INFORMACIÓN BÁSICA

Nombre del Curso	Fecha de diligenciamiento(dd/mm/aaaa)	Sección(es)	Periodo académico
Computación Científica en IEE	15/02/2015	1-2	201610

Nombre de la práctica:	Cómputo de Matrices	**Práctica No.:**	3
Profesor(es):	Nestor Peña Traslaviña	**Asistente(es) Graduado(s):**	Daniel Felipe Duarte Sánchez

Semana de la práctica (1-16)	Versión de la guía	Nomenclatura del espacio a utilizar
5	2.0	ML-108A / ML-107

CONTENIDO DE LA GUÍA

Objetivos

- Reconocer algunos algoritmos de cómputo de soluciones de sistemas lineales.
- Identificar casos de cómputo de matrices que requieren consideraciones respecto a la unidad de punto flotante de la máquina.
- Reconocer casos de aplicación y los diversos métodos que ofrece Matlab para solución de sistemas lineales.

Procedimiento de la práctica de laboratorio

1. Algoritmos de Eliminación de Gauss con Pivoteo Parcial y Sustitución Regresiva

Implemente en MATLAB los algoritmos de eliminación gaussiana con pivoteo parcial (Script 1) y sustitución hacia atrás (Script 2). Asegúrese de entender el funcionamiento detallado de los algoritmos. En particular, tenga en cuenta las buenas prácticas de programación en cuanto al manejo matricial, el control de errores, la documentación del código y el formato de presentación.

```matlab
%%Gaussian elimination with partial pivoting
function [Ar br]=gaussian_elimination(A,b)
n=size(A,2);                         %Columns of A
for j=1:n-1                          %Loop over columns
    [pivot,k]=max(abs(A(j:n,j)));    %Find the pivot element in column j.
                                     %Pivot is the largest value of an entry;
                                     %k+j-1 is its index
    if(pivot<=eps)                     %If all entries in the column are 0,
        disp('Matrix is Singular');%return with an error message
        break;
    end
                                     %Otherwise
    temp=A(j,:);                     %Interchange rows j and k+j-1
    A(j,:)=A(k+j-1,:);
    A(k+j-1,:)=temp;

    tempb=b(j);
    b(j)=b(k+j-1);
    b(k+j-1)=tempb;

    for i=j+1:n                      %Loop over rows below j
        mult=A(i,j)/A(j,j);          %Subtract this multiple of row j from
                                     %row i to make A(i,j)=0
        A(i,j:n)= A(i,j:n) - mult*A(j,j:n);
        b(i)=b(i)- mult*b(j);

    end
end
Ar=A;
br=b;
end
```

Figure 1. Eliminación Gaussiana con pivoteo parcial[1]

```matlab
%%Back substitution
function X=back_substitution(A,B)

n=size(A,2);                                        %Columns of A

X=zeros(n,1);                                       %Initialization of X
X(n)=B(n)/A(n,n);                                   %Calculation of X(n)

for k=n-1:-1:1                                      %Backward Loop over columns

    X(k)= (B(k)-A(k,k+1:n)*X(k+1:n))/A(k,k);       %Substract from B(k) all x
                                                    %found until iteration k
                                                    %multiplied by its
                                                    %coefficients. Divide by
                                                    %the coeffcient of X(k);
                                                    %that is A(k,k).
end
end
```

Figure 2. Sustitución Regresiva[2]

2. Validación

Considere el sistema descrito por las siguientes ecuaciones:

$$A = [a_{ij}] \quad 1 \leq i.j \leq n \quad donde \quad a_{ij} = \frac{1}{(i+j-1)} \quad para \;\; i,j = 1,2,3,\dots,n.$$

$$b = [b_i] \quad 1 \leq i \leq n \quad donde \quad b_i = \sum_{j=1}^{n} \frac{j}{(i+j-1)} \quad para \;\; i = 1,2,3,\dots,n.$$

La solución del sistema lineal de ecuaciones $A\boldsymbol{x} = \boldsymbol{b}$ es $\boldsymbol{x} = [1,2,3,\dots,n]$. Es decir, $x_i = i$ para $i = 1,2,3,\dots,n$.

Genere un script, con parámetro de entrada n, donde calcule A y b. Trabaje con precisión sencilla (Hint: Use la función *single ()*). Luego utilice el algoritmo implementado en el punto anterior para calcular la solución del sistema $A\boldsymbol{x} = \boldsymbol{b}$.

Para varios valores de n (i.e $3,5,7,9,10,13,20,50$) compare la solución obtenida por el algoritmo con la solución exacta (Hint: Visualice los resultados en formato *long*). Concluya al respecto.

3. Aplicación

Obtenga el sistema de ecuaciones de nodos para el circuito resistivo de la figura 3. Utilice los algoritmos implementados para resolverlo (halle todos los voltajes y corrientes). Verifique su respuesta.
Compare con las funciones propias de MATLAB para la resolución de sistemas de ecuaciones lineales:

- *rref()*
- *x=inv(A)*b*
- *linsolve()*
- *x=A\b* ó *mldivide()*

Para usar estas funciones consulte la documentación de MATLAB con el comando help seguido del nombre de la función. Para obtener información más detallada busque la función en el panel de ayuda (Help -> Function Browser). Tenga en cuenta los parámetros de entrada, salida y las condiciones de cada método. Determine el método que utiliza cada función para hallar la solución del sistema.

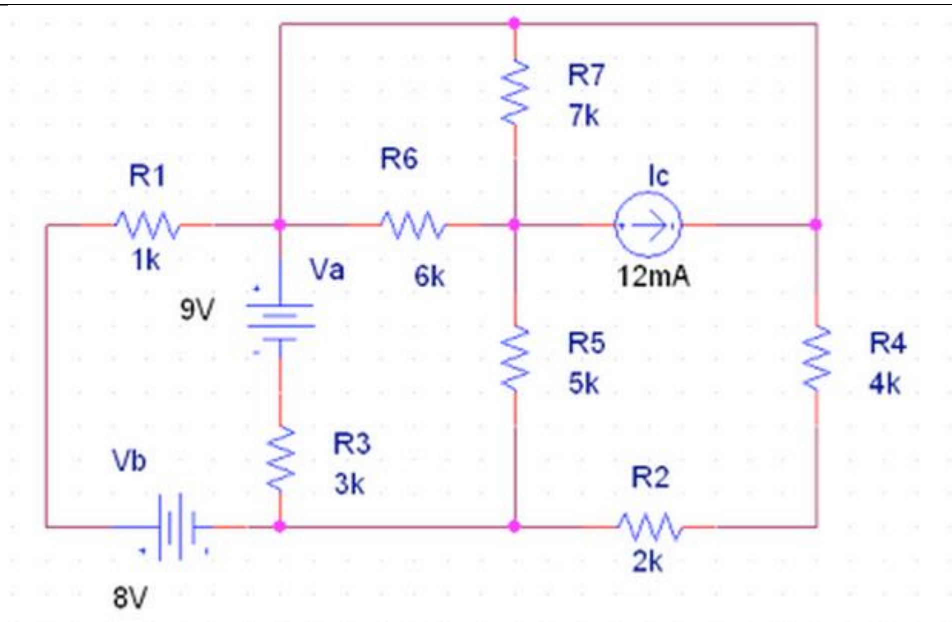

Figure 3. Circuito Resistivo

Bibliografía recomendada

[1] Greenbaum, Anne, and Timothy P. Chartier. Numerical methods: design, analysis, and computer implementation of algorithms. Princeton, N.J.: Princeton University Press, 2012.

[2] Mathews, John H., and Kurtis D. Fink. Métodos numéricos con MATLAB. 3a ed. Madrid [etc.: Prentice Hall, 2000.