18.06 Problem Set 2 - Solutions

Due Wednesday, 19 September 2007 at 4 pm in 2-106.

Problem 1: (10=5+5) (a) Do problem 12 from section 2.4 (P67) in your book.

First $AB = \begin{pmatrix} a & 0 \\ c & 0 \end{pmatrix}$, $BA = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix}$ and $AB = BA$ implies $b = c = 0$.

Now $AC = \begin{pmatrix} 0 & a \\ 0 & c \end{pmatrix} = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}$, $CA = \begin{pmatrix} c & d \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & d \\ 0 & 0 \end{pmatrix}$, so $AC = CA$ and $b = c = 0$ implies $a = d$.

Thus $A = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} = a \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

(b) Do problem 36 from section 2.4 (P70) in your book.

It is easy to compute

$$A \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} a+b & a+b \\ c+d & c+d \end{pmatrix},$$

and

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} A = \begin{pmatrix} a+c & b+d \\ a+c & b+d \end{pmatrix}.$$

Thus we have $a+b = a+c = b+d$ and $c+d = a+c = b+d$. Both are equivalent to $a = d, b = c$. Thus $A = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$ for arbitrary a and b.

Problem 2: (10) Do problem 19 from section 2.5 (P80) in your book.

We use Gauss-Jordan elimination to find the inverse. Begin with

$$\begin{pmatrix} 4 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 4 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Add the first three rows to the last row, we get

$$\begin{pmatrix} 4 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Then add the last row to the first three rows, we get

$$\begin{pmatrix} 5 & 0 & 0 & 0 & 2 & 1 & 1 & 1 \\ 0 & 5 & 0 & 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 5 & 0 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Divide the first three rows by 5, we get

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 2/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 1/5 & 2/5 & 1/5 & 1/5 \\ 0 & 0 & 1 & 0 & 1/5 & 1/5 & 2/5 & 1/5 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Finally add negative of the first three rows to the last row, we get

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 2/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 1/5 & 2/5 & 1/5 & 1/5 \\ 0 & 0 & 1 & 0 & 1/5 & 1/5 & 2/5 & 1/5 \\ 0 & 0 & 0 & 1 & 1/5 & 1/5 & 1/5 & 2/5 \end{pmatrix}$$

Thus $a = 2/5$ and $b = 1/5$.

(Another way: since $\begin{pmatrix} a & b & b & a \\ b & a & b & a \\ b & b & a & a \\ b & b & b & b \end{pmatrix}$ is the inverse, we multiply it to the matrix 5∗eye(4)-ones(4,4), then the (1,1) element is $1 = 4a - 3b$, and the (1,2) element is $0 = 2b - a$. This implies $a = 2/5$ and $b = 1/5$.)

For the matrix 6∗eye(5)-ones(5,5), one can show by exactly the same way that $a = 1/3 = 2/6$ and $b = 1/6$. And in general for k∗eye($k-1$)-ones($k-1, k-1$), we have $a = 2/k, b = 1/k$.

Problem 3: (10) Do problem 9 from section 2.6 (P92) in your book.

For the first one, we have

$$\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} \begin{pmatrix} d & e \\ 0 & f \end{pmatrix} = \begin{pmatrix} d & e \\ ld & le + f \end{pmatrix}.$$

The first column gives $d = 0$ while $dl = 2$, which is impossible.

For the second one, we have

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l & 1 & 0 \\ m & n & 1 \end{pmatrix} \begin{pmatrix} d & e & g \\ 0 & f & h \\ 0 & 0 & i \end{pmatrix} = \begin{pmatrix} d & e & g \\ dl & le+f & * \\ dm & me+nf & * \end{pmatrix}.$$

The first row gives $d = 1, e = 1, g = 0$, and thus the first column gives $l = 1, m = 1$. Now the (2,2) element gives $1 = le + f = 1 + f$, which implies $f = 0$, and thus the (3,2) element is $me + nf = 1 + 0 = 1$, contradiction.

Problem 4: (10) Do problem 24 from section 2.6 (P94) in your book.

The new elimination method looks like

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 5 \\ 2 & 1 & 1 & 0 & 8 \\ 0 & 1 & 3 & 2 & 8 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix} \overset{E_{21}}{\rightsquigarrow} \begin{pmatrix} 1 & 1 & 0 & 0 & 5 \\ 0 & -1 & 1 & 0 & -2 \\ 0 & 1 & 3 & 2 & 8 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix} \overset{E_{34}}{\rightsquigarrow} \begin{pmatrix} 1 & 1 & 0 & 0 & 5 \\ 0 & -1 & 1 & 0 & -2 \\ 0 & 1 & 1 & 0 & 4 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix} \overset{E_{32}}{\rightsquigarrow} \begin{pmatrix} 1 & 1 & 0 & 0 & 5 \\ 0 & -1 & 1 & 0 & -2 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix}$$

Thus we have
$x_3 = 1$,
$x_3 + x_4 = 2$ implies $x_4 = 1$,
$-x_2 + x_3 = -2$ implies $x_2 = 3$,
$x_1 + x_2 = 5$ implies $x_1 = 2$.

Finally we point out that this "new" elimination method is not new: it is equivalent to the standard elimination if you reorder the row first.

Problem 5: (10) Show that the product of two lower triangular matrices is a lower triangular matrix. What are the diagonal entries of this product? (This tells us why the product of the elimination matrices, or their inverses, is lower triangular, since the elimination matrices are lower triangular.)

Proof: Let $A = (a_{ij})$ and $B = (b_{ij})$ be two lower triangular matrices, i.e. $a_{ij} = b_{ij} = 0$ for all $i < j$. Let $C = (c_{ij})$ be the product of A and B, then we have

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

Now assume $i < j$, then no matter what k is, either $i < k$ or $k < j$ (otherwise $i \geq k \geq j$, contradicts with $i < j$). Thus either $a_{ik} = 0$ or $b_{kj} = 0$. This implies $c_{ij} = 0$ for all $i < j$. In other words, C is a lower triangular matrix.

For the diagonal entries, we have

$$c_{ii} = \sum_{k=1}^{n} a_{ik} b_{ki} = a_{ii} b_{ii},$$

since the other terms in the summation above vanishes, by the same argument. Thus the diagonal entries of AB is just the product of the diagonal entries of A and B.

Problem 6: (10=5+5) Let $A = \begin{pmatrix} 1 & 3 & -1 \\ 2 & 5 & 1 \\ 3 & 4 & 2 \end{pmatrix}$.

(a) Use Gauss-Jordan elimination to calculate A^{-1}.

By Gauss-Jordan elimination,

$$\begin{pmatrix} 1 & 3 & -1 & 1 & 0 & 0 \\ 2 & 5 & 1 & 0 & 1 & 0 \\ 3 & 4 & 2 & 0 & 0 & 1 \end{pmatrix} \overset{E_{21}}{\rightsquigarrow} \begin{pmatrix} 1 & 3 & -1 & 1 & 0 & 0 \\ 0 & -1 & 3 & -2 & 1 & 0 \\ 3 & 4 & 2 & 0 & 0 & 1 \end{pmatrix}$$

$$\overset{E_{31}}{\rightsquigarrow} \begin{pmatrix} 1 & 3 & -1 & 1 & 0 & 0 \\ 0 & -1 & 3 & -2 & 1 & 0 \\ 0 & -5 & 5 & -3 & 0 & 1 \end{pmatrix}$$

$$\overset{E_{32}}{\rightsquigarrow} \begin{pmatrix} 1 & 3 & -1 & 1 & 0 & 0 \\ 0 & -1 & 3 & -2 & 1 & 0 \\ 0 & 0 & -10 & 7 & -5 & 1 \end{pmatrix}$$

$$\rightsquigarrow \begin{pmatrix} 1 & 3 & -1 & 1 & 0 & 0 \\ 0 & 1 & -3 & 2 & -1 & 0 \\ 0 & 0 & 1 & -7/10 & 1/2 & -1/10 \end{pmatrix}$$

$$\rightsquigarrow \begin{pmatrix} 1 & 3 & 0 & 3/10 & 1/2 & -1/10 \\ 0 & 1 & 0 & -1/10 & 1/2 & -3/10 \\ 0 & 0 & 1 & -7/10 & 1/2 & -1/10 \end{pmatrix}$$

$$\rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & 3/5 & -1 & 4/5 \\ 0 & 1 & 0 & -1/10 & 1/2 & -3/10 \\ 0 & 0 & 1 & -7/10 & 1/2 & -1/10 \end{pmatrix}$$

(b) Find the LU decomposition of A.

We have seen from above that

$$U = \begin{pmatrix} 1 & 3 & -1 \\ 0 & -1 & 3 \\ 0 & 0 & -10 \end{pmatrix},$$

and moreover, since

$$E_{21} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, E_{31} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}, E_{32} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -5 & 1 \end{pmatrix},$$

we get

$$L = E_{21}^{-1} E_{31}^{-1} E_{32}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 5 & 1 \end{pmatrix}.$$

Problem 7: (10=2+2+2+2+2) True of false (give a reason if true; give a counterexample if false):

(a) The block matrix $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$ is always symmetric.

False.

An counterexample: $A = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}$, then $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 \end{pmatrix}$, which is definitely not symmetric.

In fact, since $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}^T = \begin{pmatrix} A^T & 0 \\ 0 & A^T \end{pmatrix}$, it is symmetric if and only if A is symmetric.

(b) If A and B are $n \times n$ symmetric matrices, so is AB.

False.

An counterexample: let $A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, both are symmetric, however, $AB = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, not symmetric.

In fact, if A and B are symmetric, then $(AB)^T = B^T A^T = BA$, thus AB is symmetric if and only if $AB = (AB)^T = BA$, i.e. A and B commute.

(c) If A is symmetric and $I + A$ is invertible, then $(I + A)^{-1}$ is symmetric.

True.

Proof: Since A is symmetric and I is symmetric, we see $I + A$ is symmetric. Now

$$((I + A)^{-1})^T = ((I + A)^T)^{-1} = (I + A)^{-1}$$

is symmetric.

(d) If A is n by n symmetric matrix, then $\mathbf{x} \cdot A^2\mathbf{y} = (A\mathbf{x}) \cdot (A\mathbf{y})$ for any $n \times 1$ vectors $\mathbf{x}$ and $\mathbf{y}$.

True.

Proof: We have $\mathbf{x} \cdot A^2\mathbf{y} = (AA\mathbf{y})^T\mathbf{x} = (A\mathbf{y})^T A^T\mathbf{x} = (A\mathbf{y})^T A\mathbf{x} = A\mathbf{x} \cdot A\mathbf{y}$.

(e) Any $n \times n$ matrix can be written as a sum of a symmetric matrix and an anti-symmetric matrix. (*anti-symmetric means* $A^T = -A$.)

True.

Proof: For any matrix A, it is easy to see that $\frac{A+A^T}{2}$ is symmetric, while $\frac{A-A^T}{2}$ is anti-symmetric. Now $A = \frac{A+A^T}{2} + \frac{A-A^T}{2}$ is a sum of a symmetric matrix and an anti-symmetric matrix.

Problem 8: (10=5+5) (a) A matrix A is *idempotent* if $A^2 = A$. Show that $A = \begin{pmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{pmatrix}$ is idempotent.

By direct computation, we have

$$A^2 = \begin{pmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{pmatrix} \begin{pmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{pmatrix} = \begin{pmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{pmatrix}.$$

(b) A matrix A is *nilpotent of class* p if $A^p = 0$ but $A^{p-1} \neq 0$. Show that $A = \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix}$ is nilpotent.

By direct computation, we have

$$A^2 = \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 3 & 9 \\ -1 & -1 & -3 \end{pmatrix},$$

and

$$A^3 = A^2A = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 3 & 9 \\ -1 & -1 & -3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Thus A is nilpotent of class 3.

Problem 9: (10=5+5) Construct a random 1000×1000 matrix A in Matlab, and a random right hand side $\mathbf{b}$:

```
>> A = rand(1000,1000);
>> b = rand(1000,1);
```

We can solve $Ax = \mathbf{b}$ in two ways: by

```
>> x =A \ b ;
```

which (inside Matlab) uses elimination (via an LU decomposition) or by

```
>> x= inv(A) * b;
```

which explicitly computes the inverse of the matrix A.
(a) try both methods of computing $\mathbf{x}$, timing them with the tic and toc commands as in problem set 1. Which is faster? Is this consistent with how hard it is to apply the two methods by hand, using the techniques we learned in class?

I tried 8 times, and my output is

```
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.276290 seconds.
Elapsed time is 0.668278 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.281526 seconds.
Elapsed time is 0.665426 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.279839 seconds.
Elapsed time is 0.667069 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.282267 seconds.
Elapsed time is 0.669732 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.280064 seconds.
Elapsed time is 0.667090 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.274248 seconds.
```

Elapsed time is 0.664005 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.277578 seconds.
Elapsed time is 0.665961 seconds.
>> A=rand(1000, 1000);b=rand(1000,1);tic;x=A\b;toc,tic;y=inv(A)*b;toc
Elapsed time is 0.273459 seconds.
Elapsed time is 0.723578 seconds.

As we can see, the first method is much faster. The reason is that in the first method, we only need to do Gauss elimination and then substitution back; while in the second case we need to do Gauss-Jordan elimination which takes more time.

(b) In a perfect world, these would give the same answer, and in both cases $A\mathbf{x}$ would equal $\mathbf{b}$, but this is not the case in reality. Computers store only about 15 decimal places of each number and (almost) every arithmetic operation involves some roundoff error. Compute the error $|A\mathbf{x} - \mathbf{b}|/|\mathbf{b}|$, via

```
>> norm(A*x - b) / norm(b)
```

for both methods of computing $\mathbf{x}$ above. Repeat for a few random matrices A. Which method seems to be more accurate in the face of roundoff error? Why do you think that might be?

I tried 8 times, and what I got is
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 3.0528e-13
ans = 1.0520e-12
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 1.9263e-13
ans = 4.1999e-13
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 1.5826e-13
ans = 1.2437e-12
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 2.2853e-13
ans = 5.7380e-13
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b),

norm(A*y-b)/norm(b)
ans = 1.3031e-12
ans = 3.6893e-12
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 5.9353e-13
ans = 8.9797e-13
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 7.2618e-13
ans = 9.3491e-13
>> A=rand(1000, 1000); b=rand(1000, 1); x=A\b; y=inv(A)*b; norm(A*x-b)/norm(b), norm(A*y-b)/norm(b)
ans = 2.3028e-13
ans = 8.2731e-13

It turns out that the first way is more accurate. The reason is that the first method involves less computations, and thus less roundoff error.

Problem 10: (10=4+(3+3)) Matlab has a command $inv(A)$ to compute the inverse of a matrix A. It also has a command $pinv(A)$ to compute something rather mysterious, a "pseudo-inverse" of A. Without poring over the Matlab documentation, let's try to figure out what $pinv$ does.
(a) Verify that inv and $pinv$ do the same thing (up to roundoff error, $14-15$ decimal places) if A is invertible. e.g. try a couple random 10×10 matrices.

I tried a couple of examples, and the first 5 digits are all the same for a random 10×10 matrices. To find out the difference, I computed the norm of $inv(A) - pinv(A)$, and get:
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 6.1877e-15
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 4.3995e-14
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 2.8663e-14
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 2.8025e-14
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 1.7367e-14

```
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 8.1975e-15
>> A=rand(10,10);norm(inv(A)-pinv(A))
ans = 1.1836e-13
```

I also tried a couple of examples with bigger random matrices, and get

```
>> A=rand(100,100);norm(inv(A)-pinv(A))
ans = 1.1544e-11
>> A=rand(100,100);norm(inv(A)-pinv(A))
ans = 6.7407e-11
>> A=rand(100,100);norm(inv(A)-pinv(A))
ans = 1.2312e-12
>> A=rand(100,100);norm(inv(A)-pinv(A))
ans = 3.9565e-13
>> A=rand(100,100);norm(inv(A)-pinv(A))
ans = 1.1716e-12
>> A=rand(100,100);norm(inv(A)-pinv(A))
ans = 1.7170e-13
>> A=rand(500,500);norm(inv(A)-pinv(A))
ans = 6.1505e-11
>> A=rand(500,500);norm(inv(A)-pinv(A))
ans = 4.1010e-10
>> A=rand(500,500);norm(inv(A)-pinv(A))
ans = 9.6185e-12
```

So as we can see, they are almost the same matrix; and the difference is larger if the matrix is bigger, as we expected.

(b) Consider a singular matrix

```
>> A = [1 2; 2 4]
```

The equation $A\mathbf{x} = \mathbf{b}$ has no solution for most right-hand sides $\mathbf{b}$; pick one such right-hand-side $\mathbf{b}$ (i.e. with no solution $\mathbf{x}$). However $pinv(A)$ gives something(!), although we don't know yet what it means, and we can compute

```
>> x= pinv(A) * b;
```

Let's try to figure out what this x could mean. If we compute $A * \mathbf{x}$, we won't get $\mathbf{b}$ (we can't!), but we get something else, call it bx:

```
>> bx = A * x;
```

For comparison, let's compute what we get when we try 100 random **x** values, call it br:

```
>> br = A * rand(2, 100);
```

To try to understand this, plot **b**, bx, and br, using the command:

```
>>plot(b(1),b(2),'rx', bx(1),bx(2),'bo', br(1,:),br(2,:),'k.')
```

which puts a red "x" where **b** is, a blue "o" where bx is, and a black dot at each br. You will need to type:

```
>> axis equal
```

to force Matlab to plot the x and y axes with equal scaling. Now:
i) The br values have a simple pattern in the plot. Why?

The br values lies on the line $y = 2x$. The line is the column space of A, and br must lies in this column space.

ii) Infer the rule that Matlab used to pick $\mathbf{x} = pinv(A) * \mathbf{b}$. (You don't have to figure out how to compute $pinv(A)$ or x. Just describe what distinguishes bx.)
We will return to this kind of singular problem later in the term.

I used different points $\mathbf{b} = [0; 10], [0; 3], [1; 4], [3; 2]$, and get

>> A=[1 2;2 4];b=[0;10]; x=pinv(A)*b; bx=A*x; br=A*rand(2,100); plot(b(1), b(2), 'rx', bx(1), bx(2), 'bo', br(1,:), br(2,:), 'k.'), axis equal

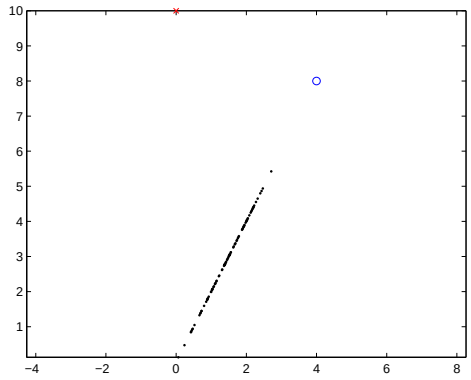

Figure 1: $\mathbf{b} = [0; 10]$

>> A=[1 2;2 4];b=[0;3]; x=pinv(A)*b; bx=A*x; br=A*rand(2,100); plot(b(1), b(2), 'rx', bx(1), bx(2), 'bo', br(1,:), br(2,:), 'k.'), axis equal

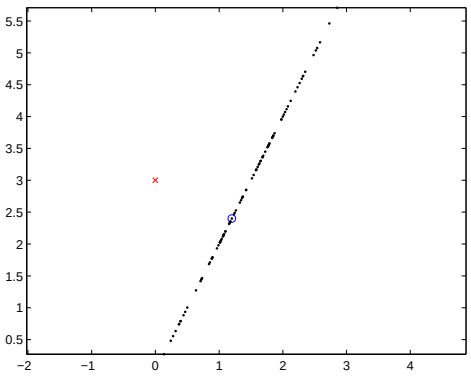

Figure 2: $\mathbf{b} = [0; 3]$

>> A=[1 2;2 4];b=[1;4]; x=pinv(A)*b; bx=A*x; br=A*rand(2,100); plot(b(1), b(2), 'rx', bx(1), bx(2), 'bo', br(1,:), br(2,:), 'k.'), axis equal

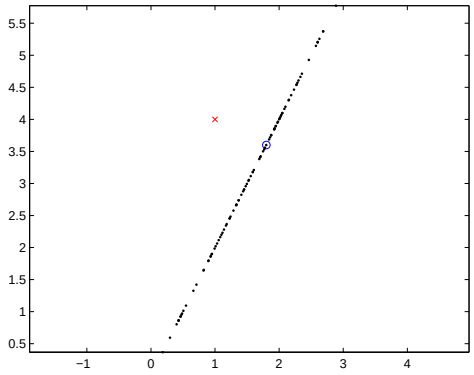

Figure 3: $\mathbf{b} = [1; 4]$

>> A=[1 2;2 4];b=[3;2]; x=pinv(A)*b; bx=A*x; br=A*rand(2,100); plot(b(1), b(2), 'rx', bx(1), bx(2), 'bo', br(1,:), br(2,:), 'k.'), axis equal

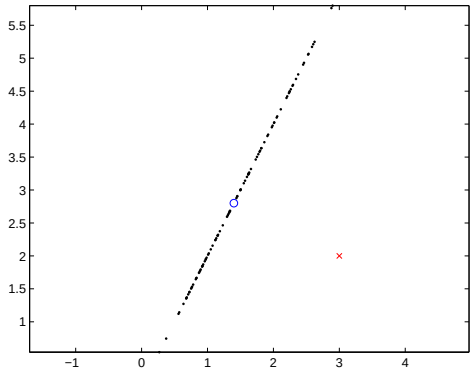

Figure 4: $\mathbf{b} = [3; 2]$

It turns out that the blue circle (the point bx) is the nearest point on the line $y = 2x$ to the red "x"(the point $\mathbf{b}$). (you must use the command "axis equal", otherwise it is hard to find this.)