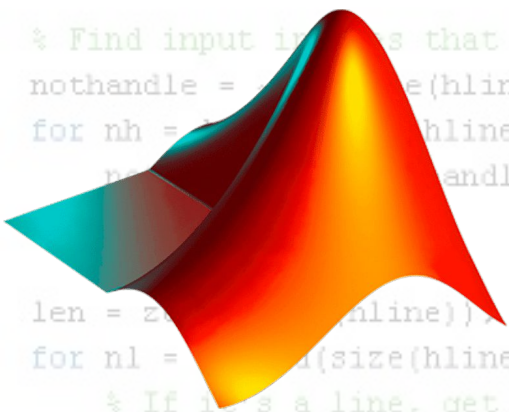


```
13 % [len,dims] = lengthofline([h1 h2])
14
15 % Find input lines that are not line objects
16 - nothandle = ~isline(hline);
17 - for nh = find(nothandle)
18 -     nothandle(nh) = ~strcmp('line',lower(get(hline(nh),'type')));
19 -
20
21 - len = 2;
22 - for nl = find(nothandle)
23     % If it's a line, get the data and compute the length
24 -     if ~notline(nl)
25 -         flds = get(hline(nl));
26 -         fdata = {flds.'YData','ZData'};
27 -         for nd = 1:length(fdata)
28 -             data{nd} = getfield(flds,fdata{nd});
29 -         end
30 % If there's no 3rd dimension, or all the data in one dimension is
31 % unique, then consider it to be a 2D line.
32 -         if isempty(data{3}) || ...
33             (length(unique(data{1}{:}))=1 ||
34              length(unique(data{2}{:}))=1 || ...
```

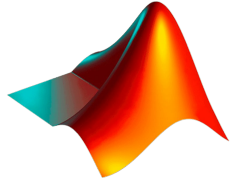


# MATLAB®

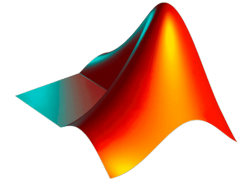
## A Glimpse into the Future of Humanity?

Matthew Gallagher, Ryan Hessler, Brandon Mitchell

# Why MATLAB?



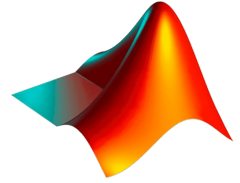
- Why not MATLAB?
- Inform the populace of *the* superior high-level language
- Taught and used here at Tech
- Useful even if you aren't a programmer



# But what is MATLAB?

- Proprietary, multi-paradigm, weakly-typed, interpreted\*, and imperative programming language
- *Matrix Laboratory*
  - Most functions are vectorized
- Early versions were just matrix calculators in FORTRAN
  - Invented by Cleve Moler
  - 71 built-in functions
  - Distributed for free in 1979
- Rewritten in C in early '80s
  - Still retains some FORTRAN syntax
  - The birth of the MATLAB programming language
- Commercially released in 1984
  - Became successful due to toolboxes (similar to libraries)


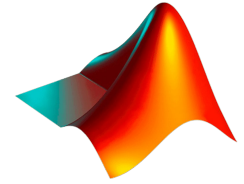
# But what is MATLAB?



*“MATLAB is a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models.”*

*-MathWorks, MATLAB's Developer*

# But what is MATLAB?



*“MATLAB is a programming and numeric computing platform used by thousands of engineers and scientists to analyze data, develop algorithms, and create models.”*

*MATLAB's Developer*

The screenshot displays the MATLAB R2018a software interface. The top menu bar includes options like HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The main workspace is divided into three primary sections:

- Current Folder:** Shows a file explorer view of the current directory, listing various files and subfolders.
- Editor:** Contains the MATLAB script `modern2_blog.m`. The script text is as follows:
 

```

1 %% MATLAB History, Modern MATLAB, part 2
2 % The ACM Special Interest Group on Programming Languages, SIGPLAN,
3 % expects to hold the fourth in a series of conferences on
4 % the History of Programming Languages in 2020, see
5 % <https://hop14.sigplan.org/HOPL-IV>. The first drafts of
6 % papers are to be submitted by August 2018. That long lead time
7 % gives me the opportunity to write a detailed history of MATLAB.
8 % I plan to write the paper in sections, which I'll post in
9 % this blog as they are available.
10 %
11 % This is the sixth such installment and the second of a three-part
12 % post about modern MATLAB.
13 %
14 % MATLAB is not just a Matrix Laboratory any more.
15 % It has evolved over the past 35 years into a rich technical
16 % computing environment. Here are some samples of the features
17 % now available.
      
```
- Command Window:** Shows the execution of MATLAB commands:
 

```

t =
    1/(4*cos(x) + 5)
    1
-----
    4 cos(x) + 5

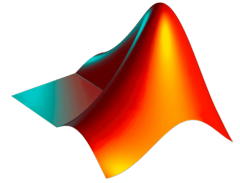
f2 =
(4*cos(x))/(4*cos(x) + 5)^2 + (32*sin(x)^2)/(4*cos(x) + 5)^3
g =
-8/(tan(x/2)^2 + 9)
e =
8/(tan(x/2)^2 + 9) + 1/(4*cos(x) + 5)
s =
    1
fx Trial>>
      
```
- Workspace:** A table listing variables and their properties:
 

Name	Value	Max
ax	1x1 Axes	
c	1x1 duration	
d	7	7
e	1x1 sym	
f	1x1 sym	
f2	1x1 sym	
g	1x1 sym	
G	1x1 digraph	
m	5	5
Math101	1x1 struct	
microc	1x1 duration	
microcentury_long	1x1 duration	
microcentury_short	1x1 duration	
mu	5	5
px	1x1 Line	
s	1x1 sym	
sinc	@(x)sin(x)/x	
t	403x1 double	30
T	3x3 table	
t0	1x1 datetime	
t1	1x1 datetime	
t2	1x1 datetime	
tspan	[0,30]	30
vdp	@(t,y)[0,1-1,mu*(1-y...	
x	1x1 sym	
X	268x400x3 uint8	255
y	403x2 double	7.6460
y0	[0,0.0100]	0.0100

The status bar at the bottom indicates the current position (Ln 18, Col 1) and the number of trial days remaining (208).

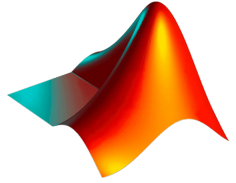
Current MATLAB interface.

# But why MATLAB?



- Originally designed for linear algebra
  - Built in functions for integration, differentiation, interpolation, complex number support, etc.
  - Really good at math and complex calculations
- Analysis, exploration, and visualization of data
  - Built in graphing and plotting support
  - Simulink for system modeling and simulation
- Image editing and manipulation
- Embedded systems and control systems
  - Auto-generation of C and HDL code based off of a model diagram
- MATLAB App Designer
  - MATLAB Web App Server™
  - Standalone apps with MATLAB Runtime and MATLAB Compiler
- Parallel computing

# So, what's the catch?



- Performance heavy and bulky
  - Can use a lot of memory
- Not useful for certain types of applications
- Can be intimidating for first time users
- Quite pricey
  - GNU Octave, Julia, Python can be alternatives



- Standard
- Education
- Home
- Student

### Individual

Select this license if you are an end user and you want to operate, install, and administer the software yourself. Your organization can also designate an administrator to manage a group of Individual licenses for the convenience of centralized administration.

[» Learn more](#)

Network Named User

Concurrent

### Parallel Computing Toolbox

- USD 1,000** i  
Perpetual license
- USD 400** i  
Annual license

[Buy now](#)

**Required Products:**

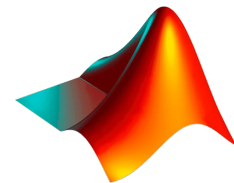
MATLAB

[-View another product-](#) ▼

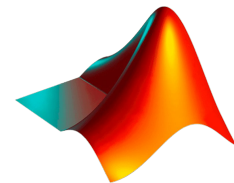
Price applies for purchase and use in United States. For pricing in other regions [contact sales](#). Pricing excludes TAX/VAT.

One big one. Cheaper for students, educational institutions.

# Data Types



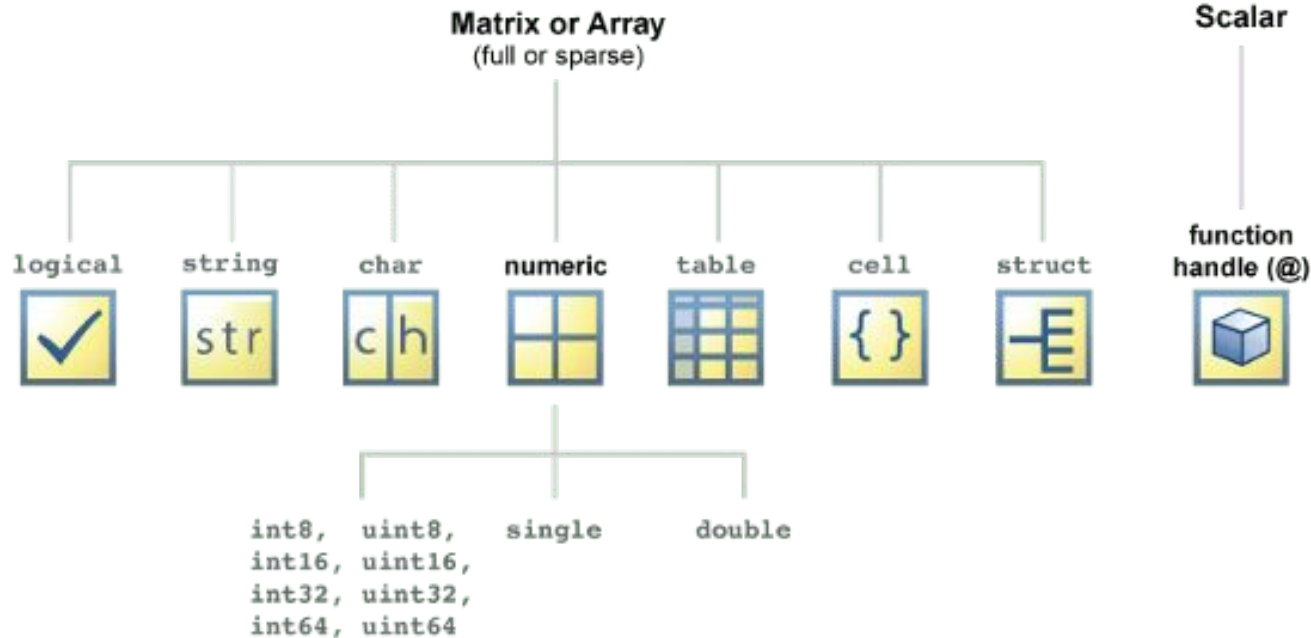
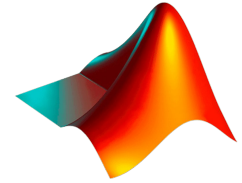
- Logical
  - Boolean
- String
  - “this is a string”
- Char
  - ‘this is a char array’
- Numeric
- Tables
  - “Use tables to store mixed-type data conveniently, access data with numeric or named indexing, and to store metadata.”



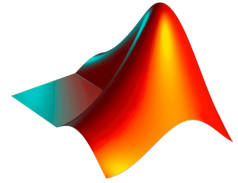
# Data Types (cont.)

- Cell
  - Array in which each element can be of differing size and type
  - Use for jagged arrays
  - Index with () returns a cell, index with {} to get value held
- Struct
  - Like structs in C
  - Structure arrays and logical indexing work together very well
- Function Handles
  - Always a scalar (1 x 1 matrix)

# Data Types (cont.)



# Syntax / Unique Features



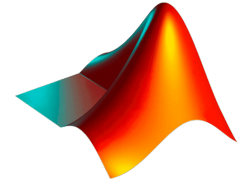
```
>> x = 17
x =
    17

>> x = 'hat'
x =
    hat

>> x = [3*4, pi/2]
x =
    12.0000    1.5708

>> y = 3*sin(x)
y =
   -1.6097    3.0000
```

# Syntax / Unique Features



```
>> x = 5
```

```
x =
```

```
5
```

```
>> y = 5;
```

```
>> x = 5; y = 5; z = 5;
```

```
>> x = 5, y = 5, z = 5
```

```
x =
```

```
5
```

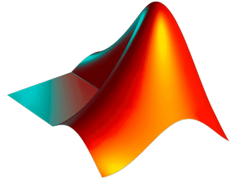
```
y =
```

```
5
```

```
z =
```

```
5
```

# Syntax / Unique Features



---

```
>> x = 6; y = 5, x = x * y
```

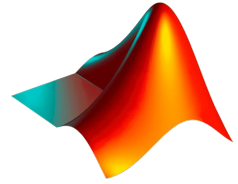
```
y =
```

```
5
```

```
x =
```

```
30
```

# Syntax / Unique Features



```
>> x = [1 2 3 4 5]
x =

     1     2     3     4     5

>> y = 1:3:18
y =

     1     4     7    10    13    16

>> z = 1:8
z =

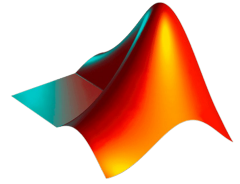
     1     2     3     4     5     6     7     8

>> a = linspace(1, 5, 10)
a =

    1.0000    1.4444    1.8889    2.3333    2.7778    3.2222    3.6667    4.1111    4.5556    5.0000
```



# Syntax / Unique Features



```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

```
A =
```

```
16  3  2 13
```

```
 5 10 11  8
```

```
 9  6  7 12
```

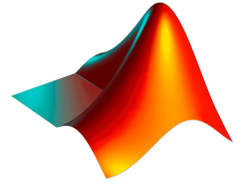
```
 4 15 14  1
```

```
>> A(2,3)
```

```
ans =
```

```
11
```

# Syntax / Unique Features



```
>> A(2:4, 3:4)
ans =

    11     8
     7    12
    14     1

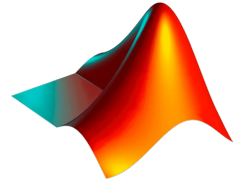
>> A(2:end, 3:end)
ans =

    11     8
     7    12
    14     1

>> A([1, 3:4], [1, end])
ans =

    16    13
     9    12
     4     1
```

# Syntax / Unique Features



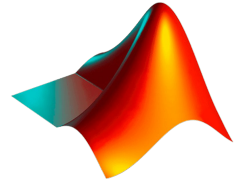
```
>> x = randi([1 10], 10)
x =

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

>> x(:, [1 4 5 7:9])
ans =

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

# Syntax / Unique Features



```
>> doc randi
>> x = randi([0 5], 5)
x =

     5     2     4     2     1
     5     1     2     2     2
     2     5     2     2     2
     2     1     3     0     2
     0     2     0     4     2

>> y = randi([0 5], 5)
y =

     4     4     5     0     0
     5     2     5     4     3
     3     5     2     0     3
     5     4     1     4     3
     5     4     5     2     3

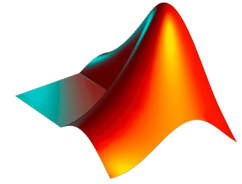
>> x + y
ans =

     9     6     9     2     1
    10     3     7     6     5
     5    10     4     2     5
     7     5     4     4     5
     5     6     5     6     5

>> x ./ y
ans =

    1.25000    0.50000    0.80000         Inf         Inf
    1.00000    0.50000    0.40000    0.50000    0.66667
    0.66667    1.00000    1.00000         Inf    0.66667
    0.40000    0.25000    3.00000    0.00000    0.66667
    0.00000    0.50000    0.00000    2.00000    0.66667
```

# Syntax / Unique Features



```
>> x = 1:5
x =

     1     2     3     4     5

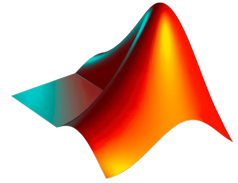
>> sin(0)
ans = 0
>> sin(x)
ans =

     0.84147     0.90930     0.14112    -0.75680    -0.95892

>> sin(x) + x
ans =

     1.8415     2.9093     3.1411     3.2432     4.0411
```

# Syntax / Unique Features



```
>> grades = randi([0, 100], 1, 15)
grades =

    35    38    27    38    11    61    95    98    94    68    7    17    78    37    65

>> failed = grades(grades < 60)
failed =

    35    38    27    38    11    7    17    37

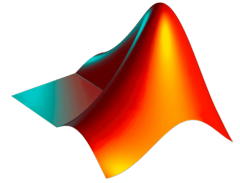
>> passed = grades(grades >= 60)
passed =

    61    95    98    94    68    78    65

>> [failed passed]
ans =

    35    38    27    38    11    7    17    37    61    95    98    94    68    78    65
```

# Syntax / Unique Features



```
>> grades >= 60
```

```
ans =
```

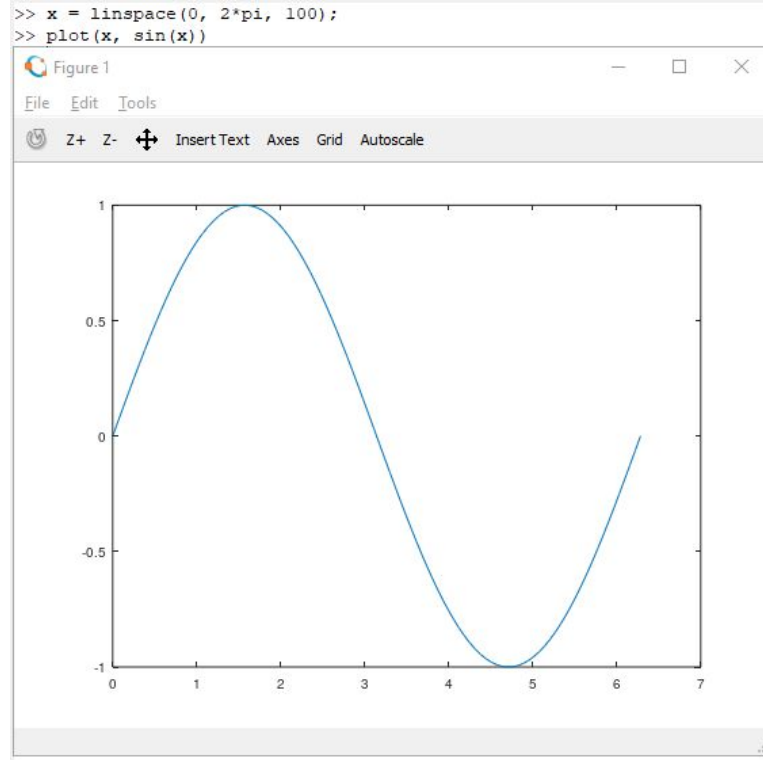
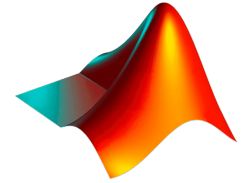
```
0 0 0 0 0 1 1 1 1 1 0 0 1 0 1
```

```
>> grades(grades < 60) = 70
```

```
grades =
```

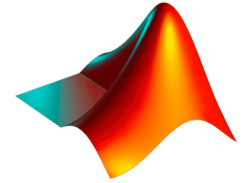
```
70 70 70 70 70 61 95 98 94 68 70 70 78 70 65
```

# Syntax / Unique Features

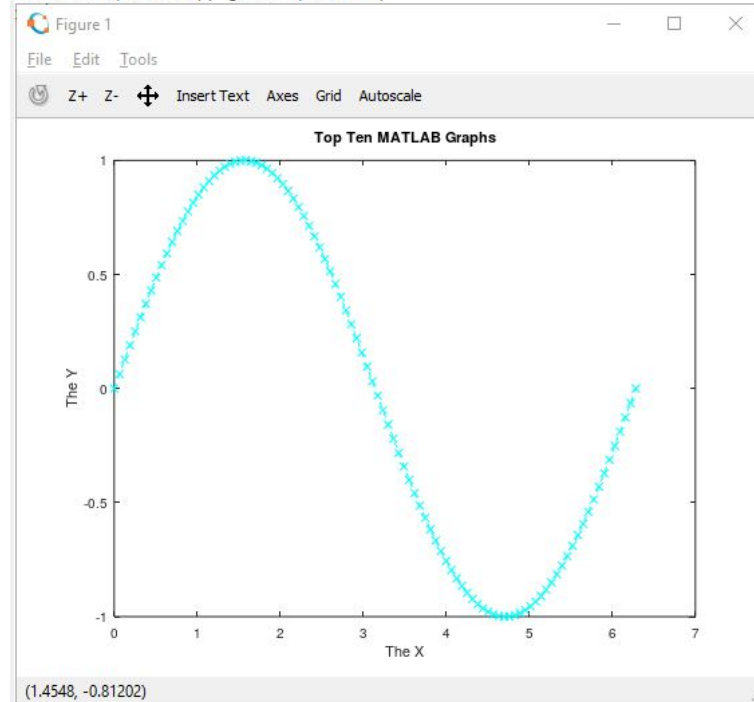




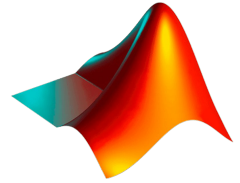
# Syntax / Unique Features



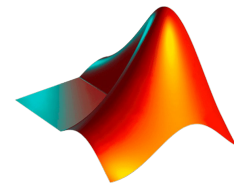
```
>> plot(x, sin(x), '--xc')  
>> title('Top Ten MATLAB Graphs')  
>> xlabel('The X'), ylabel('The Y')
```



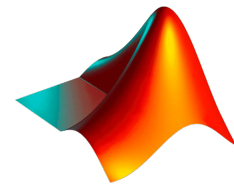
# Conclusion



- Proprietary, multi-paradigm, weakly-typed, interpreted\*, and imperative programming language
- Has many uses ...
  - Linear algebra
  - Data analysis
  - Embedded systems
- ... but also several drawbacks
  - Cost
  - Can be confusing at first
- Syntax is unique and designed with matrices in mind
- Nonetheless, is a unique and cool language that will someday take over as the #1 language.

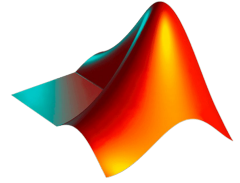


*Any questions?*



Workshop Time!!!

# Sources



<https://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>

<https://en.wikipedia.org/wiki/MATLAB>

<https://www.mathworks.com/discovery/what-is-matlab.html>

[https://www.mathworks.com/help/matlab/import\\_export/exporting-to-excel-spreadsheets.html](https://www.mathworks.com/help/matlab/import_export/exporting-to-excel-spreadsheets.html)

[https://www.mathworks.com/help/matlab/matlab\\_prog/fundamental-matlab-classes.html](https://www.mathworks.com/help/matlab/matlab_prog/fundamental-matlab-classes.html)

<https://www.mathworks.com/help/matlab/ref/keyboard.html>

<https://www.mathworks.com/help/matlab/ref/varargout.html>

<https://www.mathworks.com/help/matlab/tables.html>

<https://www.mathworks.com/products/compiler.html>

<https://www.mathworks.com/products/compiler/matlab-runtime.html>

<https://www.mathworks.com/products/matlab.html>

<https://www.mathworks.com/products/matlab/app-designer.html>

<https://www.mathworks.com/products/simulink.html>

[https://www.tutorialspoint.com/matlab/matlab\\_operators.htm](https://www.tutorialspoint.com/matlab/matlab_operators.htm)