Solution Set 1, 18.06 Fall '11

1. (a) Do Problem 17 from 2.1. Treat the vectors as *column* vectors.

   Solution. $P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ and $Q = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$. $\square$

   (b) Calculate PQ. Calculate QP. Think about the significance of the answers (no explanation necessary).

   Solution. Both of these give the identity matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. (The idea to get from this problem is that if you do an operation Q that "undoes" the operation P, then P also "undoes" the operation Q (i.e. if $PQ = I$ then $QP = I$). In 2.5, you learned that P and Q are called *inverses*. You'see see this on top of page 83. Warning: it is **not** true that $PQ = QP$ for any matrices. We're only talking about the case $PQ = I$.) $\square$

2. Do Problem 12 from 2.2.

   Solution. (Even though the problem asks for equations, I'm going to use matrices. The reason I'm doing this Convince tyourself that matrices *are just a shorthand for the equations* when we use them to solve equations - this is the point of section 2.3. If you don't get this, think about it first. You'll obviously get full credit if you wrote the actual equations and not matrices) The matrix equations we get are (including the original one),

$$\begin{pmatrix} 2 & 3 & 1 \\ 4 & 7 & 5 \\ 0 & -2 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 20 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & -2 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 8 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 8 \end{pmatrix}.$$

   The last one corresponds to the system of equations

$$\begin{aligned} 2x + 3y + z &= 8 \\ y + 3z &= 4 \\ 8z &= 8. \end{aligned}$$

   The pivots here are the $2x, y$, and $8z$ on the diagonal. Back substitution gives $z = 1$ from the last line, then the second line is $y + 3 = 4$, so $y = 1$. Finally, the first line is $2x + 3 + 1 = 8$, so $x = 2$. $\square$

3. Do Problem 16 from 2.3.

Solution. (a) First, assign variables: let x be X's age, and y be Y's age. The information we get is $x = 2y$ (equivalently, $x - 2y = 0$) and $x + y = 33$. This corresponds to the matrix equation $\begin{pmatrix} 1 & -2 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 33 \end{pmatrix}$. Solving gives $x = 22$ and $y = 11$.

(b) First, we assign variables: here the "right" variables are m and c, not x and y, because m and c are what we are trying to *solve* for. The equations are given by the line condition they satisfy, so they are $2m + c = 5$ and $3m + c = 7$. This corresponds to the matrix equation $\begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}\begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$. Solving gives $m = 2$ and $c = 1$.

□

4. (a) Recall that in Gauss-Jordan we took matrices $M = [A\ I]$, where I is the identity, and performed transformations to get $N = E_1 E_2 \cdots M = [I\ A^{-1}]$. Suppose you applied the same eliminations to the matrix $M = [A\ B]$ where A is the same as before but B is a more general matrix than the identity I. We should still get a matrix $[I\ X]$. What is X in terms of A and B?

   Solution. Multiplication of matrices can be done block by block. Even though we changed B to something else, the fact remains that we applied the same transformtions $E = E_1 E_2 \cdots$ as before. The left block corresponding to A in M was transformed to I, so the multiplication must be by A^{-1} (equivalently, $E = A^{-1}$!). This is why the right block corresponding to I in M was trasnformed to A^{-1}. Thus, whatever B was, it must become $A^{-1}B$, which is the answer. Sanity check yourself that when $B = I$ we do indeed get A^{-1} as expected. □

   (b) Write some code that, given $M = [A\ B]$, produces the matrix $[I\ X]$. *If you did the previous part correctly, you do NOT have to (or want to) do this with actual row operations, as long as you get the same result as if you did. The solution should be only a couple of lines.* (hint for MATLAB users: the backslash is helpful).

   Solution. [See MATLAB code] □

   (c) Show that the code works (by attaching the output) for two matrices of block form $M = [A\ B]$ with different dimensions.

   Solution. [See MATLAB code] □

5. Do problem 10 from section 2.5.

Solution. Note that A takes $\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$ to $\begin{pmatrix} 2w \\ 3z \\ 4y \\ 5x \end{pmatrix}$, so A^{-1} must be the matrix that takes $\begin{pmatrix} 2w \\ 3z \\ 4y \\ 5x \end{pmatrix}$ to $\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$. We can do a substitution $w' = 2w, z' = 3z$, etc. to see that A^{-1} takes $\begin{pmatrix} w' \\ z' \\ y' \\ x' \end{pmatrix}$ to $\begin{pmatrix} x'/5 \\ y'/4 \\ z'/3 \\ w'/2 \end{pmatrix}$, so it must be the matrix $\begin{pmatrix} 0 & 0 & 0 & 1/5 \\ 0 & 0 & 1/4 & 0 \\ 0 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \end{pmatrix}$.

For B, a useful framework to keep in mind is block multiplication. Think of B as the block matrix $\begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}$. Each of the 2 by 2 blocks B_i has an inverse; note that multiplying B by $\begin{pmatrix} B_1^{-1} & 0 \\ 0 & B_2^{-1} \end{pmatrix}$ gives exactly the identity. You've learned a formula for inverses of these small matrices, so you can get $B_1^{-1} = 1/(3*3-2*4)\begin{pmatrix} 3 & -2 \\ -4 & 3 \end{pmatrix} = \begin{pmatrix} 3 & -2 \\ -4 & 3 \end{pmatrix}$ and $B_2^{-1} = \begin{pmatrix} 6 & -5 \\ -7 & 6 \end{pmatrix}$. Therefore, the answer is $\begin{pmatrix} 3 & -2 & 0 & 0 \\ -4 & 3 & 0 & 0 \\ 0 & 0 & 6 & -5 \\ 0 & 0 & -7 & 6 \end{pmatrix}$.

For either of these problems (and tests, etc.) don't forget that inverses are *unique*; so if you have a guess, you can just assert that your guess is the inverse A^{-1} by just computing $AA^{-1} = 1$. □

6. Do problem 18 from section 2.5.

   Solution. If B is the inverse of A^2, then $A^2(B) = I$. But note we can rewrite this as $A(AB) = I$, so AB is the inverse of A also. □

7. For every $n = 1, 2, 3, \ldots$, there is a very important matrix (with complex entries) known as the fourier matrix F_n. We will modify it slightly and work with $G_n = F_n/\sqrt{n}$. (In MATLAB it is obtained by `G_n=fft(eye(n))/sqrt(n)`. In other languages you can use some version of FourierMatrix or write loops to make the (i,j)-th entry

$$f(i,j) = \exp(-2ij\pi\sqrt{-1}/n)/\sqrt{n}$$

   for $i, j = 1, ..., n$)

   (a) Find $(G_n)^4$, via a computer, for any 3 values of n. Make a conjecture for what $(G_n)^4$ is and do not try to prove it(the proof is outside the scope of the class, but there should be no doubt to the conjecture).

   Solution. [See MATLAB code] You should see that the answer is probably I. □

(b) Assuming your conjecture is true, what is $(G_n)^{-1}$ in terms of G_n? Your expression should be a purely algebraic expression with no negative signs or words.

Solution. Note that $G_n^3(G_n) = G_n^4 = I$, so The inverse is G_n^3. □

(c) Let M be a random matrix of dimension n (in MATLAB this can be done by `M=rand(n)`). What is the relationship between $G_n M$ and $(G_n)^9 M$? The actual use of the computer is optional if you can justify your reasoning.

Solution. [See MATLAB code] You should notice that $G_n^8(G_n M) = G_n^9 M$, but $G_n^8 = (G_n^4)^2 = I^2 = I$, so these two matrices should be the same. □

8. Do Problem 2.5 number 40... with a twist. This problem can be done: 1) by hand, 2) by guessing with a numerical computation, or 3) by a symbolic computation. Do the problem in 2 of these 3 ways. Some hints/help:

- If you are trying to be a good guesser, use $a = 2$, $b = 3$, and $c = 10$ and see how good a pattern detector you are. This can be done by hand (good practice!) or MATLAB, where you'd do something like
```
A=eye(4)-diag([2 3 10],1)
inv(A).
```
- If you want to do symbolic computation in MATLAB, if the symbolic toolbox is available you can do
```
syms a b c
A=eye(4)-diag([a b c],1).
```
- If you want to do symbolic computation in Mathematica, you can do
```
A={{1,-a,0,0},{0,1,-b,0},{0,0,1,-c},{0,0,0,1}}
MatrixForm[Inverse[A]].
```

Solution. (a) (by hand) The matrix is $\begin{pmatrix} 1 & -a & 0 & 0 \\ 0 & 1 & -b & 0 \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{pmatrix}$. It takes the vector $\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$ to the vector $\begin{pmatrix} x-ay \\ y-bz \\ z-cw \\ w \end{pmatrix}$. Let's define this to be the vector $\begin{pmatrix} r \\ s \\ t \\ u \end{pmatrix}$, and we can back-substitute to find the values of x, y, z, w in terms of r, s, t, u.

We start with $w = u$. Then $z = t+cw = t+cu$, $y = s+bz = s+bt+bcu$, and $x = r+ay = r+as+abt+abcu$. Therefore, the inverse matrix is $\begin{pmatrix} 1 & a & ab & abc \\ 0 & 1 & b & bc \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

Test this by multiplication.

(b) (by guessing) [See MATLAB code]

(c) (by symbolic calculation) [See MATLAB code]

□

MATLAB code

```
%%%%%%%%%%%%%%%%%%%
%Problem 4 (b),(c)%
%%%%%%%%%%%%%%%%%%%

% This can be done in matlab's command window, a function file,
% or (for the hackers) an anonymous function.

M=rand(3,5);
m=size(M,1); M(1:m,1:m)\M
ans =
    1.0000         0         0   -2.5775   -1.3591
   -0.0000    1.0000         0    3.0365    2.0130
   -0.0000         0    1.0000    1.0462    0.8110

% In a file you can type
% function gj=eliminate(M)
%     m=size(M,1); gj=M(1:m,1:m)\M

% or if you want to be truly fancy
gj=@(M) M(:,1:size(M,1))\M
gj =
    @(M)M(:,1:size(M,1))\M
gj(M)
ans =
    1.0000         0         0   -2.5775   -1.3591
   -0.0000    1.0000         0    3.0365    2.0130
   -0.0000         0    1.0000    1.0462    0.8110
%% The "@" sign allows you to define functions in the cmd window

M=rand(4,7);
M(:,1:size(M,1))\M
ans =
  Columns 1 through 6
    1.0000   -0.0000         0         0   -0.3555    0.2786
         0    1.0000         0         0   -0.5936    0.7520
         0    0.0000    1.0000         0    0.3097    0.0227
         0    0.0000   -0.0000    1.0000    0.9243    0.1050
  Column 7
    0.7749
```

```
    0.2643
   -0.5538
    0.4798
gj(M)
ans =
  Columns 1 through 6
    1.0000   -0.0000         0         0   -0.3555    0.2786
         0    1.0000         0         0   -0.5936    0.7520
         0    0.0000    1.0000         0    0.3097    0.0227
         0    0.0000   -0.0000    1.0000    0.9243    0.1050
  Column 7
    0.7749
    0.2643
   -0.5538
    0.4798

%%%%%%%%%%%%%
% Problem 7 %
%%%%%%%%%%%%%

n=3; G3=fft(eye(n))/sqrt(n);
G3^4
ans =
    1.0000         0         0
         0    1.0000    0.0000
         0    0.0000    1.0000
n=4; G4=fft(eye(n))/sqrt(n);
G4^4
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
n=5; G5=fft(eye(n))/sqrt(n);
G5^4
ans =
  Columns 1 through 3
   1.0000             0.0000             -0.0000
   0.0000             1.0000             -0.0000 + 0.0000i
  -0.0000 - 0.0000i  -0.0000 - 0.0000i   1.0000 + 0.0000i
  -0.0000 + 0.0000i  -0.0000 + 0.0000i   0.0000 - 0.0000i
   0.0000             0.0000             -0.0000 + 0.0000i
  Columns 4 through 5
  -0.0000             0.0000
  -0.0000 - 0.0000i   0.0000
```

```
   0.0000 + 0.0000i  -0.0000 - 0.0000i
   1.0000 + 0.0000i  -0.0000 + 0.0000i
  -0.0000 - 0.0000i   1.0000

% Looks kinda like an identity matrix. Technical note for the curious:
% MATLAB writes 1 vs 1.0000 depending on whether it computes 1 exactly or
% 1 with a rounding error.  Similarly the imaginary part of 1 could have
% rounding errors.

%%%%%%%%%%%
%Problem 8%
%%%%%%%%%%%

A=eye(4)-diag([2 3 10],1);
A
A =
     1    -2     0     0
     0     1    -3     0
     0     0     1   -10
     0     0     0     1
inv(A)
ans =
     1     2     6    60
     0     1     3    30
     0     0     1    10
     0     0     0     1

% This should be fairly suggestive - it looks like the last column are all
% multiples of $10$, which can't be a coincidence. It is intuitive to then
% guess the answer as we've done by hand.

syms a b c
A=eye(4)-diag([a b c],1);
inv(A)
ans =
[ 1, a, a*b, a*b*c]
[ 0, 1,   b,   b*c]
[ 0, 0,   1,     c]
[ 0, 0,   0,     1]
diary off
```