# ErsteSchritteMitPython

November 13, 2021

[1]:
```
1+2+3+4+5+6+7+8+9+10
```

[1]: 55

[2]:
```
for i in range(10):
    print(i,i**3)
```

```
0 0
1 1
2 8
3 27
4 64
5 125
6 216
7 343
8 512
9 729
```

[3]:
```
dummy = 5
```

[4]:
```
dummy**5
```

[4]: 3125

Das war bisher nur ein kleiner Test…

## 1 Erste Schritte mit dem JupyterLab

Wir wollen Python und JupyterLab nutzen, um die Funktion $f(x) = \sin\left(\frac{1}{x}\right)$ zu plotten.

Eingabe für die Formel:

`$f(x) = \sin\left(\frac{1}{x}\right)$`

(eine LaTeX-Formel).

Wir laden die Bibliothek für symbolische Rechnungen: **sympy**.

[5]:
```
import sympy as sy
import math as m
```

Sowohl `sp.sqrt(2)` als auch `m.sqrt(2)` berechnen $\sqrt{2}$'

[6]:
```
m.sqrt(2)
```

[6]:
```
1.4142135623730951
```

[7]:
```
sy.sqrt(2)
```

[7]: $\sqrt{2}$

Merke: "math" rechnet numerisch, "sympy" rechnet symbolisch.

### 1.1 Symbolisches Rechnen mit sympy

**Ziel:** Wir möchten gerne den Grenzwert

$$\lim_{x\to 0}\frac{\sin(x)}{x}=1$$

ausrechnen. Die vorige abgesetzte LaTeX-Formel wird als

```
$$\lim_{x\to 0}\frac{\sin(x)}{x}=1$$
```

eingegeben.

Für den Grenzwert einer Funktion $f(x)$ für $x$ gegen $x0$ ist die sympy-Routine

```
sy.limit(f(x), x, x0)
```

zuständig. Wir testen das:

Die Variable $x$ sollte zunächst als symbolische Variable definiert werden, sonst gibt es eine Fehlermeldung:

[9]:
```
sy.limit(sy.sin(x)/x, x, 0)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/var/folders/_j/x3kvwld16x1885ns96zyqdb00000gn/T/ipykernel_5742/2862830907.py in
 ↪<module>
----> 1 sy.limit(sy.sin(x)/x, x, 0)

NameError: name 'x' is not defined
```

Erklären der symbolische Variable $x$

[10]:
```
x = sy.S('x')
```

Jetzt sollte alles klappen:

[11]:
```
sy.limit(sy.sin(x)/x, x, 0)
```

[11]: 1

Hierbei **muss** man den "symbolischen Sinus aus der Bibliothek sympynehmen!

[13]:
```
m.sin(30)
```

[13]:
```
-0.9880316240928618
```

[14]:
```
sy.limit(m.sin(x)/x, x, 0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/var/folders/_j/x3kvwld16x1885ns96zyqdb00000gn/T/ipykernel_5742/899962075.py in␣
 ↪<module>
----> 1 sy.limit(m.sin(x)/x, x, 0)

~/opt/anaconda3/lib/python3.8/site-packages/sympy/core/expr.py in __float__(self)
    357         if result.is_number and result.as_real_imag()[1]:
    358             raise TypeError("can't convert complex to float")
--> 359         raise TypeError("can't convert expression to float")
    360 
    361     def __complex__(self):

TypeError: can't convert expression to float
```

Alles in Ordnung mit sy.sin

[15]:
```
sy.limit(sy.sin(x)/x, x, 0)
```

[15]: 1

## 1.2 Zeichnen von Funktionen

Wir wollen die Funktion

$$f(x) = \frac{\sin(x)}{x}$$

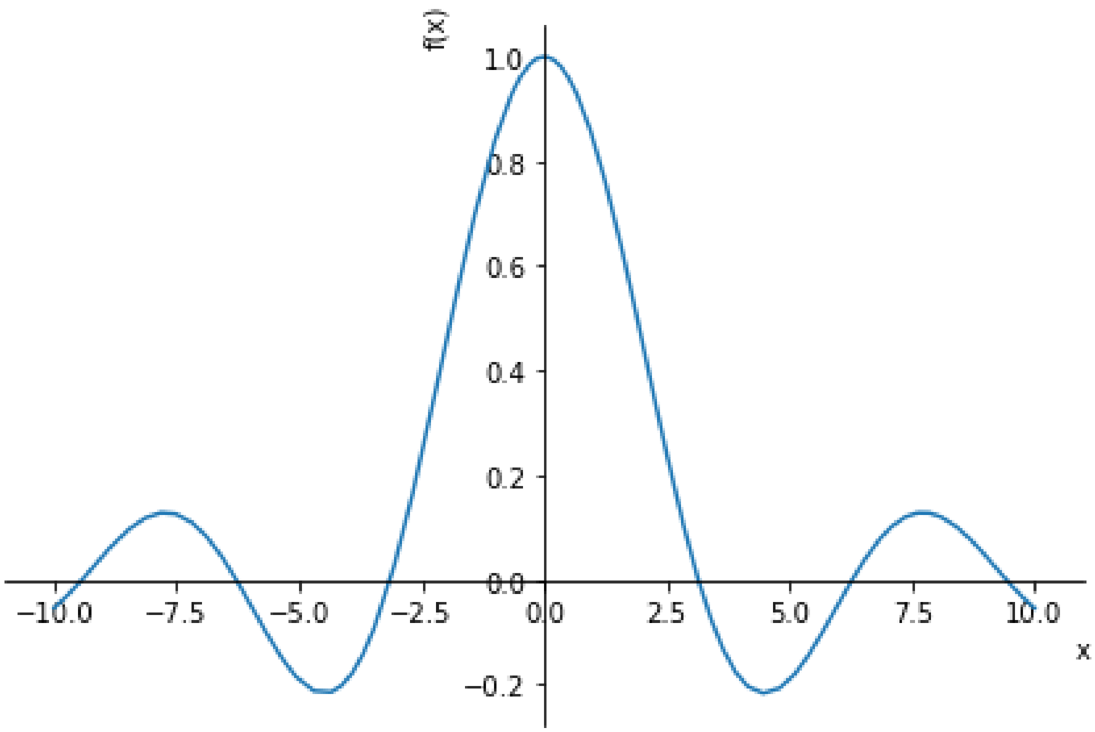
zeichnen.

Zuständig für das Zeichnen einer (symbolischen) Funktion $f(x)$ ist der Befehl

`sy.plot(f(x))`

In unserem Fall also:

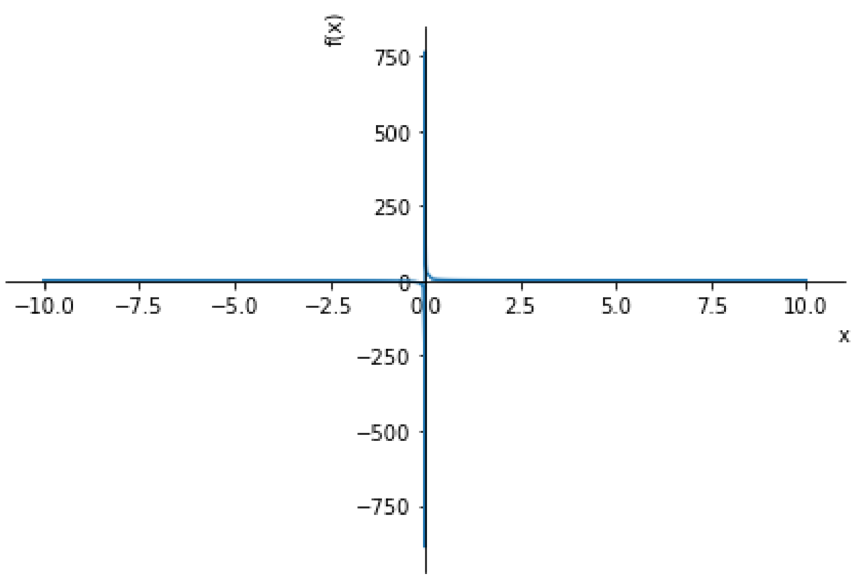
[71]:
```
sy.plot(sy.sin(x)/x)
```




[71]: <sympy.plotting.plot.Plot at 0x7f7f595932e0>

Das klappt im Allgemeinen auch recht gut. Allerdings ist in vielen Fällen ein wenig Handarbeit angesagt. So wird die Hyperbel $f(x) = 1/x$ nicht gerade überzeugend dargestellt:

[72]:
```
sy.plot(1/x)
```

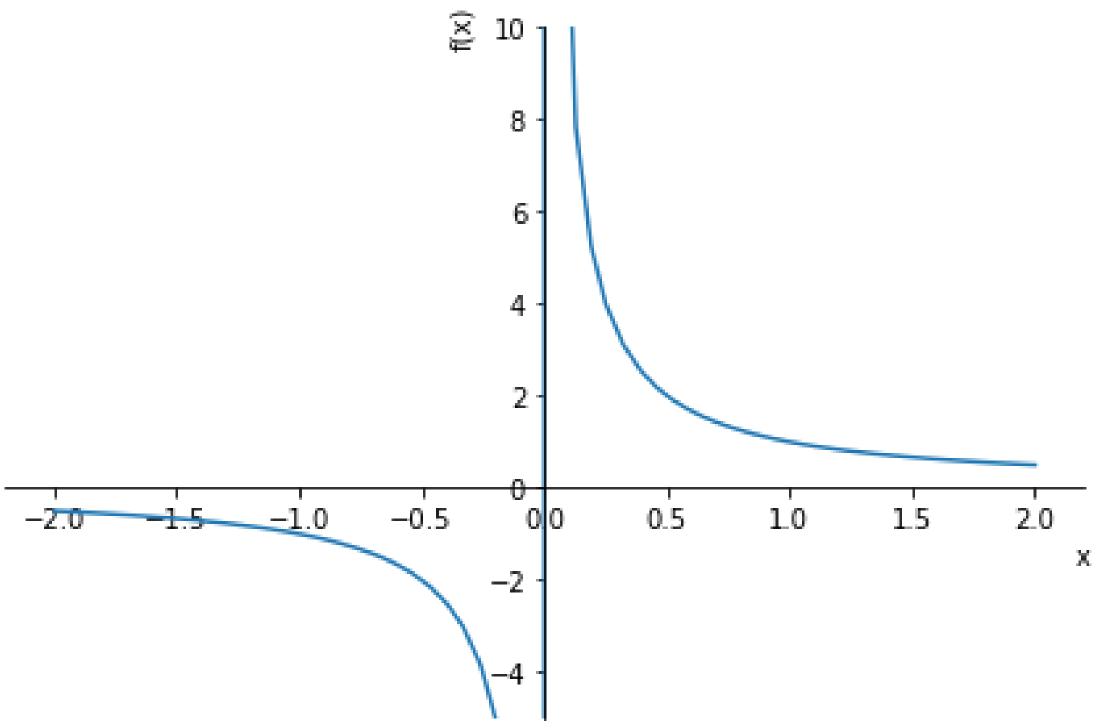


[72]: <sympy.plotting.plot.Plot at 0x7f7f4b440d00>

sympy nimmt standardmäßig das Intervall $[-10, 10]$ als Bereich für die $x$-Werte und passt den $y$-Bereich hieran an. Wir hätten unsere Hyperbel aber lieber für $-2 \leq x \leq 2$, wobei die $y$–Achse auf den Bereich $-5 \leq y \leq 10$ eingeschränkt werden soll.

Für die Einschränkung der x-Werte ist die Option `(x,xmin,xmax)` zuständig; für die Einschränkung der y-Achse nimmt man die Option `ylim = (ymin,ymax)` hinzu.

[78]:
```
sy.plot(1/x, (x,-2,2), ylim = (-5,10))
```




[78]: <sympy.plotting.plot.Plot at 0x7f7f1a306940>

### 1.3 Eine hässliche Funktion

Wir wollen jetzt die Funktion

$$f(x) = \sin \frac{1}{x}$$

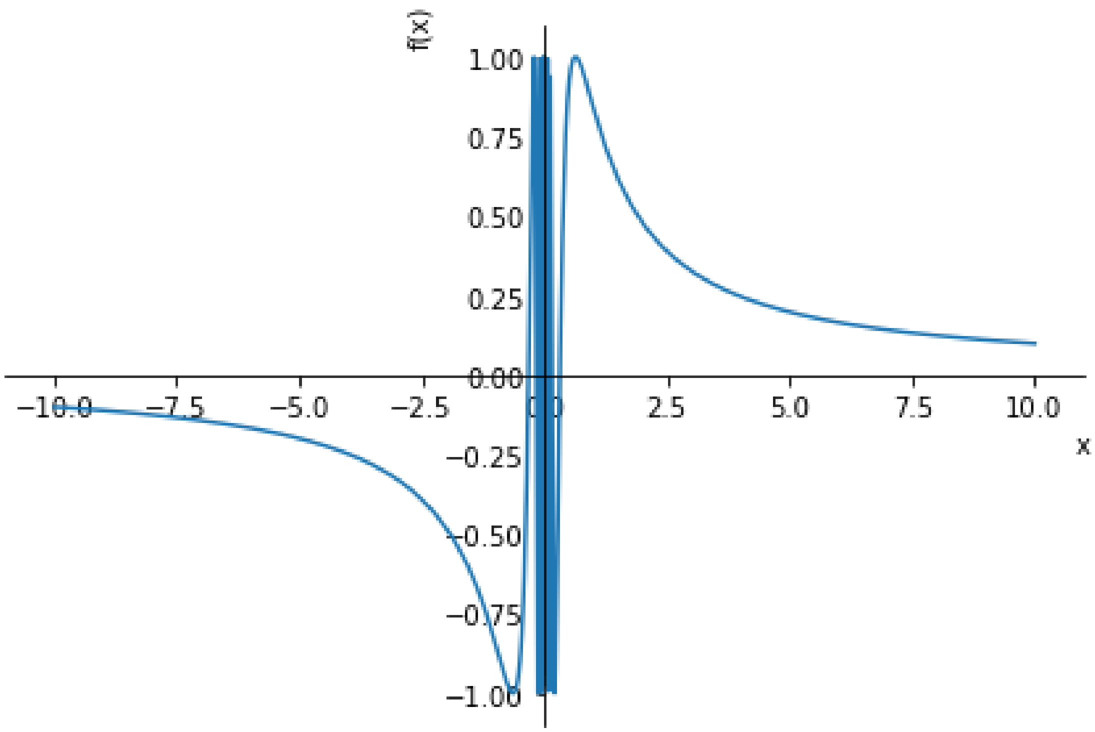
zeichnen. Dabei interessiert uns vor allem der Punkt $x = 0$ (denn dort wird es spannend). Sehen wir uns zunächst an, was sympy zum Grenzwert sagt:

[80]:
```
sy.limit(sy.sin(1/x), x, 0)
```

[80]: $\langle -1, 1 \rangle$

Hm, das scheint etwas merkwürdig zu sein! Wie also sieht die Funktion um $x = 0$ aus? Wir gehen zunächst naiv an die Sache ran:

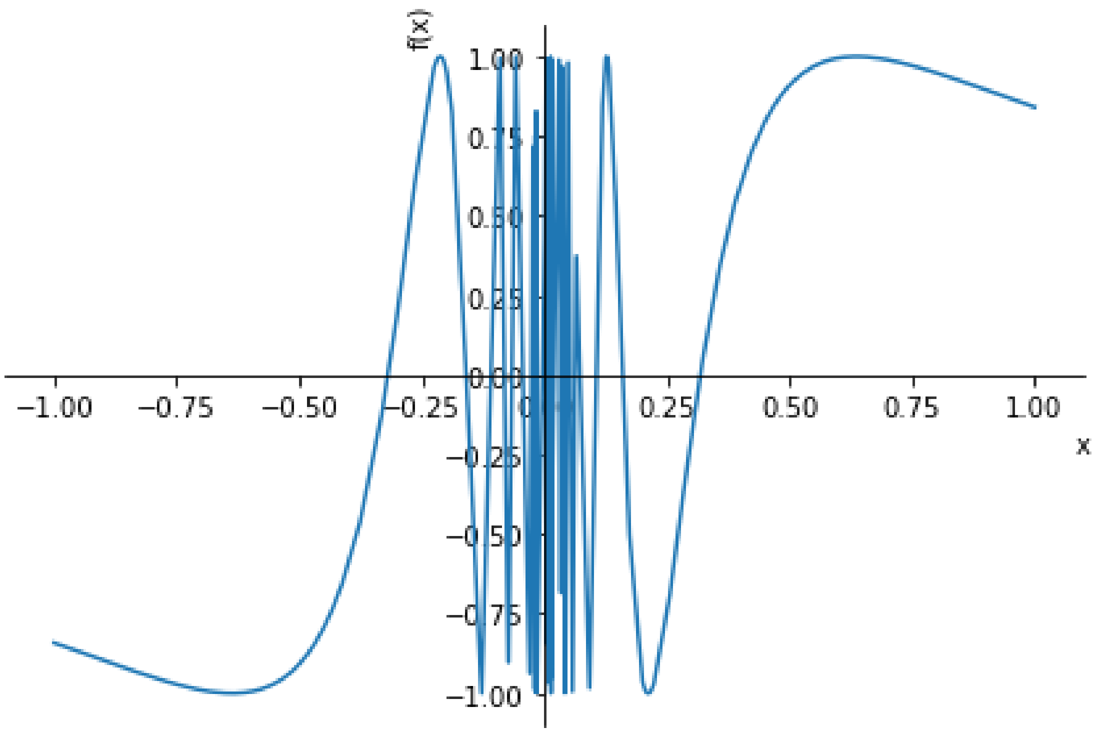
[81]:
```
sy.plot(sy.sin(1/x))
```




[81]: <sympy.plotting.plot.Plot at 0x7f7f597de400>

Offenbar würde $-1 \leq x \leq 1$ hier der bessere $x$-Bereich sein:

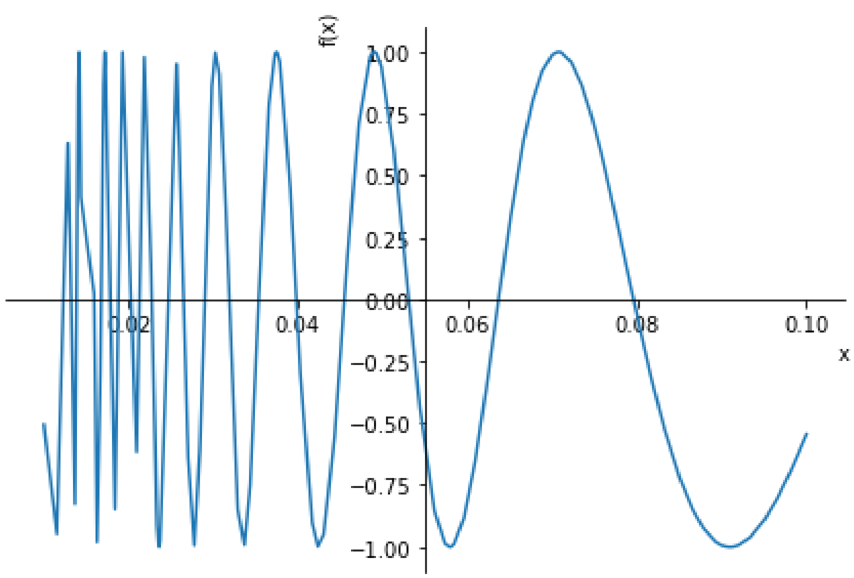
[82]:
```
sy.plot(sy.sin(1/x), (x,-1,1))
```




[82]: <sympy.plotting.plot.Plot at 0x7f7f29514d30>

Das zappelt arg schlimm herum. Versuchen wir uns also an einer Darstellung rechts von der 0:

[83]:
```
sy.plot(sy.sin(1/x), (x,0.01,0.1))
```



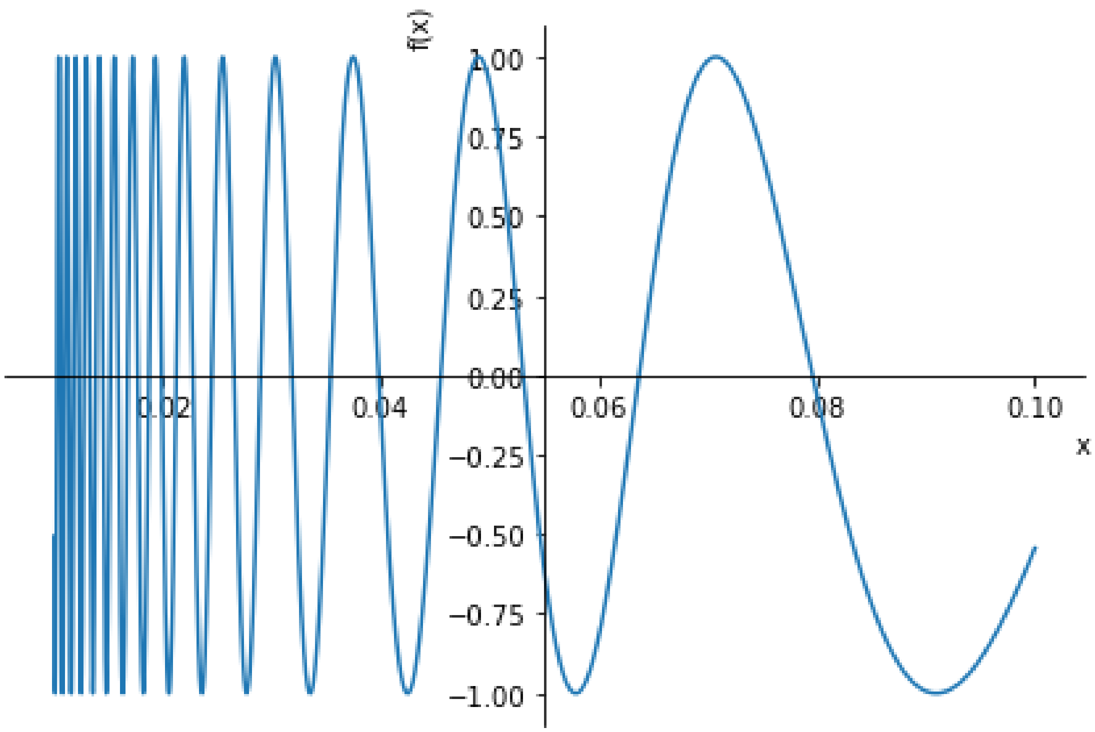
[83]: <sympy.plotting.plot.Plot at 0x7f7f59807580>

Das ist nicht überzeugend - die Maxima sollten bei $y = 1$ liegen. Schließlich kann der Sinus nicht größer werden, und es gilt

$$\sin\frac{1}{x_n} = 1 \text{ für } x_n = \frac{1}{\frac{\pi}{2} + 2n\pi} \text{ mit } x \in \mathbb{N}.$$

Daher müssen wir `sympy`zwingen, für die Darstellung mehr $x$-Werte zu nehmen, wofür wir dem plot-Befehl einige Optionen mit auf dem Weg geben müssen. Zunächst einmal soll `plot` die $x$-Werte für die Zeichnung nicht selbst auswählen. Dies geschieht durch die option `adaptive = False`: Die $x$-Werte der Punkte der Zeichung werden nun im gleichen Abstand verteilt. Außerdem sollen ausreichend viele Punkte gezeichnet werden, was durch die Option `nb_of_points = n` (mit einer recht großen Zahl `n`) erzwungen wird:

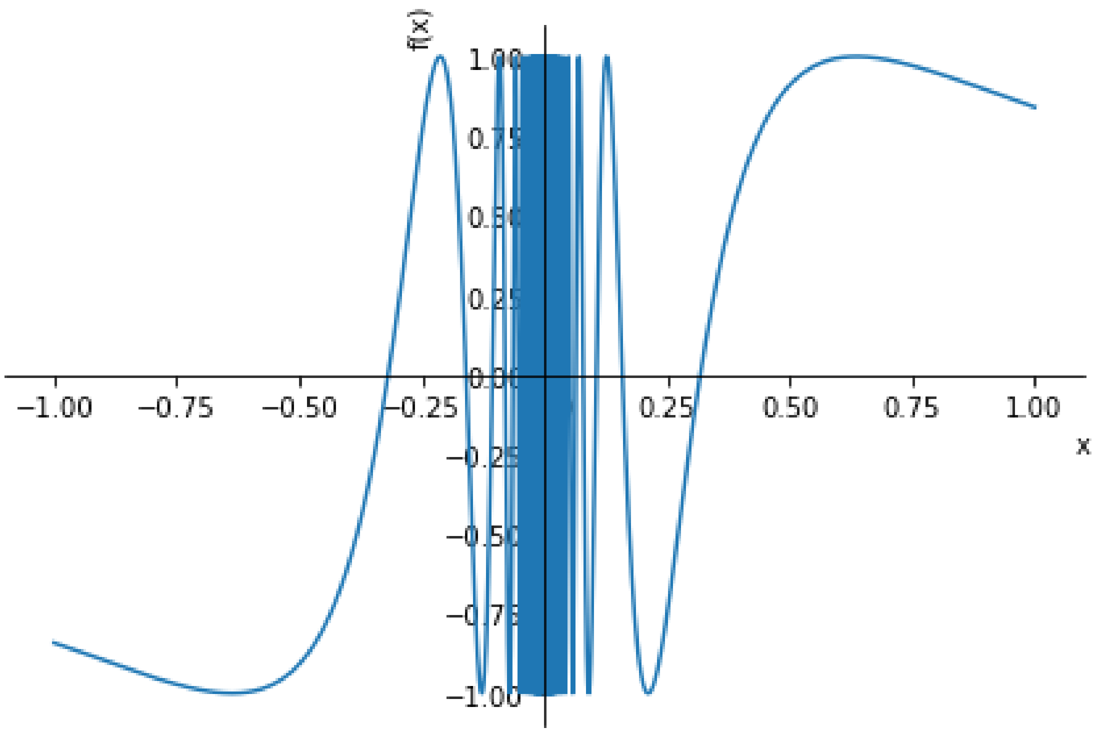
[94]:
```
sy.plot(sy.sin(1/x), (x,0.01,0.1), adaptive = False, nb_of_points = 100000)
```




[94]: <sympy.plotting.plot.Plot at 0x7f7f1ac3c6a0>

Das ist schon viel besser und zeigt das "Herumzappeln" der Funktion nahe $x = 0$ wesentlich besser.

[99]:
```
sy.plot(sy.sin(1/x), (x,-1,1), adaptive = False, nb_of_points = 100000)
```




[99]: `<sympy.plotting.plot.Plot at 0x7f7f1b8f0ee0>`

So sehen wir das Verhalten der Funktion $f(x) = \sin(1/x)$ bei $x = 0$ deutlich besser: Diese schwankt zwischen $-1$ und 1 bei $x = 0$ - und zwar unabhängig davon, *wie* nahe wir uns an dieser Stelle befinden. Mit anderen Worten: Für jedes $\delta > 0$ nimmt $f(x) = \sin(1/x)$ auf dem Intervall $(-\delta, \delta)$ *sämtliche* Werte zwischen $-1$ und 1 an. Kein Wunder, dass der Grenzwert nicht existiert und wir die Funktion nicht stetig in $x = 0$ ergänzen können. Aber eine "Sprungstelle" haben wir dort sicher auch nicht...

[100]:
```
sy.limit(sy.sin(1/x), x, 0)
```

[100]: $\langle -1, 1 \rangle$

[ ]:
```
```