

## Examples: Solving Linear Algebra Problems in Matlab

A good way to work through these: Copy and paste them into Matlab's editor as a script file, then see if Matlab produces the correct answers.

### Remember:

"Computers don't do what you want them to do, computers do what you tell them to do. These are often not the same thing."

1. Given the matrix below, (a) Find a basis for the column space, (b) find a basis for the row space (as column vectors), and (c) find a basis for the null space. Finally, (d) show that the row space is orthogonal to the column space.

$$\begin{bmatrix} 1 & 4 & 8 & -3 & -7 \\ -1 & 2 & 7 & 3 & 4 \\ -2 & 2 & 9 & 5 & 5 \\ 3 & 6 & 9 & -5 & -2 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 3 \\ 0 \\ -1 \\ 6 \end{bmatrix}$$

In Matlab, we will find the basis vectors and assign them as columns of a matrix. For example, we'll do the column space by first row reducing to find the pivot columns:

```
A=[3 -1 7 3 9;-2 2 -2 7 5;-5 9 3 3 4;-2 6 6 3 7]
B=rrref(A)           % We see that the 1st, 2d and 4th cols are basic
B1=A(:, [1,2,4]);    %B1 holds the basis for the column space

B2=B( 1:3, :)' ;     %B2 is formed from the first three rows of the rref (then
                    % transposed to make them column vectors. This forms
                    % basis for the row space.

B3=null(A,'r');       %Computes a basis for the null space (the 'r' option is to
                    % make the approximations rational numbers - otherwise,
                    % the basis vectors are orthonormal (hard to read).

% Not necessary, but it is sometimes good to check your dimensions:
size(B1)
size(B2)
size(B3)
% The columns of B2 (the row space) should be orth to cols of B3 (null space):
B2'*B3

2. Example: Let  $U$  be the matrix given below. Normalize the columns, then find project  $\mathbf{x}$  into the column
space spanned by those two vectors (See 6.3, Example 2).

U=[2 -2; 5 1 ; -1 1]; y=[1 2 3]';

%What does U.*U do? What happens when you "sum" a matrix?
colnorms=sqrt(sum(U.*U))

% Find out what repmat does- try repmat(colnorms,3,1) and compare to
% repmat(colnorms,2,3) for example
U=U./repmat(colnorms,3,1)

% Check that the matrix has orthonormal columns:
U'*U

% Find the coordinates of y with respect to the two columns:
coords=U'*y
% Find the projection:
projection=U*coords
% Or do it all at once:
projection=U*U'*y
```