ENGR 1181  |  MATLAB 2: Array Creation

Preparation Material

Learning Objectives

1. Demonstrate proper convention for assigning arrays to a variable (e.g., space, semi-colons, colon operators, linspace function)
2. Apply the transpose operator correctly

Topics

Students will read Chapter 2.1 – 2.4 of the MATLAB book before coming to class. This preparation material is provided to supplement this reading.

Students will learn basic understanding of arrays and how to create them (vectors and matrices) in MATLAB. This material contains the following:

1. Key Definitions & Ideas
2. Vectors in MATLAB
3. Matrix Definition and Creation
4. Transpose
5. Array Examples

1. Key Definitions & Ideas

**Scalar** – Single value, e.g. 3 or 45 or 35.07. A scalar variable can be identified within one column and one vector of a set of data.

**Vector** – String of values (or a string of scalars)

   **Row vector** – one row and multiple columns

   **Column vector** – multiple rows and one column

**Matrix** – multiple rows and multiple columns (group of vectors)

**Array** - The fundamental type of vector form MATLAB uses to store data.

In MATLAB, a vector can be created by typing the elements (numbers) inside square brackets [], separating them by a comma or space(s).
Arrays are used in many applications.

- Arrays of numbers can represent a vector.
- An array can be a position vector. The location of point $P$ in a three dimensional space can be represented by the three Cartesian coordinates 2, 4, and 5.
- Arrays can represent data.

The following are a graphical representation of a position vector, and an example of data arrays.

![Graphical representation of a position vector](image)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>127</td>
<td>130</td>
<td>136</td>
<td>145</td>
<td>158</td>
<td>178</td>
<td>211</td>
</tr>
</tbody>
</table>

### 2. Vectors in MATLAB

The year and population data can be entered as vectors in **rows**:

- Population = [127 130 136 145 158 178 211]

...or it can be entered as vectors in **columns**:
Column vectors can be created two ways: with a semi-colon or with the 'Enter' key.
Creating equal step spacing in a vector:

A vector in which the first term is $m$, the spacing is $q$ and the last term is $n$ can be created by typing $[m : q : n]$. This can be described as $[\text{start} : \text{step} : \text{end}]$.

If spacing ($q$) is omitted, then it is simply $[m : n]$ and the default step is 1. This can be described as $[\text{start} : \text{end}]$.

Alternatively, the command linspace() can be used to properly space the variables in a vector. This is useful when the spacing is difficult to identify, but the number of needed data points is known. This is written as linspace( $m$, $n$, $r$ ), where the first term is $m$, the last term is $n$, and the number of data points is $r$.

Let's say someone knows the coordinates on a grid and wants to evenly space 7 items from location 2.5 to 14.5. He/she may not know the exact spacing, especially if design plans change the number of poles is reduced to 6 or 8. To save time and errors, the designer can use linspace() the following way:
Note that calculating \((14.5 - 2.5)/7 = 1.714286\) doesn’t work properly because the end points are included in \texttt{linspace}(). The proper math would actually be \((14.5 - 2.5)/6\) or referenced as \((n - m) / (r - 1)\).

### 3. Matrix Definition and Creation

A \((m \times n)\), or “\(m\) by \(n\)”, matrix has \(m\) rows and \(n\) columns. The number of rows and columns may be the same or different. \((m \times n)\) is called the **size** of the matrix.

The following matrix \(G\) has two rows and four columns \((2 \times 4)\), or “2 by 4”

\[
\begin{array}{cccc}
31 & 26 & 14 & 18 \\
3 & 51 & 20 & 11
\end{array}
\]

A matrix is created by typing the elements (numbers) row by row inside square brackets \([\ ]\). Similar to a column vector, the multiple rows in a matrix can be created with a semicolon or the ‘Enter’ key.

\[
>> a = [5 \ 35 \ 43; \ 4 \ 76 \ 81; \ 21 \ 32 \ 40]
\]

\[
a = \\
5 \ 35 \ 43 \\
4 \ 76 \ 81 \\
21 \ 32 \ 40
\]
**4. Transpose**

The transpose operation is executed by adding an apostrophe (') after the vector or matrix:

For a vector: Converts a row vector to a column vector, or converts a column vector to a row vector.

For a matrix: Interchanges the rows and columns. For example, a 2 x 4 matrix becomes a 4 x 2 matrix.
ENGR 1181 MATLAB: Array Creation

Preparation Material

```
>> c = [1 2 3 4; 5 5 5 5; 6 7 8 9]
c =
   1     2     3     4
   5     5     5     5
   6     7     8     9

>> d = c'
d =
   1     5     6
   2     5     7
   3     5     8
   4     5     9
```
5. Array Examples

```matlab
>> a = 7
a =
    7

>> E = 3
E =
    3

>> d = [5 a+E 4 E^2]
d =
    5     10     4     9

>> g = [a a^2 13; a*E 1 a^E]
g =
    7     49    13
    21     1   343

>> V = [a a^2 13; a*E 1 a^E]'
V =
    7     21
    49     1
   13    343
```