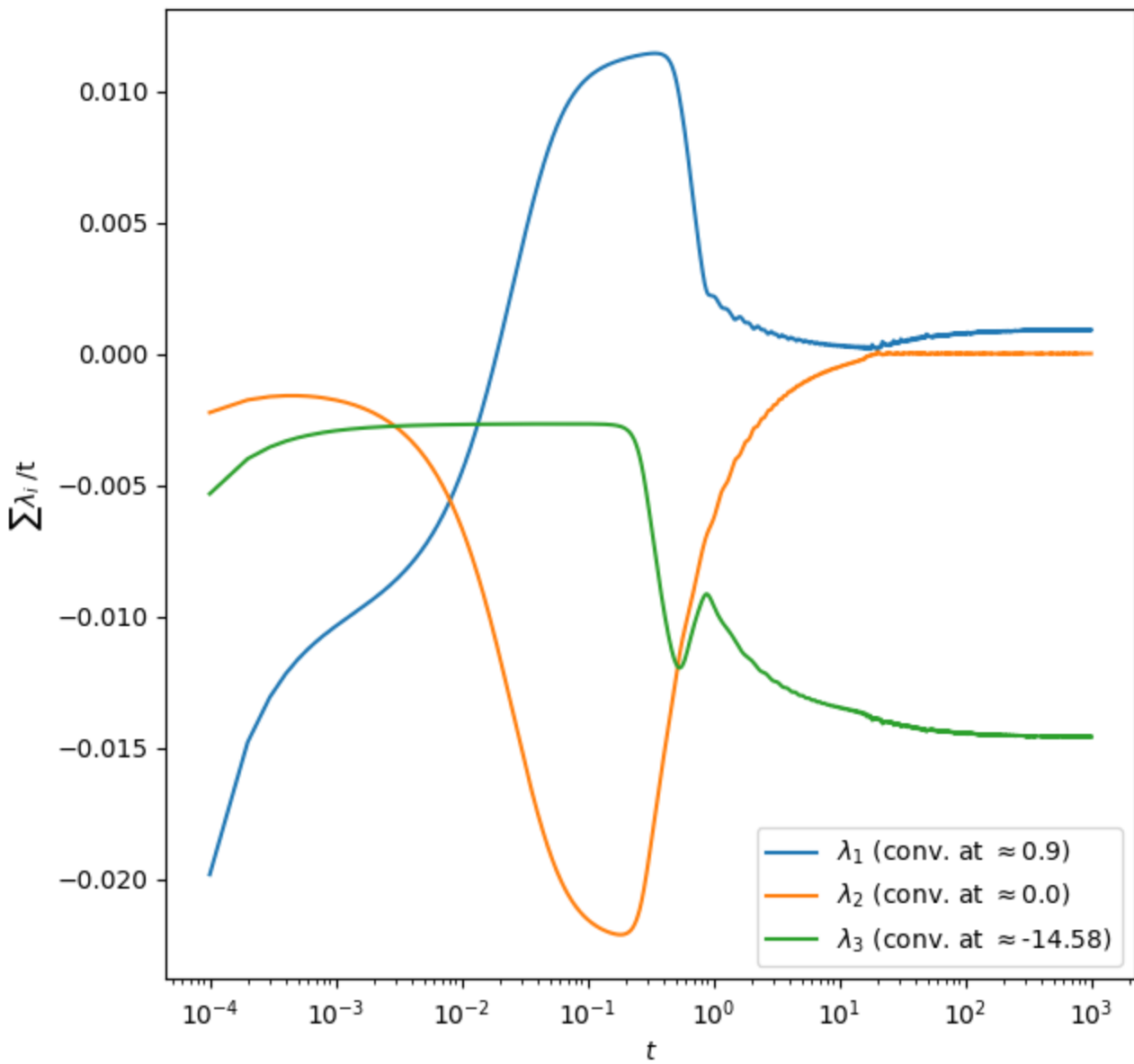
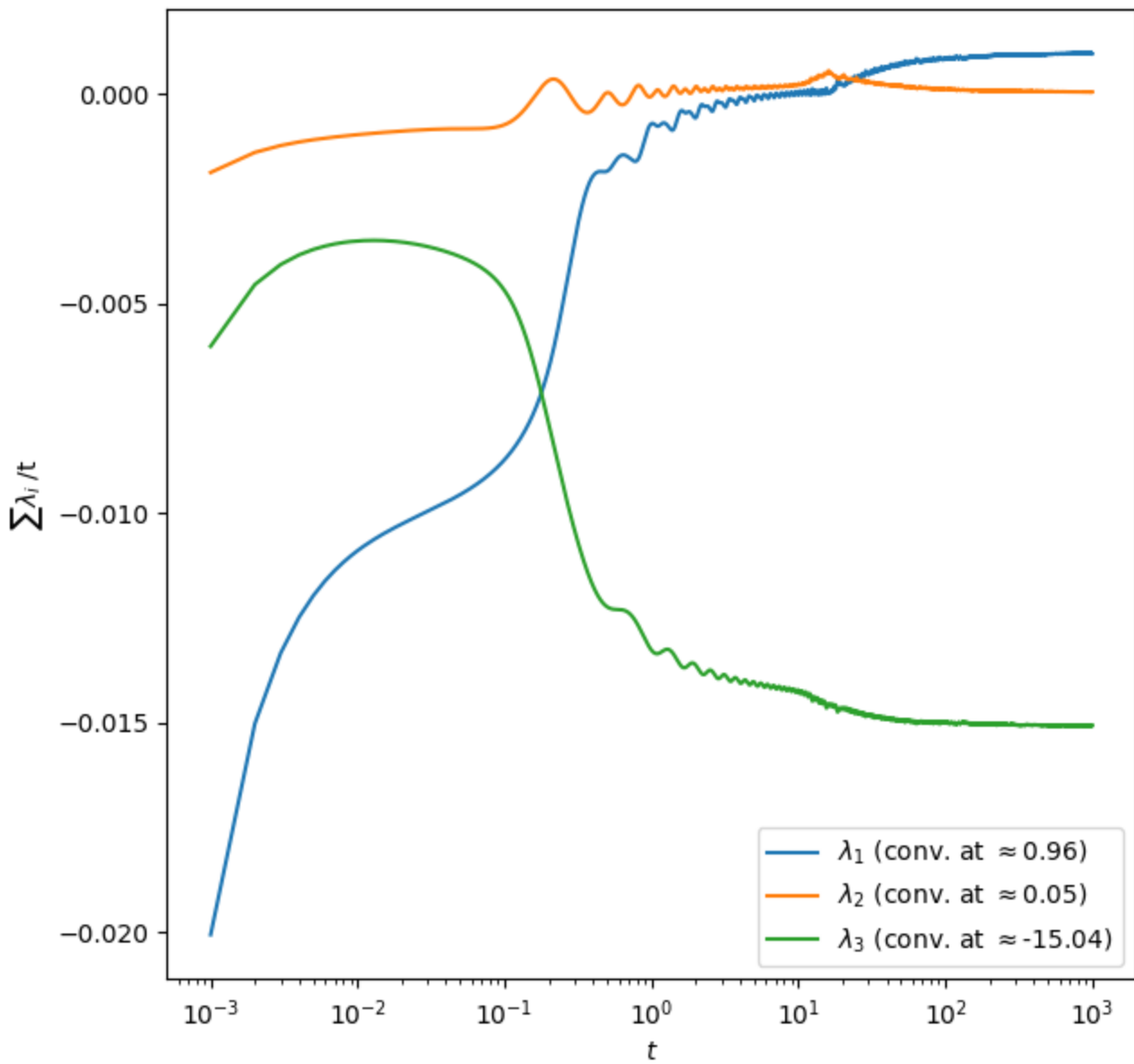


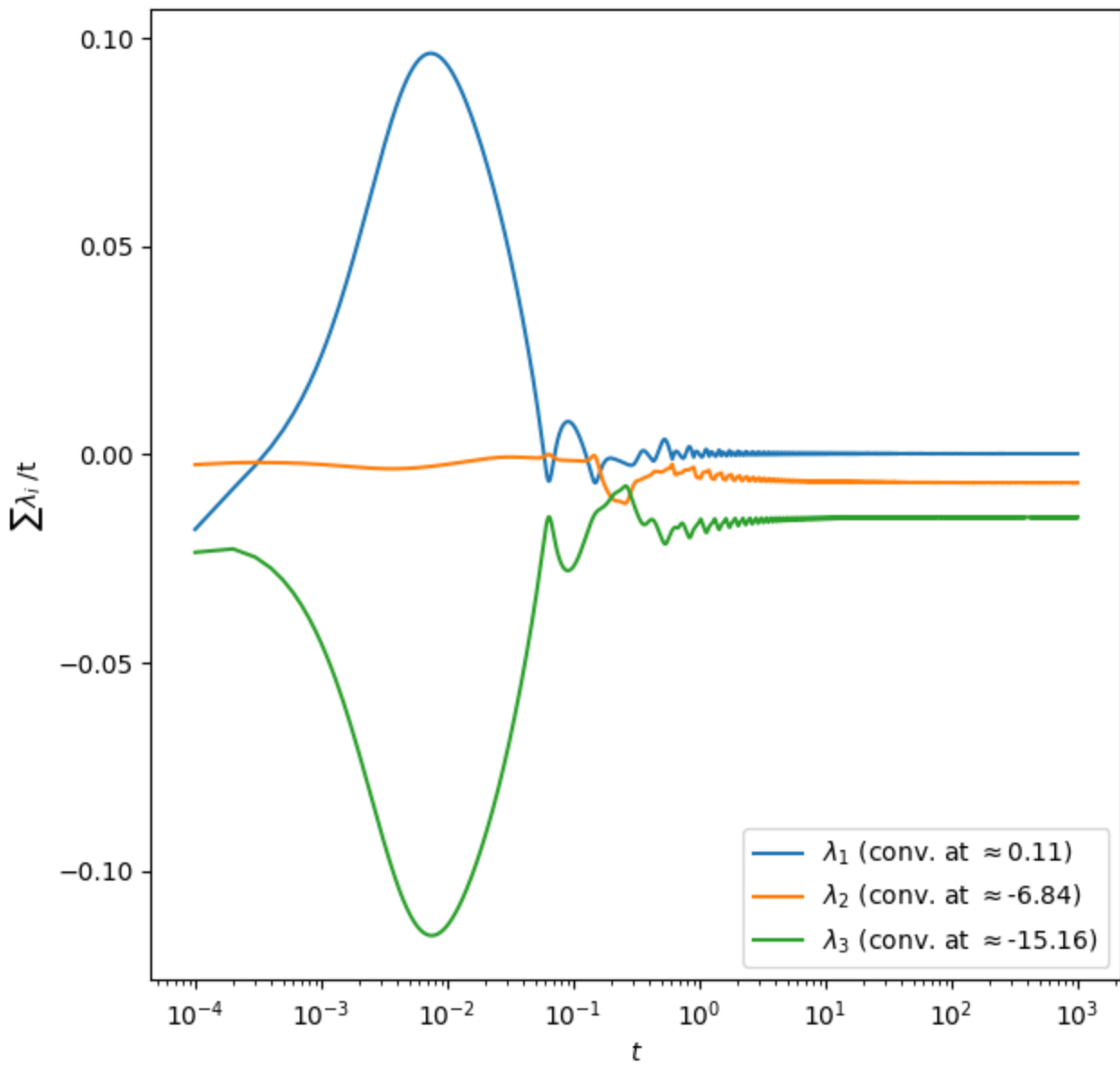
3.3ab



3.3cd



3.3ef



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4 from scipy.linalg import qr
5 import sys
6
7 title = '3.3cd'
8 sigma = 10
9 r = 28
10 b = 3
11
12 def dynamical_system(IC, t):
13     x = IC[0]
14     y = IC[1]
15     z = IC[2]
16
17     dxdt = sigma*(y-x)
18     dydt = r*x-y-x*z
19     dzdt = x*y-b*z
20
21     return [dxdt, dydt, dzdt]
22
23 # Numerical integration
24 T = 10**3
25 dt = 10**-3
26 t_array = np.arange(0,T,dt)
27
28 x = 0.01
29 y = 0.01
30 z = 0.01
31
32 IC = [x,y,z]
33 eqs = odeint(dynamical_system, IC, t_array)
34
35 cut_tail = 2000
36
37 x = eqs[cut_tail:,0]
38 y = eqs[cut_tail:,1]
39 z = eqs[cut_tail:,2]
40
41 new_t_array = t_array[cut_tail:]-cut_tail*dt
42 new_T = T-cut_tail*dt
43
44 I = np.identity(3)
45 Q = I.copy()
46
47 lambda_one = np.zeros_like(new_t_array)
48 lambda_two = lambda_one.copy()
49 lambda_three = lambda_one.copy()
50
51 print('No memory error')
52
53 for t in range(len(new_t_array)):
54
55     J11 = -sigma
56     J12 = sigma
57     J13 = 0
58
59     J21 = r-z[t]
60     J22 = -1
61     J23 = -x[t]
62
63     J31 = y[t]
64     J32 = x[t]
65     J33 = -b
66
67     J = np.array([[J11,J12,J13],\
68                  [J21,J22,J23],\
69                  [J31,J32,J33]])
70
71     M = I+J*dt
72     Q,R = qr(np.matmul(M,Q))
73
74     lambda_one[t] = lambda_one[t-1] + np.log(np.absolute(R[0,0]))/new_T
75     lambda_two[t] = lambda_two[t-1] + np.log(np.absolute(R[1,1]))/new_T
76     lambda_three[t] = lambda_three[t-1] + np.log(np.absolute(R[2,2]))/new_T
77
78     if t % 100000 == 0:

```

```

79     print(t)
80
81 print('Done: QR')
82
83 lambda_one_converged = lambda_one[-1]
84 lambda_two_converged = lambda_two[-1]
85 lambda_three_converged = lambda_three[-1]
86
87 print('Lambda 1: ', lambda_one_converged)
88 print('Lambda 2: ', lambda_two_converged)
89 print('Lambda 3: ', lambda_three_converged)
90 print('Sum of lambdas: ', lambda_one[-1]+lambda_two[-1]+lambda_three[-1])
91
92 # # Trajectory
93 # fig1, ax1 = plt.subplots(subplot_kw={"projection": "3d"}, figsize=(7,7))
94 # ax1.plot(x, y, z, '-', linewidth=0.1)
95
96 # Lambdas
97 fig2, ax2 = plt.subplots(figsize=(7,7))
98 ax2.plot(new_t_array[1:], np.divide(lambda_one[1:],new_t_array[1:]), '-',
99        linewidth=1.5, label=r'$\lambda_1$ (conv. at $\approx$' +
100        '{}'.format(np.round(lambda_one_converged,2)))
101 ax2.plot(new_t_array[1:], np.divide(lambda_two[1:],new_t_array[1:]), '-',
102        linewidth=1.5, label=r'$\lambda_2$ (conv. at $\approx$' +
103        '{}'.format(np.round(lambda_two_converged,2)))
104 ax2.plot(new_t_array[1:], np.divide(lambda_three[1:],new_t_array[1:]), '-',
105        linewidth=1.5, label=r'$\lambda_3$ (conv. at $\approx$' +
106        '{}'.format(np.round(lambda_three_converged,2)))
107 ax2.set_xscale('log')
108 ax2.set_xlabel('$t$')
109 ax2.set_ylabel('$\sum \lambda_i$ /t')
110 ax2.set_box_aspect(1)
111 ax2.set_title(title)
112
113 plt.legend(loc="lower right", prop={'size': 10})
114 plt.savefig('Dynamical systems/DS HW3/3.3/' + title + '.png', bbox_inches='tight')
115 # plt.show()

```