Linear Matrix Equations – Part 1

Mark A. Austin

University of Maryland

austin@umd.edu

ENCE 201, Fall Semester 2023

October 6, 2023

Overview

Part 4

Linear Matrix Equations

Linear Matrix Equations

Matrix Form. The matrix counterpart of 1 is $[A] \cdot [X] = [B]$, where

$$[A] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & & \vdots \\ a_{m1} & \cdots & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (2)$$

Points to note:

- Matrices A and X have dimensions $(m \times n)$ and $(n \times 1)$, respectively.
- Column vector B has dimensions $(n \times 1)$.

Summary of Results

Summary of Results

Classification of Solutions

The results are:

- A unique solution $\{X\} = \left[A^{-1}\right] \cdot \{B\}$ exists when $\left[A^{-1}\right]$ exists (i.e., $\det[A] \neq 0$).
- The equations are inconsistent when $[A]$ is singular and $\operatorname{rank}[A|B] \neq \operatorname{rank}[A]$.
- If $\operatorname{rank}[A|B]$ equals $\operatorname{rank}[A]$, then there are an infinite number of solutions.

Working Example

Problem Statement/Theoretical Considerations

Working Example: Consider the matrix equations:

$$\begin{bmatrix} 3 & -6 & 7 \\ 9 & 0 & -5 \\ 5 & -8 & 6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -4 \end{bmatrix} \tag{43}$$

Part I: Theoretical Considerations:

- A unique solution $\{X\} = \left[A^{-1}\right] \cdot \{B\}$ exists when $\left[A^{-1}\right]$ exists (i.e., det $[A] \neq 0$). Expanding det(A) about the first row gives:

$$\begin{aligned} det(A) &= 3\text{det}\begin{bmatrix} 0 & -5 \\ -8 & 6 \end{bmatrix} + 6\text{det}\begin{bmatrix} 9 & -5 \\ 5 & 6 \end{bmatrix} + 7\text{det}\begin{bmatrix} 9 & 0 \\ 5 & -8 \end{bmatrix}, \\ &= 3(0-40) + 6(54+25) + 7(-72-0) = -150. \end{aligned} \tag{44}$$

Working with NumPy

Part II: Program Source Code:

```
# ==================================================================
# TestMatrixEquations01.py: Compute solution to matrix equations.
#
# Written by: Mark Austin                          November 2022
# ==================================================================

import numpy as np
from numpy.linalg import matrix_rank

# Function to print two-dimensional matrices ...

def PrintMatrix(name, a):
    print("Matrix: {:s} ".format(name) );
    for row in a:
        for col in row:
            print("{:8.3f}".format(col), end=" ")
        print("")

# main method ...

def main():
    print("--- Enter TestMatrixEquations01.main()       ... ");
    print("--- ======================================== ... ");

    print("--- Part 1: Create test matrices ... ");
```

Working with NumPy

Part II: Program Source Code: (Continued) ...

```
    A = np.array( [ [ 3, -6,  7],
                    [ 9,  0, -5],
                    [ 5, -8,  6] ]);
    PrintMatrix("A", A);

    B = np.array([ [3], [3], [-4] ]);
    PrintMatrix("B", B);

    print("--- Part 2: Check properties of matrix A ... ");

    rank = matrix_rank(A)
    det  = np.linalg.det(A)

    print("--- Matrix A: rank = {:f}, det = {:f}  ...".format(rank, det) );

    print("--- Part 3: Solve A.x = B ... ");

    x = np.linalg.solve(A, B)
    PrintMatrix("x", x);

    print("--- ======================================== ... ");
    print("--- Leave TestMatrixEquations01.main()       ... ");

# call the main method ...

main()
```

Working with NumPy

Part III: Program Output:

```
# Part 1: Create test matrices ...

Matrix: A
   3.000   -6.000    7.000
   9.000    0.000   -5.000
   5.000   -8.000    6.000

Matrix: B
   3.000
   3.000
  -4.000

# Part 2: Check properties of matrix A ...

Matrix A: rank = 3.000000, det = -150.000000  ...
```

```
# Part 3: Solve A.x = B ...

Matrix: x
   2.000
   4.000
   3.000
```