Homework 2 - Solutions

Introduction to Linear Algebra

Material from Lectures 2 and 3

Due Thursday, January 19, 2023

2.1 (✠1.03) Let A, B, C, D be $n \times n$ matrices that are invertible. Find the inverses of the following block matrices.

(a) $\begin{bmatrix} I & 0 \\ 0 & D \end{bmatrix}$ (b) $\begin{bmatrix} I & B \\ 0 & D \end{bmatrix}$ (c) $\begin{bmatrix} A & 0 \\ I & D \end{bmatrix}$

The inverse of each matrix M can be found by trial and error, or can be found by using four matrix equations from $I = MM^{-1}$, where $M^{-1} = \left[\begin{smallmatrix} X & Y \\ Z & W \end{smallmatrix}\right]$ is some unknown block matrix.

(a) For the first matrix:

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} X & Y \\ Z & W \end{bmatrix} = \begin{bmatrix} IX + 0Z & IY + 0W \\ 0X + DZ & 0Y + DW \end{bmatrix} = \begin{bmatrix} X & Y \\ DZ & DW \end{bmatrix},$$

which, comparing the four entries on the far left and far right, means that

$$\begin{aligned} X = I, &\quad Y = 0, \\ DZ = 0, &\quad DW = I. \end{aligned}$$

Hence $X = I$ and $Y = 0$. Multiplying $DZ = 0$ by D^{-1} on both sides we get $D^{-1}DZ = D^{-1}0$, which simplifies to $Z = 0$. The last equation similarly gives us $W = D^{-1}$. Hence the desired inverse is

$$\begin{bmatrix} I & 0 \\ 0 & D \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ 0 & D^{-1} \end{bmatrix}.$$

(b) For the second matrix:

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & B \\ 0 & D \end{bmatrix} \begin{bmatrix} X & Y \\ Z & W \end{bmatrix} = \begin{bmatrix} IX + BZ & IY + BW \\ 0X + DZ & 0Y + DW \end{bmatrix} = \begin{bmatrix} X + BZ & Y + BW \\ DZ & DW \end{bmatrix},$$

which, comparing the four entries on the far left and far right, means that

$$\begin{aligned} X + BZ = I, &\quad Y + BW = 0, \\ DZ = 0, &\quad DW = I. \end{aligned}$$

By the equation $DZ = 0$, we get that $Z = 0$. By the equation $DW = I$, we get that $W = D^{-1}$. Knowing that $Z = 0$, the exuation $X + BZ = I$ simplifies to $X = I$. The final eqaution is simply $Y = -BD^{-1}$. The desired inverse is then

$$\begin{bmatrix} I & B \\ 0 & D \end{bmatrix}^{-1} = \begin{bmatrix} I & -BD^{-1} \\ 0 & D^{-1} \end{bmatrix}.$$

(c) For the third matrix:

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ I & D \end{bmatrix} \begin{bmatrix} X & Y \\ Z & W \end{bmatrix} = \begin{bmatrix} AX + 0Z & AY + 0W \\ IX + DZ & IY + DW \end{bmatrix} = \begin{bmatrix} AX & AY \\ X + DZ & Y + DW \end{bmatrix}$$

which, comparing the four entries on the far left and far right, means that

$$\begin{aligned} AX &= I, & AY &= 0, \\ X + DZ &= 0, & Y + DW &= I. \end{aligned}$$

Hence $X = A^{-1}$, $Y = 0$, $W = D^{-1}$, and $Z = -D^{-1}A^{-1}$, so the inverse is

$$\begin{bmatrix} A & 0 \\ 1 & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -D^{-1}A^{-1} & D^{-1} \end{bmatrix}.$$

2.5 *Removed*

2.6 (✠1.05) For each part of this question, construct a Python function with the given name.

(a) Make a function `ones_counter(matrix)` that takes in a matrix, in the form of a `numpy` array, and returns the number of entries that are `1`.

(b) Make a function `thick_diagonal(rownum, colnum)` that takes in two positive integers and returns a matrix, in the form of a `numpy` array, having `rownum` rows and `colnum` columns, and zero everywhere except on the diagonal and just above and just below it. For example, `thick_diagonal(5, 10)` should return the following matrix:

```
array([[1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
       [1, 1, 1, 0, 0, 0, 0, 0, 0, 0],
       [0, 1, 1, 1, 0, 0, 0, 0, 0, 0],
       [0, 0, 1, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 1, 0, 0, 0, 0]])
```

The code given is not the only way to do the problem. Questions are checked automatically on test cases.

(a) The suggested approach is to take advantage of the `np.count_nonzero` function, which counts the number of nonzero entries in a matrix. By shifting all the elements in the input matrix, precisely the elements `1` become `0`.

```
def ones_counter(matrix):
    shape = matrix.shape
    difference = np.ones(shape) - matrix
    return shape[0]*shape[1] - np.count_nonzero(difference)
```

(b) The suggested approach is to take advantage of the `np.eye` function, which creates a matrix with ones on the diagonal. Along with the `np.hstack` and `vstack` functions, which stack input matrices horizontally and vertically, respectively, the desired matrix can be constructed as the sum of three other matrices.

```
def thick_diagonal(rownum, colnum):
    mat_center = np.eye(rownum,colnum)
    mat_below = np.vstack((np.zeros((1,colnum)), np.eye(rownum-1,colnum)))
    mat_above = np.hstack((np.zeros((rownum,1)), np.eye(rownum,colnum-1)))
    return mat_below + mat_center + mat_above
```

3.3 (✠1.07) Let $a, b, c \in \mathbf{R}$ be nonzero numbers, and consider the matrix $A = \begin{bmatrix} a & b & c \\ a & 2b & 3c \\ a & 3b & 6c \end{bmatrix}$.

(a) Give the elementary matrices which, when they are multiplied on the left of A, leave A with zeros below the diagonal (not above).

(b) Let E be the product of the elementary matrices you computed in the first part of this question, and suppose that you began with an equation $A\mathbf{x} = \mathbf{b}$. If $E\mathbf{b} = \begin{bmatrix} 47 \\ 4 \\ 7 \end{bmatrix}$, what is $\mathbf{b}$? What are a, b, c?

(a) To get rid of the a in the $(2,1)$-position, A should be multiplied by $\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. To get rid of the a in the $(3,1)$-position, A should be multiplied by $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$. Then we get

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}}_{E_2} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_1} \begin{bmatrix} a & b & c \\ a & 2b & 3c \\ a & 3b & 6c \end{bmatrix} = \begin{bmatrix} a & b & c \\ 0 & b & 2c \\ 0 & 2b & 5c \end{bmatrix}.$$

Now, to get rid of the $2b$ in the $(3,2)$-position, this matrix should be multiplied by $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$, which gives

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}}_{E_3} \begin{bmatrix} a & b & c \\ 0 & b & 2c \\ 0 & 2b & 5c \end{bmatrix} = \begin{bmatrix} a & b & c \\ 0 & b & 2c \\ 0 & 0 & c \end{bmatrix}.$$

The necessary elementary matrices are hence E_1, E_2, E_3.

(b) It is possible to determine $\mathbf{b}$ explicitly. First note that

$$E = E_3 E_2 E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -2 & 1 \end{bmatrix},$$

so

$$\begin{bmatrix} 47 \\ 4 \\ 7 \end{bmatrix} = E\mathbf{b} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - b_1 \\ b_1 + b_3 - 2b_2 \end{bmatrix}.$$

From the first line we get $b_1 = 47$. From the second line we get $4 = b_2 - 47$, or $b_2 = 51$. From the third line, we get $7 = 47 + b_3 - 2 \cdot 51$, or $b_3 = 62$.

It is not possible to determine a, b, c explicitly. Stating this qualifies for full points.

It is possible to determine a, b, c implicitly, which we could do by comparing $E\mathbf{b}$ with the reduced A, by the equation

$$\begin{bmatrix} a & b & c \\ 0 & b & 2c \\ 0 & 0 & c \end{bmatrix} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\mathbf{x}} = \begin{bmatrix} 47 \\ 4 \\ 7 \end{bmatrix}.$$

From the third line, we get $c = \frac{7}{x_3}$. From the second line, we get $b = \frac{-10}{x_2}$. From the first line, we get $a = \frac{50}{x_1}$. For these to be true, neither of x_1, x_2, x_3 may be zero.