

```

# E:\TP info 24-25\TP3.py
001 | # Importations
002 |
003 | import numpy as np
004 | import matplotlib.pyplot as plt
005 |
006 | # Exercice 1 Balayage
007 | def balayage(f,a,e):
008 |     k = 0
009 |     while f(a)*f(a+k*e) > 0:
010 |         k += 1
011 |     return (a+(k-1)*e + a+k*e)/2, k
012 |
013 | ## 1
014 | def f1(x):
015 |     return x**2-2
016 |
017 | a,e = 1,10**(-6)
018 | print(balayage(f1,a,e))
019 |
020 | ## 2
021 | def f2(x):
022 |     return x**3-2*x-5
023 |
024 | Lx = np.linspace(0,3)
025 | Ly = f2(Lx)
026 | plt.grid()
027 | plt.plot(Lx,Ly)
028 | plt.show()
029 |
030 | a,e = 2,10**(-6)
031 | print(balayage(f2,a,e))
032 |
033 | ## 3
034 | def f3(x):
035 |     return 1-2**(x**2-6*x+9)
036 | Lx = np.linspace(-5,5)
037 | Ly = f3(Lx)
038 | plt.grid()
039 | plt.plot(Lx,Ly)
040 | plt.show()
041 |
042 | a,e = 2,10**(-6)
043 | print(balayage(f3,a,e))
044 |
045 | ## Exercice 2 : Dichotomie
046 | def dico(f,a,b,e):
047 |     k = 0
048 |     while abs(a-b) > e:
049 |         c = (a+b)/2

```

```

050|         if f(a)*f(c)<0:
051|             b = c
052|         else:
053|             a = c
054|             k += 1
055|     return c, k
056|
057| def f1(x):
058|     return x**2-2
059|
060| a,b,e = 1,2,10**(-6)
061| print(dicho(f1,a,b,e))
062|
063| ## 1
064| def f1(x):
065|     return x**5-x**4-5*x**3-x**2+4*x+3
066|
067| Lx = np.linspace(-2,2)
068| Ly = f1(Lx)
069| plt.grid()
070| plt.plot(Lx,Ly)
071| plt.show()
072|
073| e = 10**(-9)
074| for (a,b) in [(-2,-1),(1,2)]:
075|     print(dicho(f1,a,b,e))
076|
077| ## 2
078| def f2(x):
079|     return (x**2)*(4-x**2)-4/(x**2+1)
080|
081| # f2 est paire
082| Lx = np.linspace(0,2)
083| Ly = f2(Lx)
084| plt.grid()
085| plt.plot(Lx,Ly)
086| plt.show()
087|
088| e = 10**(-9)
089| for (a,b) in [(0,1),(1,2)]:
090|     print(dicho(f2,a,b,e))
091|
092| ## 3
093| def f3(x):
094|     return np.exp(-x)-x-2
095|
096| Lx = np.linspace(-1,1)
097| Ly = f3(Lx)
098| plt.grid()
099| plt.plot(Lx,Ly)

```

```

100| plt.show()
101|
102| a,b,e = -1,0,10**(-9)
103| print(dicho(f3,a,b,e))
104|
105| ## 4
106| def f4(x):
107|     return np.log(4-x**2)-x
108|
109| Lx = np.linspace(0,1.5)
110| Ly = f4(Lx)
111| plt.grid()
112| plt.plot(Lx,Ly)
113| plt.show()
114|
115| a,b,e = 1,2,10**(-9)
116| print(dicho(f3,a,b,e))
117|
118| ## 5
119| def f5(x):
120|     return np.sqrt(x**2-x+1)-2*np.sin(np.pi*x)
121|
122| Lx = np.linspace(-10,10,1000)
123| Ly = f5(Lx)
124| plt.grid()
125| plt.plot(Lx,Ly)
126| plt.show()
127|
128| e = 10**(-9)
129| for (a,b) in [(0,0.5),(0.5,1)]:
130|     print(dicho(f5,a,b,e))
131|
132| # Exercice 3
133| ## 1
134| def Lu(n):
135|     L = []
136|     for k in range(3,n+1):
137|         def f(x):
138|             return x**k + x**2 + 2*x - 1
139|         L.append(dicho(f,0,1,10**(-5))[0])
140|     return L
141|
142| import matplotlib.pyplot as plt
143| n = 15
144| Ln = range(3,n+1)
145| Lu = Lu(n)
146| plt.plot(Ln,Lu,marker = '.')
147| plt.show()
148|
149| ## 2

```

```

150| def f(x):
151|     return x**n*(x-1)-np.exp(-x)
152|
153| import matplotlib.pyplot as plt
154| import numpy as np
155| Lx = np.linspace(1.1,1.2,1000)
156| for n in range(3,50):
157|     Ly = f(Lx)
158|     plt.plot(Lx,Ly)
159|     plt.grid()
160|     plt.show()
161|
162| def Lu(n):
163|     L = []
164|     for k in range(3,n+1):
165|         def f(x):
166|             return x**k*(x-1)-np.exp(-x)
167|         L.append(dicho(f,1,2,10**(-5))[0])
168|     return L
169|
170| n = 100
171| Ln = range(3,n+1)
172| Lu = Lu(n)
173| plt.plot(Ln,Lu,marker = '.')
174| plt.show()
175|
176| ## Méthode de Newton
177| def Newton(f,fprime,a,e):
178|     x0 = a
179|     x1 = x0-f(x0)/fprime(x0)
180|     n = 1
181|     while abs(x1-x0) > e :
182|         x0, x1 = x1, x1-f(x1)/fprime(x1)
183|         n += 1
184|     return x1,n
185|
186| # Exercice 5
187| ## 1
188| def f1(x):
189|     return x**3-30
190|
191| def f1prime(x):
192|     return 3*x**2
193|
194| a, e = 1, 10**(-6)
195| print(Newton(f1,f1prime,a,e)[0])
196|
197| ## 2
198| def f2(x):
199|     return x**7-1000

```

```

200|
201| def f2prime(x):
202|     return 7*x**6
203|
204| a, e = 1, 10**(-6)
205| print(Newton(f2,f2prime,a,e)[0])
206|
207| ## Exerce 6
208| def f3(x):
209|     return 2*np.sin(x) - x
210|
211| Lx = np.linspace(0,2)
212| Ly = f3(Lx)
213| plt.plot(Lx,Ly)
214| plt.show()
215| ##
216| def f3prime(x):
217|     return 2*np.cos(x) - 1
218|
219| a, e = 1, 10**(-6)
220| print(Newton(f3,f3prime,a,e)[0])

```