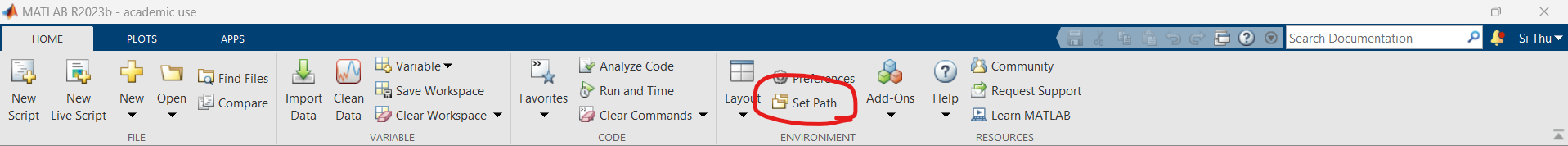
MATLAB for Linear Algebra (MA1508E)

**DO NOT USE THE COMMAND “clear”. RESTART PROGRAM IF USED!**

For setup **(For STRANGERS):**

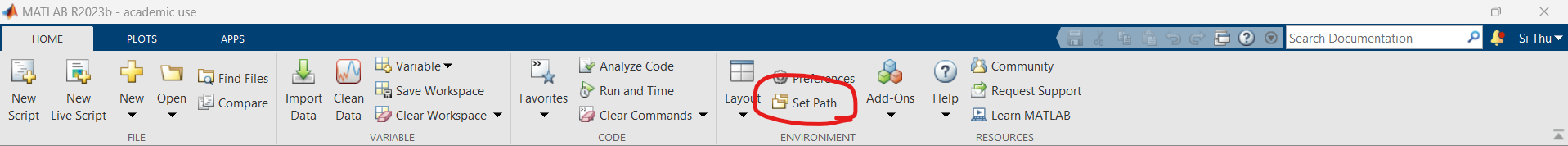
1. Set the path to inside the MA1508E codes folder



1. In MATLAB, type **“ MA1508E”**
2. **To call functions: example 🡪 obj.rref(A) / obj.performERO / obj.leftInverse(A)**

For setup **(For ADMINS)**:

1. Set the path to inside the MA1508E codes folder



1. In MATLAB, type **“ m = MA1508E”**
2. **Password : SECRET**
3. **To call functions: example 🡪 m.rref(A) / m.performERO / m.leftInverse(A)**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Function** | **Parameters** | **Description** |
| 1 | rref(A) | A – matrix to be performed on | Returns the rref of the matrix and the corresponding steps to derive the rref |
| 2 | performERO | No parameters | Performs ERO on matrices in a more interactive way  **\*\*Note:**  **DO NOT use i or j as symbols. These values are taken for their imaginary values. If you really need to use them, capitalize them.** |
| 3 | leftInverse(A) | A – matrix to be performed on | Checks if A has a left inverse. Returns left inverse if it exists, else returns false. |
| 4 | rightInverse(A) | A – matrix to be performed on | Checks if A has a right inverse. Returns right inverse if it exists, else returns false. |
| 5 | checkSpanSubset(A,B) | A – Set that is checked against  B – Set to be checked | Checks if the span of B is a subset of the span of A. Returns the initial augmented matrix, final matrix and the conclusion.  **🡪 This can be used to check if vectors are also included in the span of A**  **Steps:**   * **Put vectors in B** * **Look at the rref of augmented matrix** * **Look for consistent or inconsistent solutions** |
| 6 | checkSpanEquality(A,B) | A,B – Sets of vectors | Checks if the span of B is equal to the span of A. Returns initial and final matrices as well as the conclusion |
| 7 | findHomogeneousSolution(A) | A – matrix representing the set of vectors to be performed on | Returns the rref of the matrix as well as the homogeneous solution. Each column of the output represents 1 parameter.  **\* Can be used to find basis**  **\*There are times where using column space will be better.** |
| 8 | checkLinearIndependence(A) | A – matrix representing the set of vectors to be performed on | Returns true if the set of vectors is linearly independent. Return false otherwise. |
| 9 | vectorRelativeToBasis(S,V) | S – matrix representing the basis  V – Vector that you want to check | Returns the vector relative to the basis S. (V)s |
| 10 | reverseRelativeToBasis(S,Vs) | S – matrix representing basis  Vs – Vector relative to S | Returns the initial vector V, given Vs |
| 11 | checkBasisForSubspace(S,V) | S – matrix representing the basis  V – matrix representing the subspace | Returns the Boolean of whether S is a basis for V. If it is, return by which definition it was proven.  **\* Can also be used to find out if 2 bases are both bases for the same subspace. (just put the 2 bases as S and V)** |
| 12 | findColumnSpaceBasis(S) | S – matrix representing the set of vectors | Returns the column space of S |
| 13 | findRowSpaceBasis(S) | S – matrix representing the set of vectors | Returns the row space of S |
| 14 | extendBasisToHigherSpace(S,n) | S – matrix representing the current basis  n - R^n | Returns the vectors to be added to extend the basis to span higher n values.  **🡪 can check if S spans Rn** |
| 15 | isOrthogonalSet(S) | S – set of vectors | Returns whether the set is orthogonal |
| 16 | isOrthonormalSet(S) | S – set of vectors | Returns whether the set is orthonormal |
| 17 | isOrthogonalTo(S, V) | S – Set of vectors to check against  V – Vector to be checked | Returns whether V is orthogonal to S |
| 18 | toOrthonormalSet(S) | S – An orthogonal set | Normalizes the set of vectors in S |
| 19 | orthogonalProj(S, V) | S – The set of vectors to be projected on  V – vector to project | Returns the projection of V onto column space of S. |
| 20 | decomposeVector(S, V) | S – set of vectors (subspace)  V – vector to decompose | Returns the vector V in Vp and Vn |
| 21 | gramSchmidtOrthogonalization(V) | V – vectors to be orthogonalized and normalized | Return all steps used to derive the final orthoganalized set of vectors |
| 22 | factorizeOnSetOfVectors(V) | V – An orthonormal set of vectors to be factorized into QR (same as Tutorial 9) | Returns the factorized form of V. |
| 23 | findLeastSquareSolution(A, b) | A – A matrix whoose columns form the set of vectors  b – a vector for which the least square solution is sought | Returns the least square solution as well as the general solution if it exists. |
| 24 | **eig(A)**  **🡪 USED MAINLY FOR MCQ WHERE NO WORKINGS ARE REQUIRED**    🡪 Can also call it as  [P,D] = m.eig(A)  🡪 Can also use it to find out whether multiple values are eigenvalues  m.eig(A,1,2,3….)  **🡪 Can also check whether A is diagonalizable**  **🡪 Check the section on “combination of spanning sets” and if it have same number of colums as D, A is diagonalizable** | A – A square matrix to operate on | Returns:   * Whether eigenvalue is associated * PDP-1 * Homogeneous system used (if relevant) * Rref (if relevant) * Identifies whether A is orthogonally diagonalizable **(OD)**   **If A is OD (execute manually):**   * Find the spanning sets of all eigenvalues. **Type: m.eig(A,v1,v2,v3,…..)** * **Alternatively, you can also look for the Processed Eigenvectors (P).** * Execute GSP to get P. * D remains unchanged |
| 25 | **eigComputationWithWorkings(A)** | A – A square matrix to operate on | Returns:   * Characteristic equation * the steps taken to get the P and D values. * A check on whether A is orthogonally diagonalizable and which vectors to use GSP * A check on whether there are repeated eigenvectors. If there are, it will tell you to go and compute it manually. No P and D values given |
| 26 | eigenvectorODESolutions(A) | A – constant coefficient matrix (y’ = Ay) | Returns all eigenvalues with their eigenvectors and solutions  **🡪 This can be used to find the fundemental set of solutions**  **🡪 Limitation: only prints 1 solution for a repeated eigenvalue** |
| 27 | computeWronskian(S) | S – set of vectors to be computed | Returns the Wronskian value and whether the set of vectors are linearly independent. |
| 28 | findGeneralizedEigenvector(A) | A – matrix to be computed | Returns the generalized eigenvector and steps used to obtain it |
| 29 | findSubspaceIntersection(A,B) | A,B – Matrix of basis vectors forming 2 subspaces | Returns the intersection between 2 subspaces with the steps and explanation |