Levy Flights

Lorenzo Maria Perrone[1, *]
1 Department of Physics, EPFL
*** E-mail: lorenzo.perrone@epfl.ch**

1 Exercise 2: Simulation of Lévy Flights

In this section we are going to analyse the behaviour of a Lévy flight $x(n)$. The conditional probability distribution of the increments $y_n \equiv x_{n+1} - x_n$ is given by:

$$P(y) = \frac{2b}{\pi(b^2 + y^{1+\mu})} \quad \mu \geq 0 \tag{1}$$

We start by considering the special case $\mu = 1$ (Cauchy distribution).

1.1 First order moment

It will be seen that for a Cauchy distribution, the first order moment $\langle x(n) \rangle$ doesn't converge for large n. We show this by averaging over many trajectories the positions $x_i = y_1 + y_2 + ... + y_{i-1}$ and plotting as a function of n. The result is in Fig. (1)

```
row = 1e3;column = 1e5;
G = zeros(row,column);
for kk=1:row
    Y = abs(cauc(b,column));
    for jj=2:column
        G(kk, jj) = Y(jj-1) + G(kk, jj-1);
    end
end
G_mean = mean(G,1);
s = @(x) x.*log(x);
t = [0:column-1];
figure;plot([0:column-1],G_mean)
title('Cauchy distribution <x(n)> n=1e5');xlabel('Step n');
ylabel('Position x');
```

As it is possible to see, the mean doesn't converge to a finite value.

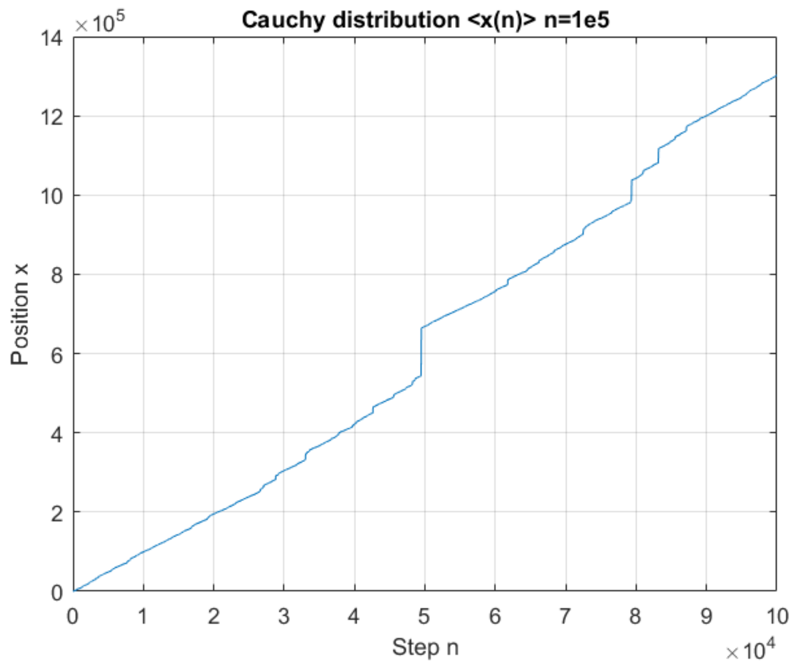

Figure 1. First order moment

1.2 Scaling behaviour

The cauchy distribution for the increment y presents with a scaling behaviour for the average $\langle x(n)\rangle \sim n\log(n)$. This has a correspondent interpretation in the anomalous diffusion. The scaling relation is shown in Fig. (2).

```
figure;plot([0:column-1],G_mean, 'DisplayName','<x(n)>');
hold on;plot(t,s(t), '-r', 'DisplayName','nlog(n)')
title('Cauchy distribution - Scaling behaviour <x(n)> n=1e5');
xlabel('Step n');
ylabel('Position x');legend('show');
```

1.3 Convergence to Levy law for large n

1.3.1 $\mu = 1$ - Cauchy process

Let's define the normalized variable $u_n \equiv \frac{x_n}{n^{1/\mu}} = \frac{x_n}{n}$. We want to show that $P(u_n)$ for large n has a limit defined by $L(u)$:

$$L(u) = \frac{b}{u^{3/2}} exp(-\pi b^2/u) \tag{2}$$

To do so, we consider M different realizations of a Cauchy process (M = 1000) and study the values x_N^m of them, with N = 1e5, $m = 1, ..., M$. Then, we divide x_N^m by N and plot the histogram of the data (see Fig. (3)). As we see, the agreement between $L(u)$ and the histogram of the data is pretty good, even though there seems to be a *shift* in the x-coordinate. This could be due to a non-perfect implementation of the Cauchy distribution in the following:

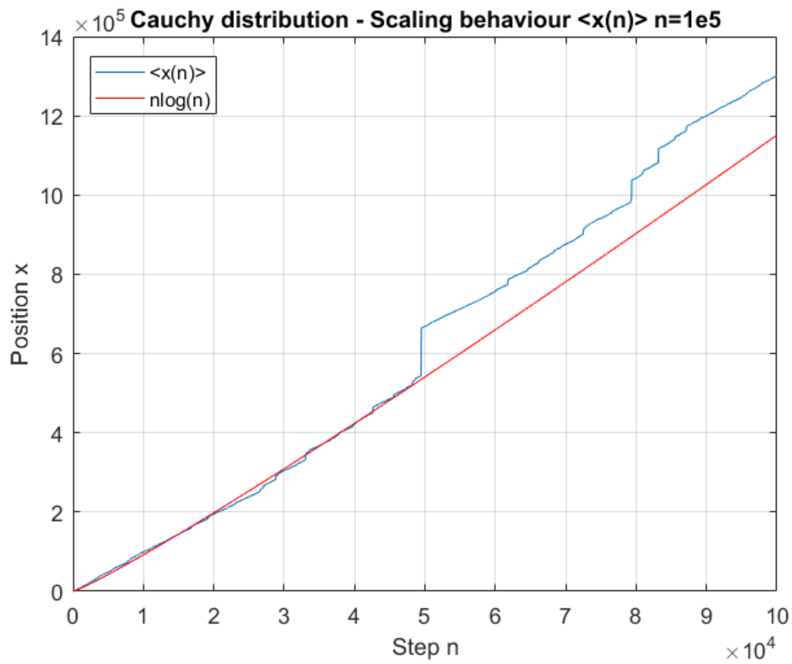

Figure 2. Scaling relation for a Cauchy process

```
function y = cauc(b,N)
    y = b*tan(0.5*pi*(rand(N,1)));
end

L = @(x) x.^(-3/2).*exp(-pi./x);
GG = G(:,end)/(column);
edges = [0 1:0.5:50 51:2:100];
[nn, xx] = hist(GG, edges);
figure;
histogram(GG,edges, 'Normalization','probability', 'DisplayName', 'hist(u)');
hold on; plot([0:0.5:xx(end)],L([0:0.5:xx(end)]), '-r', 'DisplayName', 'Levy Fu
title('Convergence to Levy Function for \mu = 1');xlabel('u_n = x_n / n');
ylabel('Probability P(u)');legend('show');
hold on
```

1.3.2 $\mu = 1/2$

To deal with this point, we used a "trick" that became clear after doing Exercise 3. It could be useful, then, to check the explanation given there.
By extracting random numbers from an exponential distribution, we can obtain a Levy probability distribution for any value of μ including 0.5. As we can see from Fig. (4), also in this case the agreement between the Levy function and the histogram is extremely good, and without shifts of any sort as well (1e4 trajectories have been computed for 1e4 steps).

Having read Exercise 3, the following code is self-explaining:

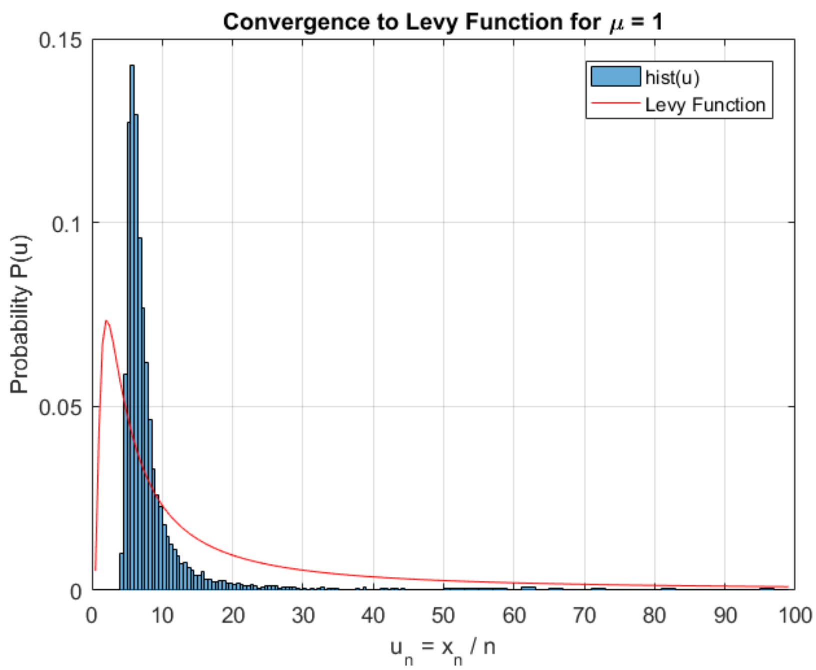

Figure 3. Convergence for mu = 1

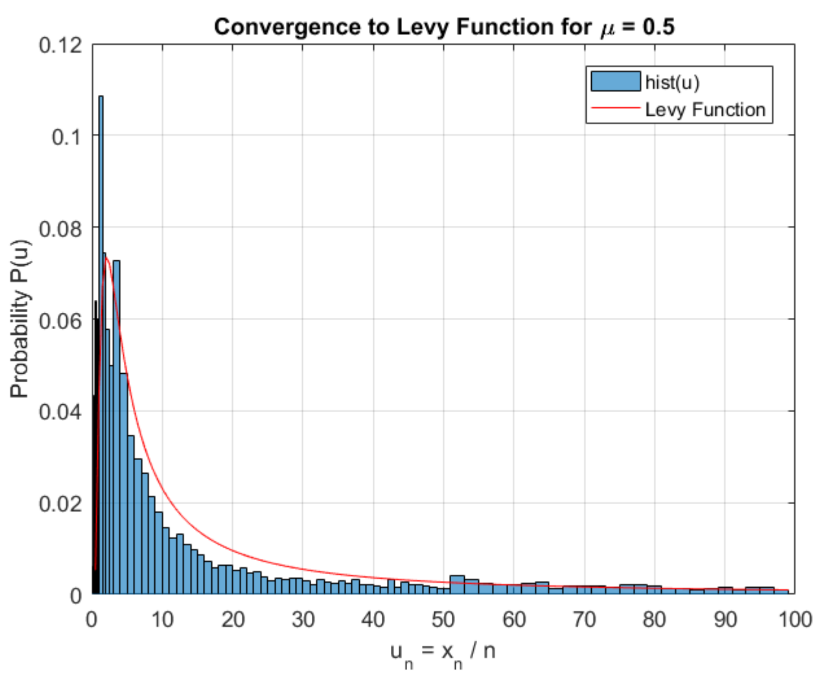

Figure 4. Convergence for mu = 0.5

```
pd=makedist('Exponential','mu',1);
row = 1e3;column = 1e5;
kT = 0.5;
b = 1;
G = zeros(row,column);
V = random(pd, row, column);tau=exp(V/kT);
for kk=1:row
    %Y = abs(cauc(b,column));
    for jj=2:column
        G(kk, jj) = tau(kk, jj-1) + G(kk, jj-1);
    end
end
G_mean = mean(G,1);
t = [0:column-1];
L = @(x) x.^(-3/2).*exp(-pi./x);
GG = G(:,end)/(column^2);
edges = [0:0.2:1 1:0.5:3 3:1:50 51:2:100];
[nn, xx] = hist(GG, edges);
figure;
histogram(GG,edges, 'Normalization','probability', 'DisplayName', 'hist(u)');
hold on; plot([0:0.5:xx(end)],L([0:0.5:xx(end)]), '-r', 'DisplayName', 'Levy Fu
title('Convergence to Levy Function for \mu = 0.5');xlabel('u_n = x_n / n');
ylabel('Probability P(u)');legend('show');
hold on
```

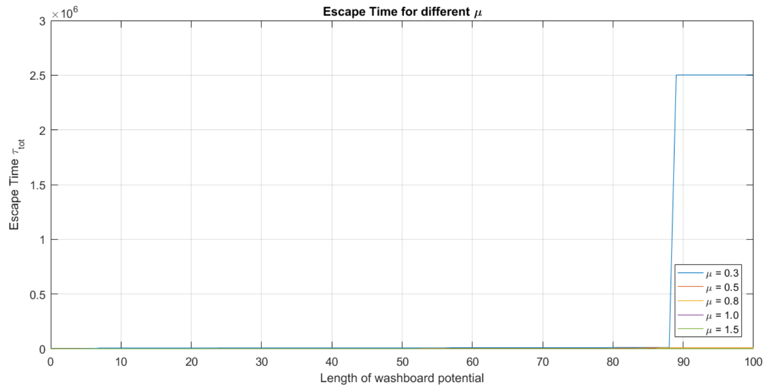

Figure 5. Comparison of the escape times for different values of mu

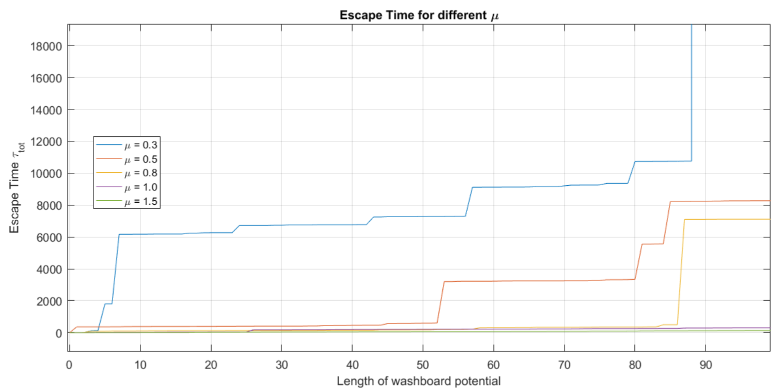

Figure 6. Comparison of the escape times for different values of mu - zoom

2 Arrhenius cascade

In this section we present the plots commented on the hand-typed paper. Additionally, we attach the code used in generating random numbers from a given distribution.

```
    mu = mu_array(ss);
p = @(x) (mu./(mu^2 + x.^(1+mu))).*heaviside(x);
norma = integral(p, 0, Inf);
p = @(x) ((mu./(mu^2 + x.^(1+mu))).*heaviside(x))/norma;
cum_int = @(y) integral(p,0,y);
N = 1e3;
Nrand = 100;
rn = rand(Nrand,1);
position = zeros(Nrand,1);
for jj=1:Nrand
    disp(jj)
    step = 0;
    a = -2;b = 7;
    counter = 0;
```

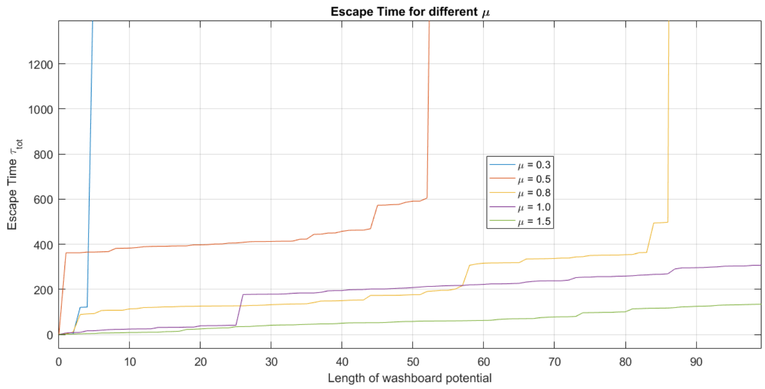

Figure 7. Comparison of the escape times for different values of mu - zoom

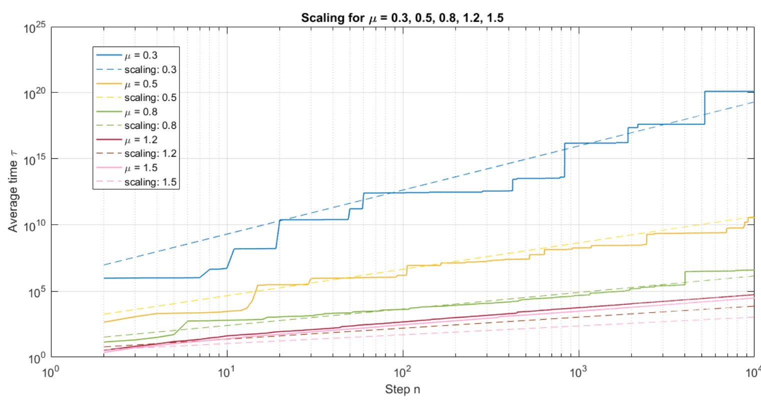

Figure 8. Scaling for different mu

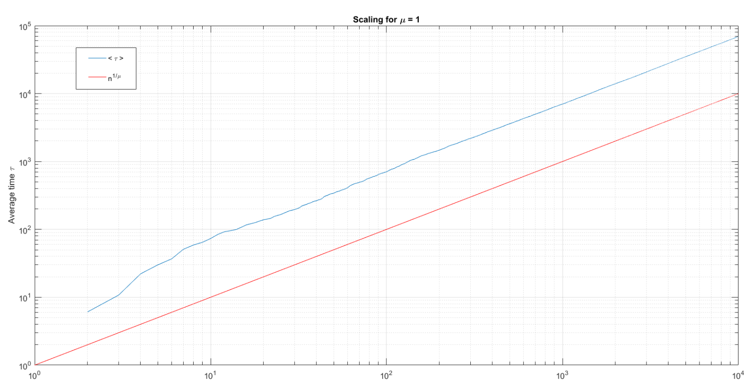

Figure 9. Scaling for mu = 1

```
    while step == 0 && counter < 100;
    %here we create a grid both on the x-axis
    %and on the correspondent image of the CDF
        x = logspace(a,b,N);
        I = zeros(N,1);
        for kk=1:N
            I(kk) = cum_int(x(kk));
        end
        %we want the last value of x which is smaller than CDF^-1(rand)
        id = find(I <rn(jj));
        %check if the extremal value of x is too small
        if ~isempty(id)
            id = id(end);
            %check if the minimal value of x is too large
            if ~(id == N)
                z1 = x(id);z2 = x(id+1);
                if (rn(jj) - I(id)) <= 1e-2
                    step = z1;
                else
                    a = log10(z1);b = log10(z2);
                        %t = logspace(log10(z1),log10(z2),1e5);
                end
            else
                z1 = x(end);z2 = x(end)*100;
                a = log10(z1);b = log10(z2);
            end
        else
            z1 = x(1)/100;z2 = x(1);
            a = log10(z1);b = log10(z2);
        end
         counter = counter + 1;
    end
    position(jj) = step;
end
for mm=2:Nrand
    position(mm) = position(mm)+position(mm-1);
end
position = [0 position']';
plot([0:Nrand], position, '-', 'DisplayName', ['\mu = ', sprintf('%1.1f',mu
hold on
```