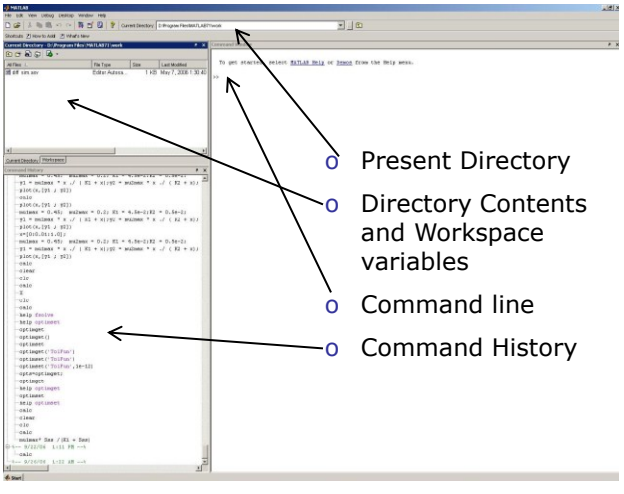
Computational Methods
CMSC/AMSC 460

Ramani Duraiswami,
Dept. of Computer Science

Matlab Windows

- Command line Interface (Main Window)
- Editor Window



Matrices in Matlab

- Entering a Matrix:

```
>> A = [ 0 -0.8 -0.6 ; 0.8 -0.36 0.48 ; 0.6 0.48 -0.64]
A =

         0   -0.8000   -0.6000
    0.8000   -0.3600    0.4800
    0.6000    0.4800   -0.6400
```

- Matrix referencing:

```
>> A(1,2)
ans =

   -0.8000

>> A(2,:)
ans =

    0.8000   -0.3600    0.4800
```

```
>> A(:,1)

ans =

         0
    0.8000
    0.6000
```

- Matrix Operations:

```
>> A+A;
>> A.*A;
>> 3*A;
>> A*A

        ans =

   -1.0000         0         0
         0   -0.2800   -0.9600
         0   -0.9600    0.2800
```

Built-in functions

- Determinant

```
>> det(A)
ans =

   -1.000
```

- Rank

```
>> rank(A)
ans =

    3
```

- Inverting a Matrix

```
>> inv(A)

ans =

   -0.0000    0.8000    0.6000
   -0.8000   -0.3600    0.4800
   -0.6000    0.4800   -0.6400
```

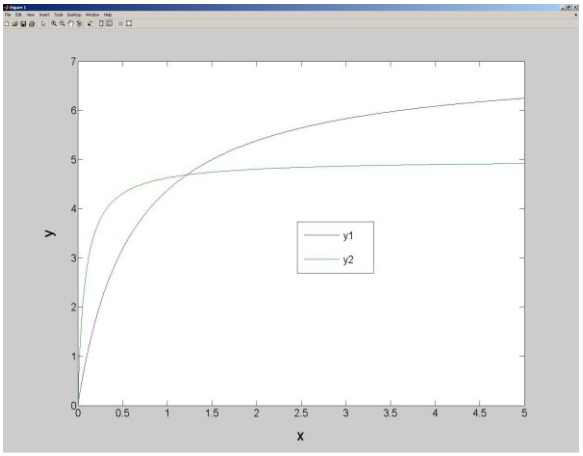
- Transpose of a Matrix

```
>> A'

ans =

         0    0.8000    0.6000
   -0.8000   -0.3600    0.4800
   -0.6000    0.4800   -0.6400
```

Plotting a function

$$y_1 = \frac{7x}{0.6 + x} \qquad y_2 = \frac{5x}{0.08 + x}$$

```
>> x = [ 0:0.01:5];
>> y1 = 7 * x ./ ( 0.6 + x );
>> y2 = 5 * x ./ ( 0.08 + x );
>> plot(x,y1,x,y2)
>> legend('y1','y2')
```



Introduction to MATLAB

- Vectors, Matrices, Syntax
- Vector operations, including the \dot commands
  - length, size, linspace, logspace, size, rand, randn, randperm
- Special vectors and matrices: zeros, ones, eye, magic
- Scripts and functions
  - Diary
- Graphing:
  - plot, special fonts, plot3, semilogx, semilogy, title, xlabel, ylabel, axis, grid, legend, subplot,
- Formatted output:
  - Sprintf, ;, disp, input
- Programming:
  - for, if, while, &, |, ~
- General/misc commands
  - ginput set, size, max, sum, close, figure, hist, any, all , floor, fix, round,
- Graphical programming and callbacks

Matrix-vector product

- Matrix-vector multiplication

$$\mathbf{M} \bullet \mathbf{v} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13} \\ \mathbf{M}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23} \\ \mathbf{M}_{31} & \mathbf{M}_{32} & \mathbf{M}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix}$$

- Recall how to do matrix multiplication
- How many operations does this matrix vector product take?
- How many operations does a general matrix vector product take?

Ways to implement a matrix vector product

- Access matrix
  - Element-by-element along rows
  - Element-by-element along columns
  - As column vectors
  - As row vectors
- Discuss advantages

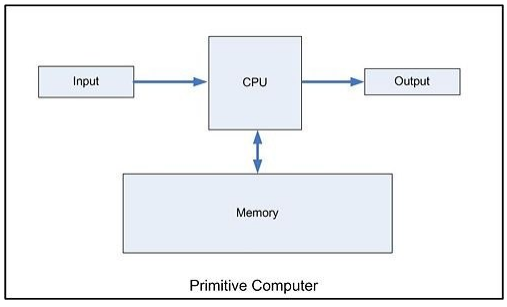
```
[m,n]=size(A);
y = zeros(m,1);
for i=1:m,
   for j=1:n,
        y(i) = y(i) + A(i,j)*x(j);
   end
end
```

```
[m,n]=size(A);
y = zeros(m,1);
for i=1:m,
     y(i) = A(i,:) * x;
end
```

```
[m,n]=size(A);
y = zeros(m,1);
for j=1:n,
    y = y + A(:,j)*x(j);
end
```

Good workmen know their tools

- Primitive model



Memory Hierarchy