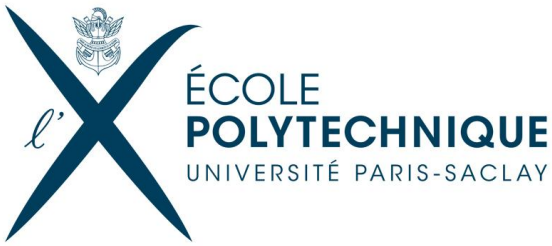


Optimal investment strategy in high-frequency trading

Bachir EL KHADIR, Mohamed Anas OUAALINE

Supervised by : Emmanuel BACRY, Iacopo MASTROMATTEO

Paris, 25/11/2014

Abstract

Given the state of an order book, we want to place purchase orders. Through this document, we try to define an efficient and optimal strategy in order to reduce the cost of such orders. Since it is quite complex to define this kind of strategies, we focus on the case where we have only one purchase order to make and a time horizon before which the trade must be executed. We have the choice between executing the order directly (market order), or placing a limit order. Since we are never sure that a limit order will be executed, we have to remove it after a certain time k and place a market order instead at the best offer available. Notice that the case where $k = 0$ is when a market order is executed immediately. The cost related to these kind of strategies can be of two types:

- Cost related to the spread when executing a market order.
- Cost related to the uncertainty of future price movements that we denote c.

While the first is easily measurable, the second is dependent on many factors: volatility of the market, the risk aversion of the trader... We will discuss this issue further in the following sections. The goal of this document becomes:

Given the state of an order book and a single purchase order we want to make, what it is the best choice of k to minimize the cost of the trade ?

Contents

1 The model and its dynamic **3**
1.1 The model . . . 3
1.2 The dynamic . . . 3

2 The optimization problem **4**

3 Solving the minimization problem **5**
3.1 First time of reach for a sum of a Markov chain . . . 5
3.2 Expectation of stopped $(S_k)_k$ . . . 6

4 Numerical Results **7**
4.1 Model results . . . 7
4.2 Results from market data and validation . . . 9

5 Model limitations and possible improvements **11**

A Technical proofs **12**
A.1 Recurrence equation of $(g^s_{-1})_s$ . . . 12
A.2 Taylor expansion of $(g^s_{-1})_s$ . . . 12

B The law of a sum of dependent random variables **13**

1 The model and its dynamic

Let us present the basic model which our study is based on.

1.1 The model

The order book is represented by its best ask and bid queues. The state of the limit order book is represented by :

- the bid price s_t^a and the ask price s_t^b
- the size of the bid queue q_t^a representing the outstanding limit buy orders at the bid, and
- the size of the ask queue q_t^b representing the outstanding limit sell orders at the ask

The bid and ask prices are multiples of the tick size $\delta = s_t^a - s_t^b$ which we consider constant. The state of the limit order book is thus described by the triplet $X_t = (s_t^b, q_t^b, q_t^a)$ which takes values in the discrete state space $\delta\mathbb{Z} \times \mathbb{N}^2$. With no loss of generality, we can assume that :

$$\delta = 1 \tag{1}$$

We assume that orders arriving at the best bid/ask follow a poissonian process and are all of the same size 1:

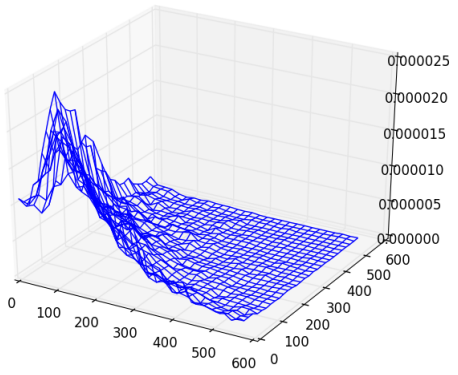
- A market/cancelation order arriving at the q_t^a (resp q_t^b) reduces its size by 1 if it is not empty
- A limit order arriving at the q_t^a (resp q_t^b) adds 1 to its size

Therefore q_t^a and q_t^b follow a birth/death process. We denote :

- $(\tau_i)_{i\geq 1}$ the sequence of random variables which stands for times between successive price changes. [1]
- $(X_i)_{i\geq 1}$ the successive moves in the price. Note that according to our hypothesis ($\delta = 1$) : $X_i \in \{-1, 1\}$.
- $(S_k)_{i\geq 0}$ the cumulative moves in the price after the n^{th} change, ie $S_n = \sum 1 \leq i \leq nX_i$. [2]

1.2 The dynamic

The state of the order book is modified by order book events. When the bid (resp. ask) queue is empty, the price moves down (resp. up), and (q_t^a, q_t^b) (resp.(q_t^b, q_t^a) is generated from a distribution $f(x, y)\mathrm{d}x\mathrm{d}y$ (resp. $\tilde{f}(x, y)\mathrm{d}x\mathrm{d}y$) independently from the rest of the historical variables. The next image is a graphical representation of f when estimated from historical data.



This property give the market a markovian property: the new state of the market after a price move does only depend on the direction of this move. Therefore, $(X_i)_{i\geq 0}$ is markov process such as :

- $\forall\, n \geq 0 \quad X_n \in \{-1, 1\}$
- $\forall\, i \geq 0 \quad \mathbb{P}(X_{i+1} = 1 | X_i = 1) = \mathbb{P}(X_{i+1} = -1 | X_i = -1) = p = 1 - q$

Given the symmetry of the market, we may assume that $f(x, y) = \tilde{f}(y, x)$. This assumption makes the process $(\tau_i)_i$ a sequence of IID random variables independent from $(X_i)_i$.

[1] Cont and Larrard's article presents an expression for the cumulative distribution function of each τ_i.
[2] see Annexe B for the generating function of S_n

2 The optimization problem

Recall how our strategy works:

- A limit order is placed at t=0 at the best bid.
- after some time $T = \sum_{i \leq k} \tau_i$, if the order is not executed, we cancel it and make a market order at the ask side.

k is the parameter of the strategy we seek to optimize. It stands for the maximum number of moves in price we are willing to wait for. Let's call it Strategy(k) . Let N be the first time where the sequence $(S_n)_{n \geq 0}$ achieves -1. Thereby :

$$N = \inf\{n \geq 0, S_n = -1\}$$

Finally, we denote c_{wait} the waiting cost per unit of time. The problem could be seen as a minimization problem :

$$\min_{k \in \mathbb{N}} \quad f(k) := \min_{k \in \mathbb{N}} \quad \mathbb{E}\left[\sum_{i=1}^{k \wedge N} \tau_i c_{wait} + (S_{k \wedge N)} + 1)\delta\right] \tag{2}$$

We shall notice that $(\tau_i)_{(i>0)}$ are assumed to be independent. The are also assumed to be independent of N. Thus we can rewrite f and the problem becomes :

$$\min_{k \in \mathbb{N}} \quad f(k) = \min_{k \in \mathbb{N}} \quad \mathbb{E}\left[(k \wedge N)\mathbb{E}(\tau)c_{wait} + (S_{k \wedge N} + 1)\right] \tag{3}$$

Since $\mathbb{E}(\tau)$ is a constant we can denote $c = \mathbb{E}(\tau)c_{wait}$ which is the waiting cost per price move. And the minimization problem finally becomes :

$$\min_{k \in \mathbb{N}} \quad f(k) = \min_{k \in \mathbb{N}} \quad c\mathbb{E}\left[k \wedge N\right] + \mathbb{E}\left[S_{k \wedge N}\right] + 1 \tag{4}$$

In order to solve this problem we need to do 2 different calculations :

- $\mathbb{E}\left[k \wedge N\right]$: we are going to explicitly calculate the Law N
- $\mathbb{E}\left[S_{k \wedge N}\right]$: since S_i and N are not independent, the law is quite hard to be explicitly calculated. Consequently, we are going to use a Monte-Carlo Algorithm.

The parameter c:
In the previous section we have denoted :

$$c = \mathbb{E}(\tau)c_{wait}$$

It is an exogenous parameter which has to manually fixed. We notice that through its definition c depends on $\mathbb{E}(\tau)$ which is the average time between two moves in the price. However we can be more accurate and see that it also depends on the volatility of the market and the operator's risk aversion. We also notice that c needs to be lower than the thick δ otherwise it is not interesting to wait for a move in the price. Thus we came up with the following remarks.

- $c = \mathbb{E}(\tau)F(\sigma, v, r)$ where $\mathbb{E}(\tau)$ is the average time between two moves in the price, σ is the market volatility, v is the operator's risk aversion and r is some unknown parameters.
- $c < \delta$ where δ is the thick.
- c is increasing in σ : The more the market is volatile the more it risky is to wait.
- c is increasing in v : The more the operator is averse to risk the more likely he is to over-rate the waiting cost.

In our model we assume that $\mathbb{E}(\tau), \sigma, v$ and r do not depends on k.

3 Solving the minimization problem

3.1 First time of reach for a sum of a Markov chain

In this section we address the issue of finding the law of the variable N. Let us define the probability-generating function of N such as :

$$g_s^x(z) = \mathbb{E}(z^N | S_0 = s, X_0 = x) \quad \forall\ x \in \{-1, 1\}\ ,\ s \in \mathbb{N}\ ,\ z \in [0, 1]$$

so that

$$\mathbb{E}(z^N | S_0 = s) = \alpha g_s^1(z) + (1 - \alpha) g_s^{-1}$$

where $\alpha := \mathbb{P}(X_0 = -1)$. Given that (S_n, X_n) is a markov chain, we can infer that for all $x \in \{-1, 1\}$, $s \in \mathbb{N}$ and $z \in]0, 1[$ [3] :

$$\begin{cases} g_s^1 = z(p g_{s+1}^1 + q g_{s-1}^{-1}) & (5) \\ g_s^{-1} = z(q g_{s+1}^1 + p g_{s-1}^{-1}) & (6) \end{cases}$$

Denoting :

$$u = \frac{1}{2}\left(\frac{1}{zp} + z(2 - \frac{1}{p})\right) \text{ [4]} \quad (7)$$

and by imposing a Dirichlet boundary condition, we can prove [5] that g_s^1 dynamic is given by the following second order system :

$$\begin{cases} g_{s+1}^{-1} - 2u g_s^{-1} + g_{s-1}^{-1} = 0 & (8) \\ g_{-1}^{-1} = 1 & (9) \\ \lim\limits_{s \to +\infty} g_s^{-1} = 0 & (10) \end{cases}$$

Which is a classical linear recurrent sequence of order 2. The solution is given by :

$$g_s^{-1} = \left(u - \sqrt{u^2 - 1}\right)^{s+1} \quad (11)$$

Changing 1 to -1 in (5) and (6) we see that g_s^1 follow the same dynamic as g_s^{-1} if we change p to $1 - p$, therefore we can deduce all the properties of the first from the second. Since g_s^x is the probability-generating function of N , the law of N conditional to $S_0 = 0$ and $X_0 = x$ is given by :

$$\begin{cases} g_s^{-1}(z) = \sum\limits_{j=0}^{+\infty} \mathbb{P}_{X_0=-1}(N = j | S_0 = s) z^j & (12) \\ g_s^1(z) = \sum\limits_{j=0}^{+\infty} \mathbb{P}_{X_0=1}(N = j | S_0 = s) z^j & (13) \end{cases}$$

$\mathbb{P}_{X_0=-1}(N = j | S_0 = s)$ can be calculated using Taylor expansion from the analytical expression of $g_s^{-1}(z)$ we have just found, and $\mathbb{P}_{X_0=1}(N = j | S_0 = s)$ can be deduced as discussed above.

[3] For clarity, we omit to note the dependence of g in z.
[4] u ≥ 1
[5] see Annexe

3.2 Expectation of stopped $(S_k)_k$

$E[S_{k\wedge N}]$ is calculated using Monte Carlo algorithm. [6]

```cpp
#include <iostream>
#include <iomanip>
#include <string>
#include <map>
#include <random>

using namespace std;

int main()
{
  int k = 5;
  double p = 0.2;

  // Number of independant simulations
  int M = 1e6;

  // generator of X_i
  random_device rd;
  mt19937 gen(rd());
  map<int, bernoulli_distribution> alea;
  // generator of X_i conditional to  X_{i-1} = 1
  alea[1] = std::bernoulli_distribution(p);
  // generator of X_i conditional to  X_{i-1} = -1
  alea[-1] = std::bernoulli_distribution(1 - p);

  double mean = 0;
  for (int m = 0; m < M; m++) {
    int S = 0;
    int X0 = 1;
    for (int i = 0; i < k && S >= 0; i++) {
      X0 = 2*alea[X0](gen)-1;
      S += X0;
    }
    mean += S;
  }

  mean /= M;
  cout << mean << endl;

return 0;
}
```

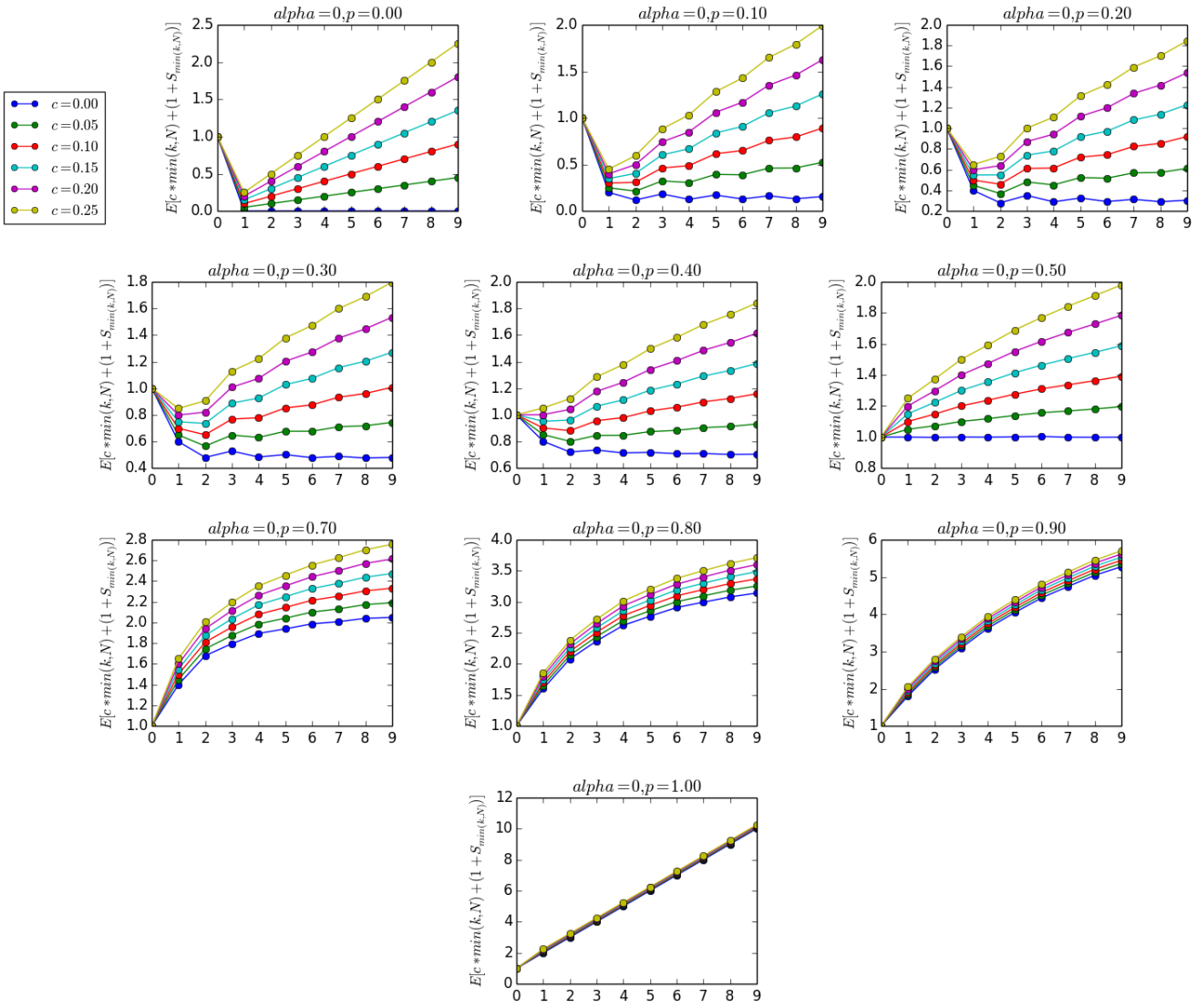
[6] All the codes used in this project are available at the link: https://github.com/maroxe/SchoolProjects/tree/master/EA

4 Numerical Results

4.1 Model results

Here are the results for different combination of the parameters (p, c, α).

When $X_0 = 1$, ie $\alpha = 0$

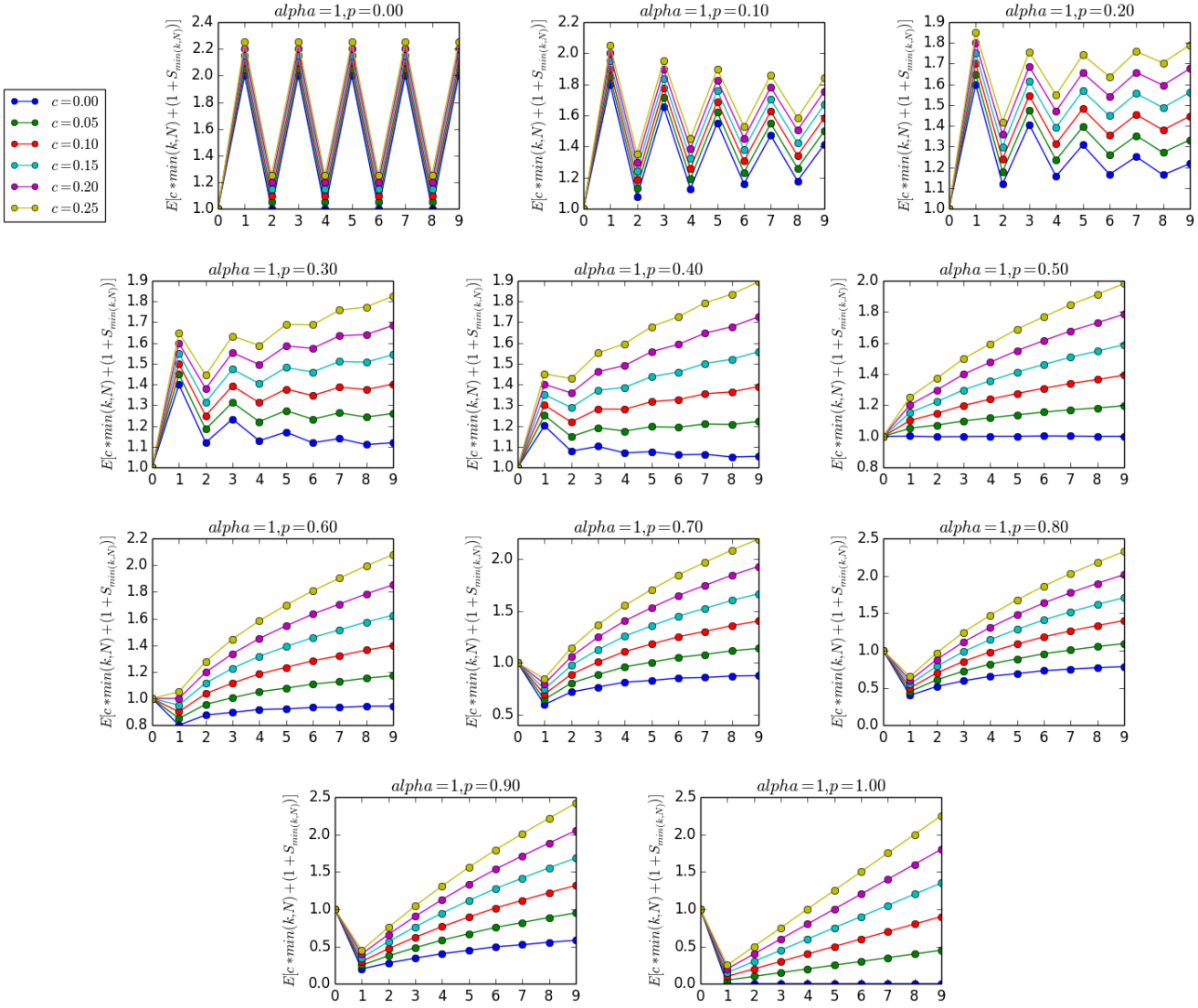


It is interesting to see that the behavior changes depending on wether $p < 1/2$ or not. Indeed, since $X_0 = -1$, for low values of p, there is a big chance that the trend changes:

$$p = \mathbb{P}(X_1 = 1 | X_0 = 1)$$

and therefore it is more profitable to buy directly at the opposite side. The critical case when $p = 0.5$ is interesting since it is the case where the $(X_i)_i$ are independent. The random walk $(S_{N \wedge k})_k$ is a martingale whose expectation is constant. When $c = 0$, all strategies are equivalent, and as soon as $c > 0$, the additional cost of waiting makes it better to place a market order.

When $X_0 = -1$, ie $\alpha = 1$



We see the exact opposite effect of p on the graphs when $X_0 = -1$, ie it is interesting to consider placing an order limit only when X_0 and X_1 are positively correlated ($p > \frac{1}{2}$).

4.2 Results from market data and validation

All experiments are done using the historical data of the Bund. Let's first try to estimate some parameters of the model.

$$p = \mathbb{P}(X_{i+1} = X_i) \approx \frac{\sum_{i \leq M} 1_{\{X_{i+1}=X_i\}}}{M}$$

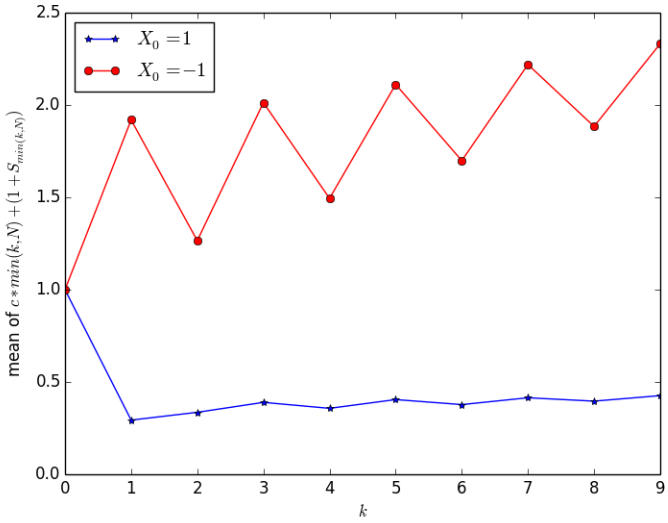
The following progam gives $p \approx 0.2$

```
r=pickle.load(input)
X = r['BidPrice']
X = X[1:] - X[:-1]
X = X[ abs(X) > 1e-3] # Keep only the entries where the price changes
X = np.array(map(lambda x: 1 if x > 0 else -1, X))
p = np.count_nonzero(X[1:] == X[:-1]) / len(X)
```

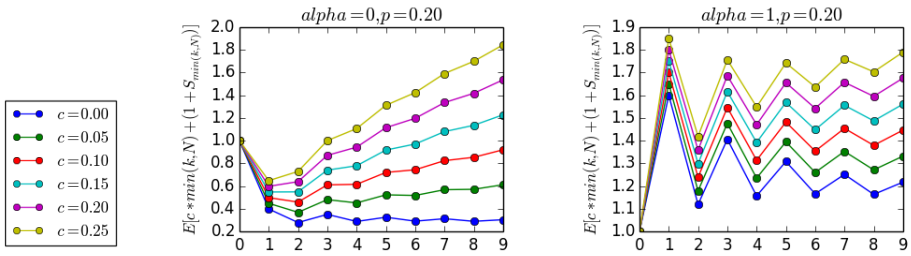
The same method gives

$$\mathbb{E}[\tau] \approx 3.75ms$$

Here is the mean cost when applying Strategy(k) for $k \in \{0, ..., 9\}$ periods using $c = 0.05$:



The shape of the graph is very similar to the ones we found in the previous section corresponding to $p = 0.2$



And here is the python program responsible for this calculation:

```
K = 10
c = 0.05
# Y_1[k] Stores the mean cost when applying strategy(k) | X0 = 1
# Y__1[k] Stores the mean cost when applying strategy(k) | X0 = -1
Y_1, Y__1 = [], []
```

```
for k in range(K):
        # count[1] stores the cost of strategy(k) when X0=1
        # the same for count[-1]
        cout = {1: [], -1: [] }
        for i, X0 in enumerate(X):
                S = 0
                j = 1
                try:
                        # strategy(k) continues while the number of changes < k and S has not reached -1
                        while j <= k and S >= 0:
                                S += X[i + j]
                                j += 1
                        cout[X0].append(  (min(j, k), S+1) )
                except IndexError:
                        # End of data, the strategy is interrupted
                        continue
        Y_1.append(np.mean(map(lambda (N, S): N*c+S, cout[1])))
        Y__1.append(np.mean(map(lambda (N, S): N*c+S, cout[-1])))
        # Draw Y_1 and Y__1
```

5 Model limitations and possible improvements

Our model presents three main limitations which can be summarized by the following :

- $\mathbb{E}\left(S_{k\wedge N}\right)$ has been computed through a Monte Carlo Algorithm. While we were able to compute efficiently the law of S_i [7], we could not find a simple analytical expression. Indeed, S_i and N are dependent and the calculus becomes quickly hard to do.
- Our model is markovian in the sense that we suppose that each move in price depends only on the previous. This choice has been made to reduce the complexity of the calculus in particular for the law of N. This choice is also induced by a previous hypothesis that says that the state of the order book at the best bid and best ask level is randomly and independently drawn whenever the price moves. To improve the model at this point, we could consider other levels of the bid and ask queues. A more comprehensive, and thus more complex model, would take in consideration the effect of autocorrelation previous trades on the intensity of arriaval of orders. Hawkes processes can be considered for this.
- In our strategy, the maximum number of moves in price we are willing to wait for is bounded by 10. Whereas it could be interesting to wait more. We have made this arbitrary choice just to make a proof of concept. We could go much further with values of k above 100. The calculus remains quickly feasible.

[7] see Annexe B

A Technical proofs

A.1 Recurrence equation of $(g_{-1}^s)_s$

The purpose of this section if to find the equation verified by $(g_{-1}^s)_s$. Equations (5)-(6) implies that:

$$\begin{aligned} g_{s+1}^1 &= \frac{1}{zp} g_s^1 - \frac{q}{p} g_{s-1}^{-1} \\ &= \frac{1}{zq} g_s^{-1} - \frac{p}{q} g_{s-1}^{-1} \end{aligned}$$

Thereby :$g_s^{-1} = \frac{q}{p} g_s^1 + \frac{z(p^2-q^2)}{p} g_{s-1}^{-1}$. Leading finally to: $g_{s+1}^{-1} = (\frac{1}{zp} + z(2-\frac{1}{p}))g_s^{-1} - g_{s-1}^{-1}$.

A.2 Taylor expansion of $(g_{-1}^s)_s$

Recall that:$g_{s+1}^{-1} = (\frac{1}{zp} + z(2-\frac{1}{p}))g_s^{-1} - g_{s-1}^{-1}$. Let us do the following denotations :

$$\begin{cases} u = \frac{1}{2}(\frac{1}{zp} + z(2-\frac{1}{p}) = az + \frac{b}{z} & (14) \\ a = 1 - \frac{1}{2p}, b = \frac{1}{2p} & (15) \end{cases}$$

Using a generalized Newton's formula we get :

$$\begin{aligned} (u^2-1)^{\frac{1}{2}} &= \sum_i \binom{\frac{1}{2}}{i} (-1)^i z^{2i-1} (b+az^2)^{1-2i} \\ &= \sum_i \binom{\frac{1}{2}}{i} (-1)^i z^{2i-1} \sum_j \binom{1-2i}{j} b^{1-2i-j} a^j z^{2j} \\ &= \sum_{i,j} \binom{\frac{1}{2}}{i} \binom{1-2i}{j} (-1)^i b^{1-2i-j} a^j z^{2(i+j)-1} \\ &= \sum_k \left(\sum_{2(i+j)-1=k} \binom{\frac{1}{2}}{i} \binom{1-2i}{j} (-1)^i b^{1-2i-j} a^j \right) z^k \\ &= \sum_r \left(\sum_{i+j=r} \binom{\frac{1}{2}}{i} \binom{1-2i}{j} (-1)^i b^{1-2i-j} a^j \right) z^{2r-1} \\ &= \sum_r \left(\sum_{i+j=r} \binom{\frac{1}{2}}{i} \binom{1-2i}{j} (-1)^i b^{1-2i-j} (1-b)^j \right) z^{2r-1} \\ &= \sum_r c_{2r-1} z^{2r-1} \end{aligned}$$

On the one hand, for even k :

$$c_k = 0$$

On the other hand, for odd $k \geq -1$ we get :

$$c_k = \sum_{2(i+j)=k+1} \binom{\frac{1}{2}}{i} \binom{1-2i}{j} (-1)^i b^{1-2i-j} (1-b)^j$$

In particular:

$$\begin{cases} c_{-1} = \binom{\frac{1}{2}}{0} \binom{1}{0} (-1)^0 b^1 (1-b)^0 = b & (16) \\ c_1 = \sum_{i+j=1} \binom{\frac{1}{2}}{i} \binom{1-2i}{j} (-1)^i b^{1-2i-j} (1-b)^j = a - p & (17) \end{cases}$$

Thereby

$$\begin{aligned} g_1^0(z) &= u - (u^2-1)^{\frac{1}{2}} \\ &= (a-c_1)z - (b+c_{-1})z^{-1} - \sum_{k>1} c_k z^k \\ &= \sum_{k\geq 1} d_k z^k \end{aligned}$$

Where:

$$d_{2r+1} = \begin{cases} p & \text{if} \quad r=0 \\ -\sum_{i\leq r+1} \binom{\frac{1}{2}}{i}\binom{1-2i}{1+r-i}(-1)^i(2p)^{2i-1}(2p-1)^{1+r-i} & \text{else} \end{cases}$$

and

$$d_2 r = 0$$

B The law of a sum of dependent random variables

We define a Markov Chain $(X_n)_{n\geq 0} \in \{-1,1\}^{\mathbb{N}}$ whose transition probabilities are defined by

$$\mathbb{P}(X_{i+1}=1|X_i=1) = \mathbb{P}(Xi+1=-1|X_i=-1) = p = 1-q$$

Its sum is denoted $(S_n)_{n\geq 0}$. In order to get the law of the r.v $(S_n)_{n\geq 0}$ we define its conditional probability-generating function as :

$$f_n^x(z) = \mathbb{E}(z^{S_n}|X_0=x) \quad \forall\ x\in\{-1,1\}\ ,\ n\in\mathbb{N}\ ,\ z\in[0,1]$$

Let us consider $x\in\{-1,1\}$, $n\in\mathbb{N}$, $z\in]0,1]$[8]. Thanks to the fact that (S_n,X_n) is a markov chain, we can write :

$$f_{n+1}^x(z) = z\mathbb{E}(z^{S_n}|X_0=1)\mathbb{P}(X_1=1|X_0=x) + \frac{1}{z}\mathbb{E}(z^{S_n}|X_0=-1)\mathbb{P}(X_1=-1|X_0=x) \tag{18}$$

Leading thus to the following system with the appropriate initial conditions :

$$\begin{cases} f_{n+1}^1(z) = zpf_n^1(z) + \frac{1}{z}qf_n^{-1}(z) & (19) \\ f_{n+1}^{-1}(z) = zqf_n^1(z) + \frac{1}{z}pf_n^{-1}(z) & (20) \\ f_0^1(z) = f_0^{-1}(z) = 1 & (21) \\ f_1^1(z) = zp + \frac{1}{z}q & (22) \\ f_1^{-1}(z) = zq + \frac{1}{z}p & (23) \end{cases}$$

We then prove that f_n^1 and f_n^{-1} follow the same equation, ie for $x\in\{-1,1\}$

$$f_{n+2}^x - p(\frac{1}{z}+z)f_{n+1}^x + (2p-1)f_n^x = 0 \tag{24}$$

Which is a classical linear recurrent sequence of order 2. The solution is given by :

$$f_n^x(z) = \frac{f_1^x - v + \sqrt{u^2-(2p-1)}}{2\sqrt{v^2-(2p-1)}}\left((v+\sqrt{v^2-(2p-1)})^n - (v-\sqrt{v^2-(2p-1)})^n)\right) + (v-\sqrt{v^2-(2p-1)})^n \tag{25}$$

Where :

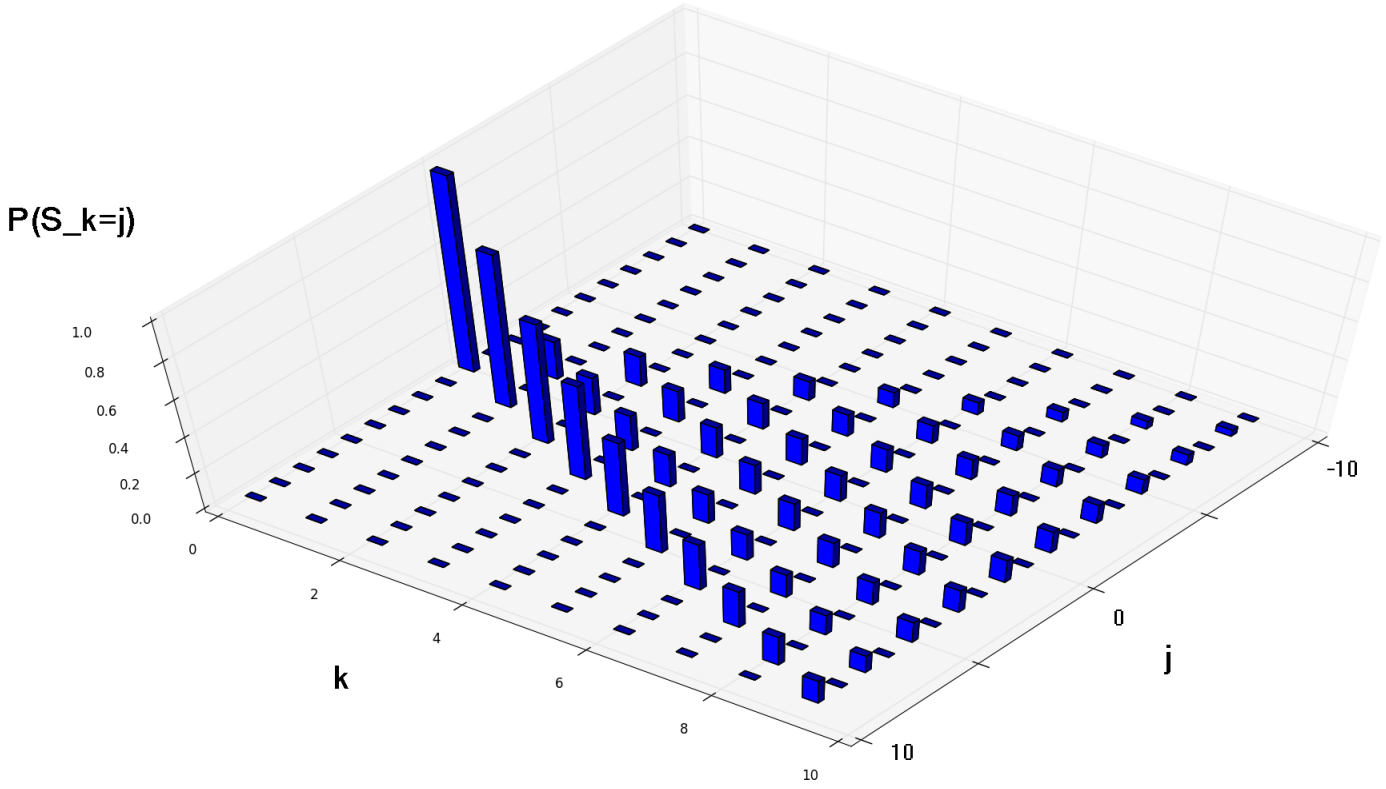
$$v = \frac{p}{2}(z+\frac{1}{z})$$

[8]The case where z=0 can be obtained by continuity.

f_n^x is the probability-generating function of $(S_n)_{n\geq 0}$. We shall notice that $\forall n \in \mathbb{N} - n \leq S_n \leq n$. We then get :

$$\begin{cases} \mathbb{E}_{X_0=x}(S_n) = \dfrac{\mathrm{df_n^x}}{\mathrm{d}z}(z=1) & (26) \\ f_n^x(z) = \mathbb{E}(z^{S_n}|X_0 = x) = \displaystyle\sum_{-\infty}^{+\infty} \mathbb{P}(S_n = k)z^k = \sum_{-n}^{+n} \mathbb{P}(S_n = k)z^k & (27) \end{cases}$$

We use Maple to compute the Taylor expansion, we get for example for $X_0 = -1$ and $p = 0.2$:



We can see that $\mathbb{P}(S_0 = 0) = 1$ and $\mathbb{P}(S_1 = 1) = 1 - \mathbb{P}(S_1 = -1) = 0.2$

References

- Rama Cont, Adrien de Larrard (2010), Price dynamics in a Markovian limit order market
- Rama Cont, Sasha Stoikov, Rishi Talreja, A stochastic model for order book dynamics
- J. Doyne Farmer, Laszlo Gillemot, Giulia Iori, Supriya Krishnamurthy, D. Eric Smith, Marcus G. Daniels, A Random Order Placement Model of Price Formation in the Continuous Double Auction
- Weibing Huang,, Charles-Albert Lehalle, and Mathieu Rosenbaum (2014), Simulating and analyzing order book data: The queue-reactive model