| \cos \theta = \|\nabla f_k\| \cos \theta$.

Luego, el mejor valor de θ para $\cos \theta$ es π , $\cos \pi = -1$, es decir:

$$\mathbf{p} = \frac{-\nabla f_k}{\|\nabla f_k\|},$$

de esta forma \mathbf{p} es ortogonal a las curvas de nivel de f .

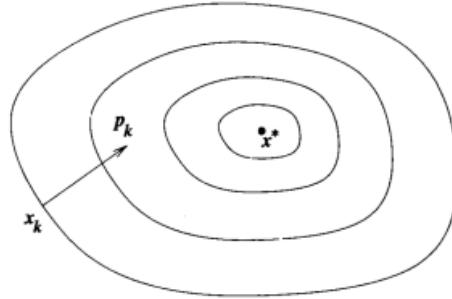


Figura 2.8: Dirección de descenso acelerado para una función de dos variables.

Para una correcta α_k (suficientemente pequeña), cualquier vector \mathbf{v}_k tal que $\mathbf{v}_k \cdot \mathbf{p}_k > 1$ con $\|\mathbf{v}_k\| = 1$ es también una dirección de descenso.

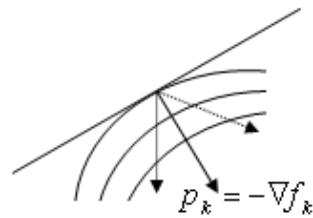


Figura 2.9: Direcciones de descenso.

En el Teorema de Taylor:

$$f(\mathbf{x}_k + \epsilon \mathbf{v}_k) = f(\mathbf{x}_k) + \epsilon \mathbf{p}_k^T \nabla f(\mathbf{x}_k) + O(\epsilon^2). \quad (2.4)$$

Nota: Notación de orden. $O(\cdot)$, $o(\cdot)$ y $\Omega(\cdot)$.

Dadas dos secuencias infinitas de escalares $\{\eta_k\}$ y $\{\nu_k\}$,

- Si $\eta_k = O(\nu_k) \rightarrow \|\eta_k\| \leq C\|\nu_k\|$, donde C es una constante positiva y para toda k suficientemente grande.
- Si $\eta_k = o(\nu_k) \rightarrow$ la secuencia $\left\{ \frac{\eta_k}{\nu_k} \right\} \rightarrow 0$, es decir, $\lim_{k \rightarrow \infty} \frac{\eta_k}{\nu_k} = 0$.
- Si $\eta_k = \Omega(\nu_k) \rightarrow$, hay dos constantes c_1 y c_2 , con $0 \leq c_1 \leq c_2 < \infty$ tales que $c_1\|\nu_k\| \leq \|\eta_k\| \leq c_2\|\nu_k\|$.

Ejemplo:

- $\eta_k = O(1)$ indica que $\eta_k \leq C$ para una $C > 0$ y para toda k .
- $\eta_k = o(1) \rightarrow \lim_{k \rightarrow \infty} \eta_k = 0$

Continuando: $O(\epsilon^2)$ en (2.4) es pequeño para una ϵ^2 suficientemente pequeña y \mathbf{p}_k es una dirección de descenso: $(f(\mathbf{x}_k + \epsilon\mathbf{p}_k) < f(\mathbf{x}_k))$.

Dirección de Newton.

Expansión de segundo orden de Taylor:

$$f(\mathbf{x}_k + \mathbf{p}) \approx f_k + \mathbf{p}^T \nabla f_k + \frac{1}{2} \mathbf{p}^T \nabla^2 f_k \mathbf{p} \stackrel{def}{=} m_k(\mathbf{p}).$$

Asumiendo que $\nabla^2 f_k$ es definida positiva, \mathbf{p} es obtenido tal que minimiza $m_k(\mathbf{p})$ (el tercer término está aproximado respecto al teorema de Taylor).

Haciendo $\frac{\partial m_k(\mathbf{p})}{\partial \mathbf{p}} = 0$, es decir: $\nabla f_k + \nabla^2 f_k \mathbf{p}_k^N = 0$, luego:

$$\mathbf{p}_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k.$$

Esta es una muy buena dirección si en realidad la aproximación de Taylor es buena, es decir, $f(\mathbf{x}_k + \mathbf{p})$ es casi cuadrática.

Inconvenientes:

- Si $\nabla^2 f_k$ no es definida positiva, $(\nabla^2 f_k)^{-1}$ no existe y el método de Newton no puede ser definido.

- Una mala aproximación de segundo orden hace que el método falle.
- El principal problema es el cálculo del Hessiano $\nabla^2 f_k)$
- Cálculo de inversa.

Una forma de evitar el cálculo de la inversa es resolver mediante métodos lineales:

$$\nabla^2 f(\mathbf{x}_k) \mathbf{p} = -\nabla f(\mathbf{x}_k).$$

El cálculo del Hessiano es en general costoso, luego es posible aproximararlo como lo hacen los métodos Quasi-Newton. $H(x) \approx \nabla^2 f(\mathbf{x}_k)$.

Ejemplo de cálculo del Hessiano.

Sean $F : \Re^n \leftarrow \Re$, y $\mathbf{x} = (x_1, \dots, x_n)^T$.

$$\min F(\mathbf{x})$$

Cálculo del gradiente:

$$G(\mathbf{x}) = \nabla F(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}}[F(\mathbf{x})].$$

El Hessiano:

$$\begin{aligned} H(\mathbf{x}) &= \nabla^2 F(\mathbf{x}) = \frac{\partial}{\partial x_j} \{ \nabla F(\mathbf{x}) \} = \frac{\partial}{\partial x_j \partial x_i} \partial F(\mathbf{x}) = \frac{\partial}{\partial x_k} \left[\sum_j \frac{\partial}{\partial x_j} F_i(\mathbf{x}) \right] \\ H(\mathbf{x}) &= \sum_i \left[\frac{\partial}{\partial x_i} F_i(\mathbf{x}) \frac{\partial}{\partial x_j} F_i(\mathbf{x}) + F_j(\mathbf{x}) \frac{\partial^2}{\partial x_i \partial x_j} F(\mathbf{x}) \right] \\ \hat{H}(\mathbf{x}) &= \sum \frac{\partial}{\partial x_k} F_i(\mathbf{x}) \frac{\partial}{\partial x_j} F_i(\mathbf{x}) \end{aligned}$$

Razones de convergencia.

Sea $\{\mathbf{x}_k\}$ una secuencia en \Re^n que converge a \mathbf{x}^* . Se dice que la secuencia tiene convergencia

- Q-lineal si hay una constante $r \in (0, 1)$ tal que:

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} \leq r,$$

(es decir, la distancia a la solución decrece en al menos en factor constante) para toda k suficientemente grande.

- Q-superlineal si

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0,$$

- Q-quadratic (cuadrática) si

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq M,$$

para toda k suficientemente grande y para alguna $M > 0$.

- Q-orden de convergencia ρ ($\rho > 1$) si

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|^\rho} \leq M,$$

para toda k suficientemente grande y para alguna $M > 0$.

R-Razones de convergencia¹(Convergencia más débil).

R-lineal: Sea $\{\nu_k\}$ una secuencia de escalares no negativos tal que $\|\mathbf{x}_k - \mathbf{x}^*\| \leq \nu_k$ para toda k y $\{\nu_k\}$ converge Q-linealmente a cero. Se dice que la secuencia $\{\|\mathbf{x}_k - \mathbf{x}^*\|\}$ es dominada por $\{\nu_k\}$.

¹R-root, raíz.

Capítulo 3

Métodos de Búsqueda en Línea.

La ecuación de recurrencia está dada por:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k. \quad (3.1)$$

En cada iteración: $\begin{cases} * \text{determina la línea de búsqueda } \mathbf{p}_k, \\ * \text{determina el tamaño de paso } \alpha_k. \end{cases}$

\mathbf{p}_k debe ser una dirección de descenso, es decir, $\mathbf{p}_k^T \nabla f_k < 0$ (f es reducida en cada iteración).

En general:

$$\mathbf{p}_k = -B_k^{-1} \nabla f_k, \quad (3.2)$$

donde B_k es una matriz simétrica y no singular.

La diferencia de los métodos de búsqueda en línea radica principalmente en la elección de B_k :

- Máximo descenso: $B_k = \frac{1}{\alpha_k} I$, con I = matriz identidad.
- Newton: $B_k = \nabla^2 f(\mathbf{x}_k)$.
- Quasi-Newton: B_k = aproximación del Hessiano.

Nota: De (3.2) y B definida positiva se tiene:

$$\mathbf{p}_k^T \nabla f_k = -\nabla f_k^T B_k^{-1} \nabla f_k < 0$$

y por lo tanto, \mathbf{p}_k es una dirección de descenso.

En los siguientes capítulos se verá como elegir B , pero antes, se verá la forma de determinar el tamaño de paso α_k .

3.1. Tamaño de paso.

Dada una dirección de descenso \mathbf{p}_k , podemos definir la función ϕ :

$$\phi(\alpha_k) = f(\mathbf{x}_k + \alpha_k \mathbf{p}_k), \quad \alpha_k > 0. \quad (3.3)$$

La idea es que:

$$\alpha_k = \min_{\alpha} \phi(\alpha). \quad (3.4)$$

En general, la minimización en (3.4) es cara.

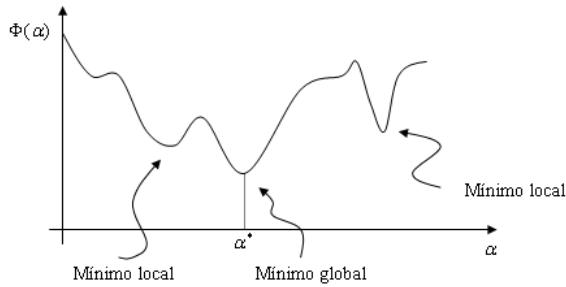


Figura 3.1: Mínimización de Φ .

En vez de minimizar, se elige una secuencia de α 's y se acepta hasta que ciertas condiciones se cumplan.

Ejemplo:

Dados el intervalo $\alpha \in [a, b]$ por bisección o interpolación.

La condición de descenso en α_k :

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k),$$

exige reducción, pero no “suficiente reducción”.

Condiciones de Wolfe.

α_k debe primeramente garantizar “suficiente descenso”:

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^T \mathbf{p}_k \text{ para alguna } c_1 \in (0, 1). \quad (3.5)$$

Nota: El lado derecho es la aproximación de primer orden de Taylor si $c_1 = 1$.

Entonces, la reducción a lo largo de \mathbf{p}_k debe ser proporcional al tamaño del paso.

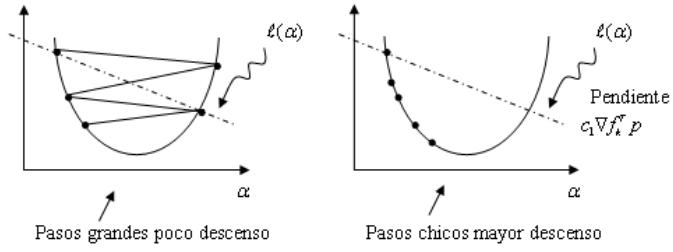


Figura 3.2: Tamaños de paso.

A (3.5) también se le denomina condición de Armijo. (De acuerdo con Nocedal, c_1 generalmente es pequeña: $c_1 = 10^{-4}$).

Condición de curvatura.

El tamaño de paso α_k debe satisfacer:

$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k \geq c_2 \nabla f_k^T \mathbf{p}_k, \quad (3.6)$$

para alguna constante $c_2 \in (c_1, 1)$, con c_1 en (3.5).

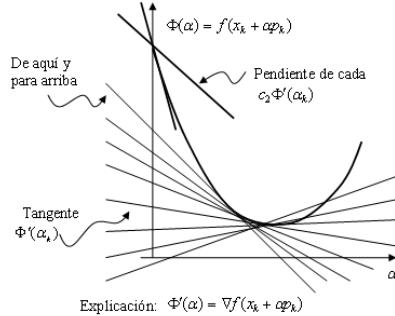


Figura 3.3: Condición de curvatura.

¡Lo que se busca es que la pendiente en el nuevo sitio $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$ sea más positiva que en \mathbf{x}_k ! al menos c_2 veces.

Luego, las condiciones de Wolfe son:

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^T \mathbf{p}_k \quad (3.7)$$

$$\mathbf{p}_k^T \nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \geq c_2 \nabla f_k^T \mathbf{p}_k \text{ con } 0 < c_1 < c_2 < 1. \quad (3.8)$$

Condiciones “fuertes” de Wolfe.

α_k debe satisfacer:

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^T \mathbf{p}_k \quad (3.9)$$

$$\|\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)\| \geq c_2 \|\nabla f_k^T \mathbf{p}_k\| \text{ con } 0 < c_1 < c_2 < 1. \quad (3.10)$$

La diferencia con las condiciones anteriores normales (ó débiles) de Wolfe es que no se aceptan derivadas $\phi'(\alpha_k)$ muy positivas.

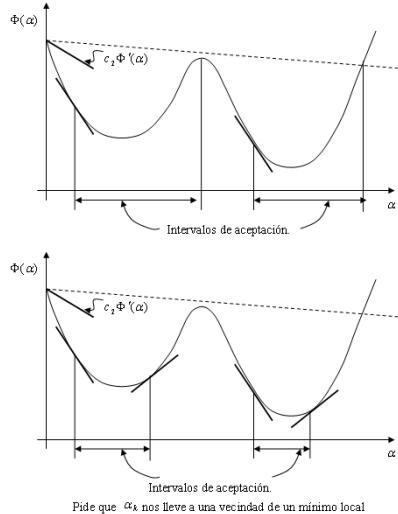


Figura 3.4: Condiciones “Fuertes” de Wolf.

Las condiciones de Wolfe son invariantes a escala, es decir, multiplicar la función f por un escalar no afecta.

Nota: Valores típicos de c_1 y c_2 .

	Newton	Gradiente Conjugado
c_1	10^{-4}	10^{-4}
c_2	0.9	0.1

Condiciones de Goldstein.

α debe satisfacer:

$$f(\mathbf{x}_k) + (1 - c) \alpha_k \nabla f_k^T \mathbf{p}_k \leq f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c \alpha_k \nabla f_k^T \mathbf{x}_k, \quad (3.11)$$

con $0 < c < \frac{1}{2}$.

Primera inecuación: Condición de suficiente descenso.

Segunda inecuación: Introduce control en el tamaño de paso.

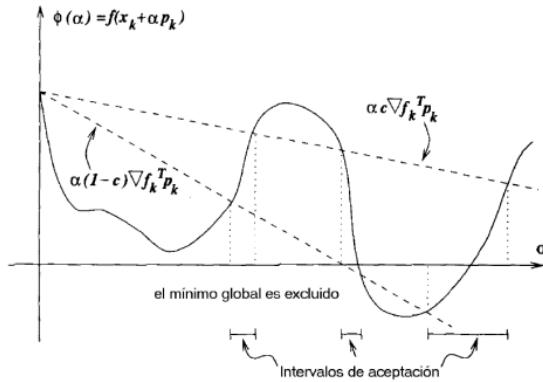


Figura 3.5: Condiciones de Goldstein.

Backtracking.

Backtracking es un algoritmo para elegir α 's dispensando (3.8).

Algoritmo:

```

Elija  $\bar{\alpha} > 0$ ,  $\rho$ , y  $c \in (0, 1)$  y ponga  $\alpha \leftarrow \bar{\alpha}$ 
Repita hasta  $f(\mathbf{x}_k + \alpha \mathbf{p}_k) \leq f(\mathbf{x}_k) + c\alpha \nabla f_k^T \mathbf{x}_k$ 
     $\alpha \leftarrow \rho\alpha$ 
End (Repetir)
Termine  $\alpha_k \leftarrow \alpha$ 

```

La idea es iniciar con α grande o con un método Newton! ($\alpha = 1$) y $\rho \in [\rho_{lo}, \rho_{hi}]$ con $0 < \rho_{lo} < \rho_{hi} < 1$.

Método de descenso. (elección de \mathbf{p}_k).

- Por coordenadas.
- Máximo descenso.

Máximo descenso.

Problema en una dimensión.

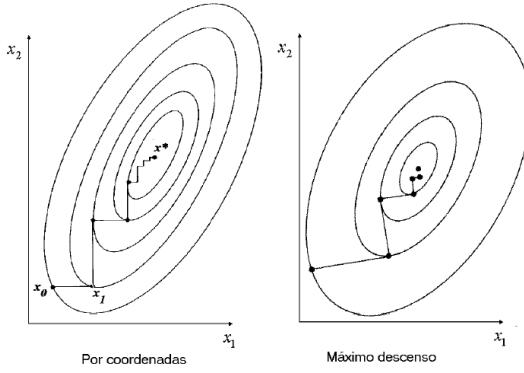


Figura 3.6: Métodos de descenso.

$$\alpha_k = \min_{\alpha} \phi(\alpha),$$

donde $\phi(\alpha) = f(\mathbf{x}_k) + \alpha \mathbf{p}_k$.

Caso cuadrático:

$$m(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k) \approx f(\mathbf{x}_k) + \alpha \nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \frac{1}{2} \alpha^2 \mathbf{p}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{p}_k$$

Solución cuadrática:

$$\min_{\alpha} m(\alpha), \text{ entonces: } \frac{\partial m(\alpha)}{\partial \alpha} = \nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \alpha \mathbf{p}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{p}_k = 0 \Rightarrow \alpha = -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{p}_k}{\mathbf{p}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{p}_k}$$

Otra forma:

Suponga $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{b}^T \mathbf{x}_k + c$, donde Q es una matriz definida positiva, entonces $f'(\mathbf{x}_k) = Q \mathbf{x}_k + \mathbf{b}$, luego α_k resuelve:

$$Q(\mathbf{x}_k + \alpha_k \mathbf{p}_k) + \mathbf{b} = 0$$

de donde:

$$\underbrace{Q \mathbf{x}_k + \mathbf{b}}_{f'(\mathbf{x}_k)} + \alpha_k Q \mathbf{p}_k = 0,$$

multiplicando por \mathbf{p}_k se tiene:

$$\alpha_k \mathbf{p}_k^T Q \mathbf{p}_k + \nabla f_k^T \mathbf{p}_k = 0$$

de aquí que:

$$\alpha_k = -\frac{\nabla f_k^T \mathbf{p}_k}{\mathbf{p}_k^T Q \mathbf{p}_k}.$$

Una forma más:

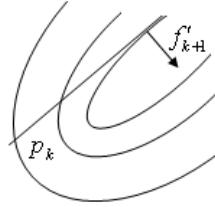


Figura 3.7: Máximo descenso.

Si α_k minimiza en la dirección \mathbf{p}_k , entonces: $(f'_{k+1})^T \mathbf{p}_k = 0$, es decir, son ortogonales, luego:

$$[Q(\mathbf{x}_k + \alpha \mathbf{p}_k) + \mathbf{b}]^T \mathbf{p}_k = [Q(\mathbf{x}_k + \alpha \mathbf{p}_k)]^T \mathbf{p}_k + \mathbf{b}^T \mathbf{p}_k = (Q\mathbf{x}_k + \mathbf{b})^T \mathbf{p}_k + \alpha \mathbf{p}_k^T Q \mathbf{p}_k = 0$$

luego:

$$\nabla f_k^T \mathbf{p}_k + \alpha \mathbf{p}_k^T Q \mathbf{p}_k = 0$$

de donde:

$$\alpha = -\frac{\nabla f_k^T \mathbf{p}_k}{\mathbf{p}_k^T Q \mathbf{p}_k}.$$

3.2. Algoritmos para selección del tamaño de paso.

Interpolación.

Suponga que el tamaño de paso inicial es α_0 (initial guess). Si se cumple:

$$\phi(\alpha_0) \leq \phi(0) + c_1 \alpha_0 \phi'(0),$$

se acepta, es decir, $\alpha_k = \alpha_0$. Si no, sabemos que en el intervalo $[0, \alpha_0]$ existe un tamaño de paso aceptable, luego se puede calcular una estimación (segunda aproximación) por interpolación cuadrática, minimizando:

$$\phi_q(\alpha) = \left(\frac{\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2} \right) \alpha^2 + \phi'(0)\alpha + \phi(0) \quad (3.12)$$

donde $\phi_q(0)$ es una construcción que cumple: $\phi_q(0) = \phi(0)$, $\phi'_q(0) = \phi'(0)$ y $\phi''_q(0) = \phi''(0)$

Nota:

- Aproximación de 2º orden:

$$\phi_q(\alpha) \approx \phi(0) + \alpha \phi'(0) + \frac{1}{2} \alpha^2 \phi''(0)$$

- Aproximación de 2^a derivada por diferencias finitas:

$$\phi'(\frac{\alpha_0}{2}) \approx \frac{\phi(\alpha_0) - \phi(0)}{\alpha_0}$$

$$\phi''(\frac{\alpha_0}{2}) = \frac{(\frac{\phi(\alpha_0) - \phi(0)}{\alpha_0})}{\frac{\alpha_0}{2}} = 2 \left[\frac{\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2} \right].$$

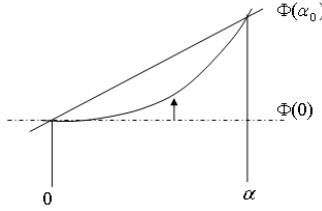


Figura 3.8: Aproximación de segunda derivada por diferencias finitas.

Luego, el minimizador de la cuadrática define el nuevo valor α_1 :

$$\min_{\alpha} \phi_q(\alpha) \rightarrow \alpha_1 = \frac{-\phi'(0)\alpha_0^2}{2[\phi(\alpha_0) - \phi(0) - \alpha_0\phi'(0)]}.$$

Si se cumple la condición de suficiente descenso, se acepta α_1 . Si no cumple la condición de suficiente descenso, es decir, si

$$\phi(\alpha_1) \neq \phi(0) + c_1 \alpha_0 \phi'(0),$$

α_2 se calcula como minimizador de ϕ_c que es la aproximación cúbica

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \alpha\phi'(0) + \phi(0),$$

donde:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_0^2 \alpha_1^2 (\alpha_1 - \alpha_0)} \begin{bmatrix} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{bmatrix} \begin{bmatrix} \phi(\alpha_1) - \phi(0) - \alpha_1 \phi'(0) \\ \phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0) \end{bmatrix}.$$

Diferenciando ϕ_c respecto a α_2 , el valor que cae en el intervalo $[0, \alpha_1]$ es:

$$\alpha_2 = \frac{-b + \sqrt{b^2 - 3a\phi'(0)}}{3a}.$$

El proceso (interpolación cúbica) se repite hasta que la condición de suficiente descenso se cumpla.

Otras estrategias:

Otro caso: Dado un intervalo $[a, b]$: $a \leq \alpha_k \leq b$, y asumiendo conocidos $\phi(\alpha_{i-1})$, $\phi'(\alpha_{i-1})$, $\phi(\alpha_i)$ y $\phi'(\alpha_i)$; utilizando una interpolación cúbica:

$$\alpha_{i+1} = \alpha_i - (\alpha_i - \alpha_{i-1}) \left[\frac{\phi'(\alpha_i) + d_2 - d_1}{\phi'(\alpha_i) - \phi'(\alpha_{i-1}) + 2d_2} \right]$$

con

$$\begin{aligned} d_1 &= \phi'(\alpha_{i-1}) + \phi'(\alpha_i) - 3\frac{\phi_{i-1} - \phi(\alpha_i)}{\alpha_{i-1} - \alpha_i} \\ d_2 &= [d_1^2 - \phi'(\alpha_{i-1})\phi'(\alpha_i)]^{\frac{1}{2}}. \end{aligned}$$

Para el tamaño de paso inicial:

- Asumir que el cambio de primer orden en la función al iterar \mathbf{x}_k será la misma que la obtenida en el paso previo. En otras palabras, escoger α_0 tal que $\alpha_0 \nabla f_k^T \mathbf{p}_k = \alpha_{k-1} \nabla f_{k-1}^T \mathbf{p}_{k-1}$. De aquí que:

$$\alpha_0 = \alpha_{k-1} \frac{\nabla f_{k-1}^T \mathbf{p}_{k-1}}{\nabla f_k^T \mathbf{p}_k}$$

- Interpolación cuadrática de f_{k-1}, f_k y $\phi'(0) = \nabla f_k^T \mathbf{p}_k$ y definir α_0 como su minimizador.

$$\alpha_0 = \frac{2(f_k - f_{k-1})}{\phi'(0)}.$$

Converge superlinealmente, esto es, si tomamos: $\alpha_0 = \min(1, 1.01\alpha_0)$, este paso será eventualmente aceptado.

Búsqueda en línea para las condiciones de Wolfe.

Para c_1, c_2 tal que $0 < c_1 < c_2 < 1$ y \mathbf{p} es una dirección de descenso y f es acotada por abajo en la dirección \mathbf{p} , ($f(\mathbf{x}) > c > -\infty \forall \mathbf{x}$).

El algoritmo tiene dos etapas:

- intenta una α_1 inicial y se mantiene incrementando hasta encontrar un paso aceptable ó un intervalo que lo contiene.
- Si se tiene un intervalo, invoca a una función **zoom** que sucesivamente decrece el tamaño del intervalo hasta que un tamaño de paso es identificado.

Algoritmo 3.2 (Búsqueda en línea).

Set $\alpha_0 \leftarrow 0$, choose $\alpha_1 > 0$ and α_{\max}

$i \leftarrow 1$

Repeat

Evaluate $\phi(\alpha_i)$;

(i) **If** [$\phi(\alpha_i) > \phi(0) + c_1 \alpha_i \phi'(0)$ or $(\phi(\alpha_i) \geq \phi(\alpha_{i-1}) \text{ and } i > 1)$]

$\alpha_* \leftarrow \text{zoom}(\alpha_{i-1}, \alpha_i)$ and **stop**;

Evaluate $\phi'(\alpha_i)$;

(ii) **If** $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$
set $\alpha_* \leftarrow \alpha_i$ and **stop**;
(iii) **If** $\phi'(\alpha_i) \geq 0$
set $\alpha_* \leftarrow \text{zoom}(\alpha_i, \alpha_{i-1})$ and **stop**;
Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$
 $i \leftarrow i + 1$;

End(Repeat).

En la llamada a **zoom** el intervalo (α_{i-1}, α_i) contiene el paso deseado.

- (i) Viola suficiente descenso,
- (ii) $\phi(\alpha_i) \geq \phi(\alpha_{i-1})$,
- (iii) $\phi'(\alpha_j) \geq 0$.

Función zoom $(\alpha_{lo}, \alpha_{hi})$ donde $\alpha_{lo} \leq \alpha_* \leq \alpha_{hi}$

Repeat

Interpolar usando (cuadrática, cúbica ó bisección) para encontrar un α_j de prueba: $\alpha_{lo} \leq \alpha_j \leq \alpha_{hi}$;

Evaluar $\phi(\alpha_j)$;

If $[\phi(\alpha_j) > \phi(0) + c_i \alpha_j \phi'(0) \text{ or } \phi(\alpha_j) \geq \phi(\alpha_{lo})]$

$\alpha_{hi} \leftarrow \alpha_j$;

else

Evaluate $\phi'(\alpha_j)$;

If $|\phi'(\alpha_j)| \leq -c_2\phi'(0)$

set $\alpha_* \leftarrow \alpha_j$ and **stop**;

If $\phi'(\alpha_j)(\alpha_{hi} - \alpha_{lo}) \geq 0$

$\alpha_{hi} \leftarrow \alpha_{lo}$

$\alpha_{lo} \leftarrow \alpha_j$

End (Repeat)

Capítulo 4

Métodos de Región Verdadera.

Métodos de búsqueda en línea.

- Generan una dirección de descenso.
- Encuentran un tamaño de paso α .

Los Métodos de Región Verdadera (MRV) definen una región alrededor de la iteración actual y luego ajustan el modelo y lo minimizan. Los MRV eligen simultáneamente la dirección de descenso y el tamaño de paso.

El tamaño ó radio de la región verdadera es clave:

- muy chica pierde la oportunidad de hacer un buen avance en la minimización de la función objetivo.
- muy grande, el modelo puede ser muy diferente de la función objetivo, lo que complica la minimización en la región verdadera.

Construcción de un modelo de f :

$$m_k(\mathbf{p}) = f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f_k \mathbf{p},$$

donde $\nabla^2 f_k$ es el Hessiano de f ; o mejor aún:

$$m_k(\mathbf{p}) = f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_k \mathbf{p},$$

donde B_k es una matriz definida positiva. En general esperamos $B_k \approx \nabla^2 f_k$.

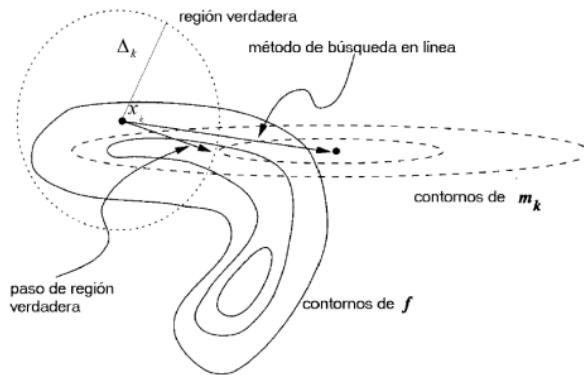


Figura 4.1: Métodos de Región Válida.

El problema de región verdadera se puede escribir como:

$$\min_{\mathbf{p} \in \mathbb{R}^n} m_k(\mathbf{p}) = f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T b_k \mathbf{p} \quad \text{S.A. } \|\mathbf{p}\| \leq \Delta_k, \quad (4.1)$$

donde Δ_k es el radio de la región verdadera, y $\|\cdot\|$ es la norma euclíadiana, ($\|\mathbf{p}\|^2 = \mathbf{p}^T \mathbf{p}$).

MRV consiste en minimizaciones sucesivas de (4.1).

El caso trivial de (4.1) es cuando B_k es definida positiva y $\|B_k^{-1} \nabla f_k\| \leq \Delta_k$, en este caso:

$$\mathbf{p}_k^B = -B_k^{-1} \nabla f_k,$$

y \mathbf{p}_k^B es el paso completo. En otros casos, esa solución no es obvia.

Nosotros buscamos una solución aproximada (como en el caso de los métodos de búsqueda en línea).

Radio de la Región Verdadera.

Definimos la razón ρ_k como:

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{p}_k)}{m_k(0) - m_k(\mathbf{p}_k)}. \quad (4.2)$$

En (4.2) el numerador es la reducción real, el denominador la reducción predicha (> 0) y \mathbf{p}_k es el minimizador de m_k .

Para ρ_k tenemos lo siguiente:

- Si $\rho_k < 0$ entonces $f_k < f(\mathbf{x}_k + \mathbf{p}_k)$. Es necesario reducir Δ_k y desechar \mathbf{p}_k .

- Si $\rho \approx 1$ el modelo m_k “ajusta” bien a f_k y se puede expandir el radio en la siguiente iteración.
- Si $\rho \approx 0$ ajustar Δ_k (reducir).
- Si $0 < \rho_k \leq 1$ acepto Δ_k (no modificar la región).
- Si $\rho > 1$ acepto Δ_k (posiblemente aumentar Δ_k).

Algoritmo 4.1 para ajustar Δ_k .

Dado $\bar{\Delta} > 0$ (radio máximo), $\Delta_0 \in (0, \bar{\Delta})$ y $\eta \in [0, \frac{1}{4}]$;
for $k = 0, 1, 2, \dots$

 Obtener \mathbf{p}_k (aproximadamente) resolviendo (4.1) (satisfaciendo $\rho_k > 0$);

 Evaluar ρ_k de (4.2);

if $\rho_k < \frac{1}{4}$

$$\Delta_{k+1} = \frac{1}{4} \|\mathbf{p}_k\|$$

else

if $\rho_k > \frac{3}{4}$ and $\|\mathbf{p}_k\| = \Delta_k$

$$\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$$

else

$$\Delta_{k+1} = \Delta_k;$$

if $\rho_k > \eta$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$$

else

$$\mathbf{x}_{k+1} = \mathbf{x}_k$$

end (for)

Nota:

El tamaño de radio se incrementa solo si $\|\mathbf{p}_k\| = \Delta_k$. Si es menor, lo dejamos.

En la siguiente sección se verán algunas estrategias para aproximar el mínimo para (4.1).

4.1. Punto de Cauchy.

Un método para calcular \mathbf{p}_k es punto de Cauchy.

Cálculo del punto de Cauchy:

Encuentre \mathbf{p}_k^s que resuelve la versión lineal de (4.1), esto es:

$$\mathbf{p}_k^s = \arg \min_{\mathbf{p} \in \Re^n} f_k + \nabla f_k^T \mathbf{p} \quad \text{S.A. } \mathbf{p}^T \mathbf{p} \leq \Delta_k^2 \quad (4.3)$$

Calcular el escalar $\tau_k > 0$ que minimiza $m_k(\tau \mathbf{p}_k^s)$ sujeta a la región verdadera, esto es:

$$\tau_k = \arg \min_{\tau > 0} m_k(\tau \mathbf{p}_k^s) \quad \text{S.A. } \|\tau \mathbf{p}_k^s\| \leq \Delta_k; \quad (4.4)$$

Hacer $\mathbf{p}_k^C = \tau_k \mathbf{p}_k^s$.

De hecho, es fácil escribir una forma cerrada del punto de Cauchy. La solución de (4.3) es simplemente:

$$\mathbf{p}_k^s = -\frac{\Delta_k}{\|\nabla f_k\|} \nabla f_k$$

Para obtener τ_k explícitamente, consideraremos los casos $\nabla f_k^T B_k \nabla f_k \leq 0$ y $\nabla f_k^T B_k \nabla f_k > 0$ separadamente. El primer caso es cuando la solución cae dentro de la región verdadera, y en este caso:

$$\mathbf{p}_k^C = -\tau_k \frac{\Delta_k}{\|\nabla f_k\|} \nabla f_k$$

donde

$$\tau_k = \begin{cases} 1 & \text{si } \nabla f_k^T B_k \nabla f_k \leq 0 \\ \min \left(\frac{\|\nabla f_k\|^3}{\Delta_k \nabla f_k^T B_k \nabla f_k}, 1 \right) & \text{en otro caso.} \end{cases}.$$

En el segundo caso $m_k(\tau \mathbf{p}_k^s)$ es convexa y cuadrática, entonces τ_k es el minimizador de la cuadrática o el valor de frontera 1, esto es:

$$\tau_k = \min_{\tau} f_k + \tau (\mathbf{p}_k^s)^T \nabla f_k + \frac{1}{2} \tau^2 (\mathbf{p}_k^s)^T B_k \mathbf{p}_k^s.$$

Derivando respecto a τ , e igualando a cero, tenemos:

$$(\mathbf{p}_k^s)^T \nabla f_k + \tau (\mathbf{p}_k^s)^T B_k \mathbf{p}_k^s = 0$$

de donde:

$$\tau_k = \frac{-(\mathbf{p}_k^s)^T \nabla f_k}{(\mathbf{p}_k^s)^T B_k \mathbf{p}_k^s}$$

sustituyendo

$$\mathbf{p}_k^s = -\Delta_k \frac{\nabla f_k}{\|\nabla f_k\|}.$$

tenemos:

$$\tau_k = \frac{\Delta_k \|\nabla f_k\|}{\frac{\Delta_k^2}{\|\nabla f_k\|^2} [\nabla f_k^T B_k \nabla f_k]} = \frac{\|\nabla f_k\|^3}{\Delta_k [\nabla f_k^T B_k \nabla f_k]}.$$

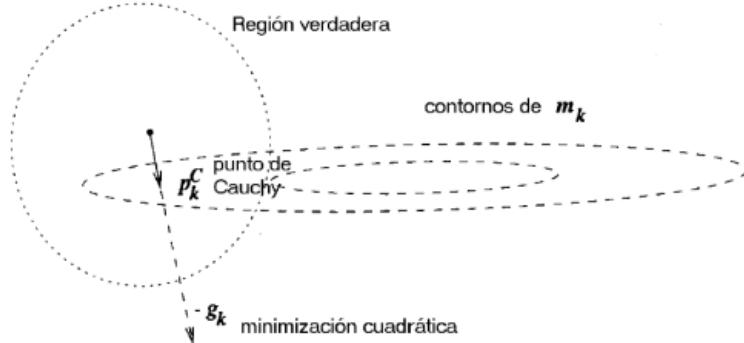


Figura 4.2: Punto de Cauchy.

El método Dogleg. (“del camino torcido”, Dogleg = pata de perro).

Revisar el efecto del radio Δ de la región verdadera en:

$$\min_{\mathbf{p} \in \mathbb{R}^n} m_k(\mathbf{p}) \stackrel{def}{=} f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_k \mathbf{p} \quad (4.5)$$

donde B_k es el Hessiano.

- Si Δ es muy pequeña el efecto del término cuadrático es despreciable, luego, $m_k(\mathbf{p})$ puede aproximarse muy bien por el término lineal y la solución óptima estará dada por el punto de Cauchy:

$$\mathbf{p}^*(\Delta) \approx -\Delta \frac{\nabla f}{\|\nabla f\|} \text{ para } \Delta \text{ pequeña, con } \|\mathbf{p}\| \leq \Delta.$$

- Para Δ suficientemente grande, si B es definida positiva, el minimizador es el punto de Newton, dado por:

$$\mathbf{p}_k^B = -B_k^{-1} \nabla f_k$$

y la solución óptima será:

$$\mathbf{p}^*(\Delta) = \mathbf{p}_k^B \quad \text{si } \|\mathbf{p}_k\| \leq \Delta.$$

- Para Δ intermedios se tiene una trayectoria curva.

Una mejor aproximación que el punto de Cauchy para Δ 's no necesariamente chicas es:

$$\mathbf{p}^U = -\frac{g^T g}{g^T B g} g \quad \text{donde } g \stackrel{def}{=} \nabla f \quad (4.6)$$

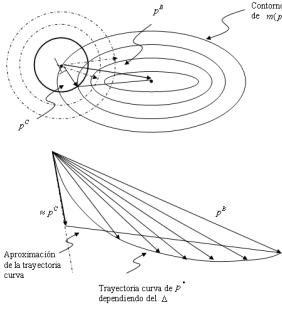


Figura 4.3: Punto de Cauchy y solución de Newton.

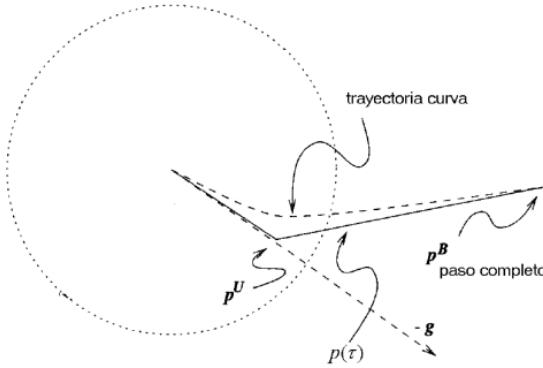


Figura 4.4: Método de Dogleg.

que es la solución de $m_k(\mathbf{p})$ en la dirección de ∇f_k sin restricciones.

Es posible crear un conjunto de soluciones usando \mathbf{p}^U y \mathbf{p}^B (con combinaciones lineales de ellas.)

Dadas \mathbf{p}^U y \mathbf{p}^B , sea $\tau \in [0, 2]$

$$\mathbf{p}(\tau) = \begin{cases} \tau \mathbf{p}^U & 0 \leq \tau \leq 1 \\ \mathbf{p}^U + (\tau - 1)(\mathbf{p}^B - \mathbf{p}^U) & \text{en otro caso} \end{cases}. \quad (4.7)$$

El método Dogleg elige \mathbf{p} como el minimizador del modelo m :

$$\min_{\mathbf{p} \in \Re^n} m(\mathbf{p}) = f_k + \mathbf{g}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B \mathbf{p},$$

a lo largo de la trayectoria definida por (4.7) sujeto a la restricción de la región verdadera.

Dado que la trayectoria intersecta a lo mucho una vez la frontera de la región verdadera, este punto

puede ser calculado analíticamente.

Es posible demostrar que $m(0) \geq m(\mathbf{p}^U) \geq m(\mathbf{p}^B)$.

Lema 4.1.

Sea B definida positiva. Entonces:

- i) $\|\mathbf{p}(\tau)\|$ es una función incremental de τ , y
- ii) $m(\mathbf{p}(\tau))$ es una función decremental de τ .

Prueba:

Para $\tau \in [0, 1]$ es fácil probar y ver para $\tau \in (1, 2]$.

Para i) definamos:

$$\begin{aligned} h(\alpha) &= \frac{1}{2} \|\mathbf{p}^U + \alpha(\mathbf{p}^B - \mathbf{p}^U)\|^2 \text{ con } \alpha > 0 \\ &= \frac{1}{2} \|\mathbf{p}^U\|^2 + \alpha (\mathbf{p}^U)^T (\mathbf{p}^B - \mathbf{p}^U) + \frac{1}{2} \alpha^2 \|\mathbf{p}^B - \mathbf{p}^U\|^2 \end{aligned}$$

luego calculando $h'(\alpha)$:

$$h'(\alpha) = \mathbf{p}^U (\mathbf{p}^B - \mathbf{p}^U) + \alpha \|\mathbf{p}^B - \mathbf{p}^U\|^2$$

tenemos:

$$\begin{aligned} h'(\alpha) &\geq \mathbf{p}^U (\mathbf{p}^B - \mathbf{p}^U) = -\frac{g^T g}{g^T B g} g^T \left(-\frac{g^T g}{g^T B g} g + B^{-1} g \right) = \\ &= g^T g \frac{g^T B^{-1} g}{g^T B g} \left(1 - \frac{(g^T g)^2}{(g^T B g)(g^T B^{-1} g)} \right) \geq 0. \end{aligned}$$

Para ii) definimos $\hat{h}(\alpha) = m(\tilde{p}(1 + \alpha))$ y se muestra que $\hat{h}'(\alpha) \leq 0$ para $\alpha \in [0, 1]$ sustituyendo (4.7) en (4.5) y diferenciando respecto al argumento.

Luego, siguiendo del Lema 4.1: $\widetilde{\mathbf{p}}(\tau)$ intersecta la región en un punto si $\|\mathbf{p}^B\| \geq \Delta$ y en ningn lugar en otro caso.

El valor de τ para el caso de intersección resuelve la ecuación cuadrática:

$$\left\| \underbrace{\mathbf{p}^U + (\tau - 1)(\mathbf{p}^B - \mathbf{p}^U)}_{vector al cuadrado} \right\|^2 = \underbrace{\Delta^2}_{radio}$$

Algoritmo Dogleg. Finalmente el algoritmo está dado por:

Dada una Δ calculada con el procedimiento de región verdadera:

a) Calcular $\mathbf{p}^U, \mathbf{p}^B$:

$$\begin{aligned}\mathbf{p}^U &= -\frac{g^T g}{g^T B g} \left(\frac{g}{\|g\|} \right) \quad \text{con } g \stackrel{\text{def}}{=} \nabla f_k \\ \mathbf{p}^B &= -B^{-1}g.\end{aligned}$$

b) Luego:

$$\text{Si } \|\mathbf{p}^B\| \leq \Delta \rightarrow \mathbf{p}^* = \mathbf{p}^B$$

$$\text{Si } \|\mathbf{p}^U\| \geq \Delta \rightarrow \mathbf{p}^* = -\Delta \frac{g}{\|g\|}$$

Si no (otros): $\mathbf{p}^* = \mathbf{p}(\tau^*)$ donde τ resuelve $\|\mathbf{p}^U + (\tau - 1)(\mathbf{p}^B - \mathbf{p}^U)\|^2 = \Delta^2$, S.A. $\tau \in [1, 2]$.

Capítulo 5

Métodos de Gradiente Conjugado.

Utilidad:

- Resolver sistemas lineales grandes.
- Problemas de optimización cuadrática (no lineal).

Tres definiciones importantes:

- A es definida positiva y simétrica si: $\mathbf{x}^T A \mathbf{y} = \mathbf{y}^T A \mathbf{x}$ (simétrica), y $\mathbf{x}^T A \mathbf{x} \geq 0$ ($\mathbf{x}^T A \mathbf{x} = 0 \Leftrightarrow \mathbf{x} = 0$)
- Si $\mathbf{x}^T A \mathbf{y} = 0$ y $\mathbf{x} \neq \mathbf{y}$ entonces \mathbf{x} y \mathbf{y} son A -conjugados.
- Si $\mathbf{y}^T \mathbf{x} = \mathbf{x}^T \mathbf{y} = 0$ y $\mathbf{x} \neq \mathbf{y}$ entonces \mathbf{x} y \mathbf{y} son ortogonales.

Máximo descenso

Problema con A definida positiva:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b}, \quad (5.1)$$

equivale a resolver:

$$\mathbf{g}(\mathbf{x}) \stackrel{\text{def}}{=} \nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = 0. \quad (5.2)$$

Fórmula iterativa de actualización de \mathbf{x} :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \quad (5.3)$$

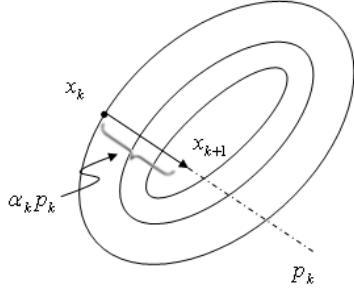


Figura 5.1: Actualización de \mathbf{x} en método de descenso.

donde α_k es el tamaño de paso y \mathbf{p}_k es la dirección de descenso.

Asumiendo \mathbf{p}_k como una dirección de descenso, el cálculo del tamaño de paso α_k satisface:

$$\alpha_k^* = \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{p}_k),$$

o lo que es lo mismo:

$$\mathbf{p}_k^T \mathbf{g}_{k+1} = 0 \Rightarrow \mathbf{p}_k^T \mathbf{g}(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = 0.$$

El gradiente en \mathbf{x}_{k+1} es ortogonal a la dirección de descenso actual.

$$\begin{aligned} \mathbf{g}_{k+1}^T \mathbf{p}_k &= 0 \Rightarrow (\mathbf{A}\mathbf{x}_{k+1} - \mathbf{b})^T \mathbf{p}_k = 0 \\ [\mathbf{A}(\mathbf{x}_k + \alpha_k \mathbf{p}_k) - \mathbf{b}]^T \mathbf{p}_k &= 0 \\ [\mathbf{A}\mathbf{x}_k + \alpha_k \mathbf{A}\mathbf{p}_k - \mathbf{b}]^T \mathbf{p}_k &= 0 \\ \mathbf{p}_k^T [\alpha_k \mathbf{A}\mathbf{p}_k + \mathbf{g}_k] &= 0 \end{aligned}$$

luego:

$$\alpha_k = \frac{\mathbf{p}_k^T \mathbf{g}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} \quad (5.4)$$

En particular para $\mathbf{p}_k = -\mathbf{g}_k$:

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T \mathbf{A} \mathbf{g}_k}. \quad (5.5)$$

5.1. Método de Gradiente Conjugado lineal.

Direcciones principales de A definida positiva.

Si descendemos por las direcciones correspondientes a los eigenvectores podemos minimizar f (o resolver g) en n iteraciones! usando pasos de máximo descenso.

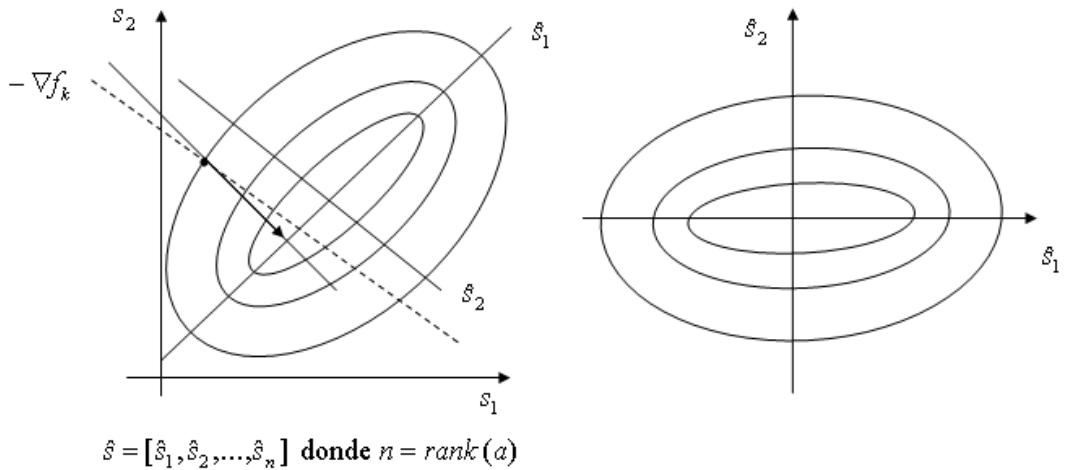


Figura 5.2: Direcciones principales.

desafortunadamente, el cálculo de los eigenvectores (vectores propios) es tan costoso como resolver el sistema con el método de Gauss.

Direcciones conjugadas de A definida positiva.

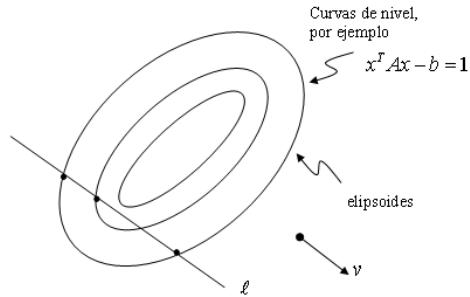


Figura 5.3: Direcciones conjugadas.

- Una línea ℓ orientada conforme un vector cualquiera \mathbf{v} , cruza a lo más en dos puntos cualquier curva de nivel de la forma cuadrática $\mathbf{x}^T A \mathbf{x}$.
- Cada línea ℓ_j de la familia de curvas $L = \ell_1, \dots, \ell_n$ orientadas por \mathbf{v}_j , también cruzan las curvas de nivel de $\mathbf{x}^T A \mathbf{x}$ en a lo más dos puntos.
- Los puntos medios de las cuerdas están alineados y orientados por \mathbf{v}_2 .
- El punto medio de la cuerda que pasa por el origen o centro de las curvas de nivel $\mathbf{x}^* \mathbf{x}$ es \mathbf{x}^* .

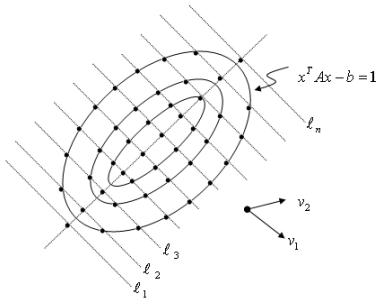


Figura 5.4: Propiedades de direcciones conjugadas.

Las direcciones \mathbf{v}_2 y \mathbf{v}_1 son A -conjugadas, es decir:

$$\mathbf{v}_2^T A \mathbf{v}_1 = 0, \quad (5.6)$$

¡no necesariamente ortogonales! ($\mathbf{v}_1^T \mathbf{v}_2 \neq 0$).

El número de direcciones conjugadas de una matriz A positiva definida es igual a $\text{rank}(A)$ (rango de A).

De nuestro ejemplo, vemos que las familias de direcciones conjugadas posibles en A es infinito y depende de mi vector inicial \mathbf{v}_1 .

El procedimiento de máximo descenso a lo largo de las direcciones conjugadas minimiza f en el número de pasos igual al número de direcciones conjugadas $\text{rank}(A)$.

¡El problema ahora es como calcular las direcciones conjugadas!.

Sea \mathbf{p}_k la dirección anterior de descenso, la nueva dirección \mathbf{p}_{k+1} debe ser conjugada, es decir,

$$\mathbf{p}_k^T A \mathbf{p}_{k+1} = 0. \quad (5.7)$$

La nueva dirección \mathbf{p}_{k+1} la elegimos como una combinación lineal del gradiente en \mathbf{x}_{k+1} y \mathbf{p}_k :

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \beta \mathbf{p}_k, \quad (5.8)$$

sujeta a (5.7):

$$\mathbf{p}_{k+1}^T A (-\mathbf{g}_{k+1} + \beta \mathbf{p}_k) = -\mathbf{g}_{k+1}^T A \mathbf{p}_k + \beta \mathbf{p}_k^T A \mathbf{p}_k.$$

Por lo tanto:

$$\beta = \frac{-\mathbf{g}_{k+1}^T A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}. \quad (5.9)$$

Si A es positiva definida, el cálculo del gradiente puede ser fácilmente actualizado usando:

$$\mathbf{g}_{k+1} = A\mathbf{x}_{k+1} - \mathbf{b} = A(\mathbf{x}_k + \alpha_k \mathbf{p}_k) - \mathbf{b} = A\mathbf{x}_k - \mathbf{b} + \alpha_k A \mathbf{p}_k.$$

entonces:

$$\mathbf{g}_{k+1} = \mathbf{g}_k + \alpha_k A \mathbf{p}_k. \quad (5.10)$$

De (5.10) podemos ver que $\alpha_k A \mathbf{p}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.

Luego, la fórmula para β ,

$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k},$$

puede ser simplificada, de (5.10) tenemos:

$$A \mathbf{p}_k = \frac{1}{\alpha_k} (\mathbf{g}_{k+1} - \mathbf{g}_k),$$

luego

$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \frac{1}{\alpha_k} (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{p}_k^T \frac{1}{\alpha_k} (\mathbf{g}_{k+1} - \mathbf{g}_k)},$$

donde

$$\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k) = \mathbf{g}_{k+1}^T \mathbf{g}_{k+1} - \mathbf{g}_{k+1}^T \mathbf{g}_k.$$

Teorema 5.2. Sea $\mathbf{x}_0 \in \Re^n$ un punto de inicio y suponga que la secuencia $\{\mathbf{x}_n\}$ es generada por el algoritmo de direcciones conjugadas, entonces:

$$\mathbf{g}_k^T \mathbf{p}_i = 0 \quad \text{para } i = 0, 1, \dots, k-1 \quad (5.11)$$

“Todos los residuales son ortogonales a todas las direcciones previas de búsqueda”.

Prueba:

Asumiendo que α_k es el minimizador. Tenemos que $\mathbf{g}_1^T \mathbf{p}_0 = 0$, luego por inducción construimos la hipótesis $\mathbf{g}_{k-1}^T \mathbf{p}_i = 0$, luego:

$$\mathbf{p}_{k-1}^T \mathbf{g}_k = \mathbf{p}_{k-1}^T (\mathbf{g}_{k-1} + \alpha_{k-1} \mathbf{p}_{k-1}^T A \mathbf{p}_{k-1}) = 0,$$

para la i -ésima dirección (\mathbf{p}_i con $i = 0, 1, \dots, k-1$):

$$\mathbf{p}_i^T \mathbf{g}_k = \underbrace{\mathbf{p}_i^T \mathbf{g}_{k-1}}_{=0} + \alpha_{k-1} \underbrace{\mathbf{p}_i^T A \mathbf{p}_{k-1}}_{=0} = 0.$$

Por lo tanto: $\mathbf{p}_i^T \mathbf{g}_k = 0$. \diamond

Nota: $\mathbf{p}_i^T A \mathbf{p}_{k-1} = 0$ porque todas las direcciones son conjugadas.

Teorema 5.3.

Todos los gradientes (residuos) son ortogonales, esto es:

$$\mathbf{g}_k^T \mathbf{g}_i = 0 \quad \text{para } i = 0, 1, \dots, k-1. \quad (5.12)$$

Prueba:

Por el Teorema 5.2 tenemos que $\mathbf{g}_k^T \mathbf{p}_i = 0$ para $i = 0, 1, \dots, k-1$. Usando $\mathbf{p}_i = -\mathbf{g}_i^T + \beta \mathbf{p}_{i-1}$, tenemos:

$$\mathbf{g}_k^T (-\mathbf{g}_i + \beta \mathbf{p}_{i-1}) = -\mathbf{g}_k^T \mathbf{g}_i + \beta_i \underbrace{\mathbf{g}_k^T \mathbf{p}_i}_{=0 \text{ por } \text{teo. 5.2.}} = 0.$$

Por lo tanto, $\mathbf{g}_k^T \mathbf{g}_i = 0$. \diamond

Usando el teorema 5.3 y 5.2 y $\mathbf{p}_k = -\mathbf{g}_{k-1} + \beta \mathbf{p}_{k-1}$

$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_{k+1}}. \quad (5.13)$$

Algoritmo 5.1. Gradiente Conjugado. (versión preliminar).

Dado \mathbf{x}_0 ;

Poner $\mathbf{g}_0 = A\mathbf{x}_0$, $\mathbf{p}_0 = -\mathbf{g}_0$ y $k = 0$;

while $\mathbf{g}_k \neq 0$ ($k < \text{rank}(A)$ y $\|\mathbf{g}_k\| < \epsilon$)

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{g}_{k+1} = A\mathbf{x}_{k+1} - \mathbf{b}$$

$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$$

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \beta_{k+1} \mathbf{p}_k$$

$$k = k + 1$$

end (while)

Algoritmo 5.2. Versión práctica de GC-lineal

Dado \mathbf{x}_0 ;

Poner $\mathbf{g}_0 = A\mathbf{x}_0$, $\mathbf{p}_0 = -\mathbf{g}_0$ y $k = 0$;

while $\mathbf{g}_k \neq 0$ ($k < \text{rank}(A)$ y $\|\mathbf{g}_k\| < \epsilon$)

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{g}_{k+1} = \mathbf{g}_k + \alpha_k A \mathbf{p}_k$$

$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$$

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \beta_{k+1} \mathbf{p}_k$$

```

k = k + 1
end (while)

```

Sistemas grandes y ralos.

A es una matriz con pocos elementos distintos de cero y con rango $\text{rank}(A)$ grande (por ejemplo digamos $A_{256 \times 256}$ o mayor) y bandada (sólo tres diagonales son distintas de cero):

$$F(\hat{g}) = \sum_x (g(x) - \hat{g}(x))^2 + \lambda \sum_x \|\nabla \hat{g}(x)\|^2. \quad \dots(1)$$

Suponda el problema donde $g(x)$ es una función discreta de N muestras ($x = 0, 1, \dots, N-1$). Calcular $\hat{g}(x)$ usando (1).

$$F(\hat{g}) = \sum_{x=0}^{N-1} (g(x) - \hat{g}(x))^2 + \lambda \sum_{x=1}^{N-1} (\hat{g}(x) - \hat{g}(x-1))^2.$$

Del procedimiento de minimización:

$$\frac{1}{2} \frac{\partial F(\hat{g})}{\partial \hat{g}(k)} = 0,$$

tenemos:

$$(g(k) - \hat{g}(k))(-1) + \lambda(\hat{g}(k) - \hat{g}(k-1)) - \lambda(\hat{g}(k+1) - \hat{g}(k)) = 0$$

$$\hat{g}(k) - g(k) + 2\lambda\hat{g}(k) - \lambda\hat{g}(k-1) - \lambda\hat{g}(k+1) = 0$$

$$\begin{pmatrix} 1+\lambda & -\lambda & & & \\ -\lambda & 1+2\lambda & -\lambda & & \\ & -\lambda & 1+2\lambda & -\lambda & \\ & & \ddots & & \\ & & & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} \hat{g}(0) \\ \vdots \\ \hat{g}(k-1) \\ \hat{g}(k) \\ \vdots \\ \hat{g}(N-1) \end{pmatrix} = \begin{pmatrix} g(0) \\ \vdots \\ g(k-1) \\ g(k) \\ \vdots \\ g(N-1) \end{pmatrix}.$$

Almacenar A es muy costoso, más eficiente es contar con una función que calcule el producto de A por un vector \mathbf{z} .

```

void producto_Az(float *Az, float *z, int N)
{
    for (int i = 0; i < N; i++)
    {
        float producto= z[i];
        if (i!= 0)
            producto += lambda*(z[i] - z[i - 1]);
    }
}

```

```

if ( $i < N - 1$ )
    producto += lambda*(z[i] - z[i + 1]);
    Az[i] = producto;
} // end for
}

```

En el caso de Newton: $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_{k+1}$ donde:

$$\begin{aligned}\delta_{k+1} &= A^{-1}\mathbf{g}_k. \\ \delta_{k+1} &= \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T A^{-1} \mathbf{g}_k} \mathbf{g}_k \quad \text{si } A \approx I. \\ \widehat{g}(k) &= \frac{g(k) + \lambda(\widehat{g}(k+1) + \widehat{g}(k-1))}{1 + 2\lambda} \quad (\text{Gauss - Seidel}).\end{aligned}$$

Precondicionamiento.

Cambiar el vector de incógnitas \mathbf{x} por $\widehat{\mathbf{x}}$ por medio de una matriz C no singular:

$$\widehat{\mathbf{x}} = C\mathbf{x} \rightarrow \mathbf{x} = C^{-1}\widehat{\mathbf{x}},$$

luego, del problema cuadrático:

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

tenemos:

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

Calculando el gradiente (para minimizar):

$$(C^{-T} A C^{-1})\widehat{\mathbf{x}} = C^{-T} \mathbf{b}.$$

Ahora, la razón de convergencia dependerá de los eigenvalores de la matriz $C^{-T} A C^{-1}$ en vez de sólo A !

Debemos elegir C tal que la nueva matriz sea más conveniente, es decir, que sea fácil de invertir y que el número de condición de A sea mucho mayor que el número de condición de $C^{-T} A C^{-1}$, esto es: $K(C^{-T} A C^{-1}) \ll K(A)$.

Nota: Número de condición de una matriz A :

$$K(A) = \frac{\lambda_{1A}}{\lambda_{nA}}$$

donde λ_1 es el eigenvalor mayor, λ_n es el eigenvalor menor y $n = \text{rank}(A)$.

Definiendo $M = C^T C$ y aplicando los pasos anteriores para el GC podemos llegar al siguiente algoritmo:

Algoritmo 5.3. GC-Precondicionado

Given \mathbf{x}_0 , the preconditioner M ;

Set $\mathbf{g}_0 = A\mathbf{x}_0 - \mathbf{b}$;

Solve $M\mathbf{y}_0 = \mathbf{g}_0$ for \mathbf{y}_0 ;

Set $\mathbf{p}_0 = -\mathbf{g}_0$, $k = 0$;

while ($\|\mathbf{g}_0\| > \epsilon \ \&\& \ k < N$)

{

$$\alpha_k = \frac{\mathbf{g}_k^T \mathbf{y}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{g}_{k+1} = \mathbf{g}_k + \alpha_k A \mathbf{p}_k$$

$$M\mathbf{y}_{k+1} = \mathbf{g}_{k+1} \text{ (Solve for } \mathbf{y}_{k+1})$$

$$\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{y}_{k+1}}{\mathbf{g}_k^T \mathbf{y}_k}$$

$$\mathbf{p}_{k+1} = -\mathbf{y}_{k+1} + \beta_{k+1} \mathbf{p}_k$$

$$k = k + 1$$

}

Para $M = I$ GC-Precondicionado es GC.

Precondicionadores.

No hay un mejor precondicionador.

Puntos recomendables:

- M eficiente, esto es: $M \approx A^{-1}$ ó $M^{-1}A \approx I$.
- Fácil de calcular (rápidamente).
- Requerimientos mínimos de almacenamiento.
- Fácil de resolver: $M\mathbf{y} = \mathbf{g}$.

La elección del precondicionador depende del problema.

$M\mathbf{y} = \mathbf{g}$ es una simplificación al problema $A\mathbf{x} = \mathbf{b}$. Por ejemplo en caso de imágenes ó ecuaciones diferenciales parciales (un problema más grueso).

Precondicionadores de tipo general.

- Symetric onerrelaxation.
- Incomplete Cholesky (tal vez el más efectivo)

$$\begin{aligned} A &= L^T L && \text{Cholesky} \\ A &\approx L^L && \text{Incompleto.} \end{aligned}$$

Nota en convergencia de GC.

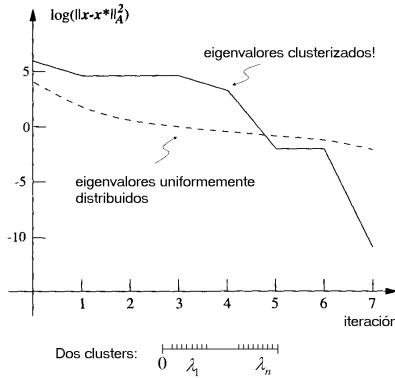


Figura 5.5: Convergencia de GC.

- Gradiente Conjugado:

$$\|\mathbf{x}_k - \mathbf{x}^*\|_A \leq \left(\frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1} \right)^{2k} \|\mathbf{x}_0 - \mathbf{x}^*\|_A$$

- Steepest descent (en fuertemente convexas cuadráticas):

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|_A^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \|\mathbf{x}_k - \mathbf{x}^*\|_A^2$$

En precondicionador con factorización de Cholesky: $A = L^T L$. Si aproximamos L por \tilde{L} (más rala que L), tenemos que $A \approx LL^T$. Eligiendo $C = \tilde{L}^T$, obtenemos $M = \tilde{L}\tilde{L}^T$ y

$$C^{-T}AC^{-1} = \tilde{L}^{-1}LL^T\tilde{L}^{-T} \approx I,$$

luego la distribución de eigenvalores de $C^{-T}AC^{-1}$ es favorable. En realidad no se calcula M explícitamente, sino que resolvemos el sistema $M\mathbf{y} = \mathbf{g}$ usando dos sustituciones triangulares de \tilde{L} .

Problema: Qué tanto se debe truncar el Cholesky? Depende del problema!!

5.2. Gradiente Conjugado No Lineal.

El algoritmo GC fue estudiado para minimizar funciones cuadráticas convexas. Ahora lo veremos en su adaptación a:

- Funciones generales convexas.
- Funciones no lineales.

Propuesta de Fletcher-Reeves (FR).

- α_k (que minimiza ϕ a lo largo de \mathbf{p}_k) necesita ser calculada con un algoritmo de búsqueda en línea.
- \mathbf{g}_k es el gradiente de ϕ en la iteración k .

Algoritmo 5.4. FR-GC (No Lineal)

Dada \mathbf{x}_0 ;

Evaluate $f_0 = f(\mathbf{x}_0)$; $\nabla f_0 = \nabla f(\mathbf{x}_0)$;

Set $\mathbf{p}_0 = -\nabla f_0$, $k = 0$;

while ($\|\nabla \mathbf{f}_k\| > \epsilon$)

{

 Compute α_k as the minimum of $\phi(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$

 Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$

 Evaluate ∇f_{k+1} (ó \mathbf{g}_{k+1})

$$\beta_{k+1}^{FR} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$$

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \beta_{k+1}^{FR} \mathbf{p}_k$$

$$k = k + 1$$

}

Si f es fuertemente convexa y cuadrática y α_k es el minimizador exacto, el algoritmo se reduce al GC-lineal.

Nota: En la elección de α_k , dado que $\mathbf{p}_k = -\mathbf{g}_k + \beta_k^{FR} \mathbf{p}_{k-1}$, luego

$$\mathbf{g}_k^T \mathbf{p}_k = -\|\mathbf{g}_k\|^2 + \beta_k^{FR} \mathbf{g}_k^T \mathbf{p}_{k-1} \quad (5.14)$$

Note que si α_k es el minimizador exacto $\mathbf{g}_k^T \mathbf{p}_{k-1} = 0$ y tenemos que $\mathbf{g}_k^T \mathbf{p}_k < 0$, luego \mathbf{p}_k es una dirección de descenso, pero si la búsqueda en línea no es exacta el segundo término en (5.14) puede dominar y hacer que $\mathbf{g}_k^T \mathbf{p}_k > 0$, lo que implica que \mathbf{p}_k no sea una dirección de descenso. Esto se resuelve imponiendo las condiciones fuertes de Wolfe.

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^T \mathbf{p}_k \quad (5.15)$$

$$|\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k| \leq c_2 |\nabla f_k^T \mathbf{p}_k| \quad (5.16)$$

donde $0 < c_1 < c_2 < \frac{1}{2}$.

“Cualquier procedimiento de búsqueda en línea que da una α_k que satisface las condiciones fuertes de Wolfe asegura que \mathbf{p}_k sea una dirección de descenso de f ”.

El GC de Polak Ribiére. (No lineal).

Es una variante de GC-FR que difiere (como muchos otros) de la forma en que se elige β_k :

$$\beta_k^{PR} = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^T \mathbf{g}_k}. \quad (5.17)$$

En el algoritmo 5.3, si f es fuertemente cuadrática convexa, $\mathbf{g}_{k+1}^T \mathbf{g}_k = 0$ y se tiene:

$$\beta_k^{PR} = \beta_k^{FR}.$$

¡(5.17) y las condiciones de Wolfe (5.15 y 5.16) no garantizan que \mathbf{p}_k sea una dirección de descenso!. Luego, si definimos el algoritmo GC-PR⁺, con:

$$\beta_k^+ = \max \{ \beta_{k-1}^{PR}, 0 \},$$

ahora si, con las condiciones de Wolfe (5.15 y 5.16) \mathbf{p}_k será una dirección de descenso.

La propuesta de GC-Hestene-Stiefel (GC-HS) es:

$$\beta_{k+1}^{HS} = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{(\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{p}_k},$$

da un algoritmo con propiedades teóricas y prácticas de convergencia similares al GC-PR.

Reinicios.

Una forma de “resetear” la memoria de GC es reiniciarlo cada n iteraciones ó cuando dos gradientes consecutivos sean muy lejanos de la ortogonalidad, es decir: $\frac{|\mathbf{g}_k^T \mathbf{g}_{k-1}|}{\mathbf{g}_k^T \mathbf{g}_k} \geq \nu$ (típicamente $\nu = 0.1$).

NUMERICAL PERFORMANCE

Table 5.1 illustrates the performance of Algorithms FR-CG, PR-CG, and PR+ without restarts. For these tests, the parameters in the strong Wolfe conditions (5.42) were chosen to be $c_1 = 10^{-4}$ and $c_2 = 0.1$. The iterations were terminated when

$$\|\nabla f_k\|_\infty < 10^{-5}(1 + |f_k|),$$

or after 10,000 iterations (the latter is denoted by a *).

The final column, headed “mod,” indicates the number of iterations of Algorithm PR+ for which the adjustment (5.44) was needed to ensure that $\beta_k^{*k} \geq 0$. An examination of the results of Algorithm FR-CG on problem GENROS shows that the method takes very short steps far from the solution that lead to tiny improvements in the objective function.

The Polak–Ribi  re algorithm, or its variation PR+, are not *always* more efficient than Algorithm FR-CG, and it has the slight disadvantage of requiring one more vector of storage. Nevertheless, we recommend that users choose Algorithm PR-CG or PR+ whenever possible.

BEHAVIOR OF THE FLETCHER–REEVES METHOD

We now investigate the Fletcher–Reeves algorithm, Algorithm 5.4, a little more closely, proving that it is globally convergent and explaining some of its observed inefficiencies.

The following result gives conditions on the line search that guarantee that all search directions are descent directions. It assumes that the level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ is

Table 5.1 Iterations and function/gradient evaluations required by three nonlinear conjugate gradient methods on a set of test problems.

Problem	<i>n</i>	Alg FR it/f-g	Alg PR it/f-g	Alg PR+ it/f-g	mod
CALCVAR3	200	2808/5617	2631/5263	2631/5263	0
GENROS	500	*	1068/2151	1067/2149	1
XPOWSING	1000	533/1102	212/473	97/229	3
TRIDIA1	1000	264/531	262/527	262/527	0
MSQRT1	1000	422/849	113/231	113/231	0
XPOWELL	1000	568/1175	212/473	97/229	3
TRIGON	1000	231/467	40/92	40/92	0

Problema: Minimizar $f U(f(x, y))$, donde:

$$U(f(x, y)) = \sum_{(x,y)} (f(x, y) - g(x, y))^2 + \lambda \left[\sum_{(x,y),(x-1,y)} (f(x, y) - f(x-1, y))^2 + \sum_{(x,y),(x,y-1)} (f(x, y) - f(x, y-1))^2 \right].$$

$$\text{DFT}\{f(x-a, y)\} = F(u, v)e^{-\frac{j2\pi ua}{N}}$$

Tarea: Resolver con GC-PR (16x16)

$$U(\theta) = \sum_{\mathbf{x}} (f_1(\mathbf{x}) - f_2(\mathbf{x} + \mathbf{X}\theta))^2.$$

Para esto, aproxime $\frac{\partial g(x)}{\partial x} \approx g(x) - g(x-1)$. Por ejemplo:

$$\frac{\partial}{\partial \theta_1} f_2(\mathbf{x} + \mathbf{X}\theta) = f_2(\mathbf{x} + \mathbf{X}\theta) - f_2 \left(\mathbf{x} + \mathbf{X} \begin{pmatrix} \theta_1 - 1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{pmatrix} \right).$$

Probar con $\delta = 0.001, 0.1$.

Capítulo 6

Introducción al Cálculo Variacional.

- El Cálculo Variacional nos muestra como encontrar extremos de funciones.
- El extremo de expresiones que dependen de funciones más que de parámetros.
- Estas expresiones se denominan funcionales.
- Los extremos están condicionados por ecuaciones diferenciales en vez de por ecuaciones algebraicas ordinarias.

6.1. Problema sin restricciones.

Considere la integral simple:

$$I = \int_{x_1}^{x_2} F(x, f, f') dx.$$

Aquí F depende de la función desconocida f y su derivada f' .

Asumamos que f debe pasar por los puntos $f_1 = f(x_1)$ y $f_2 = f(x_2)$, ahora suponga que $f(x)$ es la solución del problema extremo (mínimo o máximo).

Esperamos que una pequeña variación en f^* no cambie la integral significativamente.

Ahora usamos la función de prueba $\eta(x)$. Si agregamos $\epsilon\eta(x)$ a $f(x)$, la integral cambiará en un monto proporcional a ϵ^2 para pequeños valores de ϵ . Si variamos linealmente ϵ , incrementaremos o decrementaremos el valor de la integral y de esta manera no será un punto extremo.

Nosotros buscamos $\frac{\partial I}{\partial \epsilon}|_{\epsilon=0} = 0$ (extremo), lo cual es cierto para cualquier función de prueba η .

En nuestro problema tenemos $\eta(x_1) = 0$ y $\eta(x_2) = 0$ para satisfacer las condiciones de frontera. Si reemplazamos $f(x)$ por $f(x) + \epsilon\eta(x)$, y $f'(x)$ por $f'(x) + \epsilon\eta'(x)$, la integral se convierte:

$$I = \int_{x_1}^{x_2} F(x, f + \epsilon\eta, f' + \epsilon\eta') dx.$$

Si F es diferenciable expandemos por series de Taylor:

$$F(x, f + \epsilon\eta, f' + \epsilon\eta') = F(x, f, f') + \epsilon \frac{\partial}{\partial f} F(x, f, f')\eta + \epsilon \frac{\partial}{\partial f'} F(x, f, f')\eta' + e,$$

donde e contiene los términos de altas potencias, luego

$$I = \int_{x_1}^{x_2} (F + \epsilon\eta(x)F_f + \epsilon\eta'F_{f'} + e) dx,$$

diferenciando con respecto a ϵ e igualando a cero:

$$\int_{x_1}^{x_2} (\eta F_f + \eta' F_{f'}) dx = 0.$$

Usando integración por partes ($\int f(x)g(x)dx = F(x)g(x) - \int F(x)g'(x)dx$), vemos que:

$$\int_{x_1}^{x_2} \eta' F_{f'} dx = [\eta(x)F_{f'}]_{x_1}^{x_2} - \int_{x_1}^{x_2} \eta(x) \frac{d}{dx} F_{f'} dx.$$

Dadas las condiciones de frontera, el primer término es cero, luego la integral se convierte en:

$$\int_{x_1}^{x_2} \eta(x)(F_f - \frac{d}{dx} F_{f'}) dx + [\eta(x)F_{f'}]_{x_1}^{x_2}.$$

Lo cual es cierto para todas las funciones de prueba $\eta(x)$, entonces la forma de conseguir el cero es:

$$F_f - \frac{d}{dx} F_{f'} = 0.$$

Esta es llamada la ecuación de Euler para el problema.

Generalizaciones.

La condición natural de frontera (Newman) $F_{f'} = 0$ en $x = x_1$ y $x = x_2$ también resulta en la misma ecuación de Euler.

$$I = \int \int_D F(x, y, f, f_x, f_y) dx dy.$$

Usando condiciones de frontera de Newman:

$$F_f - \frac{\partial}{\partial x} F_{f_x} - \frac{\partial}{\partial y} F_{f_y} = 0$$

La ecuación de Euler resulta en una ecuación diferencial parcial.

Problema a resolver con GC-Precondicionado.

Usando como precondicionador un esquema multigrud con gradiente conjugado precondicionado y el precondicionador un GC (ya no precondicionado).

$$U(f) = \int_x \int_y \{(f(x,y) - g(x,y))^2 + \lambda |\nabla f(x,y)|^2\} dx dy. \quad \dots(1)$$

La ecuación de Euler correspondiente es:

$$f - g - \lambda \nabla^2 f = 0. \quad \dots(2)$$

Aproximando por primeras diferencias:

$$\frac{\partial}{\partial x} f(x,y) = \lim_{h \rightarrow 0} \frac{f(x,y) - f(x-h,y)}{h} \approx f(x,y) - f(x-1,h) \stackrel{def}{=} \Delta_x^- f.$$

Usando $\Delta_x^+ f - \Delta_x^- f = f(x+1,y) - 2f(x,y) + f(x-1,y)$ (diferencias adelantadas y atrasadas), tenemos:

$$\frac{\partial^2}{\partial x^2} f(x,y) = \lim_{h \rightarrow 0} \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h},$$

luego (2) se puede escribir como:

$$\begin{aligned} & f(x,y) + \lambda[-f(x+1,y) + 2f(x,y) - f(x-1,y)]_{(x+1,y),(x,y),(x-1,y) \in L} \leftarrow \text{cond. de frontera} \\ & + \lambda[-f(x,y+1) + 2f(x,y) - f(x,y-1)]_{(x,y+1),(x,y),(x,y-1) \in L} \leftarrow \text{cond. de frontera} \\ & = g(x,y). \end{aligned}$$

En una dimensión esto corresponde al sistema lineal

$$\begin{bmatrix} 1 & -\lambda & & & & \\ -\lambda & 1+2\lambda & -\lambda & & & \\ & -\lambda & 1+2\lambda & -\lambda & & \\ & & \ddots & & & \\ & & & -\lambda & 1 & \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(n-2) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(n-2) \\ g(n-1) \end{bmatrix}$$

El cual es un sistema lineal $Af = g$. (Las componentes $A_{0,0}$ y $A_{n-1,n-1}$ representan las condiciones de frontera).

Ahora discretizando desde un inicio la funcional:

$$U(f) = \sum_{x,y} [f(x,y) - g(x,y)]^2 + \lambda \sum_{stackrel{(x,y)(x-1,y) \in L}{}} (f(x,y) - f(x-1,y))^2 \\ + \lambda \sum_{stackrel{(x,y)(x,y-1) \in L}{}} (f(x,y) - f(x,y-1))^2.$$

Obtenemos de $\frac{\partial U(f)}{\partial f(x,y)} = 0$:

$$f(x,y) - g(x,y) + \lambda(f(x,y) - f(x-1,y))(1)_{(x,y),(x-1,y) \in L} \\ + \lambda(f(x+1,y) - f(x,y))(-1)_{(x,y),(x+1,y) \in L} \\ + \lambda(f(x,y) - f(x,y-1))(1)_{(x,y),(x,y-1) \in L} \\ + \lambda(f(x,y+1) - f(x,y))(-1)_{(x,y),(x,y+1) \in L} \\ = 0.$$

¡Impusimos las condiciones de frontera al discretizar!.

Capítulo 7

Métodos de Newton Prácticos.

Recordando el método de Newton, la fórmula de recurrencia es:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad // \alpha_k = 1,$$

luego, para calcular la dirección de descenso \mathbf{p}_k :

$$f(\mathbf{x}_k + \mathbf{p}_k) \approx f_k + \mathbf{p}_k^T \nabla f_k + \frac{1}{2} \mathbf{p}_k^T \nabla^2 f_k \mathbf{p}_k \stackrel{\text{def}}{=} m_k(\mathbf{p}),$$

donde $f_k \stackrel{\text{def}}{=} f(\mathbf{x}_k)$

Luego, haciendo $\frac{\partial}{\partial \mathbf{p}} m_k(\mathbf{p}) = 0$, es decir:

$$\nabla f_k + \nabla^2 \mathbf{p} = 0,$$

luego, el paso de Newton \mathbf{p}^N resuelve el sistema

$$\nabla^2 f_k \mathbf{p}_k^N = -\nabla f_k. \quad (7.1)$$

Note que si el Hessiano $\nabla^2 f_k$ es definido positivo, podemos usar un método iterativo (GC) para resolver \mathbf{p}_k^N . En otro caso, si $\nabla^2 f_k$ no es positivo definido o \mathbf{x}_k es cercano a un punto singular, \mathbf{p}_k^N puede ser una dirección de ascenso o ser excesivamente largo.

Hay dos estrategias para resolver esto:

- La primera estrategia es resolver (7.1) con GC y terminar si una curvatura negativa es encontrada, usando:

$$\begin{cases} -\text{Búsqueda en línea.} \\ -\text{Región verdadera (válida o de confianza)} \end{cases}$$

Este método es denominado Newton-GC.

- La segunda estrategia es modificar la matriz Hessiana $\nabla^2 f_k$ antes o durante la solución al sistema (7.1) de tal forma que garanticemos sea positiva definida. Este método se denomina Newton modificado.

Otros puntos referentes al método de Newton:

- Si el sistema (7.1) no es resuelto exactamente, se calcula una $\mathbf{p}_k \approx \mathbf{p}_k^N$ y se denomina búsqueda inexacta.
- Generalmente el mayor problema es calcular la matriz Hessiana $\nabla^2 f_k$, si no es disponible en su forma analítica es posible usar técnicas de diferenciación automáticas.

$$\text{Ingredientes} \quad \left\{ \begin{array}{l} - \text{Explotar la esparcidad (ralez).} \\ - \text{Búsqueda inexacta.} - \text{Técnicas de diferenciación.} \end{array} \right.$$

Los métodos de Newton son los más eficientes y poderosos para resolver problemas sin restricciones ya sea grandes o pequeños.

7.1. Newton con pasos inexactos.

Cuándo detener las iteraciones?. Si nos basamos en el residual

$$\mathbf{r}_k = \nabla^2 f_k \mathbf{p}_k + \nabla f_k, \quad (7.2)$$

donde \mathbf{p}_k es el paso inexacto. Note que (7.2) es sensible a la escala, luego nosotros terminamos la iteración del “solver” cuando

$$\|\mathbf{r}_k\| \leq \eta_k \|\nabla f_k\|, \quad (7.3)$$

(para reducir el efecto de la escala) donde la secuencia $\{\eta_k\}$ (con $0 < \eta_k < 1$ para toda k) es denominada la secuencia de force.

La convergencia de los métodos de Newton depende de la secuencia de force elegida (ver libro Nocedal). Para obtener convergencia:

- superlineal $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$
- cuadrática $\eta_k = \min(0.5, \|\nabla f_k\|)$.

Esto para cuando \mathbf{x}_k está “cerca” de \mathbf{x}^* . Recordemos que para \mathbf{x}_k lejos de \mathbf{x}^* los métodos de Newton oscilan o inclusive no convergen.

Hasta ahora hemos asumido que $\alpha_k = 1$ en

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k,$$

donde α_k es el tamaño de paso.

Podemos calcular α_k tal que corresponda a un método de búsqueda en línea o de región de confianza.

7.2. Métodos de Newton con búsqueda en línea.

El tamaño de paso α_k puede ser seleccionado tal que satisfaga

- a) Condiciones de Wolfe
- b) Condiciones de Goldstein
- c) Backtracking - Armijo.

En todos los casos, $\alpha_k = 1$ debe ser la primera opción.

Newton-GC con búsqueda en línea.

\mathbf{p}_k es calculada usando GC truncado, parándolo cuando se satisfaga (7.3) ó en cuanto una dirección de búsqueda del GC con curvatura negativa sea generada ($\nabla^2 f_k$ puede no ser positiva definida y hace fallar el GC).

Para usar GC (que resuelve sistemas de la forma $A\mathbf{x} = \mathbf{b}$), hacemos $A = \nabla^2 f_k$, $\mathbf{b} = -\nabla f_k$ y $\mathbf{x} = \mathbf{p}_k$.

Denotamos por $\{\mathbf{x}^{(i)}\}$ y $\{\mathbf{p}^{(i)}\}$ las secuencias de solución y dirección de descenso internas de la iteración de GC, usadas para encontrar \mathbf{p}_k (diferentes) y \mathbf{x}_k (la $\mathbf{x}^{(i)*}$ sera \mathbf{p}_k). No confundir con $\{\mathbf{x}_k\}$ y $\{\mathbf{p}_k\}$ del método de Newton.

Hay tres requerimientos en GC para resolver (7.1)

- (a) El punto de inicio de la iteración en GC es $\mathbf{x}^{(0)} = 0$.
- (b) Prueba de curvatura no negativa. Si la dirección $\mathbf{p}^{(i)}$ generada por GC satisface:

$$(\mathbf{p}^{(i)})^T A \mathbf{p}^{(i)} \leq 0, \quad (7.4)$$

chequemos si es la primera iteración ($i = 0$), si si, completamos la iteración (primera). Si no, detener el GC y $\mathbf{x}^{(i)}$ es el resultado \mathbf{x}^f (final).

- (c) El paso de Newton $\mathbf{p}_k = \mathbf{x}^{(f)}$ donde $\mathbf{x}^{(f)}$ es la iteración final de GC.

Si la condición (7.4) no se satisface, continuamos con nuestra iteración de GC.

Algoritmo 7.1. Método Newton-GC con búsqueda en línea.

Dado un punto inicial \mathbf{x}_0 ;

for $k = 1, 2, \dots$

Calcular la dirección de descenso \mathbf{p}_k mediante GC para $\nabla^2 f_k \mathbf{p}_k = -\nabla f_k$, con punto de inicio $\mathbf{x}^{(0)}$ y terminar GC cuando $\|\mathbf{r}_k\| \leq \min(0.5, \|\nabla f_k\|)\|\nabla f_k\|$, ó si se encuentra una curvatura en $\mathbf{p}^{(i)}$ no positiva.

Poner $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$, donde α_k satisface las condiciones de Wolfe, Goldstein ó Armijo-Backtracking.

end (for)

Este método se comporta bien para problemas grandes, sin embargo, cuando el Hessiano $\nabla^2 f_k$ es casi singular, la iteración Newton-GC puede ser excesivamente larga requiriendo muchas evaluaciones de la función.

Una mejor opción será un método de región de confianza Newton-GC.

Métodos de Newton Modificado.

- En muchas aplicaciones es deseable usar técnicas de álgebra lineal para resolver (7.1).
- Si el Hessiano no es definido positivo ó es casi singular, este puede ser modificado antes o durante el proceso de solución para asegurar que \mathbf{p}_k satisfaga idénticamente (7.1) excepto por que la matriz de coeficientes es reemplazada por una aproximación.
- El Hessiano modificado:

$$\left\{ \begin{array}{l} \text{-- Matriz diagonal positiva.} \\ \text{-- Matriz completa del verdadero Hessiano.} \end{array} \right.$$

----- *Paréntesis* -----

La información de la transformada rápida de Fourier (FFT) está en el Numerical Recipes C. La FFT es un algoritmo para calcular eficientemente la transformada discreta de Fourier y está definida para datos unidimensionales cuyo tamaño es potencia de 2.

Además la DTF impone condiciones de frontera periódicas.

$$F(u) = \mathfrak{F}\{f\} = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) = \mathfrak{F}^{-1}\{F\} = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux}du$$

$$F_u = DFT(f) = \sum_{x=0}^{N-1} f_x e^{-\frac{j2\pi ux}{N}}$$

$$f_i = IDFT(F) = \frac{1}{N} \sum_{u=0}^{N-1} F_u e^{\frac{j2\pi ux}{N}}$$

donde $j = \sqrt{-1}$.

$$f_{xy} - \lambda(f_{x+1,y} - 2f_{x,y} + f_{x-1,y}) - \lambda(f_{x,y+1} - 2f_{x,y} + f_{x,y-1}) = g_{xy} \quad \dots(1)$$

$$\begin{aligned} DFT\{f_{x+1,y}\} &= \sum_{x=0}^{N-1} \left[\sum_{y=0}^{M-1} f_{x+1,y} e^{-\frac{j2\pi vy}{N}} \right] e^{-\frac{j2\pi ux}{N}} = \sum_{x=0}^{N-1} \widehat{f}_{x+1,v} e^{-\frac{j2\pi ux}{N}} = \sum_{\widehat{x}=1}^N \widehat{f}_{\widehat{x},v} e^{-\frac{j2\pi u(\widehat{x}-1)}{N}} \\ &= \sum_{\widehat{x}=1}^N \widehat{f}_{\widehat{x},v} e^{\frac{j2\pi u}{N}} e^{-\frac{j2\pi u\widehat{x}}{N}} = e^{\frac{j2\pi u}{N}} \sum_{\widehat{x}=1}^N \widehat{f}_{\widehat{x},v} e^{-\frac{j2\pi u\widehat{x}}{N}} = e^{\frac{j2\pi u}{N}} F_{uv} \end{aligned}$$

Aplicando DFT a (1):

$$\begin{aligned} F_{u,v} - \lambda \left[F_{u,v} e^{\frac{j2\pi u}{N}} - 2F_{u,v} + F_{u,v} e^{-\frac{j2\pi u}{N}} \right] - \lambda \left[F_{u,v} e^{\frac{j2\pi v}{M}} - 2F_{u,v} + F_{u,v} e^{-\frac{j2\pi v}{M}} \right] &= G_{u,v} \\ F_{u,v} \left[1 + \lambda \left(2 - 2 \cos \left(\frac{2\pi u}{N} \right) \right) + \lambda \left(2 - 2 \cos \left(\frac{2\pi v}{M} \right) \right) \right] &= G_{u,v} \\ F_{u,v} \left[1 + 2\lambda \left(1 - \cos \left(\frac{2\pi u}{N} \right) \right) + 2\lambda \left(1 - \cos \left(\frac{2\pi v}{M} \right) \right) \right] &= G_{u,v} \end{aligned}$$

Así pues: $F_{uv} = \frac{G_{uv}}{H_{uv}}$, y podemos resolver $My = r$ haciendo:

$$\begin{cases} R = \text{FFT}(r) \\ Y = \frac{R}{H} \\ y = \text{IFFT}(Y) \end{cases}$$

-----Fin de Paréntesis-----

Algoritmo 7.2. Newton Modificado con búsqueda en línea.

Dado un punto inicial \mathbf{x}_0 ;

for $k = 1, 2, \dots$

Factorice la matriz $B_k = \nabla_f^T \mathbf{x}_k + E_k$, donde $E_k = 0$ si $\nabla^2 f(\mathbf{x}_k)$ es suficientemente definido positivo. En otro caso E_k se elige tal que B_k sea suficientemente definido positivo.

Ponga $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ donde α_k satisface las condiciones de Wolfe, Goldstein o Armijo-Backtracking.

end (for)

La elección de E_k es crucial para la eficiencia del método.

----- *Paréntesis* -----

Tarea: Problema 6.1 del Nocedal. Programe Newton-GC para

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{x} + 0.25\sigma(\mathbf{x}^T A \mathbf{x})^2$$

donde σ es un parámetro y

$$A = \begin{bmatrix} 5 & 1 & 0 & 0.5 \\ 1 & 4 & 0.5 & 0 \\ 0 & 0.5 & 3 & 0 \\ 0.5 & 0 & 0 & 2 \end{bmatrix}$$

que es estrictamente convexa.

Punto inicial: $\mathbf{x} = [\cos 70^\circ, \sin 70^\circ, \cos 70^\circ, \sin 70^\circ]^T$. Tratar $\sigma = 1$ y mayores.

----- *Fin de Paréntesis* -----

7.3. Técnicas de Modificación del Hessiano.

Modificación de Eigenvalores.

Considere la factorización (SVD)

$$\nabla^2 f_k = Q \Lambda Q^T = \sum_{i=1}^n \lambda_i \mathbf{q}_i \mathbf{q}_i^T$$

donde:

Λ = Matrix diagonal de eigenvalores = $Diag(\lambda_1, \lambda_2, \dots, \lambda_n)$,

Q son los eigenvectores por columna y

Q^T son los eigenvectores por renglón.

Una opción es tomar los eigenvalores negativos y hacerlos positivos tal que $\nabla^2 f_k$ sea ahora definido positivo.

Por ejemplo:

$$\nabla_k^f = Diag(10, 3, -1) \text{ y } \nabla f_k = (1, -3, 2).$$

Luego, el determinante de $\nabla^2 f_k$ es negativo, luego, no es definido positivo, entonces hacer:

$$E_k = \text{Diag}(0, 0, +\lambda_3 + \delta),$$

donde δ se elige como $\delta = \sqrt{u}$ con u igual a la mayor precisión de la computadora (por ejemplo: $\sqrt{u} = 10^{-8}$), luego:

$$B_k = \text{diag}(10, 3, \sqrt{u}).$$

Note que B_k es claramente definida positiva.

Añadiendo un múltiplo de la identidad.

$$E_k = \nabla^2 f_k + \tau I,$$

con τ constante. En E_k recorre todos los eigenvalores.

Por ejemplo:

$$\tau = \max(0, \delta - \lambda_{\min}(A)). \quad (7.5)$$

Asegurando que I produzca una matriz suficientemente definida positiva.

- El problema con (7.5) es que en general no tenemos una buena estimación del eigenvalor más pequeño de ∇_k^f .
- Una estrategia: el eigenvalor mayor en valor absoluto de una matriz A está acotado por la norma de Frobenius $\|A\|_F$, definida como: $\|A\|_F^2 = \sum_{ij} (a_{ij})^2$.

Cholesky con Identidad Aumentada. Sea $A \stackrel{\text{def}}{=} \nabla^2 f_k$.

Poner $\beta = \|A\|_F$
if $\min(a_{ii}) > 0$

$\tau_0 \leftarrow 0$

else

$\tau_0 \leftarrow \frac{\beta}{2}$

end (if)

for $k = 1, 2, \dots$

Usar Cholesky para factorizar en L , $LL^T = A + \tau_k I$

if La factorización fue exitosa

stop y regrese L

```

else
     $\tau_{k+1} \leftarrow \max(2\tau_k, \frac{\beta}{2})$ 
end (if)
end (for)

```

Esta estrategia es simple, pero presenta dos problemas:

- τ puede ser innecesariamente grande.
- Cada iteración de τ_k requiere una factorización numérica completa de $A + \tau_k I$

Factorización de Cholesky Modificada.

- Garantiza que los factores de Cholesky existan y con norma cercana al Hessiano actual.
- No modifica el Hessiano si es suficientemente definido positivo.

Sea una matriz A simétrica y positiva definida, luego, $A = LDL^T$ donde L =triangular inferior con unos en la diagonal y D es una matriz diagonal.

Algoritmo 7.4. Cholesky ($A = LDL^T$)

```

for  $j = 1, 2, \dots, n$ 
     $c_{jj} = a_{jj} - \sum_{s=1}^{j-1} d_s \ell_{js}^2;$ 
     $d_j = c_{jj};$ 
    for  $i = j + 1, \dots, n$ 
         $c_{ij} \leftarrow - \sum_{s=1}^{j-1} d_s \ell_{is} \ell_{js};$ 
         $\ell_{ij} \leftarrow \frac{c_{ij}}{d_j}$ 
    end (for)
end (for)

```

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} \quad D = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}$$

Nota: Si se desea la factorización estándar $A = MM^T$ hacemos $M = LD^{\frac{1}{2}}$. Si A es indefinida, $A = LDL^T$ puede no existir y el algoritmo 7.4 es numéricamente inestable porque L y D pueden convertirse arbitrariamente grandes.

El siguiente algoritmo modifica A tal que:

- 1** No sea muy diferente de la A original.
- 2** Los elementos de D y L no sean muy grandes.
- 3** Todos los elementos en D son suficientemente positivos.

Algoritmo 7.5. Cholesky Modificado (página 148)

Dados $\delta > 0$, $\beta > 0$

for $k = 1, 2, \dots, n$

$$c_{kk} = a_{kk}$$

end (for)

Encontrar el índice q tal que $|c_{qq}| \geq |c_{ii}|$

Intercambiar renglón y columna i y q para $q \neq i$

for $j = 1, 2, \dots, n$ (calcula la j -ésima columna de L)

for $s = 1, \dots, j - 1$

$$\ell_{js} \leftarrow \frac{c_{js}}{d_s};$$

end (for)

for $i = j + 1, \dots, n$

$$c_{ij} = a_{ij} - \sum_{s=1}^{j-1} \ell_{js} c_{is};$$

end (for)

$$\theta_j \leftarrow 0;$$

if $j \leq n$

$$\theta_j \leftarrow \max_{j < i \leq n} |c_{ij}|;$$

end (if)

$$d_j \leftarrow \max\{|c_{jj}|, (\frac{\theta_j}{\beta})^2, \delta\}$$

if $j < n$

for $i = j + 1, \dots, n$

$$c_{ii} = c_{ii} - \frac{c_{ij}^2}{d_j};$$

end (for)

end (if)

end (for)

Parámetros:

Sea u la precisión de la computadora, entonces

$$\delta = u \max(\gamma(A) + \xi(A), 1),$$

donde $\gamma(A)$ y $\xi(A)$ son:

- $\gamma(A) = \max_{1 \leq i \leq n} |a_{ii}| \leftarrow$ máximo en la diagonal de A.
- $\xi(A) = \max_{i \neq j} |a_{ij}| \leftarrow$ máximo fuera de la diagonal de A.

Para β (sugieren Gill, Murray & Wright)

$$\beta = \max \left(\gamma(A), \frac{\xi(A)}{\sqrt{n^2 - 1}}, u \right)^{\frac{1}{2}}.$$

Otras modificaciones (ver Nocedal):

- Gershgorin's
- Factorización simétrica indefinida.

7.4. Métodos de Newton de Región Válida.

A diferencia de los métodos de búsquedas, no se requiere que el Hessiano sea definido positivo.

Podemos usar directamente $b_k = \nabla^2 f_k$ para resolver el modelo:

$$\min_{\mathbf{p} \in \mathbb{R}^n} m_k(\mathbf{p}) \stackrel{def}{=} f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_k \mathbf{p} \quad \text{S.A. } \|\mathbf{p}\| \leq \Delta_k$$

Si $\nabla^2 f_k$ es no definido positivo, podemos usar una modificación (ver lo anterior) que si lo sea (B_k).

El paso de dogleg está (como antes) dado por:

$$\tilde{\mathbf{p}}(\tau) = \begin{cases} \tau \mathbf{p}^U & 0 \leq \tau \leq 1 \\ \mathbf{p}^U + (\tau - 1)(\mathbf{p}^B - \mathbf{p}^U) & 1 \leq \tau \leq 2 \end{cases},$$

y B_k una aproximación del Hessiano $\nabla^2 f_k$, luego:

$$\min_{\mathbf{p}} m_k(\mathbf{p}) = f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_k \mathbf{p} \quad \text{S.A. } \|\mathbf{p}\| \leq \Delta_k,$$

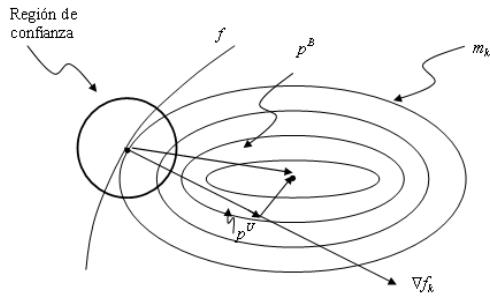


Figura 7.1: Paso de dogleg.

donde $\mathbf{p} \in \text{span}\{\nabla f_k, \mathbf{p}^B\}$ ($\text{span}\{\nabla f_k, \mathbf{p}^B\}$ es el subespacio que contiene todas las combinaciones lineales de ∇f_k y \mathbf{p}^B).

----- *Paréntesis -----*

Definición:

Subespacio de Krylov de grado k para \mathbf{r}_0 :

$$\mathcal{K}(\mathbf{r}_0; k) \stackrel{\text{def}}{=} \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^k \mathbf{r}_0\},$$

donde $\text{span}\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_k\}$ es el conjunto de todas las combinaciones lineales de $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_k$.

----- *Fin de Paréntesis -----*

Capítulo 8

Cálculo Numérico de Derivadas.

Muchas veces las derivadas de funciones no pueden ser calculadas analíticamente o son muy difíciles. Algunas técnicas para calcularlas son:

- Diferenciación Finita.
- Diferenciación Automática.
- Diferenciación simbólica.

Diferenciación Finita:

Tiene sus raíces en la serie de Taylor.

$$\frac{\partial f}{\partial x_i} \approx \frac{f(\mathbf{x} + \epsilon \mathbf{e}_i) - f(\mathbf{x} - \epsilon \mathbf{e}_i)}{2\epsilon},$$

donde ϵ es un escalar pequeño y \mathbf{e}_i es el i -ésimo vector unitario. Por ejemplo, en \mathbb{R}^3 , $\mathbf{e}_1 = [1, 0, 0]^T$.

Diferenciación Automática:

Descompone el código para evaluar la función en operaciones aritméticas elementales a las cuales le aplica la regla de la cadena.

Diferenciación Simbólica:

Manipulación de la expresión algebraica de la función para producir una nueva expresión algebraica de la derivada.

8.1. Aproximación por diferencia finita.

Una forma popular de diferenciación es la derivada de un lado.

$$\frac{\partial f}{\partial x_i} \approx \frac{f(\mathbf{x} + \epsilon \mathbf{e}_i) - f(\mathbf{x})}{\epsilon} \quad (8.1)$$

ó

$$\frac{\partial f}{\partial x_i} \approx \frac{f(\mathbf{x}) - f(\mathbf{x} - \epsilon \mathbf{e}_i)}{\epsilon}.$$

El gradiente se calcula aplicando la fórmula para $i = 1, 2, \dots, n$ donde $\mathbf{x} \in \mathbb{R}^n$.

Análisis de la diferenciación finita del Teorema de Taylor:

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{x}^T \nabla^2 f(\mathbf{x} + t\mathbf{p}) \mathbf{p} \quad \text{para alguna } t \in (0, 1). \quad (8.2)$$

Si decimos que $L = \max_{\mathbf{x} \in \Omega} \|\nabla^2 f(\mathbf{x})\|$, donde Ω es una región de interés, luego el último término en (8.2) está acotado por $\frac{1}{2} \|\mathbf{p}\|^2$, por lo tanto:

$$\|f(\mathbf{x} + \mathbf{p}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T \mathbf{p}\| \leq \left(\frac{L}{2} \right) \|\mathbf{p}\|^2. \quad (8.3)$$

Ahora elegimos $\mathbf{p} = \epsilon \mathbf{e}_i$; tenemos que:

$$\nabla f(\mathbf{x})^T \mathbf{p} = \epsilon \nabla f(\mathbf{x})^T \mathbf{e}_i = \epsilon \frac{\partial f}{\partial x_i}.$$

Arreglando (8.3):

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) = \frac{f(\mathbf{x} + \epsilon \mathbf{e}_i) - f(\mathbf{x})}{\epsilon} + \delta_\epsilon \quad (8.4)$$

donde $|\delta_\epsilon| \leq \left(\frac{L}{2} \right) \epsilon$.

Note que (8.1) ignora el término de error δ_ϵ . Este término se vuelve más pequeño en tanto ϵ tiende a cero.

¡Importante la elección del parámetro ϵ ! debe ser lo más pequeño posible.

No estamos considerando errores numéricos aún (precisión de la computadora).

Es decir, la “pequeñez” de ϵ tiene un límite inferior determinado por la precisión de la computadora, en la cual se calcula la derivada numérica.

Consideremos a u la precisión de la computadora (es decir, $u = 10^{-16}$), luego sea L_f el valor máximo que toma $|f|$ en la región de interés y $\text{comp}(\mathbf{x})$ el valor calculado de \mathbf{x} :

$$\begin{aligned} |\text{comp}(f(\mathbf{x})) - f(\mathbf{x})| &\leq uL_f \\ |\text{comp}(f(\mathbf{x} + \epsilon\mathbf{e}_i)) - f(\mathbf{x} + \epsilon\mathbf{x}_i)| &\leq uL_f. \end{aligned}$$

Si usamos el valor calculado de f en (8.4) en vez de su valor exacto, tenemos que el error estará aconditado por:

$$\text{comp}(\delta_\epsilon) = \left(\frac{L}{2}\right)\epsilon + 2u\frac{L_f}{\epsilon}.$$

Note que el término $\left(\frac{L}{2}\right)\epsilon$ de la ecuación anterior es pequeño para ϵ chicas, mientras que el término $2u\frac{L_f}{\epsilon}$ crece con ϵ chicas.

Buscamos que ϵ haga este error lo más pequeño posible, esto es:

$$\epsilon^2 = \frac{4L_f u}{L},$$

donde L_f/L es la razón del valor de la función y sus segundas derivadas (su “curvatura”); asumimos que toma un valor moderado (modesto), es decir, es una función suave.

Concluimos que la ϵ cercana óptimamente es $\epsilon = \sqrt{u}$ que es el valor que se usa en los paquetes de optimización.

Las derivadas centrales son más precisas. Por el Teorema de Taylor:

$$\begin{aligned} f(\mathbf{x} + \epsilon\mathbf{e}_i) &= f(\mathbf{x}) + \epsilon\frac{\partial f}{\partial x_i} + \frac{1}{2}\epsilon^2\frac{\partial^2 f}{\partial x_i^2} + O(\epsilon^3) \\ f(\mathbf{x} - \epsilon\mathbf{e}_i) &= f(\mathbf{x}) - \epsilon\frac{\partial f}{\partial x_i} + \frac{1}{2}\epsilon^2\frac{\partial^2 f}{\partial x_i^2} + O(\epsilon^3). \end{aligned}$$

Restando ambas expresiones (note que las $O(\epsilon^3)$ no son iguales) y dividiendo por 2ϵ :

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) = \frac{f(\mathbf{x} + \epsilon\mathbf{e}_i) - f(\mathbf{x} - \epsilon\mathbf{e}_i)}{2\epsilon} + O(\epsilon^2).$$

Obviamente $O(\epsilon^2)$ es menor que $O(\epsilon)$ (¡para ϵ pequeñas es menor que 1!).

Introducción a mínimos cuadrados no lineales.

Considera la forma (mínimos cuadrados no lineales):

$$f(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^m r_j^2(\mathbf{x}) m,$$

donde r_j es una función suave de \Re^n a \Re . Denominamos r_j el residual y asumimos que $m \geq n$.

Es decir: sea $\mathbf{r}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_m(\mathbf{x}))^T$ una función vectorial, luego

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{r}^T(\mathbf{x}) \mathbf{r}(\mathbf{x})$$

y su Jacobiano (gradiente)

$$\nabla f(x) = \mathbf{r}^T(\mathbf{x}) \nabla \mathbf{r}(x),$$

luego el Jacobiano de \mathbf{r} , $J = \nabla \mathbf{r}(\mathbf{x})$!

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial r_j}{\partial x_i} \\ \vdots \\ \end{bmatrix}_{j=1, \dots, m} \quad i=1, \dots, n$$

Por el Teorema de Taylor

$$\mathbf{r}(\mathbf{x} + \mathbf{p}) = \mathbf{r}(\mathbf{x}) + \nabla \mathbf{r}^T \mathbf{x} \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 \mathbf{r}(\mathbf{x} + t\mathbf{p}) \mathbf{p} \quad \text{para alguna } t \in (0, 1),$$

luego

$$\|\mathbf{r}(\mathbf{x} + \mathbf{p}) - \mathbf{r}(\mathbf{x}) - J(\mathbf{x}) \mathbf{p}\| \leq \frac{L}{2} \|\mathbf{p}\|^2,$$

donde L es la constante de Lipschitz para J en la región de interés, luego el producto Jacobiano, J por un vector \mathbf{p} puede aproximarse por:

$$J(\mathbf{x}) \mathbf{p} \approx \frac{\mathbf{r}(\mathbf{x} + \epsilon \mathbf{p}) - \mathbf{r}(\mathbf{x})}{\epsilon},$$

eliendo una ϵ pequeña, con precisión $O(\epsilon)$.

Ahora, si lo que queremos es una aproximación completa del Jacobiano $J(\mathbf{x})$, podemos calcular una columna a la vez haciendo $\mathbf{p} = \mathbf{e}_i$. La i -ésima columna estará dada por:

$$\frac{\partial \mathbf{r}}{\partial x_i}(\mathbf{x}) \approx \frac{\mathbf{r}(\mathbf{x} + \epsilon \mathbf{e}_i) - \mathbf{r}(\mathbf{x})}{\epsilon}.$$

El Jacobiano puede ser, entonces, estimado mediante $n+1$ evaluaciones de la función \mathbf{r} .

Tarea: GC-Membrana con el producto Ax aproximado numéricamente.

Aproximación del Jacobiano de funciones objetivo ralas.

Sea $U(\mathbf{f}) = \sum_{(i,i-1) \in L} (f_i - f_{i-1})^2$, luego

$$\begin{aligned} \frac{\partial U(\mathbf{f})}{\partial f_0} &= \frac{U(\mathbf{f} + \epsilon \mathbf{e}_0) - U(\mathbf{f})}{\epsilon} = \frac{(f_1 - f_0 - \epsilon)^2 + \sum_{(i,i-1) \in L: i-1 \neq 0} (f_i - f_{i-1})^2}{\epsilon} - \dots \\ &\quad - \frac{(f_1 - f_0)^2 + \sum_{(i,i-1) \in L: i-1 \neq 0} (f_i - f_{i-1})^2}{\epsilon} = \frac{(f_1 - f_0 - \epsilon)^2 - (f_1 - f_0)^2}{\epsilon}. \end{aligned}$$

Para $k \neq 0, N-1$:

$$\frac{\partial U(\mathbf{f})}{\partial f_k} = \frac{(f_k - f_{k-1} + \epsilon)^2 + (f_{k+1} - f_k - \epsilon)^2}{\epsilon} - \dots - \frac{(f_k - f_{k-1})^2 - (f_{k+1} - f_k)^2}{\epsilon}.$$

Aproximación del Hessiano $\nabla^2 f(\mathbf{x})$.

Cuando las segundas derivadas existen y son Lipschitz continuas, el Teorema de Taylor implica que:

$$\nabla f(\mathbf{x} + \mathbf{p}) = \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})\mathbf{p} + O(\|\mathbf{p}\|^2).$$

Sustituyendo $\mathbf{p} = \epsilon \mathbf{e}_i$:

$$\nabla^2 f(\mathbf{x})\mathbf{e}_i = \frac{\partial(\nabla f)}{\partial x_i}(\mathbf{x}) \approx \frac{\nabla f(\mathbf{x} + \epsilon \mathbf{e}_i) + \nabla f(\mathbf{x})}{\epsilon},$$

con aproximación $O(\epsilon)$, asumiendo que se cuenta con $\nabla f(\mathbf{x})$ o una aproximación, tenemos la aproximación de un renglón del Hessiano \tilde{H}_i . Podemos usar también diferencias centrales (con $O(\epsilon^2)$) con el costo adicional de calcular $\nabla f(\mathbf{x} - \epsilon \mathbf{e}_i)$.

Debido a la aproximación, \tilde{H} puede no ser exactamente simétrica, esto lo podemos resolver haciendo:

$$H \leftarrow \frac{\tilde{H}^T + \tilde{H}}{2}.$$

El Algoritmo Newton-GC no requiere conocimiento del Hessiano, sino del producto $\nabla^2 f(\mathbf{x})\mathbf{p}$ para un vector dado \mathbf{p} , luego:

$$\nabla^2 f(\mathbf{x})\mathbf{p} \approx \frac{\nabla f(\mathbf{x} + \epsilon \mathbf{p}) - \nabla f(\mathbf{x})}{\epsilon},$$

con aproximación $O(\epsilon)$.

Si los gradientes no están disponibles, podemos usar el Teorema de Taylor de nuevo para derivar la fórmula de aproximación del Hessiano.

Usando la fórmula (7.8):

$$\begin{aligned} f(\mathbf{x} + \mathbf{p}) &= f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{x} + t\mathbf{p}) \mathbf{p} \quad \text{para alguna } t \in (0, 1) \\ &= f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{x}) \mathbf{p} + O(\|\mathbf{p}\|^3), \end{aligned}$$

y sustituyendo $\mathbf{p} = \epsilon \mathbf{e}_i$, $\mathbf{p} = \epsilon \mathbf{e}_j$ y $\mathbf{p} = \epsilon(\mathbf{e}_i + \mathbf{e}_j)$ en esa fórmula y combinando los resultados apropiadamente:

$$\frac{\partial f}{\partial x_i \partial x_j}(\mathbf{x}) = \frac{f(\mathbf{x} + \epsilon \mathbf{e}_i + \epsilon \mathbf{e}_j) - f(\mathbf{x} + \epsilon \mathbf{e}_i) - f(\mathbf{x} + \epsilon \mathbf{e}_j) + f(\mathbf{x})}{\epsilon^2} + O(\epsilon).$$

Capítulo 9

Métodos Quasi - Newton.

- Propuestos por Davison en los 1950's.
- Publicado (aceptado) hasta 1991 SIAM Journal on Optimization.
- Estudiado y analizado por Broyden, Fletcher, Goldfar & Powell. Analizan y estudian la propiedad de convergencia.

9.1. El método BFGS.

El método Quasi-Newton más popular es el BFGS (Broyden, Fletcher, Goldfar, Shano).

Considere el modelo cuadrático:

$$m_k(\mathbf{p}) = f_k + \nabla f_k^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_k \mathbf{p},$$

donde $B_k \in \Re^{n \times n}$ es una matriz simétrica definida positiva. B_k requiere ser actualizada o recalculada en cada iteración.

La dirección de búsqueda se encuentra resolviendo:

$$\mathbf{p}_k = -B_k^{-1} \nabla f_k,$$

donde α_k satisface Wolfe ó Goldstein.

A diferencia de los métodos de Newton, B_k es una aproximación del verdadero Hessiano para toda k .

Davidon $\rightarrow \begin{cases} - \text{No recalcula desde cero } B_k \text{ en cada iteración.} \\ - \text{Aprovecha } B_{k-1} \text{ y la actualiza eficientemente.} \end{cases}$

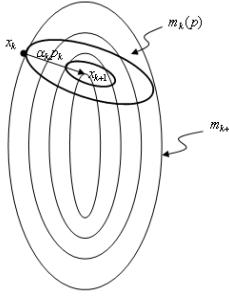


Figura 9.1: Construcción de nuevo modelo.

Construir un nuevo modelo en \mathbf{x}_{k+1} basado en B_{k+1} .

$$m_{k+1}(\mathbf{p}) = f_{k+1} + \nabla f_{k+1}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_{k+1} \mathbf{p},$$

Con dos restricciones:

- i) el gradiente (pendiente) de m_{k+1} en \mathbf{x}_{k+1} sea igual a ∇f_{k+1}
- ii) el gradiente de m_{k+1} en \mathbf{x}_k sea igual a ∇f_k .

¡La primera condición es gratis! $\nabla m_{k+1}(0) = \nabla f_{k+1}$. De la segunda condición tenemos:

$$\nabla_{\mathbf{p}} m_{k+1}(\mathbf{p}) = \nabla f_{k+1} + B_{k+1} \mathbf{p}.$$

Evaluándolo en \mathbf{x}_k , el punto anterior $(-\alpha_k \mathbf{p}_k)$, tenemos que:

$$\nabla_{\mathbf{p}} m_{k+1}(-\alpha_k \mathbf{p}_k) = \nabla f_{k+1} - \alpha_k B_{k+1} \mathbf{p}_k = \nabla f_k.$$

Rearreglando tenemos:

$$B_{k+1} \alpha_k \mathbf{p}_k = \nabla f_{k+1} - \nabla f_k.$$

Definiendo:

$$\begin{aligned} \mathbf{y}_k &= \nabla f_{k+1} - \nabla f_k \\ \mathbf{s}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{p}_k, \end{aligned}$$

tenemos:

$$B_{k+1} \mathbf{s}_k = \mathbf{y}_k \leftarrow \text{Ecuación secante!}.$$

Lo que nos dice es que el Hessiano aproximado B_{k+1} debe mapear \mathbf{s}_k en \mathbf{y}_k (restricción). Multiplicando ambos lados por \mathbf{s}_k^T :

$$\mathbf{s}_k^T B_{k+1} \mathbf{s}_k = \mathbf{s}_k^T \mathbf{y}_k,$$

el lado izquierdo es > 0 si B_{k+1} es simétrica definida positiva y luego

$$\mathbf{s}_k^T \mathbf{y}_k > 0, \quad (9.1)$$

que es la condición de curvatura.

Nota: (9.1) es una restricción, si la función es fuertemente convexa es fácil de satisfacer, si no, es necesario forzarla explícitamente usando condiciones de Wolfe en el cálculo de α .

Si hacemos esto:

$$\mathbf{x}_k^T \mathbf{s}_k \geq (c_2 - 1)\alpha_k \nabla f_k^T \mathbf{p}_k,$$

donde $c_2 < 1$ y $(c_2 - 1) < 1$ y $\nabla f_k^T \mathbf{p}_k < 1$ dado que \mathbf{p}_k es una dirección de descenso.

Si se satisface la condición de curvatura, luego la ecuación secante impone n restricciones, pero B_k (simétrica) tiene $n(n + 1)/2$ grados de libertad.

El requerimiento de definida positiva impone n restricciones adicionales, pero $n(n + 1)/2 > 2n$, luego Davidon propuso que B_{k+1} sea parecida a la anterior B_k como restricción!

Es decir:

$$\min_B \|B - B_k\| \text{ Sujeto a } B = B^T, B\mathbf{s}_k = \mathbf{y}_k. \quad (9.2)$$

La norma usada (de las muchas posibles para matrices) fue la norma pesada de Frobenius:

$$\|A\|_W = \|W^{1/2} A W^{1/2}\|_F,$$

donde

$$\|C\|_F = \sum_i \sum_j c_{ij}^2,$$

donde W es una matriz cualquiera que satisface $W^{1/2} \mathbf{s}_k = \mathbf{y}_k$.

Se usa como peso (W) el Hessiano promedio desde \mathbf{x}_k a \mathbf{x}_{k+1} ,

$$W = \bar{G}_k = \left[\int_0^1 \nabla^2 f(\mathbf{x}_k + \tau \alpha_k \mathbf{p}_k) d\tau \right].$$

Cada uno de los Hessianos satisface $\nabla^2 f_k \mathbf{s}_k = \mathbf{y}_k$.

La solución a (9.2) encontrada por Davidon es:

$$B_{k+1} = (I - \gamma_k \mathbf{y}_k \mathbf{s}_k^T) B_k (I - \gamma_k \mathbf{s}_k \mathbf{y}_k^T) + \gamma_k \mathbf{y}_k \mathbf{y}_k^T,$$

con

$$\gamma_k = \frac{1}{\mathbf{y}_k \mathbf{s}_k}.$$

Nota: Fórmula de Morrison-Woodbury.

Sea la matriz $\bar{A} = A + ab^T$, con $a, b \in \mathbb{R}^n$ y no singular, luego:

$$\bar{A}^{-1} = A^{-1} - \frac{A^{-1}ab^TA^{-1}}{1 + b^TA^{-1}b}.$$

También sean $U, V \in \mathbb{R}^{e^{n \times p}}$ para $p \in [0, n]$, y sea

$$\hat{A} = A + UV^T,$$

luego

$$\hat{A}^{-1} = A^{-1} - A^{-1}U(I + V^{-1}A^{-1}U)^{-1}V^TA^{-1}.$$

Sea $H = B^{-1}$. Por lo tanto, si queremos calcular H_{k+1} , usamos la fórmula de Morrison-Woodbury:

$$H_{k+1} = H_k - \frac{H_k \mathbf{y}_k \mathbf{y}_k^T H_k}{\mathbf{y}_k^T H_k \mathbf{y}_k} + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}.$$

Si en vez del problema original (Ecuación secante) $B_{k+1} \mathbf{s}_k = \mathbf{y}_k$, se resuelve el problema $H_{k+1} \mathbf{y}_k = \mathbf{s}_k$ y procediendo de igual manera:

$$\min_H \|H - H_k\| \quad \text{Sujeto a } H = H^T, \quad H \mathbf{y}_k = \mathbf{s}_k,$$

con la norma de Frobenius pesada de nuevo, tenemos:

$$(\text{BFGS}) \quad H_{k+1} = (I - \rho_k \mathbf{s}_k \mathbf{y}_k^T) H_k (I - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T, \quad (9.3)$$

donde

$$\rho_k = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}.$$

Existen variantes para problemas grandes donde el almacenar H es completamente prohibitivo.

Algoritmo BFGS

Dado el punto inicial \mathbf{x}_0 , tolerancia $\epsilon > 0$ y una aproximación inicial H_0 ;

Hacer $k \leftarrow 0$

Mientras $\|\nabla f_k\| > \epsilon$

Calcular $\mathbf{p}_k = -H_k \nabla f_k$

Poner $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ con α_k satisfaciendo condiciones de Wolfe.

Definir

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$$

$$\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$$

Calcular H_{k+1} usando (BFGS) (9.3)

$$k = k + 1$$

end (mientras)

Una buena elección de H_0 es:

$$H_0 \leftarrow \frac{\mathbf{y}_k^T \mathbf{s}_k}{\mathbf{y}_k^T \mathbf{y}_k} I,$$

donde $\mathbf{y}_k^T \mathbf{y}_k$ intenta calcular la dimensión de $(\nabla f_{k+1})^{-1}$.

Capítulo 10

Mínimos Cuadrados No Lineales.

La función objetivo tiene la forma:

$$f(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^m r_j^2(\mathbf{x}) \quad (10.1)$$

donde cada fracción r_j es una función suave, $r_j : \Re^n \rightarrow \Re$. r_j es conocido como residual.

Es común encontrarlos en:

- Química
- Física
- Finanzas
- Economía
- Ingeniería Civil (estructuras)
- Inteligencia Artificial (visión, aprendizaje)

Son métodos robustos que explotan la estructura especial de f y sus derivadas. Su forma especial lo hace más fácil de resolver que otros problemas de optimización sin restricciones.

Sea el vector de residuo $\mathbf{r} : \Re^n \rightarrow \Re^m$ definido por:

$$\mathbf{r}(\mathbf{x}) = [r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_m(\mathbf{x})]^T, \quad (10.2)$$

luego f puede ser escrita como:

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|_2^2.$$

La derivada de f se expresa en términos del Jacobiano de \mathbf{r} ($J(\mathbf{x})$):

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial r_j}{\partial x_i} \end{bmatrix}_{\substack{j=1, \dots, m \\ i=1, \dots, n}}, \quad (10.3)$$

luego:

$$\nabla f(\mathbf{x}) = \sum_{j=1}^m r_j(\mathbf{x}) \nabla r_j(\mathbf{x}) = J(\mathbf{x})^T \mathbf{r}(\mathbf{x}), \quad (10.4)$$

y

$$\nabla^2 f(\mathbf{x}) = \sum_{j=1}^m \nabla r_j(\mathbf{x}) \nabla r_j(\mathbf{x}) + \sum_{j=1}^m \nabla r_j(\mathbf{x}) \nabla^2 r_j(\mathbf{x})$$

entonces:

$$\nabla^2 f(\mathbf{x}) = J(\mathbf{x})^T J(\mathbf{x}) + \sum_{j=1}^m r_j(\mathbf{x}) \nabla^2 r_j(\mathbf{x}). \quad (10.5)$$

Note que calcular el Jacobiano de $\mathbf{r}(\mathbf{x})$, J , permite calcular el gradiente $\nabla f(\mathbf{x})$ y la primera parte del Hessiano $\nabla^2 f(\mathbf{x})$.

Si definimos por $S(\mathbf{x})$ el Hessiano de $r_j(\mathbf{x})$, (10.5) se puede escribir como:)

$$\nabla^2 f(\mathbf{x}) = J(\mathbf{x})^T J(\mathbf{x}) + S(\mathbf{x}) \mathbf{r}(\mathbf{x}).$$

En general, el término $J^T J$ es más importante porque:

- Casi linearidad del modelo, cercana a la solución $\nabla^2 r_j(\mathbf{x})$ es pequeño.
- Por pequeños residuos de \mathbf{r} .

Varios algoritmos explotan esta estructura del Hessiano.

Entonces es posible aproximar el Hessiano de f por

$$\nabla^2 f(\mathbf{x}) \approx J^T(\mathbf{x}) J(\mathbf{x}),$$

este método se conoce como Gauss-Newton (es un Quasi-Newton).

Para problemas de gran escala, sin embargo, resulta a veces más conveniente calcular el gradiente en forma numérica.

Fundamentos del Modelado.

Sean los datos y_{t_i} de un experimento

t	y
1	✓
1.1	✓
2	✓
3	✓
4.5	✓
7	✓

Considera que dada nuestra experiencia, la siguiente función:

$$\phi(\mathbf{x}, t) = x_1 + x_2 t + e^{-x_3 t}$$

representa el comportamiento de los datos medidos experimentalmente, donde:

- $\mathbf{x} = [x_1, x_2, x_3]$ es el vector de parámetros (\mathbb{R}^3),
- $t \in \mathbb{R}$ representa el tiempo y nuestras mediciones y
- $y_j \in \mathbb{R}$ las m mediciones realizadas, para $j = 1, \dots, m$, $\mathbf{y} \in \mathbb{R}^m$.

Luego:

$$f(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^m [y_j - \phi(\mathbf{x}, t_j)]^2,$$

ó podemos

$$f(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^m [y_j - \phi(\mathbf{x}, t_j)]^6,$$

ó bien,

$$f(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^m |y_j - \phi(\mathbf{x}, t_j)|.$$

Porqué elegimos mínimos cuadrados?

Definamos el error

$$\epsilon_j = y_j - \phi(\mathbf{x}, t_j),$$

si asumimos que los ϵ_j son independientes e indécticamente distribuidas con varianza σ^2 y con distribución de probabilidad $g_\sigma(\cdot)$.

La verosimilitud de un conjunto de observaciones particulares dados los parámetros del modelo, está dada por:

$$P(\mathbf{y}; \mathbf{x}, \sigma) = \prod_{j=1}^m g_\sigma(\epsilon_j) = \prod_{j=1}^m g_\sigma(y_j - \phi(\mathbf{x}; t_j)).$$

Un criterio para elegir los mejores parámetros \mathbf{x} serán aquellos que maximicen $P(\mathbf{y}; \mathbf{x}, \sigma)$, esto se denomina Maximizador de la Verosimilitud (ML).

Asumamos que $g_\sigma(t)$ es la distribución normal:

$$g_\sigma(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

luego

$$P(\mathbf{y}; \mathbf{x}, \sigma) = (2\pi\sigma^2)^{-m/2} \exp\left[-\frac{1}{2} \sum_{j=1}^m \frac{(y_j - \phi(\mathbf{x}, t_j))^2}{\sigma^2}\right],$$

de aquí vemos que el estimador de ML se obtiene usando el criterio de mínimos cuadrados.

Mínimos Cuadrados.

Caso especial: r_j es lineal, luego el jacobiano J es constante

$$f(\mathbf{x}) = \frac{1}{2} \|J(\mathbf{x}) + \mathbf{r}\|_2^2, \quad (10.6)$$

donde $\mathbf{r} = \mathbf{r}(0)$ (\mathbf{r} dado no es el residuo, son datos), luego:

$$\begin{aligned} \nabla f(\mathbf{x}) &= J^T(J\mathbf{x} + \mathbf{r}), \\ \nabla^2 f(\mathbf{x}) &= J^T J. \end{aligned}$$

Note que el segundo término de (10.5) desaparece porque $\nabla^2 f_j = 0 \forall i$.

(10.6) es convexa (no siempre satisfecho en el caso no lineal). La solución se encuentra haciendo:

$$\nabla^* f(\mathbf{x}) = 0,$$

luego, esto debe satisfacer

$$J^T J \mathbf{x}^* = -J^T \mathbf{x}, \quad (10.7)$$

el cual es un problema de Álgebra Lineal.

Algoritmo para problemas de Mínimos Cuadrados no Lineales.

Los métodos tipo Newton requieren:

$$\nabla^2 f(\mathbf{x}_k) \mathbf{p} = -\nabla f(\mathbf{x}_k)$$

luego el Gauss-Newton simplifica (Quasi-Newton)

$$\nabla f^2(\mathbf{x}_k) \approx J^T(\mathbf{x}_k) J(\mathbf{x}) = J_k^T J_k. \quad (10.8)$$

Requiere sólo calcular el Jacobiano J_k !

No es necesario calcular el Hessiano individual $\nabla^2 r_i, i = 1, 2, 3, \dots, m$.

Ventajas del Método de Gauss-Newton.

1. No es necesario calcular los Hessianos individuales $\nabla^2 r_j(\mathbf{x})$, para $j = 1, 2, \dots, m$.
2. En muchas situaciones el término $J^T J$ es más significativo por lo que la solución no cambia drásticamente, ni el desempeño del algoritmo se ve disminuido si se omite el término $\sum_{j=1}^m r_j \nabla^2 r_j$, esto es en casos que $|r_j(\mathbf{x})| \|nabla^2 r_j(\mathbf{x})\|$ es significativamente menor que los eigenvalores de $J^T J$.

Ocurre cuando:

- el residual r_j es pequeño (pequeños residuos) ó
 - r_j es casi lineal.
3. Si $J(\mathbf{x})$ es de rango completo y el gradiente $\nabla f(\mathbf{x})$ es no cero, la dirección de descenso \mathbf{p}^{GN} es una dirección de descenso de $f(\mathbf{x})$, y por lo tanto una dirección válida para hacer búsqueda en línea.

$$(\mathbf{p}_k^{GN})^T \nabla f_k = (\mathbf{p}_k^{GN})^T J_k \mathbf{r}_k = -(\mathbf{p}_k^{GN})^T J_k^T J_k \mathbf{p}_k = -\|J_k \mathbf{p}_k^{GN}\|^2 \leq 0,$$

donde la desigualdad es estricta, a menos que $\nabla f_k = 0!$ (ver (10.4)).

4. Ventaja: La ecuación de Gauss-Newton es la solución al problema de mínimos cuadrados

$$\min_{\mathbf{p}} \|J_k \mathbf{p} + f_k\|_2^2, \quad (10.9)$$

es decir, \mathbf{p} (el paso) puede ser calculado resolviendo (10.9) con cualquier método.

De (10.9) podemos deducir otra motivación más al método Gauss-Newton. En vez de formar un modelo cuadrático de $f(\mathbf{x})$, estamos usando un modelo lineal de la función vectorial $\mathbf{r}(\mathbf{x} + \mathbf{p}) \approx \mathbf{r}(\mathbf{x}) + J(\mathbf{x})\mathbf{p}$, luego el paso \mathbf{p} se calcula sustituyendo este modelo lineal en la expresión:

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|_2^2,$$

y minimizar con respecto a \mathbf{p} .

Levenberg-Marquart

Gauss-Newton requiere que $J^T J$ sea definida positiva. Si esto no es el caso: (LM) propone

$$\tilde{H}(\mathbf{x}) \approx J^T(\mathbf{x}) J(\mathbf{x}) + \lambda I,$$

donde λ se escoge tal que $H(\mathbf{x})$ sea positiva definida. Puede usarse una motivación de región verdadera.

Newton con región verdadera \Rightarrow LM.

Problema con Gauss-Newton-GC

$$U(t) = \sum_r (f_r - g_r)^2 + \sum_{\langle r, s \rangle} \rho^2 (f_r - f_s).$$

- 1.** Verlo en 1D.
- 2.** Asumir el caso simple de $\rho(x) = x$.
- 3.** Analizar quien es J_r y cómo se calcula $J^T J_r$.
- 4.** Generalizarlo para cualquier $\rho(\mathbf{x})$.