AMS 27L **LAB #3** **Winter 2009**

Linear Algebra and Matrix Operations

Objectives:

1. To learn matrix operations using Matlab
2. To solve linear systems of equations
3. To practice coding skills in Matlab

1 Linear Algebra in Matlab

The name MatLab comes from "MATrix LABoratory" so Matlab is very good at linear algebra.

Problem #1: Type commands (a)-(h) into the command line and press `<Enter>` to see the result.

(a) A = [1,2,3;4,5,6;7,8,9] *Create the square matrix 'A'*

(b) A(3,2) *Reference the number in the* 3^{rd} *row,* 2^{nd} *column.*

(c) A(1,1:2) *Reference only part of the matrix*

(d) B = A(1,1:end) *Create a row vector from part of 'A'*

(e) B, transpose(B) *Find the transpose of matrix B.*

(f) C = [A; B] *Shortcut for vertical concatenation of matrices - vertcat().*

(g) D = [A transpose(B)] *Shortcut for horizontal concatenation of matrices - horzcat().*

(h) size(B) *Get the dimensions of a matrix*

(i) 2*A *Multiply each element of 'A' by 2*

(j) A*A *Perform matrix multiplication*

(k) A.*A *Multiply matrices of same dimension element to element*

(l) det(A) *Compute the determinant of a matrix*

(m) inv(A) *Taking the inverse of a matrix.*

(n) diag(A) *Get the diagonal elements of a matrix*

(o) $[V, E]$ = eig(A) *Solving for eigenvalues and eigenvectors of A*

(p) C = eye(3) *Create the square 3x3 identity matrix*

(q) C = [1,2,3] *Create a row vector*

(r) D = [4;5;6] *Create a column vector*

2 Determined Systems of Equations – A Review

A system is *determined* if it has the same number of equations and unknowns. It is *underdetermined* if it has fewer equations than unknowns, and *overdetermined* if it has more equations than unknowns. In this section we examine determined systems.

There are three possibilities for the set of all solutions to a system of linear equations:

- there are no solutions
- there are infinitely many solutions
- there is a unique solution

The next example illustrates the use of the Matlab command " `rref`" to solve determined systems.

Example: *Find all solutions to the system*

$$\begin{aligned} 4x_1 - 4x_2 - 8x_3 &= 27 \\ 2x_2 + 2x_3 &= -6 \\ x_1 - 2x_2 - 3x_3 &= 10 \end{aligned}$$

This system of equations is determined, since it has three unknown and three equations.

Set up the augmented matrix,

```
>> M=[4 -4 -8 27; 0 2 2 -6;1 -2 -3 10]
```

and find its reduced row echelon form.

```
>> rref(M)
ans =
    1      0      -1      0
    0      1      -1      0
    0      0       0      1
```

The last row of the reduced row echelon form represents the equation $0x_1+0x_2+0x_3 = 1$, which clearly has no solutions. If there is no solution which satisfies this last equation, then there certainly cannot be a solution which satisfies all three equations represented by the matrix. Consequently, the system has no solutions. A system that has solutions is called a *consistent* system. Since this system has no solutions it is *inconsistent*.

Problem #2 Let

$$A = \begin{pmatrix} 1 & 5 & 2 & 6 \\ 2 & 3 & -1 & -6 \\ 0 & -3 & 10 & 12 \\ 4 & 4 & 8 & -8 \end{pmatrix}, \quad b = \begin{pmatrix} 10 \\ -13 \\ 15 \\ 1 \end{pmatrix}$$

Using Matlab, do the following matrix operations:

(i) Set up the matrix multiplications $c = A\,b$ and $d = b^T\,A$. Compare cd with dc, and $A\,A^T$ with $A^T\,A$. What does the comparison illustrate?

(ii) Set up the augmented matrix for the system of equations, $Ax = b$, and compute the reduced row echelon form.

(iii) Compute the solution, if one exists, to the linear system, $Ax = b$.

(iv) Is the system underdetermined, determined or overdetermined? Is the system consistent or inconsistent?

Problem #3 Let

$$\begin{aligned} 4x_1 - 4x_2 - 8x_3 &= 4 \\ 2x_2 + 2x_3 &= 2 \\ x_1 - 2x_2 - 3x_3 &= 0 \end{aligned}$$

(i) Find all solutions to the linear system, if they exist.

(ii) Is the system underdetermined, determined or overdetermined? Is the system consistent or inconsistent?

3 Determinants – A Review

Let's recall some facts about determinants. The determinant is defined for any square matrix, i.e. any matrix which has the same number of rows and columns. For two by two and three by three matrices we have the formulas:

$$det\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

and

$$det\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = aei - afh - bdi + cdh + bfg - ceg.$$

Problem #4 Use the system of equations in Problem #3 to answer the following:

(i) Compute the determinant and inverse of the coefficient matrix by hand. (*Remember:* To compute the inverse, attach the identity matrix onto the right hand side of the coefficient matrix and reduce the left hand side to the identity matrix).

(ii) Now using Matlab, compute the determinant and inverse of the coefficient matrix.

(iii) Find the determinant of the coefficient matrix in Problem #2 using Matlab.

(iv) What is the meaning of a non-zero determinant?

---

4 Matrix Algebra

Problem #5: Complete the following matrix operations, using Matlab. Let

$$A = \begin{pmatrix} 1 & -2 & 1 \\ 2 & 3 & -1 \\ 0 & -3 & 10 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 3 & -1 \\ 0 & -1 & 1 \\ 1 & -2 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 10 \\ 3 \\ -1 \end{pmatrix}$$

(a) Compute the determinant of A and B.

(b) Compute $\det(AB)$ and $\det(BA)$. Are they equal? Is this fact true for any pair of square matrices?

(c) Matlab can actually solve linear systems using a variety of methods, one of which is Gaussian elimination. The construction using that method is "A \ b". Compute the solution to $Ax = b$ using this method. Compare the solution to the method of finding the reduced echelon form of the augmented matrix. Are they the same?

5 Programming Loops

In some calculations it is useful to do things recursively. We may repeat commands by placing them in a loop; the construction is as follows. We choose a loop variable n and then increment n from 1 to N:

```
>> for n=1:N
.... "blah, blah" ....
end
```

"blah, blah" denotes the commands we want to repeat N times, and "end" terminates the loop. The "blah, blah" commands can be just about anything you want. For illustration, let's say we want to multiply the column vector x=[1;2;3] by the matrix

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 4 & 3 & 5 \\ 1 & 8 & 7 \end{pmatrix}$$

divide the result by 10, and then add the vector $y = [1; 0; 1]$, and repeat this operation twelve times. We can typeset this in Matlab as follows:

```
>> clear x; clear y; clear A; clear xi;
>> x=[1;2;3]; y=[1;0;1]; A=[1,2,1;4,3,5;1,8,7];
>> for n=1:12
x=A*x/10+y;
xi(:,n)=x;
end
```

>> Note how "xi" records all the intermediate results (from n=1 to n=12)<<

The above code is an example of a *repeated* "linear transformation". After executing the loop, we can plot the results. Actually, Matlab is smart enough to figure out by itself whether to plot the rows or columns of xi depending on our instructions. For example, to plot the components of the vector x against their index, we could enter "plot(1:3,xi)". Alternatively, we could track how each element changes under the matrix operation by entering "plot(1:12,xi)".

Problem #6 Type and run the above example of a "FOR loop" in Matlab. Perform the plot commands "plot(1:3,xi)" and "plot(1:12,xi)" in two separate figures. Explain what Matlab is plotting for each of the plots.

Problem #7 A mixing protocol for 5 containers of solvent consists of a repetition of the following procedure. The contents of each container are removed and divided up into pre-defined fractions. The various fractions are then poured back into the containers in a pre-assigned way. The idea is to continually mix up the solvents in the five containers, such that the concentration of added chemicals can be maintained at a constant level.

Don't worry if you did not catch all the details of the previous paragraph. Let us formulate the problem mathematically, and hopefully that will clarify things for you. Let p_n denote the row vector describing how much solvent is in each container at the n^{th} repetition of the mixing protocol. The rules for dividing up the contents and then refilling them tell us that $p_n = M\, p_{n-1}$, where

$$M = \begin{pmatrix} 0 & 0 & \frac{2}{7} & \frac{1}{2} & \frac{1}{9} \\ \frac{1}{5} & 0 & \frac{1}{7} & \frac{1}{4} & \frac{1}{9} \\ \frac{2}{5} & \frac{1}{4} & \frac{1}{7} & \frac{1}{4} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{2}{7} & 0 & \frac{1}{3} \\ \frac{2}{5} & \frac{1}{4} & \frac{1}{7} & 0 & \frac{1}{9} \end{pmatrix}$$

The matrix M is given to us by the person in charge of the mixing protocol. Assume that at the beginning all the solvent is inside the first container, that is, $p_1 = [1;0;0;0;0]$. How can we determine the distribution of the solvent after the first application of the mixing protocol? Well, we simply use our math, and compute:

$$p_2 = M\, p_1 \quad where\ \ p_1 = [1;0;0;0;0], \quad \mathrm{r}ight?$$

Using matrix multiplication, we may determine p_2, then p_3, and so on. Now, let's do the following:

(a) Compute the components of p_n against n for the first 15 mixes and draw a graphics showing them. (*Hint:* Use a loop to calculate the progression of mixes, and plot the amount of solvent for each container on a single graph with p_n on the y-axis and n on the x-axis.) What do you observe in the picture?

(b) Compute the eigenvalues of M. Check that one of them is 1, whereas the other four have absolute values that are less than one. The special eigenvalue 1 has an associated column eigenvector that solves the equation $v = Mv$. Indicate how this eigenvector is related to the solution p_n after many mixes, and hence deduce the components of v. (*Hint:* To compute the eigenvalues and eigenvectors of M, use $[V, E] = \text{eig}(M)$, where E will be a diagonal matrix containing the eigenvalues, and the column vectors of V are the corresponding eigenvectors).

(c) Which container has the highest concentration of solvent after many mixes?

(d) Which container is the fullest after many mixes? or, equivalently, which container contains the most solvent?(*i.e.* which component of p_n is the largest for large n?

Quit MATLAB by clicking on the `File` menu in the upper left corner and choosing `Exit`. Please remember to **Log Off** (from the "Start" menu in the lower left of the screen).