Notebook d'usage des codes de bissection_exclusion

August 19, 2020

1 Exemple d'usage de nos algorithmes et discussion

```
In [2]: from ore_algebra import *
        from ore_algebra import DifferentialOperators
        from sage.rings.rational_field import QQ
        Rx.<x> = QQ['x']
        A.<Dx> = OreAlgebra(Rx, 'Dx')

In [3]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt

In [4]: def plot_function_(dop,ini,name,color_fun):
            x=[-20+k*1e-2 for k in range(2000)]
            y=[]
            for k in range (2000):
                y.append(dop.numerical_solution(ini,[0,-20+k*1e-2],1e-10).mid())
            plt.plot(x,y,color=color_fun,label=name)
            plt.legend()
            plt.show()
```

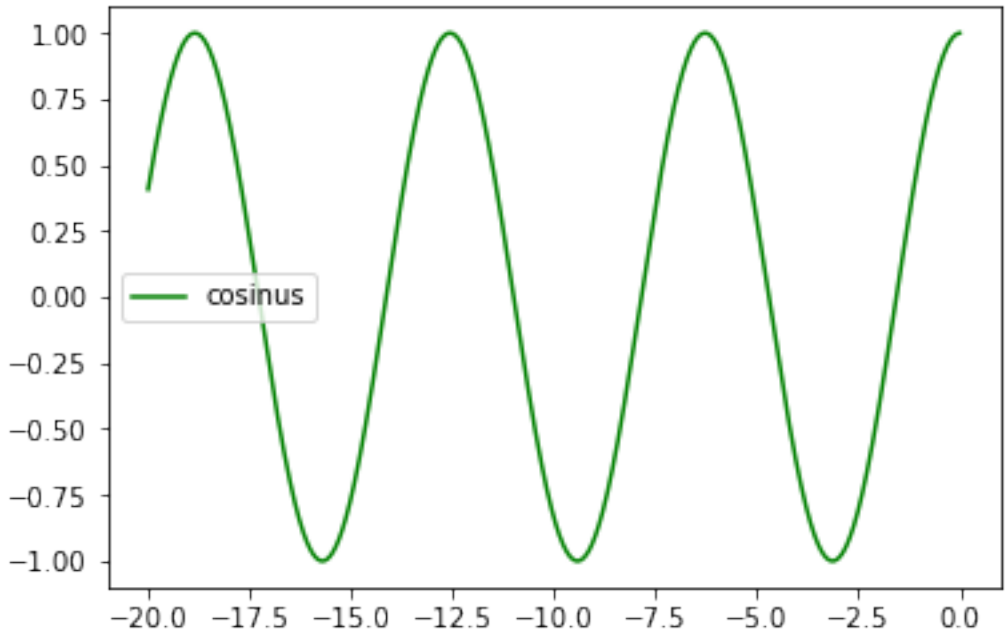
2 #Temps d'exécution sur plusieurs fonctions D-finies

On introduit ici quelques fonctions, la première fonction traitée sera la fonction cosinus, la fonction d'Airy, et les trois dernières fonctions sont des fonctions quelconques

```
In [5]: dop_1=Dx^2+1
        ini_1=[1,0]
        dop_2=Dx^2-x
        ini_2=[RealBallField(500)((gamma(2/3)*3^(2/3))^(-1)),RealBallField(500)(-(gamma(1/3)*3^(
        dop_3=Dx^5-x*Dx^2+(x^2-3)*Dx
        ini_3=[1,-10,4,3,-1]
        dop_4=((1/10)*x^2-50)*Dx^4-x*Dx+1
        ini_4=[1,10,4,-4]
        dop_5=Dx^5+10*x^2*Dx^3-1
        ini_5=[1,-10,10,3,-5]
```

2.0.1 La fonction cosinus:

```
In [6]: %time plot_function_(dop_1,ini_1,"cosinus",'green')
```




```
CPU times: user 41.6 s, sys: 15 ms, total: 41.6 s
Wall time: 41.6 s
```

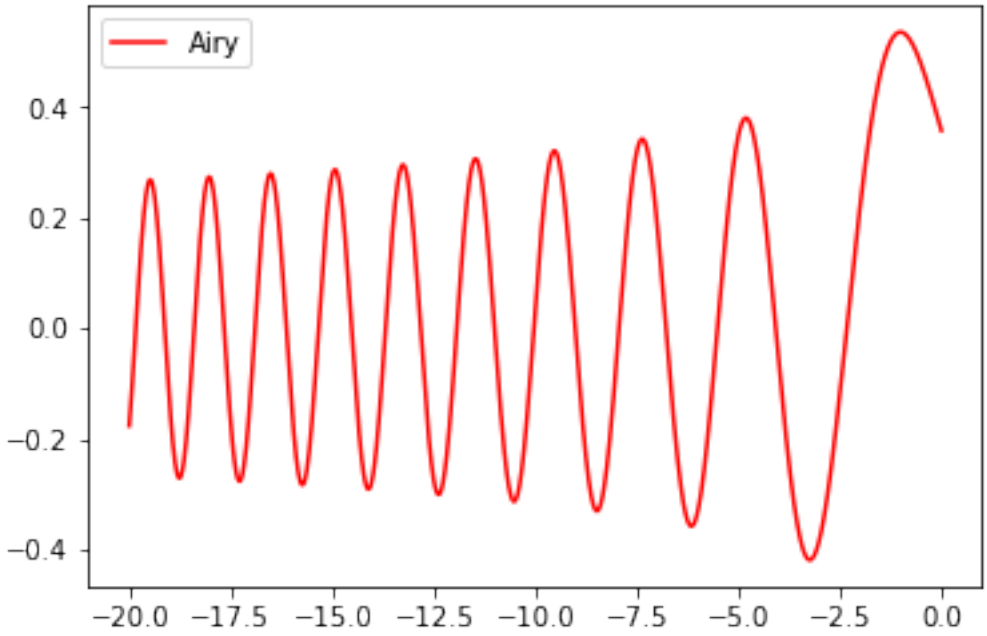
```
In [7]: %time find_certified_zeros(dop_1,ini_1,[-20,0])

CPU times: user 2.75 s, sys: 0 ns, total: 2.75 s
Wall time: 2.75 s
```

```
Out[7]: [[[-17.278778, -17.277527],
          [-14.137263, -14.136012],
          [-10.995878, -10.990871],
          [-7.8547742, -7.8447495],
          [-4.7158491, -4.6957720],
          [-1.5709168, -1.5659098]],
         []]
```

La fonction d'Airy:

2.0.2 La fonction d'Airy:

```
In [8]: %time plot_function_(dop_2,ini_2,"Airy",'red')
```




```
CPU times: user 30.1 s, sys: 16 ms, total: 30.1 s
Wall time: 30.1 s
```

```
In [9]: %time find_certified_zeros(dop_2,ini_2,[-20,0])
```

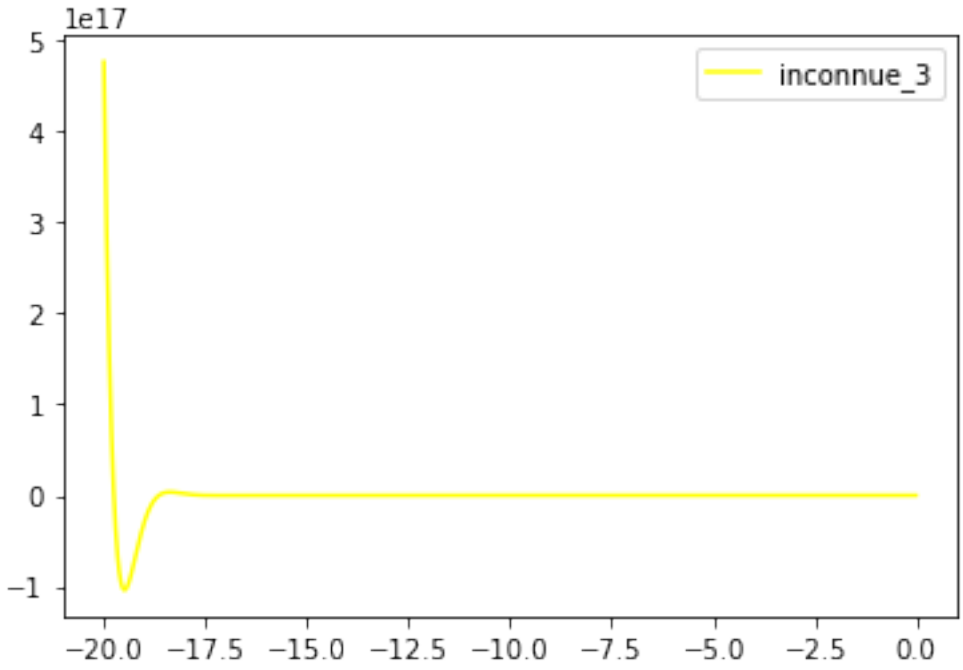
```
CPU times: user 8.17 s, sys: 0 ns, total: 8.17 s
Wall time: 8.16 s
```

```
Out[9]: [[[-19.838131, -19.837818],
          [-19.128438, -19.118350],
          [-18.402094, -18.391974],
          [-17.662035, -17.657009],
          [-16.905971, -16.900939],
          [-16.133182, -16.130677],
          [-15.340759, -15.340720],
          [-14.528346, -14.518213],
          [-13.691571, -13.689063],
          [-12.829289, -12.826783],
          [-11.936136, -11.925991],
          [-11.008575, -11.008262],
          [-10.040317, -10.037808],
          [-9.0227386, -9.0177030],
          [-7.9488629, -7.9285447],
          [-6.7867435, -6.7865873],
```

```
 [-5.5216326, -5.5115159],
 [-4.0891845, -4.0841758],
 [-2.3381267, -2.3356225]],
[]]
```

2.0.3 Une fonction quelconque

```
In [10]: %time plot_function_(dop_3,ini_3,"inconnue_3",'yellow')
```




```
CPU times: user 39 s, sys: 15 ms, total: 39.1 s
Wall time: 39.1 s
```

```
In [11]: %time find_certified_zeros(dop_3,ini_3,[-20,-1])

CPU times: user 8.87 s, sys: 0 ns, total: 8.87 s
Wall time: 8.87 s
```

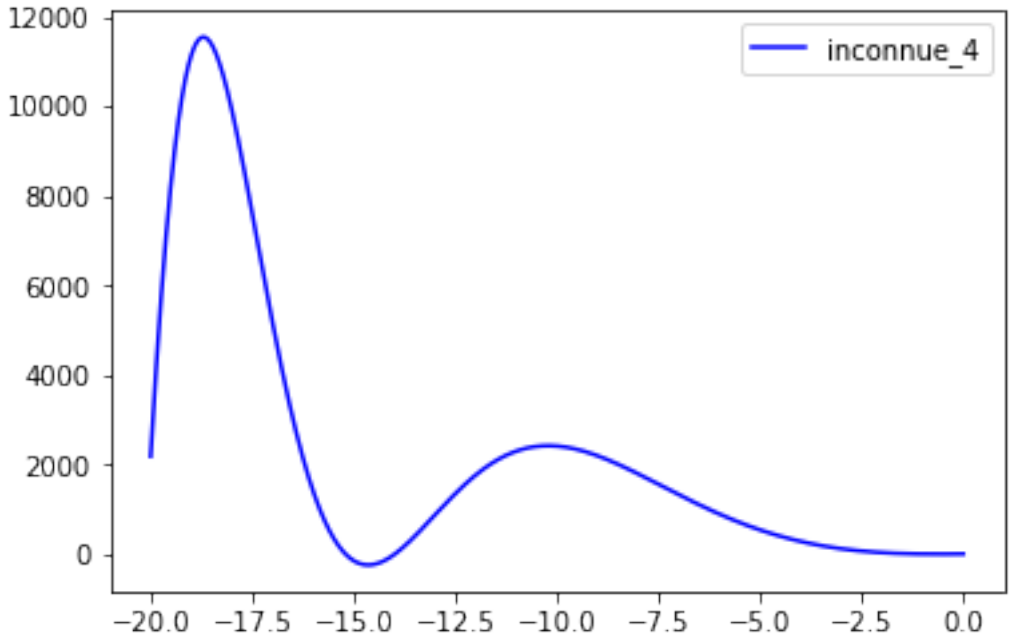
```
Out[11]: [[[-19.775961, -19.773568],
           [-18.669915, -18.665124],
           [-17.525513, -17.515790],
           [-16.337883, -16.333097],
           [-15.099798, -15.095003],
           [-13.801360, -13.801063],
           [-12.430876, -12.429687],
```

```
 [-10.968377, -10.965994],
 [-9.3830740, -9.3827770],
 [-7.6211096, -7.6199208],
 [-5.6337266, -5.6289529]],
[]]
```

C'est pour ce genre de fonction que notre algorithme est utile.

2.0.4 Encore une fonction quelconque

```
In [12]: %time plot_function_(dop_4,ini_4,"inconnue_4",'blue')
```




```
CPU times: user 41.4 s, sys: 16 ms, total: 41.4 s
Wall time: 41.4 s
```

```
In [13]: %time find_certified_zeros(dop_4,ini_4,[-20,0])

CPU times: user 1.05 s, sys: 0 ns, total: 1.05 s
Wall time: 1.03 s
```

```
Out[13]: [[[-0.100003662109375, -0.0997506713867188]], []]
```

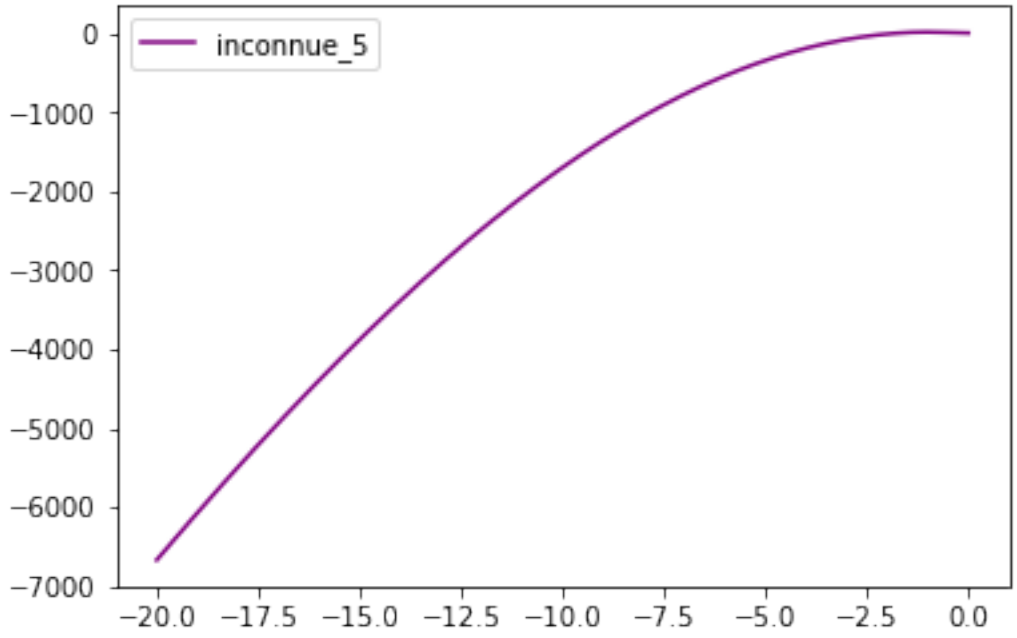
Remarque on dirait que notre algorithme a manqué deux zéros autours de 15, vérifions.

```
In [17]: dop_4.numerical_solution(ini_4,[0,-14.5],1e-30)
```

```
Out[17]: [-142.3705396298176 +/- 4.25e-14]

In [21]: dop_4.numerical_solution(ini_4,[0,-10])

Out[21]: [-93.380659891825512 +/- 4.13e-16]

In [22]: dop_4.numerical_solution(ini_4,[0,-18])

Out[22]: [-164.32774868043777 +/- 2.17e-15]
```

Ça n'est pas le cas, mais la représentation graphique de cette fonction n'est pas correcte.

```
In [14]: %time plot_function_(dop_5,ini_5,"inconnue_5",'purple')
```




```
CPU times: user 3min 20s, sys: 16 ms, total: 3min 20s
Wall time: 3min 20s

In [15]: %time find_certified_zeros(dop_5,ini_5,[-20,0])

CPU times: user 14.3 s, sys: 0 ns, total: 14.3 s
Wall time: 14.4 s

Out[15]: [[[-11.372607, -11.372451]], []]

In [18]: dop_5.numerical_solution(ini_5,[0,-11.372607])

Out[18]: [-0.0005020508433997 +/- 5.46e-17]
```

Le résultat semble correct, mais ne correspond pas à ce qu'on observe graphiquement.