

```
# Centrale_2015_ecrit.py
```

```
01 |
02 | G, N = 1, 10
03 |
04 | # IB
05 | def smul(x, L):
06 |     return [x*u for u in L]
07 | # IC1
08 | def vsum(L1, L2):
09 |     return [L1[k] + L2[k] for k in range(len(L1))]
10 | # IC2
11 | def vdif(L1, L2):
12 |     return [L1[k] - L2[k] for k in range(len(L1))]
13 |
14 |
15 | # IIB2 l'équation est autonome, tmin et tmax inutiles
16 | def euler(y0, z0, h, n, f):
17 |     yi, zi = y0, z0
18 |     y, z = [y0], [z0]
19 |     for i in range(1, n):
20 |         yi, zi = yi + h*zi, zi + h*f(yi)
21 |         y.append(yi)
22 |         z.append(zi)
23 |     return y, z
24 |
25 |
26 | # IIC1
27 | def verlet(y0, z0, h, n, f):
28 |     yi, zi = y0, z0
29 |     y, z = [y0], [z0]
30 |     for i in range(1, n):
31 |         fi = f(yi)
32 |         yi = yi + h*zi + (h**2*fi)/2
33 |         zi = zi + h*(fi + f(yi))/2
34 |         y.append(yi)
35 |         z.append(zi)
36 |     return y, z
37 |
38 |
39 | # IIIA2
40 | def force2(m1, p1, m2, p2):
41 |     d2 = sum([x**2 for x in vdif(p1, p2)])
42 |     cte = m1*m2*G / d2**1.5
43 |     return smul(cte, vdif(p2, p1))
44 |
45 |
```

```

46| # IIIA3
47| def forceN(j, m, pos):
48|     force = 3*[0]
49|     for k in range(N):
50|         if k != j:
51|             force_k = force2(m[j], pos[j], m[k],
pos[k])
52|             force = vsom(force, force_k)
53|     return force
54|
55| # IIIB2
56| def pos_suiv(m, pos, vit, h):
57|     resultat = []
58|     for j in range(N):
59|         acceleration = smul(forceN(j, m, pos), 1/m[j])
60|         pos_s_j = vsom(vsom(pos[j], smul(h, vit[j])),
smul(h**2/2, acceleration))
61|         resultat.append(pos_s_j)
62|     return resultat
63|
64|
65| import matplotlib.pyplot as plt
66| import numpy as np
67|
68| def f(y):
69|     return - 4*(np.pi)**2*y
70|
71| y0, z0 = 3, 0
72| n, h = 100, 1/33
73|
74| y, z = euler(y0, z0, h, n, f)
75| plt.plot(y, z)
76|
77| y, z = verlet(y0, z0, h, n, f)
78| plt.plot(y, z)
79| plt.show()
80|

```