

14. Calcul formel

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.

Le but de ce TP est de découvrir la bibliothèque de calcul formel *Sympy*.

Exemple 1. Calcul littéral

```
>>> from sympy import*
>>> x=symbols('x')
>>> y=symbols('y')
>>> expand((x+y)**3)
x**3 + 3*x**2*y + 3*x*y**2 + y**3
>>> factor(x**3-y**3)
(x - y)*(x**2 + x*y + y**2)
>>> simplify(x**2/(1+x)-1/(1+x))
x - 1
```

Exercice 1. Factoriser $(ac - bd)^2 + (ad + bc)^2$.

Exemple 2. Étude de fonction

```
>>> f=exp(x)/(2*x**2+1)
>>> plot(f, (x,-5,5))
<sympy.plotting.plot.Plot object at 0x0000000009D20588>
>>> limit(f,x,-infty)
0
>>> limit(f,x,+infty)
oo
>>> df=diff(f,x)
>>> factor(df)
(2*x**2 - 4*x + 1)*exp(x)/(2*x**2 + 1)**2
>>> solve(2*x**2-4*x+1>0,x)
And(Or(re(x) < -sqrt(2)/2 + 1, re(x) > sqrt(2)/2 + 1), im(x) == 0)
```

Exercice 2. Étudier les variations de la fonction $f : x \mapsto xe^{-x^2}$.

Exemple 3. Développements limités

```
>>> series(exp(x),x,0,5)
1 + x + x**2/2 + x**3/6 + x**4/24 + O(x**5)
```

Exercice 3. Montrer que la courbe représentative de la fonction $f : x \mapsto x\sqrt{\frac{x+1}{x-1}}$ admet une asymptote oblique en $+\infty$. (on pourra utiliser la commande `f.subs(x,1/X)`)

Exemple 4. Algèbre linéaire

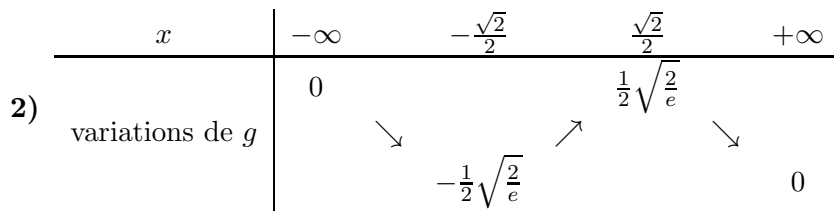
```
>>> M=Matrix([[1,2,3],[4,5,6],[7,8,9]])
>>> U=Matrix([x,y,z])
>>> M*U
Matrix([
 [ x + 2*y + 3*z],
 [4*x + 5*y + 6*z],
 [7*x + 8*y + 9*z]])
>>> solve(M*U)
{x: z, y: -2*z}
```

Exercice 4. Montrer que $P = \begin{pmatrix} 0 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ est la matrice d'un projecteur dont on déterminera les éléments caractéristiques.

Réponses

1)

```
>>> a,b,c,d=symbols('a'),symbols('b'),symbols('c'),symbols('d')
>>> factor((a*c-b*d)**2+(a*d+b*c)**2)
(a**2 + b**2)*(c**2 + d**2)
```



```
>>> g=x*exp(-x**2)
>>> limit(g,x,-oo)
0
>>> limit(g,x,+oo)
0
>>> dg=diff(g,x)
>>> factor(dg)
-(2*x**2 - 1)*exp(-x**2)
>>> solve(-(2*x**2 - 1)>0,x)
And(-sqrt(2)/2 < re(x), im(x) == 0, re(x) < sqrt(2)/2)
>>> g.subs(x,-sqrt(2)/2)
-sqrt(2)*exp(-1/2)/2
>>> g.subs(x,sqrt(2)/2)
sqrt(2)*exp(-1/2)/2
```

3)

```
>>> x=symbols('x')
>>> f=x*sqrt((x+1)/(x-1))
>>> F=f.subs(x,1/X)
>>> S=series(F,X,0,2)
>>> S.subs(X,1/x)
1/(2*x) + 1 + x + O(x**(-2), (x, oo))
>>> limit(f-(x+1),x,oo)
0
```

4)

```
>>> P=Matrix([[0,-1,-1],[0,1,0],[0,0,1]])
>>> P*P-P
Matrix([
[0, 0, 0],
[0, 0, 0],
[0, 0, 0]])
>>> V=Matrix([[x],[y],[z]])
>>> solve(P*V)
[{z: 0, y: 0}]
>>> solve(P*V-V)
{x: -y - z}
```