Pracownia nr 4

21.11.2009

1 Zadanie : Pole koła

Obliczymy pole koła w inny sposób niż w poprzednim zadaniu. Rozważmy n punktów na okręgu o promieniu 1. Łącząc punkty łamaną otrzymamy pewien wielokąt. Im większa wartość n, tym więcej punktów i tym dokładniej wielokąt wypełnia koło ograniczone naszym okręgiem. Zatem pole wielokąta jest przybliżeniem pola koła.

1.1 Konstrukcja wielokąta

1. Jak można opisać punkt leżący na okręgu? Jego współędne x, y spełniają równania:

   $$x = r \cdot cos(\alpha) \quad y = r \cdot sin(\alpha)$$

   dla pewnego kąta α o wartości od 0 do 360 stopni i promienia r (w naszym przypadku $r = 1$).

2. Stworzymy zbiór n punktów leżących na okręgu. W tym celu utworzymy ciąg liczb, które będą odpowiadać różnym wartościom kąta α dla różnych punktów. Czyli $a_1 = \alpha_1, a_2 = \alpha_2, ...a_n = \alpha_n$.

3. Chcemy aby punkty na okręgu były położone równomiernie. Dlatego różnice pomiędzy kolejnymi wyrazmi ciągu (a_n) muszą być równe. Czyli będzie to ciąg arytmetyczny. Do stworzenia ciągu arytmetycznego możemy użyć funkcji napisanej podczas pierwszej pracowni.

4. Aby użyć funkcji **f** musimy znać jej parametry: numer elementu **n**, różnicę ciągu **r** oraz pierwszy wyraz **a1**.

5. Chcemy, aby $a_n = 360$ i $a_1 = 0$. Jak przy danym n wyznać r? Prosto:

   $$a_n = a_1 + (n-1) \cdot r \quad \text{czyli} \quad r = \frac{a_n - a_1}{n-1}$$

6. Nasz program będzie miał postać funkcji, o nazwie **pole_kola** i parametrze **n**:

   ```
   function p = pole_kola(n)
   ```

7. Na początku musimy wyznaczyć wartość różnicy ciągu zgodnie z wcześniejszym wzorem:

   ```
   r = (360-0)/(n-1);
   ```

8. Następnie używamy funkcji **f** do utworzenia tablicy **a** zawierającej pierwsze n wyrazów ciągu arytmetycznego:

   ```
   a = f(1:n,r,0);
   ```

9. Używając tablicy **a** możemy utworzyć ciągi liczb odpowiadające współrzędnym x, y kolejnych punktów na okręgu, zapiszemy je odpowiednio w tablicach **x** i **y**:

```
x = cosd(a);
y = sind(a);
```

10. Możemy teraz narysować punktyi połączone odcinkami:

    ```
    plot(x,y,'+-')
    ```

11. Zakończy funkcję słowem **end** i zapiszmy plik zawierający funckję. Zobaczmy jak działa nasza nowa funkcja:

    ```
    pole_kola(11)
    ```

1.2 Obliczenie pola wielokąta

Chcemy teraz uzupełnić naszą funkcję tak, aby obliczać pole wielokąta wyznaczonego przez punkty z okręgu. Zauważmy, że pole wielokąta jest sumą pól $n-1$ trójkątów o dwóch wierzchołkach w kolejnych punktach z okręgu i trzecim w środku okręgu.

Znamy wartość kąta leżącego przy wierzchołku w środku okręgu - to wyliczone przez nas wcześniej r. Z konstrukcji ciągu (a_n) wynika równość wszystkich tych kątów. Z kolei długość ramion tego kąta jest równa 1 ponieważ jest ona równa promieniowi okręgu. Zatem pola wszystkich trójkątów są równe. Aby policzyć pole wielokąta musimy tylko pomnożyć pole trójkąta przez liczbę trójkątow.

1. Pole trójkąta możemy policzyć ze wzoru:

   $$p = \frac{1}{2} \cdot ab \cdot sin(\alpha)$$

   dla a i b oznaczających długości boków wychodzących z wierzchołka o kącie α (w naszym przypadku $a = 1, b = 1$). Zatem pole wielokąta jest równe:

   $$p = (n-1) \cdot \frac{1}{2} \cdot sin(r)$$

2. Uzupełniamy funkcję wpisując przed końcem linię:

   ```
   p = (n - 1)*(1/2)*sind(r);
   ```

3. Przyjrzyjmy się jak wyglądają wyniki dla różnych wartości **n**:

   ```
   >> for n = 10:10:100
    pole_kola(n)
    drawnow
    pause
   end
   ```

2 Zadanie : Płatek Kocha

Zajmiemy się narysowaniem prostego fraktala, tzw. płatka Kocha (od nazwiska szwedzkiego matematyka von Kocha). Zrobimy to w dwóch krokach: po pierwsze skonstruujemy łamaną, która będzie n-tym przybliżeniem tak zwanej krzywej Kocha, po drugie używając tej krzywej narysujemy figurę geometryczną nazywaną płatkiem Kocha.

2.1 Konstrukcja krzywej Kocha

Krzywa Kocha powstaje poprzez zastąpienie odcinka łamaną złożoną z czterech odcinków, tak jak na rysunku. Następnie biorąc każdy z czterech odcinków za nowy odcinek początkowy, powtarzamy procedurę dla każdego z nich. Jeżeli będziemy to powtarzać w nieskończoność krzywa otrzymana w granicy nazywa się *krzywą Kocha*. My ograniczymy się do n-tego przybliżenia tej krzywej, a więc skonstruujemy łamaną powtarzając opisaną procedurę n razy, dla dowolnego n.

1. Załóżmy że wiemy jak wyznaczyć współrzędne punktów $(x_2, y_2), (x_3, y_3), (x_4, y_4)$ z rysunku, znając współrzędne (x_1, y_1) i (x_5, y_5). Z twierdzenie Pitagorasa możemy wyliczyć, że:

$$x_2 = \frac{2x_1 + x_5}{3}, y_2 = \frac{2y_1 + y_5}{3},$$

$$x_3 = \frac{x_1 + x_5}{2} - \frac{y_5 - y_1}{2\sqrt{3}}, y_3 = \frac{y_1 + y_5}{2} + \frac{x_5 - x_1}{2\sqrt{3}},$$

$$x_4 = \frac{2x_5 + x_1}{3}, y_4 = \frac{2y_5 + y_1}{3}$$

2. Ponieważ n ma być dowolne więc musimy stworzyć funkcję, której parametrem będzie **n**. Ponieważ funkcja będzie powtarzać te same operacje, odwołując się sama do siebie więc musi to być funkcja rekurencyjna.

3. Utwórzmy funkcję o nazwie **koch**. Będzie ona rysować łamaną dla danego **n**. Ponadto paramerami będą **x1,y1,x5,y5**.

```
function koch(x1,y1,x5,y5,n)
```

4. Założmy że **n** równe jeden oznacza pierwotny odcinek przed podzieleniem na części. A zatem możemy utworzyć warunek:

```
if ( n == 1 )
 hold on
 axis equal
 plot([x1;x5],[y1;y5],'-')
```

   Oznaczający, że dla **n** równego jeden tylko rysujemy odcinek o końcach (x_1, y_1) i (x_5, y_5).

5. W przeciwnym razie używamy znanych współrzędnych i rekurencyjnie wywołujemy naszą funkcję dla każdego z odcinków powstałych z podziału pierwotnego odcinka.

```
else
 x2=(2*x1 + x5)/3;
 y2=(2*y1 + y5)/3;
 x3=(x1 + x5)/2 - (y5 - y1)/(2*sqrt(3));
 y3=(y1 + y5)/2 + (x5 - x1)/(2*sqrt(3));
 x4=(2*x5 + x1)/3;
 y4=(2*y5 + y1)/3;
 koch(x1,y1,x2,y2,n-1)
 koch(x2,y2,x3,y3,n-1)
 koch(x3,y3,x4,y4,n-1)
 koch(x4,y4,x5,y5,n-1)
end
```

6. Pozostaje jeszcze zakończyć funkcję słowem kluczowym **end** i zobaczyć jak działa:

   ```
   >> koch(0,0,1,0,4)
   ```

   W wyniku otrzymujemy łamaną, powstałą z odcinka $(0,1)$ leżącego na osi x układu współrzędnych, która jest czwartym przybliżeniem krzywej Kocha.

2.2 Płatek Kocha

Płatek Kocha to figura, którą otrzymamy biorąc trójkąt równoboczny i każdy z jego boków przekształcając w krzywą Kocha. Umiemy już narysować przybliżenie krzywej Kocha więc narysowanie płatka będzie łatwe.

1. Zauważmy, że aby narysować płatek wystarczy trzy razy użyć naszej fukcji **koch** dla odpowiednio określonych parametrów - współrzędnych końców odcinków będących bokami trójkąta równobocznego.

2. Niech pierwszy z boków będzie identyczny z użytym do rysowania krzywej w pierwszym kroku zadania. Pozostałe wierzchołki dwóch brakujących boków łatwo wyznaczyć. Musimy tylko zadbać o prawidłową kolejność ich podawania:

   ```
   >> koch(0,0,1,0,4)
   >> koch(1,0,1/2,-sqrt(3)/2,4)
   >> koch(1/2,-sqrt(3)/2,0,0,4)
   ```