18.06 Problem Set 1 Solution

Due Wednesday, 11 February 2009 at 4 pm in 2-106.
Total: 145 points

Problem 1: If $\|\vec{v}\| = 7$ and $\|\vec{w}\| = 3$, what are the smallest and largest possible values of $\|\vec{v} + \vec{w}\|$ and $\vec{v} \cdot \vec{w}$?

$\boxed{\text{Solution}}$ (10 points)

$\|\vec{v} + \vec{w}\| \le \|\vec{v}\| + \|\vec{w}\| = 10$. $\|\vec{v} + \vec{w}\| \ge \|\vec{v}\| - \|\vec{w}\| = 4$. (5pts)

$|\vec{v} \cdot \vec{w}| \le \|\vec{v}\| \cdot \|\vec{w}\| = 21$. So $-21 \le \vec{v} \cdot \vec{w} \le 21$. (5pts)

The maximum is achieved when the vectors $\vec{v}$ and $\vec{w}$ are parallel and pointing to the same direction, for example, $\vec{v} = (7, 0, 0, \dots)$ and $\vec{w} = (3, 0, 0, \dots)$; the minimum is achieved when they are parallel but pointing to opposite directions, for instance, $\vec{v} = (7, 0, 0, \dots)$ and $\vec{w} = (-3, 0, 0, \dots)$.

REMARK: We should try not to restrict ourselves to the 3-dimensional case. The statement of this problem works for vectors in arbitrary dimensional space.

Problem 2: Let A and B be 4×4 matrices, and divide each of them into 2×2 chunks via $A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$ and $B = \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}$, where A_1 is the upper-left 2×2 corner, A_2 is the upper-right 2×2 corner, and so on. Let $C = AB$ be the 4×4 product of A and B, and similarly divide C into 2×2 chunks as $C = \begin{pmatrix} C_1 & C_2 \\ C_3 & C_4 \end{pmatrix}$.

(a) Give formulas for these 2×2 chunks $C_{1\dots4}$ in terms of matrix products and sums of the chunks $A_{1\dots4}$ and $B_{1\dots4}$ (your final formulas should *not* reference the individual numbers within those chunks).

(b) Justify your formulas by an example (come up with 4×4 matrices A and B with nonzero entries, multiply them to get C, and compare to your formulas in terms of 2×2 chunks—it is acceptable to check just one of the output 2×2 chunks, say C_2).

$\boxed{\text{Solution}}$ (15 points = 10+5)

(a) Similar to usual matrix multiplication, matrix multiplication by blocks formally has the same form.

$$C_1 = A_1B_1 + A_2B_3 \quad , \quad C_2 = A_1B_2 + A_2B_4,$$
$$C_3 = A_3B_1 + A_4B_3 \quad , \quad C_4 = A_3B_2 + A_4B_4.$$

In other words,

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} = \begin{pmatrix} A_1B_1 + A_2B_3 & A_1B_2 + A_2B_4 \\ A_3B_1 + A_4B_3 & A_3B_2 + A_4B_4 \end{pmatrix}$$

NOTE: the order of A_* and B_* cannot be reversed.

(b) For example,

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Then $C = AB = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 \\ 2 & 2 & 1 & 2 \\ 2 & 1 & 1 & 2 \end{pmatrix}$ and

$$\begin{aligned}
C_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 3 & 1 \end{pmatrix}, \\
C_2 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}, \\
C_3 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix}, \\
C_4 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}.
\end{aligned}$$

Problem 3: Invent a 3×3 "magic" matrix M_3 with entries $0, 1, \ldots, 8$, such that all rows and columns and diagonals add to 12 (e.g. the first row could be 7,2,3). Compute the products:

$$M_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \; \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} M_3, \; M_3 \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \; M_3 \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix}.$$

$\boxed{\text{Solution}}$ (10 points)

For example, take $M_3 = \begin{pmatrix} 7 & 0 & 5 \\ 2 & 4 & 6 \\ 3 & 8 & 1 \end{pmatrix}$, we have

$$M_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 12 \\ 12 \\ 12 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} M_3 = \begin{pmatrix} 12 & 12 & 12 \end{pmatrix},$$

$$M_3 \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 12 & 24 & 36 \\ 12 & 24 & 36 \\ 12 & 24 & 36 \end{pmatrix}, \qquad M_3 \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix} = \begin{pmatrix} 22 & 22 & 22 \\ 28 & 28 & 28 \\ 22 & 22 & 22 \end{pmatrix}.$$

REMARK: We get these results because multiplying by (1 1 1) is equivalent to just summing the rows (or columns), which by construction gives 12. Similarly, multiplying by (2 2 2) gives 24, etcetera. Multiplying by the matrix in the second to last problem just multiplies M_3 by each column, giving 12, 24, and 36. However, in the last part the columns are 1, 2, and 3, which does not just sum the rows of M_3 and hence does not give the same value in each row.

Problem 4: Choose a coefficient b that makes this system singular. Then choose a right-hand side g tha makes it solvable. Find two solutions in that singular case.

$$\begin{aligned} 3x + 4y &= 16 \\ 4x + by &= g \end{aligned}$$

$\boxed{\text{Solution}}$ (10 points)

We use Gauss elimination process.

$$\left(\begin{array}{cc|c} 3 & 4 & 16 \\ 4 & b & g \end{array} \right) \rightsquigarrow \left(\begin{array}{cc|c} 3 & 4 & 16 \\ 0 & b - 16/3 & g - 64/3 \end{array} \right).$$

So the system is singular if $b = 16/3$. It has solvable when $g = 64/3$. In this case, solutions to the linear system are just solutions for $3x + 4y = 16$. We may take the values of one variable arbitrarily and solve for the other. For example,

$$\begin{cases} x = 0 \\ y = (16 - 3 \cdot 0)/4 = 4 \end{cases}, \qquad \begin{cases} x = 4 \\ y = (16 - 3 \cdot 4)/4 = 1 \end{cases}.$$

Problem 5: Come up with 3×3 matrices A for problems $Ax = b$ such that:

(a) two row exchanges are needed in the elimination process to get a triangular form; then solve your system for some nonzero right-hand-side b.

(b) a row exchange is needed to keep going in elimination, but it still breaks down in a subsequent step. Give a right hand side b so that there is still a solution, and give a solution x.

$\boxed{\text{Solution}}$ (20 points = 10+10)

(a) We need a non-singular system, where the Gauss elimination method succeeds, but two row-exchanges are needed in the process. For example,

$$\left(\begin{array}{ccc|c} 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 7 \\ 1 & 0 & 1 & 5 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 0 & 1 & 5 \\ 0 & 0 & 1 & 7 \\ 0 & 1 & 1 & 3 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 0 & 1 & 5 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 7 \end{array}\right).$$

(b) We need a singular system where elimination fails – but only after at least one successful step that required a row exchange. For example,

$$\left(\begin{array}{ccc|c} 0 & 2 & 0 & 3 \\ 1 & 1 & 1 & 7 \\ 1 & 3 & 1 & 10 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 0 & 2 & 0 & 3 \\ 1 & 3 & 1 & 10 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 0 & 2 & 0 & 3 \\ 0 & 2 & 0 & 3 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Problem 6: In elimination, we do operations on *rows*, which corresponds to multiplying on the left by elimination matrices. Cal Q. Luss, a Harvard student, suggests that we should do operations on *columns* instead (e.g. subtracting a multiple of one column from another, or swapping two columns).

(a) Do a sequence of these "column elimination" operations on your 3×3 matrix from problem 5(a), and show that you can still get an upper triangular matrix. What happens if you try to do column elimination on your matrix from 5(b)?

(b) Suppose, for a matrix A, that one of our "column elimination" steps consists of subtracting 3 times column 1 from column 2. Express this operation in matrix form, as A multiplied somehow by some "column elimination" matrix. Check your answer on your 3×3 matrix from 5(a).

(c) Clearly explain to Cal why "column elimination" is *not* particularly useful for solving $Ax = b$, even though you can convert A to a triangular matrix (explain what happens to the system of equations, perhaps in terms of elimination matrices).

(d) Write down a different set of linear equations in terms of your 3×3 matrix A from 5(a) that *is* solvable by column elimination (hint: think of row vectors).

$\boxed{\text{Solution}}$ (20 points=5+5+5+5)

(a) Using column elimination from right to left and the most bottom nonzero element as the pivot element, we can get upper triangular matrices in both cases.

$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} -1 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$\begin{pmatrix} 0 & 2 & 0 \\ 1 & 1 & 1 \\ 1 & 3 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 & 2 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

In the case 5(b), we can make the first column all zero. This suggests that if a matrix is singular for row elimination then it is also singular for column elimination.

REMARK: The equivalence of being singular for row elimination and column elimination is true in general. However, the proof is not obvious and will be considered later in the course.

(b) Just as multiplication by an elimination matrix on the left can be viewed as taking linear combinations of the rows of A, multiplying by an elimination matrix on the right can be viewed as taking linear combinations of the columns of A. Hence, Subtracting 3 times column 1of A from column 2 of A is equivalent to making the second column of the elimination matrix to be $(-3\ 1\ 0)^{\mathrm{T}}$. Thus, this operation is to send A to $A \begin{pmatrix} 1 & -3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. For example,

$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & -3 & 1 \end{pmatrix}.$$

REMARK: The actual elimination matrices will be lower-triangular, not upper-triangular as in the example here, since really you subtract later columns from earlier ones and not vice-versa, as seen in part (a).

(c) Row elimination works because it corresponds to multiplying both sides of $Ax = b$ on the left by a sequence of elimination matrices to simplify A. For column elimination, we need to multiply A on the right by elimination matrices, but we can't simply multiply both sides of $Ax = b$ on the right by elimination matrices because x is in the way (in fact, it doesn't make sense to multiply a column vector by a matrix on the right).

If anything, you would have to put the elimination matrices in the "middle" of Ax as $AEE^{-1}x = UE^{-1}x$ where E is the product of the elimination matrices. Actually, you could argue that this is not completely crazy, since E and E^{-1} are lower triangular — you have effectively formed a UL factorization of A (analogous to LU). You then have an equation of the form $ULx = b$, which can be solved easily as a sequence of two triangular systems. So, Cal is vindicated after all. [Note to grader: students get full marks if they don't realize this, and just make an argument equivalent to the first paragraph.]

(d) Row elimination can be used to solve equations of the kind

$$\begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}.$$

We proceed similar to column operations as follows:

$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ \hline 1 & 2 & 3 \end{pmatrix} \rightsquigarrow \begin{pmatrix} -1 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 1 \\ \hline -2 & 2 & 3 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \\ \hline 2 & -2 & 3 \end{pmatrix}$$

So $x_1 = 2$, $-x_1 - x_2 = -2 \Rightarrow x_2 = 0$, and $x_1 + x_2 + x_3 = 3 \Rightarrow x_3 = 1$.

Problem 7: Suppose A is invertible and you exchange its first two rows to obtain a new matrix B. Is the new matrix B invertible, and how would you find B^{-1} from A^{-1}?

$\boxed{\text{Solution}}$ (10 points)

Since B is obtained from A by exchanging the first two rows, we know

$$B = MA \text{ with } M = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

So, $BA^{-1}M = MAA^{-1}M = MM = I$ and hence $B^{-1} = A^{-1}M$. In other words, B^{-1} is obtained from A^{-1} by exchanging the first two columns.

Problem 8: If the product $C = AB$ is invertible (and A and B are square), find a formula for A^{-1} that involves C^{-1} and B. (Hence, it is not possible to multiply a *non*-invertible matrix on the by another matrix and obtain an *invertible* matrix as a result.)

$\boxed{\text{Solution}}$ (10 points)

Note that $ABC^{-1} = CC^{-1} = I$. So $A^{-1} = BC^{-1}$.

Remark: In general, if $A = BC$ with B or C invertible, we can multiply both sides of the equation by B^{-1} *on left* or C^{-1} *on right*, respectively. Then, we obtain $B^{-1}A = C$ or $AC^{-1} = B$. We would like to emphasize that when dealing with matrices, we need to carefully distinguish left multiplications and right multiplications. In very rare cases, they might be equal.

Problem 9: Solve the system $Ax = b$ for

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}, \; b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

using elimination and backsubstitution. Compute A^{-1} using Gauss–Jordan, and verify that $A^{-1}b$ gives the same x.

$\boxed{\text{Solution}}$ (20 points)

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 2 & 1 & 3 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 3 \end{array}\right).$$

By backsubstitution, $x_3 = 3$, $x_2 + 3x_3 = 0$, and $x_1 = 1$. Hence, we have

$$\begin{cases} x_1 = 1 \\ x_2 = -9 \\ x_3 = 3 \end{cases}$$

Calculating A^{-1} via Gauss–Jordan.

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 1 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -2 & 1 & -3 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right).$$

So $A^{-1}b$ is given by

$$A^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & -3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ -9 \\ 3 \end{pmatrix}$$

The two results coincide.

Problem 10: Solve the equations in problem 9 using Matlab. (on Athena: `add matlab && matlab` to run Matlab). You can do this by the commands:

```
A = [1 0 0; 2 1 3; 0 0 1]
b = [1; 2; 3]
x = A \ b
```

and also check your A^{-1} by computing the inverse in Matlab with `inv(A)`.

$\boxed{\text{Solution}}$ (10 points)

```
>> A = [1 0 0; 2 1 3; 0 0 1]

A =

     1     0     0
     2     1     3
     0     0     1

>> b = [1; 2; 3]

b =

     1
     2
     3

>> x = A \ b

x =

     1
    -9
     3
```

```
>> inv(A)

ans =

     1     0     0
    -2     1    -3
     0     0     1
```

Problem 11: Now, let's solve larger systems in Matlab. Much larger systems. To save the trouble of coming up with equations by hand, we'll let Matlab choose them at random using the `rand(m,n)` command, which creates a random $m \times n$ matrix:

```
A = rand(100,100);
b = rand(100,1);
tic; x = A \ b; toc
```

Notice the semicolons (;) after the commands: this suppresses the output, which is useful if you don't want to print out the 100×100 matrix A. The above code was for 100×100. The `tic` and `toc` commands print out the time, in seconds, for the `x = A \ b` command between them. Now try doubling this to 200. Then to 400. Then to 800. Then to 1600. By what factor, on average, does the computation time increase each time you double the number of rows and columns?

$\boxed{\text{Solution}}$ (10 points)

```
>> maxNumCompThreads(1);
>> A = rand(100,100);
>> b = rand(100,1);
>> tic; x = A \ b; toc
Elapsed time is 0.000723 seconds.
>> A = rand(200,200);
>> b = rand(200,1);
>> tic; x = A \ b; toc
Elapsed time is 0.003349 seconds.
>> A = rand(400,400);
>> b = rand(400,1);
>> tic; x = A \ b; toc
```

```
Elapsed time is 0.017764 seconds.
>> A = rand(800,800);
>> b = rand(800,1);
>> tic; x = A \ b; toc
Elapsed time is 0.117945 seconds.
>> A = rand(1600,1600);
>> b = rand(1600,1);
>> tic; x = A \ b; toc
Elapsed time is 0.761077 seconds.
```

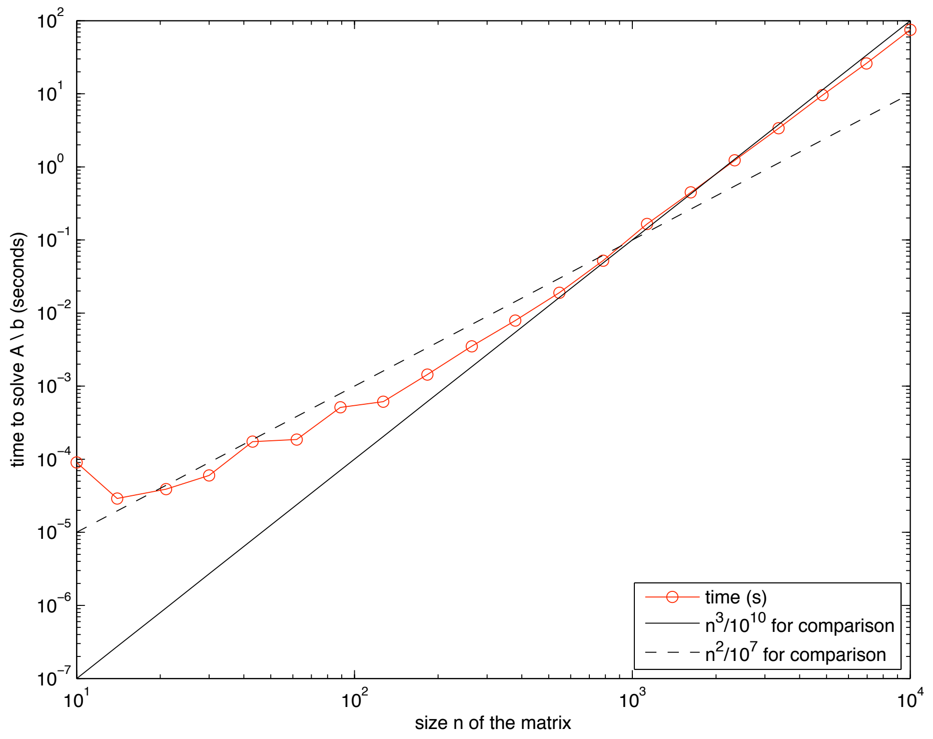
REMARK: The reason that we have the first line of the code is because we want to tell the computer to use only one processor when computing the result.

Professor Johnson wrote a loop computing more examples from n=10 to n=10000 and plotting the results, on his 2.83Gz Intel Core 2 Xeon. Here is his code

```
>> maxNumCompThreads(1)

ans =

     1

>> n = round(logspace(1,4,20))

n =

  Columns 1 through 6

          10          14          21          30          43          62

  Columns 7 through 12

          89         127         183         264         379         546

  Columns 13 through 18

         785        1129        1624        2336        3360        4833

  Columns 19 through 20

        6952       10000
```

```
>> for i = 1:length(n)
>> A = rand(n(i),n(i));
>> b = rand(n(i),1);
>> tic; x = A \ b; t(i) = toc;
>> end

>> format long
>> t

t =

  Columns 1 through 3

   0.000090000000000   0.000029000000000   0.000039000000000

  Columns 4 through 6

   0.000060000000000   0.000174000000000   0.000186000000000

  Columns 7 through 9

   0.000511000000000   0.000614000000000   0.001439000000000

  Columns 10 through 12

   0.003521000000000   0.007885000000000   0.019018000000000

  Columns 13 through 15

   0.052252000000000   0.165369000000000   0.447228000000000

  Columns 16 through 18

   1.225485000000000   3.363740000000000   9.558894000000000

  Columns 19 through 20

  26.084814000000001  75.238491999999994
```

```
>> loglog(n,t, 'ro-', n, n.^3 / 10^10, 'k-', n, n.^2 / 10^7, 'k--')
>> xlabel('size n of the matrix')
>> ylabel('time to solve A \ b (seconds)')
>> legend('time (s)', 'n^3/10^{10} for comparison', 'n^2/10^7 for
comparison','Location','SouthEast')
```



This way, from the results you can clearly see that it asymptotically approaches roughly n^3 scaling (not n^2 scaling), consistent with the prediction in class based on the operation count for Gaussian elimination. However, from the same results you can also see that the pure n^3 scaling is not obtained until fairly large n, so for the moderate n asked for by the pset the students will see less than a factor of 8 for each doubling of n, as we noticed in our data above.