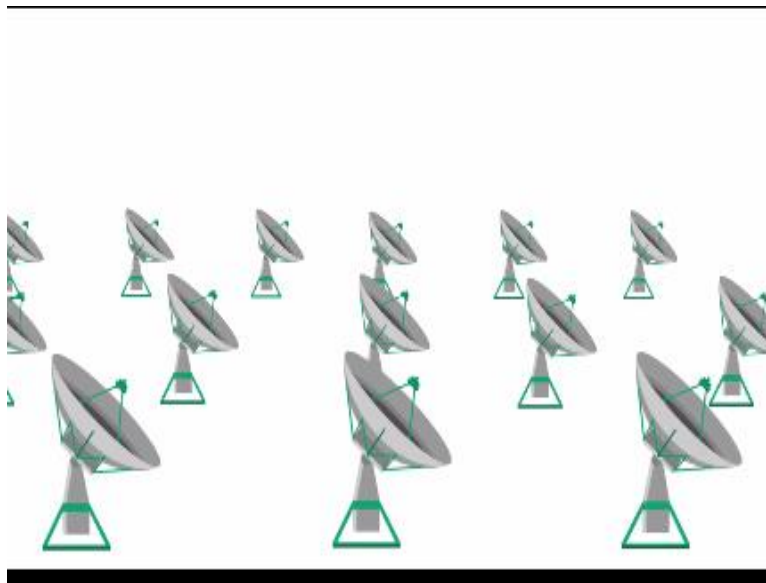


Matrix Creation Functions

Created by MathWorks for
Structural Dynamics

MathWorks

(Refer Slide Time: 00:06)




As the amount of data we collect grows will need to create some large matrices to perform calculations with the data.

(Refer Slide Time: 00:12)

```
MATLAB Commands
>> I = [1,0,0,0,0,0,0,0,0;...
0,1,0,0,0,0,0,0,0;...
0,0,1,0,0,0,0,0,0;...
0,0,0,1,0,0,0,0,0;...
0,0,0,0,1,0,0,0,0;...
0,0,0,0,0,1,0,0,0;...
0,0,0,0,0,0,1,0,0;...
0,0,0,0,0,0,0,1,0;...
0,0,0,0,0,0,0,0,1;...
0,0,0,0,0,0,0,0,1]
```

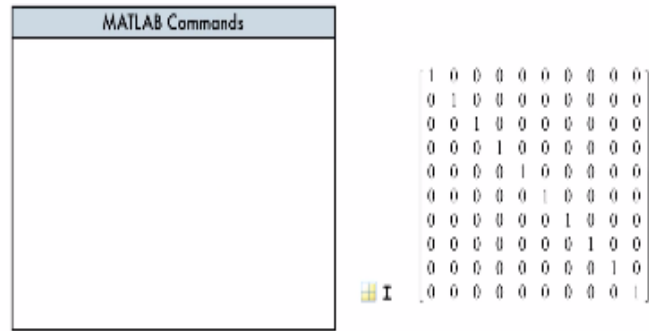
```

1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 1
```

 I

For example, we could create this 10 by 10 identity matrix like so, but who wants to do that.

(Refer Slide Time: 00:21)



The image shows a MATLAB interface. On the left is a window titled "MATLAB Commands" which is currently empty. To the right of this window is a 10x10 identity matrix, represented as a grid of 0s and 1s. The matrix is enclosed in square brackets. A small cursor icon is positioned to the left of the bottom row of the matrix.

```
[ 1 0 0 0 0 0 0 0 0 0  
0 1 0 0 0 0 0 0 0 0  
0 0 1 0 0 0 0 0 0 0  
0 0 0 1 0 0 0 0 0 0  
0 0 0 0 1 0 0 0 0 0  
0 0 0 0 0 1 0 0 0 0  
0 0 0 0 0 0 1 0 0 0  
0 0 0 0 0 0 0 1 0 0  
0 0 0 0 0 0 0 0 1 0  
0 0 0 0 0 0 0 0 0 1]
```

Fortunately there is a better way.

(Refer Slide Time: 00:24)

eye
Identity matrix

ones
Matrix of all 1's

zeros
Matrix of all 0's

rand
Uniformly distributed random numbers

randn
Normally distributed random numbers

randi
Uniformly distributed random integers

diag
Diagonal matrix

linspace
Evenly spaced vector

Functions to create many common matrices are included in MATLAB.

(Refer Slide Time: 00:28)

```
MATLAB Commands
>> I = eye(10);
```

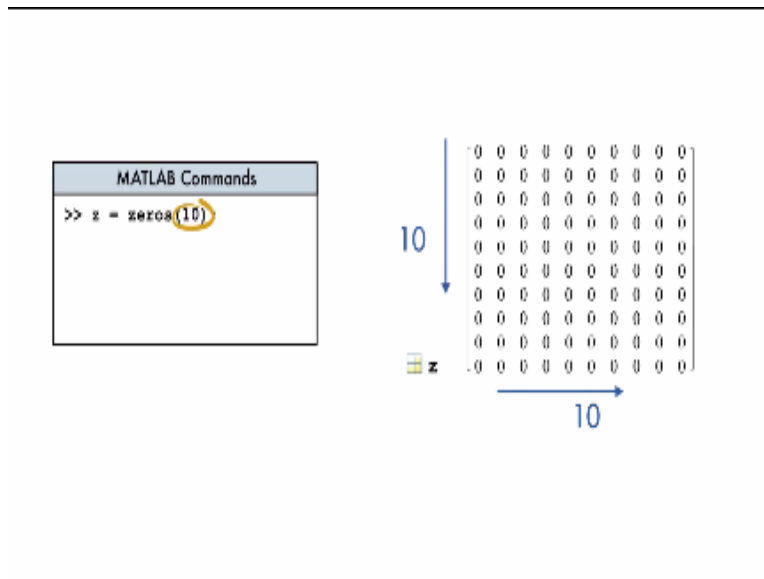
```

1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 1
```

 I

So to create an identity matrix we use the I function like this, the input variable specifies the size of the square matrix and bingo there is our 10 by 10 identity matrix, you can also use functions to initialize matrices.

(Refer Slide Time: 00:47)



For example, the zeros function creates a matrix of well zero. Notice that when using a single input the result is a square matrix here 10 rows and 10 columns.

(Refer Slide Time: 01:02)

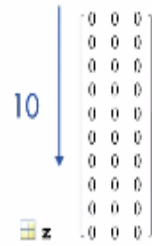
```
MATLAB Commands
>> z = zeros(10)
```

```
z =
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

But what if we need a rectangular matrix no problem we use 2 input arguments instead of one.

(Refer Slide Time: 01:10)

```
MATLAB Commands  
>> z = zeros(10,3);
```



The first input specifies the number of rows.

(Refer Slide Time: 01:13)

```
MATLAB Commands  
>> z = zeros(10,3);
```

```
z  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0 0  
3
```

And the second input the number of columns.

(Refer Slide Time: 01:16)

```
MATLAB Commands
>> z = zeros(10,3);

z =
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

We can use the exact same syntax with the ones function.

(Refer Slide Time: 01:20)

MATLAB Commands
>> z = zeros(10,3);
>> x = ones(2,4);
>> r = rand(3,2);


x $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

r $\begin{bmatrix} 0.1657 & 0.5425 \\ 0.7518 & 0.9616 \\ 0.3535 & 0.4102 \end{bmatrix}$

And rand function which creates a matrix of random numbers.

(Refer Slide Time: 01:27)

```
MATLAB Commands
>> z = zeros(10,3);
>> x = ones(2,4);
>> r = rand(3,2);
>> d = [1,0,0,0;0,2,0,0;...
0,0,3,0;0,0,0,4];
```

 d $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$

So the next time you find yourself entering a large matrix.

(Refer Slide Time: 01:32)

diag

Diagonals of matrices or (pre)diagonal elements of matrices

Syntax

```
d = diag(v)
S = diag(v,k)
```

Description

`d = diag(v)` returns a square diagonal matrix with the elements of vector `v` on the main diagonal.

`S = diag(v,k)` places the elements of vector `v` on the `k`th diagonal. `k=0` represents the main diagonal, `k>0` above the main diagonal, and `k<0` below the main diagonal.

`v = diag(S)` returns a column vector of the main diagonal elements of `S`.

`v = diag(S,k)` returns a column vector of the elements on the `k`th diagonal of `S`.

Examples

► Create Diagonal Matrices

Create a 3x3 matrix:

```
S = [2 -1 -2 -4];
```

Use `diag` to create a matrix with the elements of `v` on the main diagonal.

```
S = diag(v)
```

```
S =
```

Check the MATLAB documentation first to see if there is a function to do the job.

MathWorks

© 2015 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com / trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.