Quick introduction to MATLAB

- MATLAB is available for free on Cougar Apps under Development Tools.
    - Be sure NOT to use the light version of Citrix, as MATLAB will run smoother on the full version.
    - The Symbolic Math Toolbox has now been installed on the MATLAB in Cougar Apps:
        - If you want to be able to work with parameters in MATLAB (say as an entry in a matrix), then they must first be declared as variables as follows:
        - If you want to declare just a single parameter, say t, then you should type **syms t**
        - If you want to declare multiple parameters, say a, b, and c, then you can type **syms a b c**
        - You can also create symbolic matrices. For example, **syms A B C [4 5]** creates three symbolic 4x5 matrices.
- To enter a matrix type the entries row by row in square brackets, with entries separated by blanks, and rows separated by semicolons.
    - Example: Type **A = [2 -3 -1; 2 4 0]** to store this simple 2x3 matrix under the name **A**.
    - All rows must have the same number of entries.
    - MATLAB is case sensitive; for example **C** and **c** are different matrices.
    - MATLAB displays entries that are not integers with 5 digits precision.
        - To display all entries as fractions type **format rat**.
        - To change format back to decimal type **format short** (for 5 digits precision) or **format long** (15 digits precision).
- There are a few standard matrices that are built into MATLAB already:
    - **zeros(m,n)** is the mxn matrix with all entries being 0.
    - **ones(m,n)** is the mxn matrix with all entries being 1.
    - **eye(n)** is the nxn identity matrix.
    - Example: **B = 4*ones(2,3)** stores the 2x3 matrix all of whose entries are 4 into **B**.
- Accessing a matrix:
    - **size(A)** gives the size (# of rows, # of columns) of the matrix A.
    - To access/display a matrix just type its name.
    - To access/display the entry in row i and column j of matrix A type **A(i,j)**
    - To access/display row i type **A(i,:)**
    - To access/display column j type **A(:,j)**
    - To change an entry/row/column of matrix A set the above access command equal to whatever you want to change the entries to:
        - Example: **A(:,2)= [1 3]** changes the entries in the second column of A to 1 and 3.
- Matrix Algebra:
    - To add matrices A and B type **A+B**
    - To subtract matrices A and B type **A-B**
    - To multiply matrix A by the scalar c type **c*A**
    - To multiply matrices A and B type **A*B**
    - The kth power of the matrix A is **A^k**
    - The transpose of the matrix A is **A'**
- Vectors in R^n are just n x 1 matrices, so that all operations for vector arithmetic are the same as those for matrices shown above.
- To find the dot product of vectors u and v type **dot(u,v)** or **u'*v**
- To find the norm of vector u type **norm(u)**. (Keep in mind that this is usually irrational, so it may be better to ask for **norm(u)^2** or simply **u'*u** instead, and then your answer is the square root of that.)
- To find the projection of vector u in the direction of vector a type **(dot(u,a)/norm(a)^2)*a** or **((u'*a)/(a'*a))*a**
- The **quiver** command can be used to draw 2D-vectors. (**quiver3** works similarly for 3D.)
    - **quiver(x,y,u,v,0)** draws an arrow/vector from (x,y) to (x+u,y+v), where the 0 signals to MATLAB not to change the length of the arrow.
    - To draw multiple arrows in the same diagram collect all the corresponding x-and y-values for the starting points into two single vectors X and Y. Similarly collect x- and y-values for the direction of the arrows into

two single vectors U and V. The command then becomes **quiver(X,Y,U,V,0)**.
    - For example **quiver([0,1,0],[0,2,0],[1,3,4],[2,1,3],0)** shows graphically that [1;2]+[3;1]=[4;3].
    - Remark: even though the syntax may strike you as strange, this is quite suitable for graphing vector fields in Calc 3 or physics.
- To find the reduced row echelon form of matrix A type **rref(A)**
- To find the inverse of the n x n matrix A you can:
    - Proceed as on page 175-177, which means you type **rref([A eye(size(A))])** or just **rref([A eye(n)])**
    - Type **inv(A)**
    - Type **A^-1**
    - MATLAB uses different approaches to execute these procedures, but if the matrix is invertible the results should be the same!
- To find the determinant of matrix A type **det(A)**
- To find the characteristic polynomial of matrix A type **poly(A)**
- To find the eigenvalues of matrix A type **roots(poly(A))** or **eig(A)**.
- To find the eigenvectors and eigenvalues of matrix A type **[V,D]=eig(A)**, where the columns of V are the eigenvectors, and the corresponding eigenvalues are on the diagonal of D. However, the eigenvectors you get in this way may not be of a particularly simple form.
- Sometimes a better way of finding the eigenvectors of the n x n matrix A for eigenvalue k is to to solve (A-kI_n)=0 in Matlab by using **rref(A-k*eye(n))**.