
PPPD - Lab. 03

Copyright ©2022 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 03, grupa B, 2022/2023, autor: Michał Dębski

Uwaga: w rozwiązaniu zadania nie można używać list i definiować własnych funkcji (oprócz main).

Temat: Zbiory Julii

Treść zadania

Niech (x, y) będzie punktem na płaszczyźnie, a cx i cy liczbami rzeczywistymi. Rozważmy przekształcenie fc , które zastępuje (x, y) punktem $(x^2 - y^2 + cx, 2xy + cy)$ (jest to przekształcenie $fc(z) = z^2 + c$ na płaszczyźnie zespolonej). Przez iterowanie tego przekształcenia rozumiemy wielokrotne zastosowanie go, gdzie za każdym razem jako (x, y) podstawiamy punkt otrzymany w wyniku poprzedniej iteracji. Przez trajektorię punktu (x, y) rozumiemy ciąg punktów otrzymanych podczas iterowania przekształcenia fc , tzn. $fc((x, y)), fc(fc((x, y))), fc(fc(fc((x, y))))$, ...

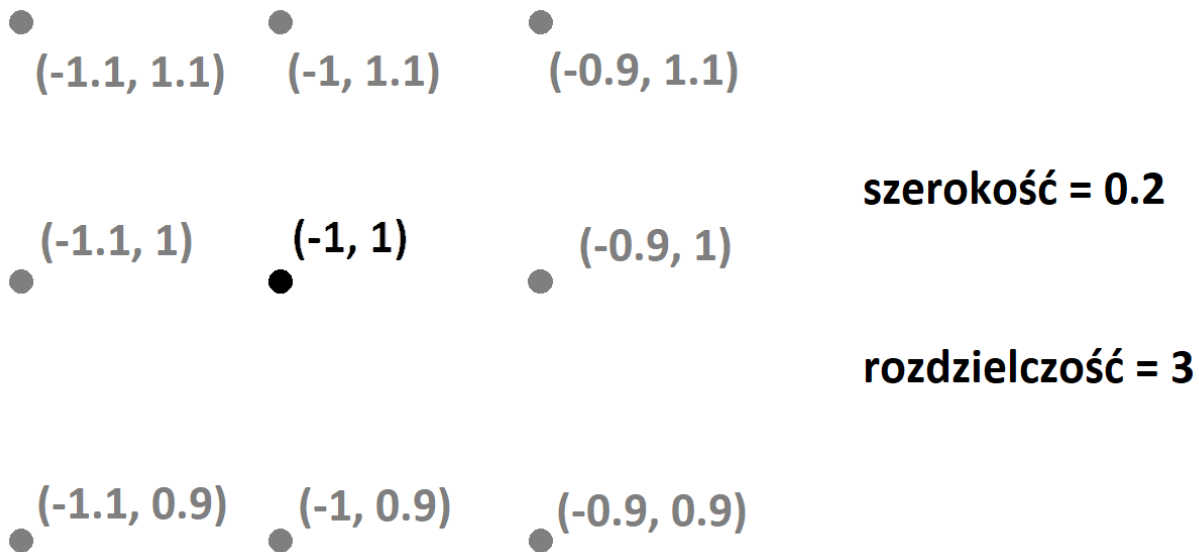
Celem zadania jest napisanie programu pozwalającego na badanie, jak zachowują się trajektorie punktów na pewnym fragmencie płaszczyzny. Należy rozważać tylko określoną liczbę punktów na każdej trajektorii i przerwać obliczanie tej trajektorii kiedy tylko oddali się ona o więcej niż 2 od początku układu współrzędnych.

Program powinien wczytać następujące dane:

- parametry cx i cy (liczby rzeczywiste)
- punkt początkowy (x_0, y_0) (liczby rzeczywiste),
- szerokość badanego obszaru (liczba rzeczywista),
- rozdzielczość wizualizacji (nieparzysta liczba naturalna)
- maksymalna liczba iteracji dla jednego punktu, tzn. maksymalna liczba punktów na każdej trajektorii (liczba całkowita).

W przypadku gdy dane są niepoprawne, program powinien wyrzucić wyjątek. W przeciwnym przypadku powinien wypisać na konsolę informacje:

- trajektorię punktu (x_0, y_0) do momentu, w którym punkt będzie w odległości przynajmniej 2 od środka układu współrzędnych lub osiągnięta zostanie maksymalna liczba iteracji
- tekstowy “rysunek” przedstawiający punkty wokół (x_0, y_0) , których trajektoria ani razu nie oddaliła się powyżej 2 od środka układu współrzędnych; punkty te powinny być oznaczone symbolem @. Środek rysunku powinien odpowiadać punktowi (x_0, y_0) , a pozostałe punkty należy dobrać tak, żeby w sumie było ich rozdzielczość^2 i były równo rozmieszczone w kwadracie o szerokości szerokość , patrz rysunek poniżej (wykonanie tej części zadania wymaga obliczenia trajektorii osobno dla każdego z rozważanych punktów)
- całkowitą liczbę iteracji przekształcenia, wykonanych podczas działania programu; liczba ta powinna uwzględniać iteracje wykonane w obu wcześniejszych etapach



Rysunek 1: Punkty, których trajektorie należy sprawdzić

Szczegóły techniczne

- Program powinien reagować na następujące błędy w danych: niedodatnia szerokość lub rozdzielność, ujemna maksymalna liczba iteracji, parzysta rozdzielność
- Program powinien przerywać iterowanie przekształcenia za pierwszym razem, kiedy punkt oddali się o więcej niż 2 od środka układu współrzędnych (żeby nie wykonywać zbędnych obliczeń)
- Na rysunku współrzędna x jest pozioma i rośnie w prawo; współrzędna y jest pionowa i rośnie w górę
- W drugiej części zadania trajektoria punktu (x_0, y_0) może być liczona od nowa

Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- Wczytanie danych i wypisanie komunikatu, jeśli są one niepoprawne - 1p
- Wypisanie trajektorii punktu początkowego - 3p
- Wydrukowanie rysunku zgodnego z wymaganiami - 4p
- Wypisanie liczby wykonanych iteracji przekształcenia - 2p

Uwagi

- Jeśli rozwiązanie nie spełnia postawionych wymagań (korzysta z list albo dodatkowo zdefiniowanych funkcji), zadanie jest oceniane na 0 punktów.
- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2p

Przykłady interakcji użytkownika z programem

Wydruk dla prawidłowych danych:

```
Podaj cx: -0.123
Podaj cy: 0.745
Podaj x0: 0
Podaj y0: 0
```

```

      Q
      QQQ
QQ      QQQ
QQ      Q      QQQ
  Q      QQQ      QQQQQQ
QQQQQQQQQQ      QQQQ
Q      QQQQQQQQ      QQQQ
Q      QQQQQQQQQQQ
  QQQQQQQQQQ
      QQQQQQ      QQQQ      Q
  Q      Q      QQQQQQQ      Q
      QQQQQQQQ      QQ
      QQQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQQQQQQQQQ
      QQQ      QQ
      Q      QQQQQQQ      Q      Q      Q
      Q      QQQQ      QQQQQ
      QQQQQQQQQQ
      QQQQQQQQQQQQ

```

```

      00000 00000000 0
      0000  000000000
000000  000  0
      000    0    00
      000      00
000
0

```

Program wykonał w sumie 10692 iteracji przekształcenia $(x,y) \leftarrow -(x^2 - y^2 + -0.123, 2xy + 0.745)$

Process finished with exit code 0

Wydruk dla błędnych danych

```

Podaj cx: 0
Podaj cy: 1
Podaj x0: -1
Podaj y0: 2
Podaj szerokość badanego obszaru: 0
Podaj rozdzielczosc: 3
Podaj maksymalną liczbę iteracji dla jednego punktu: 100
Traceback (most recent call last):
  File "2022-IAD-03B.py", line 50, in <module>
    main()
  File "2022-IAD-03B.py", line 11, in main
    raise ValueError("Błędne dane!")
ValueError: Błędne dane!

```

Wydruk dla szczególnych danych:

```

Podaj cx: 1
Podaj cy: 1
Podaj x0: 1
Podaj y0: 1
Podaj szerokość badanego obszaru: 7
Podaj rozdzielczosc: 47
Podaj maksymalną liczbę iteracji dla jednego punktu: 0
Trajektoria punktu (1.0, 1.0):

```

Program wykonał w sumie 0 iteracji przekształcenia $(x,y) \leftarrow (x^2 - y^2 + 1.0, 2xy + 1.0)$