

# Pascalsche Schnecken in polar-kartesischer Darstellung

Prof. Dr. Dörte Haftendorn: Mathematik mit MuPAD 4, Aug. 07 Update 20.08.07

[www.mathematik-verstehen.de](http://www.mathematik-verstehen.de)

<http://haftendorn.uni-lueneburg.de>

```
kreis:=t->cos(t);
```

```
r:=t->kreis(t)+k;
```

```
t → cos(t)
```

```
t → kreis(t) + k
```

```
k:=0.5: //Leinenlänge
```

```
pascal:=plot::Polar([r(t),t],t=0..ende,ende=0..2*PI,  
  LineWidth=1,LineColor=[1,0,0], Mesh=400):
```

```
pkt:=plotPoint2d([r(t),t],t=0..ende,ende=0..2*PI,PointSize=1.1):
```

```
leine:=plot::Line2d([r(t)*cos(t),r(t)*sin(t)],
```

```
[kreis(t)*cos(t),kreis(t)*sin(t)],t=0..2*PI,
```

```
  LineWidth=0.5):
```

```
pascalkart:=plot::Curve2d([t,r(t)],t=0..ende,ende=0..2*PI,
```

```
  LineWidth=1, Mesh=400, LineColor=RGB::Green):
```

```
radius2:=plot::Line2d([0,0],[r(t)*cos(t),r(t)*sin(t)],t=0..2*PI,
```

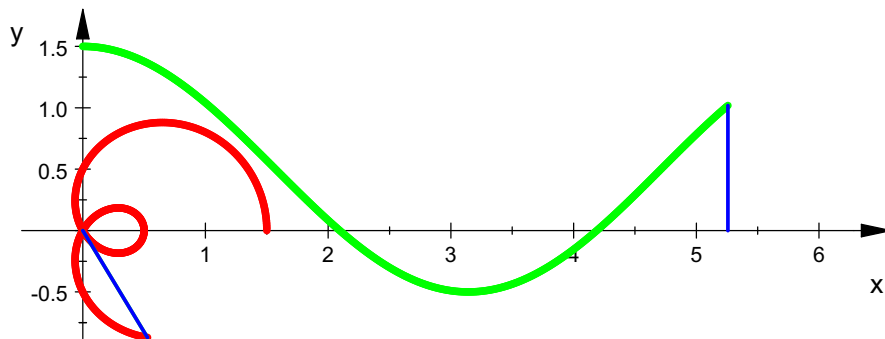
```
  LineWidth=0.5):
```

```
radiusbetrag2:=plot::Line2d([0,0],[abs(r(t))*cos(t),abs(r(t))*sin(t)],
```

```
  t=0..2*PI,LineColor=[0,1,0]):
```

```
radiusordi2:=plot::Line2d([t,0],[t,r(t)],t=0..2*PI):
```

```
plot(pascal,radiusbetrag2,radius2,pascalkart,radiusordi2,LineWidth=0.5,  
  AnimationStyle=BackAndForth);
```



## Variation der Leinenlänge

```
k:=1.5: //Leinenlänge
```

```
pascal:=plot::Polar([r(t),t],t=0..ende,ende=0..2*PI,  
  LineWidth=1,LineColor=[1,0,0], Mesh=400):
```

```
pkt:=plotPoint2d([r(t),t],t=0..ende,ende=0..2*PI,PointSize=1.1):
```

```
leine:=plot::Line2d([r(t)*cos(t),r(t)*sin(t)],
```

```
[kreis(t)*cos(t),kreis(t)*sin(t)],t=0..2*PI,
```

```
  LineWidth=0.5):
```

```
pascalkart:=plot::Curve2d([t,r(t)],t=0..ende,ende=0..2*PI,
```

```
  LineWidth=1, Mesh=400, LineColor=RGB::Green):
```

```
radius2:=plot::Line2d([0,0],[r(t)*cos(t),r(t)*sin(t)],t=0..2*PI,
```

```
  LineWidth=0.5):
```

```
radiusbetrag2:=plot::Line2d([0,0],[abs(r(t))*cos(t),abs(r(t))*sin(t)],
```

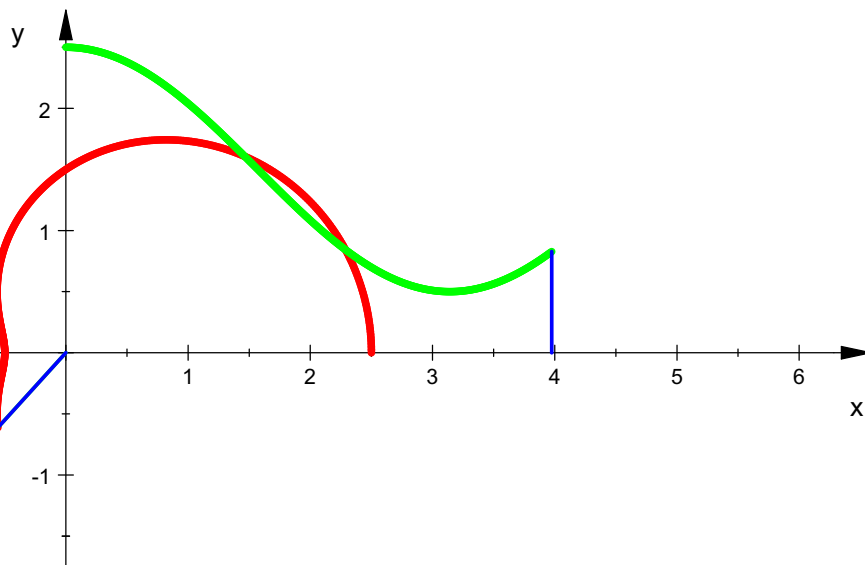
```
  t=0..2*PI,LineColor=[0,1,0]):
```

```
radiusordi2:=plot::Line2d([t,0],[t,r(t)],t=0..2*PI):
```

```

radiusordi2:=plot::Line2d([t,0],[t,r(t)],t=0..2*PI):
plot(pascal,radiusbetrag2,radius2,pascalkart,radiusordi2,LineWidth=0.5,
    AnimationStyle=BackAndForth);

```



animieren durch Anklicken!

```

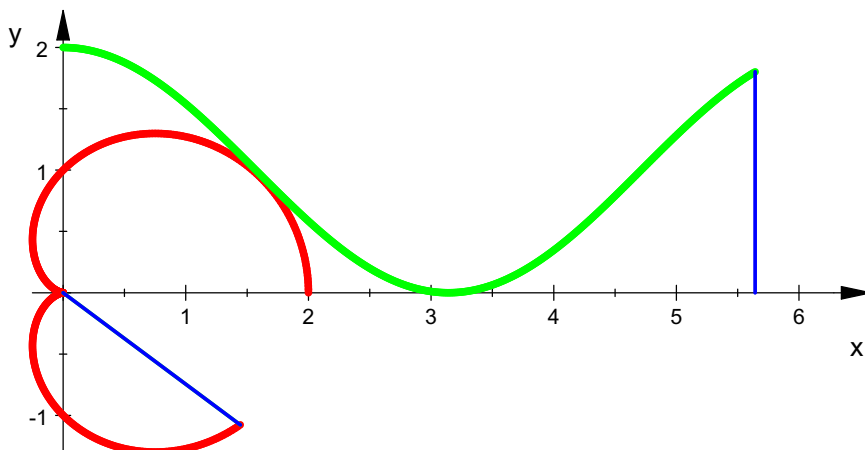
k:=1: //Leinenlänge

```

```

pascal:=plot::Polar([r(t),t],t=0..ende,ende=0..2*PI,
    LineWidth=1,LineColor=[1,0,0], Mesh=400):
pkt:=plotPoint2d([r(t),t],t=0..ende,ende=0..2*PI,PointSize=1.1):
leine:=plot::Line2d([r(t)*cos(t),r(t)*sin(t)],
[kreis(t)*cos(t),kreis(t)*sin(t)],t=0..2*PI,
    LineWidth=0.5):
pascalkart:=plot::Curve2d([t,r(t)],t=0..ende,ende=0..2*PI,
    LineWidth=1, Mesh=400, LineColor=RGB::Green):
radius2:=plot::Line2d([0,0],[r(t)*cos(t),r(t)*sin(t)],t=0..2*PI,
    LineWidth=0.5):
radiusbetrag2:=plot::Line2d([0,0],[abs(r(t))*cos(t),abs(r(t))*sin(t)],
    t=0..2*PI,LineColor=[0,1,0]):
radiusordi2:=plot::Line2d([t,0],[t,r(t)],t=0..2*PI):
plot(pascal,radiusbetrag2,radius2,pascalkart,radiusordi2,LineWidth=0.5,
    AnimationStyle=BackAndForth);

```



animieren durch Anklicken!

Ob sich beide Kurven treffen, ist in einer Extraseite untersucht.

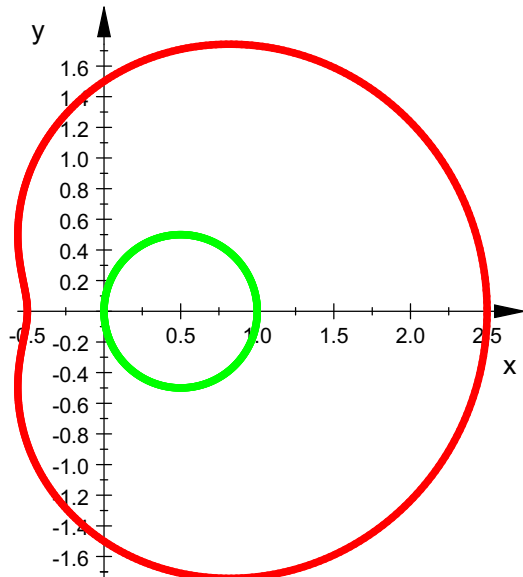
```

delete k

```

Animation der Pascalschen Schnecken allein:

```
pascalc:=plot::Polar([r(t),t],t=0..2*PI,k=-1.5..1.5,  
    LineWidth=1,LineColor=[1,0,0], Mesh=400):  
kreisg:=plot::Polar([r(t)|k=0,t],t=0..2*PI,  
    LineWidth=1,LineColor=[0,1,0], Mesh=400):  
plot(pascalc,kreisg)
```



 animieren durch Anklicken!