

Tarea 3

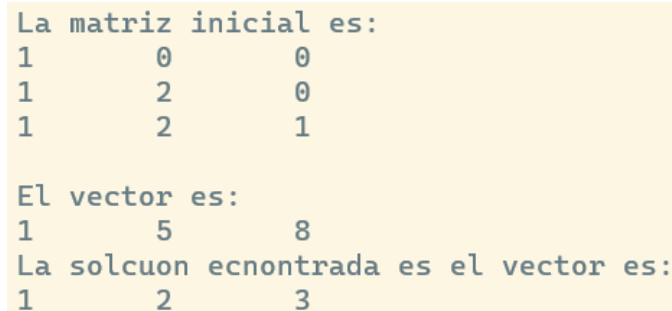


Nombre:	Jairo Saul Diaz Soto
Maestría:	Ciencias Computacionales
Modulo:	Métodos Numéricos
Instructor:	Dr. Luis Daniel Blanco Cocom
Fecha de entrega:	2023 - 09 - 10

1. Se realizó en C una rutina la cual resuelve un sistema de ecuaciones donde se tiene una matriz triangular inferior, tal que $Lx = b$, la subrutina se muestra a continuación, además se probó una matriz de ejemplo para corroborar su buen funcionamiento.

```
double *lSolve(double **matriz, double *vec, int sz) {  
    // Definimos una variable temporal  
    double temp = 0.0;  
  
    // Definimos el espacio para la solución  
    double *X;  
    X = (double *)malloc(sz * sizeof(double));  
  
    // Asignamos el valor del primer elemento, la parte trivial  
    X[0] = vec[0]/matriz[0][0];  
  
    // Recorremos para los valores de cada renglón  
    for (int i = 1; i < sz; i++) {  
        temp = 0.0;  
        // Obtenemos la parte a restar  
        for (int j = 0; j < i; j++) {  
            temp += matriz[i][j] * X[j];  
        }  
        // Asignamos el valor  
        X[i] = (vec[i] - temp) / matriz[i][i];  
    }  
    // Regresamos la dirección de memoria donde se almacena  
    return X;  
}
```

Listing 1: Método de solución para un sistema con una matriz triangular inferior.



```
La matriz inicial es:  
1      0      0  
1      2      0  
1      2      1  
  
El vector es:  
1      5      8  
La solución encontrada es el vector es:  
1      2      3
```

Figura 1: Resultado de la aplicación del método para una matriz triangular inferior.

2. Se realizó en C una rutina la cual resuelve un sistema de ecuaciones donde se tiene una matriz triangular superior, tal que $Ux = b$, la subrutina se muestra a continuación, además se probó una matriz de ejemplo para corroborar su buen funcionamiento.

```
double *uSolve(double **matriz, double *vec, int sz) {  
    // Definimos una variable temporal  
    double temp = 0.0;  
  
    // Definimos el espacio para la solución  
    double *X;  
    X = (double *)malloc(sz * sizeof(double));  
  
    // Recorremos para los valores de cada columna  
    for (int j = 1; j < sz; j++) {  
        temp = 0.0;  
        for (int i = j; i < sz; i++) {  
            temp += matriz[i][j] * X[i];  
        }  
        X[j] = (vec[j] - temp) / matriz[j][j];  
    }  
    return X;  
}
```

```

// Asignamos el valor del ultimo elemento, la parte trivial
X[sz-1] = vec[sz-1]/matriz[sz-1][sz-1];

// Recorremos para los valores de cada renglón
for (int i = sz-2; i >= 0; i--) {
    temp = 0.0;
    // Obtenemos la parte a restar
    for (int j = i; j < sz; j++) {
        temp += matriz[i][j] * X[j];
    }
    // Asignamos el valor
    X[i] = (vec[i] - temp) / matriz[i][i];
}
// Regresamos la dirección de memoria donde se almacena
return X;
}

```

Listing 2: Método de solución para un sistema con una matriz triangular superior.

La matriz inicial es:

1	2	1
0	2	1
0	0	1

El vector es:

8	5	1
---	---	---

La solución encontrada es el vector es:

3	2	1
---	---	---

Figura 2: Resultado de la aplicación del método para una matriz triangular superior.

- Se realizó en C una rutina la cual resuelve un sistema de ecuaciones mediante el método de eliminación gaussiana, la subrutina se muestra a continuación, además se probó una matriz de ejemplo para corroborar su buen funcionamiento.

```

//Subrutina de eliminacion gaussiana
double *gauss(double **matriz, double *vec, int sz){
    //Definimos el espacio donde se guarda la solución
    double *X=(double *)malloc(sz*sizeof(double));
    //definimos el multiplicador
    double m=0.0;
    //Comenzamos con el proceso de eliminacion
    for(int i=0;i<sz-1;i++){
        for(int j=i+1; j<sz; j++){
            m=matriz[j][i]/matriz[i][i];
            for(int k=0;k<sz;k++){
                matriz[j][k]=matriz[j][k]-(m*matriz[i][k]);
            }
            vec[j]=vec[j]-(vec[i]*m);
        }
    }

    //Procedemos con la solución
    X=uSolve(matriz,vec,sz);
}

```

```

    return X;
}

```

Listing 3: Método de eliminación gaussiana.

```

La matriz inicial es:
2      1      -1
3      2      2
4      3      3

El vector es:
8      14      18
La solución encontrada es:
6      -3      1

```

Figura 3: Resultado de la aplicación del método de eliminación gaussiana.

4. Se realizó en C una rutina la cual resuelve un sistema de ecuaciones mediante el método de eliminación gaussiana, con la modificación del pivoteo parcial, la subrutina se muestra a continuación, además se probó una matriz de ejemplo para corroborar su buen funcionamiento.

```

double *gauss_parc(double **matriz, double *vec, int sz){
    //Definimos el espacio donde se guarda la solución
    double *X=(double *)malloc(sz*sizeof(double));
    //definimos el multiplicador
    double m=0.0;
    //Comenzamos con el proceso de eliminación
    for(int i=0;i<sz-1;i++){

        // Encontrar el pivote parcial (el elemento de mayor valor
        // absoluto en la columna)
        double maxPivote = fabs(matriz[i][i]);
        int filaPivote = i;

        for (int k = i + 1; k < sz; k++) {
            if (fabs(matriz[k][i]) > maxPivote) {
                maxPivote = fabs(matriz[k][i]);
                filaPivote = k;
            }
        }

        // Intercambiar filas si es necesario (pivoteo parcial)
        double *temp;
        double temps;
        if (filaPivote != i) {
            temp = matriz[i];
            matriz[i] = matriz[filaPivote];
            matriz[filaPivote]=temp;
            temps = vec[i];
            vec[i] = vec[filaPivote];
            vec[filaPivote] = temps;
        }

        for(int j=i+1; j<sz; j++){

```

```

        m=matriz[j][i]/matriz[i][i];
        for(int k=0;k<sz;k++){
            matriz[j][k]=matriz[j][k]-(m*matriz[i][k]);
        }
        vec[j]=vec[j]-(vec[i]*m);
    }
}

//Procedemos con la solucion
X=uSolve(matriz,vec,sz);
return X;
}

```

Listing 4: Método de eliminación gaussiana con pivoteo parcial.

```

La matriz inicial es:
2      1      -1
3      2       2
4      3       3

El vector es:
8      14      18
La solución encontrada es:
6      -3       1

```

Figura 4: Resultado de la aplicación del método de eliminación gaussiana con pivoteo parcial.

5. a) El primer ejercicio se resolvió con eliminación gaussiana y con eliminación gaussiana con pivoteo parcial:

```

La matriz inicial es:
9      7      2      3      2
1      10     9      10     9
3      8      9      1      8
3      4      2      1      4
8      8      1      8      8

El vector es:
51     133     90     49     99
La solución encontrada es:
-6.04322      15.6692 -5.32191      -0.479881      3.89419

```

Figura 5: Solución por el método de Gauss.

- b) El segundo ejercicio también se resolvió con eliminación gaussiana y con eliminación gaussiana con pivoteo parcial:
c) El tercer ejercicio se resolvió con el método de matriz inferior:
d) El cuarto ejercicio se resolvió con el método de matriz superior:

```

La matriz inicial es:
9      7      2      3      2
1      10     9      10     9
3      8      9      1      8
3      4      2      1      4
8      8      1      8      8

El vector es:
51     133     90     49     99
La solución encontrada es:
-6.04322      15.6692 -5.32191      -0.479881      3.89419

```

Figura 6: Solución por el método de Gauss con pivoteo parcial.

```

La matriz inicial es:
2      20      18      20      18
6      16      18      2      16
18     14      4      6      4
16     16      2      16     16
6      10      4      2      8

El vector es:
532     360     204     396     172
La solución encontrada es:
2      4      6      8      10

```

Figura 7: Solución por el método de Gauss.

```

La matriz inicial es:
2      20      18      20      18
6      16      18      2      16
18     14      4      6      4
16     16      2      16     16
6      10      4      2      8

El vector es:
532     360     204     396     172
La solución encontrada es:
2      4      6      8      10

```

Figura 8: Solución por el método de Gauss con pivoteo parcial.

```

La matriz inicial es:
1      0      0      0      0
0.11111 1      0      0      0
0.33333 0.61446 1      0      0
0.33333 0.28916 -0.40984 1      0
0.88889 0.19277 -0.84016 0.29075 1

El vector es:
51      133      90      43      99
La solcuon ecnontrada es el vector es:
51      127.333 -5.2411 -12.9676      28.4875

```

Figura 9: Método de solución para matriz triangular inferior

```

La matriz inicial es:
9      7      2      3      2
0      9.22222 8.77778 9.66667 8.77778
0      0      2.93976 -5.93976      1.93976
0      0      0      -5.22951      1.59016
0      0      0      0      5.69749

El vector es:
51      127.333 -5.241 -12.9672      28.4875
La solcuon ecnontrada es el vector es:
9.0775e+199      -1.16711e+200 2.99997 4      5.00001

```

Figura 10: Método de solución para matriz triangular superior