

Introducción al cálculo numérico y ciencia de datos

Clase 2

Mg. Víctor Viana

Dr. Diego Passarella

Tacuarembó – abril de 2023

Agenda

- Resolución de sistemas de ecuaciones lineales
- Resolución de ecuaciones no lineales
- Optimización por métodos con y sin derivadas. Máximos y mínimos
- Ajuste de funciones. Mínimos cuadrados

Sistemas de ecuaciones lineales (SEL)

Muchos problemas de la física e ingenierías pueden asumirse con una relación lineal entre las variables del problema y el resultado observable (problemas térmicos, eléctricos, mecánicos, hidráulicos, discretización de ecuaciones matemáticas). Estos problemas, se representan a través de un conjunto de ecuaciones lineales que conforman un Sistema de Ecuaciones Lineales (SEL).

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \rightarrow \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$Ax = b \quad \rightarrow \quad A^{-1}Ax = A^{-1}b$$

Sistemas de ecuaciones lineales (SEL)

Los Sistemas de Ecuaciones Lineales pueden resolverse por medio de dos grandes familias de métodos:

- **Métodos directos:** Ideales para sistemas relativamente pequeños y con matrices densamente pobladas. Se basan en operar sobre la matriz A o su expandida $[A|b]$ para llevarla a una forma que sea fácilmente resoluble (matriz diagonal o triangular). Se alcanza la solución (idealmente exacta) en un número finito de operaciones que es conocido *a priori*.

Sistemas de ecuaciones lineales (SEL)

Métodos directos – **Algoritmo de remonte**: Sirve para matrices de coeficientes triangular superior (matriz que posee ceros debajo de su diagonal principal)

Algoritmo:

1)

$$x_n = \frac{b_n}{u_{nn}}$$

2) Para $i = n-1$ hasta 1

$$x_i = \frac{b_i - \sum_{j=i+1}^n x_j u_{ij}}{u_{ii}}$$

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Sistemas de ecuaciones lineales (SEL)

Métodos directos – **Eliminación de Gauss**: Se opera con el sistema completo para hacer cero a los elementos debajo de la diagonal principal. Se trabaja con la matriz expandida $A^* = [A|b]$.

$$A^* = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array} \right)$$

Algoritmo:

1) Para $k = 1$ hasta $n-1$

#recorro las primeras $n-1$ columnas

2) para $i=k+1$ hasta n

recorro todas las filas debajo de la diagonal

$$l_{ik} = \frac{a_{ik}^*}{u_{nn}}$$

3) para $j=k$ hasta $n+1$

recorro toda la i -ésima fila

$$a_{ij}^* = a_{ij}^* - l_{ik} a_{kj}^*$$

Una vez que se hace triangular la matriz A^* , se separa en matriz de coeficientes A y vector de términos independientes b . El problema se resuelve por el algoritmo de remonte.

Sistemas de ecuaciones lineales (SEL)

Métodos directos – **Factorización LU**: Se basa en expresar la matriz A, como el producto de dos matrices triangulares, L, triangular inferior y U, triangular superior, de forma que:

$$Ax = LUx = b$$

primero resuelvo el sistema: $Ly = b$

para finalmente encontrar: $Ux = y$

Algoritmo:

- 1) Utilizar el mismo algoritmo que para la eliminación de Gauss, pero solamente a la matriz A (no a la expandida).
- 2) La matriz A resultante va a ser la matriz U y los coeficientes l_{ik} pasan a ser los elementos del triángulo inferior de L (agregar una diagonal de 1's).

La factorización LU es conveniente cuando hay que resolver varios sistemas con la misma matriz A, pero con distintos vectores b.

Sistemas de ecuaciones lineales (SEL)

Otros métodos directos:

Factorización de Thomas: Factorización LU de sistemas tridiagonales

Factorización de Cholesky: Para matrices A simétricas y definidas positivas.

Se factoriza la matriz A como el producto de una matriz triangular, por su traspuesta.

Sistemas de ecuaciones lineales (SEL)

Retomando, los Sistemas de Ecuaciones Lineales pueden resolverse por medio de dos grandes familias de métodos:

- **Métodos iterativos:** Convenientes para sistemas con un gran número de incógnitas y estructuradamente ralos (muchos ceros dispuestos de forma conocida en la matriz A). Se genera una sucesión que aproxima a la solución por medio de la repetición de operaciones “matriz x vector” en cada iteración. Se llega a una aproximación de la solución con una dada tolerancia en una cantidad a priori desconocida de pasos.

Sistemas de ecuaciones lineales (SEL)

Sistemas iterativos – Descomposición de matrices

Idea: descomponer el sistema $Ax = b$ en un punto fijo de la forma $x = Gx + c$

Se descompone a la matriz como la resta de dos matrices $A = M - N$, donde M es una matriz fácilmente inversible (diagonal o triangular) y se llega a:

$$Ax = b \rightarrow [M - N]x = b$$

$$Mx = Nx + b$$

$$x = M^{-1}Nx + M^{-1}b$$

$$x^{(k)} = Gx^{(k-1)} + c \quad (\text{punto fijo})$$

Sistemas de ecuaciones lineales (SEL)

Sistemas iterativos – Descomposición de matrices

Formulación eficiente:

Si $A = [M - N] \rightarrow N = M - A$ entonces

$$x = M^{-1}Nx + M^{-1}b = M^{-1}(M - A)x + M^{-1}b$$

$$x = Ix - M^{-1}Ax + M^{-1}b$$

$$x = Ix + M^{-1}(b - Ax)$$

$$x^{(k)} = x^{(k-1)} + M^{-1}r^{(k-1)} \quad (\text{punto fijo})$$

$$\text{Con } r^{(k-1)} = b - Ax^{(k-1)}$$

Sistemas de ecuaciones lineales (SEL)

Sistemas iterativos – Descomposición de matrices

Resolución de la formulación eficiente:

Dados A y b , se define un iterante inicial $x^{(0)}$, un número máximo de iteraciones

N_{max} y una tolerancia ε , se ejecuta:

$$r^{(0)} = b - Ax^{(0)}, k = 0$$

Mientras $k < N_{max}$ y $r^{(k)} > \varepsilon \|b\|$

se resuelve $Mz = r^{(k)}$

$$x^{(k+1)} = x^{(k)} + z$$

$$r^{(k+1)} = b - Ax^{(k+1)}$$

$$k = k + 1$$

Sistemas de ecuaciones lineales (SEL)

Sistemas iterativos – Descomposición de matrices

Descomposición de Jacobi:

La matriz M pasa a ser la diagonal (D) de la matriz de coeficientes A :

Formulación eficiente:

$$x^{(k+1)} = x^{(k)} + D^{-1}r^{(k)}$$

Formulación por componentes:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) / a_{ii}$$

Sistemas de ecuaciones lineales (SEL)

Sistemas iterativos – Descomposición de matrices

Descomposición de Gauss-Seidel:

La matriz M pasa a ser el triángulo inferior de la matriz A , incluyendo a la diagonal (L):

Formulación eficiente:

$$x^{(k+1)} = x^{(k)} + L^{-1}r^{(k)}$$

Formulación por componentes:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) / a_{ii}$$

Resolución de ecuaciones no lineales

Sea una función $f(x)$ continua en un intervalo (a, b) y sea el valor α incluido en ese intervalo. α es una raíz de $f(x)$ sí y solo sí $f(\alpha) = 0$.

La separación de raíces consta en encontrar los intervalos del dominio en el que se encuentra al menos una función. Para ello se pueden utilizar métodos gráficos o analíticos ($f(a)f(b) < 0$)

Una vez que se hallan los intervalos donde se encuentra al menos una raíz, se utilizan métodos iterativos para generar una sucesión $\{x^{(k)}\}$ que converja a α .

Resolución de ecuaciones no lineales

Método de bisección:

Genera una sucesión de intervalos $I^{(k)} = [a^{(k)}, b^{(k)}]$, cada uno de la mitad de tamaño del anterior, con la condición que en cada intervalo, esté contenido α .

Algoritmo:

Dado $I^{(0)} = [a^{(0)}, b^{(0)}]$ con $(f(a^{(0)})f(b^{(0)}) < 0)$

Dados $k = 1, \varepsilon$ y N_{max}

Mientras $|I^{(k)}| > \varepsilon$ & $k < N_{max}$

$$x^{(k)} = \frac{(a^{(k)} + b^{(k)})}{2}$$

si $f(a^{(k)})f(x^{(k)}) < 0$

$$a^{(k+1)} = a^{(k)} \text{ y } b^{(k+1)} = x^{(k)}$$

si no

$$a^{(k+1)} = x^{(k)} \text{ y } b^{(k+1)} = b^{(k)}$$

$$k = k + 1$$

Resolución de ecuaciones no lineales

Método de Newton-Raphson:

Es necesario conocer la expresión de la derivada de la función. Se basa en utilizar el desarrollo de Taylor de primer orden.

$$f(x) = f(x_0) + \sum_{s=1}^{\infty} f^{(s)}(x_0) \frac{(x - x_0)^s}{s!}$$

$$f(x^{(k+1)}) = f(x^{(k)}) + (x^{(k+1)} - x^{(k)})f'(x^{(k)})$$

Si buscamos que $f(x^{(k+1)}) \rightarrow 0$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

(comentar condiciones de convergencia)

Resolución de ecuaciones no lineales

Iteración de punto fijo:

Se dice que α es un punto fijo de una función $g(x)$ si se cumple:

$$g(\alpha) = \alpha$$

Con esto en cuenta, se puede intentar generar un punto fijo a partir de $f(x)$

Idea:

llegar a la forma $f(x) = g(x) + x$ adecuada

proponer un $x^{(0)}$

iterar $x^{(k+1)} = g(x^{(k)})$ hasta $|x^{(k+1)} - x^{(k)}| < \varepsilon$

(comentar condiciones de convergencia)

Algunos métodos de optimización

Concepto general de optimización:

Sea una función $\varphi(\vec{x})$ escalar, con dominio $D \subset \mathbb{R}^n$ el mínimo de φ es un valor $\vec{\alpha}$ que cumple:

$$\varphi(\vec{\alpha}) \leq \varphi(\vec{x}) \quad \text{para todo } \vec{x} \in D$$

(este problema se plantea de forma equivalente si hablamos de un máximo y se utiliza el signo \geq)

Encontrar el valor de $\vec{\alpha}$ se puede plantear como:

$$\vec{g}(\vec{x}) = \nabla\varphi(\vec{x}) = \vec{0}$$

Para funciones de $\mathbb{R} \rightarrow \mathbb{R}$ el problema es encontrar las raíces de f'

Algunos métodos de optimización

Método de Newton:

Para encontrar el cero de $\vec{g}(\vec{x})$ se propone una estrategia similar al método de Newton-Raphson, pero para funciones de varias variables. El polinomio de primer orden que aproxima a $\vec{g}(\vec{x})$ se expresa como:

$$\vec{p}_1(\vec{x}^{(k+1)}) = \vec{g}(\vec{x}^{(k)}) + J\vec{g}(\vec{x}^{(k)})(\vec{x}^{(k+1)} - \vec{x}^{(k)}) \approx \vec{0}$$

Con $J\vec{g}(\vec{x}^{(k)})$, la matriz jacobiana de \vec{g} evaluada en $\vec{x}^{(k)}$ y dado que

$$\vec{g}(\vec{x}) = (g_1(\vec{x}), g_2(\vec{x}), g_3(\vec{x}), \dots, g_n(\vec{x})) = \nabla\varphi(\vec{x}) = \left(\frac{\partial\varphi}{\partial x_1}, \frac{\partial\varphi}{\partial x_2}, \frac{\partial\varphi}{\partial x_3}, \dots, \frac{\partial\varphi}{\partial x_n} \right)$$

La matriz jacobiana queda como:

$$J\vec{g} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} & \dots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} & \dots & \frac{\partial g_2}{\partial x_n} \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3} & \dots & \frac{\partial g_3}{\partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \frac{\partial g_n}{\partial x_3} & \dots & \frac{\partial g_n}{\partial x_n} \end{pmatrix} = H\varphi = \begin{pmatrix} \frac{\partial^2 \varphi}{\partial x_1^2} & \frac{\partial^2 \varphi}{\partial x_2 \partial x_1} & \frac{\partial^2 \varphi}{\partial x_3 \partial x_1} & \dots & \frac{\partial^2 \varphi}{\partial x_n \partial x_1} \\ \frac{\partial^2 \varphi}{\partial x_1 \partial x_2} & \frac{\partial^2 \varphi}{\partial x_2^2} & \frac{\partial^2 \varphi}{\partial x_3 \partial x_2} & \dots & \frac{\partial^2 \varphi}{\partial x_n \partial x_2} \\ \frac{\partial^2 \varphi}{\partial x_1 \partial x_3} & \frac{\partial^2 \varphi}{\partial x_2 \partial x_3} & \frac{\partial^2 \varphi}{\partial x_3^2} & \dots & \frac{\partial^2 \varphi}{\partial x_n \partial x_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \varphi}{\partial x_1 \partial x_n} & \frac{\partial^2 \varphi}{\partial x_2 \partial x_n} & \frac{\partial^2 \varphi}{\partial x_3 \partial x_n} & \dots & \frac{\partial^2 \varphi}{\partial x_n^2} \end{pmatrix}$$

Algunos métodos de optimización

De la aproximación lineal de la función, se tiene:

$$\vec{p}_1(\vec{x}^{(k+1)}) = \vec{g}(\vec{x}^{(k)}) + J\vec{g}(\vec{x}^{(k)})(\vec{x}^{(k+1)} - \vec{x}^{(k)}) \approx \vec{0}$$

Por lo que el punto fijo a iterar resulta:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - J\vec{g}^{-1}(\vec{x}^{(k)})\vec{g}(\vec{x}^{(k)})$$

Que en realidad se resuelve como:

$$J\vec{g}(\vec{x}^{(k)})z = \vec{g}(\vec{x}^{(k)})$$

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - z$$

Mencionar las dificultades de evaluar y factorizar la matriz jacobiana

Algunos métodos de optimización

Optimización sin derivadas:

En el caso que la evaluación de $J\vec{g}$ sea muy costosa computacionalmente o complicada de obtener, se pueden explorar métodos que no emplean derivadas.

Son métodos por comparación directa con heurísticas de búsqueda.

Por ejemplo, el método de Nelder-Mead, para una función n-dimensional inicia la búsqueda en un conjunto de n+1 puntos, de los cuales habrá uno que entregue el máximo y el mínimo (luego de ordenarlos). Dependiendo de la ubicación del centroide de los n mejores puntos y el peor (n+1), se realizará la búsqueda a lo largo de esa dirección.

(ejemplos a desarrollar)

Ajuste de funciones

Ajuste por mínimos cuadrados:

Supongamos que se tienen $n + 1$ pares de puntos (x_i, y_i) y se les desea ajustar un polinomio grado m , $P_m(x)$ con $(m < n)$ que cumpla la siguiente condición:

$$\sum_{i=0}^n (y_i - P_m(x_i))^2 \rightarrow \text{mínimo}$$

Siendo que el óptimo $P_m(x)$ se expresa como $a_0 + a_1x + a_2x^2 + \dots + a_mx^m$ y un $P_m(x)$ genérico se expresa como $b_0 + b_1x + b_2x^2 + \dots + b_mx^m$, el problema de ajuste por mínimos cuadrados se reescribe como:

$$\varphi(a_0, a_1, a_2, \dots, a_m) = \min_{b_i} \varphi(b_0, b_1, b_2, \dots, b_m)$$

Ajuste de funciones

El funcional a minimizar, de forma desarrollada queda:

$$\varphi(\vec{b}) = \sum_{i=0}^n y_i^2 - 2y_i(b_0 + b_1x_i + \dots + b_mx_i^m) + (b_0 + b_1x_i + \dots + b_mx_i^m)^2$$

Los coeficientes de la solución óptima $(a_0, a_1, a_2, \dots, a_m)$ se obtienen de plantear $\nabla_{\vec{b}}\varphi = \vec{0}$, con cada una de sus derivadas parciales iguales a:

$$\frac{\partial \varphi}{\partial b_j} = - \sum_{i=0}^n y_i x_i^j + b_0 \sum_{i=0}^n x_i^j + b_1 \sum_{i=0}^n x_i^{1+j} + \dots + b_m \sum_{i=0}^n x_i^{m+j} = 0$$

Con $j = 0, 1, 2, \dots, m$, de esta forma se obtiene un sistema de $m + 1$ ecuaciones lineales, para obtener los $(a_0, a_1, a_2, \dots, a_m)$ coeficientes óptimos

Ajuste de funciones

El problema de encontrar un polinomio $P_m(x)$ que ajuste de forma óptima un conjunto de datos experimentales termina siendo:

$$\begin{pmatrix} n+1 & \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \dots & \sum_{i=0}^n x_i^m \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \dots & \sum_{i=0}^n x_i^{1+m} \\ \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \sum_{i=0}^n x_i^4 & \dots & \sum_{i=0}^n x_i^{2+m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^n x_i^m & \sum_{i=0}^n x_i^{1+m} & \sum_{i=0}^n x_i^{2+m} & \dots & \sum_{i=0}^n x_i^{2m} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n y_i x_i \\ \sum_{i=0}^n y_i x_i^2 \\ \vdots \\ \sum_{i=0}^n y_i x_i^m \end{pmatrix}$$