**CTS2 LINA MATLAB Gorshkov Anton**

**Assignment 2**

Here is a code of the function:

% Write a MATLAB-function that visualises all different n solutions to the equation z.^n = q, where z and q are complex numbers and n is a positive integer. The function you write % needs to ask for n and q as input parameters and display all solutions on a circle line as output values; choose an appropriate scaling of both axis. Furthermore your function

% needs to list all solutions to z.^n = q in their algebraic representation, giving their real and imaginary parts. Test your generated MATLAB file with the following equation

% z.^7 = +2j, where j is the imaginary unit, i.e. j.^2 = -1.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [] = assignment2(varargin)

%%%%% Set default values in case when no parameters are passed %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if nargin < 2

n = 7; % Set default values of equation

q = 2i; % Set default values of equation

else

n = varargin{1}; % Set values of equation from the function parameters

q = varargin{2}; % Set values of equation from the function parameters

end

%%%%% First solution: Manual one %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% z = zeros(1, n);

% z(1) = q.^(1/n);

% for i = 1:n - 1

% z(i + 1) = z(1) \* exp(1i \* 2 \* pi \* i / n);

% disp(z(i + 1));

% disp((z(i + 1).^n));

% end

%%%%% Second solution: More general: Applicable for the (system of )(in)equalities%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

syms z % Initialize variable

eqn = z^n==q; % Initialize equation

sol = solve(eqn,z); % Initialize solutions

fprintf('%f + %fi\n', [real(sol), imag(sol)].'); % Print solutions to Command Window

plot(sol, '\*'); % Draw the solutions

hold on; % retain plots in the current axes so that new plots added to the axes do not delete existing plots.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Draw a circle line:

radius = abs(q)^(1/n); % Initialize a radius of the circle.

angle = 0: .01: 2 \* pi; % Initialize a array of angles for drawing the circle.

x = radius \* sin (angle); % Initialize a array of x-coordinates for drawing the circle.

y = radius \* cos (angle); % Initialize a array of y-coordinates for drawing the circle.

plot (x, y); % Draw all the points from the (x,y)-array.

axis equal; % use equal data unit lengths along each axis.

hold off; % set the hold state to off so that new plots added to the axes clear existing plots and reset all axes properties.

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Using MATLAB, the solutions of this equation might be found using different approaches:

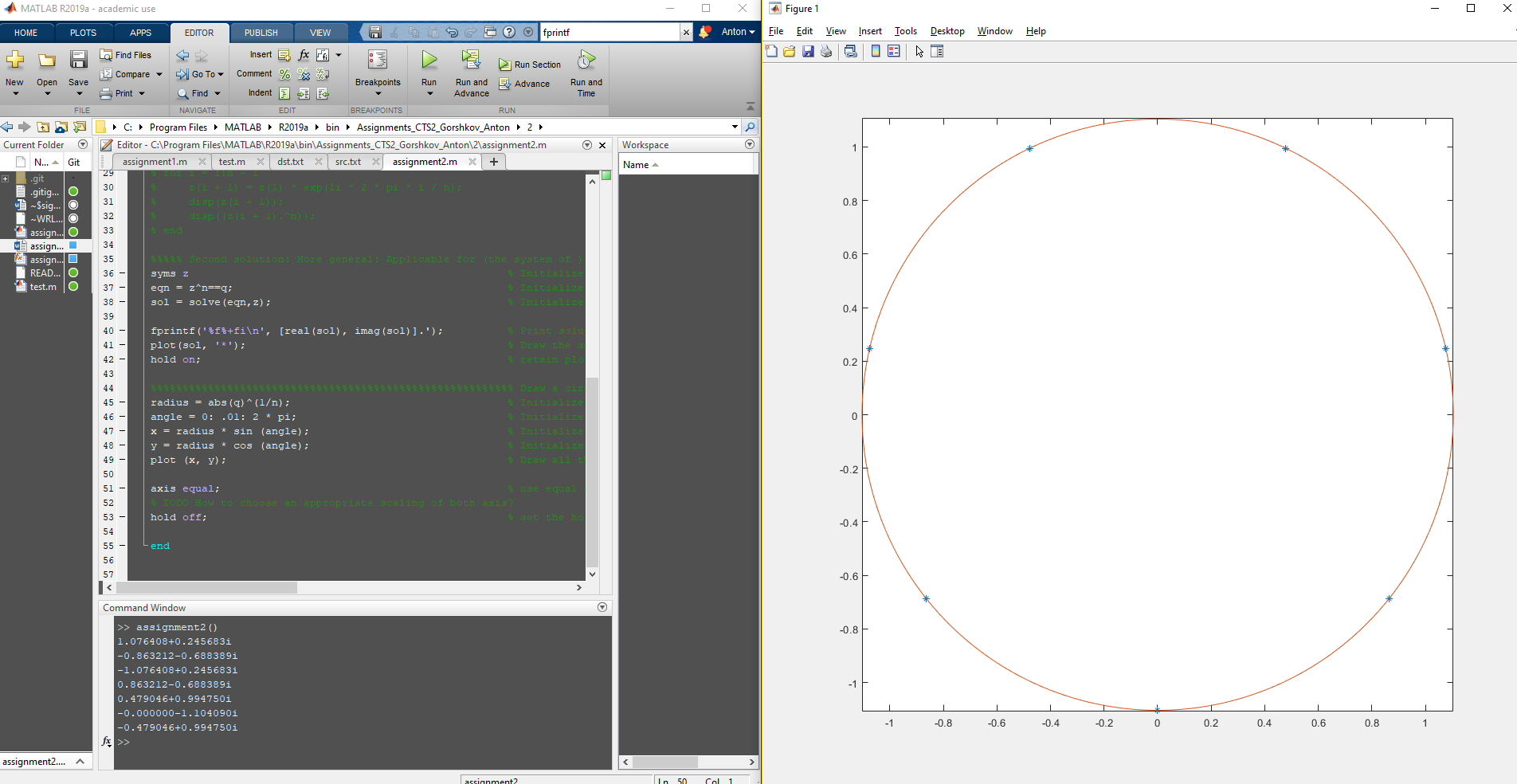
* using “manual method” (commented, is not used in my case)
* using much more general, powerful mechanism MATLAB solving system of equations/inequalities (preferable from my point of view)

It is possible to launch function in two modes:

* using particular values n, q of the equation.
* using default values n, q of the equation (7 and 2 respectively).

Let us try to execute the function without parameters (with default values).

After an execution of the function, geometric representation of complex solutions appears in the circle line.



Algebraic representation of the solutions appears as an output in the **Command Window** as follows:

1.076408+0.245683i

-0.863212-0.688389i

-1.076408+0.245683i

0.863212-0.688389i

0.479046+0.994750i

-0.000000-1.104090i

-0.479046+0.994750i

