Arrays in MATLAB

ChEn 1703

See also the wiki page for tutorials
What are Arrays?

Array: an \( n \)-dimensional collection of numbers

On paper:
\[
b = \begin{pmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_m
\end{pmatrix}
\]
\[
A = \begin{bmatrix}
  A_{11} & A_{12} & \cdots & A_{1n} \\
  A_{21} & A_{22} & \cdots & A_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  A_{m1} & A_{m2} & \cdots & A_{mn}
\end{bmatrix}
\]

Graphical representation:

Address in Matlab:
\[
b(i) \quad A(i,j) \quad C(i,j,k)
\]
Creating Arrays

**Direct assignment** - used mainly for small arrays

- **Vectors (1D Arrays):**
  - `b=[b1 b2 b3...bn];` % row-vector
  - `b=[b1; b2; b3; ... bn];` % column vector
  - `b = [b1 b2 b3 ... bn]’;` % column vector

- **Matrices (2D Arrays):**
  - `A = [a11 a12 ... a1n; a21 a22 ... a2n; ... ; am1 am2 ... amn];`
  - creates an `m x n` array (m rows, n columns)

**Shortcuts**

- **Using “:” to make arrays**
  - `a = 0:5:20;` ⇒ `0 5 10 15 20`

- **ones(m,n); zeros(m,n); eye(m,n); rand(m,n);**
  - creates arrays with “m” rows and “n” columns

- **linspace( start, end, nentries );**
  - A linearly spaced vector with “nentries” points between “start” and “end”

- “help elmat” for much more information...
Array Manipulation

Transpose: $\mathbf{A}^T$.
- Vector: row → column
- Matrix: Exchange rows & columns
- In Matlab, use the apostrophe: $\mathbf{A}'$

Dimension information for an array
- $n = \text{length}(\mathbf{a})$
- $[\text{rows}, \text{cols}] = \text{size}(\mathbf{A})$

Array addressing (indexing)
- $\mathbf{A}(i,j) \Rightarrow \text{ith row and jth column of } \mathbf{A}$.
  - Extendable to higher dimensionality
  - Also applies to vectors: $b(i) \Rightarrow \text{ith element of } b$.

```
A = [1 2 3; 4 5 6];
Atrans = A';
```

```
a = [1 5 10 2 3];
n = \text{length}(a);
[nr, nc] = \text{size}(a);
```

```
A = [1 2; 3 4; 5 6];
[nr, nc] = \text{size}(A);
```

```
A = [1 2; 3 4; 5 6];
A(2,1) = 2.5;
A(3,2) = A(1,1);
```
The “Magic” Colon

Creating regularly spaced vectors:

\[ a = 1: -1: -10; \]  
% create a row vector starting at 1 and 
% decreasing in increments of -1 to -10.

\[ b = 1: 5; \]  
% create a row vector \([1 2 3 4 5]\)

\[ c = 3: 2: 6; \]  
% create a row vector \([3 5]\)

Wild card selector:

\[ A = \text{rand}(3); \]  
% create a random matrix (3x3)

\[ A(2,:) = [1 2 3]; \]  
% fill the second row with 1 2 3.

\[ A(:,1) = [2 2 2]'; \]  
% fill the first column with 2s.

\[ A(3,:) = \text{ones}(3,1); \]  
% fill the third row with ones.

\[ A(2:3,1:2) = 1; \]  
% what would this do?
Working with Arrays

Addition & subtraction (a+b    a-b)
• What are the restrictions on the sizes of a & b?
• Example: velocity components in a vector.
  ‣ v1=[2,2];  v2=[5,1];   v3=v1+v2;

Multiplication & addition
• **Elemental multiplication**  C=A.*B;
  ‣ Elements of A are multiplied by corresponding elements of B.
  ‣ C_{ij} = A_{ij} \times B_{ij}.
  ‣ Size restrictions?
• **Matrix multiplication**:  C=A*B;
  ‣ Size restrictions?
• **Elemental division**:  C=A./B;
  ‣ C_{ij} = A_{ij} / B_{ij}.

**Elemental exponentiation**:  C=A.^B;  C=A.^2;

Other elemental operations:
• exp(A); log(A); cos(A); ...
Create a MATLAB code to convert from degrees to radians for user-specified range of angles. Also show the sin and cos of the angles.

\[ \theta^\text{rad} = \theta^\circ \frac{\pi}{180} \]

<table>
<thead>
<tr>
<th>degrees</th>
<th>radians</th>
<th>cos</th>
<th>sin</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0000</td>
<td>0.1745</td>
<td>0.9848</td>
<td>0.1736</td>
</tr>
<tr>
<td>20.0000</td>
<td>0.3491</td>
<td>0.9397</td>
<td>0.3420</td>
</tr>
<tr>
<td>30.0000</td>
<td>0.5236</td>
<td>0.8660</td>
<td>0.5000</td>
</tr>
<tr>
<td>40.0000</td>
<td>0.6981</td>
<td>0.7660</td>
<td>0.6428</td>
</tr>
<tr>
<td>50.0000</td>
<td>0.8727</td>
<td>0.6428</td>
<td>0.7660</td>
</tr>
<tr>
<td>60.0000</td>
<td>1.0472</td>
<td>0.5000</td>
<td>0.8660</td>
</tr>
<tr>
<td>70.0000</td>
<td>1.2217</td>
<td>0.3420</td>
<td>0.9397</td>
</tr>
<tr>
<td>80.0000</td>
<td>1.3963</td>
<td>0.1736</td>
<td>0.9848</td>
</tr>
<tr>
<td>90.0000</td>
<td>1.5708</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>100.0000</td>
<td>1.7453</td>
<td>-0.1736</td>
<td>0.9848</td>
</tr>
</tbody>
</table>
# Array Operations

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>Determine how many elements are in a vector.</td>
</tr>
<tr>
<td>size</td>
<td>Determine the number of rows &amp; columns</td>
</tr>
<tr>
<td>linspace</td>
<td>Build a vector containing equally spaced entries</td>
</tr>
<tr>
<td>zeros</td>
<td>Build an array with given # of rows &amp; columns filled with zeros.</td>
</tr>
<tr>
<td>ones</td>
<td>Build an array with the specified number of rows/columns filled with ones.</td>
</tr>
<tr>
<td>max</td>
<td>Determine the maximum value (for a vector). For a matrix, returns a vector containing the maximum value of each column in the matrix.</td>
</tr>
<tr>
<td>min</td>
<td>Analogous to max.</td>
</tr>
<tr>
<td>sum</td>
<td>Calculate the sum of all elements in a vector. For matrices, return a row-vector containing the sum of each column.</td>
</tr>
<tr>
<td>sort</td>
<td>Sort a vector in ascending order. For matrices, sort each column in ascending order.</td>
</tr>
</tbody>
</table>

For More Information:
- Table 2.1-1 in text.
- Matlab’s `help` on each command.
- Class [wiki page](#).