

Ein iteriertes System von Funktionen

ifs.py

Ein iteriertes Funktionensystem besteht aus einer Anzahl von Transformationen in der Ebene. Man wählt einen beliebigen Ausgangspunkt (zumeist den Ursprung) und wendet dann diese Transformationen an. Man zeichnet die Punkte und wählt zufällig einen von ihnen als neuen Ausgangspunkt. Das Ergebnis ist eine Figur, die fraktale Eigenschaften aufweist (Selbstähnlichkeit).

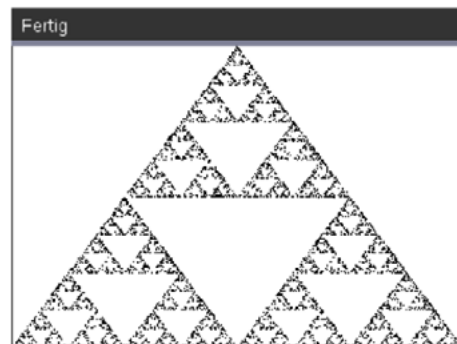
Im nächsten Programm werden die Transformationen in der komplexen Ebene beschrieben.

1. Streckung: $z \mapsto 0,5z$
2. Streckung und Schiebung: $z \mapsto 0,5z + 0,5$
1. Streckung: $z \mapsto 0,5z + (0,25 + 0,5i)$

```

1.8 1.9 1.10 *PyKurz RAD
ifs.py erfolgreich gespeichert
import tiplotlib as plt
from random import choice
n=int(input("Wieviel Punkte? "))
plt.cls();plt.window(0,1,0,1)
z=0+0j;k=0
while k<=n:
    w=[.5*z,.5*z+.5,.5*z+.25+.5j]
    for p in w:
        plt.plot(p.real,p.imag,".")
    z=choice(w);k=k+1
plt.show_plot

```



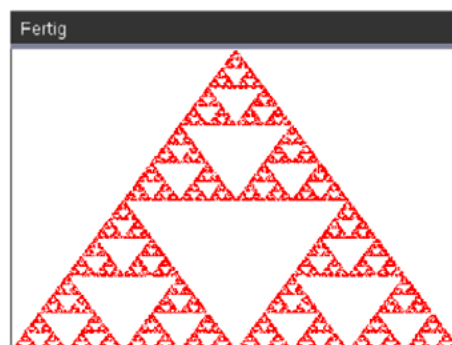
Der Benutzer wird gefragt, wie viele Punkte er zeichnen lassen will. Dann entsteht die Figur. Das Bild oben rechts besteht aus 2000 Bildpunkten.

Dieses System wird auch gerne als „Chaosspiel“ bezeichnet. Kannst du auch Drehungen implementieren? Du kannst die Punkte auch einfärben.

Informiere dich im Internet über die iterierten Funktionensysteme.

https://de.wikipedia.org/wiki/Iteriertes_Funktionensystem

Bei der Suche nach „iterated function systems“ kannst du sehr schöne Grafiken finden.



Diese Grafik besteht aus 5000 Punkten.