MatLab Commands

December 16, 2021 2:03 AM

- To declare the matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ use $\mathtt{A} = [\mathtt{1\ 2\ 3;4\ 5\ 6;7\ 8\ 9}]$. Note that MATLAB does not distinguish between augmented and non-augmented matrices, so it will be up to you to remember how to interpret the columns of the matrix.
- To calculate the reduced row echelon form of a matrix A, use `rref(A)`.
- If you have two matrices A and B with the same number of rows, you can use the command `[A B]` to create a new matrix with the columns of A followed by the columns of B. This is particularly useful when A is the coefficient matrix of a system and B is the single column of constants appearing on the other side of the equality sign.
- The command `A\b` attempts to produce a solution to the system $[A|b]$, but its behaviour can be surprising if the system is inconsistent or has more than one solution. In this course you will probably be best to use `rref([A b])` instead, and then interpret the result yourself.

- Declaring a vector in MATLAB is the same process as declaring a matrix, but with only one column. For instance, the vector $\vec{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ can be entered in MATLAB as `v = [1;2;3]`.
- Addition of vectors of the same size is accomplished in MATLAB by putting a $+$ between the vectors; for instance, if `v` and `w` are vectors of the same size then `v + w` produces their sum.
- To multiply a vector by a scalar, use the operator $*$. For instance, to compute $2\vec{v}$, enter `2 * v`.

- If `v` and `w` are vectors, their dot product is calculated by `dot(v, w)`.
- To calculate the length of the vector `v`, use either $\sqrt{\mathtt{dot(v,v)}}$ or `norm(v)`.

- If `A` is a matrix then `rank(A)` calculates the rank of A.

- If `A` and `B` are matrices of the same size, then to add them use `A + B`.
- If `A` is a matrix then $2A$ is computed by `2 * A`.
- If `A` is a matrix then A^t is computed by `A'`.
- If `A` and `B` are matrices of appropriate sizes so that AB is defined, it is computed by `A * B`.

- If `A` is a matrix then the transpose of `A` can be calculated by using `transpose(A)`, or by using `A.'`
- If `A` is a square matrix, you can find the inverse by using `inv(A)`. Be careful, though! Sometimes MATLAB will still give you an answer even if A is not invertible.

- If `A` is a square matrix then the determinant is calculated by `det(A)`.

- If `A` is a square matrix then the command `eig(A)` returns a list of the eigenvalues of A, with each one repeated according to its algebraic multiplicity.
- If `A` is a square matrix and you run the command `[P, D] = eig(A)` then D will be a diagonal matrix whose entries are the eigenvalues of A (repeated according to algebraic multiplicity) and P will be a matrix whose columns are eigenvectors ordered so that the jth column of P is an eigenvector for the eigenvalue appearing in position (j, j) of D. If it is possible, P will be made to be invertible.