Matlab Introduction

The following are a few things to get you going in Matlab. One general thing you will quickly notice is that Matlab generally requires you to think about what you are doing in terms of matrices and vectors. The suggestions below are not intended to fully describe everything to know about Matlab (not even close). It does, however, point out a few things that you will be running into later on.

1. Row vectors are defined as:

   `r1=[1 2 3 4]`

   `r2=[-10:1:10]` is a handy way to quickly generate a list of numbers in row vector form. Try `r2=[-10:1:10];` instead if you want to avoid the output from appearing.

2. Column vectors can be defined in different ways as well:

   `c1=[1;2;3;4]`

   `c2=[-10:1:10]'` generates a column vector (note the prime).

3. Define a three dimensional row vector `x=[1 2 1]` and then define a matrix `A=[1 2 3; 4 5 6; 7 8 9]`.

4. The conjugate transpose of a matrix A is obtained by the command `A'`. If the matrix A is real, then this is just the transpose. If the matrix is complex but the non-conjugate transpose is desired, then the appropriate command is `A.'` which uses "dot-prime".

5. Understand the very important difference between the usual matrix multiplication and component-wise multiplication (as well as division and powers)

   `A*A` is "regular" matrix multiplication. Do this with Matlab and make sure that this is what you expected.

   `A.*A` is component-wise multiplication (the usual operator '*' has a '.' in front '.*'). Do this with Matlab and make sure that you see what's going on. If you use one when you meant to use the other you will not get the answer you expect. If you use one when you meant to use the other on a problem where you don't know the answer, then this will be a major problem!

   Note that `A*x` is not a well-defined operation but `x*A` is and so is `A*x'` where `x'` changes the row vector x into a column vector.

6. Define a row vector `t=[0:0.1:5.0];` and use this to plot some function of t, say sin(t), by using the command `plot(t,sin(t))`. Note that the values of t must be already defined before the plot command is used. If you want a new plot try `plot(t,cos(t))`. If you would like more than one plot to appear in the same figure you can enter the command `hold on` at the matlab prompt and the same figure will be used until you enter the command `hold off`. Alternatively, the command `plot(t,sin(t),'r',t,cos(t),'g')` will plot $\sin t$ with a red line and $\cos t$ with a green line on the same graph. Also try `comet(t,cos(t))`.

7. If A is an $n \times n$ matrix and b is a column vector of length n, then the command `x=A\b` (note that this is \ and not `/`) computes the solution to $Ax = b$ using Gaussian elimination (warning messages will be returned for singular, or nearly singular matrices).

8. If A is an $m \times n$ matrix with $m > n$ and $Ax = b$ is the corresponding overdetermined system, then the command `x=A\b` solves this problem in the least squares sense. That is, the returned x minimizes $\|Ax - b\|^2$.

9. The Matlab command `[L,U]=lu(A)` computes the LU factorization of a square matrix A. If row exchanges are necessary, the matrix L will not be lower triangular. In this instance, the command `[L,U,P]=lu(A)` can be used to return the lower and upper triangular matrices L, U and the permutation matrix P where $PA = LU$.

10. The Matlab command for the Cholesky factorization `R=chol(A)` returns an upper triangular matrix R so that $R^T R = A$ when A is symmetric positive definite (similar to $A = LDL^T$ factorization, where $R = D^{1/2} L^T$). If A is not positive definite an error message is given, indicating that a non-positive pivot has been encountered.

11. The Matlab command `[V,D]=eig(A)` returns the eigenvectors in the columns of the matrix V and the eigenvalues in the diagonal elements of the matrix D. The command `d=eig(A)` returns the eigenvalues of the matrix A in the column vector d.

12. The Singular Value Decomposition of a Matrix A is calculated in Matlab using the `svd` command. `[U,S,V]=svd(A)` returns unitary matrices U and V and a diagonal matrix S containing the singular values of A. Here $A = USV^*$. If A is $m \times n$ then U is $m \times m$, V is $n \times n$ and S is $m \times n$. The command `s=svd(A)` returns just the singular values of A in the vector s.