***MATLAB, Lab 4 – Individual work***

The function RegPoly draws the polygon centered at point (x0, y0). It is circumscribed by a circle of radius r, and contains n sides. The code of RegPoly is presented below

|  |
| --- |
| function RegPoly (xo, yo, r, n)‏  *% Draws a regular polygon*  *% xo, yo - coordinates of the centre of the polygon*  *% r - the radius of the circle that circumscribes the polygon*  *% n - the number of the polygon sides*  alpha = linspace (0, 2\*pi, n+1);  x = xo + r \* cos(alpha); *% conversion of polar to Cartesian*  y = yo + r \* sin(alpha); *% as above*  d = distNp (x, y); *% we use the function “distNp”*  *% plot the polygon:*  plot (x, y); grid on;  axis equal;  legend (['d = ', num2str(d)]); |

Please note, that the function calls distNp described on the lecture, which has to be placed in the same folder as RegPoly. With the use of these functions work on the tasks below:

1. Is there any other way of computing x and y from polar coordinates? Check function *pol2cart* in help, and try to apply it in the code above. Paste the modified code below.

|  |
| --- |
| Code: |
| function RegPoly (xo, yo, r, n)  % Draws a regular polygon  % xo, yo - coordinates of the centre of the polygon  % r - the radius of the circle that circumscribes the polygon  % n - the number of the polygon sides    alpha = linspace (0, 2\*pi, n+1);  [x,y] = pol2cart(alpha,r); % conver of polar to Cartesian coordinates  d = distNp (x, y); % we use the function “distNp”    % plot the polygon:  plot (x, y); grid on;  axis equal;  legend (['d = ', num2str(d)]); |

1. How to compute the value of π with the function RegPoly? Show the result of example calculation.

|  |
| --- |
| Code: |
| function RegPoly (xo, yo, r, n)  % Draws a regular polygon  % xo, yo - coordinates of the centre of the polygon  % r - the radius of the circle that circumscribes the polygon  % n - the number of the polygon sides    alpha = linspace (0, 2\*pi, n+1);  x = xo + r \* cos(alpha); % conversion of polar to Cartesian  y = yo + r \* sin(alpha); % as above    d = distNp (x, y); % we use the function “distNp”    **AproxPi = d/(2\*r) % we aproximate the number pi by the división of**  **% the perimeter of the polygon and the diameter**  **% of the circunference**    % plot the polygon:  plot (x, y); grid on;  axis equal;  legend (['d = ', num2str(d)]); |
| Result: |
| RegPoly(0,0,10,7);  The result of this calculation is 3.0372  I use here the Arquimedes method to approximate the value of pi number. I only use the polygons inscribed in a circumference, not the circumscribed polygons. |

1. Write the function that computes the area of the polygon. An area of triangle can be calculated from formula

|  |
| --- |
| Code: |
| function a = area (xo, yo, r, n)  %area Calculate the area of the polygon    alpha = linspace (0, 2\*pi, n+1);  x = xo + r \* cos(alpha); % conversion of polar to Cartesian  y = yo + r \* sin(alpha); % as above    d = distNp (x, y)    gamma = 2\*pi/n;  lado = d/n;  apotema = (lado/2)/(tan(gamma/2));    if (n == 3)  a = 0.5\*sin(gamma)\*(lado^2); %we use the formula given in the document  elseif (n>3)  a =(d\*apotema)/2; % we use the formula perimeter per apothem  % divide by 2  end |
|  |

1. How to compute the value of π with the use of function developed in point 3? Show the result of example calculation.

|  |
| --- |
| Code: |
| function a = area (xo, yo, r, n)  %area Calculate the area of the polygon    alpha = linspace (0, 2\*pi, n+1);  x = xo + r \* cos(alpha); % conversion of polar to Cartesian  y = yo + r \* sin(alpha); % as above    d = distNp (x, y)    gamma = 2\*pi/n;  lado = d/n;  apotema = (lado/2)/(tan(gamma/2));    if (n == 3)  a = 0.5\*sin(gamma)\*(lado^2); %we use the formula given in the document  AproxPi = a/(r^2)  elseif (n>3)  a =(d\*apotema)/2; % we use the formula perimeter per apothem  AproxPi = a/(r^2) % divide by 2    end |
| Result: |
| a = area (0,0,15,13);  The result of this calculation is 3.0207  What I do here is, first of all, I use the area function to compute the area of the polygon inscribed in a circumference and then, as the area of a circle is known (A = π\*r2), I divide the area of the polygon by r2.  This is one method to approximate the number pi by the approximation of the area of a circle. |