

## **Métodos iterativos para resolución de sistemas de ecuaciones**

---

### **Introducción**

---

El objetivo de este trabajo es la implementación y experimentación con métodos iterativos para resolución de sistemas de ecuaciones y la comparación con métodos directos como eliminación gaussiana.

Los métodos iterativos permiten generar resultados aproximados en menor cantidad de pasos que un método directo.

Nos interesa poder experimentar con esta propiedad midiendo la convergencia hacia la solución y teniendo en cuenta el tiempo en el que lo hacen.

### **Métodos Iterativos**

---

A continuación definimos las operaciones de los métodos iterativos. Estos pueden expresarse de forma puramente matricial o utilizando sumatorias.

#### **Jacobi**

---

##### **Método matricial**

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$$
$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{b} + (\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)}).$$

##### **Método sumatoria**

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n.$$

#### **Gauss-Seidel**

---

##### **Método matricial**

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$$

$$\mathbf{x}^{(k+1)} = (\mathbf{D} - \mathbf{L})^{-1} (\mathbf{b} + U\mathbf{x}^{(k)}).$$

## Método sumatoria

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right), \quad i = 1, 2, \dots, n.$$

## Consignas

---

El trabajo consiste en implementar en C++, utilizando la biblioteca Eigen, los métodos iterativos de Jacobi y Gauss-Seidel tanto de forma matricial obteniendo la inversa o explícitamente con la sumatoria y establecer qué forma es más rápida. A su vez, nos interesa comparar el rendimiento de los algoritmos anteriores contra algún método directo, por lo que consideraremos el método LU de Eigen.

Para los métodos iterativos queremos medir: el error de aproximación entre el valor actual y el resultado final utilizando alguna medida (por ejemplo la norma 2) y el tiempo final de cómputo. Considerar un número de iteración máximo y el monitoreo de la diferencia absoluta entre dos iteraciones para establecer un punto de corte o una alarma si el método diverge. Para el método directo, medir el tiempo de cómputo y el error numérico con la solución real si es que lo hay.

Para la experimentación probar diseñando sistemas de ecuaciones lineales con soluciones fáciles de comprobar, por ejemplo generando con valores enteros  $A$  y  $x$  y luego encontrar  $b$ . Probar para distintos tamaños de matrices (utilizar un rango exponencial de valores) y distintos valores iniciales. Observar si los errores de los métodos iterativos tienen distintas velocidades de convergencia según alguna propiedad de la matriz, como su mal condicionamiento a los supuestos de los métodos. En último lugar, se debe mostrar la evolución del error en función de la iteración gráficamente.

Para los experimentos de medición de tiempo de ejecución, realizar suficientes repeticiones analizando la estadística de estos tiempos y determinando que resumen (promedio o mínimo) utilizar. Observar cómo escalan los tiempos con el tamaño de las matrices.

Considerar mostrar barras de errores o boxplots para enfatizar las distribuciones si es necesario.

Como consideración final generar alguna comparación entre los 3 métodos, según sus ventajas o desventajas considerando errores y tiempos.

## **IMPORTANTE**

Las implementaciones de los algoritmos deben realizarse en C++ utilizando Eigen. Establecer según el caso, si los tiempos de cómputo deben ser medidos desde C++ o python. La experimentación está pensada para ser realizada en Python.

## **Forma y Fecha de entrega**

---

1. Se debe entregar un informe hecho en Latex. Este debe incluir una introducción al problema, la solución a las consignas mostrando pseudocódigos con una explicación de su funcionamiento. Para visualizar e interpretar los resultados se sugiere el uso de tablas o figuras, las cuales tienen que estar debidamente referenciadas en el texto y tener una descripción. Por último, se debe incluir una sección de conclusiones con un resumen de lo mostrado y los principales resultados.
2. Se debe entregar los códigos que generan los resultados, figuras, etc. Si se requieren procedimientos especiales para ejecutar el código deben ser descritos.
3. La entrega es el Domingo 2 de Julio 23:59 h. Para la entrega se facilitará un formulario online donde deberán adjuntar el informe y los archivos.
4. Recuperatorio: Viernes 14 de Julio hasta las 23.59 hs, enviando el trabajo corregido.