

# MATLAB Introduction

## Introduction:

Here we are introduced to MATLAB, which we will use throughout our education for matrix and vector numerical computing. MATLAB is also an essential tool for statistics, data analysis, and visualization. Lastly, and very relevant to electrical engineering, MATLAB can be used for signal and image processing, communications, and control design.

## Question 1:

### Results:

The Cartesian representations for the complex number 1-5 in order are:  $1.2247 + 1.2247i$ ,  $1.7321 + i$ ,  $-1.2247 + 1.2247i$ ,  $-2$ ,  $-2$ . Their magnitudes are:  $1.7321$ ,  $2$ ,  $1.7321$ ,  $2$ ,  $2$ .

### Code:

```
>> exp_to_cart = @(r, th) (r .* cos(th)) + (r .* sin(th) * i);
>> r = [sqrt(3), 2, sqrt(3), 2, 2];
>> th = [pi/4, pi/6, 3*pi/4, pi, -pi];
>> a = exp_to_cart(r, th)

a =

Columns 1 through 3

    1.2247 + 1.2247i    1.7321 + 1.0000i   -1.2247 + 1.2247i

Columns 4 through 5

   -2.0000 + 0.0000i   -2.0000 - 0.0000i

>> abs(a)

ans =

    1.7321    2.0000    1.7321    2.0000    2.0000
```

### Comments about nature of results, graphs, code:

The conversion from exponential form to Cartesian uses Euler's Formula and the magnitude is simply the distance of the complex number from the origin, calculated using the Pythagorean Theorem. This shows how MATLAB can be used to quickly compute the answers to many different questions organized into a vector. The last two complex numbers, despite having different angles, are the same, since they each rotate the same distance in "different directions".

## Questions 2:

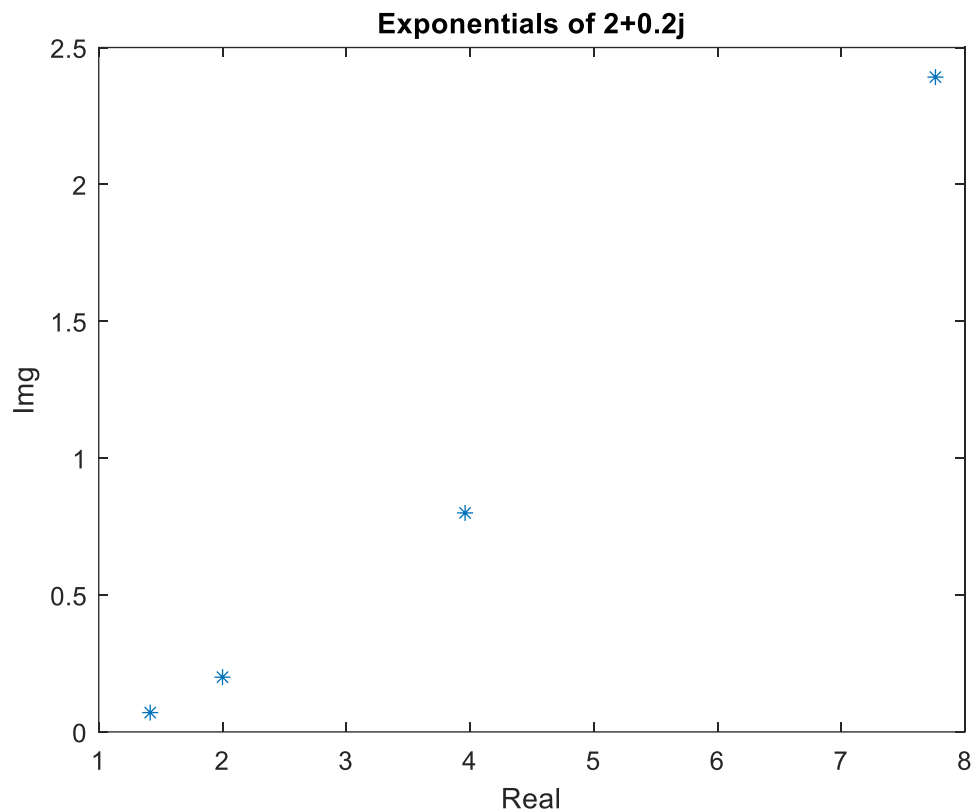
### Results:

See the plot. The value with  $\text{Real}() = 2$  is  $Z$  on the graph,  $Z^2$  is second largest,  $Z^3$  is the largest, and  $Z^{0.5}$  is the smallest.

### Code:

```
>> z = 2 + 0.2 * j;  
>> z2 = z^2;  
>> z3 = z^3;  
>> z4 = z^0.5;  
>> plot([z, z2, z3, z4], '*')  
>> xlabel("Real")  
>> ylabel("Img")  
>> title("Exponentials of 2+0.2j")
```

### Plot:



### Comments:

Z grows as its exponential increases and shrinks with fractional exponentials similar to real numbers.

## Question 3:

### Results:

The daily temperatures (maximum, average, and minimum) at the start of each month over the last 36 months are displayed on the plot.

### Code:

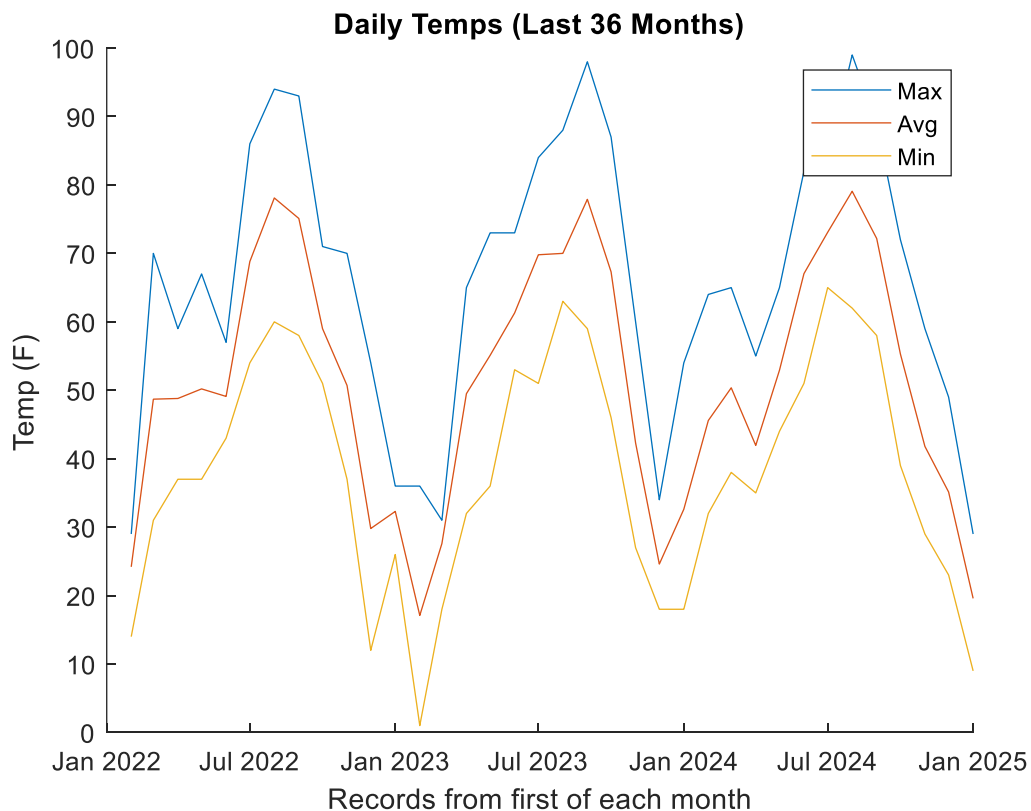
```
>> % These values were painstakingly copied from the website by hand!  
>> max_temp = [29 49 59 72 89 99 84 82 65 55 65  
64 54 34 60 87 98 88 84 73 73 65 31 36 36 54  
70 71 93 94 86 57 67 59 70 29];  
>> avg_temp = [19.59 35.13 41.8 55.29 72.17 79.08 73.08 67.04 52.92 41.94  
50.36 45.58 32.61 24.6 42.3 67.3 77.9 70.0 69.8 61.3 55.1 49.5 27.6 17.1 32.3
```

```

29.8 50.7 59.0 75.1 78.1 68.8 49.1 50.2 48.8 48.7 24.2];
>> min_temp = [9      23      29      39      58      62      65      51      44      35      38
32      18      18      27      46      59      63      51      53      36      32      18      1      26      12
37      51      58      60      54      43      37      37      31      14];
>> max_temp = flip(max_temp); % So the array goes from 2022 to 2025
>> avg_temp = flip(avg_temp);
>> min_temp = flip(min_temp);
>> % The code below makes a vector of dates from 2025 to 2022
>> x_dates = [datetime(2025,1,1) flip(datetime(2024,1:12,1))
flip(datetime(2023,1:12,1)) flip(datetime(2022,2:12,1))];
>> x_dates = flip(x_dates); % So array goes from 2022 to 2025
>> hold on
>> plot(x_dates, max_temp)
>> plot(x_dates, avg_temp)
>> plot(x_dates, min_temp)
>> xlabel("Records from first of each month")
>> ylabel("Temp (F)")
>> legend("Max", "Avg", "Min")
>> title("Daily Temps (Last 36 Months)")
>> hold off

```

### Plot:



### Comments:

All three recordings on each first day of the month show a cyclic pattern every year, with winter centered around January and summer around August. There is a slight trend in the data showing a general increase in temperatures over time, although more data would be needed to confirm this. The readings also seem very erratic in early-2022 and winter 2022-2023 for some reason not included in the graph data alone.

## Question 4:

### Results:

MATLAB is consistent with all the rules of linear algebra and element-wise multiplication of matrixes is different from traditional matrix-matrix multiplication. Traditional summation is already equivalent to element-wise summation (and it is implicitly the same in MATLAB, with no separate syntax for element-wise summation).

### Code:

```
>> A = [1 0 8; 7 9 -8; -6 5 4]
```

```
A =
```

1	0	8
7	9	-8
-6	5	4

```
>> B = [1 2 3; 4 -5 6; 7 8 8]
```

```
B =
```

1	2	3
4	-5	6
7	8	8

```
>> C = [7 9 7; 6 5 8; 7 -6 5]
```

```
C =
```

7	9	7
6	5	8
7	-6	5

```
>> A*B
```

```
ans =
```

57	66	67
-13	-95	11
42	-5	44

```
>> A.*B
```

```
ans =
```

1	0	24
28	-45	-48
-42	40	32

```
>> A+B
```

```
ans =
```

2	2	11
11	4	-2
1	13	12

```
>> A.+B
```

```
A.+B
```

```
↑
```

Invalid use of operator.

>> A+C

ans =

8	9	15
13	14	0
1	-1	9

>> C+A

ans =

8	9	15
13	14	0
1	-1	9

>> (A+C)+B

ans =

9	11	18
17	9	6
8	7	17

>> A+(C+B)

ans =

9	11	18
17	9	6
8	7	17

>> 4\*(A+B)

ans =

8	8	44
44	16	-8
4	52	48

>> 4\*A+4\*B

ans =

8	8	44
44	16	-8
4	52	48

>> B\*(C+A)

ans =

37	34	42
-27	-40	114
168	167	177

>> B\*C+B\*A

ans =

37	34	42
-27	-40	114
168	167	177

**Comments:**

- a)  $A*B$  is traditional matrix multiplication, which is equivalent to finding a matrix which performs the same linear transformation as both inputs in order.  $A.*B$  is element-wise, which multiplies the elements at each index shared between the matrixes.
- b) Summation is element-wise;  $A.+B$  is invalid syntax as  $A+B$  is already element-wise.
- a) Matrix addition is commutative (see console log).
- b) Matrix addition is associative (ibid.).
- c) Scalar distributive multiplication works (ibid.).
- d) Matrix distributive multiplication works (ibid.).

**Conclusion:**

In this lab, I learned the basics of linear algebra, data analysis, and complex arithmetic using MATLAB. Learning how to use `datetime()` for question 3 was new to me and hopefully it will be useful in the future. This also includes `flip()`, which I learned doing this lab. Copying the data for question 3 took most of the time and was very repetitive. I successfully used MATLAB in many of the different applications mentioned above.