Lecture 2

Matrix manipulations

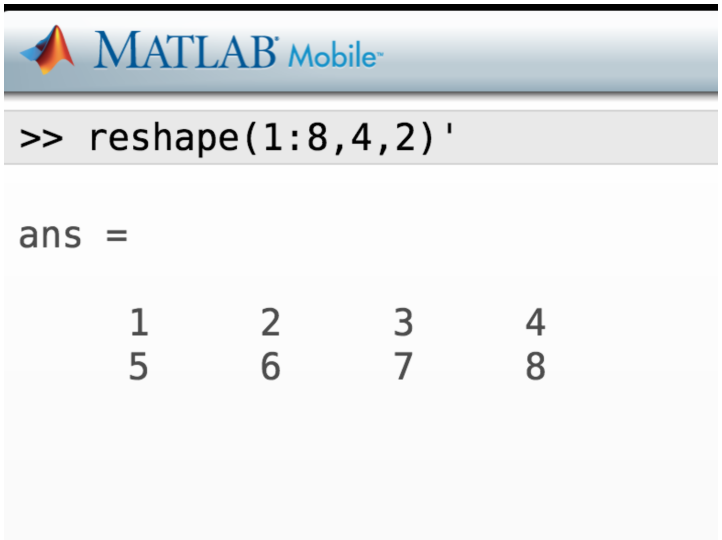
- Sub-matrices
- Reproduction
- Reshape
- Determinant, inversion

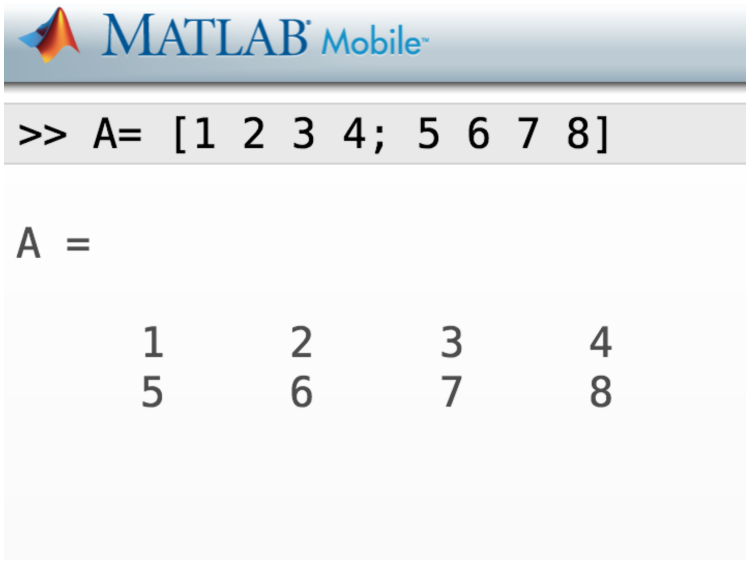
Flow control

- if, if else, find

Matrices

- A= [1 2 3 4; 5 6 7 8]
- reshape(1:8,4,2)'





Rows

- >> A(1,:)

```
ans =
   1     2     3     4
```

- >> A(2,:)

```
ans =
   5     6     7     8
```

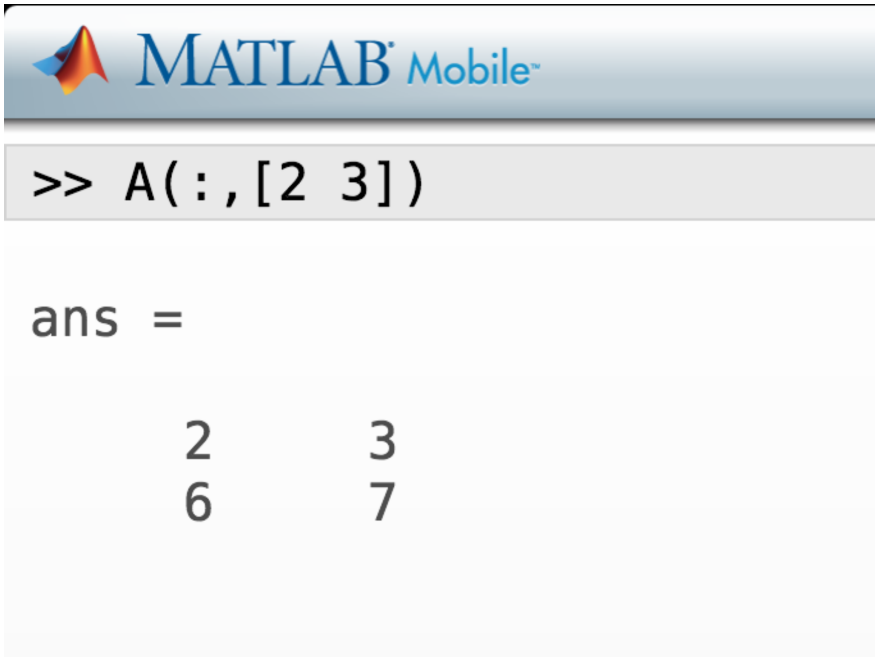
Columns

- >> A(:,1)

```
ans =
     1
     5
```

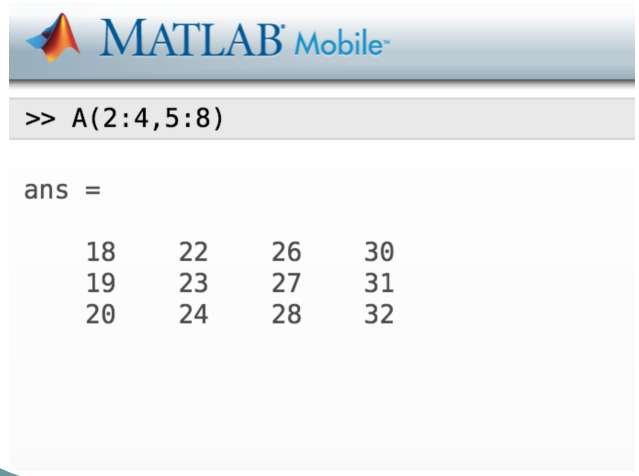
- >> A(:,[2 3])

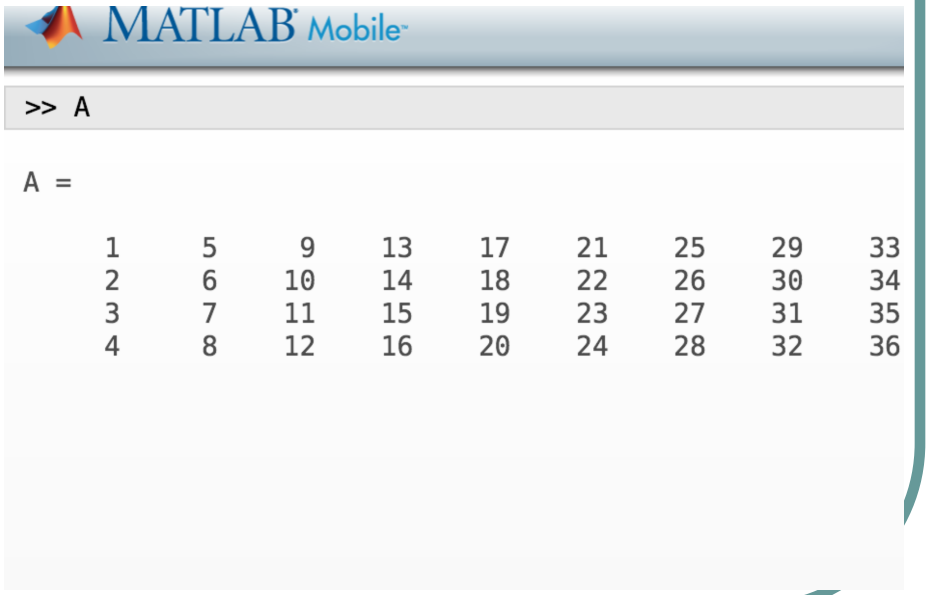
```
ans =
     2     3
     6     7
```



Sub-matrix

- a=1:1:36
- A=reshape(a,4,9)
- % Extract rows 2:4 and columns 5:8 of A
- A(2:4,5:8)





Reshape

```
v=1:12;
reshape(v,3,4)

ans =

     1     4     7    10
     2     5     8    11
     3     6     9    12
```

Reshape

- a=1:1:36
- m=4;n=9
- A=reshape(a,4,9)
- % reshape vector a to matrix A
- % A is composed of m rows and n columns

Gray image

cmw2.bmp

```
I=imread('cmw2.bmp');
imshow(I)
```

```
I=imread('cmw2.bmp');
image(I)
```

MATLAB Mobile

```
>> dir

.            .session    Shared
..           Published   cmw2.bmp
```

```
>> I=imread('cmw2.bmp');
>> image(I)
```

Sub-matrices

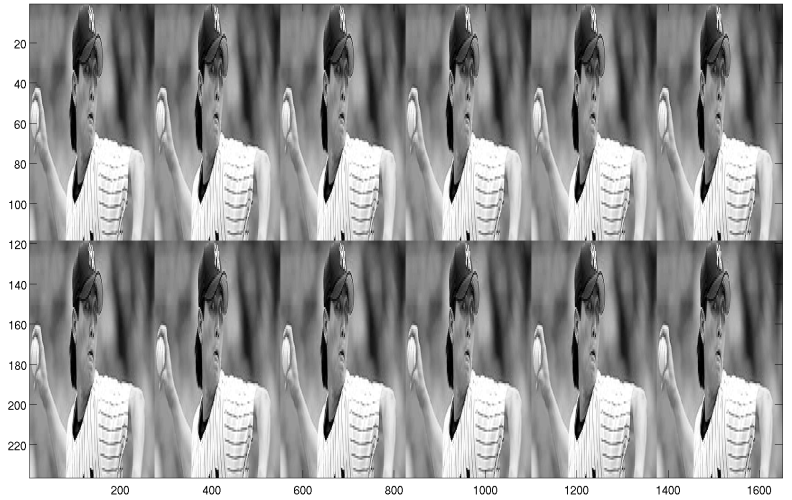
```
I=imread('cmw2.bmp');
image(I);
[m,n]=size(I);
m2=ceil(m/2);
n2=ceil(n/3*2);
Is=I(1:m2,1:n2); figure
imagesc(Is)
colormap(gray)
```

MATLAB Mobile

```
>> I=imread('cmw2.bmp');
image(I);
[m,n]=size(I);
m2=ceil(m/2);
n2=ceil(n/3*2);
Is=I(1:m2,1:n2); figure
imagesc(Is)
colormap(gray)
```

Repeat

```
I2=repmat(Is,2,3);
imagesc(I2);
colormap(gray)
```



Repmat

- A=[1 2;3 4]
- m=2;n=3;
- repmat(A,m,n)
- % repeat A vertically m times
- % repeat the result horizontally n times

MATLAB Mobile

```
>> A=[1 2;3 4]
m=2;n=3;
repmat(A,m,n)
```

```
A =

     1     2
     3     4


ans =

     1     2     1     2     1     2
     3     4     3     4     3     4
     1     2     1     2     1     2
     3     4     3     4     3     4
```

Get Rows

- A=reshape(1:16,4,4)
- a=3;b=1;c=2;
- A([a b c],:)
- % get rows specified by [a b c]

```
>> A=reshape(1:16,4,4)
a=3;b=1;c=2;
A([a b c],: )

A =

     1     5     9    13
     2     6    10    14
     3     7    11    15
     4     8    12    16


ans =

     3     7    11    15
     1     5     9    13
     2     6    10    14
```

Get Columns

- A=reshape(1:16,4,4)
- a=3;b=1;c=2;
- A(:,[a b c])
- % get columns specified by [a b c]

```
>> A=reshape(1:16,4,4)
a=3;b=1;c=2;
A(:,[a b c])
```

```
A =

     1     5     9    13
     2     6    10    14
     3     7    11    15
     4     8    12    16


ans =

     9     1     5
    10     2     6
    11     3     7
    12     4     8
```

Row exchange

```
V=reshape(1:12,3,4)
V([3 2],:)=V([2 3],:)

V =

     1     4     7    10
     3     6     9    12
     2     5     8    11
```

Column exchange

```
V=reshape(1:12,3,4)

V(:,[2 3])=V(:,[3 2])

V =

     1     7     4    10
     2     8     5    11
     3     9     6    12
```

Column sum

```
V=reshape(1:12,3,4)

V =

     1     4     7    10
     2     5     8    11
     3     6     9    12

>> sum(V)

ans =

     6    15    24    33
```

Column sum

```
V=reshape(1:12,3,4)

V =

     1     4     7    10
     2     5     8    11
     3     6     9    12

>> sum(V,1)

ans =

     6    15    24    33
```

Row sum

```
V=reshape(1:12,3,4)

V =

     1     4     7    10
     2     5     8    11
     3     6     9    12

>> sum(V,2)

ans =

    22
    26
    30
```

Mean of columns

```
V=reshape(1:12,3,4)

V =

     1     4     7    10
     2     5     8    11
     3     6     9    12

>> mean(V)

ans =

     2     5     8    11
```

Mean of rows

```
V=reshape(1:12,3,4)

V =

     1     4     7    10
     2     5     8    11
     3     6     9    12

>> mean(V,2)
```

Inversion

```
A=reshape(1:4,2,2);
B=inv(A)

B =

   -2.0000    1.5000
    1.0000   -0.5000
```

Linear system

$$2x + y - z = 1$$

$$-3x - 2y + 5z = 0$$

$$x + y + z = 5$$

inv

```
A=[2 1 -1;-3 -2 5;1 1 1];
b=[1 0 5]'
```

```
inv(A)*b

ans =

   -1.6000
    5.4000
    1.2000
```

Left devision

A=[2 1 -1;-3 -2 5;1 1 1];
b=[1 0 5]'

- A\b
- % Left devision
- % Ax=b could be solved by left devision

Determinant

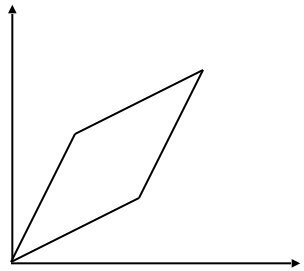
```
A=reshape(1:4,2,2);
>> det(A)

ans =

    -2
```

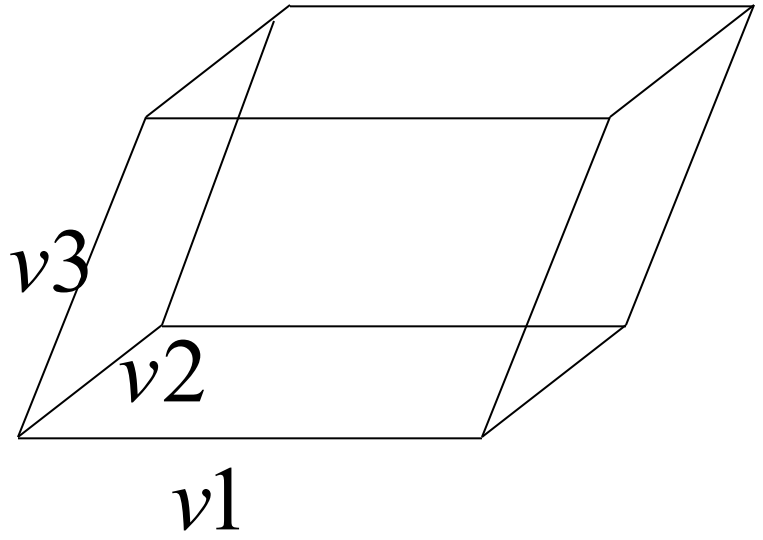
Area of a parallelogram

- v1,v2 denotes two vectors in a plane
- Calculate the area of the parallelogram determined by v1 and v2



Volume of a parallelepiped

A=[v1;v2;v3]
det(A)



```
reshape(randperm(9),3,3)

ans =

     5     9     6
     2     1     8
     7     3     4
```

Append Horizontally

```
>> A=reshape(randperm(9),3,3);
>> B=reshape(randperm(9),3,3);
>> C=[A B]

C =

     6     2     9     5     4     9
     3     4     5     6     3     7
     8     7     1     8     1     2
```

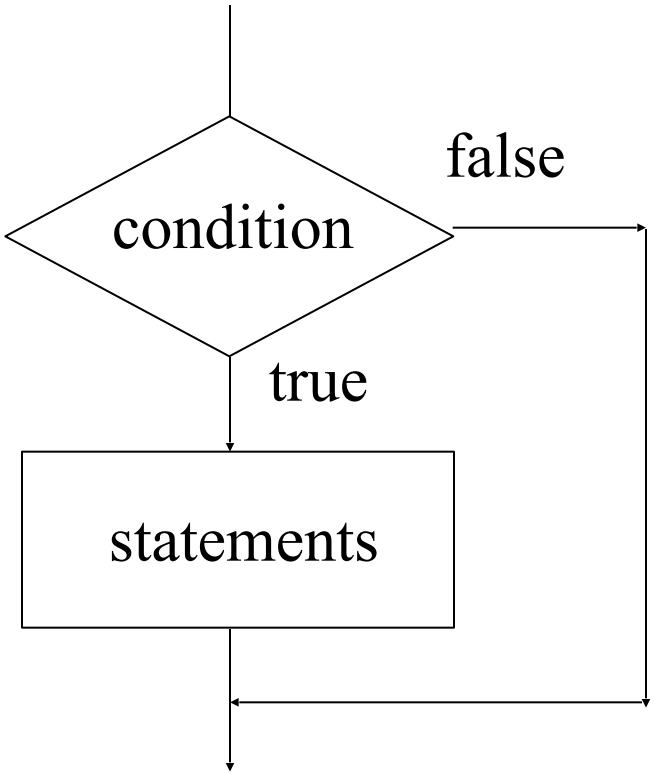
Append Vertically

```
>> C=[A;B]

A =

     6     2     9
     3     4     5
     8     7     1
     5     4     9
     6     3     7
     8     1     2
```

if



if else