▷ Project 1. Implement Gram-Schmidt as a Matlab Function

In more detail, create a Matlab m-file GramSchmidt.m in which you define a Matlab function GramSchmidt(M) taking as input a rectangular matrix M of real numbers of arbitrary size $m \times n$, and assuming that the m rows of M are linearly independent, it should transform M into another $m \times n$ matrix in which the rows are orthonormal, and moreover such that the subspace spanned by the first k rows of the output matrix is the same as the space spanned by the first k rows of the input matrix. Clearly, in writting your algorithm, you will need to know the number of rows, m and the number of columns n of M. You can find these out using the Matlab size function. In fact, size(M) returns (m,n) while size(M,1) returns m and size(M,2) returns n. Your algorithm will have to do some sort of loop, iterating over each row in order. Be sure to test your function on a number of different matrices of various sizes. What happens to your function if you give it as input a matrix with linearly dependent rows. (Ideally it should report this fact and not just return garbage!)

Addenda

- The project is a little ambiguous. It asks for a function M-File that **transforms** the input matrix to a matrix with orthogonal rows. One legitimate interpretation (actually the one I had in mind) was that the input matrix should actually be changed by the function into one with orthogonal rows. However, if you prefer to write your function so that it does not actually change the input matrix but it instead returns a different matrix having orthogonal rows, that is acceptable. (But you should realize that there are often good reasons to do the orthogonalization "in place". For example, if the input matrix is very large (e.g., 10000×10000) then there might be a memory problem in creating a second matrix of that size.)
- Please put your name in a comment near the top of the M-File.
- Another comment (coming just after the line declaring the function) should explain in detail just what the function does and how to call it. This will be printed in the command window when a user types "help GramSchmidt".
- Several of you have asked how to handle the problem of an input matrix in which the rows are not linearly independent. You will detect this in the course of the computation by finding that the norm of a certain vector u is zero, so you cannot normalize u to get the next element of the orthonormal set. (By the way, you should be careful to check not just for $\|u\|$ being exactly zero, but say for $\|u\| < 0.00001$. The reason is that because of roundoff and other errors the computation will be too unreliable if the vectors are "almost linearly dependent".) In this case you should not try to return any matrix and should just use the disp() function to display an error string to tell the user about the problem. Something like:
  ```
  disp('Input matrix rows are dependent; orthonormalization impossible.')
  ```