lektion7

January 23, 2020

Table of Contents

1 Lektion 7

[1]:
```
from sympy import *
from IPython.display import display
init_printing()
import matplotlib.pyplot as plt
import numpy as np
#%matplotlib notebook
%matplotlib inline
%config InlineBackend.figure_format = 'svg'
x, y, z, a, b, c, d = symbols('x y z a b c d')
```

1.1 Polynome I

[2]:
```
p = x**2 + 3
q = 2*x + 2
d, r = div(p, q)
display(p)
```

```
display(q)
display(d)
r
```

$x^2 + 3$

$2x + 2$

$\frac{x}{2} - \frac{1}{2}$

[2]: 4

[3]:
```
q = x + y**2
p = 1
degree(q, x), degree(q, y) , degree(p, x)
```

[3]: $(1,\ 2,\ 0)$

[4]:
```
degree(p, x)
```

[4]: 0

1.2 Polynome II

[5]:
```
p = poly(x**2 + 3, domain = QQ)
p
```

[5]: $\text{Poly}\left(x^2 + 3, x, domain = \mathbb{Q}\right)$

[6]:
```
degree(p) # per Default wird der Grad bzgl. x ausgegeben
```

[6]: 2

[7]:
```
q = poly(2*x + 2, x, domain = QQ)
q
```

[7]: $\text{Poly}\left(2x + 2, x, domain = \mathbb{Q}\right)$

1.2.1 Polynomdivision

[8]:
```
div(p, q)
```

[8]: $\left(\text{Poly}\left(\frac{1}{2}x - \frac{1}{2}, x, domain = \mathbb{Q}\right),\ \text{Poly}\left(4, x, domain = \mathbb{Q}\right)\right)$

[9]:
```
quo(p, q)
```

[9]: $\text{Poly}\left(\frac{1}{2}x - \frac{1}{2}, x, domain = \mathbb{Q}\right)$

[10]:
```
rem(p, q)
```

[10]: $\text{Poly}\left(4, x, domain = \mathbb{Q}\right)$

[11]:
```
p = poly(x**2 +3, domain=ZZ)
p
```

[11]: $\text{Poly}\left(x^2 + 3, x, domain = \mathbb{Z}\right)$

[12]:
```
q = poly(2*x+2,x,domain = ZZ)
q
```

[12]: $\text{Poly}\left(2x + 2, x, domain = \mathbb{Z}\right)$

[13]:
```
div(p, q, domain = ZZ)
```

[13]: $\left(\text{Poly}\left(0, x, domain = \mathbb{Z}\right),\ \text{Poly}\left(x^2 + 3, x, domain = \mathbb{Z}\right)\right)$

1.2.2 Polynomgrad

[14]:
```
p = poly(2*x**3 + 3*x*y)
degree(p)
```

[14]: 3

[15]:
```
degree(p, x)
```

[15]: 3

[16]:
```
degree(p, y)
```

[16]: 1

[17]:
```
r = poly(0, x, domain = QQ)
#r = poly(1,x,domain=QQ)
r
```

[17]: $\text{Poly}\left(0, x, domain = \mathbb{Q}\right)$

[18]:
```
degree(r) # Achtung
```

[18]: $-\infty$

1.2.3 Koeffizienten von Polynomen

[19]:
```
p = prod([x-j for j in range(5)])
p
```

[19]: $x\left(x-4\right)\left(x-3\right)\left(x-2\right)\left(x-1\right)$

[20]:
```
p.expand()
```

[20]: $x^5 - 10x^4 + 35x^3 - 50x^2 + 24x$

[21]:
```
Poly(p).all_coeffs()
```

[21]: $[1,\ -10,\ 35,\ -50,\ 24,\ 0]$

[22]:
```
p.expand().coeff(x, 4)
```

[22]: -10

[23]:
```
 Poly(p).coeff_monomial(x**4)
```

[23]: -10

1.2.4 Wurzeln von Polynomen

[24]:
```
p = x**4 - x**2
roots(p)
```

[24]: $\{-1 : 1,\ 0 : 2,\ 1 : 1\}$

[25]:
```
roots(a*x**2 + b*x + c, x)
```

[25]: $\left\{-\frac{b}{2a} - \frac{\sqrt{-4ac+b^2}}{2a} : 1,\ -\frac{b}{2a} + \frac{\sqrt{-4ac+b^2}}{2a} : 1\right\}$

[26]:
```
p = Poly(a*x**3+b*x**2+c*x+d, x)
roots(p, x)
```

[26]: $\left\{ - \frac{-\frac{3(2c+1)}{2a} + \frac{b^2}{a^2}}{3\sqrt[3]{\frac{\sqrt{-4\left(-\frac{3(2c+1)}{2a}+\frac{b^2}{a^2}\right)^3+\left(-\frac{27}{2a}-\frac{9b(2c+1)}{2a^2}+\frac{2b^3}{a^3}\right)^2}}{2} - \frac{27}{4a} - \frac{9b(2c+1)}{4a^2} + \frac{b^3}{a^3}}} - \frac{\sqrt[3]{\frac{\sqrt{-4\left(-\frac{3(2c+1)}{2a}+\frac{b^2}{a^2}\right)^3+\left(-\frac{27}{2a}-\frac{9b(2c+1)}{2a^2}+\frac{2b^3}{a^3}\right)^2}}{2} - \right.}{3}$

[27]:
```
p = Poly([1, -5, 4, 4, 3, 9], x) # Polynom aus der Liste der Koeffizienten
p
```

[27]: $\mathrm{Poly}\left(x^5 - 5x^4 + 4x^3 + 4x^2 + 3x + 9, x, domain = \mathbb{Z}\right)$

[28]:
```
real_roots(p, x)
```

[28]: $[-1,\ 3,\ 3]$

[29]:
```
roots(p, x)
```

[29]: $\{-1 : 1,\ 3 : 2,\ -i : 1,\ i : 1\}$

[30]:
```
p = Poly(x**5 - 20*x + 7)
p
```

[30]: $\text{Poly}\left(x^5 - 20x + 7, x, domain = \mathbb{Z}\right)$

[31]:
```
roots(p) # das klappt so nicht
```

[31]: $\{\}$

[ ]:

[32]:
```
for r in nroots(p):
    display(r)  # aber numerisch geht es
```

-2.19444446392152

0.350263599776741

2.01630911327803

$-0.0860641245666237 - 2.12350978358086i$

$-0.0860641245666237 + 2.12350978358086i$

1.3 Lösen von Gleichungen (solveset)

[33]:
```
gl = Eq((x-1)**2, 4-x)
gl
```

[33]: $(x-1)^2 = 4 - x$

[34]:
```
gl.lhs
```

[34]: $(x-1)^2$

[35]:
```
gl.rhs
```

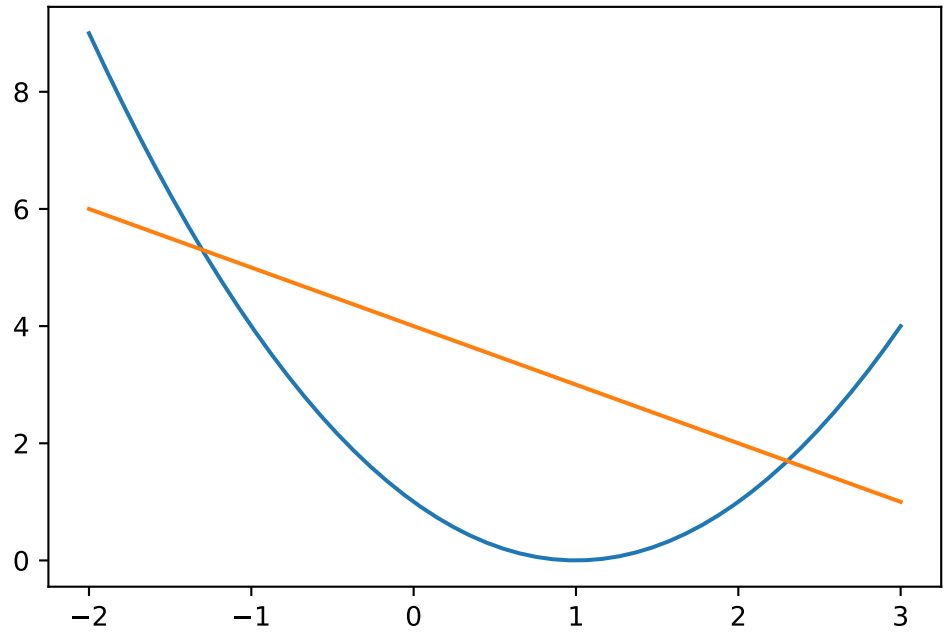
[35]: $4 - x$

[36]:
```
Lsgn = solveset(gl, x)
Lsgn
```

[36]:

$$\left\{\frac{1}{2} - \frac{\sqrt{13}}{2}, \frac{1}{2} + \frac{\sqrt{13}}{2}\right\}$$

[37]:
```
fig = plt.figure()
ax = fig.gca()
xn = np.linspace(-2,3)
ax.plot(xn,lambdify(x,gl.lhs)(xn))
ax.plot(xn,lambdify(x,gl.rhs)(xn))
plt.show()
```



[38]:
```
# Test durch einsetzen
for lsg in Lsgn:
    display(gl.subs(x, lsg))
```

True

True

[39]:
```
solveset((x-1)**2 - 4+x, x)  # loest (x-1)**2-4+x = 0
```

[39]:
$$\left\{\frac{1}{2} - \frac{\sqrt{13}}{2}, \frac{1}{2} + \frac{\sqrt{13}}{2}\right\}$$

[40]:
```
glt = Eq(sin(x), cos(x))
glt
```

[40]:
$$\sin(x) = \cos(x)$$

[41]:
```
Lsg = solveset(glt)
Lsg
```

[41]:
$$\left\{2n\pi + \frac{5\pi}{4} \mid n \in \mathbb{Z}\right\} \cup \left\{2n\pi + \frac{\pi}{4} \mid n \in \mathbb{Z}\right\}$$

[42]:
```
for i, lsg in zip(range(7), Lsg):
    display(lsg)
```

$$\frac{5\pi}{4}$$
$$\frac{\pi}{4}$$
$$\frac{13\pi}{4}$$
$$\frac{9\pi}{4}$$
$$-\frac{3\pi}{4}$$
$$-\frac{7\pi}{4}$$
$$\frac{21\pi}{4}$$

[43]:
```
print(Lsg)
```

```
Union(ImageSet(Lambda(_n, 2*_n*pi + 5*pi/4), Integers), ImageSet(Lambda(_n,
2*_n*pi + pi/4), Integers))
```

[44]:
```
glt = Eq(sin(2*x), cos(x))
glt
```

[44]:
$$\sin(2x) = \cos(x)$$

[45]:
```
Lsg = solveset(glt, x)
Lsg
```

[45]:
$$\left\{2n\pi + \frac{\pi}{2} \mid n \in \mathbb{Z}\right\} \cup \left\{2n\pi + \frac{3\pi}{2} \mid n \in \mathbb{Z}\right\} \cup \left\{2n\pi + \frac{5\pi}{6} \mid n \in \mathbb{Z}\right\} \cup \left\{2n\pi + \frac{\pi}{6} \mid n \in \mathbb{Z}\right\}$$

[46]:
```
Lsg = solveset(glt, x, Interval(-5,10)) #Loesung in einem Intervall
Lsg
```

[46]:
$$\left\{-\frac{3\pi}{2}, -\frac{7\pi}{6}, -\frac{\pi}{2}, \frac{\pi}{6}, \frac{\pi}{2}, \frac{5\pi}{6}, \frac{3\pi}{2}, \frac{13\pi}{6}, \frac{5\pi}{2}, \frac{17\pi}{6}\right\}$$

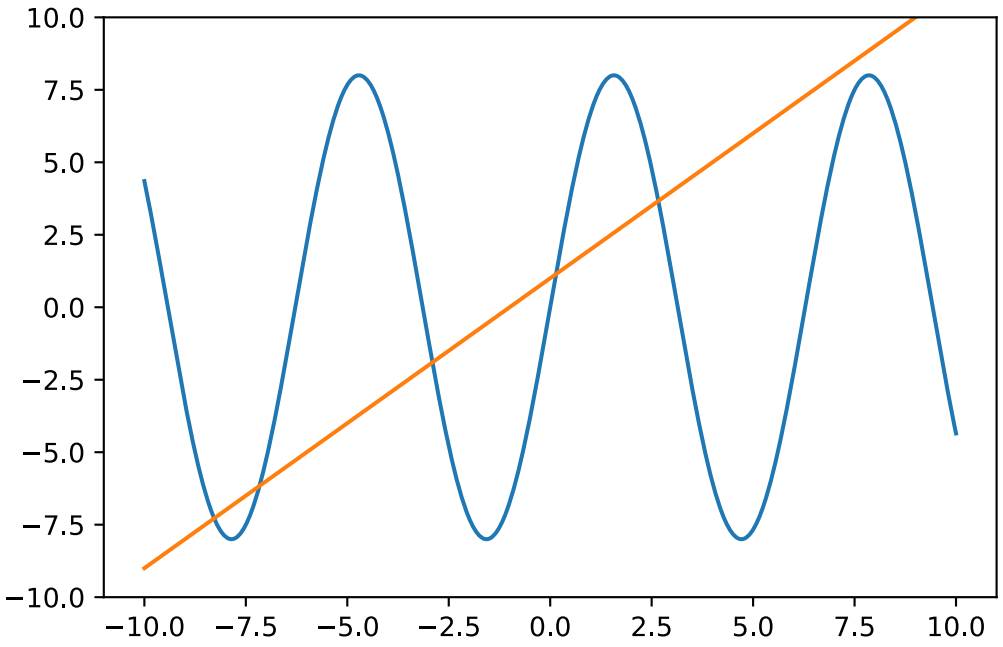
[47]:
```
glt = Eq(8*sin(x), x+1)
```

[48]:
```
Lsgt = solveset(glt)
Lsgt
```

[48]:
$$\{x \mid x \in \mathbb{C} \wedge -x + 8\sin(x) - 1 = 0\}$$

[49]:
```
gl = glt.lhs
gr = glt.rhs
xn = np.linspace(-10, 10, 500)
gln = lambdify(x, gl)
grn = lambdify(x, gr)
fig = plt.figure()
plt.plot(xn, gln(xn), xn, grn(xn))
plt.ylim((-10, 10))
```

[49]:
$$(-10,\ 10)$$



[50]:
```
solveset(exp(x), x) # exp(x) == 0
```

[50]:
$$\emptyset$$

[51]:
```
solveset(Eq(exp(x**2), 1)) # exp(x**2) = 1
```

[51]:
$$\left\{x \mid x \in \mathbb{C} \wedge e^{x^2} - 1 = 0\right\}$$

[52]:
```
solveset(exp(x)+x, x)
```

[52]:
$$\{x \mid x \in \mathbb{C} \wedge x + e^{x} = 0\}$$

[53]:
```
solveset(x**2+1, x, domain=Reals)
```

[53]:
$$\emptyset$$

[54]:
```
solveset(log(x+1)+log(x-2), x) # log(x+1)+log(x-2) = 0
```

[54]:
$$\left\{\frac{1}{2} - \frac{\sqrt{13}}{2}, \frac{1}{2} + \frac{\sqrt{13}}{2}\right\}$$

[55]:
```
solveset(Eq(5**(x-2), 3**(2*x+1)), x) # 5**(x-2) = 3**(2*x+1)
```

[55]:
$$\left\{-\frac{\log(3) + 2\log(5)}{-\log(5) + 2\log(3)}\right\}$$

1.3.1 Wurzeln von Polynomen

[56]:
```
p = Eq(x**5 - 20*x + 7, 0)
Lsg = solveset(p)
Lsg
```

[56]:
$$\left\{\text{CRootOf}\left(x^5 - 20x + 7, 0\right), \text{CRootOf}\left(x^5 - 20x + 7, 1\right), \text{CRootOf}\left(x^5 - 20x + 7, 2\right), \text{CRootOf}\left(x^5 - 20x + 7, 3\right), \text{CR}\right.$$

[57]:
```
Lsg = solveset(p, domain = Reals)
Lsg
```

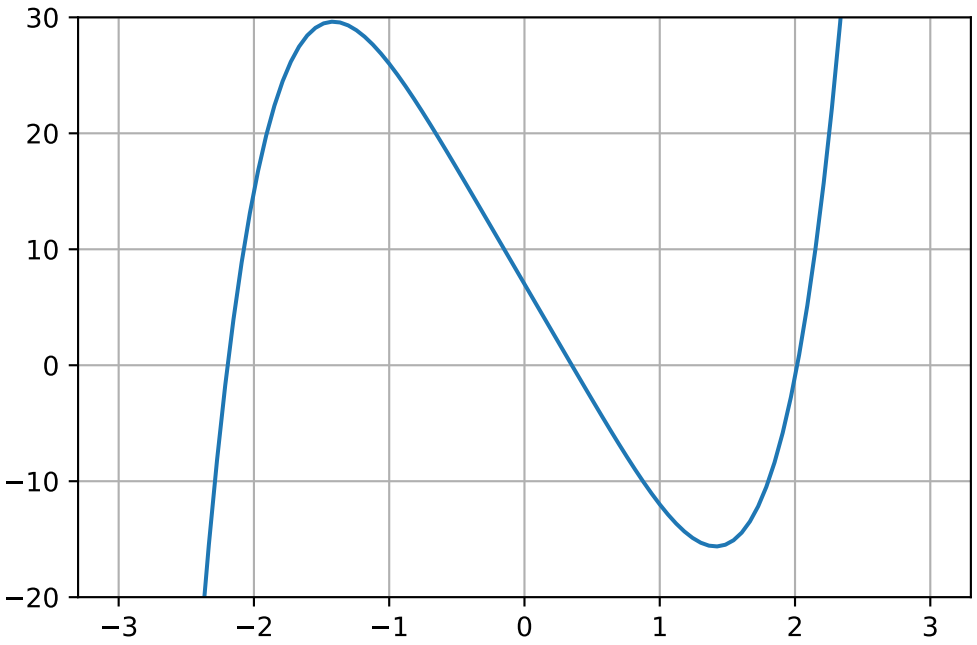
[57]:
$$\left\{\text{CRootOf}\left(x^5 - 20x + 7, 0\right), \text{CRootOf}\left(x^5 - 20x + 7, 1\right), \text{CRootOf}\left(x^5 - 20x + 7, 2\right)\right\}$$

[58]:
```
[l.n() for l in Lsg]
```

[58]:
$$[-2.19444446392152,\ 0.350263599776741,\ 2.01630911327803]$$

[59]:
```
fig = plt.figure()
ax = fig.gca()
xn = np.linspace(-3, 3, 100)
ax.plot(xn, lambdify(x, p.lhs)(xn))
ax.grid()
ax.set_ylim(-20, 30)
```

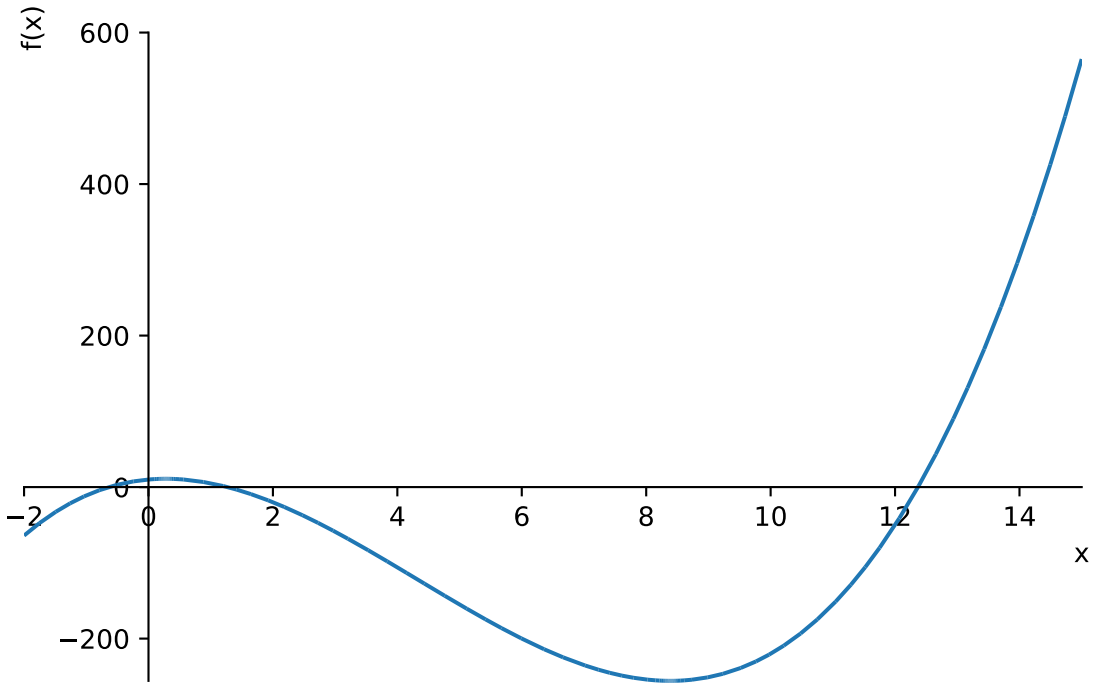
[59]:
$$(-20,\ 30)$$



[60]:
```
p = x**3 - 13*x**2 + 7*x + 10
gl = Eq(p, 0)
lsgn = solveset(gl)
lsgn
```

[60]:
$$\left\{\frac{13}{3} + \frac{4\sqrt{37}\cos\left(\frac{\operatorname{atan}\left(\frac{3\sqrt{227127}}{3305}\right)}{3}\right)}{3}, -\frac{\sqrt{37}\cos\left(\frac{\operatorname{atan}\left(\frac{3\sqrt{227127}}{3305}\right)}{3}\right)}{3} + \frac{148\,\mathrm{re}\left(\frac{1}{\left(-\frac{1}{2}-\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{227127}i}{18}}}\right)}{9} + \frac{\sqrt{111}\sin\left(\right.\right.$$

[61]:
```
plot(p,(x,-2,15))
```



[61]:
```
<sympy.plotting.plot.Plot at 0x7f15ca966e90>
```

[62]:
```
solveset(gl, domain=Reals)
```

[62]:
$$\mathbb{R} \cap \left\{ \frac{13}{3} + \frac{4\sqrt{37}\cos\left(\frac{\operatorname{atan}\left(\frac{3\sqrt{227127}}{3305}\right)}{3}\right)}{3}, -\frac{\sqrt{37}\cos\left(\frac{\operatorname{atan}\left(\frac{3\sqrt{227127}}{3305}\right)}{3}\right)}{3} + \frac{148\,\mathrm{re}\left(\frac{1}{\left(-\frac{1}{2}-\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{227127}i}{18}}}\right)}{9} + \frac{\sqrt{111}\,\mathrm{si}}{}\right.$$

[63]:
```
[p.subs(x,l).simplify() for l in lsgn]
```

[63]:
$$\left[-\frac{592\sqrt{37}\cos\left(\frac{\operatorname{atan}\left(\frac{3\sqrt{227127}}{3305}\right)}{3}\right)}{9} - \frac{3305}{27} + \frac{2368\sqrt{37}\cos^3\left(\frac{\operatorname{atan}\left(\frac{3\sqrt{227127}}{3305}\right)}{3}\right)}{27},\ 0,\ 0 \right]$$

[64]:
```
[p.subs(x,l).simplify().n() for l in lsgn]
```

[64]:
$$\left[-6.0 \cdot 10^{-123},\ 0,\ 0\right]$$

[65]:
```
wurzeln = roots(p)
wurzeln
```

[65]:
$$\left\{\frac{13}{3}+\left(-\frac{1}{2}-\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{227127}i}{18}}+\frac{148}{9\left(-\frac{1}{2}-\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{227127}i}{18}}}:1,\ \frac{13}{3}+\frac{148}{9\left(-\frac{1}{2}+\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{}}{}}}\right.$$

[66]:
```
[p.subs(x, w).simplify() for w in wurzeln]
```

[66]:
$$[0,\ 0,\ 0]$$

1.4 Lösen von Gleichungen (solve)

[67]:
```
glg = Eq(x**5 - 20*x + 7, 0)
lsg = solve(glg, x)
lsg
```

[67]:
$$\left[\text{CRootOf}\left(x^5-20x+7,0\right),\ \text{CRootOf}\left(x^5-20x+7,1\right),\ \text{CRootOf}\left(x^5-20x+7,2\right),\ \text{CRootOf}\left(x^5-20x+7,3\right),\right.$$

[68]:
```
p = x**3 - 13*x**2 + 7*x + 10
gl = Eq(p,0)
lsgn = solve(gl) # das ist die Darstellung der Loesung wie bei roots
lsgn
```

[68]:
$$\left[\frac{13}{3}+\left(-\frac{1}{2}-\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{227127}i}{18}}+\frac{148}{9\left(-\frac{1}{2}-\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{227127}i}{18}}},\ \frac{13}{3}+\frac{148}{9\left(-\frac{1}{2}+\frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{3305}{54}+\frac{\sqrt{2271}}{18}}}\right.$$

[69]:
```
solve(Eq(sin(x),cos(x)),x) # hier fehlen jede Menge Lösungen;  hier ist solve␣
↪schlechter als solveset
```

[69]:
$$\left[-\frac{3\pi}{4},\ \frac{\pi}{4}\right]$$

[70]:
```
solve(Eq(8*sin(x), x+1), x)
```

```
␣
↪---------------------------------------------------------------------------

NotImplementedError                       Traceback (most recent call last)

<ipython-input-70-7f6d2f67274b> in <module>
----> 1 solve(Eq(8*sin(x), x+1), x)

/local/schaedle/miniconda/lib/python3.7/site-packages/sympy/solvers/
↪solvers.py in solve(f, *symbols, **flags)
   1172     ␣
↪#########################################################################
```

```
   1173     if bare_f:
-> 1174         solution = _solve(f[0], *symbols, **flags)
   1175     else:
   1176         solution = _solve_system(f, symbols, **flags)


/local/schaedle/miniconda/lib/python3.7/site-packages/sympy/solvers/
↪solvers.py in _solve(f, *symbols, **flags)
   1746
   1747     if result is False:
-> 1748         raise NotImplementedError('\n'.join([msg, not_impl_msg % f]))
   1749
   1750     if flags.get('simplify', True):


NotImplementedError: multiple generators [x, sin(x)]
No algorithms are implemented to solve equation -x + 8*sin(x) - 1
```

[71]:
```
solve(Eq(exp(x**2), 1), x)  # hier ist solve besser als solveset
```

[71]: $[0]$

[72]:
```
solve(Eq(exp(x), 0), x)
```

[72]: $[]$

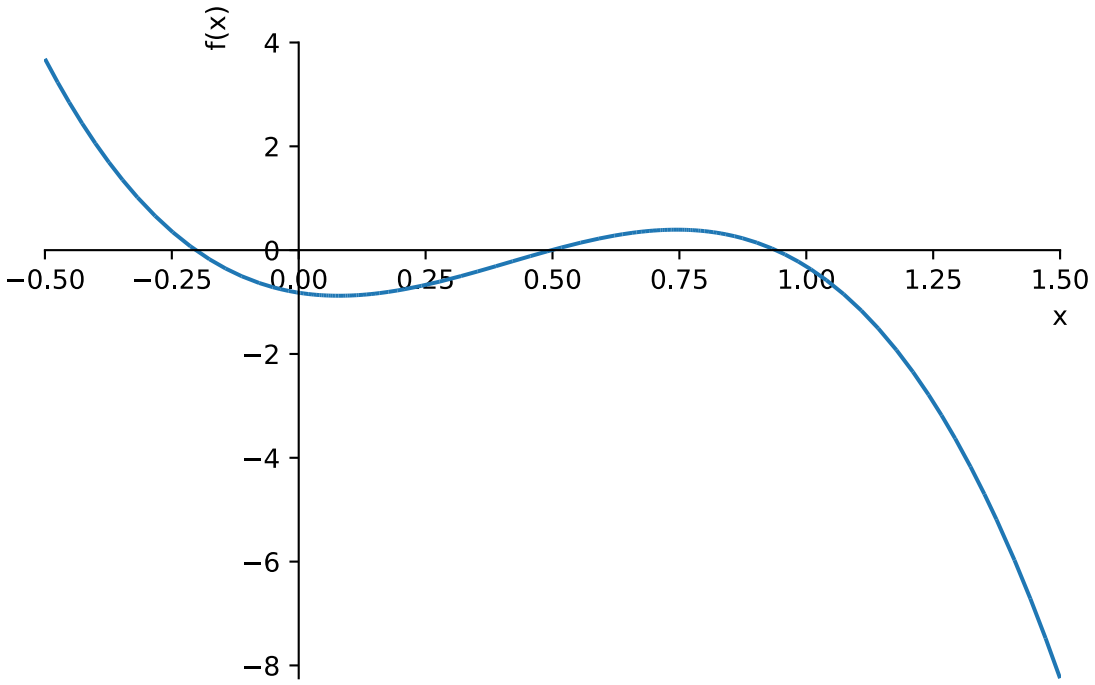
[73]:
```
solve(Eq(2**(2-x), 3**(2*x-1)), x)
```

[73]: $\left[\frac{\log(12)}{\log(18)}\right]$

[74]:
```
solve(exp(x)+x, x)  # hier ist solve besser als solveset
```

[74]: $[-W(1)]$

[75]:
```
n = int(3)
a = Rational(-1, 2)
b = 3
f = 1/(2**n*factorial(n)) *(1-x)**(-a) *(1+x)**(-b) * diff((1-x)**a * (1+x)**b␣
↪* (1-x**2)**n,x, n)
f
f.simplify()
plot(f,(x, -1/2, 3/2))
```



[75]:
```
<sympy.plotting.plot.Plot at 0x7f15d1a7c9d0>
```

[76]:
```
sols = solve(f.simplify())
sols
```

[76]:
$$\left[\frac{7}{17} - \frac{96}{289\left(-\frac{1}{2} - \frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{24192}{63869} + \frac{3456i}{3757}}} - \frac{\left(-\frac{1}{2} - \frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{24192}{63869} + \frac{3456i}{3757}}}{3}, \quad \frac{7}{17} - \frac{\left(-\frac{1}{2} + \frac{\sqrt{3}i}{2}\right)\sqrt[3]{\frac{24192}{63869} + \frac{3456i}{3757}}}{3} - \frac{}{289\left(\right.}\right.$$

[77]:
```
[im(sol).simplify() for sol in sols]
```

[77]:
$$[0,\ 0,\ 0]$$

[78]:
```
[re(sol).simplify() for sol in sols]
```

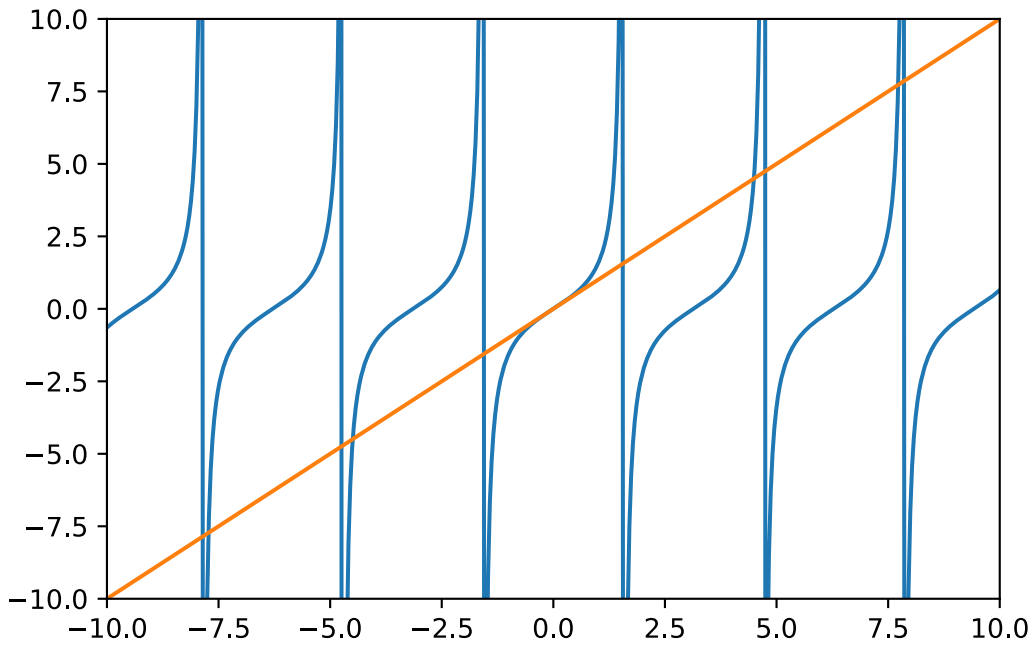
[78]:
$$\left[\frac{8\sqrt{2}\sin\left(-\frac{\operatorname{atan}\left(\frac{17}{7}\right)}{3} + \frac{\pi}{6}\right)}{17} + \frac{7}{17}, \quad \frac{7}{17} + \frac{8\sqrt{2}\sin\left(\frac{\operatorname{atan}\left(\frac{17}{7}\right)}{3} + \frac{\pi}{6}\right)}{17}, \quad -\frac{8\sqrt{2}\cos\left(\frac{\operatorname{atan}\left(\frac{17}{7}\right)}{3}\right)}{17} + \frac{7}{17}\right]$$

1.5 Numerische Lösung von Gleichungen

[79]:
```
def glp(lb, ub, gl, nn = 500):
    ''' Zeichne Gleichung gl in (lb,ub)'''
    fig = plt.figure()
    ax = fig.gca()
    xn = np.linspace(lb, ub, nn)
    ax.plot(xn, lambdify(x, gl.lhs)(xn))
    ax.plot(xn, lambdify(x, gl.rhs)(xn))
    return fig, ax
```

[80]:
```
gl = Eq(tan(x), x)
fig, ax = glp(-10, 10, gl)
ax.set_ylim([-10, 10])
ax.axis([-10, 10, -10, 10])
```

[80]: $[-10,\ 10,\ -10,\ 10]$



[81]:
```
solveset(gl)
```

[81]: $\{x \mid x \in \mathbb{C} \wedge -x + \tan(x) = 0\}$

[82]:
```
nsolve(gl, 1 )
```

[82]: 0.000348227174421857

[83]:
```
nsolve(gl, (np.pi, 1.499*np.pi), solver='bisect')
```

[83]:
4.49340945790906