Solve a set of linear algebraic equations with Gauss-Seidel iteration Method.
Instructor: Nam Sun Wang

Define the Gauss-Seidel algorithm for $A \cdot x = b$
A=square matrix
b=column vector
x_0=vector of initial guess (not needed, because there is only one solution for a linear system)
ε=tolerance in x
N=maximum number of iterations (already set to N=999)

```
gaussseidel(A,b,ε) :=  "start with x=0 ------------------------------------------------------------------------"
                       n ← last(b)
                       x_n ← 0
                       "iterate with Gauss-Seidel's formula up to a maximum of 999 times --------"
                       for iterate ∈ 1 .. 999
                           "save x before writing over it"
                           x old ← x
                           for j ∈ 0 .. n
                               x_j ← 1/A_{j,j} · ( b_j − Σ_{k=0}^{n} if(k=j, 0, A_{j,k}·x_k) )
                           "done if each & every element of x does not change much (relative error is small
                           break if [ Π_{i=0}^{n} | (x_i − x old_i) / x_i | < ε ]
                       return x
```

$$x_j \leftarrow \frac{1}{A_{j,j}} \cdot \left(b_j - \sum_{k=0}^{n} \mathrm{if}(k = j, 0, A_{j,k} \cdot x_k) \right)$$

$$\text{break if } \left[\prod_{i=0}^{n} \left| \frac{x_i - x\,old_i}{x_i} \right| < \varepsilon \right]$$

In the function definition below, we check for absolute error in x.

```
gaussseidel(A,b,ε) :=  "start with x=0 -----------------------------------------------------------------------"
                       n ← last(b)
                       x_n ← 0
                       "iterate with Gauss-Seidel's formula up to a maximum of 999 times --------"
                       for iterate ∈ 1 .. 999
                           "save x before writing over it"
                           x_old ← x
                           for j ∈ 0 .. n
                               x_j ← (1/A_{j,j}) · ( b_j − Σ_{k=0}^{n} if(k=j, 0, A_{j,k}·x_k) )
                           "done if x does not change much (i.e., absolute change-error is small enough )"
                           break if |x − x_old| < ε
                       return x
```

Another definition of Gauss-Seidel's algorithm, with a slightly stopping criterion based on the vertical error (and there is no need to save the old value of x)

```
gaussseidel2(A,b,ε) := "start with x=0 -----------------------------------------------------------------------"
                       n ← last(b)
                       x_n ← 0
                       "iterate with Gauss-Seidel's formula up to a maximum of 999 times --------"
                       for iterate ∈ 1 .. 999
                           for j ∈ 0 .. n
                               x_j ← (1/A_{j,j}) · ( b_j − Σ_{k=0}^{n} if(k=j, 0, A_{j,k}·x_k) )
                           "done if Ax is sufficiently close to b"
                           break if |A·x − b| < ε
                       return x
```

Example

check

$$A := \begin{bmatrix} 5 & 0 & 6 \\ 3 & -4 & 0 \\ 0 & 3 & 5 \end{bmatrix} \quad b := \begin{bmatrix} -0.329193 \\ -2.34066 \\ 1.20736 \end{bmatrix} \quad x := \mathrm{gaussseidel}\left(A, b, 10^{-7}\right) \quad x = \begin{bmatrix} 0.143 \\ 0.692 \\ -0.174 \end{bmatrix} \quad A \cdot x = \begin{bmatrix} -0.3291932 \\ -2.34066 \\ 1.20736 \end{bmatrix}$$

However, the above definition cannot handle an arbitrary A that has 0 in the diagonal elment $A_{j,j}=0$

$$A := \begin{bmatrix} 0 & 3 & 5 \\ 3 & -4 & 0 \\ 5 & 0 & 6 \end{bmatrix} \qquad b := \begin{bmatrix} 1.20736 \\ -2.34066 \\ -0.329193 \end{bmatrix} \qquad x := \text{gaussseidel}\left(A, b, 10^{-7}\right)$$

The definition below adds the row-swapping steps

```
gaussseidel(A,b,ε) := | "start with x=0 ------------------------------------------------------------------------"
                      | n ← last(b)
                      | x_n ← 0
                      | "swap rows in A & b"
                      | for j ∈ 0 .. n
                      |   | "Find the largest element in each column"
                      |   | Amax ← 0
                      |   | for i ∈ j .. n
                      |   |   if Amax < |A_{i,j}|
                      |   |     | Amax ← |A_{i,j}|
                      |   |     | imax ← i
                      |   | "swap imax-th row with jth row"
                      |   | for k ∈ 0 .. n
                      |   |   | temp ← A_{imax,k}
                      |   |   | A_{imax,k} ← A_{j,k}
                      |   |   | A_{j,k} ← temp
                      |   | temp ← b_imax
                      |   | b_imax ← b_j
                      |   | b_j ← temp
                      | "iterate with Gauss-Seidel's formula up to a maximum of 999 times --------"
                      | for iterate ∈ 1 .. 999
                      |   | "save x before writing over it"
                      |   | x_old ← x
                      |   | for j ∈ 0 .. n
                      |   |   x_j ← (1 / A_{j,j}) · ( b_j − Σ_{k=0}^{n} if(k=j, 0, A_{j,k}·x_k) )
                      |   | "done if x does not change much (i.e., absolute change-error is small enough )"
                      |   | break if |x − x_old| < ε
```

$$x_j \leftarrow \frac{1}{A_{j,j}} \cdot \left(b_j - \sum_{k=0}^{n} \text{if}\left(k = j, 0, A_{j,k} \cdot x_k\right) \right)$$

return x

Example.

check

$$A := \begin{bmatrix} 0 & 3 & 5 \\ 3 & -4 & 0 \\ 5 & 0 & 6 \end{bmatrix} \quad b := \begin{bmatrix} 1.20736 \\ -2.34066 \\ -0.329193 \end{bmatrix} \quad x := \text{gaussseidel}\left(A, b, 10^{-7}\right) \quad x = \begin{bmatrix} 0.143 \\ 0.692 \\ -0.174 \end{bmatrix} \quad A \cdot x = \begin{bmatrix} 1.20736 \\ -2.34066 \\ -0.3291932 \end{bmatrix}$$

Iterate with matrix formula (rather than scalar formula)

$A = I + L + U$ separate the given matrix A into different parts

$A \cdot x = (I + L + U) \cdot x = b$ The "x" in Lx comes from the new iteration; whereas, the "x" in Ux comes from the old iteration.

$x = b - L \cdot x - U \cdot x$

$(I + L) \cdot x = b - U \cdot x$

$x = (I + L)^{-1} \cdot (b - U \cdot x)$ ... Gauss-Seidel iteration formula in matrix form

```
gaussseidel3(A,b,ε) :=
  "start with x=0 -----------------------------------------------------------------------"
  n ← last(b)
  x_n ← 0
  "swap rows in A & b"
  for j ∈ 0..n
      "Find the largest element in each column"
      Amax ← 0
      for i ∈ 0..n
        if Amax < |A_{i,j}|
            Amax ← |A_{i,j}|
            imax ← i
      "swap imax-th row with jth row"
      for k ∈ 0..n
          temp ← A_{imax,k}
          A_{imax,k} ← A_{j,k}
          A_{j,k} ← temp
      temp ← b_imax
      b_imax ← b_j
      b_j ← temp
  "normalize each row to the diagonal element"
  for i ∈ 0..n
      diagonal ← A_{i,i}
      for j ∈ 0..n
          A_{i,j} ← A_{i,j} / diagonal
      b_i ← b_i / ...
```

```
    diagonal
"decompose A=I+L+U=L'+U (not LU decomposition)"
L' ← A
for i ∈ 0 .. n − 1
  for j ∈ i + 1 .. n
    U[i,j] ← L'[i,j]
    L'[i,j] ← 0
U[n,n] ← 0
"iterate with Gauss-Seidel's formula up to a maximum of 999 times --------"
for iterate ∈ 1 .. 999
  "save x before writing over it"
  x_old ← x
  x ← L'^-1·(b − U·x)
  "done if x does not change much (i.e., absolute change-error is small enough )"
  break if |x − x_old| < ε
return x
```

Example.

check

$$A := \begin{bmatrix} 0 & 3 & 5 \\ 3 & -4 & 0 \\ 5 & 0 & 6 \end{bmatrix} \quad b := \begin{bmatrix} 1.20736 \\ -2.34066 \\ -0.329193 \end{bmatrix} \quad x := \text{gaussseidel3}\left(A, b, 10^{-7}\right) \quad x = \begin{bmatrix} 0.143 \\ 0.692 \\ -0.174 \end{bmatrix} \quad A \cdot x = \begin{bmatrix} 1.20736 \\ -2.34066 \\ -0.3291932 \end{bmatrix}$$

Jacobi iteration is another scheme closely related to Gauss-Seidel. Within each iteration, the x variables are updated sequentially in Gauss-Seidel; whereas, the x variables are all updated simulteneously in Jacobi.

$A = I + L + U$ separate the given matrix A into different parts

$A \cdot x = x - (I - A) \cdot x = b$

$x = b + (I - A) \cdot x$ ... Jacobi iteration formula in matrix form

```
jacobi3(A,b,ε) :=  "start with x=0 -------------------------------------------------------------------------"
                   n ← last(b)
                   x_n ← 0
                   "swap rows in A & b"
                   for j ∈ 0 .. n
                       "Find the largest element in each column"
                       Amax ← 0
                       for i ∈ j .. n
                         if Amax < |A_i,j|
                             Amax ← |A_i,j|
                             imax ← i
                       "swap imax-th row with jth row"
                       for k ∈ 0 .. n
                           temp ← A_imax,k
                           A_imax,k ← A_j,k
                           A_j,k ← temp
                       temp ← b_imax
                       b_imax ← b_j
                       b_j ← temp
                   "normalize each row to the diagonal element"
                   for i ∈ 0 .. n
                       diagonal ← A_i,i
                       for j ∈ 0 .. n
                         A_i,j ← A_i,j / diagonal
                       b_i ← b_i / diagonal
                       I_i,i ← 1
```

$_{i,i}$

"iterate with Gauss-Seidel's formula up to a maximum of 999 times --------"

for iterate $\in$ 1 .. 999

"save x before writing over it"

$x_{old} \leftarrow x$

$x \leftarrow b + (I - A)\cdot x$

"done if x does not change much (i.e., absolute change-error is small enough)"

break if $|x - x_{old}| < \varepsilon$

return x

Example.

check

$$A := \begin{bmatrix} 0 & 3 & 5 \\ 3 & -4 & 0 \\ 5 & 0 & 6 \end{bmatrix} \quad b := \begin{bmatrix} 1.20736 \\ -2.34066 \\ -0.329193 \end{bmatrix} \quad x := \text{jacobi3}\left(A, b, 10^{-7}\right) \quad x = \begin{bmatrix} 0.143 \\ 0.692 \\ -0.174 \end{bmatrix} \quad A\cdot x = \begin{bmatrix} 1.20736 \\ -2.3406602 \\ -0.3291934 \end{bmatrix}$$