Lecture 10

Numerical Techniques: Solving Sets of Equations - Linear Algebra and Matrix Computing I

Kirchhoff's Rules

IKC-MH.55 *Scientific Computing with Python* at December 29, 2023

Dr. Cem Özdoğan
Engineering Sciences Department
İzmir Kâtip Çelebi University

Contents

Solving Sets of Equations

- Solving sets of linear equations and eigenvalue problems are the most frequently used numerical procedures when real-world situations are modelled.
- Analytical solution may be feasible when the number of unknowns is small.
- However, computers outperforms to solve large systems of linear equations such as with 100 unknowns in a reasonable time.

1 Matrices and Vectors

Reviews concepts of matrices and vectors in preparation.

2 Elimination Methods

Describes classical methods that *change a system of equations to form*s that allow getting the solution by back-substitution. How the errors of the solution can be minimized.

3 The Inverse of a Matrix

4 Iterative Methods

Matrices and Vectors I

- When a system of equations has more than two or three equations, it is difficult to discuss them without using *matrices* and *vectors*.
- A *matrix* is a rectangular array of numbers in which not only the value of the number is important but also its position in the array.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} = [a_{ij}]\,, \quad \begin{array}{l} i = 1, 2, \dots, n, \\ j = 1, 2, \dots, m \end{array}$$

- An $m \times n$ matrix times as $n \times 1$ vector gives an $m \times 1$ product $(m \times \mathbf{nn} \times 1)$.
- The general relation for $\boxed{Ax = b}$ is

$$b_i = \sum_{k=1}^{No.of.cols.} a_{ik} x_k, \qquad i = 1, 2, \dots, \#\ of\ rows$$

where A is a matrix, x and b are vectors (column vectors).

Matrices and Vectors II

- This definition of matrix multiplication permits us to write the set of linear equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n
\end{aligned}
$$

  - If the equations in any two rows is interchanged, the solution does not change.
  - Multiplying the equation in any row by a constant does not change the solution.
  - Adding or subtracting the equation in a row to another row does not change the solution.

Matrices and Vectors III

- Much more simply in matrix notation, as $Ax = b$ where

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- Example,

Matrix notation:

$$\begin{bmatrix} 3 & 2 & 4 \\ 1 & -2 & 0 \\ -1 & 3 & 2 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ -7 \\ 2 \end{bmatrix} \Leftrightarrow$$

Set of equations:

$$\begin{aligned} 3x_1 + 2x_2 + 4x_3 &= 14 \\ x_1 - 2x_2 &= -7 \\ -x_1 + 3x_2 + 2x_3 &= 2 \end{aligned}$$

- **Square matrices** are particularly important when a system of equations is to be solved.

Some Special Matrices and Their Properties I

- **Symmetric matrix.**
  A square matrix is called a **symmetric** matrix when the pairs of elements in similar positions across the diagonal are equal.

$$\begin{bmatrix} 1 & x & y \\ x & 2 & z \\ y & z & 3 \end{bmatrix}$$

- The **transpose** of a matrix is the matrix obtained by writing the rows as columns or by writing the columns as rows.
  - The symbol for the transpose of matrix A is A^T.

$$A = \begin{bmatrix} 3 & -1 & 4 \\ 0 & 2 & 3 \\ 1 & 1 & 2 \end{bmatrix}, \qquad A^T = \begin{bmatrix} 3 & 0 & 1 \\ -1 & 2 & 1 \\ 4 & -3 & 2 \end{bmatrix}$$

- If all the elements above/below the diagonal are zero, a matrix is called ***lower/upper-triangular*** (L/U);

$$L = \begin{bmatrix} x & 0 & 0 \\ x & x & 0 \\ x & x & x \end{bmatrix}, \qquad U = \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{bmatrix}$$

- **We will deal with square matrices.**

Some Special Matrices and Their Properties II

- **Sparse matrix.** In some important applied problems, only a few of the elements are nonzero.
- Such a matrix is termed a **sparse** matrix and procedures that take advantage of this sparseness are of value.
- Division of matrices is not defined, but we will discuss the **inverse** of a matrix.
- The **determinant** of a square matrix is a number.
  - The method of calculating determinants is a lot of work if the matrix is of large size.
  - Methods that triangularize a matrix, as described in next section, are much better ways to get the determinant.
- If a matrix, B, is triangular (either upper or lower), its determinant is just the product of the diagonal elements:

$$det(B) = \Pi B_{ii}, \qquad i = 1, \ldots, n$$

$$det \begin{vmatrix} 4 & 0 & 0 \\ 6 & -2 & 0 \\ 1 & -3 & 5 \end{vmatrix} = -40$$

If we have a square matrix and the coefficients of the determinant are nonzero, there is a unique solution.

Some Special Matrices and Their Properties III

- Determinants can be used to obtain the **characteristic polynomial** and the **eigenvalues** of a matrix, which are the roots of that polynomial.
- If a matrix is triangular, **its eigenvalues are equal to the diagonal elements.**
- This follows from the fact that
  - its determinant is just the product of the diagonal elements and
  - its characteristic polynomial is the product of the terms $(a_{ii} - \lambda)$ with i going from 1 to n, the number of rows of the matrix.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix},$$

$$det(A - \lambda I) = det \begin{vmatrix} 1-\lambda & 2 & 3 \\ 0 & 4-\lambda & 5 \\ 0 & 0 & 6-\lambda \end{vmatrix} = (1-\lambda)(4-\lambda)(6-\lambda)$$

whose roots are clearly 1, 4, and 6.

- It does not matter if the matrix is upper- or lower-triangular.

Elimination Methods I

- Solve a set of linear (variables with first power) equations.
- If we have a system of equations that is of an *upper-triangular* form

$$\begin{aligned} 5x_1 + 3x_2 - 2x_3 &= -3 \\ 6x_2 + x_3 &= -1 \\ 2x_3 &= 10 \end{aligned}$$

  Then, we have the solution as: $x_1 = 2, \; x_2 = -1, \; x_3 = 5$

- If NOT: <u>Change the matrix of coefficients</u> $\Longrightarrow$ <u>upper-triangular</u>. Consider this example of three equations:

$$\begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -3x_1 - x_2 + 4x_3 &= 8 \\ x_1 - x_2 + 3x_3 &= 13 \end{aligned} \qquad \begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -10x_2 + 19x_3 &= 77 \\ -2x_2 + 11x_3 &= 37 \end{aligned} \qquad \begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -10x_2 + 19x_3 &= 77 \\ -72x_3 &= -216 \end{aligned}$$

  Now we have a **triangular system** and the solution is readily obtained;

  1. obviously $x_3 = 3$ from the third equation,
  2. and **back-substitution** into the second equation gives $x_2 = -2$.
  3. We continue with back-substitution by substituting both x_2, and x_3 into the first equation to get $x_1 = 2$.

- Notice to the values in this example. They are getting bigger!

Elimination Methods II

- The essence of any elimination method is to reduce the coefficient matrix to a triangular matrix and then use back-substitution to get the solution.
- We now present the same problem, solved in exactly the same way, in matrix notation;

$$\begin{bmatrix} 4 & -2 & 1 \\ -3 & -1 & 4 \\ 1 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15 \\ 8 \\ 13 \end{bmatrix}$$

- So we work with the matrix of coefficients *augmented* with the right-hand-side vector.
- We perform elementary row transformations to convert A to upper-triangular form:

$$\begin{bmatrix} 4 & -2 & 1 & 15 \\ -3 & -1 & 4 & 8 \\ 1 & -1 & 3 & 13 \end{bmatrix}, \qquad \begin{matrix} 3R_1 + 4R_2 \rightarrow \\ (-1)R_1 + 4R_3 \rightarrow \end{matrix} \qquad \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -10 & 19 & 77 \\ 0 & -2 & 11 & 37 \end{bmatrix},$$

$$2R_2 - 10R_3 \rightarrow \qquad \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -10 & 19 & 77 \\ 0 & 0 & 72 & -216 \end{bmatrix} \Rightarrow \qquad \begin{aligned} 4x_1 - 2x_2 + x_3 &= 15 \\ -10x_2 + 19x_3 &= 77 \\ -72x_3 &= -216 \end{aligned}$$

Elimination Methods III

- The back-substitution step can be performed quite mechanically by solving the equations in reverse order. That is, $x_3 = 3, x_2 = -2, x_1 = 2$. *Same solution with the non-matrix notation*.
- During the triangularization step, if a **zero** is encountered on the diagonal, we cannot use that row to eliminate coefficients below that zero element.
  - However, in that case, we can continue by **interchanging rows** and eventually achieve an upper-triangular matrix of coefficients.
- The real trouble is finding a zero on the diagonal after we have triangularized.
  - If that occurs, the back-substitution fails, for we cannot divide by zero.
  - It also means that the determinant is zero. There is no solution.

Elimination Methods IV

Possible approaches for the solution of the following system of equations with coefficient matrix A.

1 ***Cramer's rule***.

$$\begin{aligned} 6x_1 - 3x_2 + x_3 &= 11 \\ 2x_1 + x_2 - 8x_3 &= -15 \\ x_1 - 7x_2 + x_3 &= 10 \end{aligned} \qquad A = \begin{bmatrix} 6 & -3 & 1 \\ 2 & 1 & -8 \\ 1 & -7 & 1 \end{bmatrix},$$

- Let's denote the matrix by A_j in which the right-hand-side vector are substituted to the j^{th} column of A matrix.
- e.g., the solution for x_1 is expressed as:

$$x_1 = \frac{det(A_1)}{det(A)} = \frac{det \begin{vmatrix} 11 & -3 & 1 \\ -15 & 1 & -8 \\ 10 & -7 & 1 \end{vmatrix}}{det \begin{vmatrix} 6 & -3 & 1 \\ 2 & 1 & -8 \\ 1 & -7 & 1 \end{vmatrix}} = \frac{-285}{-285} = 1$$

Similarly, the solutions x_2 and x_3 can be written.

Elimination Methods V

- This method is very convenient when the number of unknowns is as few as 3-5.
- However, it is not feasible for systems with a large number of unknowns since determinant calculation requires many multiplication operations.
- For example, calculating a 20×20 determinant with Cramer's rule requires $\approx 10^{20}$ multiplications!

2. ***Inverse matrix***. Another solution is to use the A^{-1} matrix, which is the inverse of the A matrix.

- Multiplying both sides of the equation $A\vec{x} = \vec{b}$ by A^{-1} and considering that $A^{-1}A = 1$,

$$
\begin{aligned}
A^{-1}A\vec{x} &= A^{-1}\vec{b} \\
\underbrace{A^{-1}A}_{1}\vec{x} &= A^{-1}\vec{b} \\
\vec{x} &= A^{-1}\vec{b}
\end{aligned}
$$

- However, this approach is also not reasonable since calculating matrix inverses requires a large number of operations.

Gaussian Elimination I

- **Therefore, adequate methods should be used in linear equation system solutions such as without calculating determinant or inverse matrix.**
- Two of the most useful methods are **Gaussian elimination** and **L-U decomposition**.
- *Elimination Methods.* While it may be satisfactory for hand computations with small systems, it is inadequate for a large system (numbers may getting bigger!).
- The method that is called *Gaussian elimination* avoids this by subtracting a_{i1}/a_{11} times the first equation from the i^{th} equation to make the transformed numbers in the first column equal to zero.
- We do similarly for the rest of the columns.
- Observe that zeros may be created in the diagonal positions even if they are not present in the original matrix of coefficients.

Gaussian Elimination II

- A useful strategy to avoid (if possible) such zero divisors in the diagonal positions is to rearrange the equations so as to put the coefficient of largest magnitude on the diagonal at each step.
- This is called **pivoting**.
- *Generalization*. Let the system of equations with N unknowns be given as:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n
\end{aligned}
$$

  - **First Stage**. Let $a_{11} \neq 0$ (If not, pivoting), then multiply the 1^{st} equation by (a_2/a_{11})) and subtract it from the 2^{nd} equation.
  - **Next,** again multiply the 1^{st} equation by (a_{31}/a_{11}) and subtract from the 3^{rd} equation.
  - **Repeat** this procedure up to the n^{th} equation.

Gaussian Elimination III

- As a result, the variable x_1 is eliminated in the other equations and the new system of equations becomes:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
\left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \ldots + \left(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}\right)x_n &= b_2 - \frac{a_{21}}{a_{11}}b_1 \\
\left(a_{32} - \frac{a_{31}}{a_{11}}a_{13}\right)x_2 + \ldots + \left(a_{3n} - \frac{a_{31}}{a_{11}}a_{1n}\right)x_n &= b_3 - \frac{a_{31}}{a_{11}}b_1 \\
&\vdots \\
\left(a_{n2} - \frac{a_{n1}}{a_{11}}a_{1n}\right)x_2 + \ldots + \left(a_{nn} - \frac{a_{nn}}{a_{11}}a_{1n}\right)x_n &= b_n - \frac{a_{n1}}{a_{11}}b_1
\end{aligned}
$$

- Notice that in this new system of equations at **first stage**, the coefficient of the j^{th} term in the i^{th} row and the constant b_i are as follows:

$$
\begin{aligned}
a_{ij}^{(1)} &= a_{ij} - \frac{a_{i1}}{a_{11}}a_{1j} \qquad (i, j = 2, \ldots, n) \\
b_i^{(1)} &= b_i - \frac{a_{i1}}{a_{11}}b_i \qquad (i = 2, \ldots, n)
\end{aligned}
$$

Gaussian Elimination IV

- **Next,** let $a_{22}^{(1)} \neq 0$ (If not, pivoting) in this new system of equations, then multiply the 2^{nd} equation by $(a_{32}^{(1)}/a_{22}^{(1)})$) and subtract it from the 3^{rd} equation.
- **Repeat** this procedure up to equation n as $n-1$ times to obtain the upper-triangular form:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)} \\
a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)} \\
&\vdots \\
a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}
\end{aligned}
$$

- As seen, the number of unknowns decreases by one in the 2^{nd} and other equations as 1^{st} stage, decreases once again in the 3^{rd} and other equations as 2^{nd} stage and so on. In $(N-1)^{th}$ stage, a single unknown is obtained.

Gaussian Elimination V

- Then, the j^{th} coefficient of the i^{th} equation as k^{th} stage is as follows:

$$\boxed{a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)} a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}} \qquad (i, j = 1, \ldots, n) \qquad (1)$$

$$\boxed{b_{i}^{(k)} = b_{i}^{(k-1)} - \frac{a_{ik}^{(k-1)} b_{k}^{(k-1)}}{a_{kk}^{(k-1)}}} \qquad (i = 1, \ldots, n) \qquad (2)$$

- After reaching to the upper-triangular form, the solution is almost readily obtained.
- From the last equation in the upper-triangular form:

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)}$$

- All other unknowns are obtained consequently by backward-substitution. The general expression would be:

$$\boxed{x_k = \frac{1}{a_{kk}^{(k-1)}} \left[b_k^{(k-1)} - \sum_{j=k+1}^{n} a_{kj}^{(k-1)} x_j \right]} \quad (k = n-1, \ldots, 1) \quad (3)$$

Gaussian Elimination VI

- Repeat the example of the previous section,

$$\begin{bmatrix} 4 & -2 & 1 & 15 \\ -3 & -1 & 4 & 8 \\ 1 & -1 & 3 & 13 \end{bmatrix}, \qquad \begin{matrix} \\ R_2 - (-3/4)R_1 \rightarrow \\ R_3 - (1/4)R_1 \rightarrow \end{matrix} \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -2.5 & 4.75 & 19.25 \\ 0 & -0.5 & 2.75 & 9.25 \end{bmatrix},$$

$$R_3 - (-0.5/-2.5)R_2 \rightarrow \qquad \begin{bmatrix} 4 & -2 & 1 & 15 \\ 0 & -2.5 & 4.75 & 19.25 \\ 0 & 0 & 1.8 & 5.40 \end{bmatrix}$$

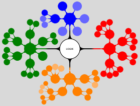
- The method we have just illustrated is called *Gaussian elimination*.
- In this example, **no pivoting was required** to make the largest coefficients be on the diagonal.
- Back-substitution, gives us $x_3 = 3,\ x_2 = -2,\ x_1 = 2$

Gaussian Elimination VII

Example py-file: Show steps in Gaussian elimination and back substitution without pivoting. myGEshow.py

```
Tolerance value in pivoting is 1.110223e-15.
Begin forward elimination with Augmented system:
[[ 4. -2.  1. 15.]
 [-3. -1.  4.  8.]
 [ 1. -1.  3. 13.]]
Stage 0
After elimination in column 0 with pivot = 4.000000
[[ 4.   -2.    1.   15.  ]
 [ 0.   -2.5   4.75 19.25]
 [ 0.   -0.5   2.75  9.25]]
Stage 1
After elimination in column 1 with pivot = -2.500000
[[ 4.   -2.    1.   15.  ]
 [ 0.   -2.5   4.75 19.25]
 [ 0.    0.    1.8   5.4 ]]
Stage 2
After elimination in column 2 with pivot = 1.800000
[[ 4.   -2.    1.   15.  ]
 [ 0.   -2.5   4.75 19.25]
 [ 0.    0.    1.8   5.4 ]]
x2=3.000000
x1=(19.250000-14.250000)/-2.500000)=-2.000000
x0=(15.000000-7.000000)/4.000000)=2.000000
Solution Vector is  [ 2. -2.  3.]
```

Figure: Steps in Gaussian elimination and back substitution without pivoting.



Gaussian Elimination VIII

- if we had stored the ratio of coefficients in place of zero (we show these in parentheses), our final form would have been

$$\begin{bmatrix} 4 & -2 & 1 & 15 \\ (-0.75) & -2.5 & 4.75 & 19.25 \\ (0.25) & (0.20) & 1.8 & 5.40 \end{bmatrix}$$

- The original matrix can be written as the product:

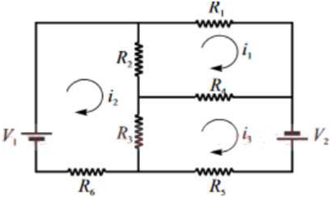
$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -0.75 & 1 & 0 \\ 0.25 & 0.20 & 1 \end{bmatrix}}_{L} * \underbrace{\begin{bmatrix} 4 & -2 & 1 \\ 0 & -2.5 & 4.75 \\ 0 & 0 & 1.8 \end{bmatrix}}_{U}$$

- This procedure is called a ***LU-decomposition of A***. ($A = L * U$)
- We have $det(A) = det(U) = (4) * (-2.5) * (1.8) = -18$
- When there are m row interchanges

$$\boxed{det(A) = (-1)^m * u_{11} * \ldots * u_{nn}}$$

Kirchhoff's Rules I

Find currents at each loop by using Kirchhoff's Junction & Loop Rules:



3 unkowns, 3 equations

$$
\begin{aligned}
(R_1 + R_2 + R_4)i_1 - R_2 i_2 - R_4 i_3 &= 0 \\
-R_2 i_1 + (R_2 + R_3 + R_6)i_2 - R_3 i_3 &= V_1 \\
-R_4 i_1 - R_3 i_2 + (R_3 + R_4 + R_5)i_3 &= V_2
\end{aligned}
$$

With the values of $R_1 = R_2 = 1\ \Omega$, $R_3 = R_4 = R_5 = R_6 = 2\ \Omega$ and $V_1 = 1\ V$, $V_2 = 5\ V$. System of linear equations becomes as follows:

$$
\begin{aligned}
4i_1 - i_2 - 2i_3 &= 0 \\
-i_1 + 5i_2 - 2i_3 &= 1 \\
-2i_1 - 2i_2 + 6i_3 &= 5
\end{aligned}
$$

Example py-file: Kirchhoff's Rules in Gaussian elimination & back substitution. No pivoting. myKirchhoff.py

Kirchhoff's Rules II

```
Tolerance value in pivoting is 1.110223e-15.
Gaussian Elimination with Augmented system:
[[ 4. -1. -2.  0.]
 [-1.  5. -2.  1.]
 [-2. -2.  6.  5.]]
--- Stage 0
After elimination in column 0 with pivot = 4.000000
[[ 4.   -1.   -2.    0.  ]
 [ 0.    4.75 -2.5   1.  ]
 [ 0.   -2.5   5.    5.  ]]
--- Stage 1
After elimination in column 1 with pivot = 4.750000
[[ 4.         -1.         -2.          0.        ]
 [ 0.          4.75       -2.5         1.        ]
 [ 0.          0.          3.68421053  5.52631579]]
--- Stage 2
After elimination in column 2 with pivot = 3.684211
[[ 4.         -1.         -2.          0.        ]
 [ 0.          4.75       -2.5         1.        ]
 [ 0.          0.          3.68421053  5.52631579]]
Backward substitution
x2=1.500000
x1=(1.000000--3.750000)/4.750000)=1.000000
x0=(0.000000--4.000000)/4.000000)=1.000000
MyGEshow - Solution Vector :  [[1.  1.  1.5]]
NumPy Solve - Solution Vector : [1.  1.  1.5]
```

Figure: Kirchhoff's Rules in Gaussian elimination & back substitution. No pivoting.

Gaussian Elimination IX

Example. Solve the following system of equations using Gaussian elimination.

$$\begin{array}{rrrrl} & 2x_2 & & +x_4 & = 0 \\ 2x_1 & +2x_2 & +3x_3 & +2x_4 & = -2 \\ 4x_1 & -3x_2 & & x_4 & = -7 \\ 6x_1 & +x_2 & -6x_3 & -5x_4 & = 6 \end{array}$$

In addition, obtain the determinant of the coefficient matrix and the *LU* decomposition of this matrix.

1 The augmented coefficient matrix is

$$\begin{bmatrix} 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{bmatrix}$$

2 We cannot permit a **zero** in the a_{11} position because that element is the pivot in reducing the first column.

3 We could interchange the first row with any of the other rows to avoid a zero divisor, but interchanging the first and fourth rows is our best choice. This gives

$$\begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & 1.6667 & 5 & 3.6667 & -4 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

Gaussian Elimination X

4 We again interchange before reducing the second column, not because we have a zero divisor, but because we want to preserve accuracy. Interchanging the second and third rows puts the element of largest magnitude on the diagonal.

$$\begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 1.6667 & 5 & 3.6667 & -4 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

5 Now we reduce in the second column

$$\begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 0 & 6.8182 & 5.6364 & -9.0001 \\ 0 & 0 & 2.1818 & 3.3636 & -5.9999 \end{bmatrix}$$

6 No interchange is indicated in the third column. Reducing, we get

$$\begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & -11 \\ 0 & 0 & 6.8182 & 5.6364 & -9.0001 \\ 0 & 0 & 0 & 1.5600 & -3.1199 \end{bmatrix}$$

Gaussian Elimination XI

7 Back-substitution gives

$$x_1 = -0.50000, x_2 = 1.0000, x_3 = 0.33325, x_4 = -1.9999.$$

- The **correct (exact)** answers are $x_1 = -1/2, x_2 = 1, x_3 = 1/3, x_4 = -2$.
- In this calculation we have carried five significant figures and rounded each calculation.
- Even so, we do not have five-digit accuracy in the answers. The discrepancy is due to **round off**.
- **Example py-file:** Show steps in Gauss elimination and back substitution with pivoting. myGEPivShow.py

Gaussian Elimination XII

```
Tolerance value in pivoting is 1.110223e-15.
Gaussian Elimination with Augmented system:
[[ 0.  2.  0.  1.  0.]
 [ 2.  2.  3.  2. -2.]
 [ 4. -3.  0.  1. -7.]
 [ 6.  1. -6. -5.  6.]]
--- Stage 0
Swap rows 0 and 3;  new pivot = 6.000000
After elimination in column 0 with pivot = 6.000000
[[  6.          1.          -6.          -5.          6.        ]
 [  0.          1.66666667   5.          3.66666667  -4.        ]
 [  0.         -3.66666667   4.          4.33333333 -11.        ]
 [  0.          2.           0.          1.          0.        ]]
--- Stage 1
Swap rows 1 and 2;  new pivot = -3.666667
After elimination in column 1 with pivot = -3.666667
[[  6.          1.          -6.          -5.          6.        ]
 [  0.         -3.66666667   4.          4.33333333 -11.        ]
 [  0.          0.          6.81818182   5.63636364  -9.        ]
 [  0.          0.          2.18181818   3.36363636  -6.        ]]
--- Stage 2
After elimination in column 2 with pivot = 6.818182
[[  6.          1.          -6.          -5.          6.        ]
 [  0.         -3.66666667   4.          4.33333333 -11.        ]
 [  0.          0.          6.81818182   5.63636364  -9.        ]
 [  0.          0.          0.          1.56        -3.12      ]]
--- Stage 3
After elimination in column 3 with pivot = 1.560000
[[  6.          1.          -6.          -5.          6.        ]
 [  0.         -3.66666667   4.          4.33333333 -11.        ]
 [  0.          0.          6.81818182   5.63636364  -9.        ]
 [  0.          0.          0.          1.56        -3.12      ]]
Backward substitution
x3=-2.000000
x2=(-9.000000--11.272727)/6.818182)=0.333333
x1=(-11.000000--7.333333)/-3.666667)=1.000000
x0=(6.000000-9.000000)/6.000000)=-0.500000
MyGEPivShow - Solution Vector :  [[-0.5         1.          0.33333333 -2.        ]]
NumPy Solve - Solution Vector : [-0.5         1.          0.33333333 -2.        ]
```

Figure: Steps in Gaussian elimination and back substitution with pivoting.