

Musterlösung zu Aufgabe 15.1

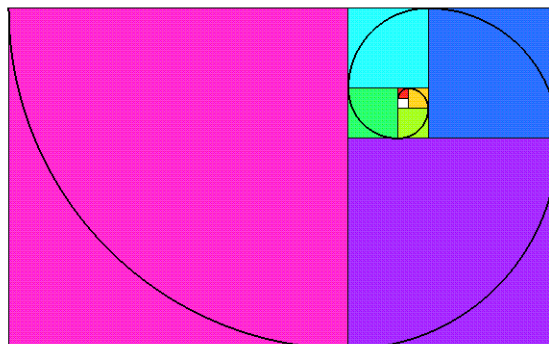
```
> restart;  
with(LinearAlgebra):with(plots):with(plottools):
```

Wir kopieren einiges aus der Lösung von David und Rudi. Die Wahl dieser Gruppe ist willkürlich.

```
> a:=proc(n)  
  option remember;  
  if n=1 then return 1; end if;  
  if n=2 then return 1; end if;  
  if n>=3 then  
    procname(n-1)+procname(n-2);  
  end if;  
end proc;  
> Viertelkreis:=proc(m,r,q)  
  plot([m[1]+r*cos(t),m[2]+r*sin(t), t=Pi-q*Pi/2..Pi/2-q*Pi/2],  
  color=black,thickness=2);  
end proc;  
> zeichne:=proc(n)  
  local M,p,r,i,q;  
  p := table():  
  r := table():  
  M := [0,0]:  
  for i from 0 to n do  
    q := ((i-1) mod 4) + 1;  
    p[i] := Viertelkreis(M,a(i+2),q);  
    if q = 4 then  
      r[i] := rectangle(M,M+[-a(i+2),a(i+2)],color=COLOR(HUE, i/(n+1)));  
      M[2] := M[2] - a(i+1);  
    elif q = 1 then  
      r[i] := rectangle(M,M+[a(i+2),a(i+2)],color=COLOR(HUE, i/(n+1)));  
      M[1] := M[1] - a(i+1);  
    elif q = 2 then  
      r[i] := rectangle(M,M+[a(i+2),-a(i+2)],color=COLOR(HUE, i/(n+1)));  
      M[2] := M[2] + a(i+1);  
    else  
      r[i] := rectangle(M,M+[-a(i+2),-a(i+2)],color=COLOR(HUE, i/(n+1)));  
      M[1] := M[1] + a(i+1);  
    end if;  
  end do;  
  display([op(convert(r,'list')),op(convert(p,'list'))],scaling=constrained,axes=none);  
end proc;
```

Wir testen die Routine und stellen fest, dass nach unserer Numerierung eine Spirale bis zum Punkt P_9 gezeichnet wird, wenn wir `zeichne(7)` aufrufen.

```
> zeichne(7);
```



Wir definieren uns eine Abbildung $i \rightarrow P(i) = \langle x_i, y_i \rangle$:

```
> P := i -> <x[i],y[i]>:
```

So rechnet Maple für uns die Koordinaten x_i und y_i aus:

```
> P_i := P(i-1) - (P(i-2) - P(i-3)) + <<0,-1>|<1,0>>.(P(i-1) - P(i-2)):
x[i] := P_i[1];
y[i] := P_i[2];
```

$$x_i := x_{i-1} - x_{i-2} + x_{i-3} + y_{i-1} - y_{i-2}$$

$$y_i := y_{i-1} - y_{i-2} + y_{i-3} - x_{i-1} + x_{i-2}$$

Daraus lesen wir die folgende Matrix A ab:

```
> A :=
<<0,0,0,0,1,0>|<0,0,0,0,0,1>|<1,0,0,0,-1,1>|<0,1,0,0,-1,-1>|<0,0,1,0,1,-1>|<0,0,0,1,1,1>>;
```

$$A := \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 \\ 0 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

Um zu überprüfen, dass wir richtig abgelesen haben, definieren wir eine Abbildung v :

Zur Erinnerung: A soll erfüllen, dass für alle i in \mathbb{N} : $A v(i) = v(i+1)$.

Motivation: Dann gilt (trivialer Induktionsbeweis): $A^n v(0) = v(n)$. Unser Ziel ist es, eine geschlossene Form für A^n zu finden, diese dann mit dem konstanten $v(0)$ zu multiplizieren, und so eine geschlossene Form für $v(n)$ zu erhalten, dass als erste und zweite Koordinaten x_n und y_n enthält. Auf diese Art erhalten wir die **parametrisierte** Spirale $n \rightarrow \langle x_n, y_n \rangle$.

```
> v := i -> <x[i],y[i],x[i+1],y[i+1],x[i+2],y[i+2]>:
```

Und machen die Probe:

Hinweis: Der Befehl `op` ist jeweils nötig, damit **nicht** überprüft wird, ob es **dasselbe** Objekt ist, sondern nur, ob die Inhalte **identisch** sind.

```
> A.v(i-3);
evalb( op(A.v(i-3)) = op(v(i-2)) );
```

$$\begin{bmatrix} x_{i-2} \\ y_{i-2} \\ x_{i-1} \\ y_{i-1} \\ x_{i-1} - x_{i-2} + x_{i-3} + y_{i-1} - y_{i-2} \\ y_{i-1} - y_{i-2} + y_{i-3} - x_{i-1} + x_{i-2} \end{bmatrix}$$

true

Jetzt aber weiter mit der Aufgabe. Wir berechnen die Jordan'sche Normalform J von A sowie die Übergangsmatrix Q mit

$$Q J Q^{(-1)} = A.$$

```
> J,Q := JordanForm(A,output=['J','Q']):
J;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}I\sqrt{5} - \frac{1}{2}I & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2}I\sqrt{5} + \frac{1}{2}I & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}I + \frac{1}{2}I\sqrt{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2}I - \frac{1}{2}I\sqrt{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Zunächst vereinfachen wir Q wie folgt:

Zur Erinnerung: `@` steht für Hintereinanderausführung von Funktionen, `@@` für wiederholtes Ausführen der selben Funktion.

```
> Q:=Map((rationalize@expand)@@2,Q):
```

Überprüfen von J und Q :

So kommt jeweils ein Zahlenwust heraus:

```
> #Q.J.Q^(-1);
#simplify(%);
```

So nicht:

```
> simplify(Q.J.Q^(-1) - A);  
map(simplify@expand@rationalize,Q.J.Q^(-1));
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 \\ 0 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

Folgender Befehl liefert eine geschlossene Form für A^n :

Erläuterung: Maple kann Ausdrücke der Form A^n für variables n nicht vereinfachen, da das i.a. auch gar nicht geht. Wir haben nun eine Zerlegung $A = Q J Q^{(-1)}$ berechnet und damit auch die Zerlegung $A^n = Q J^n Q^{(-1)}$. Der Ausdruck J^n lässt sich leicht "vereinfachen", da man eine Diagonalmatrix potenziert, indem man die Diagonalelemente (oder wie hier einfach alle) potenziert.

```
> An := map((rationalize@expand)@@2,Q.map(x->x^n,J).Q^(-1));
```

Wenn wir jetzt diese Matrix mit einem geeigneten v_0 multiplizieren, erhalten wir eine geschlossene Form für v_n . In v_0 müssen die drei Punkte P_0, P_1 und P_2 der Spirale stehen, wobei das Koordinatensystem mit dem aus dem gewählten Projekt übereinstimmen muss. In Davids und Rudis Projekt ist der Ursprung der rechte Punkt zwischen den beiden kleinen Quadraten. Also hat $P_0 = (0,-1)$, $P_1 = (-1,0)$ und $P_2 = (0,1)$.

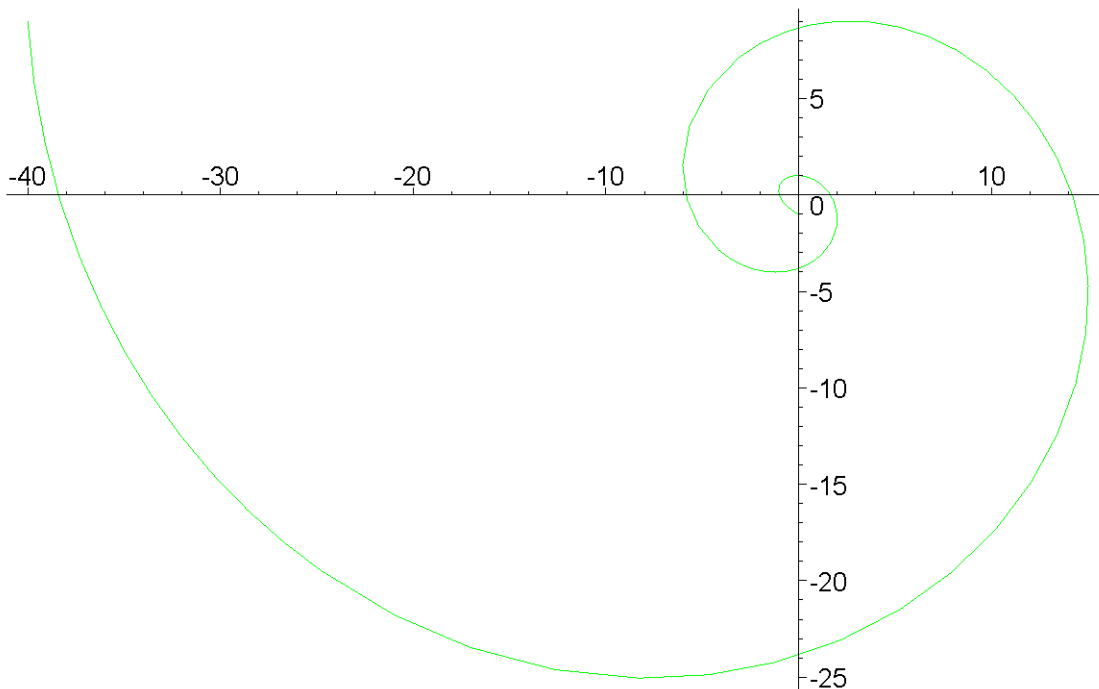
```
> v0 := <0,-1,-1,0,0,1>;  
vn := map((rationalize@expand)@@2,An.v0);
```

Jetzt können wir die parametrisierte Kurve plotten. Man beachte die Syntax ("Form") des benötigten Plotbefehls, die natürlich wie immer der Hilfe (?plot) entnommen ist.

```
> s := plot([vn[1],vn[2],n=0..9],color=green,scaling=constrained);
```

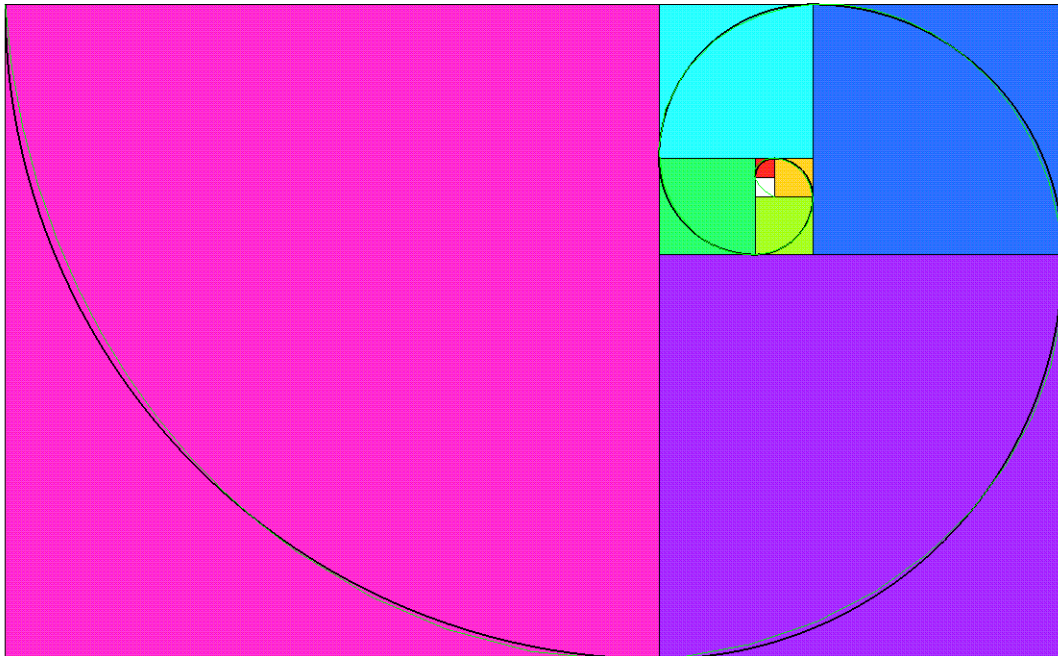
Wir zeigen die Kurve erstmal an:

```
> display(s);
```



Wir zeichnen die Spirale in das Bild mit den bunten Quadraten und den Viertelkreisen:

```
> display([s,zeichne(7)]);
```



Wir können auch leicht einen Film von unserer parametrisierten Spirale machen. Dabei läuft die Spirale außen viel schneller, weil zwischen je zwei Punkten nur eine Zeiteinheit liegt, der Abstand aber sehr schnell wächst.

```
> display( ['plot([vn[1],vn[2],n=0..N/10],color=green,scaling=constrained)'$'N'=1..100],  
insequence = true );
```