“Documentation” for the algorithm “optimal execution by Forsyth”

Arnaud LIONNET

03/03/2020,   16:50

Contents

Premia 22

1 The problem being solved and what the function does.

1.1 Optimal trade execution : notation, model, objective.

This function/algorithm is the for the numerical solving of the problem of optimal execution of an operation by an agent using the mean-square objective criterion.

1.1.1 Accountancy.

Let B_t be the *amount* of money on the risk-free bank account and α_t be the *number* of shares of the risky asset held, the price of a share being S_t, at time t.

The agent starts with an amount $\alpha_0 = \alpha_I$ shares that they have to sell (if $\alpha_I > 0$) or buy (if $\alpha_I < 0$) so as to have $\alpha_T = \alpha_F = 0$ at the deterministic terminal time T. By default we will use the language of selling shares, although all quantities are algebraic.

The bank account starts at $B_0 = 0$ and at the end is given by the random variable $B_L = L(S_T, B_T, \alpha_T)$. The reason this is not simply B_T is because we do not force the agent to use a trading strategy such that $\alpha_T = \alpha_{T^-} = 0$, i.e. everything has been sold in time. Rather, should α_T be non-zero, the position is fire-liquidated over a small time interval $\delta t << T$, so that $\alpha_{T+} = \alpha_F = 0$. The liquidation function L, which will be made explicit below, is naturally such that $L(S_T, B_T, 0) = B_T$.

1.1.2 Control, price model and price impact.

The agent's control is the trading rate $(v_t)_{0 \leq t \leq T}$. As the actions will depend on the states of the world, this process is stochastic. The dynamics of the portfolio in the risk asset is therefore given by

$$\frac{d\alpha_t^v}{dt} = v_t \qquad \Longleftrightarrow \qquad \alpha_t^v = \alpha_I + \int_0^t v_s ds.$$

The dynamics of the price of the asset is a geometric Brownian motion affected by a permanent price impact,

$$dS_t = S_t(\eta + g(v_t))dt + S_t \sigma dW_t.$$

Here, η and σ are the drift and volatility the price would have without the trading of the agent. The trading permanently affects the drift, with linear price impact function $g(v) = \kappa_p v$.

The dynamics of the bank account is given by

$$\frac{dB_t^v}{dt} = rB_t^v - v_t f(v_t) S_t.$$

Here, r is the risk-free, instantaneous interest rate, which makes the bank account grow (case of selling). The selling of $-v_t dt$ shares over $[t, t+dt]$ at price $f(v_t)S_t$ generates an extra income (note that $v_t \leq 0$ when selling). The "transaction cost and temporary price impact" function is taken to be

$$f(v) = (1 + \kappa_s \mathrm{sgn}(v)) \ \exp\left(\mathrm{sgn}(v)\kappa_t |v|^\beta\right).$$

Here, κ_s is morally half the bid-ask spread normalized by the mid-price, $\frac{S_t^a - S_t^b}{2S_t}$, and is therefore in $[0,1[$. It is assumed to be constant, and can generally be thought of as a transaction cost. The term $\exp\left(\operatorname{sgn}(v)\kappa_t |v|^\beta\right)$ is the temporary price impact, with κ_t and β being constants.

The liquidation function is given by

$$L(S,B,\alpha) = B - \left(\frac{-\alpha}{\delta t}\right) f\left(\frac{-\alpha}{\delta t}\right) S.$$

That is, $\frac{-\alpha}{\delta t}$ is the final trading rate (between T and T^+) and the buying of the remaining $\alpha_F - \alpha = -\alpha$ shares generates a last income (if $\alpha > \alpha_F$, i.e. there are still shares to sell).

1.1.3 The objective function.

The agent is a mean-variance optimizer. Given a target mean value d for the final bank account, the agent seeks to minimize the variance. We denote by $B_L^v = L(S_T, B_T^v, \alpha_T^v)$ the final amount on the bank account. So, for the given d, the optimization problem is

$$\begin{gathered} \min_v \operatorname{Var}\left[L(S_T, B_T^v, \alpha_T^v)\right] = \min_v \left\{E\left[(B_L^v)^2\right] - d^2\right\} \\ \text{under the constraints } E\left[B_L^v\right] = d \text{ and } v \text{ valued in } Z. \end{gathered} \tag{1}$$

Here, $Z = [v_{\min}, v_{\max}]$ is the set of admissible trading rates, and we will write $v \in Z$ to mean that the control process v must be valued in Z at all times. Typically, we normalize the initial number of shares and only selling/buying is allowed depending on the case. For selling, we would have $\alpha_I = 1$, $v_{\min} < 0$ and $v_{\max} = 0$. For buying, we would have $\alpha = -1$, $v_{\min} = 0$ and $v_{\max} > 0$. But the formalism can of course accommodate for other situations.

1.2 Determination of the optimal frontier vs strategy and value for a given expectation

1.2.1 The efficient frontier.

The real goal here is not so much to solve the problem (1) for a given d, but rather to draw the efficient frontier. In the (SD,E) space (SD in abscissa and E in ordinate), for each expectation d, there is a minimal standard deviation Σ_d : all points with higher Σ are not optimal and not "realized", points with lower Σ are not accessible. The curve $d \mapsto (\Sigma_d, d)$ is the efficient frontier. This is what the function returns.

Remark 1.1 (On the duality with maximization of the mean, etc.)**.** *TBDL.*

1.2.2 Lagrange multiplier.

In order to solve the problem (1), one can use a Lagrange multiplier. That is, for fixed d, we consider the problem

$$\operatorname*{opt}_{\gamma\in\mathbb{R}} \min_{v\in Z} \operatorname{Lag}(v,\gamma) = E\big[(B_L^v)^2\big] - d^2 - \gamma\big(E[B_L^v] - d\big). \tag{2}$$

The solution is at a critical point for Lag. For fixed γ, the problem of finding v such that $\frac{\partial \mathrm{Lag}}{\partial v} = 0$, $\leftrightarrow \min_{v\in Z} \operatorname{Lag}(v,\gamma)$, is easier because unconstrained. For fixed v, $\frac{\partial \mathrm{Lag}}{\partial \gamma} = 0 \Leftrightarrow$ the constraint is satisfied.

In practice, and for fixed d, the solution to problem (2) is often sought as follows. First, for fixed γ, solve the minimization problem over v :

$$\min_{v\in Z} \operatorname{Lag}(v,\gamma) = E\big[(B_L^v)^2\big] - d^2 - \gamma\big(E[B_L^v] - d\big).$$

This leads to an optimal control $v^*(\gamma)$, with an associated variance $\mathrm{Var}^*(\gamma) = \mathrm{Var}[B_L^{v^*(\gamma)}]$ and expectation $\mathrm{Exp}^*(\gamma) = E[B_L^{v^*(\gamma)}] =: d^*(\gamma)$. Then, find the γ^* such that the constraint $E[B_L^{v^*(\gamma^*)}] = d^*(\gamma^*) = d$ is satisfied.

Let us look more closely at that minimization problem over v. Notice that

$$\operatorname{Lag}(v,\gamma) = E\Big[\Big(B_L^v - \frac{\gamma}{2}\Big)^2\Big] - \frac{\gamma^2}{4} - d^2 + \gamma d. \tag{3}$$

So we have the same minimizers v^* if we solve

$$\min_{v\in Z} E\Big[\Big(B_L^v - \frac{\gamma}{2}\Big)^2\Big]. \tag{4}$$

Interestingly, this problem (4) does not depend on d. By solving it, we obtain a control $v^*(\gamma)$, the value $V(\gamma) = E\big[(B_L^{v^*(\gamma)} - \frac{\gamma}{2})^2\big]$, and we can compute $d^*(\gamma) = E[B_L^{v^*(\gamma)}]$. We can then obtain the original value $\mathrm{Var}^*(\gamma)$ for $\gamma = (d^*)^{-1}(d)$.

Now, hopefully enough, the function d^* (which does not depend on d) is continuous and strictly monotonic. Or at least a bijection. So when d spans its domain, γ spans its domain. Consequently, rather than drawing $d \mapsto \big(\mathrm{Var}^*\big((d^*)^{-1}(d)\big)^{1/2}, d\big)$, we will draw the curve $\gamma \mapsto \big(\mathrm{Var}^*(\gamma)^{1/2}, d^*(\gamma)\big)$. For this, we only need to solve the problem (4) as γ varies.

1.3 HJB approach.

1.3.1 Dependence on B_t and other state variables.

The optimal trading rate v^* at time t surely depends on the current price S_t and the number of shares α_t remaining to liquidate. It does not seem clear that it should depend on B_t. Indeed, whatever has been amassed so far will have a deterministic and therefore non-risky growth until T, and whatever will be gained and at what risk from selling between t and T depends on the present and future actions. The value V of the problem (4) can also be considered when starting from any time $t \in [0,T]$, with current price S_t and number of shares α_t.

However, the problem (4) is a target problem for B : we seek to minimize the L^2-distance between B_L and $\gamma/2$. So it sounds obvious that, for a given target, the current amount B_t on the bank account matters. In fact, problem (4) is *not* about (the expected utility of ?) the trading gains $B_L - B_0$, or $B_L - B_t$, but about bringing B close to $\gamma/2$. This argument holds also when thinking about targeting an expected wealth of d. The result does depend on the current wealth on the bank account.

So we seek for v^* and V as functions of (S_t, B_t, α_t, t) (the explicit dependence in γ being omitted).

1.3.2 The HJB PDEs.

We want to determine $V(S_t, B_t, \alpha_t, t) = E_{(S_t,B_t,\alpha_t,t)}\Big[\Big(B_L^{v^*} - \frac{\gamma}{2}\Big)^2\Big]$ and $U(S_t, B_t, \alpha_t, t) := E_{(S_t,B_t,\alpha_t,t)}\Big[B_L^{v^*} - \frac{\gamma}{2}\Big]$. V probably solves the HJB PDE

$$\begin{cases} -V_t = \mathcal{L}V + rBV_B + \min\limits_{v\in Z}\Big\{-vf(v)SV_B + vV_\alpha + g(v)SV_S\Big\} \\ V(S,B,\alpha,T) = \Big(L(S,B,\alpha) - \dfrac{\gamma}{2}\Big)^2. \end{cases}$$

Here, $\mathcal{L}f = \frac{\sigma^2}{2}f_{SS} + \eta S f_S$. Solving this produces $V(S_0, B_0, \alpha_I, 0) = V(\gamma)$ for a given γ, as well as the optimal strategy $(v^*(S_t, B_t, \alpha_t, t))_{0\le t\le T}$. Meanwhile U solves

$$\begin{cases} -U_t = \mathcal{L}U + rBU_B + \Big\{-v^*f(v^*)SU_B + v^*U_\alpha + g(v^*)SU_S\Big\} \\ U(S,B,\alpha,T) = L(S,B,\alpha) - \dfrac{\gamma}{2}. \end{cases}$$

1.4 Dependence on γ, change of variable, PDEs to solve.

1.4.1 Change of variables.

V and U depend on γ through their terminal condition. For each γ, one has to solve the PDEs above with a different terminal condition. Instead, we consider the following variable, which compares B_t to the actualized target :

$$\mathcal{B}_t = B_t - \frac{\gamma}{2}e^{-r(T-t)}.$$

Since L is affine in the variable B, we have

$$B_L = L(S_T, B_T, \alpha_T) - \frac{\gamma}{2} = L\Big(S_T, B_T - \frac{\gamma}{2}, \alpha_T\Big) = L(S_T, \mathcal{B}_T, \alpha_T) =: \mathcal{B}_L.$$

So the problem (4) is nothing but

$$\min_{v\in Z} E\big[(\mathcal{B}_L^v)^2\big]. \tag{5}$$

Rewrite as $\mathcal{V}(S_t, \mathcal{B}_t, \alpha_t, t) = V(S_t, B_t, \alpha_t, t)$ its dynamic value function, and $\mathcal{U}(S_t, \mathcal{B}_t, \alpha_t, t) = E_{(S_t,\mathcal{B}_t,\alpha_t,t)}[\mathcal{B}_L^{v^*}] = U(S_t, B_t, \alpha_t, t)$. However, for simplicity, I use the same notation for the optimal control function in the $\mathcal{B}$ variable, $v^*(S_t, \mathcal{B}_t, \alpha_t, t)$.

The controlled dynamics of $\mathcal{B}$ is easily found to be

$$\frac{d\mathcal{B}_t^v}{dt} = r\mathcal{B}_t^v - v_t f(v_t) S_t.$$

So it is the same exact same problem as with B. (Remark : I suppose one could also simply have done the change of variable directly in the PDEs, without rethinking the optimization problem.)

1.4.2 The PDEs effectively solved.

For $\mathcal{V}$:

$$\begin{cases} -\mathcal{V}_t = \mathcal{L}\mathcal{V} + r\mathcal{B}\mathcal{V}_{\mathcal{B}} + \min_{v\in Z}\Big\{ -vf(v)S\mathcal{V}_{\mathcal{B}} + v\mathcal{V}_\alpha + g(v)S\mathcal{V}_S\Big\} \\ \mathcal{V}(S,\mathcal{B},\alpha,T) = L(S,\mathcal{B},\alpha)^2. \end{cases} \tag{6}$$

For $\mathcal{U}$:

$$\begin{cases} -\mathcal{U}_t = \mathcal{L}\mathcal{U} + r\mathcal{B}\mathcal{U}_{\mathcal{B}} + \Big\{ -v^*f(v^*)S\mathcal{U}_{\mathcal{B}} + v^*\mathcal{U}_\alpha + g(v^*)S\mathcal{U}_S\Big\} \\ \mathcal{U}(S,\mathcal{B},\alpha,T) = L(S,\mathcal{B},\alpha). \end{cases} \tag{7}$$

1.4.3 Coming back to B, γ, and the efficient frontier.

We have

$$\mathcal{B}_0 = B_0 - \frac{\gamma}{2}e^{-rT} \Longleftrightarrow \gamma = 2(B_0 - \mathcal{B}_0)e^{rT}.$$

For fixed B_0, reading along the $\mathcal{B}_0$ axis at time 0 translated into reading various values of γ. *(We will actually draw the frontier parametrized by $\mathcal{B}_0$, so $\rightsquigarrow \gamma = \gamma(\mathcal{B}_0)$.)*

Then, we have

$$\mathcal{U}(S_0,\mathcal{B}_0,\alpha_I,0) = E\big[\mathcal{B}_L^{v^*}\big] = E\big[B_L^{v^*}\big] - \frac{\gamma}{2},$$

so

$$d^*(\gamma) = E\big[B_L^{v^*}\big] = \mathcal{U}(S_0,\mathcal{B}_0,\alpha_I,0) + \frac{\gamma(\mathcal{B}_0)}{2}.$$

Also, we have

$$\mathcal{V}(S_0,\mathcal{B}_0,\alpha_I,0) = E\big[(\mathcal{B}_L^{v^*})^2\big] = E\Big[\Big(B_L^{v^*} - \frac{\gamma}{2}\Big)^2\Big] = E\big[(B_L^{v^*})^2\big] - \gamma E\big[\mathcal{B}_L^{v^*}\big] + \frac{\gamma^2}{4},$$

so

$$\begin{aligned} \mathrm{Var}^*(\gamma) = E\big[(B_L^{v^*})^2\big] - E\big[B_L^{v^*}\big]^2 &= \mathcal{V}(S_0,\mathcal{B}_0,\alpha_I,0) + \gamma E\big[B_L^{v^*}\big] - \frac{\gamma^2}{4} - E\big[B_L^{v^*}\big]^2 \\ &= \mathcal{V}(S_0,\mathcal{B}_0,\alpha_I,0) + \gamma d^*(\gamma) - \frac{\gamma^2}{4} - d^*(\gamma)^2. \end{aligned}$$

(Remark : it seems to me as if, in the paper, Forsyth only computes $E\big[(B_L^{v^*})^2\big]$ ...)

1.5 Similarity and dimension-reduction (option in the algorithm).

What happens if we look at the problem from time t, with $(S_t, B_t) \leftarrow (\xi S_t, \xi B_t)$? Since $d(S_t, B_t)$ is linear in (S_t, B_t), the process $(\xi S, \xi B)$ has the exact same dynamics (same $\eta, g(v), \sigma, r, vf(v)$). In addition to that, the liquidation function L is linear in (S, B), so $L(\xi S, \xi \mathcal{B}, \alpha) = \xi L(S, \mathcal{B}, \alpha)$. These 2 arguments (which don't depend on the form of the price impact) seem enough to me to explain why

$$\mathcal{V}(\xi S_t, \xi \mathcal{B}_t, \alpha_t, t) = \xi^2 \mathcal{V}(S_t, \mathcal{B}_t, \alpha_t, t) \quad \text{and} \quad \mathcal{U}(\xi S_t, \xi \mathcal{B}_t, \alpha_t, t) = \xi \mathcal{U}(S_t, \mathcal{B}_t, \alpha_t, t).$$

This implies that, for arbitrary $\mathcal{B}^*$,

$$\begin{aligned} \mathcal{V}(S_t, \mathcal{B}_t, \alpha_t, t) &= \Big(\frac{\mathcal{B}_t}{\mathcal{B}^*}\Big)^2 \mathcal{V}\Big(\frac{\mathcal{B}^* S_t}{\mathcal{B}_t}, \mathcal{B}_t, \alpha_t, t\Big) \\ \mathcal{U}(S_t, \mathcal{B}_t, \alpha_t, t) &= \Big(\frac{\mathcal{B}_t}{\mathcal{B}^*}\Big)^2 \mathcal{U}\Big(\frac{\mathcal{B}^* S_t}{\mathcal{B}_t}, \mathcal{B}_t, \alpha_t, t\Big). \end{aligned}$$

Notice however that this can be used only for a $\mathcal{B}^*$ of the same sign as $\mathcal{B}_t$, so that $\frac{\mathcal{B}^* S_t}{\mathcal{B}_t}$ remains > 0 : otherwise the RHS is not defined.

This remark of the similarity/scaling property in the problem leads to the possibility of reducing the spatial dimension, by eliminating the $\mathcal{B}$ grid.

2 The algorithm.

We now explain the algorithm for solving the PDEs (6) and (7).

2.1 Semi-Lagrangian approach.

The approach consists in interpreting part of the equation as an ODE along some path. Consider the path $(X_t = (S_t, \mathcal{B}_t, \alpha_t))_{0 \le t \le T}$ defined, for a control $(v_t)_{0 \le t \le T}$, by

$$\frac{dS_t^v}{dt} = g(v) S_t \ , \qquad \frac{d\mathcal{B}_t^v}{dt} = r\mathcal{B}_t^v - v_t f(v_t) S_t^v \quad \text{and} \quad \frac{d\alpha_t^v}{dt} = v_t.$$

Still denote by $\mathcal{V}$ the path $\mathcal{V}_t = \mathcal{V}(S_t^v, \mathcal{B}_t^v, \alpha_t^v, t)$. We have

$$\frac{d\mathcal{V}_t}{dt} = \ldots = \mathcal{V}_t + r\mathcal{B}_t \mathcal{V}_{\mathcal{B}} + \{ - v_t f(v_t) S_t \mathcal{V}_{\mathcal{B}} + v\mathcal{V}_\alpha + g(v_t) S_t \mathcal{V}_S \} \quad \text{(evaluated at } (S_t, \mathcal{B}_t, \alpha_t, t)).$$

The PDEs (6)-(7) can therefore (...) be rewritten

$$\begin{cases} \min\limits_{v_t \in Z} \dfrac{d\mathcal{V}_t}{dt} + \mathcal{L}\mathcal{V} = 0 \\ \mathcal{V}(S, \mathcal{B}, \alpha, T) = L(S, \mathcal{B}, \alpha)^2 \end{cases} \quad \text{and} \quad \begin{cases} \dfrac{d\mathcal{U}_t^{v^*}}{dt} + \mathcal{L}\mathcal{U} = 0 \\ \mathcal{U}(S, \mathcal{B}, \alpha, T) = L(S, \mathcal{B}, \alpha) \end{cases}.$$

2.2 Domain discretization.

The PDEs are initially set on the domain $\mathcal{D} = \mathbb{R}_+ \times \mathbb{R} \times [v_{\min}, v_{\max}] \times [0, T]$. For the numerical solving of the PDEs, we need to reduce it to the bounded domain $D = [0, S_{\max}] \times [\mathcal{B}_{\min}, \mathcal{B}_{\max}] \times [\alpha_{\min}, \alpha_{\max}] \times [0, T]$. This domain is then discretized into a grid $\widehat{D} = \{x_{ijk}^n\}$ defined below.

We use N_t time-intervals, N_S intervals in the S direction, N_B intervals in the $\mathcal{B}$ direction and N_α intervals in the α direction. Define the grid points

$$\begin{aligned}
&t^n = n\Delta t \quad \text{where} \quad \Delta t = \frac{T}{N_t} \quad \text{and} \quad n \in \{0, \ldots, N_t\}, \\
&S_i = i\Delta S \quad \text{where} \quad \Delta S = \frac{S_{\max}}{N_S} \quad \text{and} \quad i \in \{0, \ldots, N_S\}, \\
&\mathcal{B}_j = \mathcal{B}_{\min} + j\Delta\mathcal{B} \quad \text{where} \quad \Delta\mathcal{B} = \frac{\mathcal{B}_{\max} - \mathcal{B}_{\min}}{N_B} \quad \text{and} \quad j \in \{0, \ldots, N_B\}, \\
&\alpha_k = \alpha_{\min} + k\Delta\alpha \quad \text{where} \quad \Delta\alpha = \frac{\alpha_{\max} - \alpha_{\min}}{N_\alpha} \quad \text{and} \quad k \in \{0, \ldots, N_\alpha\}.
\end{aligned}$$

I will also use the notation $x_{ijk} = (S_i, \mathcal{B}_j, \alpha_k)$ and $x_{ijk}^n = (S_i, \mathcal{B}_j, \alpha_k, t^n)$.

2.3 Boundary conditions.

There is nothing to do regarding the boundaries in the $\mathcal{B}$ and α directions. Indeed, only first derivatives appear in the PDEs (which can always be computed easily at the boundary), and even they won't be used because of the semi-Lagragian approach.

The set of admissible controls, however, should possibly be changed at the boundary. Denoting by Z^n_{ijk} the set of admissible controls at the node x^n_{ijk}, one should take $Z^n_{i,j,0} = [0, v_{\max}]$ and $Z^n_{i,j,N_\alpha} = [v_{\min}, 0]$. In general, with only buying or only selling actions allowed, one of these intervals will be the whole Z and the other the singleton 0. But we do not need to worry about this too much since, more generally, Z^n_{ijk} will be chosen at each point so that α driven by v remains in D.

Finally, in the S direction, we take $\mathcal{L}\mathcal{V} = 0$ at $S_0 = 0$ and $S_{N_S} = S_{\max}$.

2.4 Approximation of the second-order operator.

We use the approximation by centered finite differences. At $x^n_{ijk} = (S_i, \mathcal{B}_j, \alpha_k, t^n)$, $(\mathcal{L}\mathcal{V})(x^n_{ijk}) \approx (\mathcal{L}\mathcal{V})^n_{ijk}$ where

$$\begin{aligned}(\mathcal{L}\mathcal{V})^n_{ijk} &= \frac{\sigma^2 S_i^2}{2}\,\frac{\mathcal{V}^n_{i+1,j,k} - 2\mathcal{V}^n_{i,j,k} + \mathcal{V}^n_{i-1,j,k}}{(\Delta S)^2} + rS_i \frac{\mathcal{V}^n_{i+1,j,k} - \mathcal{V}^n_{i-1,j,k}}{2\Delta S} \\ &= \underbrace{\left(\frac{\sigma^2 S_i^2}{2(\Delta S)^2} - \frac{rS_i}{2\Delta S}\right)}_{\bar{a}_i} \mathcal{V}^n_{i-1,j,k} + \underbrace{\left(-\frac{\sigma^2 S_i^2}{(\Delta S)^2}\right)}_{\bar{b}_i} \mathcal{V}^n_{i,j,k} + \underbrace{\left(\frac{\sigma^2 S_i^2}{2(\Delta S)^2} + \frac{rS_i}{2\Delta S}\right)}_{\bar{c}_i} \mathcal{V}^n_{i+1,j,k},\end{aligned}$$

for all $j \in \{0, \dots, N_B\}$, $k \in \{0, \dots, N_\alpha\}$, and for all $i \in \{1, \dots, N_S - 1\}$. At $i = 0$ and $i = N_S$, we take $(\mathcal{L}\mathcal{V})^n_{ijk} = 0$, so $\bar{a}_i = \bar{b}_i = \bar{c}_i = 0$.

2.5 Approximation of the Lagrangian ODE.

2.5.1 Time-discretization

We look at the time interval $[t^n, t^{n+1}]$, the approximate solution being known at time t^{n+1}, and to be computed for time t^n. Given the grid point $x^n_{ijk} = (S_i, \mathcal{B}_j, \alpha_k, t^n)$, and a control v^n_{ijk}, the derivative along the path $(X_t(v^n_{ijk}))_{t^n \le t \le t^{n+1}}$ is approximated as

$$\frac{d\mathcal{V}_t}{dt} \approx \frac{1}{\Delta t}(\widehat{V}^{n+1}_{ijk} - V^n_{ijk}).$$

In the above, $\widehat{V}^{n+1}_{ijk}$ is an approximation of $\mathcal{V}$ at the point $\widehat{x}^{n+1}_{ijk} = (\widehat{S}_i, \widehat{\mathcal{B}}_j, \widehat{\alpha}_k, t^{n+1})$. This point is computed by solving the ODE for X between t^n and t^{n+1} with the frozen control v^n_{ijk}. So we obtain

$$\begin{aligned}\widehat{S}_i &= S_i e^{g(v^n_{ijk})\Delta t}, \\ \widehat{\mathcal{B}}_j &= \mathcal{B}_j e^{r\Delta t} - v^n_{ijk} f(v^n_{ijk}) S_i \frac{e^{g(v^n_{ijk})\Delta t} - e^{r\Delta t}}{g(v^n_{ijk}) - r} \\ \widehat{\alpha}_k &= \alpha_k + v^n_{ijk}\Delta t.\end{aligned}$$

If the control v^n_{ijk} is admissible (see below), $\widehat{x}^{n+1}_{ijk}$ lies in a box of the numerical domain (more specifically : its spatial component $\widehat{x}_{ijk}$ does, the time t^{n+1} is exactly on the grid). The value of $\mathcal{V}$ at this point is approximated by interpolation of the values of $\mathcal{V}^{n+1}$ at the grid points forming the vertices of the box where $\widehat{x}_{ijk}^{n+1}$ lies. This gives $\widehat{\mathcal{V}}_{ijk}^{n+1}(v_{ijk}^n)$. And then we will want to compute

$$\min_{v_t\in Z}\frac{d\mathcal{V}_t}{dt}\approx\min_{v_{ijk}^n\in Z_{ijk}^n}\frac{1}{\Delta t}(\widehat{\mathcal{V}}_{ijk}^{n+1}(v_{ijk}^n)-\mathcal{V}_{ijk}^n)=\frac{1}{\Delta t}(\min_{v_{ijk}^n\in Z_{ijk}^n}\widehat{\mathcal{V}}_{ijk}^{n+1}(v_{ijk}^n)-\mathcal{V}_{ijk}^n).$$

For $\mathcal{U}$, there is no optimization to do, we just use the $(v_{ijk}^n)^*$ computed for $\mathcal{V}$. So the approximation is

$$\frac{d\mathcal{U}_t^{v^*}}{dt}\approx\frac{1}{\Delta t}(\widehat{\mathcal{U}}_{ijk}^{n+1}((v_{ijk}^n)^*)-\mathcal{U}_{ijk}^n).$$

2.5.2 Interpolation.

For the fixed grid point x_{ijk}^n and associated $\widehat{x}_{ijk}^{n+1}$, the interpolation is done as follows. Define

$$i_0=\left\lfloor\frac{\widehat{S}_i-0}{\Delta S}\right\rfloor,\qquad \lambda_i=\frac{\widehat{S}_i-S_{i_0}}{\Delta S},$$
$$j_0=\left\lfloor\frac{\widehat{\mathcal{B}}_j-\mathcal{B}_{\min}}{\Delta\mathcal{B}}\right\rfloor,\qquad \lambda_j=\frac{\widehat{\mathcal{B}}_j-\mathcal{B}_{j_0}}{\Delta\mathcal{B}},$$
$$k_0=\left\lfloor\frac{\widehat{\alpha}_k-\alpha_{\min}}{\Delta\alpha}\right\rfloor,\qquad \lambda_k=\frac{\widehat{\alpha}_k-\alpha_{k_0}}{\Delta\alpha}.$$

Each index is in $\{0,\dots,N\}$, and each $\lambda\in[0,1[$.

Interpolation when similarity reduction is OFF.

The point $\widehat{x}_{ijk}^{n+1}$, lies : when projected on the S axis, between S_{i_0} and S_{i_0+1}, when projected on the $\mathcal{B}$ axis, between $\mathcal{B}_{j_0}$ and $\mathcal{B}_{j_0+1}$, when projected on the α axis, between α_{k_0} and α_{k_0+1}. Specifically, we have

$$\begin{aligned}\widehat{x}_{ijk}=&(1-\lambda_j)(1-\lambda_k)(1-\lambda_i)\ x_{i_0,j_0,k_0} && +(1-\lambda_j)(1-\lambda_k)\lambda_i\ x_{i_0+1,j_0,k_0}\\ &+(1-\lambda_j)\lambda_k(1-\lambda_i)\ x_{i_0,j_0,k_0+1} && +(1-\lambda_j)\lambda_k\lambda_i\ x_{i_0+1,j_0,k_0+1}\\ &+\lambda_j(1-\lambda_k)(1-\lambda_i)\ x_{i_0,j_0+1,k_0} && +\lambda_j(1-\lambda_k)\lambda_i\ x_{i_0+1,j_0+1,k_0}\\ &+\lambda_j\lambda_k(1-\lambda_i)\ x_{i_0,j_0+1,k_0+1} && +\lambda_j\lambda_k\lambda_i\ x_{i_0+1,j_0+1,k_0+1}.\end{aligned}$$

We therefore interpolate $\widehat{\mathcal{V}}_{ijk}^{n+1}(v_{ijk}^n)$ as

$$\begin{aligned}\widehat{\mathcal{V}}_{ijk}^{n+1}=&(1-\lambda_j)(1-\lambda_k)(1-\lambda_i)\ \mathcal{V}_{i_0,j_0,k_0}^{n+1} && +(1-\lambda_j)(1-\lambda_k)\lambda_i\ \mathcal{V}_{i_0+1,j_0,k_0}^{n+1}\\ &+(1-\lambda_j)\lambda_k(1-\lambda_i)\ \mathcal{V}_{i_0,j_0,k_0+1}^{n+1} && +(1-\lambda_j)\lambda_k\lambda_i\ \mathcal{V}_{i_0+1,j_0,k_0+1}^{n+1}\\ &+\lambda_j(1-\lambda_k)(1-\lambda_i)\ \mathcal{V}_{i_0,j_0+1,k_0}^{n+1} && +\lambda_j(1-\lambda_k)\lambda_i\ \mathcal{V}_{i_0+1,j_0+1,k_0}^{n+1}\\ &+\lambda_j\lambda_k(1-\lambda_i)\ \mathcal{V}_{i_0,j_0+1,k_0+1}^{n+1} && +\lambda_j\lambda_k\lambda_i\ \mathcal{V}_{i_0+1,j_0+1,k_0+1}^{n+1}.\end{aligned}$$

Remark : in order to avoid doing tests to know whether $i_0 = N_S$, $j_0 = N_B$ or $k_0 = N_\alpha$, it will be convenient to consider that the grid goes to "index max +1" and write the formula as is. It won't be a problem because, in such a case, $\lambda_i = 0$, $\lambda_j = 0$ or $\lambda_k = 0$ (respectively).

Naturally, the same is done to compute $\widehat{\mathcal{U}}_{ijk}^{n+1}((v_{ijk}^n)^*)$.

Interpolation when similarity reduction is ON.

When the similarity is used, we can reduce the dimension of the problem. In that case, the $\mathcal{B}$-grid with N_B points is replaced by a grid with 2 points. Let $\mathcal{B}^* = 1 > 0$, and define $\mathcal{B}_0 = -\mathcal{B}^*$ and $\mathcal{B}_1 = +\mathcal{B}^*$, so $j \in \{0, 1\}$.

Now, for any $x_{ijk}^n \in \widehat{D}$, and a control v_{ijk}^n, the point $\widehat{x}_{ijk}^{n+1}$ is computed as previously. The admissibility (cf below) is check at that point (though whether $\widehat{\mathcal{B}}_j \in [\mathcal{B}_{\min}, \mathcal{B}_{\max}]$ is not so important, morally, it is more $\widehat{S}_i$ and $\widehat{\alpha}_k$ that matter, but let us not change this). We then want to approximate the value of $\mathcal{V}$ at $\widehat{x}_{ijk}^{n+1}$. For this, we first note that

$$\mathcal{V}(\widehat{x}_{ijk}^{n+1}) = \mathcal{V}(\widehat{S}_i, \widehat{\mathcal{B}}_j, \widehat{\alpha}_k, t^{n+1}) = \begin{cases} \Big(\dfrac{\widehat{\mathcal{B}}_j}{\mathcal{B}_1}\Big)^2 \mathcal{V}\Big(\dfrac{\mathcal{B}_1 \widehat{S}_i}{\widehat{\mathcal{B}}_j}, \mathcal{B}_1, \widehat{\alpha}_k, t^{n+1}\Big) & \text{if } \widehat{\mathcal{B}}_j > 0, \\ \Big(\dfrac{\widehat{\mathcal{B}}_j}{\mathcal{B}_0}\Big)^2 \mathcal{V}\Big(\dfrac{\mathcal{B}_0 \widehat{S}_i}{\widehat{\mathcal{B}}_j}, \mathcal{B}_0, \widehat{\alpha}_k, t^{n+1}\Big) & \text{if } \widehat{\mathcal{B}}_j < 0. \end{cases}$$

So, define $\widetilde{\alpha}_k = \widehat{\alpha}_k$,

$$\begin{aligned} \widetilde{\mathcal{B}}_j = \mathcal{B}_1 \quad \text{and} \quad \widetilde{S}_i = \frac{\mathcal{B}_1 \widehat{S}_i}{\widehat{\mathcal{B}}_j} \quad \text{if } \widehat{\mathcal{B}}_j > 0, \\ \widetilde{\mathcal{B}}_j = \mathcal{B}_0 \quad \text{and} \quad \widetilde{S}_i = \frac{\mathcal{B}_0 \widehat{S}_i}{\widehat{\mathcal{B}}_j} \quad \text{if } \widehat{\mathcal{B}}_j < 0, \end{aligned}$$

and $\widetilde{x}_{ijk}^{n+1} = (\widetilde{S}_i, \widetilde{\mathcal{B}}_j, \widetilde{\alpha}_k, t^{n+1})$. Then interpolate the value $\widetilde{\mathcal{V}}_{ijk}^{n+1} \approx \mathcal{V}(\widetilde{x}_{ijk}^{n+1})$ as above and deduce $\widehat{\mathcal{V}}_{ijk}^{n+1} \approx \mathcal{V}(\widehat{x}_{ijk}^{n+1})$ from the similarity identity.

Essentially the same goes for $\mathcal{U}$, since

$$\mathcal{U}(\widehat{x}_{ijk}^{n+1}) = \mathcal{U}(\widehat{S}_i, \widehat{\mathcal{B}}_j, \widehat{\alpha}_k, t^{n+1}) = \begin{cases} \Big(\dfrac{\widehat{\mathcal{B}}_j}{\mathcal{B}_1}\Big) \mathcal{U}\Big(\dfrac{\mathcal{B}_1 \widehat{S}_i}{\widehat{\mathcal{B}}_j}, \mathcal{B}_1, \widehat{\alpha}_k, t^{n+1}\Big) & \text{if } \widehat{\mathcal{B}}_j > 0, \\ \Big(\dfrac{\widehat{\mathcal{B}}_j}{\mathcal{B}_0}\Big) \mathcal{U}\Big(\dfrac{\mathcal{B}_0 \widehat{S}_i}{\widehat{\mathcal{B}}_j}, \mathcal{B}_0, \widehat{\alpha}_k, t^{n+1}\Big) & \text{if } \widehat{\mathcal{B}}_j < 0. \end{cases}$$

So, first, interpolate the value $\widetilde{\mathcal{U}}_{ijk}^{n+1} \approx \mathcal{U}(\widetilde{x}_{ijk}^{n+1})$ as above and deduce $\widehat{\mathcal{U}}_{ijk}^{n+1} \approx \mathcal{U}(\widehat{x}_{ijk}^{n+1})$ from the similarity identity.

2.5.3 Solving the optimization problem.

At time t^n and at point x_{ijk}, we must solve an optimization problem over v^n_{ijk}. The admissible controls are those $v^n_{ijk} \in Z = [v_{\min}, v_{\max}]$ such that $\widehat{x}^n_{ijk} \in D$.

Forsyth suggests not to use a standard optimization procedure, over a continous set Z or Z^n_{ijk}, because the problem is likely to have many local minima. Instead, the set of controls is discretized as well. So we consider a given a number $N_v + 1$ of equidistant controls in $[v_{\min}, v_{\max}]$, identified by

$$v_l = v_{\min} + l\Delta v \quad \text{where} \quad \Delta v = \frac{v_{\max} - v_{\min}}{N_v} \quad \text{and} \quad l \in \{0, \ldots, N_v\}.$$

The optimization is then reduced to a full search over the v_l's. So the computation of $\min_{v^n_{ijk} \in Z^n_{ijk}} \widehat{\mathcal{V}}^{n+1}_{ijk}$ is done as follows. For $l = 0$ to N_v :

- Take v_l and compute $x^n_{ijk}(v_l)$. If it does not lie in the bounded domain D, then this control is not admissible and is discarded.
- If $x^n_{ijk}(v_l) \in D$, then compute $\widehat{\mathcal{V}}^{n+1}_{ijk}(v_l)$ and compare to the running minimum.

At the end, we have the optimal $(v^n_{ijk})^*$ and the $\widehat{\mathcal{V}}^{n+1}_{ijk}((v^n_{ijk})^*)$ used for computing $\mathcal{V}^n_{ijk}$.

2.6 Summary of the schemes for $\mathcal{V}$ and $\mathcal{U}$.

Putting together subsections 2.1, 2.4 and 2.5, the scheme is the following. Having computed the approximated solution at time t^{n+1}, we move to time t^n, and for each node (ijk) we define

$$\mathcal{V}^n_{ijk} = \min_{v^n_{ijk} \in Z^n_{ijk}} \widehat{\mathcal{V}}^{n+1}_{ijk}(v^n_{ijk}) + (\mathcal{L}\mathcal{V})^n_{ijk}\, \Delta t = \widehat{\mathcal{V}}^{n+1}_{ijk}((v^n_{ijk})^*) + (\mathcal{L}\mathcal{V})^n_{ijk}\, \Delta t.$$

Notice that this is an implicit scheme. Since $(\mathcal{L}\mathcal{V})^n_{ijk} = \bar{a}_i \mathcal{V}^n_{i-1,j,k} + \bar{b}_i \mathcal{V}^n_{i,j,k} + \bar{c}_i \mathcal{V}^n_{i+1,j,k}$, we end up having

$$\underbrace{(-\bar{a}_i \Delta t)}_{a_i} \mathcal{V}^n_{i-1,j,k} + \underbrace{(1 - \bar{b}_i \Delta t)}_{b_i} \mathcal{V}^n_{i,j,k} + \underbrace{(-\bar{c}_i \Delta t)}_{c_i} \mathcal{V}^n_{i+1,j,k} = \underbrace{\widehat{\mathcal{V}}^{n+1}_{ijk}((v^n_{ijk})^*)}_{d_{i,(j,k)}}.$$

From the boundary conditions we have, for $i = 0$, $a_0 =$ inexistant, $b_0 = 1$, $c_0 = 0$, while for $i = N_S$, $a_{N_S} = 0$, $b_{N_S} = 1$ and $c_{N_S} =$ inexistant.

For $\mathcal{U}$, we define

$$\mathcal{U}^n_{ijk} = \widehat{\mathcal{U}}^{n+1}_{ijk}((v^n_{ijk})^*) + (\mathcal{L}\mathcal{U})^n_{ijk}\, \Delta t.$$

So we end up solving, for each (jk), the system

$$a_i\, \mathcal{U}^n_{i-1,j,k} + b_i\, \mathcal{U}^n_{i,j,k} + c_i\, \mathcal{U}^n_{i+1,j,k} = \underbrace{\widehat{\mathcal{U}}^{n+1}_{ijk}((v^n_{ijk})^*)}_{e_{i,(j,k)}}.$$

2.7 The parameters of the algorithm

Model parameters

Variable name	Meaning
T	horizon
σ	volatility of the asset
η	natural drif of the asset
r	interest rate
κ_p	permanent impact parameter
κ_s	cost parameter
κ_t	temporary impact parameter
β	temporary impact parameter
$\text{rate}_{\max}$	max trading rate (absolute value)
δt	time of final liquidation (strenght of penalty)
S_0	initial price of asset
α_I	inital number of shares
B_0	initial amount on the bank account

In the code, $\alpha_{\min}$, $\alpha_{\max}$, $v_{\min}$ and $v_{\max}$ are not parameters, but are determined from α_I. If $\alpha_I = 1$ (or > 0), i.e. we are selling, then $\alpha_{\min} = 0$ and $\alpha_{\max} = \alpha_I$, $v_{\min} = -\text{rate}_{\max}$ and $v_{\max} = 0$. Same thing for buying.

Numerical parameters

Variable name	Meaning
$S_{\max}$	max value of S
$\mathcal{B}_{\min}$	min value of $\mathcal{B}$
$\mathcal{B}_{\max}$	max value of $\mathcal{B}$
N_t	number of time-steps
N_S	number of S-intervals
N_B	number of $\mathcal{B}$-intervals
N_α	number of α-intervals
N_v	number of v-intervals

References

[1] P. Forsyth, A Hamilton Jacobi Bellman approach to optimal trade execution. *Applied numerical mathematics*, 61:2, 241-265 (2014).