MATH 211.3 Winter Term 2024 Assignment

Assignment #00

Name: Kanwar Pannu

Student Number: 11336732

NSID: eza147

Problem 1

**2.**

>> 4+3

ans =

7

>> 2^2

ans =

4

>> sin(2\*pi)+exp(−3/2)

sin(2\*pi)+exp(−3/2)

↑

Error: Invalid text character. Check for unsupported symbol, invisible

character, or pasting of non-ASCII characters.

>> sin(2\*pi)+exp(-3/2)

ans =

0.2231

>> 5+5j

ans =

5.0000 + 5.0000i

>> atan(5/5)

ans =

0.7854

>> 10\*log10(0.5)

ans =

-3.0103

**3.**

%computing ex 1 part 3

>> 2^5/(2^5-1)

ans =

1.0323

>> (1-1/2^5)^-1

ans =

1.0323

>> (sqrt(5)-1)/(sqrt(5)+1)^2

ans =

0.1180

%In part 3 in 1st bullet point answer comes same with both expressions

**4.**

%computing ex1 part 4

>> exp(3)

ans =

20.0855

>> log(exp(3))

ans =

3

>> log10(exp(3))

ans =

1.3029

>> log10(10^5)

ans =

5

**5.**

%computing ex1 part5

>> sin(pi/6)

ans =

0.5000

>> cos(pi)

ans =

-1

>> tan(pi/2)

ans =

1.6331e+16

>> sin(pi/6)^2 + cos(pi/6)^2

ans =

1

Problem 2

**2.**

>> A=[1 3 2;2 1 1;3 2 3]

A =

1 3 2

2 1 1

3 2 3

%command to get desired B matrix

>> B = [A(1,2:3);A(2,1:2)]

B =

3 2

2 1

**3. (a)**

>> A=[1 3 2;2 1 1;3 2 3]

A =

1 3 2

2 1 1

3 2 3

>> B = [A(1,2:3);A(2,1:2)]

B =

3 2

2 1

>> A =[1 2 3; 2 2 2; -1 2 1]

A =

1 2 3

2 2 2

-1 2 1

>> B = [1 0 0; 1 1 0; 1 1 1]

B =

1 0 0

1 1 0

1 1 1

>> C = [1 1; 2 1; 1 2]

C =

1 1

2 1

1 2

>> A+B

ans =

2 2 3

3 3 2

0 3 2

>> A\*B

ans =

6 5 3

6 4 2

2 3 1

>> A+C

Error using +

Arrays have incompatible sizes for this operation.

Related documentation

>> B\*C

ans =

1 1

3 2

4 4

>> B-A

ans =

0 -2 -3

-1 -1 -2

2 -1 0

>> A\*C

ans =

8 9

8 8

4 3

>> C-B

Error using -

Arrays have incompatible sizes for this operation.

Related documentation

>> C\*A

Error using \*

Incorrect dimensions for matrix multiplication. Check

that the number of columns in the first matrix matches

the number of rows in the second matrix. To operate on

each element of the matrix individually, use TIMES

(.\*) for elementwise multiplication.

Related documentation

% Errors occur whenever the matrix sizes do not match for e.g. A is 3x3 and C is 3x2 therefore cannot add or subtract and similar problem with multiplication and division.

**3(b)**

A\*B is matrix multiplication where rules of matrix multiplication are applied.

A.\*B is element by element multiplication where corresponding elements of both matrices are multiplied.

Both operations result in same error if matrix sizes are incompatible.

**5.**

>> t = 1:1:10

t =

Columns 1 through 9

1 2 3 4 5 6 7 8 9

Column 10

10

>> theta = linspace(0,pi,32)

theta =

Columns 1 through 5

0 0.1013 0.2027 0.3040 0.4054

Columns 6 through 10

0.5067 0.6081 0.7094 0.8107 0.9121

Columns 11 through 15

1.0134 1.1148 1.2161 1.3174 1.4188

Columns 16 through 20

1.5201 1.6215 1.7228 1.8242 1.9255

Columns 21 through 25

2.0268 2.1282 2.2295 2.3309 2.4322

Columns 26 through 30

2.5335 2.6349 2.7362 2.8376 2.9389

Columns 31 through 32

3.0403 3.1416

>> x = 2\*sin(theta)

x =

Columns 1 through 5

0 0.2023 0.4026 0.5987 0.7887

Columns 6 through 10

0.9706 1.1425 1.3027 1.4496 1.5816

Columns 11 through 15

1.6973 1.7956 1.8755 1.9362 1.9769

Columns 16 through 20

1.9974 1.9974 1.9769 1.9362 1.8755

Columns 21 through 25

1.7956 1.6973 1.5816 1.4496 1.3027

Columns 26 through 30

1.1425 0.9706 0.7887 0.5987 0.4026

Columns 31 through 32

0.2023 0.0000

>> y = (t-1)/(t+1)

y =

0.7426

>> z = sin(theta.^2)/theta.^2

z =

0.0217

>>

**7.**

>> theta = [0 pi/4 pi/2 3\*pi/4 pi 5\*pi/4]

theta =

Columns 1 through 5

0 0.7854 1.5708 2.3562 3.1416

Column 6

3.9270

>> r = 2

r =

2

>> x = r\*cos(theta)

x =

Columns 1 through 5

2.0000 1.4142 0.0000 -1.4142 -2.0000

Column 6

-1.4142

>> y = r\*sin(theta)

y =

Columns 1 through 5

0 1.4142 2.0000 1.4142 0.0000

Column 6

-1.4142

>> %checking radius values on r^2

>> sqrt(x.^2 + y.^2)

ans =

2 2 2 2 2 2

Since answer is always 2 therefore and y indeed satisfy the equation of a circle, by computing the radius r = sqrt(x^2 + y^2).

**8****.**

>> r = 0.5

r =

0.5000

>> n = 10

n =

10

**%computing series from 0 to 10**

>> series = r.^(0:n)

series =

Columns 1 through 4

1.0000 0.5000 0.2500 0.1250

Columns 5 through 8

0.0625 0.0312 0.0156 0.0078

Columns 9 through 11

0.0039 0.0020 0.0010

**%sum of 0 to 10**

>> sum10 = sum(series)

sum10 =

1.9990

>> n = 50

n =

50

**%computing series from 0 to 50**

>> series = r.^(0:n)

series =

Columns 1 through 4

1.0000 0.5000 0.2500 0.1250

Columns 5 through 8

0.0625 0.0312 0.0156 0.0078

Columns 9 through 12

0.0039 0.0020 0.0010 0.0005

Columns 13 through 16

0.0002 0.0001 0.0001 0.0000

Columns 17 through 20

0.0000 0.0000 0.0000 0.0000

Columns 21 through 24

0.0000 0.0000 0.0000 0.0000

Columns 25 through 28

0.0000 0.0000 0.0000 0.0000

Columns 29 through 32

0.0000 0.0000 0.0000 0.0000

Columns 33 through 36

0.0000 0.0000 0.0000 0.0000

Columns 37 through 40

0.0000 0.0000 0.0000 0.0000

Columns 41 through 44

0.0000 0.0000 0.0000 0.0000

Columns 45 through 48

0.0000 0.0000 0.0000 0.0000

Columns 49 through 51

0.0000 0.0000 0.0000

**%sum of 0 to 50**

>> sum50 = sum(series)

sum50 =

2.0000

**%computing series from 0 to 100**

>> series = r.^(0:100)

series =

Columns 1 through 3

1.0000 0.5000 0.2500

Columns 4 through 6

0.1250 0.0625 0.0312

Columns 7 through 9

0.0156 0.0078 0.0039

Columns 10 through 12

0.0020 0.0010 0.0005

Columns 13 through 15

0.0002 0.0001 0.0001

Columns 16 through 18

0.0000 0.0000 0.0000

Columns 19 through 21

0.0000 0.0000 0.0000

Columns 22 through 24

0.0000 0.0000 0.0000

Columns 25 through 27

0.0000 0.0000 0.0000

Columns 28 through 30

0.0000 0.0000 0.0000

Columns 31 through 33

0.0000 0.0000 0.0000

Columns 34 through 36

0.0000 0.0000 0.0000

Columns 37 through 39

0.0000 0.0000 0.0000

Columns 40 through 42

0.0000 0.0000 0.0000

Columns 43 through 45

0.0000 0.0000 0.0000

Columns 46 through 48

0.0000 0.0000 0.0000

Columns 49 through 51

0.0000 0.0000 0.0000

Columns 52 through 54

0.0000 0.0000 0.0000

Columns 55 through 57

0.0000 0.0000 0.0000

Columns 58 through 60

0.0000 0.0000 0.0000

Columns 61 through 63

0.0000 0.0000 0.0000

Columns 64 through 66

0.0000 0.0000 0.0000

Columns 67 through 69

0.0000 0.0000 0.0000

Columns 70 through 72

0.0000 0.0000 0.0000

Columns 73 through 75

0.0000 0.0000 0.0000

Columns 76 through 78

0.0000 0.0000 0.0000

Columns 79 through 81

0.0000 0.0000 0.0000

Columns 82 through 84

0.0000 0.0000 0.0000

Columns 85 through 87

0.0000 0.0000 0.0000

Columns 88 through 90

0.0000 0.0000 0.0000

Columns 91 through 93

0.0000 0.0000 0.0000

Columns 94 through 96

0.0000 0.0000 0.0000

Columns 97 through 99

0.0000 0.0000 0.0000

Columns 100 through 101

0.0000 0.0000

**%sum of 0 to 100**

>> sum100 = sum(series)

sum100 =

2

**%computing limit at r=0.5**

>> limit = 1/(1-r)

limit =

2

>>

Problem 3

**1**

A screenshot of a computer

Description automatically generated

**2**A screen shot of a graph

Description automatically generated

**Problem 4**

clear;

clc;

%defining variables

P\_outside = 1;%in atm

p = 1000; %density

g = 32.2; % acc due to gravity in ft/s^2

h = 7; %height

%conversion factors

ft\_to\_m = 0.3048;

yd\_to\_m = 0.9144;

atm\_to\_Pa = 1.013E5;

P\_abs = p \* (ft\_to\_m\*g) \* (yd\_to\_m\*h) + (atm\_to\_Pa\*P\_outside);

disp(['P\_abs is ' num2str(P\_abs)]);

**Output:**

P\_abs is 164121.0356

**Problem 5**

**1.**

a = input('Enter a number ');

if a>0

disp(num2str(log(a)));

else

disp('Error! Number should be positive')

end

**2.**

kilometres = input('Enter kilometres');

% Define cost rates

rate1 = 0.50; % Cost per km for first 100 km

rate2 = 0.30; % Cost per km for next 200 km

rate3 = 0.20; % Cost per km for km above 300

% Calculate cost

if kilometres <= 100

totalCost = kilometres \* rate1;

elseif kilometres <= 300

totalCost = 100 \* rate1 + (kilometres - 100) \* rate2;

else

totalCost = 100 \* rate1 + 200 \* rate2 + (kilometres - 300) \* rate3;

end

disp(['Total Cost is ' num2str(totalCost)]);

**Problem 6**

**1(a)**

clear;

clc;

product = 1;

for i = 2:10

if mod(i,2) == 0

product = product\*i;

end

end

disp(num2str(product));

**1(b)**

clear;

clc;

product = 1;

i = 2; % Initialize counter for the while loop

while i <= 10

if mod(i,2) == 0

product = product \* i;

end

i = i + 1; % Increment counter

end

disp(num2str(product));

**4.**

clear;

clc;

% Prompt the user for a non-negative integer

n = input('Enter a non-negative integer: ');

% Check if the input is non-negative and an integer

if n < 0

disp('Error: Please enter a non-negative integer.');

else

factorial = 1;

% Compute the factorial using a for loop

for i = 1:n

factorial = factorial \* i;

end

% Display the result

disp(['Factorial of ',num2str(n),' is ', num2str(factorial)]);

end

**A close-up of a paper

Description automatically generated**

**A white lined paper with writing on it

Description automatically generated**

**A close-up of a paper

Description automatically generated**