

# Exercices sur les limites

28 octobre 2015

```
var('x')
funcslims = [
('x**2 - 2*x - 3', '+oo'),
('-x**2 + 2*x + 3', '-oo')
]

print(r'\begin{Exercise}[number=1,title={Vrai-faux}]')
for func,lim in funcslims:
    question = "Calculer la limite de  $\{\}\{\}$  en  $\{\}\{\}$ ".format(
        latex(eval(func)),latex(eval(lim)))
    print(r'\Question')
    print(question)

print(r'\end{Exercise}')
print(r'\begin{Answer}')
for func,lim in funcslims:
    print(r'\Question')
    # Décomposition en somme : on initialise une liste vide dans
    # laquelle on ajoute les éléments de la somme
    funcn = []
    for k,v in enumerate(eval(func).as_coefficients_dict()):
        funcn.append(eval('{}*{}`.format(eval(func).as_coefficients_dict()[v],v)))
    print(r"L'expression se décompose comme la somme de", end=' ')
    # On affiche les différents termes de cette somme.
    for i in range(len(funcn) - 1):
        print(r" $\{\}\{\}$ ,".format(latex(funcn[i])), end=' ')
        print(r"et  $\{\}\{\}$ .".format(latex(funcn[-1])))
    limits_mon = map(lambda x: Limit(x,'x',lim).doit(), funcn)
    # On passe à la limite pour chacun des termes (dans la liste) Voir
    # les effets de 'map' et 'lambda: '
    # Si la somme des limites n'est pas indéterminée, alors on a le
    # résultat et on l'affiche
    if sum(limits_mon) != nan:
        print(r'La somme des limites donne ici le résultat :')
        limit = Limit(funcn,'x',lim)
        reponse = " $\{\}\{\} = \{\}\{\}$ ".format(latex(limit),latex(limit.doit()))
        print(reponse)
    # Sinon on factorise
    else:
        print(r'La somme des limites ici conduit à une forme \
indéterminée. On ne peut conclure ainsi.')
        print(r'Factorisons, si cela est possible.')
```

```

# Si la factorisation «échoue», on «factorise par  $x \neq 0$ 
if factor(func) != func:
    print(r'Ici, on ne peut pas factoriser. Pour  $x \neq 0$ , on a :')
    x = Symbol('x', nonzero = True)
    funcx = expand(eval(func)/x)
    print(r'${} = x \times \left( {} \right)$.'.format(
        latex(eval(func)), latex(funcx)))
    print(r'Le deuxième facteur de ce produit a pour limite')
    funcxn = []
    for k,v in enumerate(funcx.as_coefficients_dict()):
        funcxn.append(eval('{}*{}'.format(
            funcx.as_coefficients_dict()[v],v)))
    limits_monx = map(lambda f: Limit(f,'x',lim).doit(), funcxn)
    print(r'${}$.'.format(latex(Limit(funcx,x,lim).doit())))
    limit = Limit(func,'x',lim)
    reponse = "${} = {}$".format(latex(limit), latex(limit.doit()))
    print("Le produit nous donne donc", reponse)

print(r'\end{Answer}')

```