Universidad de los Andes
Departamento de Ingeniería Eléctrica y Electrónica
IELE 2009L – Laboratorio de Computación Científica
Gerardo Andrés Riaño Briceño (201112388)

Informe de laboratorio 3

1. Validación del algoritmo

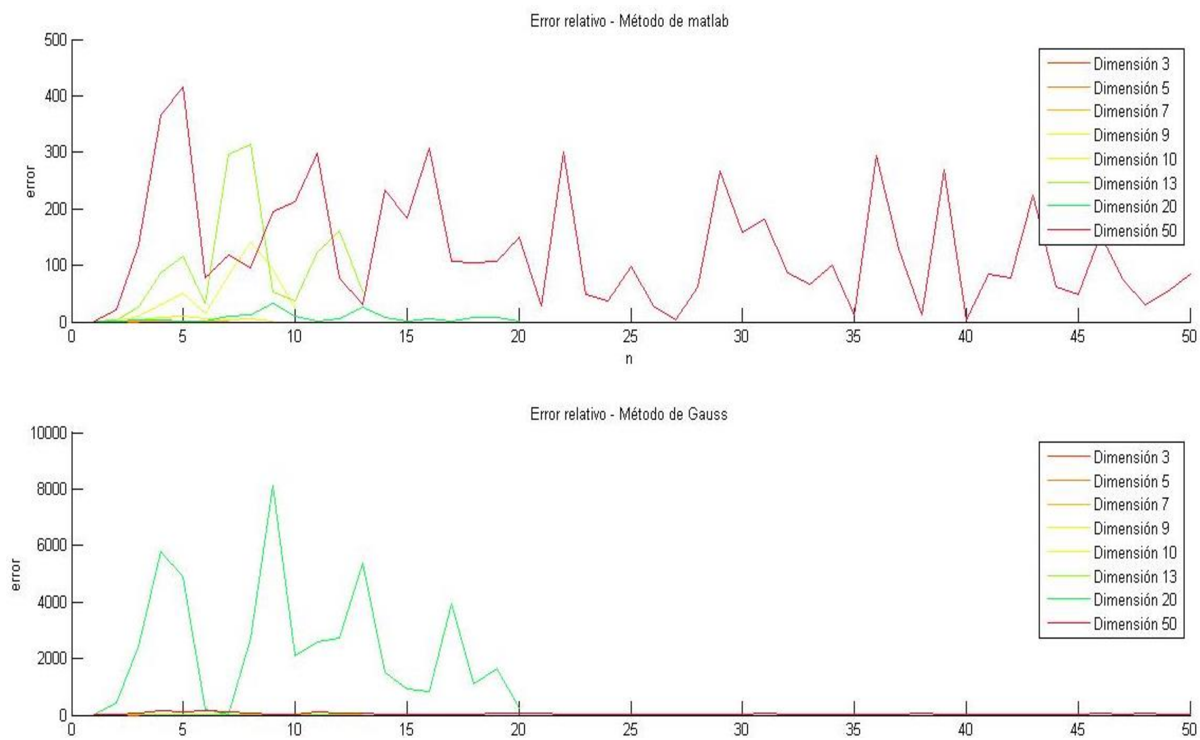

Figura 1. Error relativo de solución de sistemas de ecuaciones lineales.

Como puede observarse en la Figura 1, el error relativo aumenta a medida que aumenta la dimensión del sistema de ecuaciones. Cuando se utiliza el método de Matlab para resolver el sistema de ecuaciones, se observa que el error es mayor a cuando que en el caso en el que se usa el método de eliminación de Gauss con pivoteo parcial. A pesar de que se utiliza aritmética de punto flotante con 4 bytes en ambos casos, durante la ejecución de los métodos de Matlab, el programa arroja advertencias por mal condicionamiento de la matriz A, por lo cual, los errores que se obtienen son muy grandes debido a variaciones pequeñas en los resultados de las operaciones internas del programa. Es decir, que en el método de Matlab el error de redondeo se propaga y crece ampliamente a medida que resuelven las operaciones para llegar a la solución del sistema de ecuaciones

En el caso del método de Gauss, se tiene la ventaja de que los valores de la columna pivote se normalizan por el mayor valor relativo de los elementos de la fila, con lo cual, las operaciones internas se realizan entre números de la forma de punto flotante que no son tan cercanos entre sí, y es por esto que la propagación de error por redondeo no es crítica como en el otro método. Sin embargo, para el sistema de ecuaciones de dimensión 20, el error es bastante grande, lo cual sugiere que para éste sistema, sea necesario replantear el método de solución o aumentar la precisión numérica de los valores que se usan en la operación.

2. Aplicación del algoritmo

Siguiendo la convención propuesta en la guía y con los datos que se presentan a continuación, se obtienen las ecuaciones de nodos del circuito, las cuales conforman un sistema ecuaciones lineales con cuatro incógnitas.

Datos del circuito

$R_1 = 1k\Omega$	$R_4 = 4k\Omega$	$R_7 = 7k\Omega$	$I_C = 12\ \mathrm{m}A$
$R_2 = 2k\Omega$	$R_5 = 5k\Omega$	$V_a = 9V$	
$R_3 = 3k\Omega$	$R_6 = 6k\Omega$	$V_b = 8V$	

Ecuaciones de corriente

$$\boldsymbol{i_1} = \frac{V_A - V_b}{R_1} \qquad \boldsymbol{i_3} = \frac{V_F}{R_3} \qquad \boldsymbol{i_5} = \frac{V_C}{R_5} \qquad \boldsymbol{i_7} = \frac{V_A - V_C}{R_7}$$

$$\boldsymbol{i_2} = -\frac{V_E}{R_2} \qquad \boldsymbol{i_4} = \frac{V_A - V_E}{R_4} \qquad \boldsymbol{i_6} = \frac{V_A - V_C}{R_6}$$

Ecuaciones de nodos

$$I_C - i_1 - i_3 - i_4 - i_6 - i_7 = 0 \quad (\boldsymbol{Nodo\ A})$$

$$I_c - i_5 + i_6 + i_7 = 0 \quad (\boldsymbol{Nodo\ C})$$

$$i_2 + i_4 = 0 \quad (\boldsymbol{Nodo\ E})$$

$$V_A = V_a + V_F \ (\boldsymbol{Supernodo})$$

$$\boldsymbol{V_A}\left(-\frac{1}{R_1} - \frac{1}{R_4} - \frac{1}{R_6} - \frac{1}{R_7}\right) + \boldsymbol{V_C}\left(\frac{1}{R_6} + \frac{1}{R_7}\right) + \boldsymbol{V_F}\left(-\frac{1}{R_3}\right) + \boldsymbol{V_E}\left(\frac{1}{R_4}\right) = -I_C - \frac{V_b}{R_1}$$

$$\boldsymbol{V_A}\left(-\frac{1}{R_6} - \frac{1}{R_7}\right) + \boldsymbol{V_F}\left(\frac{1}{R_5} + \frac{1}{R_6} + \frac{1}{R_7}\right) = I_C$$

$$\boldsymbol{V_E}\left(-\frac{1}{R_2} - \frac{1}{R_4}\right) + \boldsymbol{V_A}\left(\frac{1}{R_4}\right) = 0$$

$$\boldsymbol{V_A} - \boldsymbol{V_F} = V_a$$

Forma matricial del sistema de ecuaciones

$$\mathbf{Ax} = \mathbf{b} \qquad A \in \mathrm{R}^{4x4}, \qquad x \in \mathrm{R}^{4x1}, \qquad b \in \mathrm{R}^{4x1}$$

$$\begin{pmatrix} \left(-\frac{1}{R_1}-\frac{1}{R_4}-\frac{1}{R_6}-\frac{1}{R_7}\right) & \left(\frac{1}{R_6}+\frac{1}{R_7}\right) & \left(-\frac{1}{R_3}\right) & \left(\frac{1}{R_4}\right) \\ \left(-\frac{1}{R_6}-\frac{1}{R_7}\right) & \mathbf{0} & \left(\frac{1}{R_5}+\frac{1}{R_6}+\frac{1}{R_7}\right) & \mathbf{0} \\ \left(\frac{1}{R_4}\right) & \mathbf{0} & \mathbf{0} & \left(-\frac{1}{R_2}-\frac{1}{R_4}\right) \\ \mathbf{1} & \mathbf{0} & \mathbf{-1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{V_A} \\ \boldsymbol{V_C} \\ \boldsymbol{V_F} \\ \boldsymbol{V_E} \end{pmatrix} = \begin{pmatrix} -I_C-\frac{V_b}{R_1} \\ -I_C \\ \mathbf{0} \\ V_a \end{pmatrix}$$

x = [-37.0714, -291.0330, -46.0714, -12.3571]

Descripción de los métodos de solución – se obtiene de la documentación de Matlab

- **Función rref(A)**
  Permite hallar el rango de la matriz A. Retorna una matriz reducida y un vector cuya dimensión es equivalente al rango de la matriz. Si la matriz es de rango completo la matriz reducida es igual a la identidad.

- **Función inv(A)*b**
  El programa halla la matriz inversa de A y luego realiza el producto entre la inversa y el vector b, para obtener el vector solución.

- **Función A\b**
  Utiliza diferentes métodos de solución dependiendo del tipo de matriz que se use. Si A no es cuadrada se utiliza un método de solución basado en descomposición QR. Si es cuadrada se revisa si es triangular, si no, se revisa si es triangular permutada, si no, se revisa si es Hermitiana. Si es Hermitiana, se revisa si los valores de la diagonal son reales y positivos, si es así, se usa un método de solución basado en la descomposición de Cholesky. Si es Hermitiana y los valores de la diagonal no son reales o no son positivos, se usa descomposición LDL. Si A no es Hermitiana, se revisa si es Hessenberg superior, si es así, se utiliza la solución de Hessenberg y si no, se utiliza descomposición LU.

- **Función linsolve(A, b)**
  Resuelve el sistema de ecuaciones a través de una descomposición LU, así que resuelve primero el sistema de la forma Ly = b, y con y resuelve el sistema Ux = y. Este método resulta eficiente ya que las matrices L y U son matrices triangulares, inferior y superior respectivamente, con lo cual pueden calcularse los valores del vector solución con sustitución hacia atrás.

Después de realizar las operaciones, con todos los métodos se obtiene la misma solución (utilizando una precisión de 4 cifras significativas). Sin embargo, para otro tipo de sistemas puede tenerse un mal condicionamiento del sistema de ecuaciones y puede haber grandes errores debido a la propagación de error de redondeo. Por ende, se puede inferir que el método que mayor error de redondeo puede producir es el que lleva a cabo más operaciones entre números muy cercanos.

Gerardo Andrés Riaño Briceño (201112388)

Entre los cuatro métodos estudiados, el que puede generar más error es el que consiste en hallar primero la matriz inversa y luego realizar el producto por b. Por otro lado, el método más eficiente y que garantiza menor error de redondeo es mldivide de Matlab, ya que se adapta al tipo de matriz para garantizar la mejor solución según su tipo.

Anexos – Código en Matlab

```
clc; clear all; close all;
%% Validación
    dimensiones = [3 5 7 9 10 13 20 50];

    % Definición de colores para gráficas
    cc = hsv(50);
    figure;

    for i = dimensiones
        [A, b] = generar(i);
        A = single(A);
        b = single(b);
        % Solución con método de matlab
        x_matlab = A\b;
        % Solución con algoritmo de eliminación gaussiana y sustitución
        %   hacia atrás
        [AA, bb] = gaussian_elimination(A,b);
        x_gauss = back_sustitution(AA,bb);
        % Solución real
        x_real = 1:i;

        error_matlab = abs(x_matlab - x_real')./abs(x_real');
        error_gauss = abs(x_gauss - x_real')./abs(x_real');

        % Gráfica error con método de matlab en función de la dimensión
del
        % sistema
        subplot(2,1,1); hold on;
        plot(error_matlab, 'color', cc(i,:), 'DisplayName',
sprintf('Dimensión %d', i));
        % Gráfica error con método de gauss en función de la dimensión
del
        % sistema
        subplot(2,1,2); hold on;
        plot(error_gauss, 'color', cc(i,:), 'DisplayName',
sprintf('Dimensión %d', i));
    end
    % Detalles primera gráfica
        subplot(2,1,1); legend('show');
        title('Error relativo - Método de matlab');
        xlabel('n'); ylabel('error');
    % Detalles segunda gráfica
        subplot(2,1,2); legend('show');
        title('Error relativo - Método de Gauss');
        xlabel('n'); ylabel('error');
```

Gerardo Andrés Riaño Briceño (201112388)

```
%% Aplicación

    R = (1:7)*1000; % Vector con valores de resistencias
    Va = 9;
    Vb = 8;
    Ic = 12E-3;

    % Sistema de ecuaciones lineales Ax = b

    A = [(-1/R(1)-1/R(4)-1/R(6)-1/R(7)),(1/R(6)+1/R(7)),(-1/R(3)),1/R(4);
         (1/R(6)+1/R(7)), 0, (-1/R(5)-1/R(6)-1/R(7)), 0;
         (1/R(4)), 0, 0, (-1/R(2)-1/R(4));
         1, 0, -1, 0];

    b = [(-Ic-Vb/R(1)); Ic; 0; Va];
    x = A\b;
    VA = x(1);
    VC = x(2);
    VF = x(3);
    VE = x(4);
    i1 = (VA - Vb)/R(1);
    i2 = -VE/R(2);
    i3 = VF/R(3);
    i4 = (VA - VE)/R(4);
    i5 = VC/R(5);
    i6 = (VA - VC)/R(6);
    i7 = (VA - VC)/R(7);

    % Comprobación con métodos de Matlab
    [X, jb] = rref(A);
    fprintf('\nLa matriz es de rango %d\n', length(jb));
    x_m1 = inv(A)*b;
    x_m2 = linsolve(A,b);
    x_m3 = A\b;

    % Comprobación con algoritmo de eliminación de Gauss con pivoteo
    % parcial

    [AA, bb] = gaussian_elimination(A,b);
    x_gauss = back_sustitution(AA,bb);
```