Tarea 6

CIMAT

Nombre:	**Jairo Saul Diaz Soto**
Maestría:	Ciencias Computacionales
Modulo:	Métodos Numéricos
Instructor:	Dr. Salvador Botello Rionda
Fecha de entrega:	2023 - 09 - 24

1. Introducción

Una de las técnicas utilizadas para resolver sistemas de ecuaciones lineales, es el denominado **método iterativo**, este tipo de métodos son aproximados y esta técnica se basa en la siguiente desigualdad, donde se propone que si para

$$\lim_{k\to\infty} \{\mathbf{x}^{(k)}\}$$
$$||\mathbf{x}^{(k)} - \mathbf{x}|| < \epsilon, \qquad \epsilon > 0$$
$$if \ \exists N(\epsilon) \leq k \tag{1}$$

Donde $\{\mathbf{x}^{(k)}\}$ son vectores de $\Re^n$, que constituyen una sucesión la cual se construye de acuerdo a lo siguiente

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{c}$$

Para lo cual B es una matriz denominada matriz de iteración y $\mathbf{c}$ es un vector fijo. Ahora bien, para poder determinar la condición de paro par el algoritmo, no es posible comparar la distancia entre $\mathbf{x}^{(k)}$ y la solución $\mathbf{x}$, por lo cual se define el residuo tal que:

$$\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$$

Se podría, a través del residuo y cualquier norma, determinar el distanciamiento que se tiene con la solución exacta.

1.1. Método de Jacobi

Para este método, se propone lo siguiente, si se tiene el sistema de ecuaciones $A\mathbf{x} = \mathbf{b}$, entonces se puede definir que

$$A = M - N, \quad M = D, \quad N = D - A = L + U$$

Donde D es la diagonal principal de la matriz A, L es una matriz inferior tal que $l_{ij} = -a_{ij}$ para $i \geq j$ mientras que U es una matriz triangulas superior tal que $l_{ij} = -a_{ij}$ para $j \geq i$. Con esto entonces la sucesión de vectores se define como

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

Por lo que se tiene entonces la matriz de iteración esta definida como $\mathbf{B} = D^{-1}(L+U)$ y el vector fijo esta definido como $\mathbf{c} = D^{-1}\mathbf{b}$. Para la obtención de cada uno de los elementos del k-ésimo vector se tiene que

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

Entonces, Se tiene que el algoritmo para este método es el siguiente

1.2. Método Gauss - Seidel

Bajo el mismo supuesto de descomposición $A = M - N$, ahora se redefinen como se declaran estas matrices tal que

$$M = D - L, \quad N = U$$

Algorithm 1 Método de Jacobi

Inicializar $x^{(0)}$ ▷ Vector de inicio
Inicializar tolerancia ϵ ▷ Tolerancia deseada
Inicializar número máximo de iteraciones $N_{\max}$
Inicializar contador de iteraciones $k \leftarrow 0$
while $k < N_{\max}$ **do**
    $k \leftarrow k + 1$
    for $i \leftarrow 1$ to n **do** ▷ Para cada componente de x
        $x_i^{(k)} \leftarrow \frac{1}{A_{ii}} \left(b_i - \sum_{j \neq i} A_{ij} x_j^{(k-1)} \right)$ ▷ Actualización de x_i
    end for
    if $\|x^{(k)} - x^{(k-1)}\| < \epsilon$ **then** ▷ Condición de convergencia
        Salir del bucle
    end if
end while

Para lo cual D, LyU siguen manteniendo su construcción inicial, pero ahora, la sucesión está dada como

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

Por lo tanto, la matriz de iteración resulta en $B = (D-L)^{-1}U$ y el vector fijo tal que $\mathbf{c} = (D-L)^{-1}\mathbf{b}$. Entonces, se plantea el algoritmo para este ejercicio

Algorithm 2 Método de Gauss-Seidel

Inicializar $x^{(0)}$ ▷ Vector de inicio
Inicializar tolerancia ϵ ▷ Tolerancia deseada
Inicializar número máximo de iteraciones $N_{\max}$
Inicializar contador de iteraciones $k \leftarrow 0$
while $k < N_{\max}$ **do**
    $k \leftarrow k + 1$
    for $i \leftarrow 1$ to n **do** ▷ Para cada componente de x
        $x_i^{(k)} \leftarrow \frac{1}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} x_j^{(k)} - \sum_{j>i} A_{ij} x_j^{(k-1)} \right)$ ▷ Actualización de x_i
    end for
    if $\|x^{(k)} - x^{(k-1)}\| < \epsilon$ **then** ▷ Condición de convergencia
        Salir del bucle
    end if
end while

2. Resultados

Tras la ejecución de los métodos, se llego a los siguientes resultados

3. Conclusiones

Bien, para el *smallSystem* se tiene que los métodos convergen de forma muy rápida, en este caso, para cuando el vector inicial es 0, el método de Gauss - Seidel es mejor que el método de Jacobi. Sin embargo, la situación cambia cuando se tiene el *bigSystem*, siendo que tras 50 iteraciones ninguno de los dos métodos converge, la situación aquí, aparentemente, dado que la aproximación es bastante

```
La matriz de entrada es:
3       -0.1    -0.2
0.1     7       -0.3
0.3     -0.2    10
El vector de entrada es:
7.85
-19.3
71.4
El metodo converge tras 9 iteraciones.
EL resultado solucion es
3
-2.5
7
La aproximacion es:
-1.17328e-12
-1.86873e-12
-1.77636e-12
```

Figura 1: Resultado de la aplicación del metodo de Jacobi a *smallSystem*.

```
La matriz de entrada es:
3       -0.1    -0.2
0.1     7       -0.3
0.3     -0.2    10
El vector de entrada es:
7.85
-19.3
71.4
El metodo converge tras 4 iteraciones.
EL resultado solucion es
3
-2.5
7
La aproximacion es:
-5.79753e-09
-3.40151e-09
0
```

Figura 2: Resultado de la aplicación del método de Gauss - Seidel a *smallSystem*.

buena, se presenta que al utilizarse la norma de distancias, al ser muchos valores, aunque pequeños, la suma acumulada de todos estos es difícil que se acerque a 0. En cambio si se utilizara la norma que sólo toma como norma la máxima distancia entre un par de elementos [**?**] de los vectores, esta convergería más rápidamente para ambos casos.

4. Referencias

[1] Burden, R.L., Faires, J.D. and Burden, A.M. 2015. *Numerical analysis.* Cengage learning.

[2] Cabrales, R.C. and Burgos Alarcon, D.A. 2016 *Métodos iterativos para sistemas de ecuaciones lineales, Universidad del Bío-Bío. Escuela de Pedagogía en Educación Matemática.* Available at: http://repobib.ubiobio.cl/jspui/handle/123456789/1228 (Accessed: 24 September 2023).

```
Despues de 50 iteraciones el metodo no converge.EL resultado solucion es
0.187428
0.188717
0.00673795
0.188227
0.188932
0.188177
0.130633
0.0827252
0.00673795
0.0408736
La aproximacion es:
-2.43989e-07
-4.94933e-07
0
-4.96833e-07
-5.20144e-07
-5.33108e-07
-8.57169e-07
-7.36956e-07
0
-3.73048e-07
```

Figura 3: Resultado de la aplicación del metodo de Jacobi a *bigSystem*.

```
Despues de 50 iteraciones el metodo no converge.EL resultado solucion es
0.29672
0.292942
0.00673795
0.29786
0.298864
0.297248
0.230004
0.164139
0.00673795
0.0844019
La aproximacion es:
-3.76835e-07
-7.13143e-07
0
-7.62881e-07
-7.87466e-07
-7.92659e-07
-1.23171e-06
-1.0572e-06
0
-5.25835e-07
```

Figura 4: Resultado de la aplicación del método de Gauss - Seidel a *bigSystem*.