TP4 Solving Linear Systems of Equations

1 Solve $AX = B$ for X where

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 \\ -2 & 2 & -3 & 0 \\ 1 & -2 & 2 & 4 \end{pmatrix}, B = \begin{pmatrix} -4 \\ 8 \\ 5 \\ 8 \end{pmatrix}$$

2 Solve $AX = B$ for X where

$$A = \begin{pmatrix} a_{1,1} & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & 0 & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

assuming that $a_{i,i} \neq 0$ for all $i = 1, 2, 3, 4$.

3 Write a general purpose algorithm for Forward Substitution Method.

- **a** Describe your algorithm objective?
- **b** Describe required input parameter(s)?
- **c** Describe output value(s)?
- **d** Write your algorithm body here.
- **e** Implement your developed algorithm with Python, or any other programming language.

4 Solve $AX = B$ for X where

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}, B = \begin{pmatrix} -1 \\ 1 \\ -2 \\ 2 \\ -3 \\ 3 \\ -4 \\ 4 \\ -5 \\ 5 \end{pmatrix}$$

- **a** Using the code of forward substitution method developed in the previous exercise.
- **b** Verify your result with `numpy.linalg.solve`, or `scipy.linalg.solve` from the `numpy` and `scipy` packages respectively.
- **c** Can you solve it manually?

5 Write a general purpose algorithm for Backward Substitution Method.

- **a** Describe your algorithm objective?
- **b** Describe required input parameter(s)?
- **c** Describe output value(s)?

d Write your algorithm body here.

e Implement your developed algorithm with Python, or any other programming language.

6 Use code developed from previous exercise to solve $AX = B$ for X where

$$A = \begin{pmatrix} 2 & -1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}, B = \begin{pmatrix} 13 \\ -19 \\ 25 \\ -31 \\ 37 \\ -43 \\ 49 \\ -25 \\ 18 \end{pmatrix} .X = \begin{pmatrix} 1 \\ -2 \\ 3 \\ -4 \\ 5 \\ -6 \\ 7 \\ -8 \\ 9 \end{pmatrix}$$

7 Use Gauss elimination to solve the equations $AX = B$, where

$$A = \begin{pmatrix} 2 & -3 & -1 \\ 3 & 2 & -5 \\ 2 & 4 & -1 \end{pmatrix}, B = \begin{pmatrix} 3 \\ -9 \\ -5 \end{pmatrix}$$

8 Write a general purpose algorithm for Gauss Elemination Method.

a Describe your algorithm objective?

b Describe required input parameter(s)?

c Describe output value(s)?

d Write your algorithm body here.

e Implement your developed algorithm with Python, or any other programming language.

9 Use the code developed from previous exercise to solve $AX = B$ for X where

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}.$$

a After elimination stage, $AX = B$ is transformed to $UX = C$.
Determine the upper triagular matrix U and the vector C.

b Use the code of backward substitution method to get X.

10 Let

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ 2 & -3 & 2 & 0 & 0 \\ 0 & 4 & -7 & -3 & 0 \\ 0 & 0 & -9 & -7 & 4 \\ 0 & 0 & 0 & -4 & -7 \end{pmatrix}, B = \begin{pmatrix} -5 \\ -3 \\ -1 \\ 7 \\ 5 \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}.$$

a Use Doolittle decomposition algorithm to find L and U so that $A = LU$.

b Use Forward substitution algorithm to solve $LY = B$ for Y where L is the lower triangle matrix obtained from the decomposition.

c Use Backward substitution algorithm to solve $UX = Y$ for X where U is the upper triangle matrix obtained from the decomposition.

11 Write a general purpose algorithm for Doolittle decomposition method.

a Describe your algorithm objective?

b Describe required input parameter(s)?

c Describe output value(s)?

d Write your algorithm body here.

e Implement your developed algorithm with Python, or any other programming language.

12 Use the code developed from previous exercise to solve $AX = B$ for X where

$$A = \begin{pmatrix} 2 & -1 & & & (0) \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ (0) & & & -1 & 2 \end{pmatrix}_{9\times 9}, B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 10 \end{pmatrix}_{9\times 1}, X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \\ x_9 \end{pmatrix}.$$

a From Doolittle decomposition, what is L and U?

b Solve $LY = B$ for Y.

c Solve $UX = Y$ for X.

13 We are given

$$A = \begin{pmatrix} 2 & 4 & -2 & 4 & -2 \\ 4 & 9 & -2 & 7 & -2 \\ -2 & -2 & 8 & -2 & 4 \\ 4 & 7 & -2 & 18 & -8 \\ -2 & -2 & 4 & -8 & 14 \end{pmatrix}, B = \begin{pmatrix} 2 \\ 8 \\ 8 \\ 1 \\ -4 \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}, Z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix}.$$

a Write a general purpose algorithm for Doolittle LDL^T decomposition.

b Implement the algorithm to determine L and D.

c Use the Forward substitution method to solve $LZ = B$ for Z.

d Solve $DY = Z$ for Y.

e Use the Backward substitution method to solve $L^TX = Y$ for X.

14 Let A, B, X, Y are matrices presented in exercise **13**.

a Write a general purpose algorithm for Cholesky LL^T decomposition.

b Implement the algorithm to determine L.

c Use the Forward substitution method to solve $LY = B$ for Y.

d Use the Backward substitution method to solve $L^TX = Y$ for X.

15 The objective of this exercise is to develop a special Doolittle LU decomposition for tridigonal matrix of the form

$$A = \begin{pmatrix} d_1 & e_1 & 0 & 0 & 0 & \cdots & 0 \\ c_1 & d_2 & e_2 & 0 & 0 & \cdots & 0 \\ 0 & c_2 & d_3 & e_3 & 0 & \cdots & 0 \\ 0 & 0 & c_3 & d_4 & e_4 & \cdots & 0 \\ 0 & 0 & 0 & c_4 & d_5 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & e_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & c_{n-1} & d_n \end{pmatrix}$$

and solve $AX = B$ in more efficient way by reducing the calculation complexity.

a Follow an algorithm presented in the course, develop Doolittle LU decomposition for tridiagonal matrix A called `TridiagonalDecomposition` with input

$$\begin{aligned} c &= (c_1, c_2, \dots, c_{n-1}) \\ d &= (d_1, d_2, \dots, d_{n-1}, d_n) \\ e &= (e_1, e_2, \dots, e_{n-1}) \end{aligned}$$

and output

$$\begin{aligned} c' &= (c'_1, c'_2, \dots, c'_{n-1}) \\ d' &= (d'_1, d'_2, \dots, d'_{n-1}, d'_n) \\ e' &= (e'_1, e'_2, \dots, e'_{n-1}) \end{aligned}$$

b Extend `TridiagonalDecomposition` to `TridiagonalSolve` with input

$$\begin{aligned} c &= (c_1, c_2, \dots, c_{n-1}) \\ d &= (d_1, d_2, \dots, d_{n-1}, d_n) \\ e &= (e_1, e_2, \dots, e_{n-1}) \\ B &= (b_1, b_2, \dots, b_{n-1}, b_n) \end{aligned}$$

and output $X = (x_1, x_2, \dots, x_{n-1}, x_n)$, the solution to $AX = B$.

16 Use the code developed from previous exercise to solve $AX = B$ for X where A is a tridiagonal matrix and

$$\begin{aligned} c &= (-1, -1, -1, -1) \\ d &= (2, 2, 2, 2, 2) \\ e &= (-1, -1, -1, -1) \\ B &= (5, -5, 4, -5, 5). \end{aligned}$$

That is solving for $(x_1, x_2, x_3, x_4, x_5)^T$ of the equation

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 5 \\ -5 \\ 4 \\ -5 \\ 5 \end{pmatrix}.$$

17 The objective of this exercise is to develop a special Doolittle LU decomposition for symmetric pentadigonal matrix of the form

$$A = \begin{pmatrix} d_1 & e_1 & f_1 & 0 & 0 & \cdots & 0 \\ e_1 & d_2 & e_2 & f_2 & 0 & \cdots & 0 \\ f_1 & e_2 & d_3 & e_3 & f_3 & \cdots & 0 \\ 0 & f_2 & e_3 & d_4 & e_4 & \ddots & \vdots \\ 0 & 0 & f_3 & e_4 & d_5 & \ddots & f_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & e_{n-1} \\ 0 & 0 & 0 & \cdots & f_{n-2} & e_{n-1} & d_n \end{pmatrix}$$

and solve $AX = B$ in more efficient way by reducing the calculation complexity.

a Follow an algorithm presented in the course, develop Doolittle LU decomposition for symmetric pentadiagonal matrix A called `PentadiagonalDecomposition` with input

$$\begin{aligned} d &= (d_1, d_2, \dots, d_{n-2}, d_{n-1}, d_n) \\ e &= (e_1, e_2, \dots, e_{n-2}, e_{n-1}) \\ f &= (f_1, f_2, \dots, f_{n-2}) \end{aligned}$$

and output

$$\begin{aligned} d' &= (d'_1, d'_2, \dots, d'_{n-2}, d'_{n-1}, d'_n) \\ e' &= (e'_1, e'_2, \dots, e'_{n-2}, e'_{n-1}) \\ f' &= (f'_1, f'_2, \dots, f'_{n-2}) \end{aligned}$$

b Extend `PentadiagonalDecomposition` to `PentadiagonalSolve` with input

$$\begin{aligned} d &= (d_1, d_2, \dots, d_{n-2}, d_{n-1}, d_n) \\ e &= (e_1, e_2, \dots, e_{n-2}, e_{n-1}) \\ f &= (f_1, f_2, \dots, f_{n-2}) \\ B &= (b_1, b_2, \dots, b_{n-2}, b_{n-1}, b_n) \end{aligned}$$

and output $X = (x_1, x_2, \dots, x_{n-1}, x_n)$, the solution to $AX = B$.

c Use your `PentadiagonalSolve` to solve the system of linear equations

$$\begin{cases} 7x_1 - 4x_2 + x_3 & = 1 \\ -4x_1 + 6x_2 - 4x_3 + x_4 & = 1 \\ x_{i-2} - 4x_{i-1} + 6x_i - 4x_{i+1} + x_{i+2} & = 1 \quad (i = 3, 4, \dots, 8) \\ x_7 - 4x_8 + 6x_9 - 4x_{10} & = 1 \\ x_8 - 4x_9 + 7x_{10} & = 1 \end{cases}$$

for $X = (x_1, x_2, \dots, x_{10})$.

18 Write a general purpose algorithm for Gaussian Scaled Row Pivoting.

a Describe your algorithm objective?

b Describe required input parameter(s)?

c Describe output value(s)?

d Write your algorithm body here.

e Implement your developed algorithm with Python, or any other programming language.

19 Solve $AX = B$ for X where

$$A = \begin{pmatrix} 1 & 10^{100} \\ 1 & 10^{-100} \end{pmatrix}, B = \begin{pmatrix} 10^{100} \\ 1 \end{pmatrix}$$

a using Gaussian scaled row pivoting.

b using Gaussian elimination.

20 Find the first two iterations of the Jacobi method for the following linear systems, using $X^{(0)} = \mathbf{0}$.

a $\begin{cases} 3x_1 - x_2 + x_3 = 1 \\ 3x_1 + 6x_2 + 2x_3 = 0 \\ 3x_1 + 3x_2 + 7x_3 = 4 \end{cases}$

b $\begin{cases} 10x_1 - x_2 = 9 \\ -x_1 + 10x_2 - 2x_3 = 7 \\ -2x_2 + 10x_3 = 6 \end{cases}$

21 Write a general purpose algorithm for Jacobi Iteration technique.

a Describe your algorithm objective?

b Describe required input parameter(s)?

c Describe output value(s)?

d Write your algorithm body here.

e Implement your developed algorithm with Python, or any other programming language.

22 Let

$$A = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 5 \end{pmatrix}, B = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, X^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

a Solve $AX = B$ for X using code of Jacobi method developed from previous exercise.

b How many iterations needed for which the solution is within 10^{-10} accuracy?

c Verify your result with `numpy.linalg.solve`, or `scipy.linalg.solve` from the `numpy` and `scipy` packages respectively.

23 Find the first two iterations of the Gauss-Seidel method for the following linear systems, using $X^{(0)} = \mathbf{0}$.

a $\begin{cases} 4x_1 + x_2 - x_3 = 5 \\ -x_1 + 3x_2 + x_3 = -4 \\ 2x_1 + 2x_2 + 5x_3 = 1 \end{cases}$

b $\begin{cases} -4x_1 + 2x_2 + x_3 = 8 \\ 2x_1 - 4x_2 - x_3 = -8 \\ x_2 + 2x_3 = 0 \end{cases}$

24 Write a general purpose algorithm for Gauss-Seidel Iteration technique.

- **a** Describe your algorithm objective?
- **b** Describe required input parameter(s)?
- **c** Describe output value(s)?
- **d** Write your algorithm body here.
- **e** Implement your developed algorithm with Python, or any other programming language.

25 Let

$$A = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 5 \end{pmatrix}, B = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, X^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

- **a** Solve $AX = B$ for X using the code of Gauss-Seidel method developed from previous exercise.
- **b** How many iterations needed for which the solution is within 10^{-10} accuracy?
- **c** Verify your result with `numpy.linalg.solve`, or `scipy.linalg.solve` from the `numpy` and `scipy` packages respectively.

26 The linear system

$$\begin{cases} x_1 - x_3 = 0.2 \\ -\frac{1}{2}x_1 + x_2 - x_3 = -1.425 \\ x_1 - \frac{1}{2}x_2 + x_3 = 2 \end{cases}$$

has the solution $(0.9, -0.8, 0.7)^T$.

- **a** Is the coefficient matrix

$$A = \begin{pmatrix} 1 & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{4} \\ 1 & -\frac{1}{2} & 1 \end{pmatrix}$$

  strictly diagonally dominant?
- **b** Compute the spectral radius of the Gauss-Seidel matrix T_G.
- **c** Use the Gauss-Seidel iterative method to approximate the solution to the linear system with a tolerance of 10^{-2} and a maximum of 300 iterations.
- **d** What happens in part **c** when the system is changed to

$$\begin{cases} x_1 - 2x_3 = 0.2 \\ -\frac{1}{2}x_1 + x_2 - x_3 = -1.425 \\ x_1 - \frac{1}{2}x_2 + x_3 = 2 \end{cases}$$

27 Repeat Exercise **26** using the Jacobi method.

28 Use $X^{(0)} = \mathbf{0}$, find the first two iterations of the Relaxation method for the following linear systems

$$\begin{cases} 4x_1 + x_2 - x_3 + x_4 = -2 \\ x_1 + 4x_2 - x_3 - x_4 = -1 \\ -x_1 - x_2 + 5x_3 + x_4 = 0 \\ x_1 - x_2 + x_3 + 3x_4 = 1 \end{cases}$$

a with $\omega = 1.1$.

b with $\omega = 1.3$.

29 Solve the system of linear equations

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 1 \\ -1 & 2 & -1 & \ddots & 0 \\ 0 & -1 & 2 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & -1 \\ 1 & 0 & 0 & -1 & 2 \end{pmatrix}_{n\times n} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}_{n\times 1}$$

using Conjugate Gradient method with $n = 20$.