

```

# E:\TP info 24-25\TP2 Exercices 1 à 6.py
001 | # Exercice 1
002 | import matplotlib.pyplot as plt
003 | import numpy as np
004 | ## 1
005 | def f1(x):
006 |     return x**2-x+1
007 | Lx = np.linspace(0,1)
008 | Ly = [f1(x) for x in Lx]
009 | # Ly = f1(Lx)
010 | plt.plot(Lx,Ly)
011 | plt.show()
012 |
013 | ## 2
014 | def f2(x):
015 |     return x*np.cos(2*np.pi*x)
016 | Lx = np.linspace(-1,1)
017 | Ly = [f2(x) for x in Lx]
018 | plt.plot(Lx,Ly)
019 | plt.show()
020 |
021 | ## 3
022 | def f3(x):
023 |     return x**2-2
024 | Lx = np.linspace(1,2)
025 | Ly = [f3(x) for x in Lx]
026 | plt.plot(Lx,Ly)
027 | plt.show()
028 |
029 | ## 4
030 | def f4(x):
031 |     return 1/(x**2 + 1)
032 | Lx = np.linspace(0,1)
033 | Ly = [f4(x) for x in Lx]
034 | plt.plot(Lx,Ly)
035 | plt.show()
036 |
037 |
038 | ## 5
039 | def f5(x):
040 |     return x*np.exp(-x)
041 | Lx = np.linspace(-1,1,200)
042 | Ly = [f5(x) for x in Lx]
043 | plt.plot(Lx,Ly)
044 | plt.show()
045 |
046 |
047 | ## 6
048 | def f6(x):
049 |     return np.log(1+x)

```

```

050| Lx = np.linspace(1,2)
051| Ly = [f6(x) for x in Lx]
052| plt.plot(Lx,Ly)
053| plt.show()
054|
055| ## Exercice 2
056| def Riemann(f,a,b,n):
057|     h = (b-a)/n
058|     S = 0
059|     for k in range(n):
060|         S += f(a+k*h)
061|     return h*S
062|
063| def f(t):
064|     return 4/(1+t**2)
065|
066| a, b = 0, 1
067| n = 1000
068| print(Riemann(f,a,b,n))
069|
070| def best_n(p):
071|     a, b = 0, 1
072|     n = 1
073|     while abs(Riemann(f,a,b,n)-np.pi) > 10**(-p):
074|         n += 1
075|     return n
076|
077| print(best_n(3))
078|
079| for p in range(1,5):
080|     print(best_n(p))
081| ##
082| Lp = range(1,10)
083| Lbest = [best_n(p) for p in Lp]
084| plt.plot(Lp,Lbest)
085| plt.show()
086|
087| # Exercice 3
088| ## 1
089| def f1(x):
090|     return 2-x**2
091|
092| a, b = 0, 2
093| n = 1000
094| print(Riemann(f1,a,b,n))
095|
096| ## 2
097| def f2(x):
098|     return np.log(x) - 1
099|

```

```

100| a, b = 1, 2
101| n = 1000
102| print(Riemann(f2,a,b,n))
103|
104| ## 3
105| def f3(x):
106|     return np.sqrt(x) - 2
107|
108| a, b = 1, 6
109| n = 1000
110| print(Riemann(f3,a,b,n))
111|
112| ## 4
113| def f4(x):
114|     return x - 2*np.sin(2*x)
115|
116| a, b = 0, 3
117| n = 100
118| print(Riemann(f4,a,b,n))
119|
120| ## Exercice 4
121| def u(n):
122|     def f(x):
123|         return x**2 * np.sin(n*x)
124|     return Riemann(f,0,1,100)
125|
126| def liste(n):
127|     return [u(k) for k in range(n+1)]
128|
129| n = 100
130| Ln = range(n+1)
131| Lu = liste(n)
132| plt.plot(Ln,Lu)
133| plt.show()
134|
135| # on conjecture que les deux suites extraites sont
adjacentes
136| # donc, que la suite est convergente
137|
138| # Exercice 5
139| ## 1
140| def F1(x):
141|     def f(t):
142|         return np.exp(-t*x)/(1+t)
143|     return Riemann(f,0,1,100)
144|
145| Lx = np.linspace(1,5,500)
146| Ly = [F1(x) for x in Lx]
147| plt.plot(Lx,Ly)
148| plt.show()

```

```

149|
150| ## 2
151| def F2(x):
152|     def f(t):
153|         return np.sin(x+t)/(x+t)
154|     return Riemann(f,0,1,100)
155|
156| Lx = np.linspace(2,4,200)
157| Ly = [F2(x) for x in Lx]
158| plt.plot(Lx,Ly)
159| plt.show()
160|
161| ## 3
162| def F3(x):
163|     def f(t):
164|         return np.exp(-x*(1+t**2))/(1+t**2)
165|     return Riemann(f,0,1,100)
166|
167| Lx = np.linspace(-1,1,200)
168| Ly = [F3(x) for x in Lx]
169| plt.plot(Lx,Ly)
170| plt.show()
171|
172| ## Exercice 6
173| def Rectangles(a,b,h,f):
174|     n = int(np.floor((b-a)/h))
175|     return Riemann(f,a,b,n)
176|
177| def f(t):
178|     return 4/(1+t**2)
179|
180| print(Rectangles(0,1,0.001,f))
181|
182| def rectangles(a,b,h,f):
183|     L = np.arange(a,b,h)
184|     S = 0
185|     for el in L:
186|         S += f(el)
187|     return h*S
188|
189| print(rectangles(0,1,0.001,f))
190|
191| ## Exercice 7
192| # y(x) = exp(-x)
193| import numpy as np
194| import matplotlib.pyplot as plt
195| a, b = 0, 1
196| h = 0.2
197| Lx = np.arange(a,b,h)
198| n = int(np.floor((b-a)/h))

```

```

199| Ly = [1]
200| for k in range(n-1):
201|     Ly.append((1-h)*Ly[-1])
202| print(Ly)
203|
204| for p in [1,2]:
205|     h = 10**(-p)
206|     Lx = np.arange(a,b,h)
207|     n = int(np.floor((b-a)/h))
208|     Ly = [1]
209|     for k in range(n-1):
210|         Ly.append((1-h)*Ly[-1])
211|     plt.plot(Lx,Ly)
212| Lx = np.linspace(0,1)
213| Ly = [np.exp(-x) for x in Lx]
214| plt.plot(Lx,Ly)
215| plt.show()
216|
217| ## Exercice 8
218| import numpy as np
219| import matplotlib.pyplot as plt
220| def F(x,y):
221|     return -2*x*y+x*np.exp(-x**2)*y**3
222|
223| a, b = 0, 2
224| h = 0.02
225| Lx = np.arange(a,b,h)
226| n = int(np.floor((b-a)/h))
227| Ly = [1]
228| for k in range(n-1):
229|     Ly.append((Ly[k]+h*F(Lx[k],Ly[k])))
230| plt.plot(Lx,Ly)
231|
232| def f(x):
233|     return np.sqrt(3/(np.exp(-x**2)+2*np.exp(2*x**2)))
234| Lexact = f(Lx)
235| plt.plot(Lx,Lexact)
236| plt.show()
237|
238| ## Exercice 9
239| import numpy as np
240| import matplotlib.pyplot as plt
241| def F(x,y):
242|     return -(2/x)*y+x**2
243|
244| a, b = 0.05, 2
245| h = 0.025
246| Lx = np.arange(a,b,h)
247| n = int(np.floor((b-a)/h))
248| Ly = [2/5]

```

```

249| for k in range(n-1):
250|     Ly.append((Ly[k]+h*F(Lx[k],Ly[k])))
251| plt.plot(Lx,Ly,'black')
252|
253| def f(x):
254|     return 0.001/x**2+(x**3)/5
255| Lexact = f(Lx)
256| plt.plot(Lx,Lexact,'red')
257| plt.show()
258|
259| ##
260| a, b = 1, 200
261| h = 5
262| Lx = np.arange(a,b,h)
263| n = int(np.floor((b-a)/h))
264| Ly = [2/5]
265| for k in range(n-1):
266|     Ly.append((Ly[k]+h*F(Lx[k],Ly[k])))
267| plt.plot(Lx,Ly,'black')
268|
269| def f(x):
270|     return 0.001/x**2+(x**3)/5
271| Lexact = f(Lx)
272| plt.plot(Lx,Lexact,'red')
273| plt.show()

```