

Actividad:

Métodos iterativos para la resolución de sistemas lineales: Jacobi, Gauss–Seidel y relajación

Los métodos de resolución de sistemas lineales que hemos visto durante la asignatura son métodos directos, y son útiles para la resolución de sistemas de dimensión no muy elevada. A medida que incrementamos la dimensión de los problemas, los errores de redondeo van creciendo, cosa que los hace poco aplicables. Hay dos formas de resolver este problema: aplicando un procedimiento conocido como *mejora por iteración* o usando *métodos iterativos*. En esta actividad nos introduciremos en el mundo de los métodos iterativos para resolver sistemas de la forma

$$Ax + b,$$

donde A es una matriz $n \times n$ y b un vector de tamaño n .

Los **métodos iterativos** son menos sensibles a los errores de redondeo y están más adaptados a resolver sistemas grandes, que habitualmente tienen muchos coeficientes nulos. Cuando trabajamos con métodos iterativos siempre partiremos de una primera aproximación de la solución, $x^{(0)}$. También necesitaremos una cota del error que estamos dispuestos a admitir, ϵ_0 . Todos los métodos iterativos seguirán el esquema

$$x^{(\ell+1)} = Dx^{(\ell)} + C,$$

donde D será una matriz y C un vector constantes a lo largo de las iteraciones. Tanto D como C se construirán a partir de la matriz A del sistema y del vector de términos independientes b .

Como todos los métodos iterativos, pueden presentar problemas de convergencia. La condición que nos permitirá asegurar la convergencia de estos métodos será la *predominancia diagonal* de la matriz del sistema. Decimos que una matriz cuadrada $A = (a_{ij})$ tiene **predominancia diagonal** si, para todo $i = 1, \dots, n$,

$$|a_{ii}| \geq \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}|.$$

En algunos casos, si no tenemos predominancia diagonal, podemos hacer permutaciones de filas y columnas para conseguirlo, aunque no siempre será posible.

Las matrices tridiagonales (o matrices banda en general) suelen ser buenas candidatas para estos métodos, ya que es probable que tengan predominancia diagonal por la gran cantidad de ceros que contienen.

En general, usaremos descomposiciones de la matriz A de la forma

$$A = M + N.$$

Entonces, el sistema lineal $Ax = b$ adquiere la forma

$$(M + N)x = b.$$

Manipulando la ecuación anterior, llegamos a

$$x = \underbrace{-M^{-1}N}_D x + \underbrace{M^{-1}b}_C,$$

obteniendo un esquema iterativo de la forma $x^{(\ell+1)} = Dx^{(\ell)} + C$ descrita anteriormente. En función de la elección de la descomposición $A = M + N$ llegaremos a distintos métodos iterativos.

Método de Jacobi

El método de Jacobi es un método iterativo para la resolución de sistema de ecuaciones lineales de la forma $Ax = b$. En este caso, la descomposición que elegimos es

$$A = M + N,$$

donde M es la matriz diagonal que contiene la diagonal de A , mientras que N es la matriz A sin los elementos de la diagonal:

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}}_A = \underbrace{\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}}_M + \underbrace{\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & 0 & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix}}_N.$$

En este caso, el esquema

$$x^{(\ell+1)} = \underbrace{-M^{-1}N}_{D} x^{(\ell)} + \underbrace{M^{-1}b}_C,$$

se escribe como

$$x_i^{(\ell+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(\ell)} \right).$$

Como primera aproximación de la solución se suele tomar

$$x_i^{(0)} = 0, \quad i = 1, \dots, n,$$

o bien,

$$x_i^{(0)} = \frac{b_i}{a_{ii}}, \quad i = 1, \dots, n.$$

Notemos que en el cálculo de $x^{(\ell+1)}$ solo intervienen los valores de $x^{(\ell)}$. En el método que veremos a continuación veremos que se usan los valores de $x^{(\ell+1)}$ ya calculados para calcular el resto de valores de $x^{(\ell+1)}$, lo que acelera el método.

Método de Gauss–Seidel

El **método de Gauss–Seidel** es un método iterativo muy similar al método de Jacobi. En este caso, la descomposición de la matriz A será

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}}_A = \underbrace{\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}}_{A_1} + \underbrace{\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}}_{A_2}.$$

El esquema

$$x^{(\ell+1)} = \underbrace{-A_1^{-1}A_2}_{D} x^{(\ell)} + \underbrace{A_1^{-1}b}_C,$$

que obtenemos es idéntico al método de Jacobi, con la diferencia que para calcular los valores de $x^{(\ell+1)}$ usaremos si es posible los valores de $x^{(\ell+1)}$ ya calculados:

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

El hecho de usar valores ya calculados de la iteración $\ell + 1$ acelera la velocidad de convergencia. Como primera aproximación, se suele tomar

$$x_i^{(0)} = 0, \quad i = 1, \dots, n,$$

o bien,

$$x_i^{(0)} = \frac{b_i}{a_{ii}}, \quad i = 1, \dots, n.$$

Métodos de relajación

La relajación es una pequeña modificación del método de Gauss–Seidel, pensada para acelerar todavía más la velocidad de convergencia del método. La idea es, una vez calculado un nuevo valor $x_{\text{calculado}}^{(\ell+1)}$, modificarlo ligeramente:

$$x^{(\ell+1)} = \omega x_{\text{calculado}}^{(\ell+1)} + (1 - \omega) x^{(\ell)},$$

donde ω es un número real elegido adecuadamente. La elección de ω es un problema que suele resolverse empíricamente.

- Si el valor de ω se encuentra en el intervalo $(0, 1)$, hablamos de **subrelajación**. La subrelajación se usa para que un sistema que no converge pase a converger.
- Si tomamos $\omega > 1$ (habitualmente $1 < \omega < 2$) hablamos de **sobrerelajación**. Se usa para acelerar la convergencia.
- El caso $\omega = 1$ corresponde a Gauss–Seidel.

Se pide:

1. Escribir una función `predominante_diagonal` que reciba como parámetro una matriz cuadrada y devuelva `true` o `false` en función de si la matriz tiene predominancia diagonal o no.
2. Escribir una función `jacobi` que reciba como parámetro una matriz cuadrada A , un vector de términos independientes b y un valor de tolerancia, y devuelva la solución del sistema $Ax = b$ usando el método de Jacobi, así como el número de iteraciones. La función debe comprobar que la matriz tiene predominancia diagonal.
3. Escribir una función `gauss_seidel` que reciba como parámetro una matriz cuadrada A , un vector de términos independientes b y un valor de tolerancia, y devuelva la solución del sistema $Ax = b$ usando el método de Gauss–Seidel, así como el número de iteraciones. La función debe comprobar que la matriz tiene predominancia diagonal.

4. Escribir una función **relajacion** que reciba como parámetro una matriz cuadrada A , un vector de términos independientes b , un valor de ω y un valor de tolerancia, y devuelva la solución del sistema $Ax = b$ usando el método de relajación, así como el número de iteraciones. La función debe comprobar que la matriz tiene predominancia diagonal.
5. Escribir un programa principal que resuelva un sistema lineal con predominancia diagonal usando la descomposición LU vista en clase y mediante los tres métodos anteriores comprobando que la solución coincide, y compare el número de iteraciones usado en los distintos métodos iterativos. Comprobad que el número de iteraciones requerido en Gauss–Seidel es menor que en Jacobi. Buscad un valor de ω que acelere todavía más el método de Gauss–Seidel.

Para este último apartado, usad un valor de la tolerancia de $\epsilon_0 = 10^{-10}$. Podéis usar sistema

$$\begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} -120 \\ 0 \\ -30 \\ -70 \\ 0 \\ -20 \\ -290 \\ -170 \\ -160 \end{pmatrix},$$

que tiene solución

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} 60 \\ 40 \\ 30 \\ 80 \\ 70 \\ 50 \\ 120 \\ 110 \\ 80 \end{pmatrix}.$$

Observación

Esta actividad puede hacerse en grupos de 1–3 personas. Cada miembro del grupo tiene que entregar el mismo archivo ***.zip** (ver Formato de entrega) y tanto en el código (por ejemplo con un comentario) como en el pdf debe aparecer el nombre de todos los integrantes del grupo.

Formato de entrega

La resolución numérica debe hacerse usando el lenguaje de programación C++. Si se desea entregar algún documento de texto, debe estar escrita usando \LaTeX , y el documento tiene que ser un PDF.

Tanto el documento con las posibles explicaciones como los ficheros de código deben entregarse en un único archivo comprimido ***.zip**.