Simulating the non-equilibrium thermodynamics of quantum dots
Stationary state - two single-level quantum dots in contact with three reservoirs

Bonamino Luca

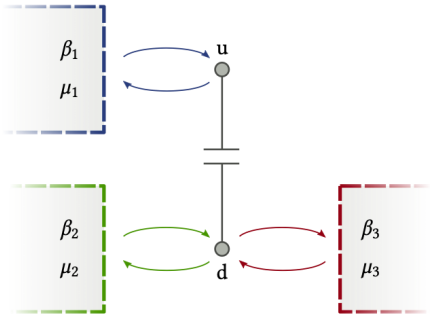


Figure 1: Set Up Scheme [1]

1 Setup

Two quantum dots identified as dot up (u) and dot down (d) with a fixed energies ε_u and ε_d and coupled with three reservoirs. The whole system is in a stationary state.
The dot u is coupled with the reservoir 1 with chemical potential μ_1 and temperature T_1, while the dot d is coupled with the reservoirs 2 and 3 with respectively chemical potentials μ_2 and μ_3 and temperatures T_2 and T_3. Electrons cannot jump from one dot to the other.

Identifying by ξ the dot ($\xi = u$ or $\xi = d$) the dots-reservoirs relations are described by the rates W_+^ξ and W_-^ξ to increase and decrease the population of the dots. The overall upward [resp. downward] rate on a dots ξ is the sum of all upwards [resp. downwards] rates from the reservoirs ν acting on the dot ξ.

$$W_+^\xi = \sum_{\nu \in \nu_\xi} W_+^\nu \tag{1}$$

$$W_-^\xi = \sum_{\nu \in \nu_\xi} W_-^\nu \tag{2}$$

where ν_ξ is the set of νs acting on the dot ξ.

The rates to increase [resp. decrease] the population of a dot from a reservoir are proportional to the Fermi-Dirac distribution [resp. one minus the Fermi-Dirac distribution] of that reservoir.

$$W_+^\xi = \sum_{\nu\in\nu_\xi} W_+^\nu = \sum_{\nu_\xi} a_\nu^\xi f_\nu \tag{3}$$

$$W_-^\xi = \sum_{\nu\in\nu_\xi} W_-^\nu = \sum_{\nu\in\nu_\xi} a_\nu^\xi (1-f_\nu) \tag{4}$$

where f_ν is the fermi distribution of the reservoir ν and a_ν^ξ is a constant taking into account tunneling factors between the reservoirs and the dots.
The reservoir 1 only interacts with the dot u, while dot reservoirs 2 and 3 interacts with the dot d

$$\xi = u \implies \nu_\xi = \{1\}$$
$$\xi = d \implies \nu_\xi = \{2,3\}$$

hence

$$\begin{aligned} W_+^u &= W_+^{(1)} \\ W_-^u &= W_-^{(1)} \end{aligned} \tag{5}$$

$$\begin{aligned} W_+^d &= W_+^{(2)} + W_+^{(3)} \\ W_-^d &= W_-^{(2)} + W_-^{(3)} \end{aligned} \tag{6}$$

The temperatures T_2 and T_3 and chemical potentials μ_2 and μ_3 are imposed to be such that $T_3 > T_2$ and that $f_3 > f_2$.

$$f_3 > f_2 \iff 1-f_3 < 1-f_2$$
$$\implies W_+^{(3)} > W_+^{(2)} \iff W_-^{(3)} < W_-^{(2)} \tag{7}$$

The intention is to study the stationary dynamics of this model.

2 A First Description

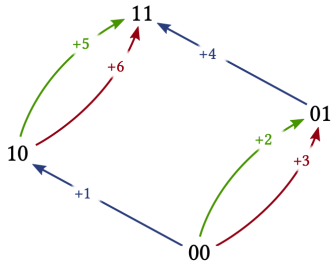


Figure 2: Transition Network [1]

The system has four states: 00, 01, 10 and 11 where the first entry refers to the occupation of the dot u and the second entry to the occupation of the dot d.

Six transitions are possible between these states

transition 1: $00 <->10,\ \nu = 1$

transition 2: $00 <->01,\ \nu = 2$

transition 2: $00 <->01,\ \nu = 3$

transition 3: $01 <->11,\ \nu = 2$

transition 3: $01 <->11,\ \nu = 3$

transition 4: $10 <->11,\ \nu = 1$

Each transition has a correspond-

ing rate from the implied reservoirs

$$\begin{aligned} W_{10,00} &= W^{(1)}_{10,00} = W^u_+ \\ W_{00,10} &= W^{(1)}_{00,10} = W^u_- \end{aligned} \tag{8}$$

$$\begin{aligned} W_{01,00} &= W^{(2)}_{01,00} + W^{(3)}_{00,01} = W^d_+ \\ W_{00,01} &= W^{(2)}_{00,01} + W^{(3)}_{00,01} = W^d_- \end{aligned} \tag{9}$$

$$\begin{aligned} W_{11,01} &= W^{(1)}_{11,01} = W^u_+ \\ W_{01,11} &= W^{(1)}_{01,11} = W^u_- \end{aligned} \tag{10}$$

$$\begin{aligned} W_{11,10} &= W^{(2)}_{11,10} + W^{(3)}_{11,10} = W^d_+ \\ W_{10,11} &= W^{(2)}_{10,11} + W^{(3)}_{10,11} = W^d_- \end{aligned} \tag{11}$$

Each transition results in a change in the population of the states of the system: defining as N the population of a state, for the transition i

$$\Delta\mathbf{N}_i = \begin{pmatrix} \Delta N^i_{00} \\ \Delta N^i_{10} \\ \Delta N^i_{01} \\ \Delta N^i_{11} \end{pmatrix}$$

Only one transition can occur at the time, the incidence matrix [1] can be hence constructed as

$$D = (\Delta\mathbf{N}_i)_{i\in\{1,2,3,4,5,6\}}$$

Hence

$$D = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{12}$$

3 Ensemble-level Description

The stationary probability distribution of a state is found by solving the master equation

$$dp_n = \sum_m W_{n,m} p_m = 0 \tag{13}$$

where $n = (0,0), (1,0), (0,1)$ or $(1,1)$. To find the probability occupation [resp. emptiness] of the two dots we need to solve for

$$\begin{aligned} p^{u,ss}_+(t) &= p^{ss}_{10}(t) + p^{ss}_{11}(t) \\ p^{u,ss}_-(t) &= p^{ss}_{01}(t) + p^{ss}_{00}(t) \end{aligned} \tag{14}$$

$$\begin{aligned} p^{d,ss}_+(t) &= p^{ss}_{01}(t) + p^{ss}_{11}(t) \\ p^{d,ss}_-(t) &= p^{ss}_{10}(t) + p^{ss}_{00}(t) \end{aligned} \tag{15}$$

$$\tag{16}$$

which is different than to solve for the stationary distribution of a two quantum dots in two different non-communicating systems. This is due to the fact that here, the probability of occupancy of a dot directly depends on the probability of emptiness of the other.

From now own, the ss index will be neglect since the stationary distribution is the only one that will appear.

The stationary currents from the reservoirs are

$$\begin{aligned} I^u_1 = I_1 =& W^{(1)}_{10,00} p_{00} - W^{(1)}_{00,10} p_{10} + \\ & W^{(1)}_{11,01} p_{01} - W^{(1)}_{01,11} p_{11} \end{aligned} \tag{17}$$

$$\begin{aligned} I^d_3 = -I^d_2 =& W^{(3)}_{01,00} p_{00} - W^{(3)}_{00,01} p_{01} + \\ & W^{(3)}_{11,10} p_{10} - W^{(3)}_{10,11} p_{11} \end{aligned} \tag{18}$$

hence

$$I^u_1 = I_1 = W^{(1)}_+ p^u_- + W^{(1)}_- p^u_+ \tag{19}$$

$$I^d_3 = I_3 = W^{(3)}_+ p^d_- + W^{(3)}_- p^d_+ \tag{20}$$

and the heat rate

$$\dot{Q}_\nu = I_{\nu,\varepsilon} - \mu_\nu I_{\nu,m} \tag{21}$$

where $I_{\nu,\varepsilon}$ is the energy current from the reservoir ν and $I_{\nu,m}$ is the matter current from the reservoir ν.
However, because of being in a stationary state, the only way to give energy to the dot is to put an electron in the dots level, hence there is a proportionality between the energy current and the matter current. The proportionality constant is the energy of the dot's level ε_u or ε_d

$$I_{1,\varepsilon} = \varepsilon_u I_{1,m} = I_1$$
$$I_{3,\varepsilon} = \varepsilon_d I_{3,m} = I_3$$

from which

$$\dot{Q}_1 = (\varepsilon_u - \mu_1) I_1 \tag{22}$$
$$\dot{Q}_3 = -\dot{Q}_2 = (\varepsilon_d - \mu_3) I_3 \tag{23}$$

The overall chemical work rate is given by

$$\dot{w}_{chem} = \sum_{\nu=r,l} I_{\nu,m}\mu_\nu = \frac{I_1}{\varepsilon_u}\mu_1 + (\mu_3 - \mu_2) I_3 \tag{24}$$

The entropy productions rate for the two dots are given by

$$\dot{S}_i^u = I_1 \ln\left(\frac{W_+^{(1)} p_-^u}{W_-^{(1)} p_+^u}\right) \tag{25}$$

and using that $I_2 = -I_3$

$$\begin{aligned}\dot{S}_i^d &= \sum_{\nu=2,3} I_\nu \ln\left(\frac{W_+^\nu (p_{00}+p_{10})}{W_-^\nu (p_{11}+p_{01})}\right) \\ &= I_3 \ln\left(\frac{W_+^{(3)} W_-^{(2)}}{W_+^{(3)} W_-^{(2)}}\right)\end{aligned} \tag{26}$$

Hence, the total entropy production rate

$$\begin{aligned}\dot{S}_i &= \dot{S}_i^u + \dot{S}_i^d \\ &= I_1 \ln\left(\frac{W_+^{(1)} p_-^u}{W_-^{(1)} p_+^u}\right) \\ &+ I_3 \ln\left(\frac{W_+^{(3)} W_-^{(2)}}{W_+^{(3)} W_-^{(2)}}\right)\end{aligned} \tag{27}$$

4 Trajectory-level Description and Simulation

The micro state of the system is described by a two-dimensional array $p = [.,.]$ where the first entry is the micro state of the dot u and the second entry is the micro state of the dot d. Because both dot's can be populated by maximum one electron, their micros states corresponds to their populations.

To simulate the dynamics of the model, the times of the transitions τ_u and τ_d of both dots are calculated via the Gillespie algorithm, using

$$\tau_\xi = \frac{\ln r_\xi}{rate_\xi} \tag{28}$$

where $\xi = u$ and d, $r_\xi \in (0,1)$ is a uniformly distributed number between 1 and 0, and

$$rate_\xi = (1 - p[i_\xi])\, W_+^\xi + p[i_\xi] W_-^\xi \tag{29}$$

where $i_u = 0$ and $i_d = 1$ are the indexes indicating the micro states of the dot u and d.

The smaller between the two times τ_u and τ_d is taken as the time of the transition.
This first step gives the knowledge of

with which of the two dots the transition is occurring.
Giving this knowledge and the state of micro state of the dot, the type of transition occurring is found by checking the relative size of a uniformly distributed number r with the rates corresponding to the dot to which the transition is occurring, following the restriction (7) given in the setup **section (1)** - see code in the **appendix (6)**.
Depending on which transition occurs the micro states of the two dot change and the change of population ΔN_i corresponding to the transition i takes the value of 1 [resp. -1] if the dot is filled [resp. emptied].
In the same way the micro state thermodynamic quantities change according to the dynamics: increasing [resp. decreasing] the value of the heat contribution by an amount of $\varepsilon_\xi - \mu_\nu$ of the dot ξ and reservoir ν taking part in the transition. The chemical work contribution at the transition is the current I_ν times chemical potential μ_ν

5 Results and Conclusions

5.1 Simulation Results

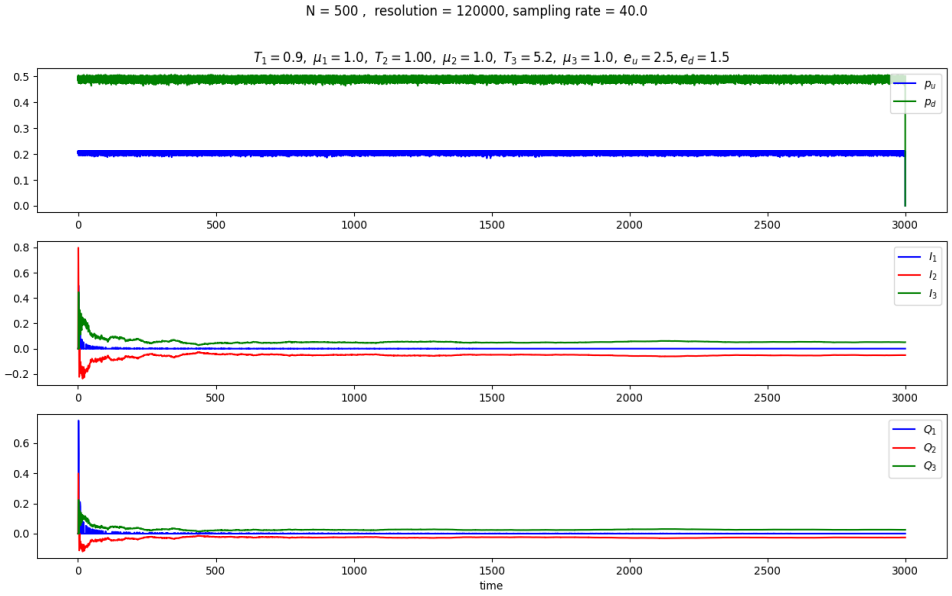
Figure 3 represents the results of the simulation of the out of equilibrium regime 3a and the equilibrium regime 3b.
In the middle and bottom plots, the blue, red and green colors are used to represent respectively the quantities from the reservoir 1, 2 and 3.
The top plots represents the occupation probability for the two dot's. For both dot's the occupation probabilities are relatively constant which it's what it's expected from a model in a stationary regime. Moreover, as expected from the setup constrains $f_1 < f_2 < f_3$, $p_u(t) < p_d(t) \forall t$.
The oscillations are due to doing the ensemble average on non-equally spaced trajectory transitions.

The quantities plotted in the middle and bottom graphs are not ensemble averages, their "evolution" is calculated from a single trajectory. These quantities take a non-zero values only at the times of transitions. Because the transitions are not equally spaced, the probability of finding more than one transition occurring at the same time in two or more different trajectories is extremely low, hence the ensemble average drops to zero. To be able to plot this quantities, we use the empirical cumulative function by integrating them. Because the process is markovian, and hence the micro state at a certain time only depends on the time just prior, their empirical distribution functions approach their expected evolution for a big number of time-steps.
The middle plots represents the currents from the two reservoirs. As expected, the currents from the reservoirs 2 and 3 in contact with the dot d are at each instant opposite to each other and equal in modulus, and the current I_2 is the one that is negative. This is correct because the matter and heat flow should go from the reservoir 3 to the reservoir 2 as imposed by the setup.
Concerning the equilibrium regime, 3b, the quantities go relatively fast to zero as no flow should be present.

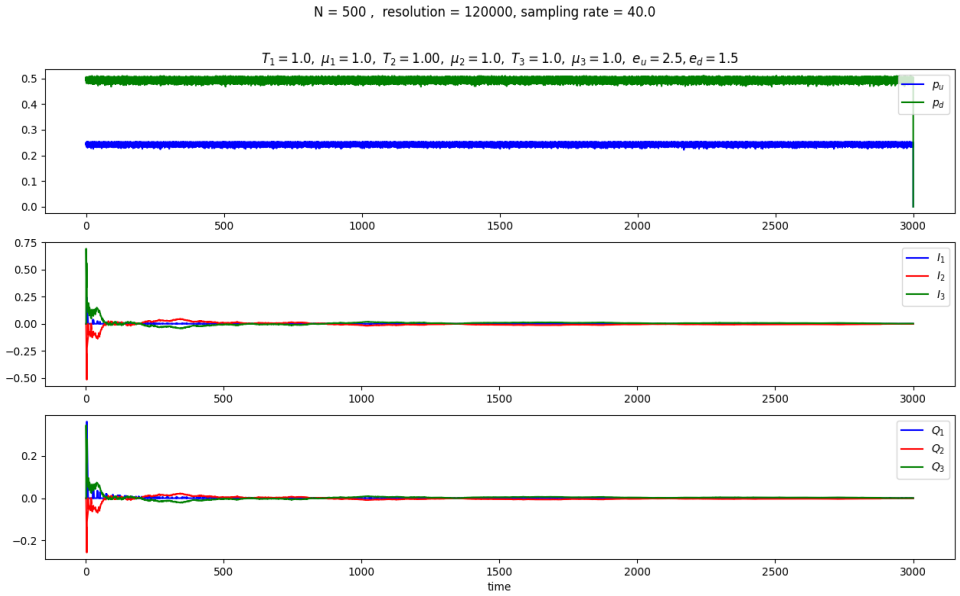
The bottom plots represent the heat rate from the three reservoirs. As expected they have the exact behavior as the currents.

5.2 Conclusions

The objective of this project was to study stationary regime models of more than one quantum dot in contact with more reservoirs and to show the possibility to find ordered behaviors even if in out of equilibrium, from which interesting proprieties may be extracted.



(a) out of equilibrium regime



(b) equilibrium regime

Figure 3: Stationary state - two single level quantum dots in contact with three reservoirs. The dot u in contact with the reservoir 1 and the dot d in contact with the reservoirs 2 and 3

6 Appendix

Full code for the simulation

```
import numpy as np
import matplotlib.pyplot as plt
import random
import sys

'''
    Reservoir Class
'''
class Reservoir:
    '''
        params: float:  T: reservoir's temperature
        params: float:  mu: reservoir's chemical potential
        params: float:  C: reservoir's-dot tunneling constant
        params: float:  e: energy of the dot in contact with the
                           reservoir
        return: Reservoir_Object
    '''
    def __init__(self, T,mu, C,e):
        self.T = T
        self.mu = mu
        self.C = C
        self.e = e
        self.up, self.down = self.rates(e)

    def fermi(self, e):
        return self.C /(1. + np.exp((e-self.mu)/self.T))

    def rates(self, e):
        return self.fermi(e), 1-self.fermi(e)


'''
    Calculates the miscrostate trajectories

    params: float:            MAXTIME:  maximum time of the
                                            simulation
    params  np_array:         times:    array of discretised time
    params: Reservoir_Object: res1:     reservoir 1
    params: Reservoir_Object: res2:     reservoir 2
    params: Reservoir_Object: res3:     reservoir 3
    reurn: tuple: (p_t, I, Q, jumps_t): occupation probability
                                          (2-D list),
                                         Currents (3-D list),
                                         Heats (3-D list),
                                         ransition times
                                          (MAXTIME-D list)
'''
def trajectory(MAXTIME, times, res1, res2, res3):
    t = 0
    p = [0,0]
    resolution = len(times)
    p_t = [[0,0] for _ in range(resolution)]
    jumps_t = [0]
    I_tau = [0,0,0]
```

```
    Q_tau = [0,0,0]
    I = {1 : [0], 2: [0], 3:[0]}
    Q = {1 : [0], 2: [0], 3:[0]}
    i = 0
    while i < resolution:
        pold = p
        while t <= times[i]:
            rate_uu = (1-p[0])* res1.up
            rate_ud = p[0] * res1.down
            rate_u = rate_uu + rate_ud
            rate_du = (1 - p[1]) * (res2.up + res3.up)
            rate_dd = p[1] * (res2.down + res3.down)
            rate_d = rate_du + rate_dd
            tau_u = - (1.0 / rate_u) * np.log(random.random())
            tau_d = - (1.0 / rate_d) * np.log(random.random())
            r = random.random()
            if tau_u < tau_d:
                t += tau_u
                if p[0] == 0:
                    I_tau = [1,0,0]
                    Q_tau = [res1.e - res1.mu, 0, 0]
                else:
                    I_tau = [-1,0,0]
                    Q_tau = [-res1.e + res1.mu, 0, 0]
                p[0] = 1- p[0]
            else:
                t += tau_d
                if p[1] == 0:
                    if r < res2.up / rate_du:
                        I_tau = [0,1,0]
                        Q_tau = [0, res2.e - res2.mu, 0]
                    else:
                        I_tau = [0,0,1]
                        Q_tau = [0, 0, res3.e - res3.mu]
                else:
                    if r < res3.down / rate_dd:
                        I_tau = [0,0,-1]
                        Q_tau = [0, 0, - res3.e + res3.mu]
                    else:
                        I_tau = [0,-1,0]
                        Q_tau = [0, -res2.e + res2.mu, 0]
                p[1] = 1 - p[1]
            jumps_t.append(t)
            for k in range(1,4):
                I[k].append(I_tau[k-1])
                Q[k].append(Q_tau[k-1])


        time_idx = np.searchsorted(times > t, True)
        corr_idx = min(resolution - 1, time_idx)
        for j in range(i, corr_idx - 1):
            p_t[j] = pold
        i = time_idx
    return p_t, I, Q, jumps_t


N = 500
```

```
MAXTIME = 3000
resolution =120000
e = 2.5
## Reservoirs
# Setup in such a way that
#res1.up < res2.up < res3.up <=> res1.down > res2.down > res3.down
res1 = Reservoir(T=0.9,mu=1.0,C=1.0,e=2.5)
res2 = Reservoir(T=1.0,mu=1.0,C=1.0, e=1.5)
res3 = Reservoir(T=5.2,mu=1.0,C=1.0, e=1.5)

print('res1.up = %s\nres1.down = %s' % (res1.up, res1.down))
print('\nres2.up = %s\nres2.down = %s' % (res2.up, res2.down))
print('\nres3.up = %s\nres3.down = %s' % (res3.up, res3.down))
if not res1.up < res2.up < res3.up and not res1.T == res2.T == res3
    .T:
    print('\nthe rates do not satisfy the conditions')
    sys.exit()


## Initiation of the variables
p_avg = [[0,0] for _ in range(resolution)]
jumped_t = []
times = np.linspace(0,MAXTIME, resolution)
sample_rate = resolution / MAXTIME
I = {1 : [0] , 2: [0], 3: [0]}
Q = {1 : [0] , 2: [0], 3: [0]}

dist_flag = False
for i in range(N):
    p_t, I_t, Q_t, jumps_t = trajectory(MAXTIME, times, res1, res2,
     res3)

    if dist_flag == False:
        jumped_t = jumps_t
        # Empirical comulative fucntion
        for i in range(1,len(I_t[1])):
            Q[1].append(sum(Q_t[1][:i]) / jumps_t[i])
            Q[2].append(sum(Q_t[2][:i]) / jumps_t[i])
            Q[3].append(sum(Q_t[3][:i]) / jumps_t[i])
            I[1].append(sum(I_t[1][:i]) / jumps_t[i])
            I[2].append(sum(I_t[2][:i]) / jumps_t[i])
            I[3].append(sum(I_t[3][:i]) / jumps_t[i])
        dist_flag = True
    ## MEAN VAlues
    for t in range(resolution):
        p_avg[t][0] += p_t[t][0] / N
        p_avg[t][1] += p_t[t][1] / N

p = {'u' : [item[0] for item in p_avg], 'd' : [item[1] for item in
    p_avg]}

#### PLOTS
fig, axs = plt.subplots(3)
plt.suptitle('N = %d ,  resolution = %.2d, sampling rate = %.1f' %
    (N, resolution, sample_rate))

```

```
164 axs[0].set_title(r"$T_1 = %.1f,\  \mu_1 = %.1f,\  T_2 = %.2f,\  \
        mu_2 = %.1f,\ T_3 = %.1f,\ \mu_3 = %.1f,\  e_u = %.1f, e_d =
        %.1f$" %(res1.T, res1.mu, res2.T, res2.mu, res3.T, res3.mu,
        res1.e, res2.e))
165 axs[0].plot(times, p['u'], label=r"$p_u$", color="blue")
166 axs[0].plot(times, p['d'], label=r'$p_d$', color="green")
167 axs[2].set_xlabel('time')
168 axs[0].legend(loc='upper right')
169
170 axs[1].plot(jumped_t[1:], I[1][1:], label=r"$I_{1}$", color="blue")
171 axs[1].plot(jumped_t[1:], I[2][1:], label=r"$I_{2}$", color="red")
172 axs[1].plot(jumped_t[1:], I[3][1:], label=r"$I_{3}$", color="green"
        )
173 axs[2].set_xlabel('time')
174 axs[1].legend(loc='upper right')
175
176 axs[2].plot(jumped_t[1:], Q[1][1:], label=r"$Q_{1}$", color="blue")
177 axs[2].plot(jumped_t[1:], Q[2][1:], label=r"$Q_{2}$", color="red")
178 axs[2].plot(jumped_t[1:], Q[3][1:], label=r"$Q_{3}$", color="green"
        )
179 axs[2].set_xlabel('time')
180 axs[2].legend(loc='upper right')
181
182 plt.show()
```

References

[1] Riccardo Rao and Massimiliano Esposito. "Conservation laws shape dissipation". In: *New Journal of Physics* 20.2 (2018), p. 023007.