Programación:
Solución de sistemas de ecuaciones lineales usando la eliminación de Gauss con pivotes diagonales

Objetivos. Programar el método de eliminación de Gauss sin pivoteo (esto es, con pivotes diagonales). Programar la solución de un sistema de ecuaciones lineales.

Requisitos. Operaciones con matrices, programación de operaciones elementales, sustitución hacia atrás.

Eliminación de Gauss con pivotes diagonales

1. Algoritmo Reduce1 (pseudocódigo).

```
Entrada: matriz A.
Variables locales: B, m, n, k, i, j;
m := número de renglones de A;
n := número de columnas de A;
B := una copia de A;
Para p := 1, ..., m - 1:
    // Comentario: usamos B[p, p] como pivote
    Para i := p + 1, ..., m:
        mu := - B[i, p] / B[p, p];
        B[i, p] := 0;
        Para j := p + 1, ..., n:
            B[i, j] := B[i, j] + mu * B[p, j];
Salida: B.
```

2. Eliminación de Gauss con pivotes diagonales. Escriba una función `Reduce1` de un argumento matricial que reduzca la matriz dada.

> Entrada: una matriz rectangular A de tamaño $m \times n$, $m \le n$.
>
> Salida: la matriz triangular superior B que se obtiene de la matriz A al aplicar la eliminación de Gauss.

Se supone que a la matriz A se puede aplicar el método de eliminación de Gauss con pivotes diagonales, es decir, para cada p en el paso número p la entrada (p, p) es no nula.

3. Comprobación. Aplique la función `Reduce1` a la siguiente matriz:

$$A = \begin{bmatrix} 2 & -1 & -2 & 3 & 1 \\ 4 & 1 & -2 & 1 & 5 \\ -2 & -2 & -1 & 4 & 3 \end{bmatrix}. \qquad \text{Respusta correcta: } \begin{bmatrix} 2 & -1 & -2 & 3 & 1 \\ 0 & 3 & 2 & -5 & 3 \\ 0 & 0 & -1 & 2 & 7 \end{bmatrix}.$$

Juntar datos en una matriz aumentada.
Sacar datos de una matriz aumentada

Ejemplo: primer método para juntar una matriz y un vector en la matriz aumentada. Construir una matriz nula y llenar sus partes usando subconjuntos de índices:

```
a = {{1, 2}, {3, 4}}; b = {-1, -2};

aext = Table[0, {2}, {3}]

aext[[All, Range[1, 2]]] = a; aext[[All, 3]] = b; MatrixForm[aext]
```

Ejemplo: segundo método para juntar una matriz y un vector en la matriz aumentada. Usar la función `MapThread`:

```
a = {{1, 2}, {3, 4}}; b = {-1, -2};

aext = MapThread[Append, {a, b}]
```

Para comprender que hace `MapThread`, se recomienda probar el siguiente ejemplo:

```
Plus[5, 3]

MapThread[Plus, {{100, 200, 300}, {4, 5, 6}}]
```

Ejemplo: primer método para sacar de la matriz aumentada la matriz de coeficientes y el vector de constantes. Usar subconjuntos de índices:

```
aext = {{1, 2, 3, 4}, {5, 6, 7, 8}}

aext[[All, Range[1, 3]]]

aext[[All, 4]]
```

Ejemplo: segundo método para sacar de la matriz aumentada la matriz de coeficientes y el vector de constantes. Aplicar las funciones `Most` y `Last` a cada fila:

```
aext = {{1, 2, 3, 4}, {5, 6, 7, 8}}

Map[Most, aext]

Map[Last, aext]
```

Solución de sistemas de ecuaciones lineales usando la eliminación de Gauss con pivotes diagonales

4. Problema: LinSolve1. Escriba una función `LinSolve1[A_, b_]` que resuelva el sistema de ecuaciones lineales $Ax = b$ usando las funciones `Reduce1` y `SolveUT`.

5. Haga la comprobación de la función `LinSolve1` con

$$\mathtt{A} = \begin{bmatrix} 2 & -1 & -2 \\ 4 & 1 & -2 \\ -2 & -2 & -1 \end{bmatrix}, \qquad \mathtt{b} = \begin{bmatrix} 7 \\ 9 \\ -4 \end{bmatrix}.$$

La respuesta correcta es $(4, -3, 2)$.

Comprobación del método de Gauss con datos aleatorios

6. RandVector. Escriba una función `RandVector[n_, vmin_, vmax_]` que construya un vector aleatorio de longitud n con componentes reales pertenecientes al intervalo $[vmin, vmax]$.

7. RandMatrix. Escriba una función `RandMatrix[m_, n_, vmin_, vmax_]` que construya una matriz aleatoria de tamaño $m \times n$ con entradas reales pertenecientes al intervalo $[vmin, vmax]$.

8. Comprobación con matrices pequeñas. Aplique la función `LinSolve1` a una matriz aleatoria y un vector aleatorios de tamaño pequeño:

```
a = RandMatrix(3, 3, -5, 5); b = RandMatrix(3, -5, 5)

x = LinSolve1(a, b)

a . x

Norm[a . x - b]
```

9. Comprobación con matrices grandes. Aplique la función `LinSolve1` a una matriz aleatoria y un vector aleatorio de tamaño grande (por ejemplo, $n = 1000$). Para estimar el error, calcule `Norm[a . x - b]`.