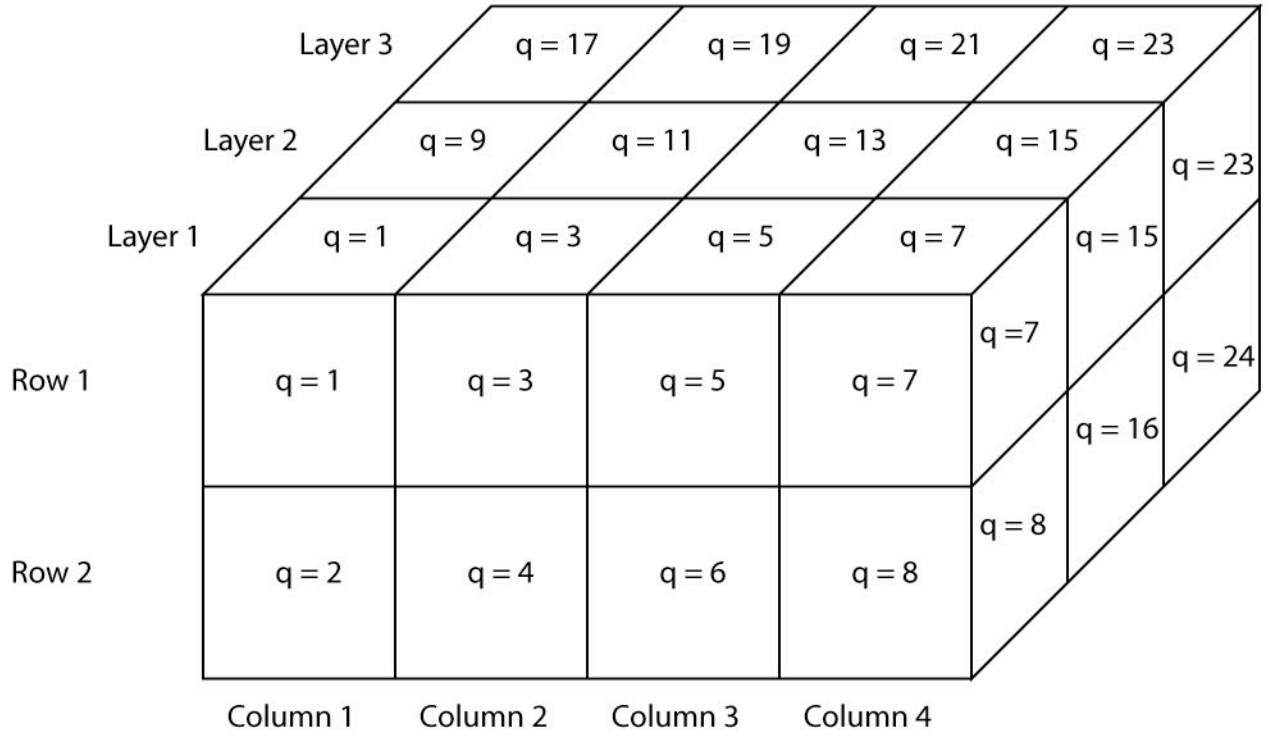
Three-dimensional Matrices

- Useful for representing a function of 3 variables [e.g., temperature in a volume; T = f(x,y,z)]
- Creating a 3-D matrix
- Size of a 3-D matrix
- Reshaping matrices
- Addressing elements in a 3-D matrix
- Creating a 3-D matrix with meshgrid
- 3-D visualization

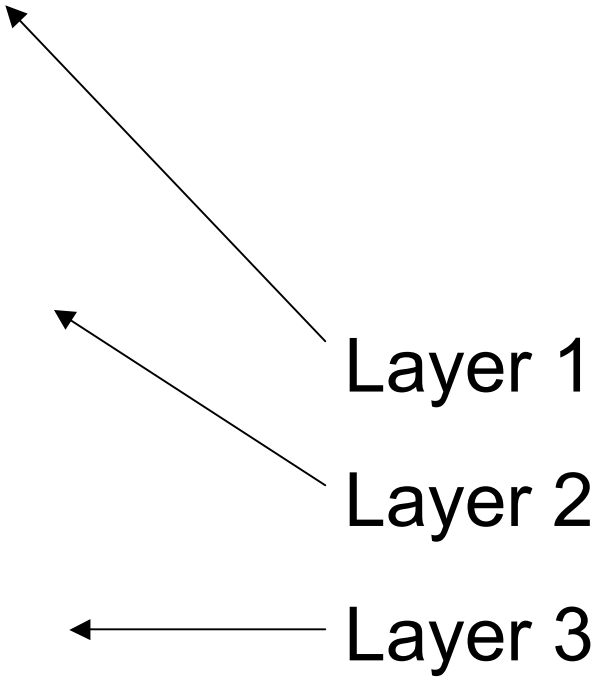
Three-dimensional Matrices



l = # of rows (here l = 2)
m = # of columns (here m = 4)
n = # of layers (here m =3)

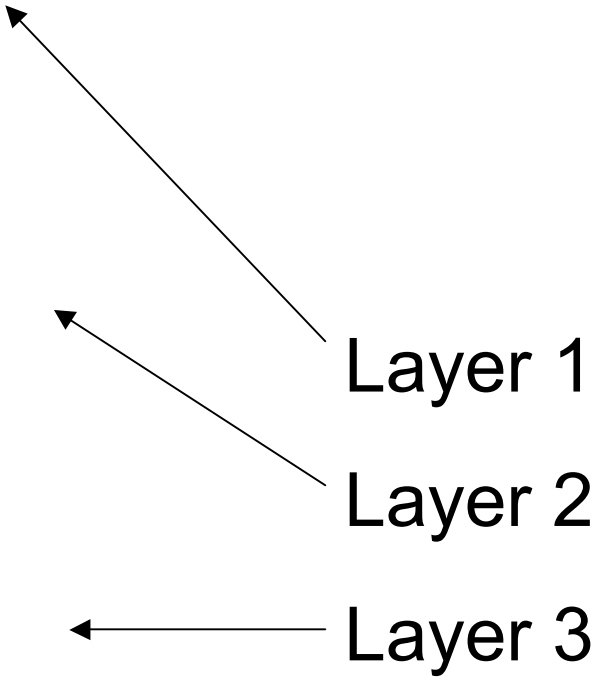
Creating 3-D Matrices (zeros)

```
>> q = zeros(2,4,3)

q(:,:,1) =

     0     0     0     0
     0     0     0     0

q(:,:,2) =

     0     0     0     0
     0     0     0     0

q(:,:,3) =

     0     0     0     0
     0     0     0     0
```



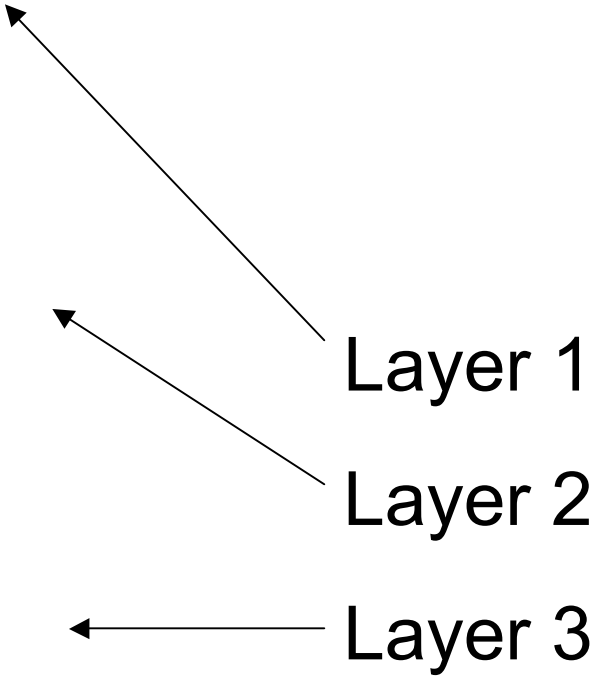
Creating 3-D Matrices (ones)

```
>> B = ones(2,4,3)

B(:,:,1) =

     1     1     1     1
     1     1     1     1

B(:,:,2) =

     1     1     1     1
     1     1     1     1

B(:,:,3) =

     1     1     1     1
     1     1     1     1
```



Creating 3-D Matrices (rand)

```
>> A = rand(2,4,3)

A(:,:,1) =

    0.9501    0.6068    0.8913    0.4565
    0.2311    0.4860    0.7621    0.0185

A(:,:,2) =

    0.8214    0.6154    0.9218    0.1763
    0.4447    0.7919    0.7382    0.4057

A(:,:,3) =

    0.9355    0.4103    0.0579    0.8132
    0.9169    0.8936    0.3529    0.0099
```



Size of a 3-D matrix

```
>> [l,m,n] = size(q)

l =

     2

m =

     4

n =

     3
```

2 rows

4 columns

3 layers

Reshaping a 3-D matrix (1)

```
x=1:24

x =

  Columns 1 through 7

    1     2     3     4     5     6     7

  Columns 8 through 14

    8     9    10    11    12    13    14

  Columns 15 through 21

   15    16    17    18    19    20    21

  Columns 22 through 24

   22    23    24
```

length(x) = 24

Reshaping a 3-D matrix (2)

```
>> y=x'

y =

     1
     2
     3
     4
     5
     6
     …
    21
    22
    23
    24
```

length(y) = 24

Reshaping a 3-D matrix (3)

```
>> q=reshape(y,2,4,3)

q(:,:,1) =

     1     3     5     7
     2     4     6     8

q(:,:,2) =

     9    11    13    15
    10    12    14    16

q(:,:,3) =

    17    19    21    23
    18    20    22    24
```

This reshapes y into a 3-D matrix with 2 rows, 4 columns, and 3 layers

# of elements in q must match # of elements in y

Layer 1

Layer 2

Layer 3

Reshaping a 3-D matrix (4)

```
>> zz = reshape(q,24,1,1)

zz =

     1
     2
     3
     4
     5
     6
     …
    21
    22
    23
    24
```

Reshaping a 3-D matrix (5)

```
>> zzz = z(:)

zzz =

     1
     2
     3
     4
     5
     6
     …
    21
    22
    23
    24
```

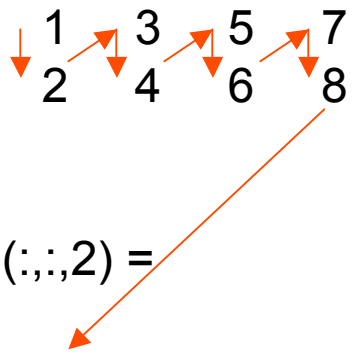
Addressing Elements in a 3-D Matrix (1)

```
>> q=q

q(:,:,1) =

     1     3     5     7
     2     4     6     8


q(:,:,2) =

     9    11    13    15
    10    12    14    16


q(:,:,3) =

    17    19    21    23
    18    20    22    24
```

```
>> qq = q(1,2,3)

qq =

    19

>> qqq = q(:,1,2:3)

qqq(:,:,1) =

     9
    10


qqq(:,:,2) =

    17
    18
```

Addressing Elements in a 3-D Matrix (2)

```
>> q=q

q(:,:,1) =

     1     3     5     7
     2     4     6     8

q(:,:,2) =

     9    11    13    15
    10    12    14    16

q(:,:,3) =

    17    19    21    23
    18    20    22    24
```



```
>> qqq = q(1)

qqq =

     1

>> qqqq = q(12)

qqqq =

    12
```

Creating a 3-D matrix with meshgrid

```
>> [x,y,z] = meshgrid(1:2,2:3,3:4)

x(:,:,1) =

     1     2
     1     2


x(:,:,2) =

     1     2
     1     2


y(:,:,1) =

     2     2
     3     3


y(:,:,2) =

     2     2
     3     3


z(:,:,1) =

     3     3
     3     3


z(:,:,2) =

     4     4
     4     4
```

3-D Visualization

- Numerous ways to visually represent functions of three variables
- See “Volume Visualization” under MATLAB Help (e.g., coneplot, slice)