18.06 Problem Set 1

Due Wednesday, 11 February 2008 at 4pm in the undergrad. math office.

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

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 $C_{1...4}$ in terms of matrix products and sums of the chunks $A_{1...4}$ and $B_{1...4}$ (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).

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:

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

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.

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

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.

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).

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}?

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.)

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.

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)`.

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?