Efficient Simulation Method for Dynamic Portfolio Selection with Transaction Cost, Liquidity Cost and Market Impact

Rongju Zhang[*], Nicolas Langrené[†], Yu Tian[‡], Zili Zhu[§], Fima Klebaner[¶] and Kais Hamza[‖]

Abstract

We develop an efficient method for solving dynamic portfolio selection problems in the presence of transaction cost, liquidity cost and market impact. Our method, based on least-squares Monte Carlo simulation, has no restriction on return dynamics, portfolio constraints, intermediate consumption and investor's objective. We model return dynamics as exogenous state variables and model portfolio weights, price dynamics and portfolio value as endogenous state variables. This separation allows for incorporation of any formation of transaction cost, liquidity cost and market impact. We first perform a forward simulation for both exogenous and endogenous state variables, then use a least-squares regression to approximate the backward recursive dynamic programs on a discrete grid of controls. Finally, we use a local interpolation and an adaptive refinement grid to enhance the optimal allocation estimates. The computational runtime of this framework grows polynomially with dimension. Its viability is illustrated on a realistic portfolio allocation example with twelve risky assets.

Keywords: dynamic portfolio selection; transaction cost; liquidity cost; market impact; least-squares Monte Carlo; optimal stochastic control

JEL Classification: G11; C61; C63; C15

MSC Classification: 93E20; 91G10; 65C05; 90C15; 90C39; 91G80; 93E35; 93E24; 62L20

[*] Corresponding author. Email: rongju.zhang@monash.edu. School of Mathematical Sciences, Monash University

[†] Real Options and Financial Risk, CSIRO

[‡] School of Mathematical Sciences, Monash University

[§] Real Options and Financial Risk, CSIRO

[¶] School of Mathematical Sciences, Monash University

[‖] School of Mathematical Sciences, Monash University

1 Introduction

Fund managers actively seek to improve their portfolio selection (also well-known as asset allocation, portfolio optimization and portfolio management) strategies using dynamic schemes. Due to analytical intractability, most solutions are based on numerical approximations. Unfortunately, up to date, the existing numerical methods are either computationally expensive or restricted in the range of applications. This paper proposes a numerical framework whose computational runtime scales polynomially with dimension and it has no restriction on return dynamics, portfolio constraints, intermediate consumption and investor's objective. In particular, this framework allows incorporation of any formation of transaction cost, liquidity cost and market impact.

The original literature on dynamic portfolio selection was started with the problems in the absence of transaction cost. The pioneers, Merton (1969), Merton (1971), Samuelson (1969) and Mossin (1968) provide closed-form solutions of optimal asset allocation strategies for long-term investors. Unfortunately, these closed-form solutions are severely restricted in terms of return dynamics and investor's preferences. To incorporate more realistic components, researchers therefore proposed a variety of numerical methods. The use of the least-squares Monte Carlo (LSMC) algorithm (Carriere (1996), Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001) on pricing American options) for solving portfolio selection problems was pioneered by Brandt, Goyal, Santa-Clara, and Stroud (2005). In the process, Brandt et al. (2005) determine a semi-closed form by solving the first order condition of the Taylor series expansion of the investor's future value function. Garlappi and Skoulakis (2010) claim that the convergence of Brandt et al. (2005)'s method is not stable and that it cannot handle problems where the control variable depends on some endogenous state variables such as portfolio wealth. Instead, they introduce a state variable decomposition method to tackle this drawback. Cong and Oosterlee (2016) use a multi-stage strategy to perform forward simulation of control variables and wealths which are iteratively updated in the backward recursive programming. Although this method allows for using wealth-dependent control variables, it is restricted to the mean-variance framework. One major limitation of these literatures is the absence of transaction cost.

In practice, every transaction incurs a cost. Several improvements have therefore been proposed to incorporate transaction costs. Examples of closed-forms are Davis and Norman (1990), Shreve and Soner (1994), Liu (2004), Gârleanu and Pedersen (2013) and Cui, Gao, Li, and Li (2014). Examples of general numerical methods are Lynch and Tan (2010) and Bao, Zhu, Langrené, and Lee (2014). However, both methods suffer at some point from the so-called 'curse of dimensionality' with respect to the number of assets, notably when discretizing the set of feasible allocations. To alleviate this problem, Cai, Judd, and Xu (2013) parallelize the discrete control grid. We refer to Cai, Judd, Thain, and Wright (2015) for the details of the parallelization technique. Still, this method does not solve the curse of dimensionality but simply shuns it by relying on pure computer power. Muthuraman and Zha (2008) solve a specific problem using a computational scheme whose runtime scales polynomially in dimension. The limitation is the lack of flexibility in solving other problems. Brown and Smith (2011) compute upper bounds of the objective function by using perfect information about the future. The complexity of their method grows more than linearly but less than polynomially in dimension, but the approximation is not accurate under return predictability. Moallemi and Sağlam (2015) introduce a class of efficient heuristic based on linear rebalancing rules, but their approximation is also not accurate.

	Objective	Dynamics	TC	LC&MI	Complexity	Accuracy
Liu (2004)	Specific	Restricted	Yes	No	Fast	Exact
Brandt et al. (2005)	General	General	No	No	>>Exp.	High
Muthuraman and Zha (2008)	Specific	Specific	Yes	No	Poly.	High
Garlappi and Skoulakis (2010)	General	General	No	No	>>Exp.	High
Lynch and Tan (2010)	General	General	Yes	No	>>Exp.	High
Brown and Smith (2011)	General	General	Yes	No	<Poly.	Low
Gârleanu and Pedersen (2013)	Specific	Restricted	Yes	Yes	Fast	Exact
Bao et al. (2014)	Restricted	General	Yes	No	>>Exp.	High
Moallemi and Sağlam (2015)	General	General	Yes	No	Fast	Low
Cong and Oosterlee (2016)	Specific	General	No	No	>>Exp.	High
This paper	General	General	Yes	Yes	Poly.	High

Table 1.1: Related literature

Beyond transaction cost, liquidity cost as well as permanent market impact is attracting more and more attention. Here are some noteworthy existing contributions, albeit restricted in applications. One of the most influential frameworks is developed by Gârleanu and Pedersen (2013), where the authors derive a closed-form optimal portfolio policy for the mean-variance framework with quadratic transaction costs such that liquidity cost and market impact are included. Following on this framework, many extensions have been proposed, for example Collin-Dufresne, Daniel, Moallemi, and Sağlam (2014), DeMiguel, Mei, and Nogales (2014) and Mei and Nogales (2015). Some other contributions dealing with liquidity cost and market impact are He and Mamaysky (2005) Vath, Mnif, and Pham (2007), Lim and Wimonkittiwat (2014) and Tsoukalas, Wang, and Giesecke (2015).

Table 1.1 summarizes notable contributions for solving dynamic portfolio selection problems. Yet, a precise, efficient and general method with transaction cost, liquidity cost and market impact is still under investigation. To a large extend, the present paper fills this gap. Our contributions are two-fold:

1. We extend the LSMC algorithm to deal with switching cost and endogenous state variable. We model return dynamics as exogenous state variables and model portfolio weights, price dynamics and portfolio value as endogenous state variables. To deal with these endogenous state variables, we rely on Kharroubi, Langrené, and Pham (2014)'s control randomization technique. This allows the incorporation of any formation of transaction cost, liquidity cost and market impact.

2. We use a local interpolation to generalize the value functions estimated on a discrete grid of portfolio weights, then perform an adaptive refinement algorithm to estimate the continuous optimal control. We show that the results produced by a local interpolation are more accurate and more efficient than a global interpolation. Moreover, under the local interpolation, a coarse grid can achieve the same accuracy as a fine grid, which substantially improves the computational efficiency.

The outline of the paper is as follows. Section 2 describes our dynamic portfolio selection problem. Section 3 describes the methodology. Section 4 discusses numerical experiments and Section 5 concludes the paper.

2 The Portfolio Optimization Problem

We consider a dynamic portfolio selection problem over a finite time horizon T. Suppose there are N assets available for investment. Let $R = \left(R_t^i\right)_{i=1,\dots,N}$ and $S_t = \left(S_t^i\right)_{i=1,\dots,N}$ respectively denote the assets' returns and prices at time $t \in [0,T]$. Denote Z_t as the vector of return predictors of R_t for $t \in [0,T]$. This vector Z_t is used to construct the dynamics of the assets. We assume discrete portfolio rebalancing times and denote $\mathcal{T} = \{1,\dots,T-1\}$ as the set of possible rebalancing times. Let $\alpha_t = \left(\alpha_t^i\right)_{i=1,\dots,N}$ be the portfolio weight in each asset at time $t \in \mathcal{T}$. Let $\mathcal{C}_t \subseteq \mathbb{R}^N$ be the set of admissible portfolio strategies at each rebalancing time $t \in \mathcal{T}$. These sets may include constraints defined by the investor (weight limits in each individual asset for example). We assume that there is no-shorting constraint and the set of admissible strategies is constant over time, i.e., $\mathcal{C} = \mathcal{C}_t$ for $t \in \mathcal{T}$. Finally, let W_t denote the portfolio value (or portfolio wealth) at time $t \in [0,T]$. We suppose that the investor tries to maximize his expected utility of final wealth $\mathbb{E}\left[U(W_T)\right]$ over all the possible strategies $\{\alpha_t \in \mathcal{C}_t\}_{t\in\mathcal{T}}$. In this paper, without loss of generality, we investigate on the constant relative risk-aversion (CRRA) utility, i.e. $U(w) = w^{1-\gamma}/(1-\gamma)$, where γ is the parameter controlling the risk-aversion level.

To sum up, we are facing a multivariate stochastic control problem. The state variables of the problem are:

- **Exogenous state variables**. The return predictors are exogenous stochastic risk factors.
- **Endogenous state variables**. The asset prices, the portfolio weights and the portfolio value are endogenous risk factors, as rebalancing the portfolio incurs transaction cost, liquidity cost and market impact.

Let $\mathcal{F} = \{\mathcal{F}_t\}_{t\in[0,T]}$ be the filtration generated by all the state variables. The objective function reads

$$V_t = \sup_{\{\alpha_s \in \mathcal{C}_s\}, s\in[t,T)} \mathbb{E}\left[U(W_T)\,|\mathcal{F}_t\,\right],\ t \in \mathcal{T}, \tag{2.1}$$

where V_t and α_t are $\mathcal{F}_t$-adapted. In particular, $V_t = V_t(X_t, \alpha_t)$ and $\alpha_t = \alpha_t(X_{t^-}, \alpha_{t^-})$ where the vector of stochastic state variables $X_t := (Z_t, R_t, S_t, W_t)$ contains all the information at time t other than the portfolio weights α_t. The main unknowns of the problem are therefore the whole rebalancing functions $\alpha_t : (x,a) \to \alpha_t\,(x,a) \in \mathcal{C}_t$ for every $t \in \mathcal{T}$.

For every rebalancing time $t \in \mathcal{T}$, let $\mathbf{q}_t = \left(q_t^i\right)_{i=1,\dots,N}$ describes the number of units held in each asset. In particular, $W_t = \mathbf{q}_t \cdot S_t = \sum_{i=1}^N q_t^i S_t^i$ where '$\cdot$' denotes the dot product between two vectors. We assume that these quantities are constant between two rebalancing dates. Let TC_t, LC_t and MI_t denote respectively transaction cost, liquidity cost and market impact at time $t \in \mathcal{T}$ when rebalancing the portfolio from $\mathbf{q}_{t-1}$ to $\mathbf{q}_t$. Define $\Delta\mathbf{q}_t := \mathbf{q}_t - \mathbf{q}_{t-1}$. In particular, $\mathrm{TC}_t = \mathrm{TC}(\Delta\mathbf{q}_t)$, $\mathrm{LC}_t = \mathrm{LC}(\Delta\mathbf{q}_t)$ and $\mathrm{MI}_t = \mathrm{MI}(\Delta\mathbf{q}_t)$, with $\mathrm{TC} : \mathbb{R}^N \to \mathbb{R}$, $\mathrm{LC} : \mathbb{R}^N \to \mathbb{R}$ and $\mathrm{MI} : \mathbb{R}^N \to \mathbb{R}^N$.

Given an initial portfolio value W_0 and initial asset price S_0, the wealth of the investor evolves as follows:

$$\begin{aligned}
S_{1^-} &= S_0 \odot R_1 & W_{1^-} &= W_0 + (S_{1^-} - S_0) \cdot \mathbf{q}_0 \\
S_1 &= S_{1^-} + \mathrm{MI}_1 & W_1 &= W_{1^-} - \mathrm{LC}_1 - \mathrm{TC}_1 + \mathrm{MI}_1 \cdot \mathbf{q}_1 \\
&\vdots \\
S_{t^-} &= S_0 \odot R_1 & W_{t^-} &= W_{t-1} + (S_{t^-} - S_{t-1}) \cdot \mathbf{q}_{t-1} \\
S_t &= S_{t^-} + \mathrm{MI}_t & W_t &= W_{t^-} - \mathrm{LC}_t - \mathrm{TC}_t + \mathrm{MI}_t \cdot \mathbf{q}_t \\
&\vdots \\
S_{(T-1)^-} &= S_{T-1} \odot R_{T-1} & W_{(T-1)^-} &= W_{T-2} + (S_{(T-1)^-} - S_{T-2}) \cdot \mathbf{q}_{T-2} \\
S_{T-1} &= S_{(T-1)^-} + \mathrm{MI}_{T-1} & W_{T-1} &= W_{(T-1)^-} - \mathrm{LC}_{T-1} - \mathrm{TC}_{T-1} + \mathrm{MI}_{T-1} \cdot \mathbf{q}_{T-1} \\
S_{T^-} &= S_{T-1} \odot R_T & W_{T^-} &= W_{T-1} + (S_{T^-} - S_{T-1}) \cdot \mathbf{q}_{T-1} \\
S_T &= S_{T^-} & W_T &= W_{T^-}
\end{aligned} \tag{2.2}$$

Remark that there is no transaction at time T[1].

There are two possible descriptions of the portfolio positions:

1. *Absolute positions* using the quantity (number of units) in each asset $\mathbf{q}_t$.

2. *Relative positions* using the proportions of wealth in each asset α_t.

One can easily switch from one description to the other, as $q_t^i S_t^i = \alpha_t^i W_t$ for each asset $i = 1, \ldots, N$ and rebalancing time $t \in \mathcal{T}$. Both descriptions will be needed, as transaction cost, liquidity cost and market impact depend on $\mathbf{q}$, while portfolio strategies are described in terms of α.

3 Extended Least-Squares Monte Carlo

This section describes the extended Least-Squares Monte Carlo (LSMC) algorithm we use to solve the dynamic portfolio selection problem in the presence of transaction cost, liquidity cost and market impact.

The LSMC algorithm, first introduced by Carriere (1996) and popularized by Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001) for American option pricing, was later extended to more general optimal switching problems such as gas storage valuation (Boogert and de Jong (2008)) or investment in power plants (Aïd, Campi, Langrené, and Pham (2014)). Brandt et al. (2005) and Bao et al. (2014) adapt this method to dynamic portfolio selection problems. These two papers mainly differ on how the optimal portfolio weights are computed in the backward loop: Brandt et al. (2005) determine a semi-closed form solution by maximizing a Taylor series expansion of the value function, while Bao et al. (2014) perform an exhaustive maximization over a fixed grid of discrete portfolio weights.

Our algorithm is made of three main parts:

1. A forward simulation for both the exogenous and endogenous risk factors (Subsection 3.2).

[1] For every $t \in \mathcal{T}$, $W_{t^-} \geq W_t$, and $W_{t^-} = W_t$ if the optimal decision is no-rebalancing, as is the case at time T.

2. A backward recursive dynamic programming loop during which the conditional expectations on a discrete grid of controls are approximated by least-squares regression, and a first-layer discrete optimal control estimate is obtained by exhaustive search (Subsection 3.3).

3. Based on the approximated conditional expectations and the first-layer discrete optimal control estimates, local interpolation and adaptive refinement grids are implemented to obtain an improved second-layer continuous optimal decision estimate (Subsection 3.4).

Before getting into the details of these three main parts, we need some preliminary remarks on how to switch from absolute portfolio positions to relative portfolio weights when transaction cost, liquidity cost and market impact are taken into account (Subsection 3.1).

3.1 Absolute position versus relative weight

As discussed in Section 2, portfolio positions can be described in terms of either relative weight α_t or absolute position $\mathbf{q}_t$, where α_t is modeled as the control variable while absolute transaction volume $\Delta\mathbf{q}_t$ is associated with transaction cost, liquidity cost and market impact. Fortunately, there is a natural one-to-one correspondence between these two descriptions, namely $\alpha_t \odot W_t = \mathbf{q}_t \odot S_t$, $t \in \mathcal{T}$, where $\odot$ denotes the element-wise multiplication between two vectors. Suppose that at time $t \in \mathcal{T}$, one wants to rebalance the portfolio from absolute position $\mathbf{q}_{t-1}$ to relative weight $\alpha_t \in \mathcal{C}_t$. Then, using the portfolio value dynamics (2.2), the following system of equations hold:

$$\alpha_t \times \left(W_{t^-} - \mathrm{TC}(\Delta\mathbf{q}_t) - \mathrm{LC}(\Delta\mathbf{q}_t) + \mathrm{MI}(\Delta\mathbf{q}_t)\cdot\mathbf{q}_t\right) = \mathbf{q}_t \odot \left(S_{t^-} + \mathrm{MI}(\Delta\mathbf{q}_t)\right). \tag{3.1}$$

This is a system of non-linear equations coupled by the wealth variable. To solve it numerically (i.e. being given α_t and $\mathbf{q}_{t-1}$, find $\mathbf{q}_t$) we use a fixed-point argument as described by Algorithm 1.

Algorithm 1 Calculate $\mathbf{q}_t$

```
**Input:** $\mathbf{q}_{t-1}$, $S_{t^-}$ , $W_{t^-}$ and $\alpha_t$
**Result:** $\mathbf{q}_t$, $S_t$ and $W_t$
Set tol
Initial guess: $\mathbf{q}_t = \alpha_t \times W_{t^-}/S_{t^-}$
**while** dist > tol **do**
    $S_t = S_{t^-} + \mathrm{MI}(\Delta\mathbf{q}_t)$
    $W_t = W_{t^-} - \mathrm{TC}(\Delta\mathbf{q}_t) - \mathrm{LC}(\Delta\mathbf{q}_t) + \mathrm{MI}(\Delta\mathbf{q}_t)\cdot\mathbf{q}_t$
    $\mathbf{q}_t^{\mathrm{aux}} = \alpha_t \times W_t/S_t$
    dist $= \sum |\mathbf{q}_t^{\mathrm{aux}} - \mathbf{q}_t|/\mathbf{q}_t^{\mathrm{aux}}$
    $\mathbf{q}_t = \mathbf{q}_t^{\mathrm{aux}}$
**end while**
```

Based on our numerical experiment, a stable solution can be reached within three iterations for a tolerance set to tol $= 10^{-4}$.

3.2 Monte Carlo simulations

The first main part of our LSMC algorithm is the forward simulation. It consists in simulating a large sample of all the stochastic risk factors. The return predictors Z_t and the asset returns $R_t = \left(R_t^i\right)_{i=1,\ldots,N}$

are exogenous risk factors, and therefore easy to simulate. By contrast, the portfolio value W_t is an endogenous risk factor, i.e. its dynamics depends on the control α_t and the asset price S_t, where S_t depends on α_t due to permanent market impact. In order to simulate W_t, which is needed to initiate the algorithm, we rely on the control randomization technique of Kharroubi et al. (2014). In summary, we first simulate a sample of return predictors Z_t, asset return R_t and random portfolio weights $\tilde{\alpha}_t$, then compute the corresponding portfolio positions $\tilde{\mathbf{q}}_t$, asset prices $\tilde{S}_t$ and portfolio value $\tilde{W}_t$ using Algorithm 1. Algorithm 2 describes this forward loop.

Algorithm 2 Forward simulation

```
Input: $S_0$, $W_0$ and $M$ (number of Monte Carlo simulations)
for each $m = 1, \dots, M$ do
  Simulate $\tilde{\alpha}_0^m$ and compute $\tilde{\mathbf{q}}_0^m = W_0/S_0$
  for each $t \in \mathcal{T}$ do
    Simulate $Z_t^m$, $R_t^m$ and $\tilde{\alpha}_t^m$
    Compute pre-decision state variables: $\tilde{S}_{t^-}^m = \tilde{S}_{t-1}^m R_t^m$ and $\tilde{W}_{t^-}^m = \tilde{S}_{t^-}^m \cdot \tilde{\mathbf{q}}_{t-1}^m$
    Compute post-decision state variables:
      Compute $\tilde{\mathbf{q}}_t^m$ from $(\tilde{\mathbf{q}}_{t-1}^m, \tilde{S}_{t^-}^m, \tilde{\alpha}_t^m, \tilde{W}_{t^-}^m)$ using Algorithm 1
      Compute $\tilde{S}_t^m = \tilde{S}_{t^-}^m + \mathrm{MI}(\Delta\tilde{\mathbf{q}}_t^m)$
      Compute $\tilde{W}_t^m = \tilde{W}_{t^-}^m - \mathrm{TC}(\Delta\tilde{\mathbf{q}}_t^m) - \mathrm{LC}(\Delta\tilde{\mathbf{q}}_t^m) + \mathrm{MI}(\Delta\tilde{\mathbf{q}}_t^m) \cdot \tilde{\mathbf{q}}_t^m$ or $\tilde{W}_t^m = \tilde{S}_t^m \cdot \tilde{\mathbf{q}}_t^m$
  end for
end for
```

3.3 Regressions and maximizations

The second part of the least-squares Monte Carlo algorithm is the regression and maximization by exhaustive search. At time T, the objective function (2.1) is equal to $V_T = U(W_T)$. Then, going one time step backward, the objective function at time $T-1$ is given by

$$\begin{aligned} V_{T-1} &= \sup_{\alpha_{T-1} \in \mathcal{C}_{T-1}} \mathbb{E}\left[V_T \,|\mathcal{F}_{T-1}\right] \\ &= \max_{a^j \in \mathcal{C}_{T-1}} \mathbb{E}\left[V_T \,\middle|X_{T-1}, \alpha_{T-1} = a^j\right] \\ &=: \max_{a^j \in \mathcal{C}_{T-1}} \mathrm{CV}_{T-1}^j(X_{T-1}). \end{aligned}$$

where $X_{T-1} := (Z_{T-1}, R_{T-1}, S_{T-1}, W_{T-1})$. In general, these conditional expectations cannot be computed explicitly, but can be approximated by regression for example, following the least-squares Monte Carlo approach. Here, an additional difficulty comes from the fact that the dynamics of the wealth W_t depends on the control variable α_t and the control-dependent asset prices S_t. To deal with it, we adopt the control randomization approach from Kharroubi et al. (2014). More precisely, we extend their approach to the optimal switching framework. In doing so, we take advantage of the discrete nature of the control set $\mathcal{C}$. Algorithm 3 describes how we proceed at time $T-1$. To make use of the whole sample, we recompute W_{T-1} and W_T for each $j|a^j \in \mathcal{C}_{T-1}$, then perform least-squares regressions to obtain estimators of the 'continuation value' CV_{T-1}^j for each $a^j \in \mathcal{C}_{T-1}$. We denote $L(x) = \big(L^1(x), \dots, L^K(x)\big)$ as the vector of basis functions for the regression.

We then iterate the same operation for $t = T-2, \dots, 0$. Again, we try to approximate the dynamic

Algorithm 3 Continuation value estimates at time $T-1$

```
Input: $Z_{T-1}^m$, $R_{T-1}^m$, $\tilde{S}_{(T-1)^-}^m$, $\tilde{\mathbf{q}}_{T-2}^m$, $\tilde{W}_{(T-1)^-}^m$, $R_T^m$ for all paths $m = 1, ..., M$
Result: $\hat{\beta}_{T-1}^j$, $\widehat{\mathrm{CV}}_{T-1}^j(x)$, for all the decisions $j | a^j \in \mathcal{C}_{T-1}$
for decisions $a^j \in \mathcal{C}_{T-1}$ do
    for paths $m = 1, ..., M$ do
        Compute $(\mathbf{q}_{T-1}^{j,m}, S_{T-1}^{j,m}, W_{T-1}^{j,m})$ from $(\mathbf{q}_{T-2}^{j,m}, S_{(T-1)^-}^{j,m}, W_{(T-1)^-}^{j,m}, \alpha_{T-1}^m = a^j)$ using Algorithm 1
        Compute terminal wealth $W_T^{j,m} = W_{T^-}^{j,m} = S_{T^-}^{j,m} \cdot \mathbf{q}_{T-1}^{j,m} = S_{T-1}^{j,m} R_T^m \cdot \mathbf{q}_{T-1}^{j,m}$
        Compute value function $\hat{V}_T^{j,m} = U(W_T^{j,m})$
    end for
    Perform least-squares regression: $\hat{\beta}_{T-1}^j = \arg\min_\beta \sum_{m=1}^M \left(\beta \cdot L(X_{T-1}^{j,m}) - \hat{V}_T^{j,m}\right)^2$
    where $X_{T-1}^{j,m} = (Z_{T-1}^m, R_{T-1}^m, S_{T-1}^{j,m}, W_{T-1}^{j,m})$
    Estimate 'continuation values': $\widehat{\mathrm{CV}}_{T-1}^j(x) = \hat{\beta}_{T-1}^j \cdot L(x)$
end for
```

programming equations

$$\begin{aligned} V_t &= \max_{a^j \in \mathcal{C}_t} \mathbb{E}\left[V_{t+1} \,\middle|\, X_t, \alpha_t = a^j\right] \\ &= \max_{a^j \in \mathcal{C}_t} \mathbb{E}\left[\max_{a^l \in \mathcal{C}_{t+1}} CV_{t+1}^l(X_{t+1}) \,\middle|\, X_t, \alpha_t = a^j\right] \end{aligned} \tag{3.2}$$

Compared to Algorithm 3, an additional loop on the portfolio weights is needed when $t < T-1$. The portfolio value W_{t+1} is not necessarily continuous, i.e. there is a potential jump due to intermediate costs. Algorithm 4 describes how to proceed.

Remark 3.1. There exists several practical tricks to improve the numerical accuracy of Algorithms 3 and 4. For example:

- The quality of the regressions may be improved by transforming the risk factors before plugging them into the basis L. Transforming the risk factors is theoretically innocuous, as if X is a real-valued random variable and $f : \mathbb{R} \to \mathbb{R}$ is a bijection, then the σ-algebras $\sigma(X)$ and $\sigma(f(X))$ are the same. In practice however, it can improve substantially the quality of the regressions. One very common choice for f is to normalize the risk factors to zero mean and unit variance. In our case, we transform W into $U(W/W_0)$. Unsurprisingly, it significantly improves the least-squares regressions, as this transform mimics the shape of the final value $V_T = U(W_T)$.

- Another potential improvement is to reduce the number of risk factors used as predictors for the regressions. Indeed, even though in theory all risk factors need to be included in order to transmit all the information available, in practice the bias-variance tradeoff means it can be beneficial to scrap the risk factors that bring little to no additional information. For example, if the returns R_t are a direct, deterministic function of the return predictors Z_t, then the returns can be removed from the vector of state variables X_t, as the information they provide would be redundant. In our tests, the wealth W_t proved to be the most important risk factor, followed by the return predictors Z_t, while the prices S_t bring no improvement (and because of the dimension increase and variance they introduce, they actually damage the estimates) and are better to be removed

Algorithm 4 Continuation value estimates at time $T-2,\dots,0$

```
Input: $Z_t^m$, $R_t^m$, $\tilde{\mathbf{q}}_{t-1}^m$, $\tilde{S}_{t^-}^m$, $\tilde{W}_{t^-}^m$, $R_{t+1}^m$, $t = T-2,\dots,1$, $m = 1,\dots,M$ and $\widehat{\mathrm{CV}}_{T-1}^j$ for each $j$ s.t. $a^j \in \mathcal{C}_{T-1}$
Result: $\hat{\beta}_t^j$, $\widehat{\mathrm{CV}}_t^j(x)$, $\hat{V}_0$
for rebalancing times $t = T-2,\dots,0$ do
  for decisions $a^j \in \mathcal{C}_t$ do
    for paths $m = 1,...,M$ do
      Compute $(\mathbf{q}_t^{j,m}, S_t^{j.m}, W_t^{j,m})$ from $(\mathbf{q}_{t-1}^{j,m}, \tilde{S}_{t^-}^{j,m}, \tilde{W}_{t^-}^{j,m}, \alpha_t^m = a^j)$ using Algorithm 1
      Compute one-step forward wealth $W_{(t+1)^-}^{j,m} = S_{(t+1)^-}^{j,m} \cdot \mathbf{q}_t^{j,m} = S_t^{j,m} R_{t+1}^m \cdot \mathbf{q}_t^{j,m}$
      for each decisions $a^l \in \mathcal{C}_{t+1}$ do
        Compute $(\mathbf{q}_{t+1}^{l,m}, S_{t+1}^{l,m}, W_{t+1}^{l,m})$ from $(\mathbf{q}_t^{j,m}, S_{(t+1)^-}^{j,m} W_{(t+1)^-}^{j,m}, \alpha_{t+1}^m = a^l)$ using Algorithm 1
        Compute ‘continuation value’ $\widehat{\mathrm{CV}}_{t+1}^{l,m} = \hat{\beta}_{t+1}^l \cdot L(X_{t+1}^{l,m})$
        where $X_{t+1}^{l,m} = (Z_{t+1}^m, R_{t+1}^m, S_{t+1}^{l,m}, W_{t+1}^{l,m})$
      end for
      Compute value function by maximizing ‘continuation value’ $\hat{V}_{t+1}^{j,m} = \max_{l \,|\, a^l \in \mathcal{C}_{t+1}} \widehat{\mathrm{CV}}_{t+1}^{l,m}$
    end for
    if $t > 0$ then
      Perform least-squares regression $\hat{\beta}_t^j = \arg\min_\beta \sum_{m=1}^M \left(\beta.L(X_t^{j,m}) - \hat{V}_{t+1}^{j,m}\right)^2$
      where $X_t^{j,m} = (Z_t^m, R_t^m, S_t^{j,m}, W_t^{j,m})$
      Formulate ‘continuation value’ $\widehat{\mathrm{CV}}_t^j(x) = \hat{\beta}_t^j \cdot L(x)$
    else
      Compute initial ‘continuation value’ $\widehat{\mathrm{CV}}_0^j = \frac{1}{M}\sum_{m=1}^M \hat{V}_1^{j,m}$
    end if
  end for
end for
Compute the initial value function by maximizing ‘continuation value’ $\hat{V}_0 = \max_{j \,|\, a^j \in \mathcal{C}_0} \widehat{\mathrm{CV}}_0^j$
Remark that there is no transaction cost, liquidity cost or market impact at time 0.
```

from the regression inputs.

- The choice of the dummy randomizations $\tilde{\alpha}$ and $\tilde{W}$ does not affect the theoretical convergence of the algorithm (cf. Kharroubi et al. (2014)). However, for a given fixed sample size M, it may affect the quality of the regressions. One possibility to substantially mitigate this effect is to run a second time Algorithms 3 and 4 with $\tilde{\alpha}$ and $\tilde{W}$ replaced by the out-of-sample estimates $\hat{\alpha}$ and $\hat{W}$ produced by the first run of the algorithms, see Cong and Oosterlee (2016) for example. After each iterations, the whole processes α and W get closer to the optimal ones α^* and W^*, though in our experiments one-single loop already converges. We will compare the purely random controls and a second-time iteration in Section 4.

- Algorithm 4 regresses on previously estimated continuation functions, in the lines of Carriere (1996) and Tsitsiklis and Van Roy (2001)'s scheme for American options, a.k.a. value function iteration (VFI). Conversely, Longstaff and Schwartz (2001) discard the continuation function estimates, using them only to estimate the optimal policy, and keep track of the sample paths under estimated optimal policy, a.k.a. policy function iteration (PFI). PFI is usually more stable, as regression errors are not compounded. One can build a PFI of Algorithm 4, but as in our case the wealth W is endogenous, it requires to recompute W_s, $s = t, \ldots, T$ for every $t \in \mathcal{T}$, increasing the computational complexity from linear to quadratic in T. Generally, the choice between VFI and PFI will depend on the problem at hand, in particular on the quality of the VFI, on the number of rebalancing times $\#\mathcal{T}$, and on the availability of a good regression basis. We will discuss the Monte Carlo convergence under these two schemes in Section 4.

3.4 Local Interpolation

The maximization procedure used for each Monte Carlo path in Algorithm 4 is a simple exhaustive search of the optimal portfolio weights, yielding an overall computational complexity of $\mathcal{O}(M \times \#\mathcal{T} \times (\#\mathcal{C})^2)$. Unfortunately, enumerating the number of possible allocations $\#\mathcal{C}$ is a so-called ‘stars and bars’ combinatorial problem, which suffers from the ‘curse of dimensionality’ for small grid mesh. Table 3.1 reports the number of strategies $\#\mathcal{C}$ for different mesh sizes and different dimensions (number of assets). For a mesh size of 1/100, which is a commonly accepted allocation precision, $\#\mathcal{C}$ explodes very quickly in dimension.

Dimension	Mesh size					
	1/2	1/4	1/8	1/16	1/32	1/100
2	3	5	9	17	33	101
3	6	15	45	153	561	5,151
4	10	35	165	969	6,545	176,851
5	15	70	495	4,845	58,905	4,598,126
6	21	126	1,287	20,349	435,897	96,560,646
7	28	210	3,003	74,613	2,760,681	1,705,904,746
8	36	330	6,435	245,157	15,380,937	26,075,972,546
9	45	495	12,870	735,471	76,904,685	352,025,629,371
10	55	715	24,310	2,042,975	350,343,565	4,263,421,511,271
			$\vdots$			
N	$\binom{N+1}{2}$	$\binom{N+3}{4}$	$\binom{N+7}{8}$	$\binom{N+15}{16}$	$\binom{N+31}{32}$	$\binom{N+99}{100}$

Table 3.1: Number of strategies

To improve the computational efficiency, we introduce a hybrid maximization approach. We first perform an exhaustive search on a coarse grid, for example with a mesh of 1/4, then we improve this first estimate by interpolation. Indeed, a continuous (i.e. outside the grid) optimal control estimate can be obtained by interpolating the value function estimates of the discrete control grid into the entire control space. For $t = T-2, ..., 0$, the Bellman equation is computed by

$$V_t = \max_{a^j \in \mathcal{C}_t} \mathbb{E}\left[\max_{a \in [0,1]^N} CV^a_{t+1}(X_{t+1}) \,\middle|\, X_t, \alpha_t = a^j \right],$$

where the inner maximization is performed over a continuous control space. The idea is to interpolate the estimated $\widehat{\mathrm{CV}}^l_{t+1}(X_{t+1})$ on $a^l \in \mathcal{C}_{t+1}$ to obtain a parametric form on the continuous control space. However, performing a plain global interpolation has several drawbacks:

- It is hard to choose an appropriate order of basis functions (for each simulation path and each time step).
- It can be hard to fit the global grid well even with high order advanced fitting (such as multivariate adaptive regression splines (MARS, Friedman (1991)) for example).
- It can be computationally expensive to fit the global grid.

See Figure 3.1 for examples of global interpolation for a three-dimensional portfolio using four different bases.

3.4.1 Local interpolation

To bypass the limitations of global interpolation, we rely on a local interpolation. Suppose we have estimated the continuation values $\widehat{\mathrm{CV}}^l_{t+1}(X_{t+1})$ for all $a^l \in \mathcal{C}_{t+1}$, then the associated first-layer optimal control is given as

$$\hat{\alpha}_{t+1} = \arg\max_{a^l \in \mathcal{C}_{t+1}} \widehat{\mathrm{CV}}^l_{t+1}(X_{t+1}).$$

From there, we can construct the local grid $\mathcal{L}(\hat{\alpha}^j_{t+1}) \subset \mathcal{C}_{t+1}$ of the neighbors of $\hat{\alpha}^j_{t+1}$ in $\mathcal{C}_{t+1}$:

$$\mathcal{L}(\hat{\alpha}_{t+1}) := \{a \in \mathcal{C}_{t+1} : \|\hat{\alpha}_{t+1} - a\|_\infty \leq \delta\},$$

where δ is the mesh size of the discrete grid $\mathcal{C}_{t+1}$. Finally, we interpolate the continuation values of the local grid, $\{\widehat{\mathrm{CV}}^l_{t+1}(X_{t+1})\}_{l \,|\, a^l \in \mathcal{L}(\hat{\alpha}_{t+1})}$, $a^l \in \mathcal{L}(\hat{\alpha}_{t+1})$ into the whole local hypercube, which then provides improved estimates of the optimal control.

Here is a simple example to illustrate the method. Suppose we are given a portfolio of one risk-free asset and one risky asset and we estimate a first discrete optimal control estimate on a 4-step control grid (i.e., $\delta = 1/4$). In this paragraph, we denote $\alpha \in [0, 1]$ as the allocation in the risky asset. Suppose this first-layer discrete optimal control is $\hat{\alpha} = 1/4$. Then, its local grid is $\mathcal{L}(\hat{\alpha} = 1/4) = \{0, 1/4, 1/2\}$. Notice that $\{0\}$ and $\{1/2\}$ belong to the 4-step control grid, which means that we have already estimated the corresponding continuation values. We interpolate these continuation values on $\mathcal{L}(\hat{\alpha} = 1/4)$ to obtain a parametric form of the continuation value, then maximize it over the interval $[0, 1/2]$.

Local interpolation has several advantages:

- *Simplicity.* A simple second order polynomial is a sensible candidate for a parametric interpolation, as the local grid only has a maximum of three points in each dimension, with the central point being the current optimal.
- *Accuracy.* As a local grid contains only few points, it is easier for the interpolation to reach a satisfactory goodness-of-fit.

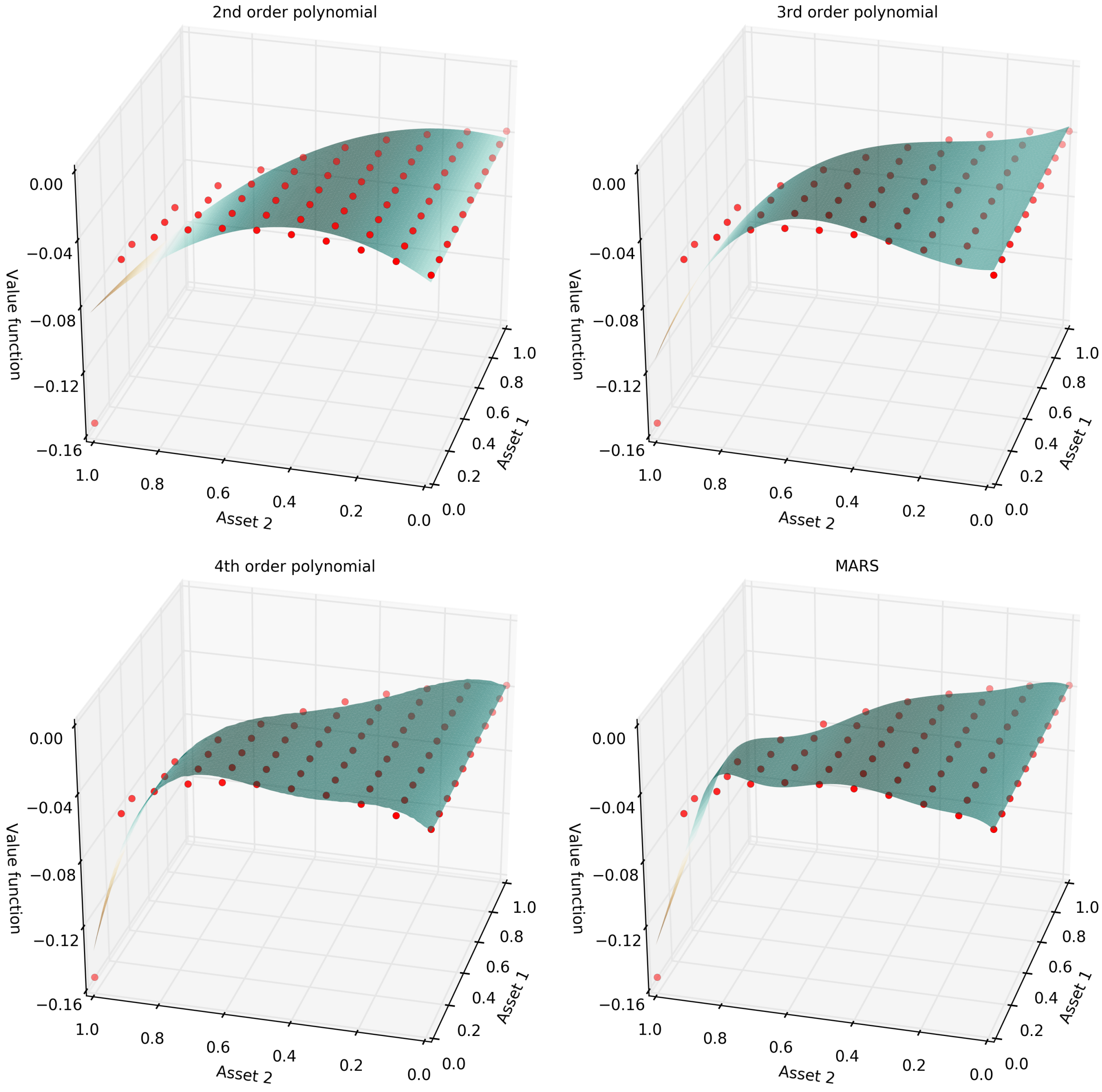

Figure 3.1: Global interpolation

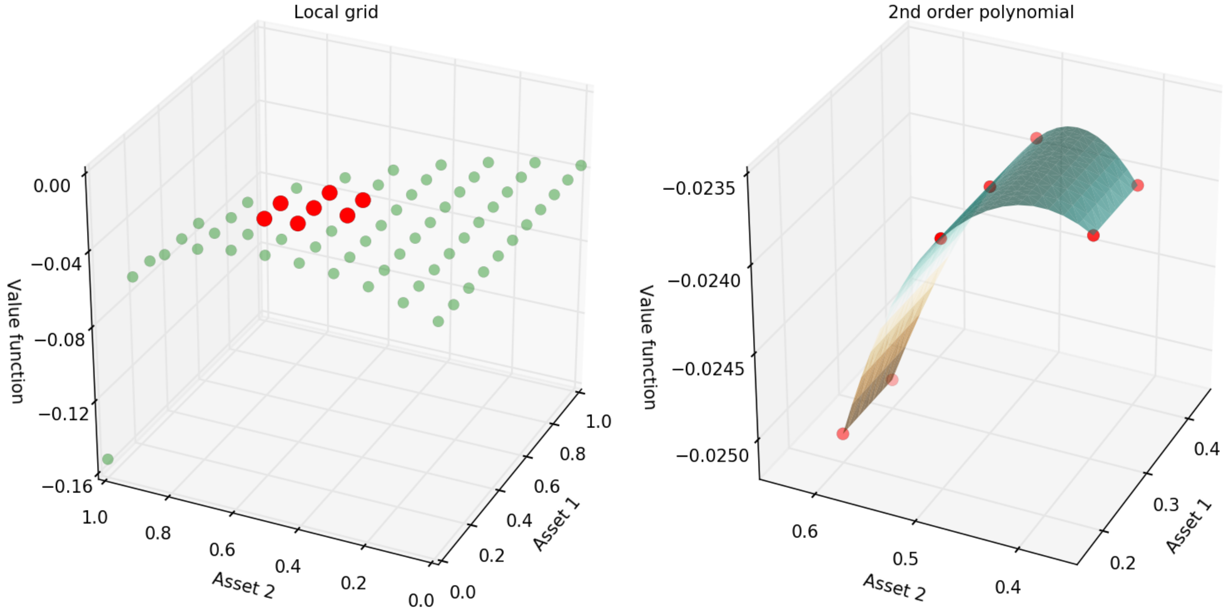

Figure 3.2: Local interpolation

- *Efficiency.* Using a local grid avoids the time-consuming task of searching for a suitable polynomial order of global interpolation. Moreover, computing the new maximum on a local grid is much more efficient.

See Figure 3.2 for an illustration of local interpolation under the same example presented in Figure 3.1.

3.4.2 Adaptive grids to maximize the interpolated value function

The final step is to compute the second-layer continuous optimal control corresponding to the highest interpolated value function. To do so, a natural idea when using second order polynomial is to use quadratic programming. The main problem with this approach is that the interpolated polynomial is not guaranteed to be concave. When convex, the maximum value would be pushed toward the boundary of the local hypercube. One may alleviate this problem by enforcing concavity of the interpolation as a constraint, at the cost of a worse goodness-of-fit of the interpolation.

Instead, we develop an adaptive refinement strategy to estimate the second-layer optimal control. As a matter of fact, adaptive controls has received a lot of attention for solving dynamic economic models; see Krueger and Kubler (2004), Winschel and Kraetzig (2010), Judd, Maliar, Maliar, and Valero (2014) and Brumm, Mikushin, Scheidegger, and Schenk (2015) for examples. All these papers found substantial improvements in the numerical efficiency. Our setting differs from the existing methods by performing adaptive controls on local grids.

Here is a simple illustration of our adaptive control. Suppose we are given a portfolio of a risk-free asset and a risky asset. We denote $\{\mathcal{A}_p\}_{p=0,1,\ldots,P}$ as the sequence of adaptive grids and denote $\{\alpha(p)\}_{p=0,\ldots,P}$ as the sequence of optimal controls associated with these adaptive grids. The initial adaptive control $\alpha(0)$ is set to the optimal decision on the coarse global grid $\mathcal{C}$. Suppose $\mathcal{L}(\alpha(0)) = \{x-h, x, x+h\}$, then the initial adaptive grid is $\mathcal{A}_0 = \{x-h, x-h/2, x, x+h/2, x+h\}$. For every refined adaptive grid, there is a maximum of two new points in each dimension. This hierarchical structure guarantees the minimum number of operations will be used in the algorithm. Below is an example of our adaptive grid refinement strategy starting with a 1/4 control mesh for $\mathcal{C}$:

$$\begin{aligned}
&\text{Suppose } \alpha(0) := \arg\max_{\mathcal{C}} CV = 0.5000, \quad \text{then} \quad \mathcal{A}_0 = \{0.2500, 0.3750, 0.5000, 0.6250, 0.7500\} \\
&\text{Suppose } \alpha(1) := \arg\max_{\mathcal{A}_0} CV = 0.3750, \quad \text{then} \quad \mathcal{A}_1 = \{0.2500, 0.3125, 0.3750, 0.4375, 0.5000\} \\
&\text{Suppose } \alpha(2) := \arg\max_{\mathcal{A}_1} CV = 0.3125, \quad \text{then} \quad \mathcal{A}_2 = \{0.2500, 0.28125, 0.3125, 0.34375, 0.3750\} \\
&\text{Suppose } \alpha(3) := \arg\max_{\mathcal{A}_2} CV = 0.3125, \quad \text{then} \quad \mathcal{A}_3 = \{0.28125, 0.296875, 0.3125, 0.32815, 0.34375\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&\qquad\quad \alpha^* := \alpha(P) = \arg\max_{\mathcal{A}_{P-1}} CV
\end{aligned}$$

One can see that it takes only a few iterations to reach a 1/100 control mesh precision.

4 Numerical experiments

In this section, we test our framework using data on equities, bonds, commodities and currencies. In Subsection 4.1, we numerically compare the Monte Carlo convergence under value function iteration (VFI, Algorithm 4) and policy function iteration (PFI). In Subsection 4.2, we test the accuracy and efficiency under local interpolation and global interpolation. In Subsection 4.3, we test the accuracy and efficiency of different mesh sizes for the control grid. In Subsection 4.4, we show how our method is able to deal with endogenous state variable by testing different instances of transaction cost, liquidity cost and market impact. Finally, we show that our dynamic strategy outperforms the market in Subsection 4.5.

Data and modeling Table 4.1 summarizes the financial instruments considered for risky assets. We calibrate a first order vector autoregressive model to monthly log-returns (i.e., $\log S_t - \log S_{t-1}$) from October 2007 to January 2016[2]. We assume the annual interest rate on the cash account is 4.5%.

[2] These data are obtained from Yahoo Finance.

Table 4.1: Risky Assets

Risky Asset	ETF Name	ETF Ticker
U.S. bond	Vanguard Total Bond Market ETF	BND
U.S. stock	SPDR S&P 500 ETF	SPY
International stock	iShares MSCI EAFE ETF	EFA
Emerging market stock	iShares MSCI Emerging Markets ETF	EEM
Gold	SPDR Gold Shares ETF	GLD
International bond	SPDR Barclays International Treasury Bond ETF	BWX
Silver	iShares Silver Trust ETF	SLV
Crude oil	U.S. Oil ETF	USO
U.S. dollar	PowerShares Deutsche Bank U.S. Dollar Bullish ETF	UUP
Euro	CurrencyShares Euro ETF	FXE
Japanese Yen	CurrencyShares Japanese Yen ETF	FXY
Australian dollar	CurrencyShares Australian dollar ETF	FXA

Certainty equivalent return For all the numerical tests, we rely on the monthly adjusted certainty equivalent return (CER) calculated by

$$\text{CER} = U^{-1}\left(\mathbb{E}\left[U(W_T)\right]\right)^{\frac{1}{T}} - 1 \approx U^{-1}\left(\frac{1}{M}\sum_{m=1}^{M} U(W_T^m)\right)^{\frac{1}{T}} - 1.$$

The magnitude of monthly returns is usually less than one percent, thus we convert the certainty equivalent return into basis points to make our comparison more intuitive.

Transaction cost, liquidity cost and market impact We assume transaction costs of 0.3% proportional to the portfolio turnover. For liquidity cost and market impact, we use a non-linear parametric form.

We model the limit order book based on the Marginal Supply-Demand Curve (MSDC) originally developed in Acerbi and Scandolo (2008). The MSDC describes the asset's market price as a function of its available market orders. Tian, Rood, and Oosterlee (2013) calibrated the MSDC to the European equity market, interpreting a 'square-root' relation for large or medium-cap equities. We use this 'square-root' MSDC to model the instantaneous order-driven market price. This MSDC reads:

$$\text{MSDC}(\Delta\mathbf{q}_t) = \begin{cases} S_\text{A} e^{k\sqrt{|\Delta\mathbf{q}_t|}} & \text{when } \Delta\mathbf{q}_t > 0 \,(\text{position increase}) \\ S_\text{B} e^{-k\sqrt{|\Delta\mathbf{q}_t|}} & \text{when } \Delta\mathbf{q}_t < 0 \,(\text{position reduction}) \end{cases}, \tag{4.1}$$

where S_B denotes the best bid price, S_A denotes the best ask price, and $\Delta\mathbf{q}_t$ denotes the position change of the investor at time t[3] and $k \in \mathbb{R}^+$ is called the liquidity risk factor. By taking integration with respect

[3] We adapted the definition of MSDC to our context of portfolio selection from the point of view of the investor. In particular, the MSDC curve is increasing, and describes how the asset price changes after a transaction of $\Delta\mathbf{q}_t$ shares.

to $\Delta \mathbf{q}_t$, we obtain an explicit parametric form of the liquidity cost generated by the portfolio rebalancing $\Delta \mathbf{q}_t$,

$$\mathrm{LC}(\Delta \mathbf{q}_t) = S_{\mathrm{A}} \left(\eta(\Delta \mathbf{q}_t) - \Delta \mathbf{q}_t \right) \mathbb{1}\{\Delta \mathbf{q}_t > 0\} + S_{\mathrm{B}} \left(\Delta \mathbf{q}_t - \eta(\Delta \mathbf{q}_t) \right) \mathbb{1}\{\Delta \mathbf{q}_t < 0\} ,$$

where

$$\eta(\Delta \mathbf{q}_t) = \frac{2}{k^2} \left(\mathrm{sign}(\Delta \mathbf{q}_t) k \sqrt{|\Delta \mathbf{q}_t|} e^{-\mathrm{sign}(\Delta \mathbf{q}_t) k \sqrt{|\Delta \mathbf{q}_t|}} + e^{-\mathrm{sign}(\Delta \mathbf{q}_t) k \sqrt{|\Delta \mathbf{q}_t|}} - 1 \right) .$$

Following on Tian, Rood, and Oosterlee (2013), we set $k_0 := 8 \times 10^{-6}$ as our base liquidity risk factor and we will discuss the impact of liquidity using multiples of k_0 in Subsection 4.4. The calibrated bid-ask spread $S_{\mathrm{A}} - S_{\mathrm{B}}$ is about 10^{-5}, small enough to have virtually no impact on the optimal portfolio allocation according to our tests. For convenience, we simply set $S_{\mathrm{B}} = S_{\mathrm{A}}$ for the numerical tests reported in this paper.

We set the permanent market impact as proportional to the temporary peak of the MSDC. Moro, Vicente, Moyano, Gerig, Farmer, Vaglica, and Lillo (2009) have found a proportional rate between 0.5 and 0.7, and 2/3 is widely used in the literature, cf. Abergel and Loeper (2013) for example. Thus, we use 2/3 as our base. In Subsection 4.4, we will analyze the sensitivity of the numerical parameters for transaction cost, liquidity risk factor and market impact.

Basis function and regression We use a simple multivariate polynomial basis. Following the first and second points of Remark 3.1, we scale all the exogenous risk factors (in our case the log-returns) by dividing by their unconditional mean. For the endogenous risk factor (the portfolio wealth W), we transform it as $U(W/W_0)$, where W_0 is the initial portfolio wealth and $U(\cdot)$ is the CRRA utility function. These quantities form the predictors used as inputs of our polynomial basis. We choose the polynomial order by observing the plots of the objective function against the regression bases at various intermediate times. The surface shape, close to linear but slightly bent, suggests a polynomial of order two would be sufficient.

Interpolation For the interpolation, we have tested simple least-squares regression, Ridge regression and Lasso regression (cf. Tibshirani (1996) for details). We observe that Ridge regression is the most stable and the most efficient. We have tested different Ridge penalty from 10^{-4} to 10^{-1} and obtained similar results. Thus, we choose to set the Ridge regression penalty to 10^{-2} for the rest of the paper.

4.1 Monte Carlo convergence

In this Subsection, we discuss the numerical convergence of Monte Carlo simulation under VFI (Value Function Iteration) and PFI (Policy Function Iteration). Van Binsbergen and Brandt (2007) claim that PFI is more stable than VFI. The value function is a conditional expectation, which involves error when the conditional expectation is evaluated by projecting realized utilities on the polynomial basis of the state variables. As a consequence, the resulting optimal control will also contain an error, and for a highly non-linear utility function, this error will be very large and will be compounded in multi-period problems. Van Binsbergen and Brandt (2007) numerically show that for low levels of risk-aversion and short investment horizons VFI and PFI are almost equivalent, but for high levels of risk-aversion and/or long investment horizon, PFI is more accurate. The reason for their inaccuracy of implementing VFI is the approximation of a non-linear value function by first order polynomial basis. However, VFI can greatly benefit from a second order polynomial basis for a sufficiently large number of simulation paths.

Garlappi and Skoulakis (2009) argue that the reason for the deterioration of VFI is the high non-linearity of the value function and show that if one transforms the value function into a less non-linear function, for example certainty equivalent, VFI also produces very accurate results. This transformation is based on the separation of endogenous state variables and exogenous state variables, which is however not applicable in general.

In fact, VFI is likely to explode when the value function itself is highly non-linear. Thus, we adjust the value function estimate by truncating the value function to its global maximum. Indeed, the CRRA utility $U(w) = w^{1-\gamma}/(1-\gamma)$ is bounded above by 0 (when $\gamma > 1$), thus the objective in (2.1) becomes

$$V_t = \sup_{\{\alpha_s \in \mathcal{C}_s\}, s \in [t,T)} \mathbb{E}\left[\min\left\{U(W_T), 0\right\} | \mathcal{F}_t\right], \ t \in \mathcal{T}.$$

We call this truncated VFI scheme (or bounded VFI scheme). The variant of Algorithm 4 is where the continuation values $\widehat{\mathrm{CV}}_t^j(x) = \hat{\beta}_t^j \cdot L(x)$ are replaced by $\min\left\{\hat{\beta}_t^j \cdot L(x), 0\right\}$. This idea of truncating the conditional expectation estimates to known bounds is crucial for Monte Carlo convergence, see for example the convergence analyses in Gobet et al. (2005) and Aïd et al. (2014).

Table 4.2 compares the CER for (unbounded) VFI, truncated VFI and PFI under different numbers of Monte Carlo paths. For both VFIs, in particular, we calculate the confidence interval for the true value function following Broadie and Glasserman (2004) and Bouchard and Warin (2012). Our results show that unbounded VFI is not stable, i.e., the algorithm struggles to converge for a highly non-linear CRRA and/or a long time horizon.

There is always a debate between VFI and PFI, where in general VFI is more efficient but PFI is more accurate. Perhaps surprisingly, our results show that the lower bound of the truncated VFI's confidence interval is higher than the estimate of PFI, meaning that truncated VFI, in addition to be more efficient than PFI, is also more accurate than PFI (the higher the lower bound the better). For this reason, we will use the lower bound of the truncated VFI scheme, with $M = 10,000$ paths, as the basis for the rest of the numerical tests.

		VFI			Truncated VFI			PFI		
Period	M	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$
$T = 3\text{M}$	10^3	$(68.6, 78.5)$	$(55.7, 74.2)$	$(45.8, 79.7)$	$(68.6, 78.5)$	$(55.7, 74.2)$	$(44.0, 79.7)$	69.7	56.3	47.1
	10^4	$(66.6, 68.4)$	$(57.6, 61.3)$	$(47.8, 61.1)$	$(66.6, 68.4)$	$(57.6, 61.3)$	$(47.8, 61.0)$	66.4	57.3	47.1
	10^5	$(64.6, 65.1)$	$(56.4, 57.7)$	$(50.7, 53.2)$	$(64.6, 65.1)$	$(56.4, 57.7)$	$(50.7, 53.2)$	64.5	56.5	50.0
	10^6	$(64.7, 64.9)$	$(56.5, 56.8)$	$(50.9, 51.5)$	$(64.7, 64.9)$	$(56.5, 56.8)$	$(50.9, 51.5)$	64.6	56.5	50.3
$T = 6\text{M}$	10^3	$(83.4, 97.0)$	$(-11.9, 394.3)$	$(-98.9, -565.9)$	$(83.4, 97.0)$	$(62.0, 95.5)$	$(50.6, 107.7)$	83.1	64.7	38.5
	10^4	$(81.6, 81.8)$	$(65.0, 72.7)$	$(-7.0, -210.4)$	$(80.8, 82.6)$	$(65.0, 72.7)$	$(50.2, 74.3)$	80.7	64.5	46.7
	10^5	$(79.0, 79.8)$	$(65.7, 67.5)$	$(55.2, 63.0)$	$(79.0, 79.8)$	$(65.7, 67.5)$	$(55.2, 62.8)$	78.9	65.4	54.1
	10^6	$(79.2, 79.5)$	$(65.9, 66.2)$	$(55.5, 58.2)$	$(79.2, 79.5)$	$(65.9, 66.2)$	$(55.5, 58.1)$	79.2	65.9	55.4
$T = 12\text{M}$	10^3	$(88.2, 118.7)$	$(-26.2, -126.2)$	$(-128.5, -195.4)$	$(88.1, 118.5)$	$(66.4, 124.5)$	$(43.7, 135.9)$	85.8	58.9	30.1
	10^4	$(86.7, 90.0)$	$(31.4, 139.2)$	$(-173.1, -238.9)$	$(86.7, 90.0)$	$(67.1, 84.5)$	$(51.0, 99.2)$	86.2	64.6	39.5
	10^5	$(85.7, 86.2)$	$(68.1, 72.1)$	$(-184.4, -308.9)$	$(85.7, 86.2)$	$(68.1, 72.1)$	$(51.4, 85.8)$	85.6	67.4	48.4
	10^6	$(85.8, 85.9)$	$(68.3, 69.1)$	$(35.7, 115.3)$	$(85.8, 85.9)$	$(68.3, 69.1)$	$(54.5, 61.9)$	85.6	68.2	53.9

Table 4.2: This table compares the Monte Carlo convergence of VFI, truncated VFI and PFI. The monthly adjusted certainty equivalent return (in basis points) is reported: a confidence interval for VFI and truncated VFI, and a lower bound estimator for PFI, under different sizes of Monte Carlo sample ($M = 10^3, 10^4, 10^5, 10^6$), different investment horizons ($T =$ 3M, 6M, 12M) and different risk-aversion parameters of the CRRA utility ($\gamma = 5, 10, 15$). A 1/4 control mesh is used to investigate a portfolio of a cash account (risk-free-rate=4.5% p.a.) and two risky assets (BND and SPY), given proportional transaction costs 0.3%, liquidity risk factor $k = 8 \times 10^{-6}$ and proportional market impact ratio 2/3.

4.2 Local interpolation

As explained in subsection 3.4, the optimal portfolio allocation computed on a discrete grid can be improved by local interpolation. In this subsection, we compare local interpolation to global interpolation in terms of accuracy and efficiency. Table 4.3 and Table 4.4 compare local interpolation with different types of global interpolations (polynomial fitting of different orders, and MARS (multivariate adaptive regression splines)). Our results show that local interpolation does improve both accuracy and efficiency.

Mesh	2nd Local	2nd Global	3rd Global	4th Global	MARS Global
1/2	64.9	65.0	64.9	64.8	63.2
1/4	65.0	64.8	65.0	64.9	64.5
1/8	65.1	64.9	64.5	−12.3	65.0
1/16	65.2	64.8	64.9	65.0	65.0
1/32	65.2	64.8	64.7	65.0	64.8

Table 4.3: This table compares the lower bound of the truncated VFI scheme for the monthly adjusted CER (in basis points) using different sizes of control mesh and different interpolation methods. A sample of $M = 10^4$ Monte Carlo paths is used to investigate a CRRA investor with $\gamma = 10$ on a portfolio of a cash account (risk-free-rate=4.5% p.a.) and two risky assets (BND and SPY) over a 6-month horizon ($T = 6\text{M}$), under 0.3% proportional transaction cost, $k = 8 \times 10^{-6}$ liquidity risk factor and 2/3 proportional market impact ratio.

Mesh	2nd Local	2nd Global	3rd Global	4th Global	MARS Global
1/2	$\tau^a = 43$ secs	$1.2 \times \tau^a$	$1.6 \times \tau^a$	$2.6 \times \tau^a$	$3.6 \times \tau^a$
1/4	$\tau^b = 1.5$ mins	$1.4 \times \tau^b$	$1.9 \times \tau^b$	$3.0 \times \tau^b$	$10.5 \times \tau^b$
1/8	$\tau^c = 5.2$ mins	$1.8 \times \tau^c$	$2.3 \times \tau^c$	$7.0 \times \tau^c$	$15.6 \times \tau^c$
1/16	$\tau^d = 24.1$ mins	$2.8 \times \tau^d$	$3.9 \times \tau^d$	$9.3 \times \tau^d$	$22.3 \times \tau^d$
1/32	$\tau^e = 2.9$ hours	$4.1 \times \tau^e$	$7.0 \times \tau^e$	$13.6 \times \tau^e$	$35.4 \times \tau^e$

Table 4.4: This table reports the computational runtime with respect to control dimension of approximating one conditional expectation with a sample of $M = 10^4$ Monte Carlo paths under different sizes of mesh and different interpolation methods. For ease of comparison, the results are reported as multiple of the fastest method (local interpolation).

Remark that the diverging value −12.3 for 4th global interpolation and 1/8 mesh suggests that a high interpolation order is unreliable.

4.3 Mesh size

In this subsection, we test the sensitivity of the accuracy (Table 4.5) and efficiency (Table 4.6) with respect to the mesh size. Table 4.5 shows that a coarse grid (1/2 or 1/4 mesh) is able to produce accurate results, where the loss in CER is only a few basis points compared with a fine grid (1/32 mesh).

		CER			Initial Weights		
Period	Mesh	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$
3M	1/2	66.6	57.5	47.3	0.51, 0.37, 0.12	0.78, 0.22, 0.00	1.00, 0.00, 0.00
	1/4	66.6	57.6	47.8	0.41, 0.31, 0.18	0.65, 0.22, 0.13	0.95, 0.05, 0.00
	1/8	66.4	57.8	48.0	0.42, 0.32, 0.16	0.64, 0.23, 0.13	0.92, 0.08, 0.00
	1/16	66.6	57.8	47.8	0.42, 0.33, 0.15	0.65, 0.24, 0.11	0.92, 0.08, 0.00
	1/32	66.7	57.8	48.0	0.41, 0.32, 0.17	0.64, 0.26, 0.10	0.92, 0.08, 0.00
6M	1/2	80.8	64.9	49.7	0.65, 0.27, 0.08	0.85, 0.15, 0.00	1.00, 0.00, 0.00
	1/4	80.8	65.0	50.2	0.58, 0.31, 0.11	0.77, 0.18, 0.05	1.00, 0.00, 0.00
	1/8	80.6	65.1	50.5	0.57, 0.31, 0.12	0.75, 0.20, 0.05	0.91, 0.09, 0.00
	1/16	80.7	65.2	50.5	0.55, 0.32, 0.13	0.74, 0.20, 0.06	0.92, 0.08, 0.00
	1/32	80.8	65.2	50.5	0.55, 0.32, 0.13	0.71, 0.22, 0.07	0.92, 0.08, 0.00
12M	1/2	86.5	66.7	51.0	0.87, 0.13, 0.00	1.00, 0.00, 0.00	1.00, 0.00, 0.00
	1/4	86.7	67.1	51.0	0.81, 0.19, 0.00	0.90, 0.10, 0.00	0.86, 0.14, 0.00
	1/8	86.5	67.2	51.2	0.82, 0.18, 0.00	0.89, 0.11, 0.00	0.88, 0.12, 0.00
	1/16	86.6	67.4	51.1	0.81, 0.19, 0.00	0.87, 0.13, 0.00	0.88, 0.12, 0.00
	1/32	86.8	67.4	51.1	0.81, 0.19, 0.00	0.87, 0.13, 0.00	0.88, 0.12, 0.00

Table 4.5: This table compares the VFI's lower limit of monthly adjusted CER (in basis points) and its initial optimal decision of using different sizes of control mesh $(1/2, 1/4, 1/8, 1/16, 1/32)$ under different investment horizons $(T = 3\text{M}, 6\text{M}, 12\text{M})$ and different risk-aversion parameters of the CRRA utility $(\gamma = 5, 10, 15)$. A sample of 10^4 Monte Carlo paths $(M = 10^4)$ is used to investigate a portfolio of a cash account (risk-free-rate=4.5% p.a.) and two risky assets (BND and SPY), given proportional transaction costs 0.3%, liquidity risk factor $k = 8 \times 10^{-6}$ and proportional market impact ratio 2/3.

Table 4.6 reports the runtime of one-time-step dynamic programming under different sizes of control mesh. The code is implemented in Python 3.4.3 on a single processor of 2.2 GHz Intel Core i7. 'N\A', short for 'Not Available', indicates the situation where the algorithm requires more than 24 hours to run. We estimate that the runtime is a polynomial function of the dimension N, i.e., $\tau_N = cN^p$, where c and p are constants. For mesh sizes 1/2, 1/4, and 1/8, we use a simple least-squares regression to fit the logarithmic runtime w.r.t. the logarithmic dimension, i.e., $\log \tau_N = \log c + p \log N$ and the estimated polynomial orders are 2.599, 4.347 and 6.451 (see Figure 4.1).

Dimension	Mesh size				
	1/2	1/4	1/8	1/16	1/32
2	19 secs	26 secs	48 secs	1.6 mins	3.2 mins
3	43 secs	1.5 mins	5.2 mins	24.1 mins	2.9 hrs
4	1.4 mins	4.5 mins	29.4 mins	8.6 hrs	N\A
5	2.4 mins	11.8 mins	2.9 hrs	N\A	N\A
6	3.9 mins	37.2 mins	13.8 hrs	N\A	N\A
7	6.1 mins	1.3 hrs	N\A	N\A	N\A
8	9.1 mins	2.2 hrs	N\A	N\A	N\A
9	14.0 mins	3.5 hrs	N\A	N\A	N\A
10	19.6 mins	6.4 hrs	N\A	N\A	N\A

Table 4.6: This table reports the computational runtime w.r.t. control dimension of approximating one-time-step's conditional expectations with a sample of $M = 10^4$ Monte Carlo paths under different sizes of mesh $(1/2, 1/4, 1/8, 1/16, 1/32)$. 'N\A', short for 'Not Available', indicates the situation where the program takes more than 24 hours to run.

4.4 Transaction cost, liquidity cost, market impact

This subsection investigates the sensitivity of the portfolio value with respect to transaction cost, liquidity cost and market impact. Table 4.7 illustrates how our framework can easily handle different transaction costs, liquidity costs and market impacts within a dynamic allocation scheme. For comparison, we benchmark our extended LSMC algorithm with the optimal allocation from an LSMC algorithm that would ignore transaction cost, liquidity cost and market impact (henceforth Blind-LSMC). The massive gain in CER of our LSMC over Blind-LSMC indicates that our algorithm properly captures the effects of intermediate costs, and that taking them into account is vital for reaching one's expected target in a real situation where these costs do occur.

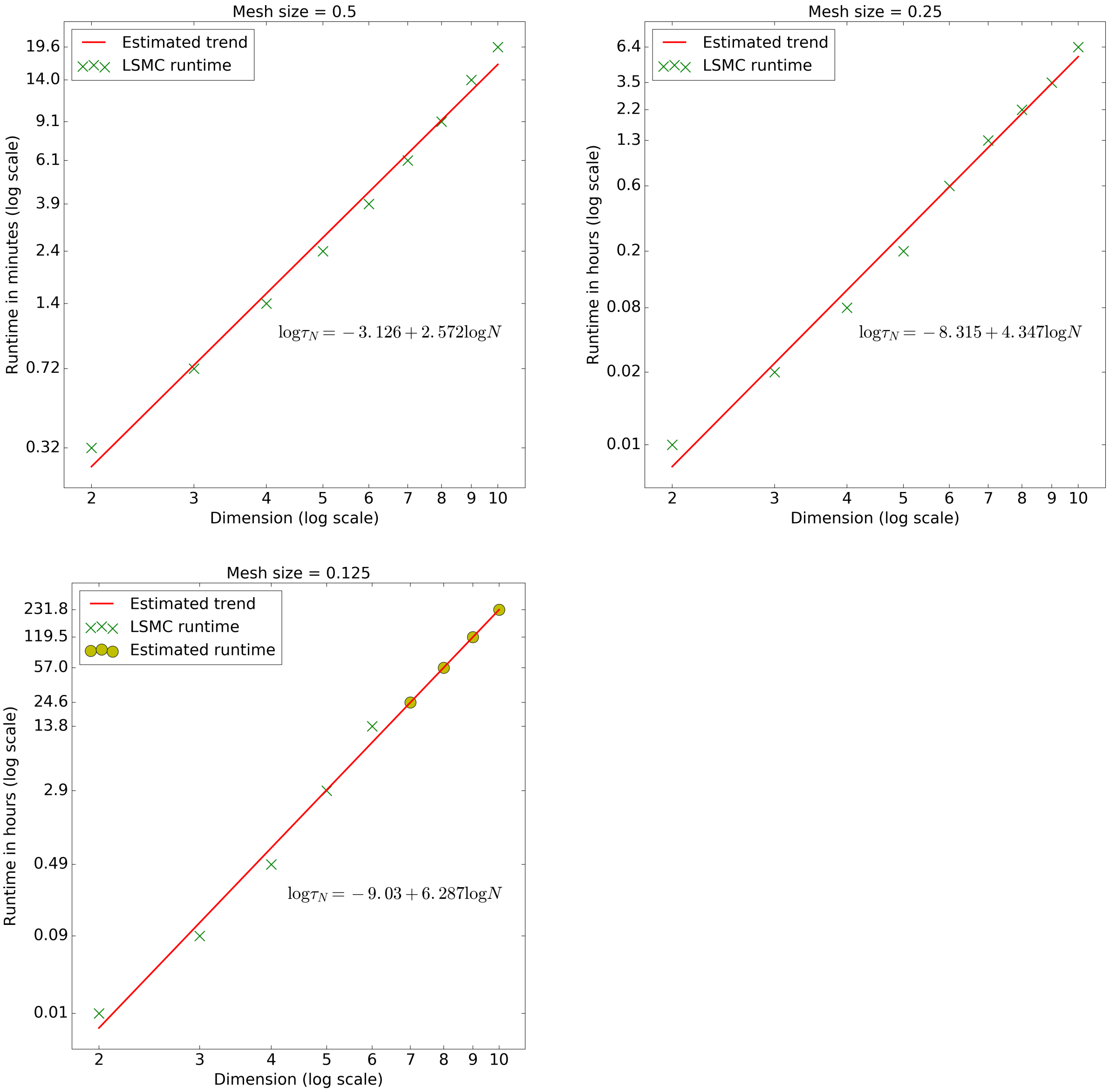

Figure 4.1: This figure shows the runtime of approximating a one-time-step conditional expectation w.r.t. control dimension, with a sample of $M = 10^4$ Monte Carlo paths, under different sizes of mesh (1/2, , 1/4 and 1/8).

		LSMC			Blind-LSMC		
Liquidity Risk Factor	MI	TC=0	TC=0.5%	TC=1%	TC=0	TC=0.5%	TC=1%
$k = 0$		81.5	59.1	49.6	81.5	48.6	14.7
	0.0	77.2	58.2	49.2	76.0	43.0	8.9
$k = 8 \times 10^{-6}$	2/3	78.7	58.7	49.3	78.0	45.0	11.0
	1.0	80.8	59.3	49.5	80.0	47.0	13.1
	0.0	65.9	54.5	46.8	24.5	-9.8	-45.1
$k = 8 \times 10^{-5}$	2/3	73.3	58.3	47.9	45.4	11.3	-23.7
	1.0	95.7	80.1	69.0	65.7	31.9	-2.9
	0.0	61.1	51.1	44.1	-37.1	-72.4	-108.7
$k = 1.6 \times 10^{-4}$	2/3	82.2	70.8	62.7	6.2	-28.9	-64.9
	1.0	126.0	110.4	105.0	46.5	11.8	-23.8

Table 4.7: This table compares the lower bound of the truncated VFI scheme for the monthly adjusted CER (in basis points) under the original LSMC and the Blind-LSMC with different parameters for transaction cost (TC), liquidity cost (liquidity risk factor k) and market impact (MI). A sample of $M = 10^4$ Monte Carlo paths and a 1/4 allocation mesh are used to investigate a CRRA investor with $\gamma = 10$ on a portfolio containing a cash account (risk-free-rate=4.5% p.a.) and two risky assets (BND and SPY) over a 6-month horizon ($T = 6$M)

As discussed in the third point of Remark 3.1, the choice of the dummy randomizations $\tilde{\alpha}$ and $\tilde{W}$ does not affect the theoretical convergence of the algorithm (Kharroubi et al. (2014)); however, for a given fixed sample size M, it may affect the quality of the regressions, especially when the switching costs are high. To prevent any issue, we follow Cong and Oosterlee (2016) by iteratively updating the initial random controls with the new optimal control obtained. We compare the results produced by one full iteration (classical truncated VFI) and two full iterations under different transaction costs, liquidity risk factors and market impact ratios. Table 4.8 shows that the results produced by one iteration and two iterations are almost identical, indicating that the algorithm converges properly with one iteration.

		One Iteration			Two Iterations		
Liquidity Risk Factor	MI	TC=0	TC=0.5%	TC=1%	TC=0	TC=0.5%	TC=1%
$k = 0$		81.5	59.1	49.6	81.5	58.9	49.5
$k = 8 \times 10^{-6}$	0.0	77.2	58.2	49.2	77.0	58.2	49.3
	2/3	78.7	58.7	49.3	78.7	58.6	49.4
	1.0	80.8	59.3	49.5	80.7	59.1	49.5
$k = 8 \times 10^{-5}$	0.0	65.9	54.5	46.8	65.9	54.7	46.9
	2/3	73.3	58.3	47.9	73.1	58.2	47.9
	1.0	95.7	80.1	69.0	95.8	80.0	69.1
$k = 1.6 \times 10^{-4}$	0.0	61.1	51.1	44.1	61.2	51.3	44.3
	2/3	82.2	70.8	62.7	82.2	71.0	62.7
	1.0	126.0	110.4	105.0	125.9	110.4	105.1

Table 4.8: This table compares the lower bound of the truncated VFI scheme for the monthly adjusted CER (in basis points) using one iteration and using two iterations with different parameters for transaction cost (TC), liquidity cost (liquidity risk factor k) and market impact (MI). A sample of $M = 10^4$ Monte Carlo paths and a 1/4 allocation mesh are used to investigate a CRRA investor with $\gamma = 10$ on a portfolio containing a cash account (risk-free-rate=4.5% p.a.) and two risky assets (BND and SPY) over a 6-month horizon ($T = 6\text{M}$)

4.5 Outperforming the market

Table 4.9 reports the performance of five different strategies for a portfolios comprising a cash account and the 12 risky assets detailed in Table 4.1. The performance of each risky asset is also provided for comparison. We compare mean return, volatility of return, five-percent quantile, probability of a negative return and the Sharpe ratio (computed as (Mean − risk-free-rate)/Volatility). The results show that dynamic strategies vastly outperform the myopic strategies (market portfolio and equal weights), as well as each individual asset, indicating the quality of dynamic portfolio allocation, and the capability of our algorithm to deal with realistic high-dimensional portfolios.

5 Conclusion

This paper develops an efficient framework for solving dynamic portfolio selection problems in the presence of transaction cost, liquidity cost and market impact.

This method is based on an extension of the Least-Squares Monte Carlo algorithm (LSMC), with no restrictions with respect to return dynamics, portfolio constraints, intermediate costs or objective function of the investor. We model return dynamics as exogenous state variables and model portfolio weights, price dynamics and portfolio value as endogenous state variables. We use the control randomization technique developed in Kharroubi et al. (2014) to deal with endogenous state variables within the LSMC framework. In particular, this generic framework is able to incorporate any formation of transaction cost, liquidity cost and market impact.

We improve the computation of the optimal allocation for every path using a local interpolation. This technique significantly improves the computational efficiency without losing accuracy. We show that the

	Mean	Volatility	5% quantile	P[return<0]	Sharpe
Portfolios:					
Dynamic Risk-Neutral	**24.92**%	20.99%	−4.69%	9.30%	0.97
Dynamic CRRA $\gamma = 5$	23.15%	16.33%	**−0.48**%	**5.41**%	**1.14**
Dynamic CRRA $\gamma = 10$	16.48%	14.03%	−1.50%	7.27%	0.85
Myopic Market Portfolio	4.69%	**6.39**%	−4.80%	23.85%	0.03
Myopic Equal Weight	−0.38%	12.18%	−17.53%	56.20%	−0.40
Individual Assets:					
SLV	**6.18**%	38.23%	−43.88%	50.02%	**0.04**
SPY	5.10%	21.14%	−25.77%	44.08%	0.03
BND	4.41%	**3.70**%	**−1.59**%	**11.49**%	−0.03
UUP	1.84%	8.43%	−11.39%	42.85%	−0.32
BWX	1.14%	8.08%	−11.58%	45.90%	−0.42
FXA	0.82%	16.68%	−24.04%	51.28%	−0.22
GLD	−0.13%	16.26%	−24.51%	53.56%	−0.28
EEM	−0.91%	29.72%	−41.56%	57.09%	−0.18
EFA	−1.31%	24.63%	−36.23%	57.04%	−0.24
FXY	−1.36%	10.69%	−18.25%	56.95%	−0.53
FXE	−2.80%	10.26%	−18.75%	62.81%	−0.71
USO	−17.55%	41.67%	−66.34%	73.95%	−0.53

Table 4.9: This table reports statistics of the out-of-sample distribution of the lower bound of the truncated VFI scheme, using $M = 10^6$ Monte Carlo paths, for portfolios comprising a cash account (risk-free rate=4.5%) and the 12 risky assets from Table 4.1 (i.e. $N = 13$ with the cash account) over the investment horizon $T = 12$M, using a 1/2 allocation mesh size plus adaptive local interpolation, given 0.3% proportional transaction cost, $k = 8 \times 10^{-6}$ as liquidity risk factor and a proportional market impact of 0.6. Mean return, volatility of return, five percent quantile, probability of a negative return (P[return<0]) and Sharpe ratio (Sharpe) are reported. Five different portfolios are reported: three optimal dynamic portfolios with different investor's risk aversion (risk-neutral, CRRA utility with $\gamma = 5$ and CRRA utility with $\gamma = 10$), the Myopic Market Portfolio, obtained from the highest Sharpe ratio on the efficient frontier, and the Myopic Equal Weight is the strategy that allocates $1/N$ percent in each asset. Each individual asset is also reported. The portfolios and assets are ranked by their expected return.

overall computational runtime of our method scales polynomially with dimension. We numerically show that the Monte Carlo convergence of value function iteration (VFI) is not stable, especially when the objective function is highly non-linear, but the convergence of truncated VFI is much more stable, and even more accurate than policy function iteration (PFI). Moreover, we find that one single simulation loop of the endogenous state variables is sufficient to produce very accurate results.

We study a realistic dynamic portfolio selection problem that includes one risk-free asset and 12 risky assets, and show that the optimal dynamic strategies computed by our method vastly outperform several benchmarks such as the market portfolio, equal weight portfolio and each individual asset of the portfolio. This demonstrates the capability of our method to deal with realistic high-dimensional portfolio allocation problems with intermediate costs, and illustrates the potential of a systematic optimal stochastic control approach for dynamic asset allocation problems.

References

Abergel, F. and G. Loeper (2013, April). Pricing and hedging contingent claims with liquidity costs and market impact. Working Paper, Available at SSRN: http://ssrn.com/abstract=2239498.

Acerbi, C. and G. Scandolo (2008). Liquidity risk theory and coherent measures of risk. *Quantitative Finance 8*, 681–692.

Aïd, R., L. Campi, N. Langrené, and H. Pham (2014). A probabilistic numerical method for optimal multiple switching problem in high dimension. *SIAM Journal on Financial Mathematics 5*(1), 191–231.

Bao, C., Z. Zhu, N. Langrené, and G. Lee (2014). Multi-period dynamic portfolio optimization through least squares learning. In *IAENG Transactions on Engineering Sciences*, pp. 29–42.

Boogert, A. and C. de Jong (2008). Gas storage valuation using a Monte Carlo method. *The Journal of Derivatives 15*(3), 81–98.

Bouchard, B. and X. Warin (2012). Monte-Carlo valorisation of American options: facts and new algorithms to improve existing methods. In P. Carmona, D. M. P., P. Hu, and N. Oudjane (Eds.), *Numerical Methods in Finance*, Volume 12, Heidelberg, pp. 215–255. Springer Proceedings in Mathematics.

Brandt, M., A. Goyal, P. Santa-Clara, and J. Stroud (2005). A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies 18*, 831–873.

Broadie, M. and P. Glasserman (2004). A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance 7*, 35–72.

Brown, D. and J. Smith (2011). Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Management Science 57*(10), 1752–1770.

Brumm, J., D. Mikushin, S. Scheidegger, and O. Schenk (2015). Scalable high-dimensional dynamic stochastic economic modeling. *Journal of Computational Science 11*, 12–25.

Cai, Y., K. L. Judd, G. Thain, and S. J. Wright (2015). Solving dynamic programming problems on a computational grid. *Computational Economics 45*(2), 261–284.

Cai, Y., K. L. Judd, and R. Xu (2013). Numerical solution of dynamic portfolio optimization with transaction costs. NBER Working Paper No. w18709.

Carriere, J. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics 19*(1), 19–30.

Collin-Dufresne, P., K. Daniel, C. C. Moallemi, and M. Sağlam (2014). Dynamic asset allocation with predictable returns and transaction costs. Working Paper.

Cong, F. and C. W. Oosterlee (2016). Multi-period mean-variance portfolio optimization based on Monte Carlo simulation. *Journal of Economic Dynamics and Control 64*, 23–38.

Cui, X., J. Gao, X. Li, and D. Li (2014). Optimal multi-period mean-variance policy under no-shorting constraint. *European Journal of Operational Research 234*(2), 459–468.

Davis, M. and A. Norman (1990). Portfolio selection with transaction costs. *Mathematics of Operations Research 15*(4), 676–713.

DeMiguel, V., X. Mei, and F. J. Nogales (2014). Multiperiod portfolio optimization with many risky assets and general transaction costs. Working Paper, Available at SSRN: http://ssrn.com/abstract=2295345.

Friedman, J, H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics 19*(1), 1–67.

Garlappi, L. and G. Skoulakis (2009). Numerical solutions to dynamic portfolio problems: The case for value function iteration using Taylor approximation. *Computational Economics 33*, 193–207.

Garlappi, L. and G. Skoulakis (2010). Solving consumption and portfolio choice problems: The state variable decomposition method. *Review of Financial Studies 23*, 3346–3400.

Gârleanu, N. and L. H. Pedersen (2013). Dynamic trading with predictable returns and transaction costs. *Journal of Finance 68*(2309-2340).

Gobet, E., J.-P. Lemor, and X. Warin (2005). A regression-based Monte Carlo method to solve Backward Stochastic Differential Equations. *The Annals of Applied Probability 15*(3), 2172–2202.

He, H. and H. Mamaysky (2005). Dynamic trading with price impact. *Journal of Economic Dynamics and Control 29*(5), 891–930.

Judd, K., L., L. Maliar, S. Maliar, and R. Valero (2014, July). Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control 44*, 92–123.

Kharroubi, I., N. Langrené, and H. Pham (2014). A numerical algorithm for fully nonlinear HJB equations: an approach by control randomization. *Monte Carlo Methods and Applications 20*(2), 145–165.

Krueger, D. and F. Kubler (2004, April). Computing equilibrium in OLG models with stochastic production. *Journal of Economic Dynamics and Control 28*(7), 1411–1436.

Lim, A. and P. Wimonkittiwat (2014). Dynamic portfolio selection with market impact costs. *Operations Research Letters 42*(5), 299–306.

Liu, H. (2004). Optimal consumption and investment with transaction costs and multiple risky assets. *Journal of Finance 59*(1), 289–338.

Longstaff, F. and E. Schwartz (2001). Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies 14*(1), 681–692.

Lynch, A., W. and S. Tan (2010). Multiple risky assets, transaction costs and return predictability: Allocation rules and implication for U.S. investors. *Journal of Financial and Quantitative Analysis 45*(4), 1015–1053.

Mei, X. and F. J. Nogales (2015). Portfolio selection with proportional transaction costs and predictability. UC3M Working Paper, ISSN 2387-0303.

Merton, R. (1971). Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory 3*, 373–413.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: the continuous-time case. *Review of Economics and Statistics 51*(3), 247–257.

Moallemi, C. C. and M. Sağlam (2015). Dynamic portfolio choice with linear rebalancing rules. Working Paper, Available at SSRN: http://ssrn.com/abstract=2011605.

Moro, R., J. Vicente, G. Moyano, L., A. Gerig, J. D. Farmer, G. Vaglica, and F. Lillo (2009). Market impact and trading profile of hidden orders in stock markets. *Physical Review E 80*, 066102–1–066102–8.

Mossin, J. (1968). Optimal multiperiod portfolio policies. *Journal of Business 41*(2), 215–229.

Muthuraman, K. and H. Zha (2008). Simulation-based portfolio optimization for large portfolios with transaction costs. *Mathematical Finance 18*(1), 115–134.

Samuelson, P. (1969). Lifetime portfolio selection by dynamic stochastic programming. *Review of Economics and Statistics 51*, 239–46.

Shreve, S., E. and M. Soner, H. (1994). Optimal investment and consumption with transaction costs. *The Annals of Applied Probability 4*(3), 609–692.

Tian, Y., R. Rood, and C. Oosterlee (2013). Efficient portfolio valuation incorporating liquidity risk. *Quantitative Finance 13*(10), 1575–1586.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Tsitsiklis, J. and B. Van Roy (2001). Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks 12*(4), 694–703.

Tsoukalas, G., J. Wang, and K. Giesecke (2015). Dynamic portfolio execution. Working Paper, Available at SSRN: http://ssrn.com/abstract=2089837.

Van Binsbergen, J. H. and M. Brandt (2007). Solving dynamic portfolio choice problems by recursing on optimized portfolio weights or on the value function? *Computational Economics 29*, 355–367.

Vath, V., M. Mnif, and H. Pham (2007). A model of optimal portfolio selection under liquidity risk and price impact. *Finance and Stochastics 11*(1), 51–90.

Winschel, V. and M. Kraetzig (2010). Solving, estimating, and selecting nonlinear dynamic models without the curse of dimensionality. *Econometrica 78*(2), 803–821.