Algebre

June 22, 2022

1 Un peu d'algèbre avec sympy.

- L'objectif est de résoudre une equation
- de calculer l'image d'un nombre par substitution d'une variable symbolique
- de calculer la dérivée d'une fonction
- de tracer les courbe d'une fonction et de sa dérivée

2 Ressources en français sur sympy:

Documentation

Notebooks de Pierre Proulx * https://pierreproulx.espaceweb.usherbrooke.ca/notebooks/python_2.html * https://pierreproulx.espaceweb.usherbrooke.ca/notebooks/python_4.html * https://pierreproulx.espaceweb.usherbrooke.ca/notebooks/MAO_1.html

Cours de Marc Buffat

Des exemples dans Wikimath, mais attention avec la mauvaise pratique:

```
from sympy import *
```

Dans ce notebook les modules seront importés explicitement.

Les références précédentes sont intéressantes, moins pour les sujets traités (équa diff, intégrales ...) que pour la façon d'utiliser sympy.

3 Résolution d'équations algébriques:

Commençons par nous donner quelques super pouvoirs en important des modules:

[2]:
```python
from sympy.solvers import solve
from sympy import Eq
from sympy import Symbol
#affichage
from sympy import latex, pprint
```

[3]:
```python
from matplotlib import pyplot as plt
```

Le second membre (à droite du signe =) des équations est égal à zéro. Pour résoudre $x^2 = 1$, on tranforme l'équation en $x^2 - 1 = 0$

On écrira le code python suivant:

```
x = Symbol('x')
solve(x**2 - 1, x)
```

x représente la variable, ici c'est un symbole

[4]:
```
x = Symbol('x')
solve(x**2 - 1, x)
```

[4]:
```
[-1, 1]
```

[5]:
```
type(x)
```

[5]:
```
sympy.core.symbol.Symbol
```

On peut quand même définir explicitement les membres gauche et droit d'une équation:

[6]:
```
eq = Eq(x**2,1)
eq
```

[6]: $x^2 = 1$

et résoudre l'équation *eq*:

[7]:
```
solutions = solve(eq,x)
solutions
```

[7]:
```
[-1, 1]
```

3.0.1 Compararons les méthodes d'affichage:

[8]:
```
from sympy import init_printing
init_printing(use_unicode=True)
```

[9]:
```
solutions = solve(x**2 -3, x)
print(solutions)
pprint(solutions)
solutions
```

```
[-sqrt(3), sqrt(3)]
[-3, 3]
```

[9]: $\left[-\sqrt{3}, \sqrt{3}\right]$

pour Python, les solutions obtenues ne sont pas des nombres, mais des symboles:

[10]:
```
print('La solution:', solutions[0],' est de type ',type(solutions[0]))
```

```
La solution: -sqrt(3)  est de type  <class 'sympy.core.mul.Mul'>
```

Mais Python peut quand même faire des calculs avec la solution et un nombre entier de *type* **int**

[11]:
```
1+solutions[0]
```

[11]: $1 - \sqrt{3}$

3.1 Fonction:

- image d'un nombre
- antécédent(s)
- dérivation
- représentation graphique

Définissons la fonction f avec $x \in \mathbb{R}$, tel que : $f(x) = x^2 + 3x - 1$

[15]:
```
from sympy import diff
from sympy.plotting import plot
```

[16]:
```
from matplotlib import pyplot as plt
```

[17]:
```
#f = Function('f')
f = x**2+3*x-1
```

Calculons l'image de 0 par f, c'est à dire $f(0)$. On remplace x par 0, on *subs*titue x par 0

[18]:
```
f.subs(x,0)
```

[18]: -1

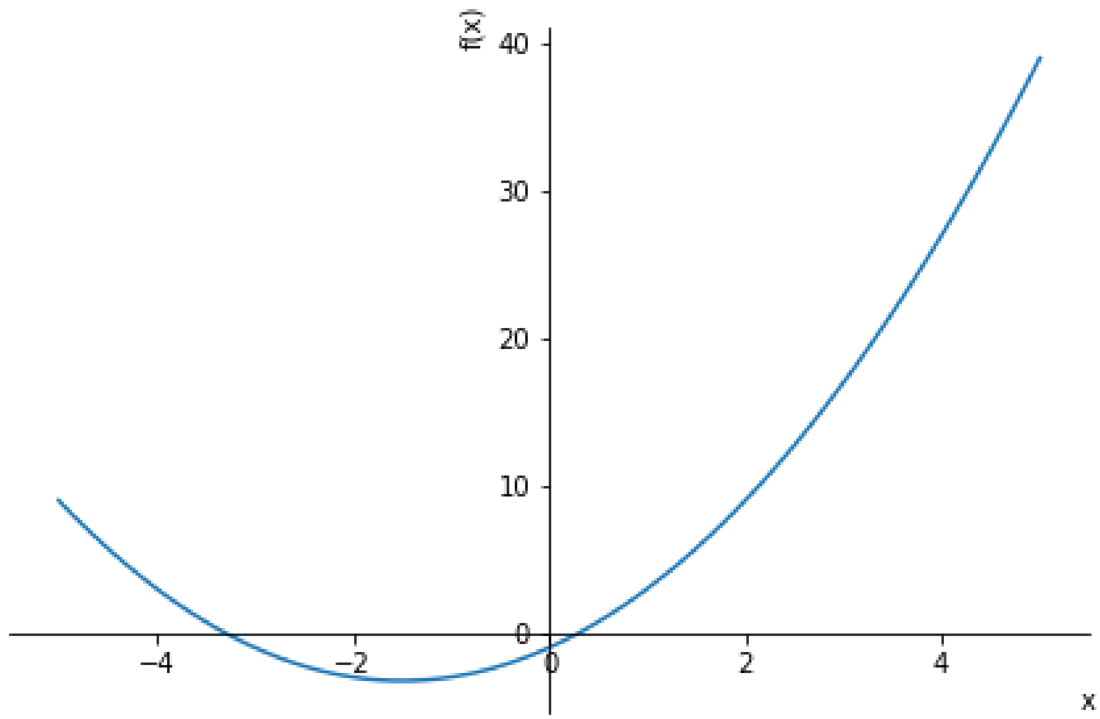
Cherchons les points l'abscisse des points d'intersection de C_f avec l'axe des abscisses, c'est à dire les points d'ordonnées 0. On résoud l'équation: $f(x) = 0$

[19]:
```
solutions = solve(f, x)
x1 = solutions[0]
x2 = solutions[1]
x1, x2
```

[19]:
$$\left(-\frac{3}{2} + \frac{\sqrt{13}}{2}, -\frac{\sqrt{13}}{2} - \frac{3}{2}\right)$$

4 Traçons la courbe représentant la fonction f

[20]:
```
C0=plot(f, (x,-5,5))
```



l'abscisse du sommet de la parabole est donné par:

$x_0 = \dfrac{x_1 + x_2}{2}$

[21]:
```
x0 = (x1+x2)/2
x0
```

[21]: $-\dfrac{3}{2}$

L'ordonnée du sommet est donc $y_0 = f(x_0)$

[22]:
```
y0 = f.subs(x, x0)
y0
```

[22]: $-\dfrac{13}{4}$

On remarque la différence à l'affichage entre:

`pprint(y0)`

et juste:

`y0`

[23]:
```
pprint(y0)
```

```
-13/4
```

5 Fonction dérivée:

Retrouvons ce résultat en dérivant f

[24]:
```
f_prime = diff(f)
print("L'expression de la fonction f est:")
print()
f
```

```
L'expression de la fonction f est:
```

[24]: $x^2 + 3x - 1$

Celle de sa fonction dérivée est:

[25]:
```
f_prime
```

[25]: $2x + 3$

[26]:
```
solutions = solve(f_prime,x)
solutions
```

[26]: $\left[-\frac{3}{2}\right]$

[27]:
```
s0 = solutions[0]
f_prime.subs(x, s0)
```

[27]: 0

[28]:
```
f.subs(x, s0)
```

[28]: $-\frac{13}{4}$

5.1 Signe d'une fonction:

Résolution d'une inéquation Signe de la dérivée:

On résoud l'inéquation,

$$f'(x) > 0$$

.

Lorsque l'agument **relationnal** est fixé à **True**, la solution de l'inéquation est donnée sous forme d'un encadrement:

$$-\frac{3}{2} < x \wedge x < \infty$$

ce qui est plus pratique que la solution donnée avec l'écriture:

$$\left(-\frac{3}{2}, \infty\right)$$

[29]:
```
from sympy.solvers.inequalities import solve_univariate_inequality
```

[30]:
```
solve_univariate_inequality(f_prime > 0, x, relational=False)
```

[30]: $\left(-\frac{3}{2}, \infty\right)$

[31]:
```
solve_univariate_inequality(f_prime > 0, x, relational=True)
```

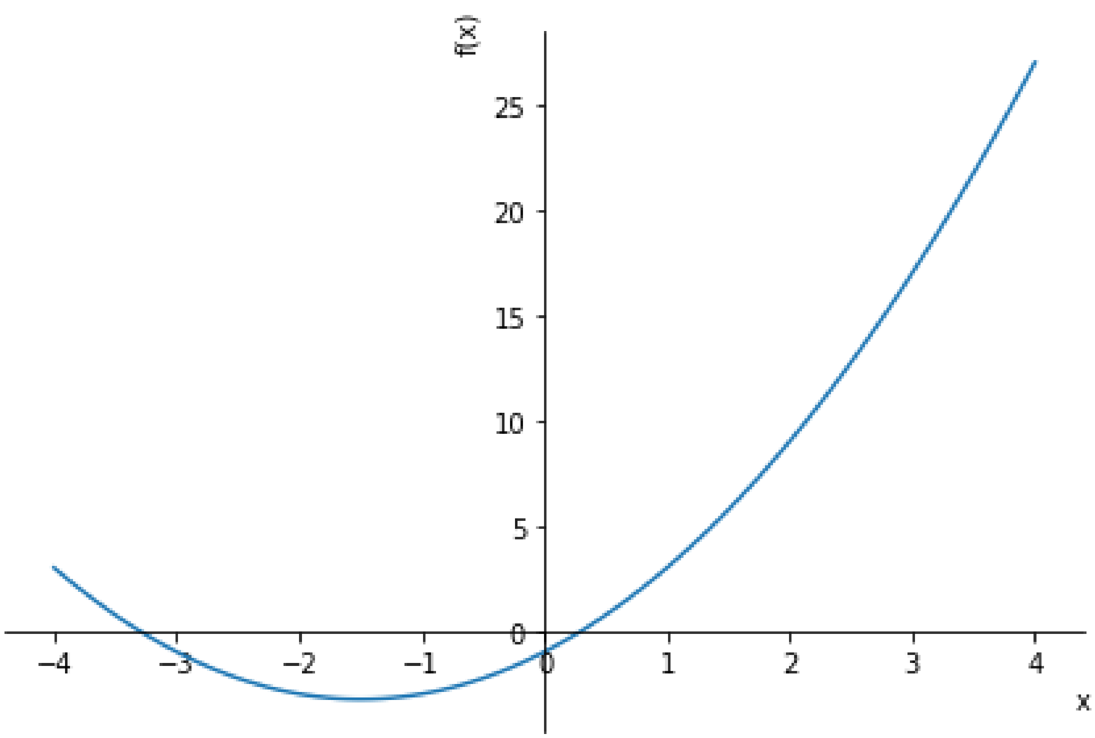
[31]: $-\frac{3}{2} < x \wedge x < \infty$

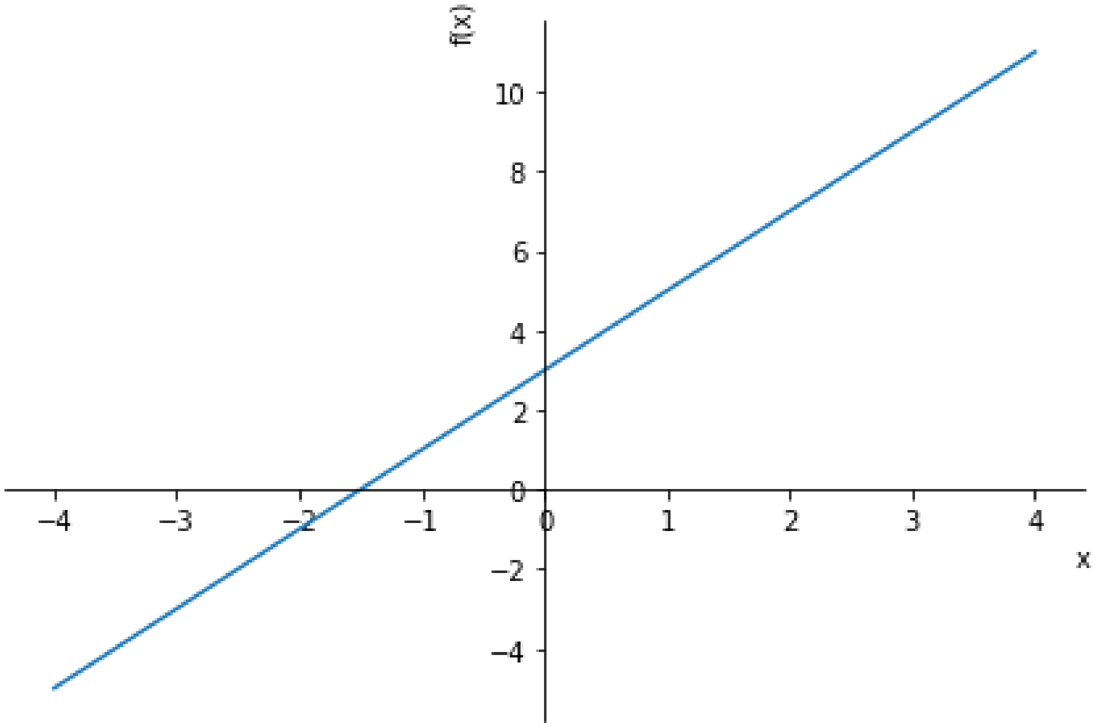
ATTENTION :Le symbole \$ $\wedge$\$*signifie***ET**, *eneffetx* est bien supérieur à $-\frac{3}{2}$ et inférieur à ∞

La dérivée est donc positive pour $x > -\frac{3}{2}$

5.2 Traçons les courbes de f et de f'

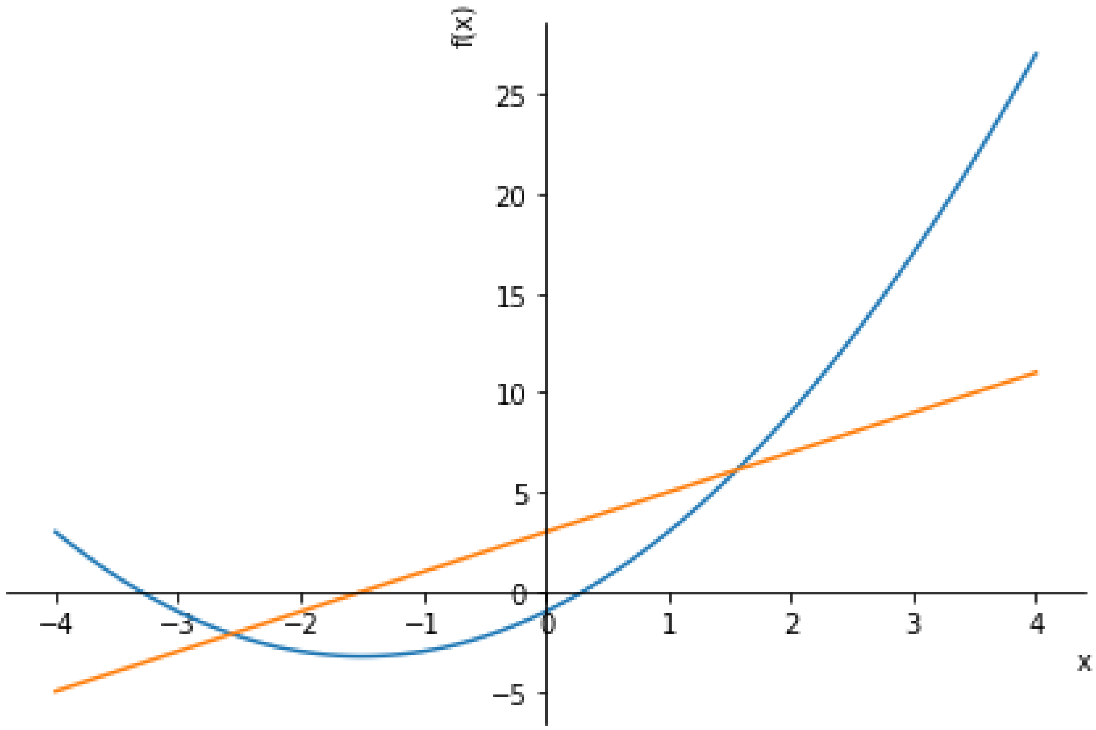
[35]:
```
C0=plot(f, (x,-4,4))
C1=plot(f_prime, (x,-4,4))
```





Une astuce pour supperposer les deux courbes:

[36]:
```
C0.extend(C1)
C0.show()
```



f(x)
25
20
15
10
5
0
−5
−4
−3
−2
−1
0
1
2
3
4
x


```
[ ]:
```