Arrays in MATLAB

ChEn 1703

See also the wiki page for tutorials

What are Arrays?

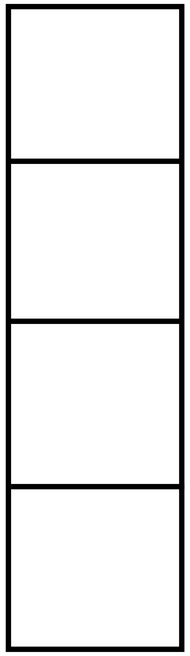
Array: an n-dimensional collection of numbers

On paper:

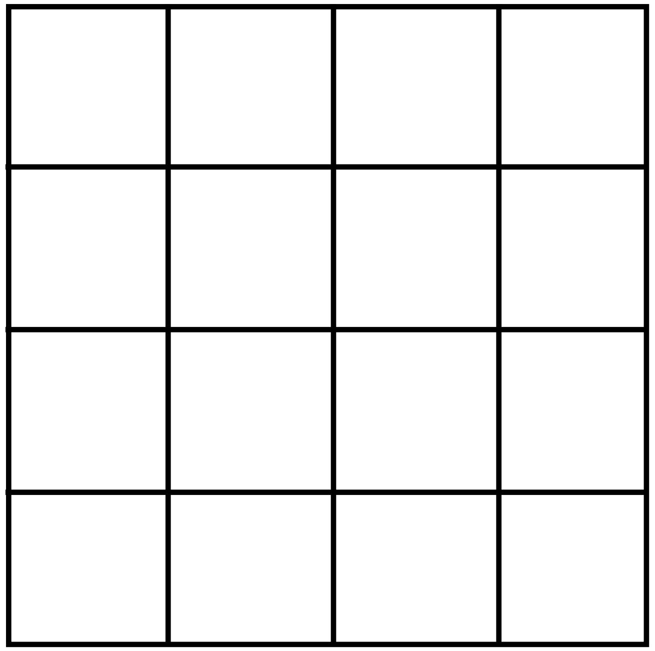
$$b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \qquad A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

Graphical representation:

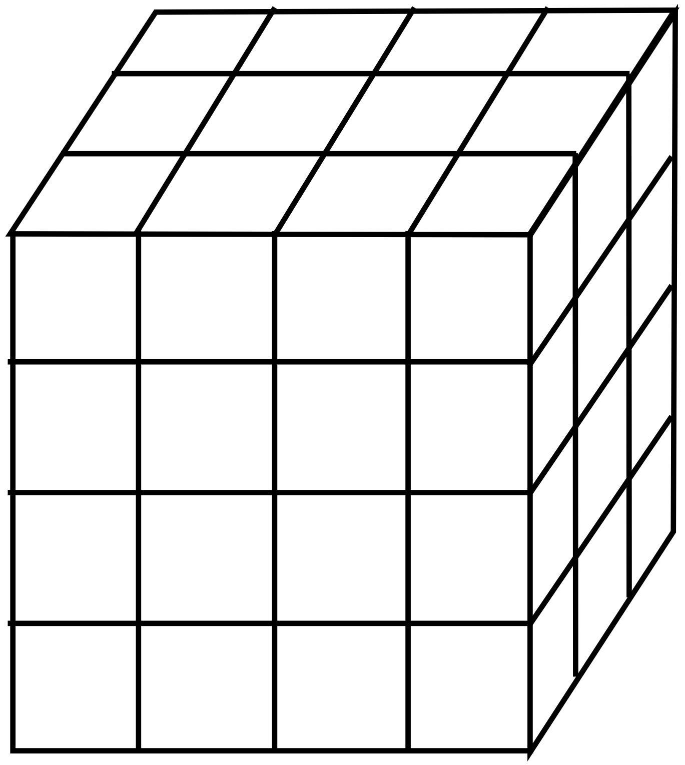
1-D



2-D



3-D



Address in Matlab:

b(i) A(i,j) 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 `mxn` 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: A^T.

- Vector: row→column
- Matrix: Exchange rows & columns
- In Matlab, use the apostrophe: `A′`

Transpose - interchange rows & columns

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

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

Dimension information for an array

- `n=length(a);`
- `[rows,cols]=size(A);`

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

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

Array addressing (indexing)

- `A(i,j)` ⇒ ith row and jth column of A.
  - Extendable to higher dimensionality
  - Also applies to vectors: `b(i)` ⇒ ith element of b.

```
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:

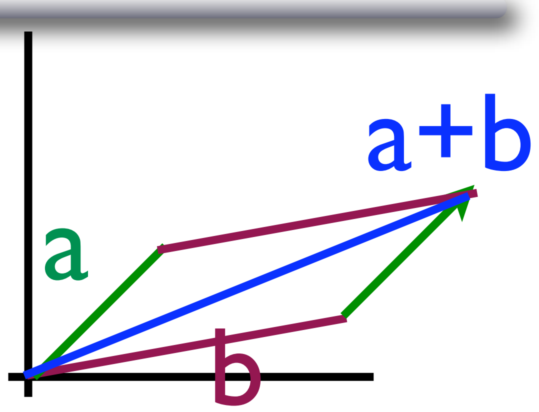
```
a=1:-1:-10;    % create a row vector starting at 1 and
               % decreasing in increments of -1 to -10.
b = 1:5;       % create a row vector [1 2 3 4 5]
c = 3:2:6;     % create a row vector [3 5]
```

Wild card selector:

```
A = 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,:) = ones(3,1);    % fill the third row with ones.

A(2:3,1:2) = 1;        % what would this do?
```

Working with Arrays

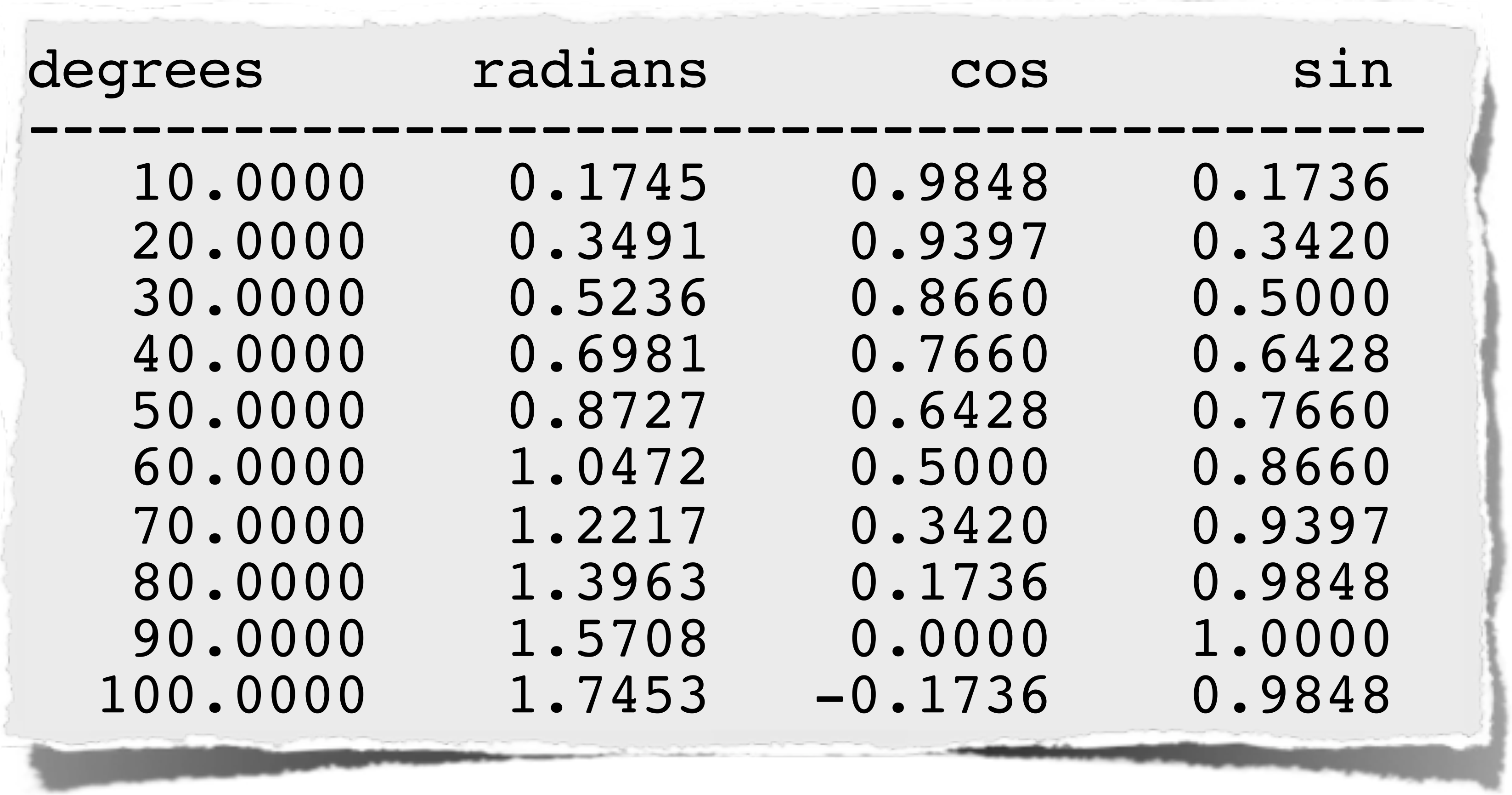


- Addition & subtraction (a+b   a-b)
  - What are the restrictions on the sizes of a & b?
  - Example: velocity components in a vector.
    - `v1=[2,2];   v2=[5,1];    v3=v1+v2;`
- Multiplication & addition
  - Elemental multiplication `C=A.*B;`
    - Elements of A are multiplied by corresponding elements of B.
    - $C_{ij} = A_{ij} * B_{ij}$.
    - 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);` ...

Example: Angle Conversion

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.

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



```
degrees        radians           cos           sin
-----------------------------------------------------
   10.0000      0.1745        0.9848        0.1736
   20.0000      0.3491        0.9397        0.3420
   30.0000      0.5236        0.8660        0.5000
   40.0000      0.6981        0.7660        0.6428
   50.0000      0.8727        0.6428        0.7660
   60.0000      1.0472        0.5000        0.8660
   70.0000      1.2217        0.3420        0.9397
   80.0000      1.3963        0.1736        0.9848
   90.0000      1.5708        0.0000        1.0000
  100.0000      1.7453       -0.1736        0.9848
```

Array Operations

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

For More Information:

- Table 2.1-1 in text.
- Matlab's `help` on each command.
- Class wiki page.