Solution Set 2, 18.06 Fall '11

1. Do Problem 7 from 2.6.

   Solution. We perform the Gauß elimination w/o row exchange, and record below the matrices E_{ij} [recalling the textbook's notation: E_{ij} adds a multiple of j'th row to i'th row, while suppressing (in notation) what number we multiply by]:

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 2 & 2 \\ 3 & 4 & 5 \end{bmatrix} \overset{E_{21}}{\sim} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 3 & 4 & 5 \end{bmatrix} \overset{E_{31}}{\sim} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 4 & 2 \end{bmatrix} \overset{E_{32}}{\sim} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} =: U.$$

   The matrices we used here, were:

$$E_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad E_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix},$$

   with inverses:

$$E_{21}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_{31}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad E_{32}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}.$$

   We now combine their efforts. Recall that the result will also be *lower triangular and with only 1's on the diagonal*, which reduces the number of entries you need to compute (a little). Note also the order in which they come:

$$L = E_{21}^{-1}E_{31}^{-1}E_{32}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}.$$

   We finally *perform a test* (as for any problem, to completely rule out error):

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 2 & 2 \\ 3 & 4 & 5 \end{bmatrix} \quad \checkmark$$

   The procedure worked, since this was the A we started with.

   □

2. Do Problem 15 from 2.6.

   Solution. Since L is triangular we can solve $L\boldsymbol{c} = \boldsymbol{b}$ by back-substitution to get $\boldsymbol{c} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. Since U is triangular we can solve $U\boldsymbol{x} = \boldsymbol{c}$ by back-substitution to get $\boldsymbol{x} = \begin{bmatrix} -5 \\ 3 \end{bmatrix}$. Now multiply

$$A = LU = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 8 & 17 \end{bmatrix}.$$

To solve $A\boldsymbol{x} = \boldsymbol{b}$ subtract 4 times the first row from the second to obtain an equivalent system

$$\begin{bmatrix} 2 & 4 \\ 0 & 1 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

which we solve by back-substitution to get $\boldsymbol{x} = \begin{bmatrix} -5 \\ 3 \end{bmatrix}$. □

3. Do these problems about permutations.

   (a) Do Problem 8 from 2.7.

   (b) For each permutation matrix of size 3, tell me what each one does to the column vector $v = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$. For example, the identity matrix of size 3 sends v to itself, and so corresponds to $v = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$.

   (c) For 3 matrices of different $n > 1$, take a permutation matrix of size n and take their $(n!)$-th powers. What do you get? Think about why. The proof is outside the scope of this course (in MATLAB, we can create a random permutation matrix with `e=eye(n); a=e(randperm(n),:)`. To evaluate a factorial, use `factorial(n)`).

Solution. (a) There are two equivalent ways to think about this. Note there is exactly one 1 in the first row; we can choose that in n ways. Now there are $n-1$ spots in the second row for the unique 1 in that row; we can choose that in $(n-1)$ ways. Repeating this argument, we have a total of $n! = n(n-1)\cdots$ choices. Alternatively, a permutation is determined uniquely by where it sends $(x_1, x_2, \ldots, x_n)$ via left multiplication. The first component of the resulting vector can be any of n choices x_i, and the second component can be any of the remaining $(n-1)$ choices, etc. This gives the same count.

(b) We use the same $v = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ for all of these. We already talked about the identity; $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ sends v to $\begin{bmatrix} x_1 \\ x_3 \\ x_2 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ sends v to $\begin{bmatrix} x_2 \\ x_3 \\ x_1 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ sends v to $\begin{bmatrix} x_2 \\ x_1 \\ x_3 \end{bmatrix}$, $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ sends v to $\begin{bmatrix} x_3 \\ x_1 \\ x_2 \end{bmatrix}$, and $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ sends v to $\begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix}$,

(c) [See MATLAB code] You'll always get the identity matrix.

□

4. Do Problem 19 from 2.7.

Solution. (a) $(R^TAR)^T = R^TA^T(R^T)^T = R^TAR$, where we used that $A = A^T$ is symmetric in the last equality. Its shape is n by n.

(b) If you multiply R^TR using row-by-column method you see that the i^{th} diagonal entry of R^TR is the inner product of the i^{th} column of R with itself and is therefore nonnegative (equal to the squared length of the i^{th} column of R). □

5. Do Problem 22 from 2.7.

Solution. **First matrix**. If we start with

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

we can swap rows 1 & 2 by using $P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, to get $PA = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix}$.

Then the (non-row-exchanging) matrix E_{31},

$$E_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}, \quad \text{with} \quad E_{31}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix},$$

followed (i.e. further to the left) by next the (non-row-exchanging) matrix E_{32},

$$E_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix}, \quad \text{with} \quad E_{32}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix},$$

together serve to yield the result:

$$U = E_{32}E_{31}PA = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \quad \text{and}$$

$$L = E_{31}^{-1}E_{32}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{bmatrix}.$$

We perform the test:

$$PA = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix}, \quad \text{while also}$$

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \checkmark$$

Second matrix. Now starting a Gauß reduction of the matrix

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

we see that we can subtract a multiple of the 1st row from the 2nd row (and then a multiple of the 1st row from the 3rd row) to reduce the matrix. But then we will end with the pivots of rows 2 & 3 in the wrong order. So, we start by applying to A the permutation matrix of rows 2 & 3:

$$P_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \text{and thus} \quad P_{23}A = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 1 \\ 2 & 4 & 1 \end{bmatrix}.$$

Now, the mentioned (non-exchanging) row operations (Note: Row numbers and notation as applied to the *new* object of interest, $P_{23}A$) are:

$$E_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}, \quad \text{with} \quad E_{31}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix},$$

followed by

$$E_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{with} \quad E_{21}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

Thus here we get

$$U = E_{21}E_{31}P_{23}A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{and}$$

$$L = E_{31}^{-1}E_{21}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}.$$

We perform the test:

$$P_{23}A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 1 \\ 2 & 4 & 1 \end{bmatrix}, \quad \text{while also}$$

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 1 \\ 2 & 4 & 1 \end{bmatrix} \checkmark$$

so we computed correctly. □

6. Suppose a portfolio consists of a credit account, a checkings account, and a savings account, with the premise that for all these accounts you can be in the negatives (overdrawing, etc.) with no limit in either direction. Convince the grader that the set of portfolios has the structure of a vector space. How many dimensions are there? What do they mean? What does addition/subtraction of two vectors mean in this space? What does multiplication of a vector by a real number mean in this space?

   Solution. This is a 3-dimensional vector space of vectors (a, b, c), where (say) a, b, c correspond to the credit balance, checkings, and savings balance, respectively, with negative numbers corresponding to money you owe the bank. Two vectors can be added to mean one has gained money (possibly negative) in the three accounts. Multiplication by a real number can correspond to scaling the amount of money, or even changing the units of money (for example, if we want to convert from dollars to cents, we can scale every number by 100). □

7. Do Problem 14 from 3.1.

   Solution. (a) The subspaces of $\mathbf{R}^2$ are the zero subspace $\mathbf{Z}$, lines through 0 and $\mathbf{R}^2$ itself.

   (b) The space of all diagonal 2×2 matrices $\mathbf{D}$ we consider are more suggestively described as all D of the form

   $$D = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix}, \quad \text{for} \quad x, y \in \mathbb{R} \quad \text{freely chosen.}$$

   Hence by part (a), we see that all subspaces of $\mathbf{D}$ are: (A) The zero subspace $\mathbf{Z}$, (B) For any given line L through 0 in $\mathbb{R}^2$, the subspace corresponding to L that consists of all the matrices whose $(x, y) \in L$, and finally (C) The collection of all diagonal 2×2 matrices $\mathbf{D}$.

   □

8. Do Problem 27 from 3.1.

   Solution. (a) False. The zero vector $\boldsymbol{b} = 0$ belongs to any subspace and is always in the column space $\boldsymbol{C}(A)$ as well.

   (b) True. If A were nonzero, it would have a nonzero column and the column space of A would have to contain that column.

   (c) True. The subspaces spanned by $v_1, \ldots, v_n$ and by $2v_1, \ldots, 2v_n$ are the same because each $v_i = \frac{1}{2}(2v_i)$ is a linear combination of the $2v_i$'s and vice versa.

   (d) False. Take $A = I$. The column space of A is nonzero by part (b) whereas that of $A - I$ is zero. □

9. (a) On a computer, time the following operations for random matrices of size $100, 200$, and 800, averaged over say 50 trials: a) matrix multiplication, b) matrix addition, c) solving $Ax = b$. With MATLAB, your code for multiplication may look like this after setting an n:

```
t=50
v=zeros(t,1)
for i=1:t
a=rand(n); b=rand(n);
tic, a*b; v(i)=toc;
end
mean(v)
```

   (b) Now, compute the *rate* of computation (single-number additions and multiplications per second) for each of these three operations on these various n. Recall that matrix multiplication, matrix addition, and equation solution require roughly $2n^3, n^2$, and $(2/3)n^3$ single-number additions and multiplications, respectively (technical fact: the rate may converge to different numbers for reasons beyond the scope of this class, like memory traffic and cache misses).

Solution. [See MATLAB code] □

10. Create random 2x2x2 3-dimensional arrays (not matrices, which are 2-dimensional arrays!) of numbers (in MATLAB, `a=rand(2,2,2);`).

   (a) Create two such 3-dimensional arrays. Add them. Multiply them by constants. Get a feel for them. Question: do these arrays form a vector space?

   (b) Suppose we choose a different set of (m, n, p) besides $(2, 2, 2)$. Are the `rand(m,n,p)`'s a vector space? (avoid any of these being 1 since MATLAB does something funny)

   (c) Can I add a `rand(2,3,5)` and a `rand(4,7,8)`?

   (d) True or false: the collection of ALL 3-dimensional arrays form a vector space.

Solution. (a) [See MATLAB code] These do form a vector space, precisely because you can add and multiply by constants.

   (b) These also form a vector space; the 2's are not special in any way.

   (c) No. You should get an error if you tried. There is no nice way to define addition on these guys.

   (d) False, because we'd need to be able to, say, add two things with different (m, n, p)'s like from the last problem.

□

MATLAB code

```
%%%%%%%%%%%%%
% Problem 3 %
%%%%%%%%%%%%%

>> n=5;
>> %% we create permutation matrices by permuting rows of the identity
>> e=eye(n);p=randperm(n);
>> a=e(p,:)

a =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1

>> p=randperm(n);
>> a=e(p,:)

>> a^factorial(n)

ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1

%%%%%%%%%%%%%
% Problem 9 %
%%%%%%%%%%%%%

%% we omit the timing (a) because it was basically given in the
%% problem set.

t=50;
v=zeros(t,1);
for n=[100 200  800]
for i=1:t
  a=rand(n); b=rand(n);
```

```
  tic, a*b; v(i)=toc;
end
[n (2*n^3)/mean(v)/1e9]
%% On an old machine, the numbers suggest about 10 gigaflops
%% (the 1e9 converts to gigaflops by dividing by 10^9)

t=50;
v=zeros(t,1);
for n=[100 200  800]
for i=1:t
  a=rand(n); b=rand(n);
  tic, a+b; v(i)=toc;
end
[n (n^2)/mean(v)/1e6]
%% On an old machine, the numbers suggest about 100 or 200 megaflops

t=50;
v=zeros(t,1);
for n=[100 200  800]
for i=1:t
  a=rand(n); b=rand(n,1);
  tic, a\b; v(i)=toc;
end
[n (2/3*n^3)/mean(v)/1e9]
%% On an old machine, the numbers suggest about 1 or 2 gigaflops
end

%%%%%%%%%%%%%%
% Problem 10 %
%%%%%%%%%%%%%%

>> a=rand(2,2,2); b=rand(2,2,2);
>> a+b; 10*a;

%% The result is suppressed here via the ';', but it was allowed (no error)

>>
>> a=rand(2,3,4); b=rand(3,2,4);
>> a+b;
??? Array dimensions must match for binary array op.

%% This error message means that adding was not okay.
```