TP10_correction

May 26, 2017

In [2]:
```
from pylab import *
from numpy import exp
from scipy.integrate import odeint
```

Activite 1

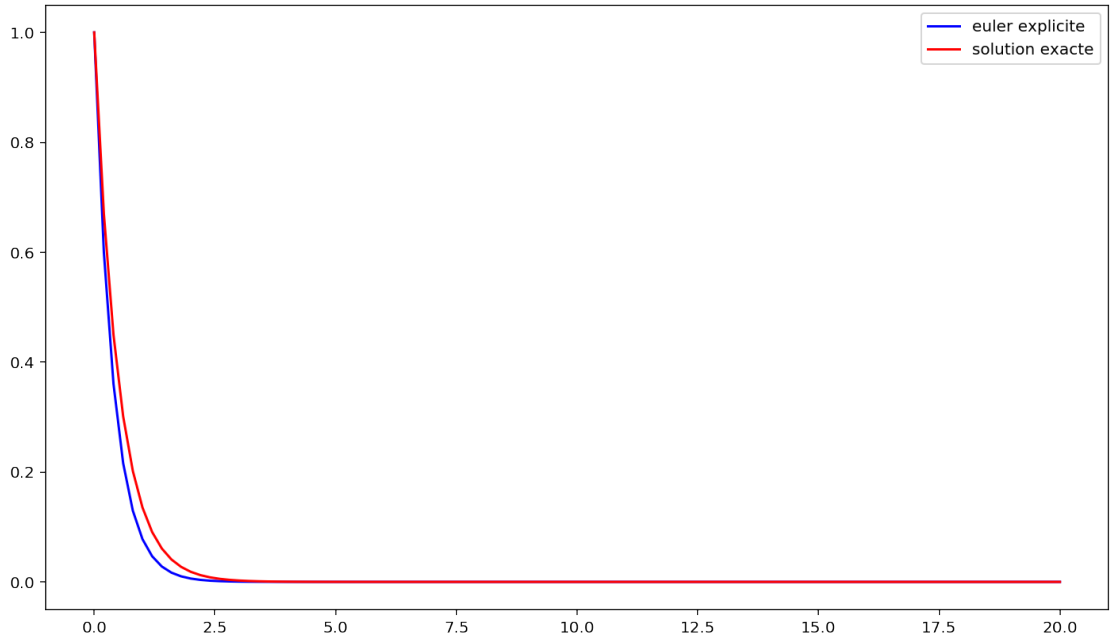
La fonction euler_exp retourne deux listes. La première correspond à la subdivision choisie. La deuxième correspond aux valeurs approchées de la solution par la méthode d'Euler explicite, au niveau des points de la subdivision du temps.

La fonction trace_exp trace les solutions exactes et approchées pour un pas et un nombre de points de la sibdivision fixés.

In [3]:
```
def euler_exp(h,N):
    L=[1]
    x=1
    T=[0]
    for k in range(1,N+1):
        x=(1-2*h)*x
        L.append(x)
        T.append(k*h)
    return array(T),L


def trace_exp(h,n):
    T=euler_exp(h,n)[0]
    plot(T,euler_exp(h,n)[1],'b',label='euler explicite')
    plot(T,exp(-2*T),'r',label='solution exacte')
    legend()
    show()
```

In [4]:
```
trace_exp(0.2,100)
```

Out[4]:



On écrit ensuite une fonction qui mesure l'erreur entre ces-deux dernières :

```
In [5]: def erreur_exp(h,n):
            return abs(euler_exp(h,n)[1]-exp(-2*euler_exp(h,n)[0]))
```

Par exemple :

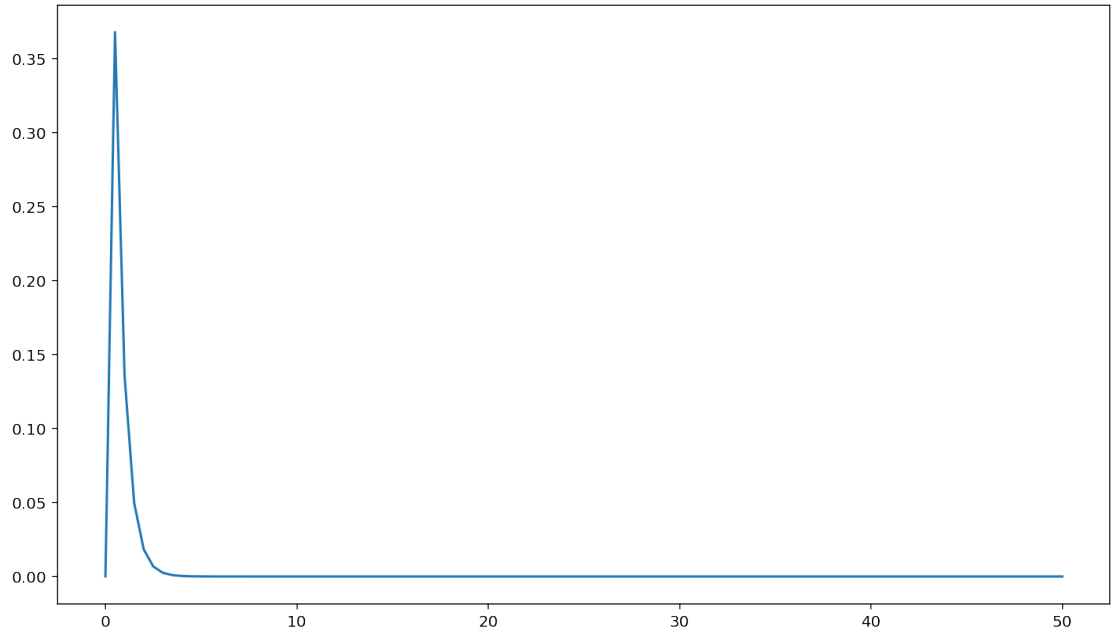
```
In [6]: erreur_exp(0.5,100)
```

```
Out[6]: array([  0.00000000e+00,   3.67879441e-01,   1.35335283e-01,
                 4.97870684e-02,   1.83156389e-02,   6.73794700e-03,
                 2.47875218e-03,   9.11881966e-04,   3.35462628e-04,
                 1.23409804e-04,   4.53999298e-05,   1.67017008e-05,
                 6.14421235e-06,   2.26032941e-06,   8.31528719e-07,
                 3.05902321e-07,   1.12535175e-07,   4.13993772e-08,
                 1.52299797e-08,   5.60279644e-09,   2.06115362e-09,
                 7.58256043e-10,   2.78946809e-10,   1.02618796e-10,
                 3.77513454e-11,   1.38879439e-11,   5.10908903e-12,
                 1.87952882e-12,   6.91440011e-13,   2.54366565e-13,
                 9.35762297e-14,   3.44247711e-14,   1.26641655e-14,
                 4.65888615e-15,   1.71390843e-15,   6.30511676e-16,
                 2.31952283e-16,   8.53304763e-17,   3.13913279e-17,
                 1.15482242e-17,   4.24835426e-18,   1.56288219e-18,
                 5.74952226e-19,   2.11513104e-19,   7.78113224e-20,
                 2.86251858e-20,   1.05306174e-20,   3.87399763e-21,
                 1.42516408e-21,   5.24288566e-22,   1.92874985e-22,
                 7.09547416e-23,   2.61027907e-23,   9.60268005e-24,
```

```
3.53262857e-24,   1.29958143e-24,   4.78089288e-25,
1.75879220e-25,   6.47023493e-26,   2.38026641e-26,
8.75651076e-27,   3.22134029e-27,   1.18506486e-27,
4.35961000e-28,   1.60381089e-28,   5.90009054e-29,
2.17052201e-29,   7.98490425e-30,   2.93748211e-30,
1.08063928e-30,   3.97544974e-31,   1.46248623e-31,
5.38018616e-32,   1.97925988e-32,   7.28129018e-33,
2.67863696e-33,   9.85415469e-34,   3.62514092e-34,
1.33361482e-34,   4.90609473e-35,   1.80485139e-35,
6.63967720e-36,   2.44260074e-36,   8.98582594e-37,
3.30570063e-37,   1.21609930e-37,   4.47377931e-38,
1.64581143e-38,   6.05460190e-39,   2.22736356e-39,
8.19401262e-40,   3.01440879e-40,   1.10893902e-40,
4.07955867e-41,   1.50078576e-41,   5.52108228e-42,
2.03109266e-42,   7.47197234e-43,   2.74878501e-43,
1.01122149e-43,   3.72007598e-44])
```

On trace cette-dernière :

```
In [7]: plot(euler_exp(0.5,100)[0],erreur_exp(0.5,100))

Out[7]: [<matplotlib.lines.Line2D at 0x7f08b0349b00>]

Out[7]:
```

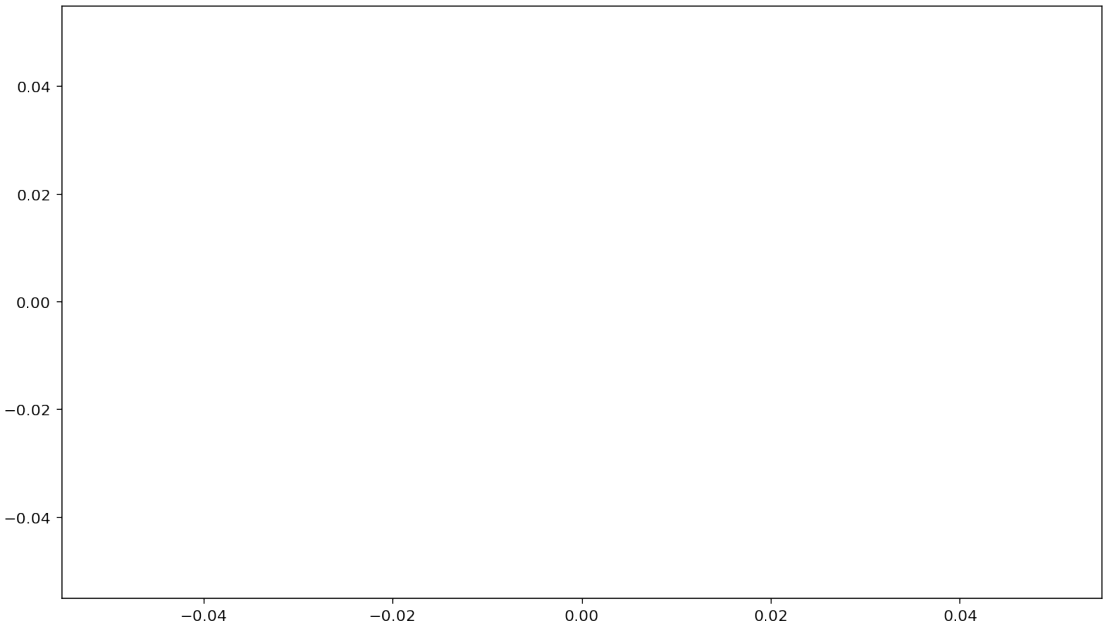


On essaye maintenant de faire varier h afin d'obtenir la limite de stabilité :

```
In [15]: for k in range(0,20,2):
             a=k/10
             print(a)
             plot(euler_exp(a,100)[0],erreur_exp(a,100))
             show()
             close()
```

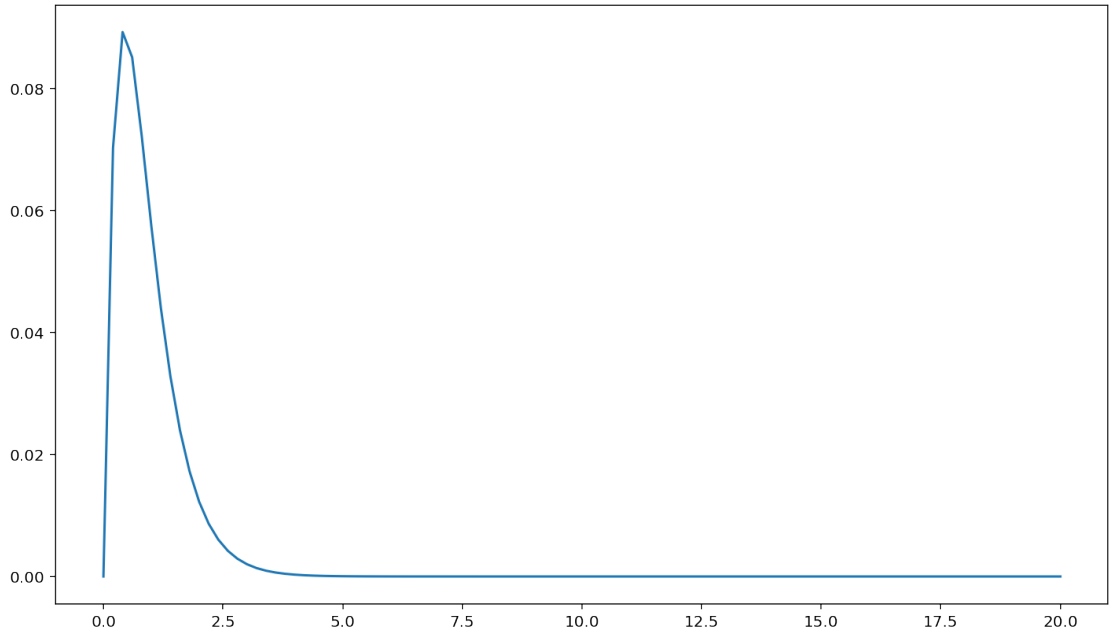
```
0.0
```

Out[15]:



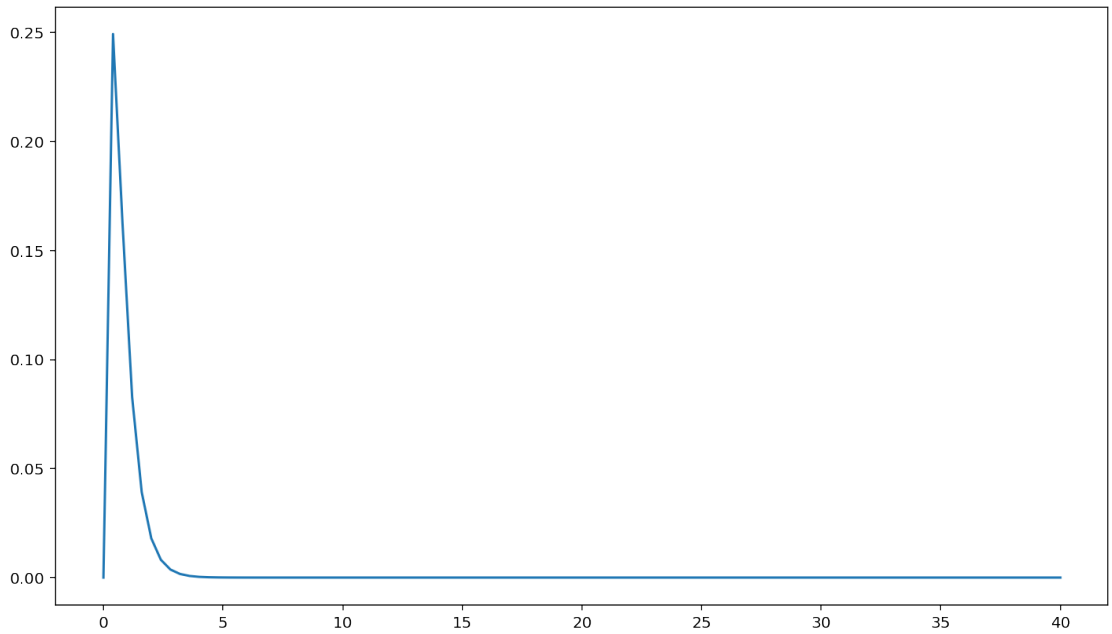

```
0.2
```

Out[15]:


0.08
0.06
0.04
0.02
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

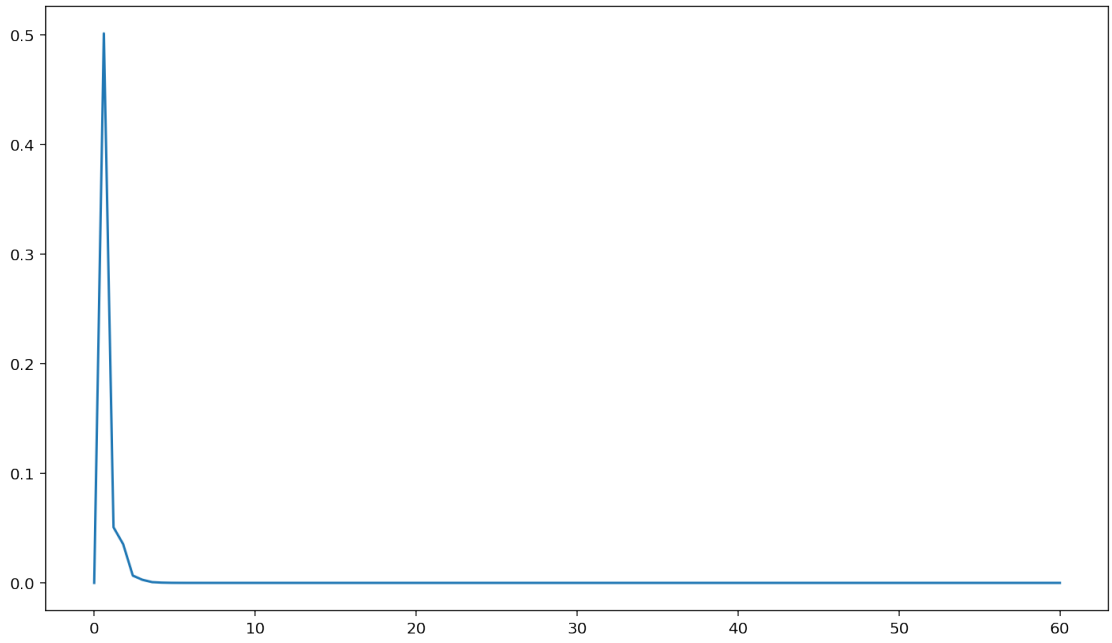
```
0.4
```

Out[15]:


0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30
35
40

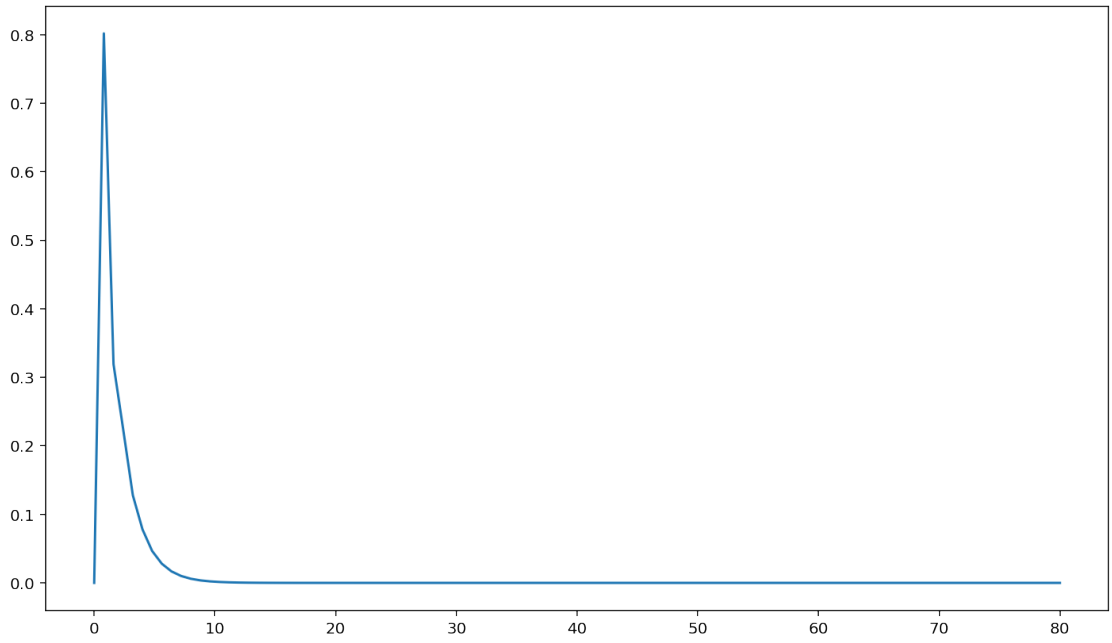
0.6

Out[15]:



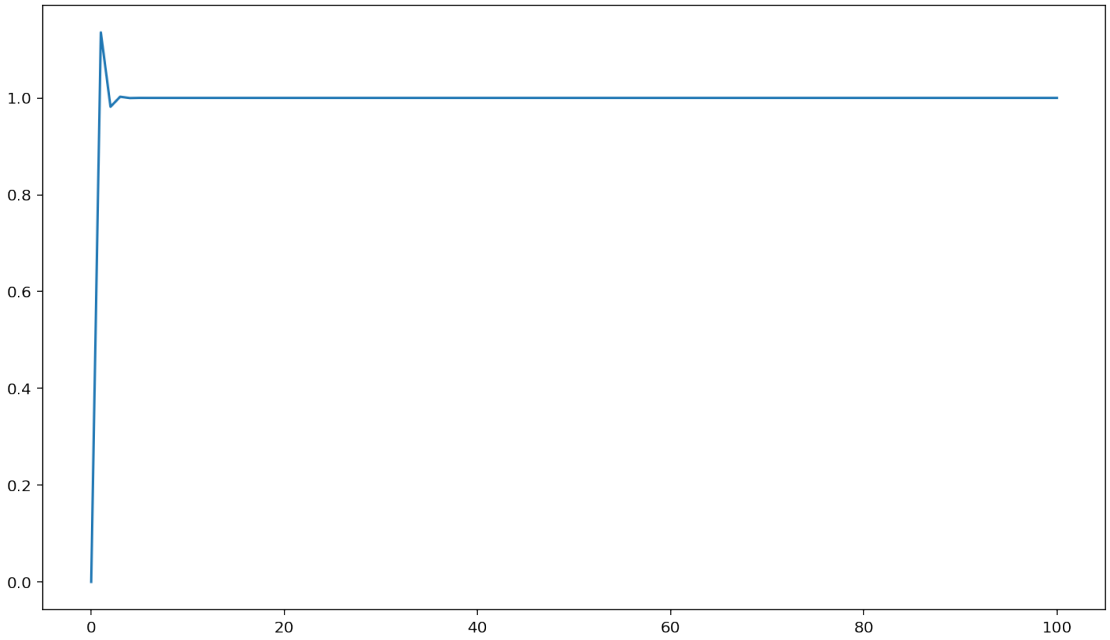
0.8

Out[15]:



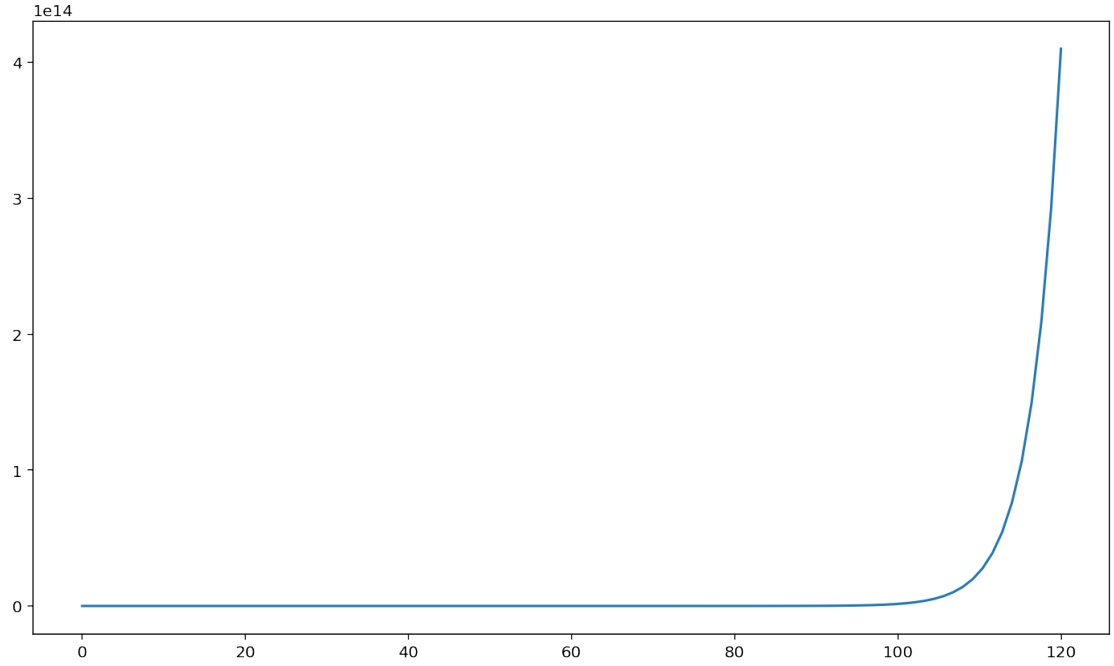
1.0

Out[15]:



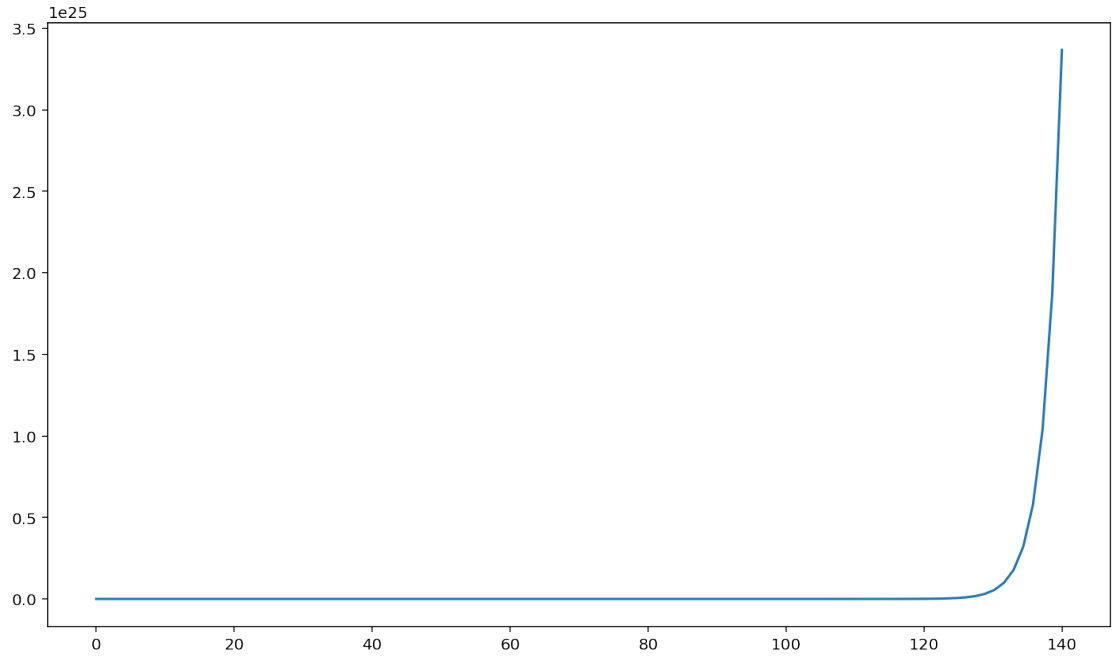
1.2

Out[15]:


1e14
4
3
2
1
0
0
20
40
60
80
100
120

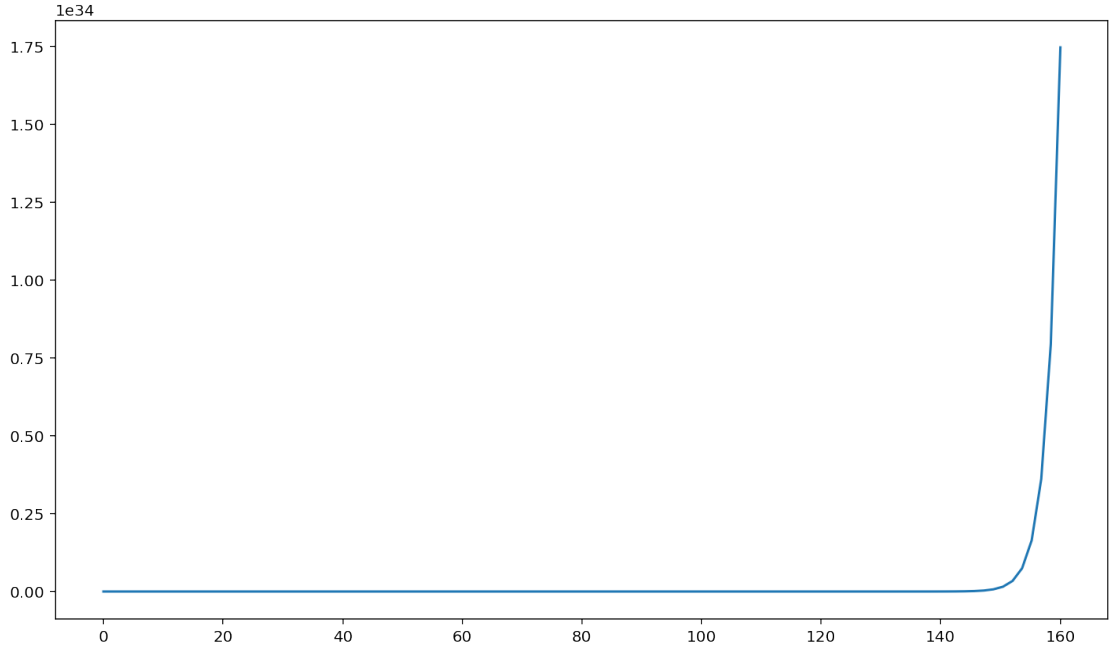
1.4

Out[15]:


1e25
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
120
140

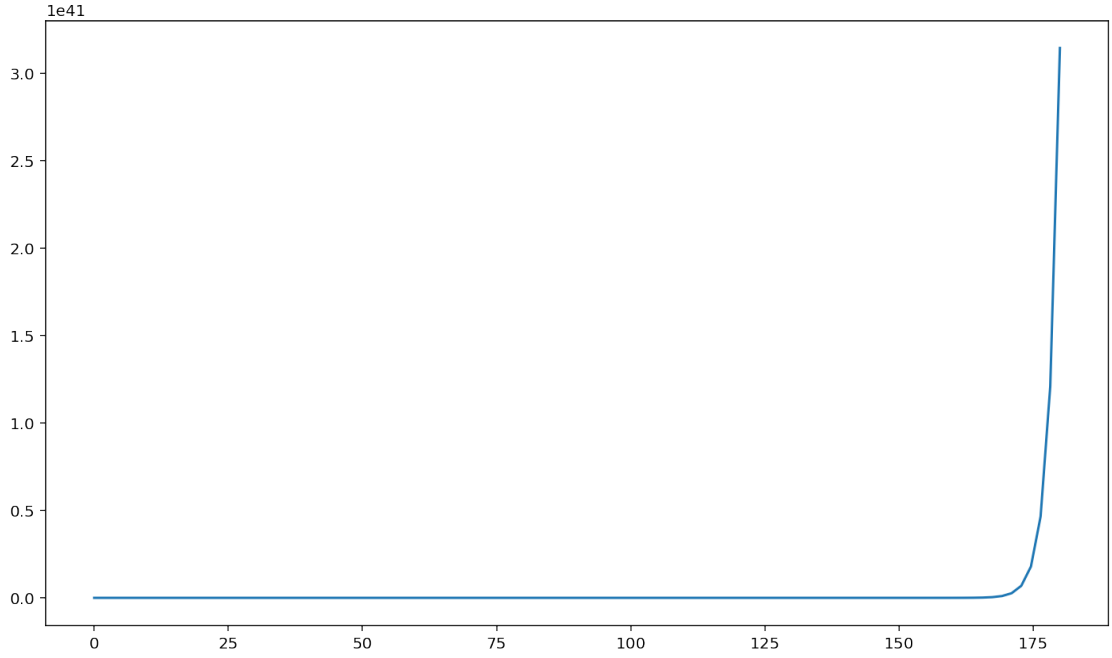
1.6

Out[15]:



1.8

Out[15]:

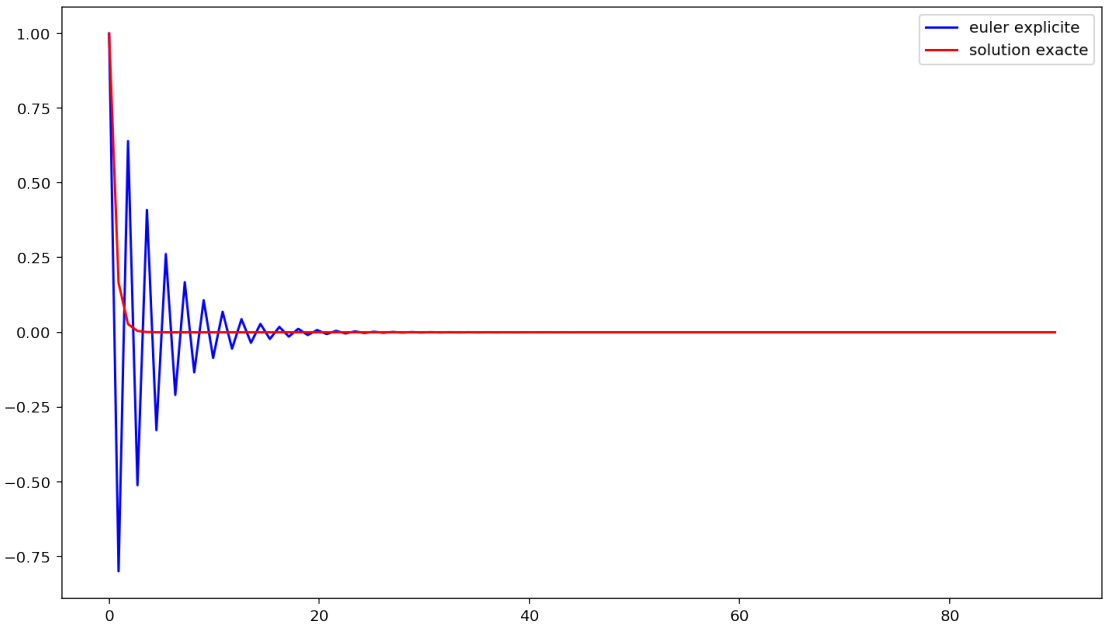


Il semblerait que pour h=1 l'erreur ne tende plus asymptotiquement vers 0. On peut tracer l'allure des solutions pour h=0.9;h=1,h=1.1

In [17]:
```
h=0.9
print(h)
trace_exp(h,100)
show()
close()
h=1
print(h)
trace_exp(h,100)
show()
close()
h=1.1
print(h)
trace_exp(h,100)
show()
close()
```

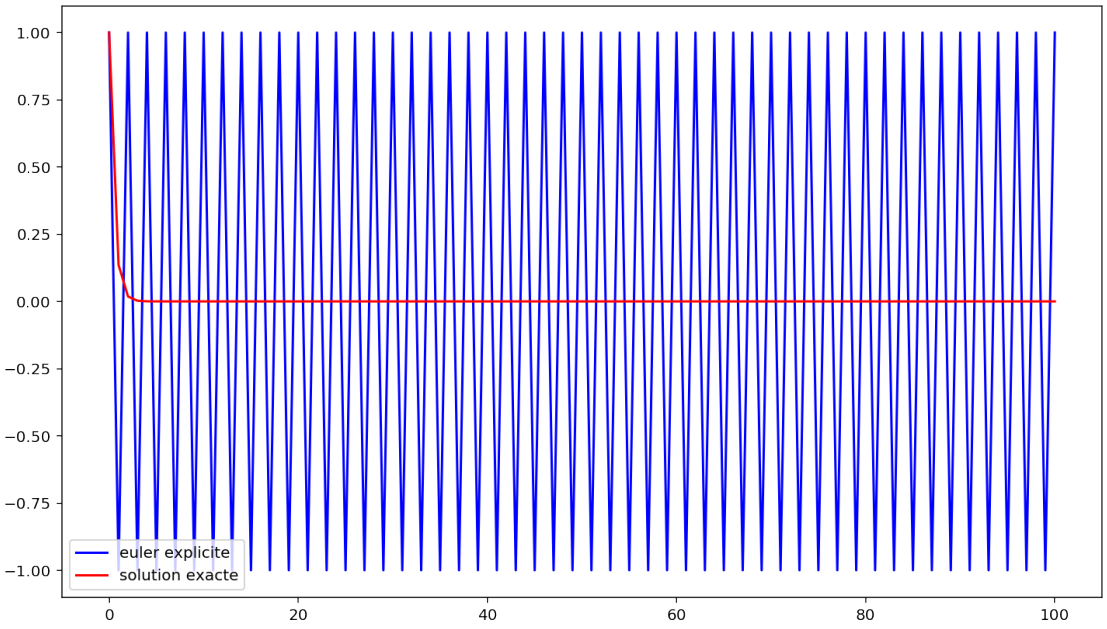
```
0.9
```

Out[17]:



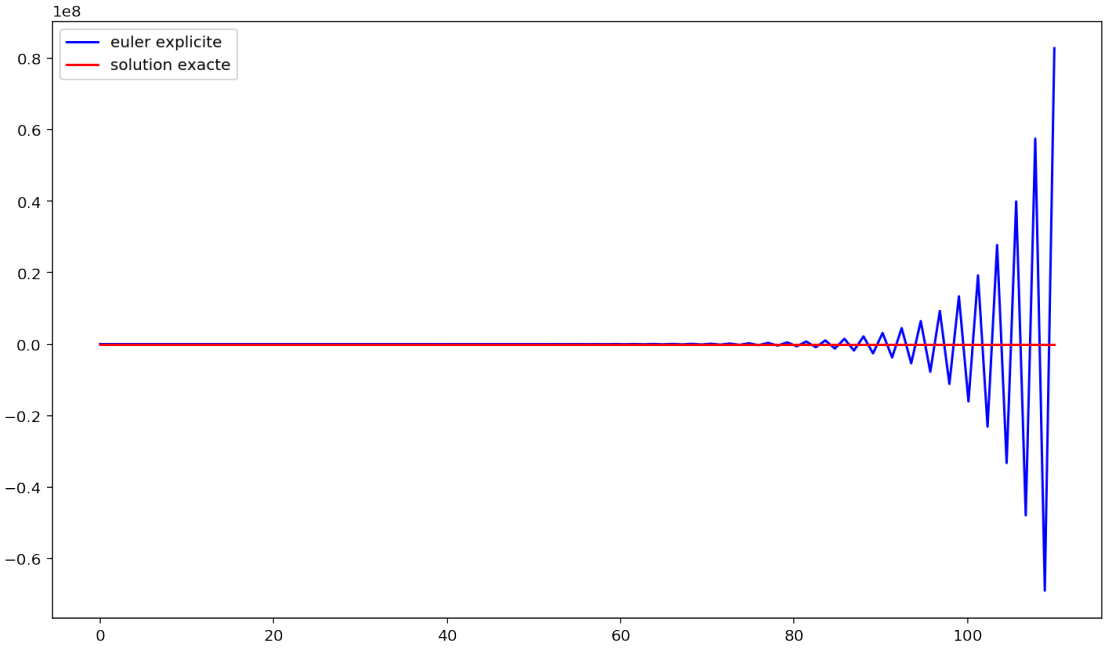

```
1
```

Out[17]:



1.1

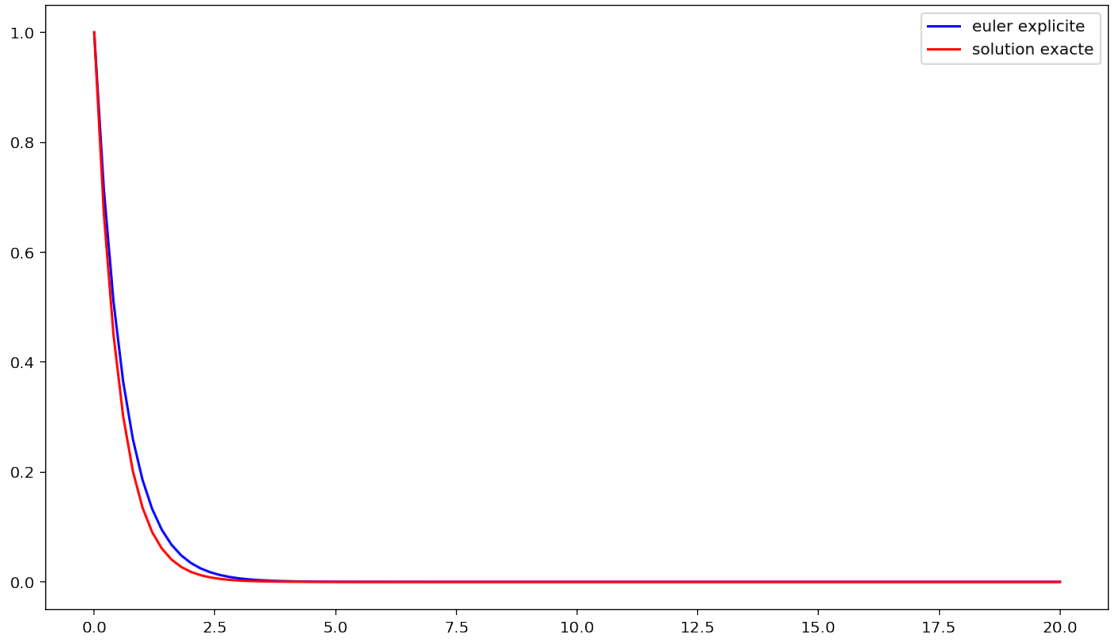
Out[17]:



La limite de stabilité mise en évidence est donc ici 1
Activité 2 : on reprend les questions précédentes avec euler implicite :

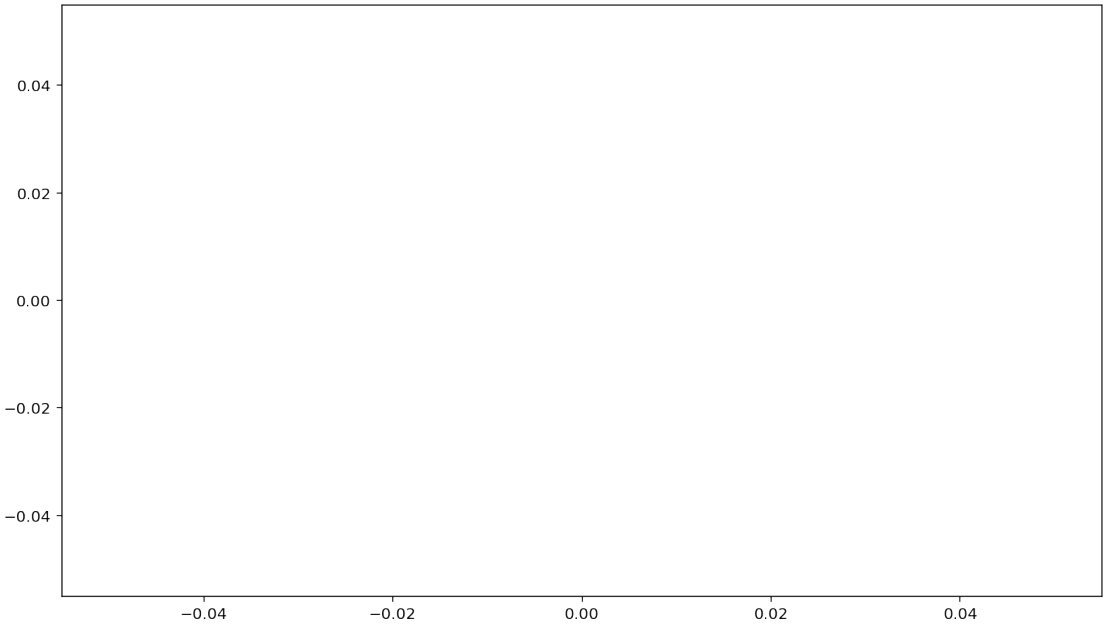
```
In [18]: def euler_imp(h,N):
             L=[1]
             x=1
             T=[0]
             for k in range(1,N+1):
                 x=(1/(1+2*h))*x
                 L.append(x)
                 T.append(k*h)
             return array(T),L


         def trace_imp(h,n):
             T=euler_imp(h,n)[0]
             plot(T,euler_imp(h,n)[1],'b',label='euler explicite')
             plot(T,exp(-2*T),'r',label='solution exacte')
             legend()
             show()

In [19]: trace_imp(0.2,100)
```

Out[19]:



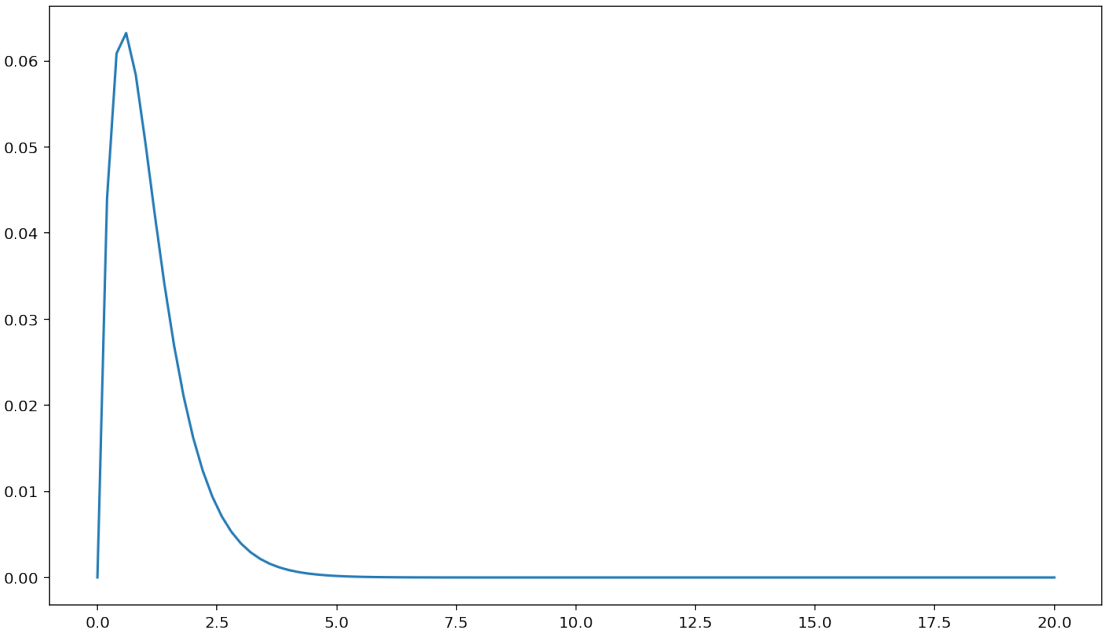
```
In [20]: def erreur_imp(h,n):
             return abs(euler_imp(h,n)[1]-exp(-2*euler_imp(h,n)[0]))

In [21]: erreur_exp(0.5,100)

Out[21]: array([  0.00000000e+00,   3.67879441e-01,   1.35335283e-01,
                  4.97870684e-02,   1.83156389e-02,   6.73794700e-03,
                  2.47875218e-03,   9.11881966e-04,   3.35462628e-04,
                  1.23409804e-04,   4.53999298e-05,   1.67017008e-05,
                  6.14421235e-06,   2.26032941e-06,   8.31528719e-07,
                  3.05902321e-07,   1.12535175e-07,   4.13993772e-08,
                  1.52299797e-08,   5.60279644e-09,   2.06115362e-09,
                  7.58256043e-10,   2.78946809e-10,   1.02618796e-10,
                  3.77513454e-11,   1.38879439e-11,   5.10908903e-12,
                  1.87952882e-12,   6.91440011e-13,   2.54366565e-13,
                  9.35762297e-14,   3.44247711e-14,   1.26641655e-14,
                  4.65888615e-15,   1.71390843e-15,   6.30511676e-16,
                  2.31952283e-16,   8.53304763e-17,   3.13913279e-17,
                  1.15482242e-17,   4.24835426e-18,   1.56288219e-18,
                  5.74952226e-19,   2.11513104e-19,   7.78113224e-20,
                  2.86251858e-20,   1.05306174e-20,   3.87399763e-21,
                  1.42516408e-21,   5.24288566e-22,   1.92874985e-22,
                  7.09547416e-23,   2.61027907e-23,   9.60268005e-24,
                  3.53262857e-24,   1.29958143e-24,   4.78089288e-25,
                  1.75879220e-25,   6.47023493e-26,   2.38026641e-26,
                  8.75651076e-27,   3.22134029e-27,   1.18506486e-27,
                  4.35961000e-28,   1.60381089e-28,   5.90009054e-29,
                  2.17052201e-29,   7.98490425e-30,   2.93748211e-30,
                  1.08063928e-30,   3.97544974e-31,   1.46248623e-31,
                  5.38018616e-32,   1.97925988e-32,   7.28129018e-33,
                  2.67863696e-33,   9.85415469e-34,   3.62514092e-34,
                  1.33361482e-34,   4.90609473e-35,   1.80485139e-35,
                  6.63967720e-36,   2.44260074e-36,   8.98582594e-37,
                  3.30570063e-37,   1.21609930e-37,   4.47377931e-38,
                  1.64581143e-38,   6.05460190e-39,   2.22736356e-39,
                  8.19401262e-40,   3.01440879e-40,   1.10893902e-40,
                  4.07955867e-41,   1.50078576e-41,   5.52108228e-42,
                  2.03109266e-42,   7.47197234e-43,   2.74878501e-43,
                  1.01122149e-43,   3.72007598e-44])

In [22]: for k in range(0,20,2):
             a=k/10
             print(a)
             plot(euler_imp(a,100)[0],erreur_imp(a,100))
             show()
             close()

0.0
```

Out[22]:



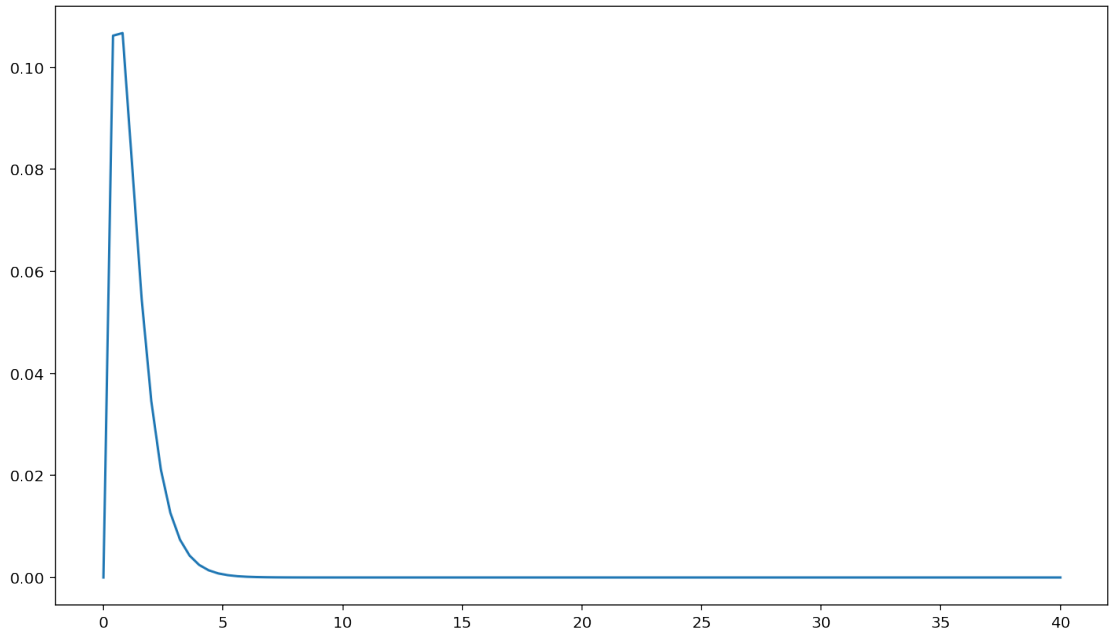

```
0.2
```

Out[22]:



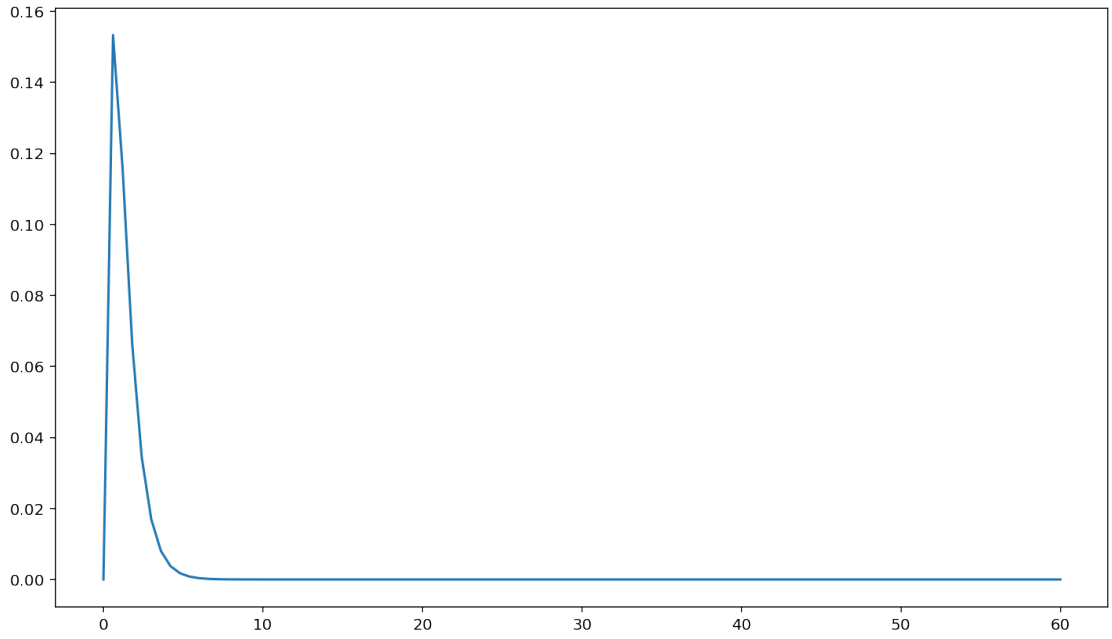
```
0.4
```

Out[22]:



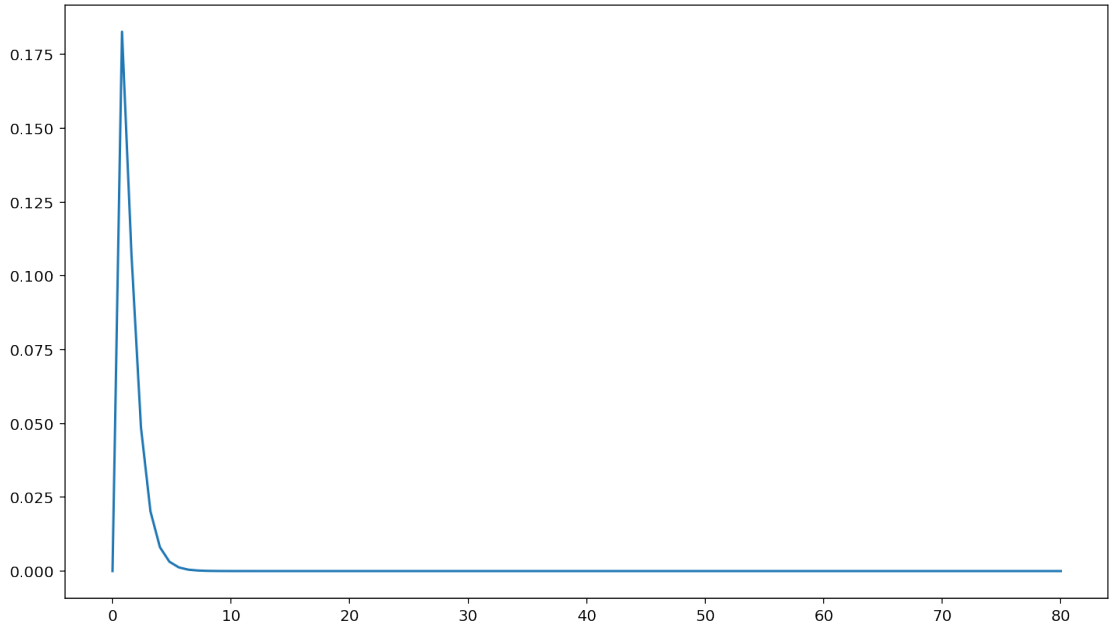

```
0.6
```

Out[22]:



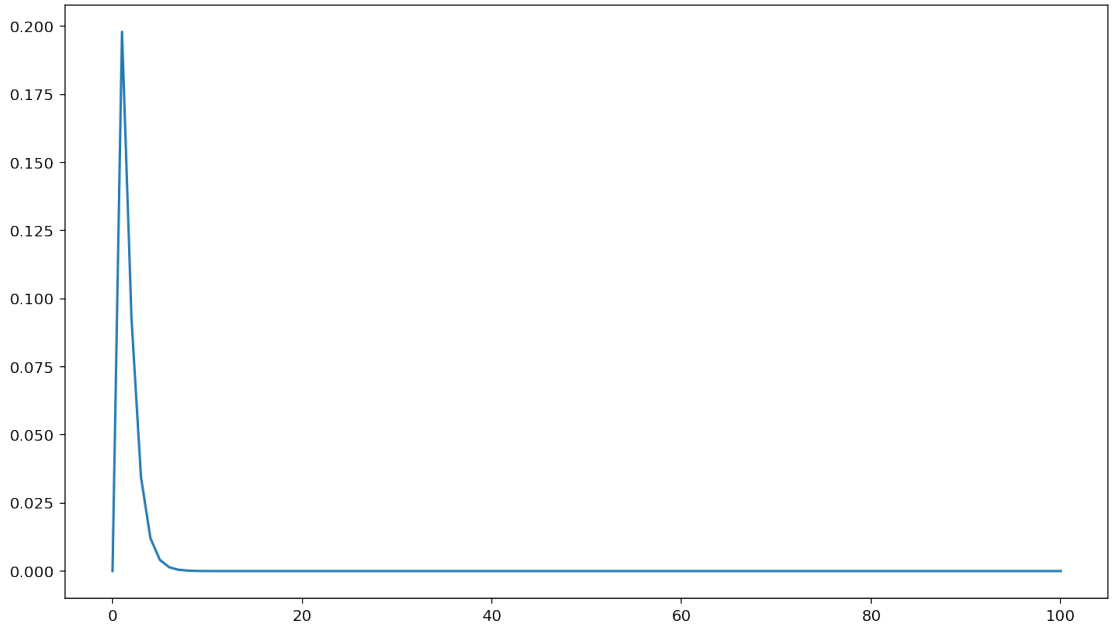
```
0.8
```

Out[22]:



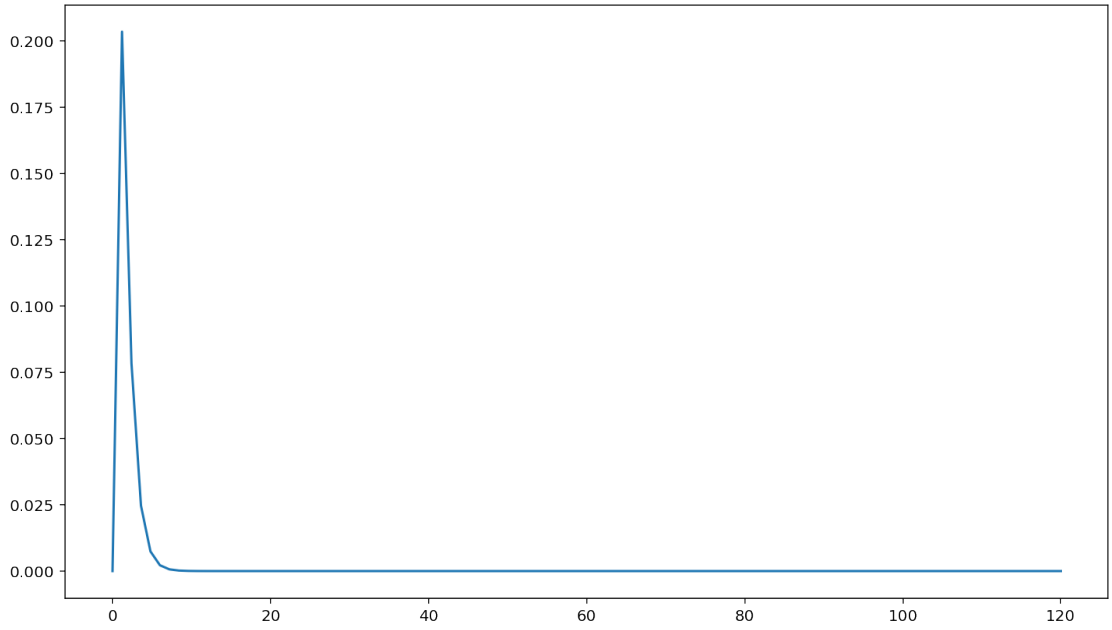

```
1.0
```

Out[22]:


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
20
40
60
80
100

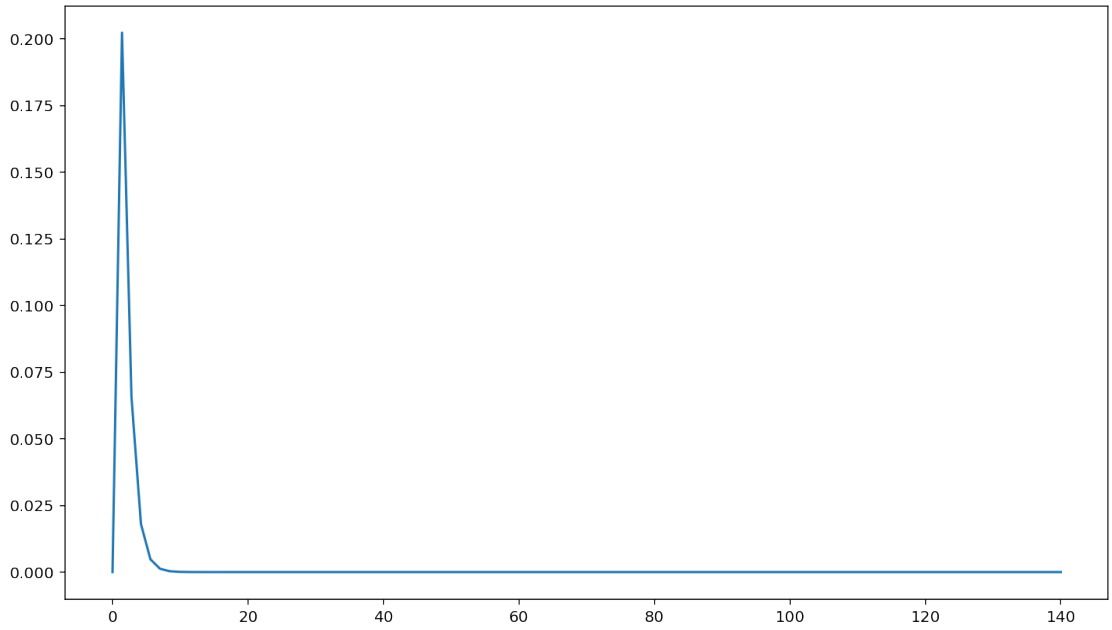
1.2

Out[22]:


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
20
40
60
80
100
120

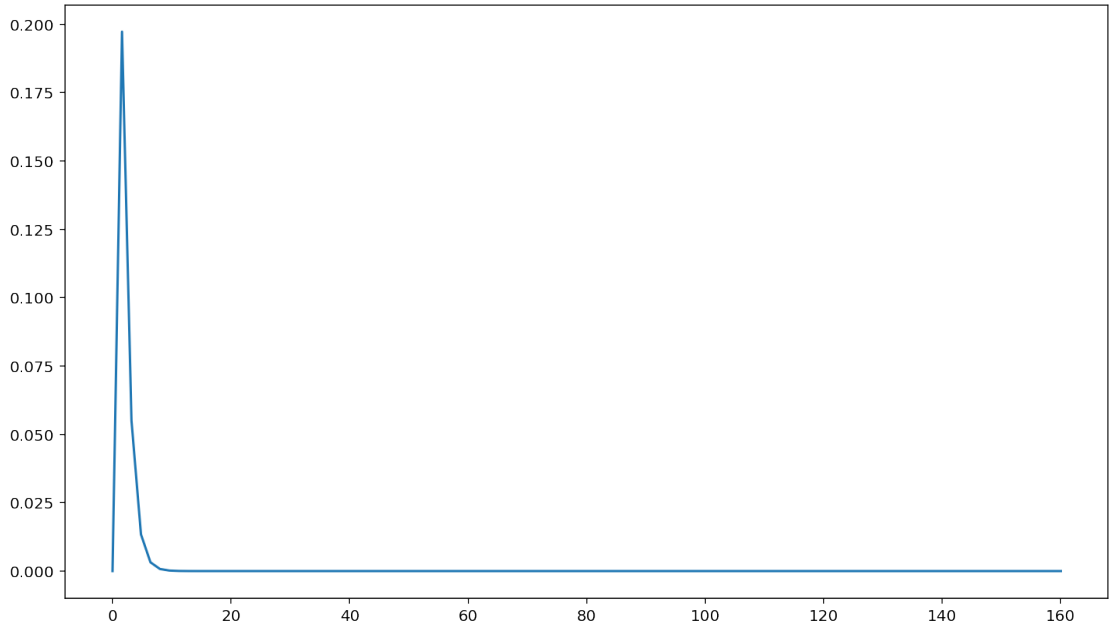
1.4

Out[22]:



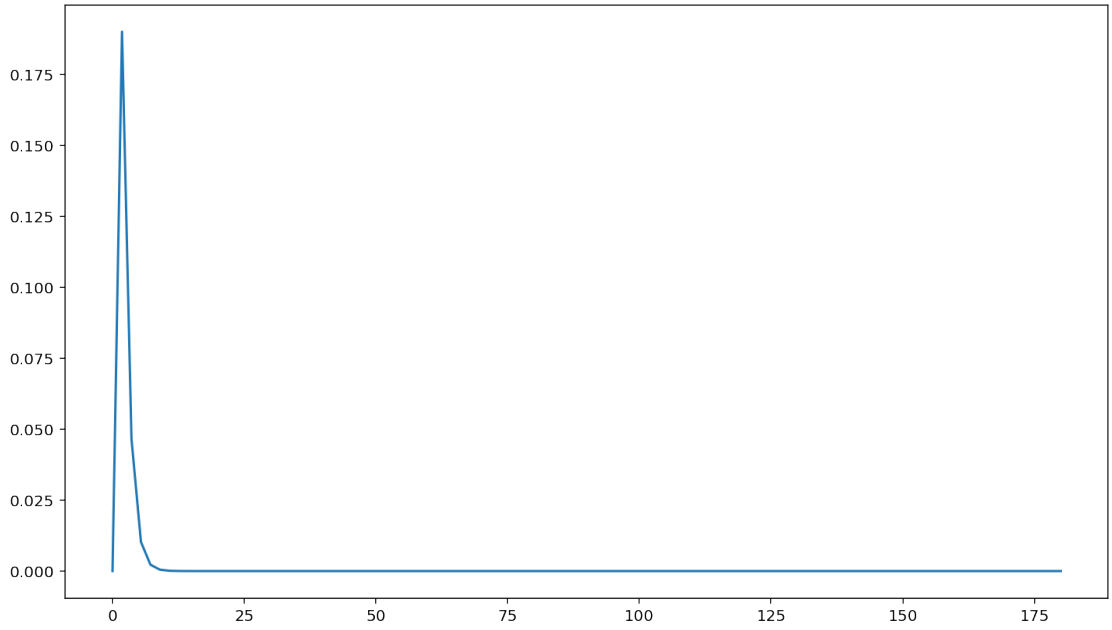
1.6

Out[22]:



```
1.8
```

Out[22]:



Ici il n'y a pas d'évolutions de stabilité à partir de h=1. Il n'y en aura pas non plus après. En effet, euler implicite consiste à coder une suite (x_k) telle que : $x_{k+1} = ax_k$ avec $a = \frac{1}{1+2h}$. Il s'agit d'une suite géométrique de raison a. Or pour $h > 0$, $a < 1$ donc a^n tend vers 0 quel que soit $h > 0$.

En revanche, pour euler explicite la raison est $r = 1 - 2h$ et pour $h > 1$, $1 - 2h < -1$ donc $|r^n|$ tend vers l'infini.

In [0]: