Juliamengen und Apfelmännchen, Iterationen, Level 5 bis 7

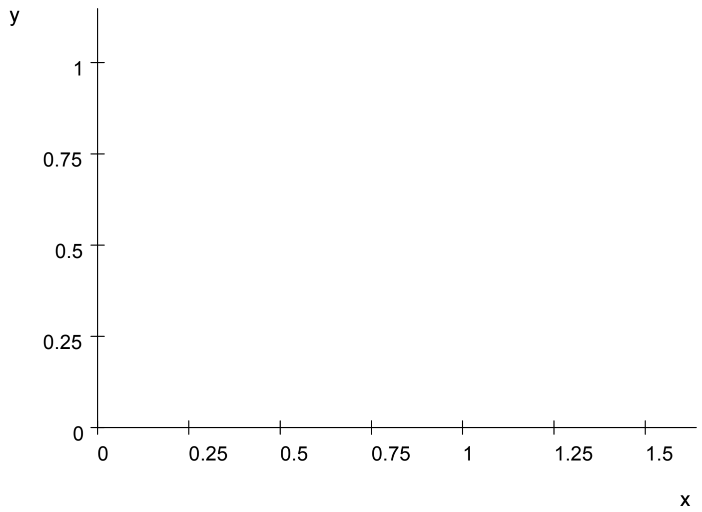
Mathematik mit MuPAD 2.5, Prof. Dr. Dörte Haftendorn 13.06.03 Version vom 16.06.03

```
Inhalt....: Juliamengen und Apfelmännchen, komplexe Iterationen
Kategorie.: Lernblatt
Mathematik: Komplexe Zahlen, Fraktale Geometrie,Juliamengen und Apfelmännchen
MuPAD.....: 2.5.0
Datum.....: 2003-06-16
Autoren...: Dörte Haftendorn <Haftendorn@uni-lueneburg.de>
Funktionen: Re, Im,  plot, plot::Point, map, f@@i
```

Level 1 Komplexe Zahlen in der gaußschen Zahlenebene ---->komplex.mnb
Level 2 Komplexe Zahlen in der Polar-Darstellung mit der Eulerschen Formel ---->komplex.mnb
Level 3 Lineare Iterationen mit Fixpunkt ---->komplex.mnb
Level 4 Quadratische Iterrationen mit f(z)=z^2 als Vorbereitung für Apfelmännchen ---->komplex.mnb
Level 5 Quadratische Iterrationen mit f(z)=z^2 +c Juliamengen
Level 6 Quadratische Iterrationen mit f(z)=z^2 +c Apfelmännchen
Level 7 Iterrationen mit f(x)=x^2 +c mit c reell, Zusammenhang von Apfelmännchen und Feigenbaum

Aus Level 1 und 2 zwei komfortable Prozeduren:

```
• zpkt:=proc(z) begin   /* Erzeugt den Punkt von z in der Gauß-Ebene */
          return(plot::Point([Re(float(z)),Im(float(z))]))
      end_proc:
  zbw:=proc(r,phi) begin  /* Polardarstellung aus Radius und Winkel im Gradmaß */
        return(r*E^(I*PI/180*phi))
      end_proc:
• plot(zpkt(0),zpkt(zbw(2,35)),zpkt(1.2+I))
```



Obiges ausführlich in --->Datei komplex.mnb
##

Level 5 Iterationen zu Juliamengen

- `delete z,c :`
  `f:=z->z^2+c`

  $$z \to z^2 + c$$

- `fix:=solve(f(z)=z,z)`

  $$\left\{\frac{1}{2} - \frac{\sqrt{1-4\cdot c}}{2}, \frac{\sqrt{1-4\cdot c}}{2} + \frac{1}{2}\right\}$$

- `fixSteigung:=f'(abs(fix))`

  $$\left\{2\cdot\left|\frac{1}{2} - \frac{\sqrt{1-4\cdot c}}{2}\right|, 2\cdot\left|\frac{\sqrt{1-4\cdot c}}{2} + \frac{1}{2}\right|\right\}$$

Außer für c=1/4 gibt es zwei Fixpunkte.

- `c:=1/4+0.1:fix; fixSteigung`

  $$\{0.5 - 0.316227766\cdot i, 0.5 + 0.316227766\cdot i\}$$

  $$\{1.183215957\}$$

Für reelle c>1/4 sind beide abstoßend.
Es gibt Punkte c, bei denen ein Fixpunkt anziehend, einer abstoßend ist.

- `c:=1/4-0.1:fix; fixSteigung`

  $\{0.183772234,\ 0.816227766\}$

  $\{0.367544468,\ 1.632455532\}$

- `c:=0.1+0.5*I:fix; fixSteigung`

  $\{1.07966168 - 0.4312860567 \cdot i,\ -0.07966167954 + 0.4312860567 \cdot i\}$

  $\{0.8771628034,\ 2.325232896\}$

-

Leider wird die Reihenfolge bei diesen Auswertungen nicht beibehalten.

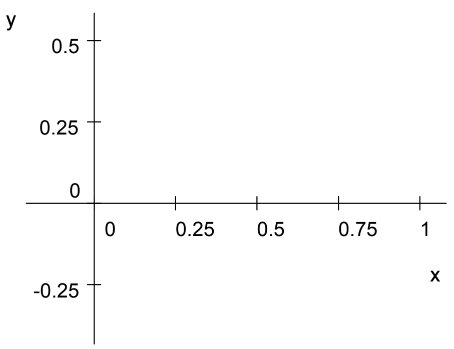
- `f'(-0.08-0.43*I) /* dies ist der anziehende Fixpunkt */`

  $-0.16 - 0.86 \cdot i$

-

Betrachtung von Iterations-Folgen von fc(z)=z^2 +c

- ```
  c:=0.1+0.5*I: z0:=0.4-0.4*I:
  liste:=(f@@i)(z0) $ i=1..100:/*Orbit von z0*/
  fixpkt:=op(map(fix,zpkt)):
  (fixpkt[1])::Color:=[1,0,1]:(fixpkt[2])::Color:=[0,0,1]:
  pkte:=map(liste,zpkt):cpkt:=zpkt(c):cpkt::Color:=[0,1,0]:
  plot(zpkt(0),pkte,cpkt,zpkt(z0),fixpkt,Scaling=Constrained)
  ```



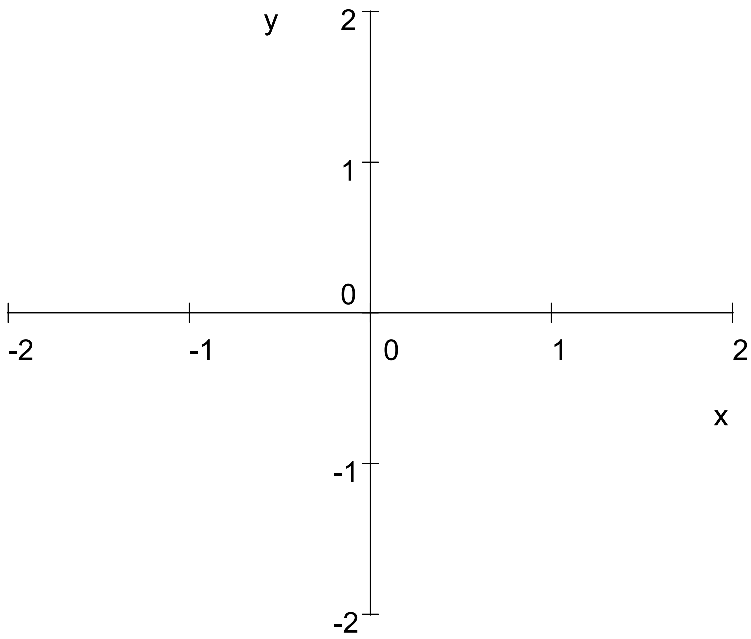
In Grün sieht man c, in Blau und Lila den anziehenden und den abstoßenden Fixpunkt
  ```

Definitionen und Sätze
Die Menge aller z0, für die die Folge fc(z0) beschränkt bleibt, heißt
Gefangenenmenge von c, oder ausgefüllte Juliamenge von c, kurz Gc.
Die Menge aller z0, für die die Folge fc(z0) gegen Unendlich strebt, heißt
Fluchtmenge, Einzugsbereich von Unendlich, kurz Uc(infinity), Uc=C/Gc
Der Rand von Gc heißt Juliamenge von c, kurz Jc.
Satz:
Gc liegt sicher innerhalb des Kreises um 0 mit Flucht-Radius max(abs(c),2).
Mit dieser Grundinformation werden im Folgenden zu fest gwähltem c für viele Punkte aus dem Quadrat von -2 bis 2 auf
beiden Achsen die nächsten Folgenglieder berechnet. Sowie ihr Betrag den Fluchtradius überschreitet,
wird der zugehörige Startpunkt z0 mit einer Farbe gefärbt, die aus dem momentanen Folgenindex berechnet wird,
der Liste der bisher untersuchten Punkte zugefügt.
Wenn nach kmax Schritten der Eintritt in den Fluchtbereich noch nicht erfolgt ist, wird z0 schwarz gefärbt.
Dies nennt man "escape-time-Algorithmus" mit Niveaulinien.

- `kmax:=10:`
- `farbpkt:=proc(z0,c)`

```
          local rr,rf,i,zz,z0pkt;
          begin
           rf:=float(max(abs(c),2));rr:=float(abs(z0));i:=0;zz:=z0;
           z0pkt:=zpkt(z0);
           z0pkt::Color:=[(kmax-i)/(kmax),1/(kmax+1)*(kmax-(1+(-1)^i/2)),0];
           /*  das waren die Vorbelegungen  */
          repeat
             if rr>rf then  /* die Folge ist in das Fluchtgebiet eingetreten */
               z0pkt::Color:=[(kmax-i)/(kmax),1/(kmax+1)*(kmax-(1+(-1)^i/2)),0];
               return(z0pkt);
             else
               i:=i+1; zz:=zz^2+c;  /* weiteres Folgenglied */
               rr:=float(abs(zz));
             end_if;
           until i=kmax+1 end_repeat;
           z0pkt::Color:=[0,0,0]; /* der Rest schwarz */
           return(z0pkt);
         end_proc:
```

- ```
  liste:=op({}):fein:=80:c:=-1:
  liste:=((liste,farbpkt(-2+4*i/fein+(-2+4*k/fein)*I,-1) $ i=0..fein )
                   $ k=0..fein): plot(liste):
  ```



-

Schwarz ist also eine grobe Nährung für die Gefangenmenge von c= -1.
Ihr Rand ist also eine grobe Nährung für die Juliamenge von c= -1.
Es gibt zu jedem c eine Juliamenge. Variiert und gezeichnet wird der Startpunkt z0, gefärbt entsprechend seiner Folge unter dem festen c.
Soll die Näherung viel besser werden, muss man zu rechnernaher Programmierung greifen.
(Winfract, Dufner: Julia.exe, apfel.exe von Ha u.a.)

-

Sätze: Die Juliamengen Jc sind punktsymmetrisch zum Ursprung.
Die Juliamenge zum konjugiert-komplexen c erhält man sowohl durch
Spiegeln an der reellen Achse als auch an der Im-Achse.
Juliamengen sind invariant unter fc.
fc ist eine "chaotische" Funktion in folgendem Sinne:
Das Bild jeder noch so kleinen Umgebung Uz eines Punktes z aus Jc ist
nach genügend vielen Iterationen aufJc verstreut.
  ```

Genauer: In jeder anderen beliebig kleinen Umgebung irgendeines Punktes von
Jc finden sich Punkte die aus Uz stammen.

•

##

Level 6 Iterationen zum Apfelmännchen

Es geht wieder um dieselbe Iteration wie eben.

```
delete z,c :
f:=z->z^2+c
```

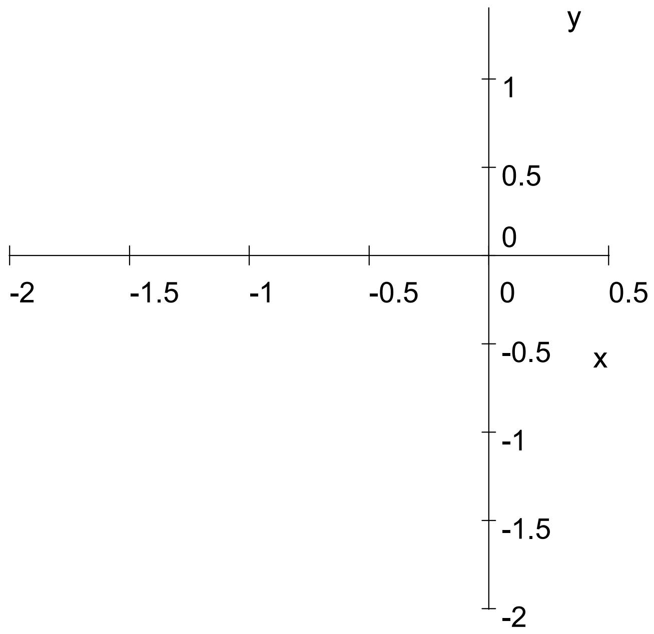
$$z \to z^2 + c$$

Hier aber wird immer bei z0=0 gestartet, variiert und gezeichnet wird c,
gefärbt entsprechend dem Verhalten der Folge fc.
Erreicht sie die Fluchtmenge (Abs(zn)>2) wird das zugehörige c mit einer Farbe gefärbt,
die aus dem momentanen Folgenindex berechnet wird,
der Liste der bisher untersuchten Punkte c zugefügt.
Wenn nach kmax Schritten der Eintritt in den Fluchtbereich noch nicht erfolgt ist, wird c schwarz gefärbt.
Dies nennt man "escape-time-Algorithmus" mit Niveaulinien.

```
kmax:=12:
```

```
apfpkt:=proc(c)
          local rr,rf,i,zz,cpkt;
          begin
           rr:=float(abs(c));i:=0;zz:=0;
           cpkt:=zpkt(c);
           cpkt::Color:=[(kmax-i)/(kmax),1/(kmax+1)*(kmax-(1+(-
  1)^i/2)),0];
           repeat
             if rr>2 then
               cpkt::Color:=[(kmax-i)/(kmax),1/(kmax+1)*(kmax-(1+(-
  1)^i/2)),0];
               return(cpkt);
             else
               i:=i+1; zz:=zz^2+c;
               rr:=float(abs(zz));
             end_if;
           until i=kmax+1 end_repeat;
           cpkt::Color:=[0,0,0];
           return(cpkt);
         end_proc:
```

```
liste:=op({}):fein:=90:
liste:=((liste,apfpkt(-2+2.5*i/fein+(-2+3.4*k/fein)*I) $ i=0..fein )
                 $ k=0..fein): plot(liste):
```



-

Definitionen und Sätze
Die Mandelbrotmenge der Iteration mit fc(z)= z^2+c (=das Apfelmännchen) ist die
Menge aller c, für die die bei 0 startende Folge beschränkt bleibt.
Hier ist eine Näherung schwarz gezeichnet.
Die Symmetrieachse ist die reelle Achse.
Das Apfelmännchen reicht ohne Unterbrechung auf der reellen Achse von -2 bis +1/4.
Das Apfelmännchen liegt in der Gauß-Ebene in einem Kreis mit Radius 2
Satz:
Genau die Juliamengen zu c aus dem Apfelmännchen sind zusammenhängend.
Dies wird auch machmal als Definition genommen. Dann ist obige Definition ein Satz.

-
-