

# Einführung in Sage - Einheit 6

## Folgen, Reihen, Potenzreihen, Vertiefung Schleifen

Jochen Schulz

Georg-August Universität Göttingen 

- 1 Folgen
- 2 Reihen
- 3 Potenzreihen
- 4 Vertiefung Schleifen ??

1 Folgen

2 Reihen

3 Potenzreihen

4 Vertiefung Schleifen ??

- Eine **reelle Zahlenfolge** kurz **Folge** genannt, ist eine Abbildung von  $\mathbb{N}$  in  $\mathbb{R}$ .
- Statt  $a : \mathbb{N} \rightarrow \mathbb{R}$  schreibt man in Anlehnung an die Vektornotation  $(a_n)_{n \in \mathbb{N}}$  oder einfach  $(a_n)_n$ .
- Natürlich kann man auch Folgen  $\mathbb{N} \rightarrow Y$  auf beliebigen Mengen  $Y$  betrachten. Aber wir beschränken uns auf den Fall  $Y = \mathbb{R}$ .
- Die Zahlen  $a_n$  heißen **Glieder** der Folge.
- Eine **Teilfolge**  $(a_{n_i})_{n_i}$  ist eine Abbildung  $a : N \rightarrow \mathbb{R}$ , wobei  $N \subset \mathbb{N}$  eine Menge mit unendlich vielen Elementen ist.

# Konvergenz von Folgen

Eine Zahlenfolge  $(a_n)_n$  ist **konvergent** gegen den **Grenzwert** oder **Limes**  $a \in \mathbb{R}$ , wenn es zu jedem  $\varepsilon > 0$  ein  $n_0 \in \mathbb{N}$  gibt, so dass für alle  $n \geq n_0$  die Abschätzung

$$|a_n - a| < \varepsilon$$

gilt. Man schreibt

$$a = \lim_{n \rightarrow \infty} a_n.$$

Eine nicht konvergente Folge nennt man **divergent**.

- Konvergiert eine Folge gegen 0, so nennt man sie eine **Nullfolge**.
- Der Grenzwert einer konvergenten Teilfolge  $(a_{n_i})_{n_i}$  heißt **Häufungspunkt**.
- Eine Folge kann keinen aber auch mehrere Häufungspunkte besitzen; konvergente Folgen haben genau einen Häufungspunkt.
- Eine **Cauchy-Folge** ist eine Folge  $(a_n)_n$  bei der für alle  $\varepsilon > 0$  ein  $n_0 \in \mathbb{N}$  existiert, so dass für alle  $n, m \geq n_0$  gilt:  $|a_n - a_m| < \varepsilon$ . In  $\mathbb{R}$  ist eine Folge konvergent, genau dann wenn sie eine Cauchy-Folge ist (Vollständigkeit).
- Eine  **$\varepsilon$ -Umgebung**  $U_\varepsilon(a)$  von  $a$  ist definiert durch

$$U_\varepsilon(a) := (a - \varepsilon, a + \varepsilon) := \{x \in \mathbb{R} \mid |x - a| < \varepsilon\}.$$

# Beispiele

$$\begin{array}{ll} a_n := \frac{1}{n+1} & 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots \\ b_n := 2^{-n} & 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots \\ c_n := 2^n & 1, 2, 4, 8, \dots \\ d_n := \left(\frac{n+2}{n+1}\right)^{n+1} & \frac{2^1}{1^1}, \frac{3^2}{2^2}, \frac{4^3}{3^3}, \dots \\ e_n = (-1)^n & 1, -1, 1, -1, \dots \end{array}$$

Die Folgen  $(a_n)_n$ ,  $(b_n)_n$  und  $(d_n)_n$  konvergieren und  $(c_n)_n$ ,  $(e_n)_n$  divergieren.

# Folgen in Sage I

Grenzwerte von Folgen  $(a_n)_n$  können in Sage mit Hilfe von

```
limit(expr(x), x = oo, dir='above')
```

berechnet werden. Dabei ist  $expr(x)$  ein Ausdruck.

**Beispiele:**

```
_ = var('n'); limit(1/(n+1), n=oo)
```

0

```
limit(((n+2)/(n+1))^(n+1), n=oo)
```

e

```
limit((-1)^n, n=oo)
```

ind



# Folgen in Sage II

```
limit(2^n, n=oo)
```

`+Infinity`

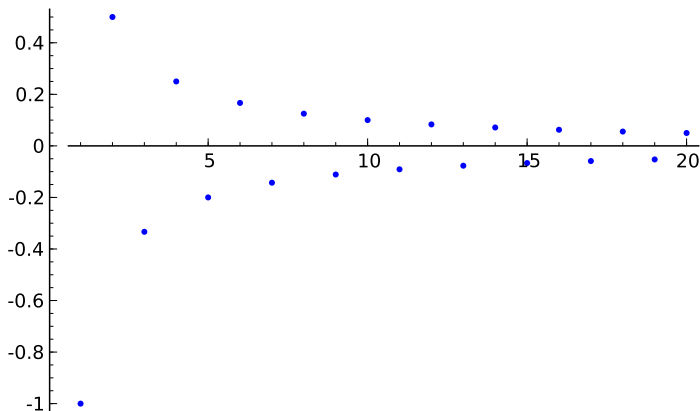
```
lim(x*sin(1/x), x=0)
```

`0`

# Visualisieren von Folgen

Folgen können in Sage durch `points` visualisiert werden.

```
var('n');  
point([(n, (-1)^n/n) for n in range(1,21)], pointsize=8)
```



# Konvergenzkriterien

- Jede monotone, beschränkte Folge konvergiert.
- Sind  $(a_n)_n$  und  $(b_n)_n$  konvergente Folgen,  $\alpha, \beta \in \mathbb{R}$ , so ist auch die Folge  $(\alpha a_n + \beta b_n)_n$  konvergent mit dem Grenzwert

$$\lim_{n \rightarrow \infty} (\alpha a_n + \beta b_n) = \alpha \lim_{n \rightarrow \infty} a_n + \beta \lim_{n \rightarrow \infty} b_n.$$

- Sind  $(a_n)_n$  und  $(b_n)_n$  konvergente Folgen, so ist auch die Folge  $(a_n b_n)_n$  konvergent mit dem Grenzwert

$$\lim_{n \rightarrow \infty} (a_n b_n) = \left( \lim_{n \rightarrow \infty} a_n \right) \cdot \left( \lim_{n \rightarrow \infty} b_n \right).$$

- Weglassen oder Hinzufügen endlich vieler Glieder verändert das Konvergenzverhalten nicht.

- (Bolzano-Weierstrass) Jede beschränkte Folge besitzt (mindestens) eine konvergente Teilfolge.
- Jede Teilfolge einer konvergenten Folge konvergiert gegen den Grenzwert der ursprünglichen Folge.
- Jede konvergente Folge ist beschränkt, d.h. es gibt ein  $K > 0$ , so dass  $|a_n| \leq K$  gilt für alle  $n \in \mathbb{N}$ .
- Seien  $(a_n)_n$  und  $(b_n)_n$  konvergente Folgen mit  $\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n$ . Dann gilt für eine Folge  $(c_n)_n$  mit  $a_n \leq c_n \leq b_n$ ,  $n \in \mathbb{N}$ , dass sie konvergiert mit  $\lim_{n \rightarrow \infty} c_n = \lim_{n \rightarrow \infty} b_n$ .

# Rekursive Folgen

Rekursive Folgen können durch rekursive Funktionen erzeugt werden.

**Beispiel:**

$$y_{n+2} := 2y_{n+1} - y_n + 2, \quad y_0 = -1, y_1 = a.$$

```
var('a')
def y(n):
    if n==0:
        return -1
    if n==1:
        return a
    return 2*y(n-1)-y(n-2)+2
y(4)
```

4\*a + 15

1 Folgen

2 **Reihen**

3 Potenzreihen

4 Vertiefung Schleifen ??

Sei  $(a_n)_n$  eine Folge reeller Zahlen. Eine (unendliche) Reihe mit den Gliedern  $a_n$ , in Zeichen

$$\sum_{n=1}^{\infty} a_n = a_1 + a_2 + a_3 + \dots,$$

ist definiert durch die Folge  $(s_n)_n$  der Partialsummen

$$s_n = \sum_{k=1}^n a_k = a_1 + a_2 + \dots + a_n$$

Der Grenzwert  $s$  der Folge  $(s_n)_n$  wird als Wert oder Summe der Reihe bezeichnet. Man schreibt

$$s = \sum_{n=1}^{\infty} a_n.$$

- Beginnt die Indizierung statt bei 1 mit einer anderen ganzen Zahl  $m$ , so wird  $\sum_{n=m}^{\infty} a_n$  entsprechend eingeführt.
- Bei Abänderung, Weglassen oder Hinzufügen endlich vieler Glieder bleiben Konvergenz und Divergenz unberührt. I.A. wird sich aber der Grenzwert ändern.
- Reihen sind eine spezielle Art von Folgen.



- Die **geometrische Reihe** ist gegeben durch  $\sum_{n=0}^{\infty} x^n$ . Die Partialsummen lauten

$$s_n = 1 + x + x^2 + \dots + x^n = \begin{cases} n + 1, & \text{falls } x = 1 \\ \frac{1-x^{n+1}}{1-x}, & \text{falls } x \neq 1 \end{cases}.$$

Also divergiert die Reihe für  $|x| \geq 1$  und konvergiert für  $|x| < 1$  mit dem Wert  $\sum_{n=0}^{\infty} = \frac{1}{1-x}$ .

- Die Reihe  $\sum_{n=1}^{\infty} \frac{1}{n^2}$  konvergiert gegen  $\pi^2/6$ .

# Beispiele II

- Die **harmonische Reihe**  $\sum_{n=1}^{\infty} \frac{1}{n}$  divergiert.
- Die **alternierende harmonische Reihe**  $\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}$  konvergiert.
- Die Reihe  $\sum_{n=1}^{\infty} \frac{1}{n^s}$  konvergiert für  $s > 1$ .
- Die Reihe  $\sum_{n=2}^{\infty} \frac{1}{n(\log n)^s}$  konvergiert für  $s > 1$  und divergiert für  $s = 1$ .

# Reihen mit Sage I

Der Befehl `sum(f,i=a..b)` sucht eine geschlossene Darstellung der Summe  $\sum_{i=a}^b f(i)$ . Dabei sind  $a, b$  ganze Zahlen, wobei auch unendlich (also `infinity`) erlaubt ist und  $f$  ist ein Ausdruck in  $i$ .

```
_ = var('k'); sum(1/k^2, k, 1, oo)
```

```
1/6*pi^2
```

```
sum((-1)^(k+1)/k, k, 1, oo)
```

```
log(2)
```

```
sum(1/k, k, 1, oo)
```

```
ValueError: Sum is divergent
```

# Reihen mit Sage II

Oft ist die Konvergenz einer Reihe abhängig von bestimmten Parametern, wie z.B. bei der geometrischen Reihe. Und je nach Parameterwert zeigt die Reihe unterschiedliches Konvergenzverhalten

```
sum(x^k,k,0,oo)
```

Is `abs(x)-1` positive, negative, or zero?

Entsprechend gibt es keine geschlossene Form. Für  $x = 1/2$  gilt jedoch

```
x = 1/2; sum(x^k,k,0,oo)
```

- Definieren der Partialsumme

```
_ = var('x,n,k')  
s = sum(x^k,k,0,n); s
```

$$(x^{(n + 1)} - 1)/(x - 1)$$

- Die ersten 5 Glieder der Partialsumme

```
assume(x<>1); [s(n=m) for m in [1..6]]
```

$$[(x^2 - 1)/(x - 1), (x^3 - 1)/(x - 1), (x^4 - 1)/(x - 1), (x^5 - 1)/(x - 1), (x^6 - 1)/(x - 1), (x^7 - 1)/(x - 1)]$$

- Bestimmen des Grenzwertes der Folge der Partialsummen

```
forget(); assume(abs(x)<1); limit(s,n=oo)
```

$-1/(x - 1)$

```
forget(); assume(x>1); limit(s,n=oo)
```

$+\text{Infinity}$

Mit der Funktion `assume` kann man Funktionen wie `expand`, `simplify` oder `solve` mitteilen, dass für gewisse Bezeichner Annahmen über ihre Bedeutung gemacht wurden.

## Beispiele:

`assume(x, 'real')`     $x$  wird auf  $\mathbb{R}$  eingeschränkt!

`assume(x>a)`     $x$  wird auf  $\{y \in \mathbb{R} \mid y > a\}$  eingeschränkt!

Ruft man `assume` mehrmals für einen Bezeichner auf, werden zusätzliche Annahmen gemacht. Sind diese widersprüchlich erhält man eine entsprechende Meldung.

- Umformungen oder Vereinfachungen für symbolische Bezeichner werden i.A. nur dann durchgeführt, wenn sie für alle komplexen Zahlen gelten. Hier kann ein Einschränken des Definitionsbereichs helfen.
- Mittels `forget(x>a)` wird die Annahme  $x>a$  gelöscht.
- Durch `assumptions()` können alle Annahmen ausgegeben werden.



# Beispiele zu assume I

```
var('c'); assumptions()
```

```
c  
[]
```

```
c = 2; assume(c>0)
```

```
AttributeError: 'bool' object has no attribute 'assume'
```

# Beispiele zu assume II

```
_ = var('c')  
assume(c, 'integer'); assumptions()
```

```
[c is integer]
```

```
sin(c*pi)
```

```
sin(pi*c)
```

```
sin(c*pi).simplify()
```

```
0
```

```
assume(x>0)  
sqrt(x^2).simplify()
```

```
x
```

# Einige Annahmen für assume

Annahme	Erklärung
'real'	$\mathbb{R}$
'rational'	$\mathbb{Q}$
'integer'	$\mathbb{Z}$
'complex'	$\mathbb{C}$
'even'	gerade Zahl
'odd'	ungerade Zahl
'increasing'	wachsend
'analytic'	analytisch

- **Cauchy Kriterium:** Eine Reihe  $\sum_{n=1}^{\infty} a_n$  konvergiert genau dann, wenn es zu jedem  $\varepsilon > 0$  ein  $n_0 \in \mathbb{N}$  gibt, so dass für alle  $m, n \geq n_0$  gilt  $|\sum_{k=m}^n a_k| < \varepsilon$ .
- **Notwendiges Kriterium:** Konvergiert eine Reihe, so bilden ihre Glieder eine Nullfolge. Dieses Kriterium ist **nicht** hinreichend!
- **Verdichtungskriterium:** Eine Reihe  $\sum_{n=1}^{\infty} a_n$  mit einer Folge nichtnegativer, monoton fallender Glieder konvergiert genau dann, wenn die Reihe  $\sum_{n=1}^{\infty} 2^n a_{2^n}$  konvergiert.

# Majorantenkriterium

- Gilt  $0 \leq c_n \leq a_n \leq b_n$  für alle  $n \in \mathbb{N}$ , so nennt man  $\sum_{n=1}^{\infty} c_n$  eine **Minorante** und  $\sum_{n=1}^{\infty} b_n$  eine **Majorante** von  $\sum_{n=1}^{\infty} a_n$ .
- Besitzt eine Reihe mit nichtnegativen Gliedern eine konvergente Majorante, so konvergiert sie.
- Besitzt eine Reihe mit nichtnegativen Gliedern dagegen eine divergente Minorante, so divergiert sie.

# Konvergenzkriterien

Die Reihe  $\sum_{n=1}^{\infty} a_n$  konvergiert, wenn...

**Quotientenkriterium:** Die Glieder positiv sind und ein  $q < 1$  existiert, so dass für  $n \in \mathbb{N}$  gilt  $\frac{a_{n+1}}{a_n} \leq q$ .

**Wurzelkriterium:** Die Glieder positiv sind und ein  $q < 1$  existiert, so dass für  $n \in \mathbb{N}$  gilt  $\sqrt[n]{a_n} \leq q$ .

**Leibnizsches Kriterium:** Die Reihe  $\sum_{n=1}^{\infty} (-1)^n a_n$  konvergiert, wenn die Folge  $(a_n)_n$  eine monoton fallende Nullfolge ist.

# Beispiele

- Betrachte  $\sum_{n=0}^{\infty} n^4 e^{-n^2}$

```
f(n) = n^4.*exp(-n*n)
g(n) = f(n+1)/f(n)
limit(g(n),n=oo)
```

0

- Betrachte  $\sum_{n=2}^{\infty} \frac{1}{n(\log n)^2}$

```
f(n) = 1/(n*(ln(n)^2))
g(n) = 2^n*f(2^n)
h(n) = 2^n*g(2^n)
limit(h(n+1)/h(n),n=oo)
```

1/2

# Absolute und bedingte Konvergenz

Eine Reihe  $\sum_{n=0}^{\infty} a_n$  heißt **absolut konvergent** genau dann wenn  $\sum_{n=0}^{\infty} |a_n|$  konvergiert.

Eine konvergente, aber nicht absolut konvergente Reihe heißt **bedingt konvergent**.

- Absolut konvergente Reihen können beliebig umgeordnet werden.
- Dies ist i.d.R. bei nicht absolut konvergenten Reihen falsch!



1 Folgen

2 Reihen

**3 Potenzreihen**

4 Vertiefung Schleifen ??

Eine **Potenzreihe** ist eine Reihe der Form

$$\sum_{n=0}^{\infty} a_n (x - x_0)^n$$

mit  $x_0 \in \mathbb{R}$ . Das Konvergenzverhalten für verschiedene  $x$  wird durch den **Konvergenzradius**

$$\rho := \frac{1}{\limsup_{n \rightarrow \infty} \sqrt[n]{|a_n|}}$$

bestimmt. Für  $|x - x_0| < \rho$  konvergiert die Potenzreihe absolut und für  $|x - x_0| > \rho$  divergiert sie.

- Ist  $a_n \neq 0$  für alle  $n > n_0$ , dann gilt für den Konvergenzradius:

$$\rho = \limsup_{n \rightarrow \infty} \frac{|a_n|}{|a_{n+1}|}.$$

- Potenzreihen konvergieren innerhalb ihres Konvergenzradius absolut.
- Die Konvergenz an den Stellen  $x_0 - \rho$  und  $x_0 + \rho$  muss bei jeder Reihe individuell geprüft werden.
- Potenzreihen sind ein mächtiges Werkzeug innerhalb der Mathematik.

# Beispiele

- $\sum_{n=1}^{\infty} \frac{x^n}{n!}$

```
f(n) = 1/factorial(n)
rho = limit(expand(f(n+1)/f(n)),n=oo); rho
```

0

Die Potenzreihe konvergiert für alle  $x \in \mathbb{R}$ .

- $\sum_{n=0}^{\infty} n^s x^n, s > 0$

```
_ = var('s'); f(n) = n^s; assume(s>0)
limit(expand(f(n)^(1/n)),n=infinity)
```

1

Der Konvergenzradius ist 1.

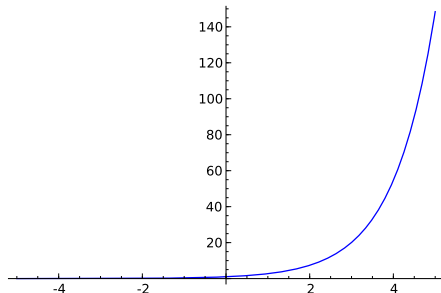
# Exponentialfunktion

Wir erklären die **Exponentialfunktion** durch

$$\exp(x) := \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, x \in \mathbb{R}.$$

Die Funktion ist auf ganz  $\mathbb{R}$  definiert. Plot:

```
plot(exp, (-5, 5))
```



# Eigenschaften der Exponentialfunktion

- Es gilt  $\exp(x + y) = \exp(x) \cdot \exp(y)$ .
- Es gilt  $\exp(x) = \lim_{n \rightarrow \infty} (1 + \frac{x}{n})^n$ .
- Es gilt  $\exp(x) = 1 / \exp(-x)$ .
- Die Umkehrfunktion auf  $\mathbb{R}_+$  der Exponentialfunktion ist die Logarithmusfunktion  $\log(x)$ . Es gilt

$$\exp(\log(x)) = x, \quad x > 0, \quad \log(\exp(x)) = x, \quad x \in \mathbb{R}.$$

- Die **allgemeine Potenz** ist durch  $a^x := \exp(x \log a)$ ,  $a \in \mathbb{R}_+$  definiert.

```
sum(x^n/factorial(n),n,0,oo)
```

$e^x$

```
exp(log(x))
```

$x$

```
_ = var('n'); limit((1+x/n)^n,n=oo)
```

$e^x$

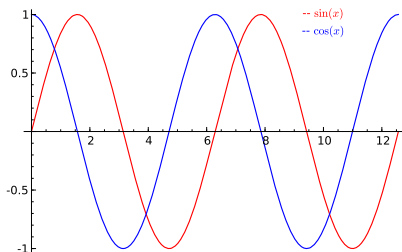
# Trigonometrische Funktionen

Die **Sinusfunktion** und die **Cosinusfunktion** sind definiert durch

$$\sin(x) := \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad \cos(x) := \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}.$$

Die Potenzreihen konvergieren für alle  $x \in \mathbb{R}$ . Plotten:

```
p = plot(sin,0,4*pi,color='red')
p += plot(cos,0,4*pi);
p += text('-- $\sin(x)$', (10, 1.0), color='red')
p += text('-- $\cos(x)$', (10, 0.85)); p.show()
```





- Es gelten die Additionstheoreme:

$$\sin(x + y) = \sin x \cos y + \cos x \sin y$$

$$\cos(x + y) = \cos x \cos y - \sin x \sin y.$$

- Es gilt:

$$\sin^2 x + \cos^2 x = 1.$$

- Wir definieren  $\pi$ , indem wir die kleinste positive Nullstelle von  $\cos(x)$  als  $\pi/2$  definieren.
- Es gilt:

$$\sin(x + \pi/2) = \cos(x)$$

$$\cos(x + \pi/2) = -\sin(x).$$

```
solve(cos(x)==0,x)
```

```
[x == 1/2*pi]
```

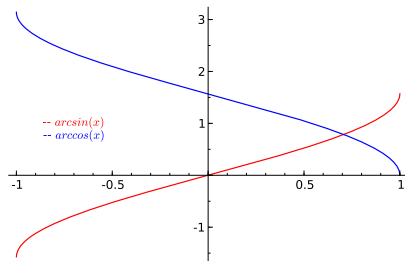
```
sin(x+pi/2).simplify()
```

```
cos(x)
```

# Weitere Eigenschaften I

- Die Umkehrfunktionen von Sinus und Cosinus werden mit  $\arcsin$  und  $\arccos$  bezeichnet. In Sage:  $\arcsin$  und  $\arccos$ . Plotten:

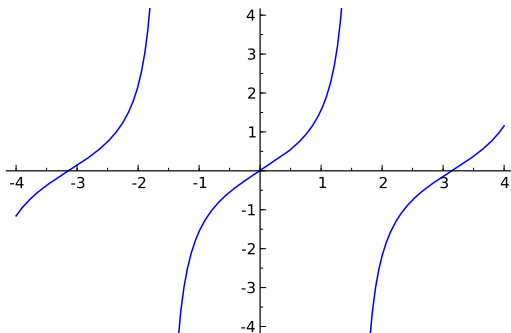
```
p = plot(arcsin,-1,1,color='red')
p += plot(arccos,-1,1);
p += text('-- $arcsin(x)$', (-0.7, 1.0), color='red')
p += text('-- $arccos(x)$', (-0.7, 0.75)); p.show()
```



# Weitere Eigenschaften II

- Der **Tangens** ist definiert durch  $\tan(x) := \frac{\sin(x)}{\cos(x)}$ .

```
plot(tan, -4, 4, detect_poles=True, ymax=4, ymin=-4)
```



1 Folgen

2 Reihen

3 Potenzreihen

**4 Vertiefung Schleifen ??**

Eine elegante Möglichkeit sind Schleifen über Listen oder Mengen.

```
L = [1..10]
for i in L:
    x = i^2
    print("Das Quadrat von {0} ist {1}").format(i,x)
```

Wir geben für die natürlichen Zahlen  $\leq 1000$  an, wieviele Zahlen  $1, 2, 3, \dots$  Teiler haben.

```
Liste = [1..1000]
def anz_teiler(n): return len(divisors(n))
Liste2 = map(anz_teiler, Liste)
for k in [1..50]:
    print "{0} , {1}".format(k, len(filter(lambda x: x ==
        k, Liste2)))
print divisors(840)
```

# Alternative Schleifenkonstruktionen

- Schleifen abwärts zählen

```
for j in reversed([2,4]):  
    print("{0}, {1}".format(x,x^j))
```

- Schrittweite modifizieren

```
for j in range(3,10,2):  
    print(x,x^j)
```

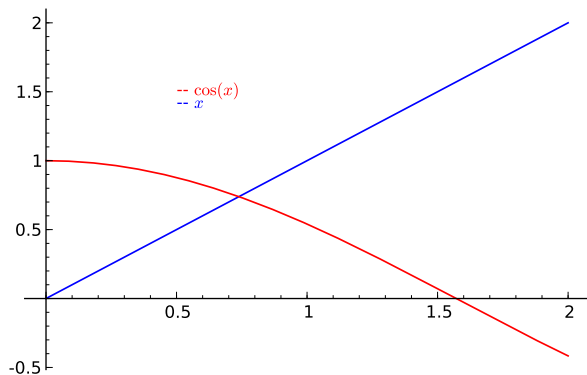


# Fixpunkt

Suche ein  $x_{\text{fix}} \in \mathbb{R}$  so dass

$$x_{\text{fix}} = \cos(x_{\text{fix}})$$

gilt.

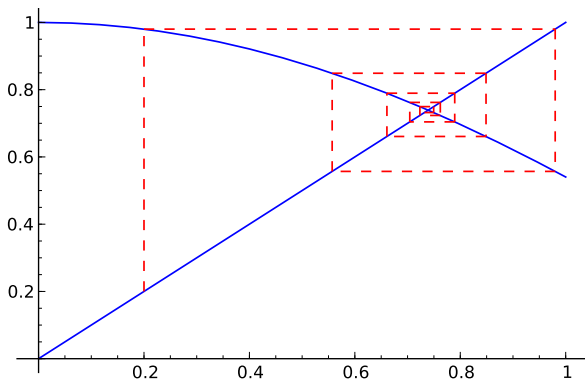


# Fixpunkt-Iteration

Fixpunkt-Iteration

$$x_{k+1} = \cos(x_k)$$

bei geeignetem Startwert  $x_0 = 0.2$ .



```
def fixpunkt(f, In, x0, n):  
    y = [x0]  
    p = plot(f, (In[0], In[1]))  
    p += plot(x, (In[0], In[1]))  
    for i in range(0, n-1):  
        y.append(float(f(y[i])))  
        p += line([ (y[i], y[i]), (y[i], y[i+1]) ],  
                  linestyle='--', color='red')  
        p += line([ (y[i], y[i+1]), (y[i+1], y[i+1]) ],  
                  linestyle='--', color='red')  
    p.show()  
    return y
```

```
fixpunkt(lambda x: cos(x), [0,1], 0.2, 10)  
  
[0.2000000000000000, 0.98006657784124163,  
 0.55696725280964243, 0.84886216565827077,  
 0.66083755111661502, 0.78947843776686832,  
 0.70421571334199318, 0.76211956176066087,  
 0.72337417210557109, 0.74957657633149311,  
 0.73197742525819132]
```