

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE  
ÁREA DE ANÁLISIS FUNCIONAL  
MÉTODOS NUMÉRICOS

Integrantes: Jefferson David Yépez Morán  
Fecha: 9 de julio de 2024

NRC:14188

Tarea 2.2

- Realice un programa que calcule las incógnitas de un sistema  $n \times n$ , y compare el tiempo de ejecución entre los métodos, Gauss parcial, LU e inversa.

- Resuelva sistemas de ecuaciones  $3 \times 3$ ,  $10 \times 10$ ,  $30 \times 30$ ,  $100 \times 100$  y  $200 \times 200$  para comparar la eficiencia de los métodos. Muestre además los resultados en una gráfica como el ejemplo de la siguiente figura:

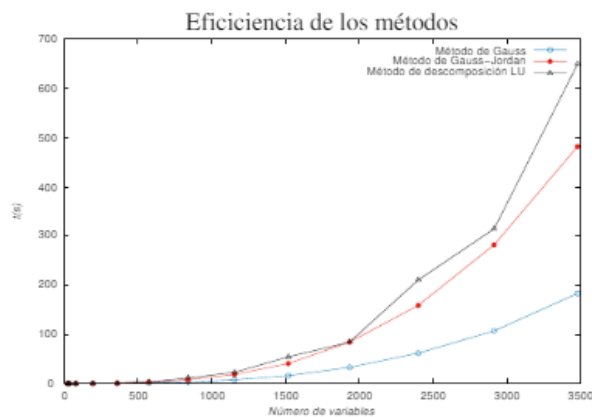


Figura 1: Enunciado

Para esta tarea, debe realizar una entrega:

Aula virtual: PDF realizado en LaTeX con los respaldos pertinentes. Además, adjuntar en el mismo documento el código fuente del algoritmo con el script solicitado. (No hay entrega en Gradescope).

Funcion:

```
1 %Autor: Jefferson Yepez
2
3 function mainTiempos()
4     comparar_tiempos();
5 end
6
7 function comparar_tiempos()
8     tamanos = [3, 10, 30, 100, 200];
9     tiempos_gauss = zeros(length(tamanos), 1);
10    tiempos_lu = zeros(length(tamanos), 1);
11    tiempos_inversa = zeros(length(tamanos), 1);
```

```

12
13     for i = 1:length(tamamos)
14         n = tamanos(i);
15         A = rand(n, n);
16         b = rand(n, 1);
17
18         % Tiempo para m todo de Gauss
19         tic;
20         gauss(A, b);
21         tiempos_gauss(i) = toc;
22
23         % Tiempo para m todo de descomposici n LU
24         tic;
25         [L, U] = lu_decomposition(A);
26         solve_lu(L, U, b);
27         tiempos_lu(i) = toc;
28
29         % Tiempo para m todo de la inversa
30         tic;
31         inversa(A, b);
32         tiempos_inversa(i) = toc;
33     end
34
35     % Graficar los resultados
36     figure;
37     plot(tamamos, tiempos_gauss, '-o', 'DisplayName', 'M todo de Gauss');
38     hold on;
39     plot(tamamos, tiempos_lu, '-x', 'DisplayName', 'M todo de descomposici n LU');
40     plot(tamamos, tiempos_inversa, '-s', 'DisplayName', 'M todo de la inversa');
41     xlabel('N mero de variables');
42     ylabel('Tiempo (s)');
43     title('Eficiencia de los m todos');
44     legend show;
45     hold off;
46 end
47
48 function x = gauss(A, b)
49     n = length(b);
50     % Eliminaci n Gaussiana
51     for k = 1:n-1
52         for i = k+1:n
53             factor = A(i,k) / A(k,k);
54             A(i,k:n) = A(i,k:n) - factor * A(k,k:n);
55             b(i) = b(i) - factor * b(k);
56         end
57     end
58     % Sustituci n regresiva
59     x = zeros(n, 1);
60     x(n) = b(n) / A(n,n);
61     for i = n-1:-1:1
62         x(i) = (b(i) - A(i,i+1:n) * x(i+1:n)) / A(i,i);
63     end
64 end
65
66 function [L, U] = lu_decomposition(A)
67     n = size(A, 1);
68     L = eye(n);
69     U = zeros(n);
70     for j = 1:n
71         for i = 1:j
72             U(i,j) = A(i,j) - L(i,1:i-1) * U(1:i-1,j);

```

```

73     end
74     for i = j+1:n
75         L(i,j) = (A(i,j) - L(i,1:j-1) * U(1:j-1,j)) / U(j,j);
76     end
77 end
78 end
79
80 function x = solve_lu(L, U, b)
81     % Sustitución progresiva para resolver  $Lc = b$ 
82     n = length(b);
83     c = zeros(n, 1);
84     for i = 1:n
85         c(i) = b(i) - L(i, 1:i-1) * c(1:i-1);
86     end
87     % Sustitución regresiva para resolver  $Ux = c$ 
88     x = zeros(n, 1);
89     for i = n:-1:1
90         x(i) = (c(i) - U(i, i+1:n) * x(i+1:n)) / U(i, i);
91     end
92 end
93
94 function x = inversa(A, b)
95     x = inv(A) * b;
96 end

```

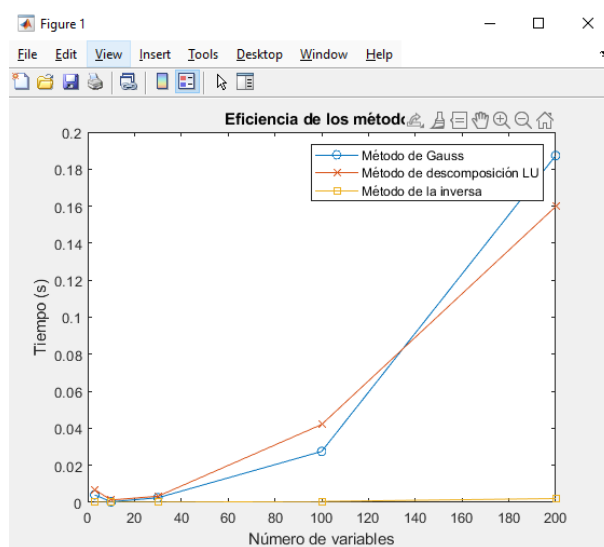


Figura 2: Enunciado