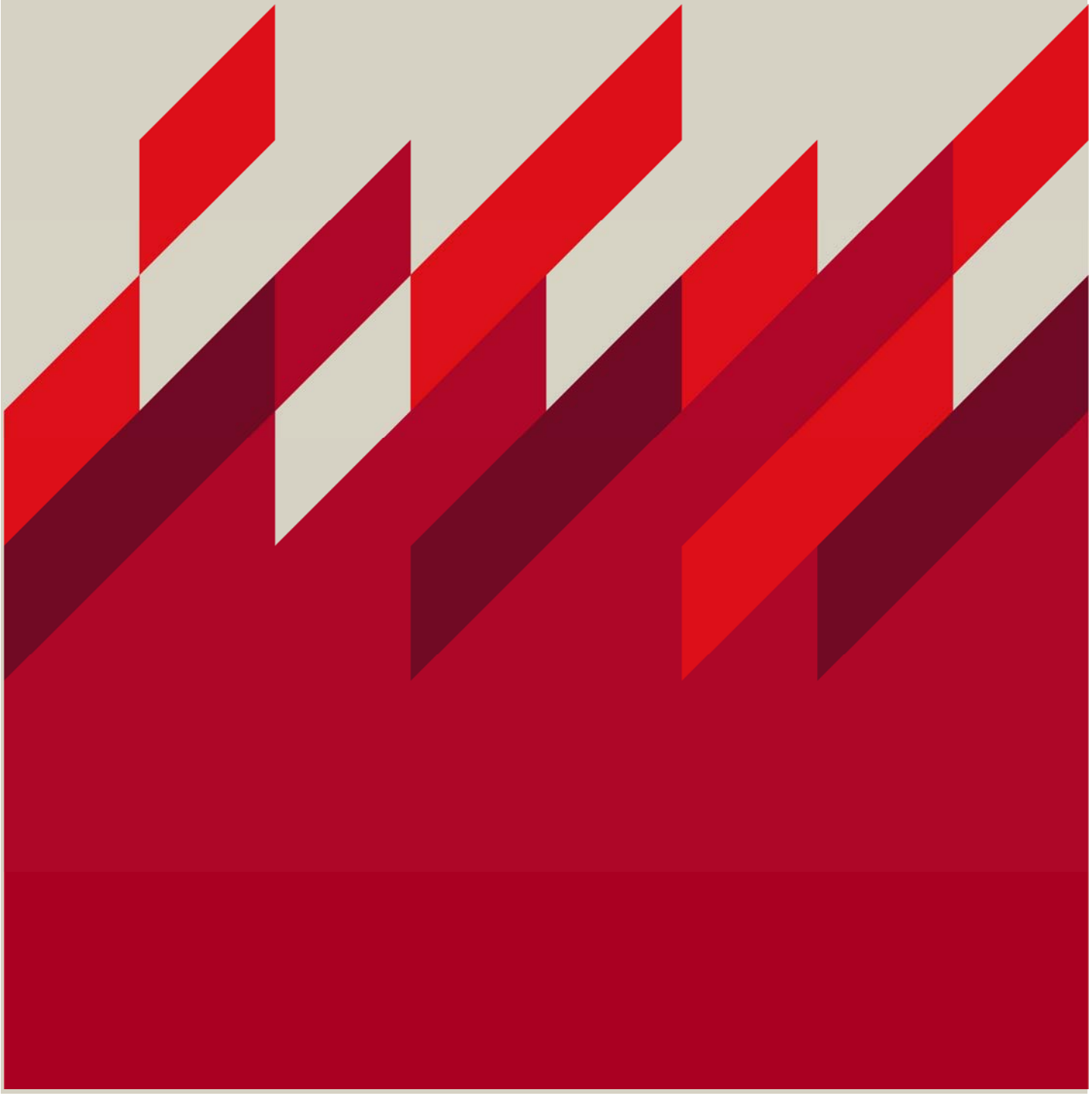
CENTRE FOR
FINANCIAL RISK
Faculty of Business and Economics

MACQUARIE University

Bias-corrected Least-Squares Monte Carlo for utility based optimal stochastic control problems

WORKING PAPER 17-08

Johan G. Andréasson, Pavel V. Shevchenko



Bias-corrected Least-Squares Monte Carlo for utility based optimal stochastic control problems

Johan G. Andréasson*†, Pavel V. Shevchenko ‡

23 December 2017

Abstract

The Least-Squares Monte Carlo method has gained popularity recent years due to its ability to handle multi-dimensional stochastic control problems without restrictions on the state dynamics, including problems with state variables affected by control. However, when applied to stochastic control problems in the multi-period expected utility models, the regression fit tends to contain errors which accumulate over time and typically blow up the numerical solution. In this paper we propose to transform the value function of stochastic control problems to improve the regression fit, and then using either the 'Smearing Estimate' or 'Controlled Heteroskedasticity' to avoid the re-transformation bias. We also present and utilise recent improvements in Least-Squares Monte Carlo algorithms such as control randomisation with policy iteration to avoid regression errors from accumulating. Presented numerical examples demonstrate that our transformation method allows for control of disturbance terms to be handled correctly and leads to an accurate solution. In addition, in the forward simulation stage of the algorithm, we propose a re-sampling of state variables at each time step instead of simulating continuous paths, to improve the exploration of the state space that also appears to be important to obtain a stable and accurate solution for expected utility models.

Keywords: Dynamic programming, Least-Squares Monte Carlo, control randomisation, stochastic control, lifecycle modelling

JEL classification: D91 (Intertemporal Household Choice; Life Cycle Models and Saving), G11 (Portfolio Choice; Investment Decisions), C61 (Optimization Techniques; Programming Models; Dynamic Analysis)

1 Introduction

Stochastic control problems are at the heart of decision making under uncertainty and are critical in many areas such as finance, health, environment, and mining. In stochastic control problems there is always a choice to be made between model complexity, such as

*Corresponding author: 11940057@uts.edu.au

†CSIRO, Australia; School of Mathematical and Physical Sciences, University of Technology, Sydney, Broadway, PO Box 123, NSW 2007, Australia; email: 11940057@uts.edu.au

‡Applied Finance & Actuarial Studies, Macquarie University; email: pavel.shevchenko@mq.edu.au

the number of state/control variables or stochastic factors, and computational cost. Analytical solutions are limited to problems with few stochastic factors with restrictions on the dynamics and dimensions, otherwise one has to revert to numerical methods. Partial Differential Equation methods suffer from the curse of dimensionality, and are practical up to two dimensions only. Numerical direct integration solutions, such as deterministic quadratures also suffer as the number of dimensions increases, but can sometimes handle more if we are willing to accept less precise solutions and longer computation times. Simulation methods are therefore favoured when the number of state variables and stochastic factors increases. The Stochastic Mesh method (Broadie and Glasserman, 2004) overcomes the dimensionality problem, but requires the transition densities of the stochastic factors to be known and suffers as the number of time steps increases. One simulation method that has received increasing interest among researchers is the Least-Squares Monte Carlo method (LSMC), due to its effectiveness in dealing with higher dimensions and because it imposes fewer restrictions on constraints and allows for flexibility in the dynamics of underlying stochastic processes. The idea is based on simulating random paths of the underlying stochastic variables over time and replacing the conditional expectation of the value function in the Bellman backward recursive solution of the stochastic control problem with an empirical least-squares regression estimate. The transition density of the underlying process is not even required to be known in closed form, which offers much more flexibility than alternative approaches.

LSMC was originally developed in Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001). The regression is generally performed on the state variables in order to approximate the value function. In the simpler case, where the state variable is exogenous (i.e. does not depend on control), the simulation and backward in time solution are rather straightforward. When considering endogenous state variables (i.e. affected by the control), however, the simulation becomes more complicated as the future states are affected by the unknown control. The extensions to LSMC that are of particular interest are methods where control variables are included in the regression basis functions[1] (Denault et al., 2013, 2017; Kharroubi et al., 2014). Kharroubi et al. (2014) allow for random control to be simulated and their algorithm (referred to as 'control randomisation') is the only theoretically justified LSMC algorithm with endogenous state variables. This algorithm provides additional benefits of parametric estimate in a feedback form of control, hence no solution grid for control is required (contrary to Denault et al. (2017)).

Naturally there are pitfalls with LSMC as well. Regression errors can accumulate over multiple time periods and can eventually blow up, and as the number of samples increases the algorithm becomes too computationally intensive. There are, however, methods to deal with such problems. The value function can either be based on the 'realised values' (Longstaff and Schwartz, 2001; Denault et al., 2013, 2017; Zhang et al., 2016b) or on the 'regression surface' (Tsitsiklis and Van Roy, 2001; Denault et al., 2017). Although Denault et al. (2017) find little difference between 'realised values' and the 'regression surface', it should be noted that the authors do not apply "true" realised values as the value function is interpolated with respect to the choice of controls. True realised values, which require re-simulation of paths after control is changed in order to calculate the value function, avoid regression errors to accumulate hence appear to be more stable over longer periods (Zhang et al., 2016b) although with the trade-off of longer computation times. In addition, basis functions can often be difficult to find and can be highly problem

[1] A basis function is an element of a particular basis for a function space, where the full function space can be expressed as a linear combination of some chosen functions.

specific. Incorrectly defined basis functions will quickly inflate the regression errors and blow up the solution of a multi-period stochastic control problem. This risk is especially high with regard to objective functions based on utility functions.

LSMC has been applied in many different fields, such as pricing American options (Longstaff and Schwartz, 2001; Tsitsiklis and Van Roy, 2001), mining and real options (Chen et al., 2015), electricity (Denault et al., 2013) and portfolio allocation (Brandt et al., 2005; Garlappi and Skoulakis, 2010; Zhang et al., 2016b) to mention a few. Research with respect to certain issues in LSMC is very diverse, such as heteroskedasticity in the regression (Fabozzi et al., 2017), avoiding re-computing realised paths (Glasserman and Yu, 2004; Nadarajah et al., 2017; Nadarajah and Secomandi, 2017) or managing discontinuity in the basis function (Langrene et al., 2015).

With regard to problems involving utility functions however, there has been very limited research. Approximating a utility function with least-squares regression poses a number of challenges when the agent is risk averse due to very high second derivatives of the utility functions. Regression directly on the value function works only when samples of the state variable are restricted to a sub-domain rather than the full domain. If the full domain is used, then the fit of the regression will be unsatisfactory in parts of the domain due to the high curvature of the value function. This will not work in the case where a control can move the state variable over the full domain, such as a consumption problem where the decision can move the state of wealth from high to zero. It can work if the control has less influence over the change in the state variable, such as in portfolio allocation problems, as the allocation of assets will not result in as significant change in wealth compared to consumption. Even then authors acknowledge problems as volatility or risk aversion increases (Brandt et al., 2005; Denault et al., 2017).

Attempts have been made to resolve this issue, such as utilising Taylor series expansions around the value function (Brandt et al., 2005; Garlappi and Skoulakis, 2010) or by transformations. However, Taylor series expansions require the utility function to be differentiable, often with a minimum of four times and also add effort to compute the derivatives. Garlappi and Skoulakis (2010) apply an inverse utility function to the value function in order to perform the regression on the transformed value function, which does indeed improve the regression fit, but results in a 're-transformation bias' due to Jensen's inequality[2]. Such transformation also ignores any volatility of stochastic variables and underestimates risk. Denault et al. (2017) on the other hand only applies the transformation when interpolating in order to use a more coarse grid for state variables, but the use of a grid voids the purpose of 'true' LSMC. Zhang et al. (2016a) suggest using basis functions with the independent variables transformed using a utility function, but if the domain covers a larger part of the curvature of the utility functions the regression errors will not be homogeneous, hence still result in a bias.

If the regression is carried out on the non-transformed value function, then control of disturbance terms is possible but the regression fit might be questionable. If the regression instead is carried out on the transformed value function, then the fit will most likely be better but optimal control related to disturbance will be unreliable. In this paper two methods are proposed to deal with the re-transformation bias, based on the characteristics of the problem, in order to account for difficulties in using LSMC with utility functions such as control of disturbance and re-transformation bias. In addition, a modification based on re-sampling state variables at each time step is suggested to the probabilistic numerical algorithm that combines dynamic programming with LSMC in

[2]Jensen's inequality states that for a random variable Z and a concave function ψ, $\psi(\mathbb{E}[Z]) \geq \mathbb{E}[\psi(Z)]$.

Kharroubi et al. (2014, 2015) to improve the exploration of the state space, which further helps with the efficiency of the method in the case of expected utility models.

The paper is structured as follows. In Section 2 the basic problem definition is stated, and Section 3 describes methods to avoid re-transformation bias. In Section 4 the LSMC algorithms are explained. The accuracy of the algorithm together with the methods to deal with the re-transformation bias and proposed re-sampling of the state variables at each time step are presented in Section 5. Finally, concluding remarks are in Section 6.

2 Problem definition

Let $t = 0, 1, ..., N$ correspond to equispaced points in time interval $[0, T]$ and $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0\leq t\leq N}, \mathbb{P})$ be a filtered complete probability space where $\mathcal{F}_t$ represents the information available up to time t. We assume that all the processes introduced below are well defined and adapted to $\{\mathcal{F}_t\}_{t\geq 0}$. Let $\pi = (\pi_t)_{t=0,...,N}$ be a control taking value in an action space $\mathcal{A} \subseteq \mathbb{R}^d$, $Z = (Z_t)_{t=1,...,N} \in \mathcal{Z} \subseteq \mathbb{R}^d$ be a disturbance term with realisation z_t and $X^\pi = (X_t^\pi)_{t=0,...,N} \in \mathcal{X} \subseteq \mathbb{R}^d$ be a controlled state variable. We also assume that the evolution of the state variable is described by a transition function

$$X_{t+1}^\pi = \mathcal{T}_t \left(X_t^\pi, \pi_t, Z_{t+1}\right), \tag{1}$$

hence the state of the next period depends on the state of the current period, the control decision and the realisation of the disturbance term.

Now consider the standard discrete dynamic programming problem with the objective to maximise the expected value of total reward function

$$V_0(x) = \sup_\pi \mathbb{E}\left[\beta^N R_N(X_N^\pi) + \sum_{t=0}^{N-1} \beta^t R_t(X_t^\pi, \pi_t) \;\middle|\; X_0^\pi = x; \pi\right], \tag{2}$$

where R_N and R_t are reward functions satisfying the integrability conditions, and β is a time discount factor over a time step. This type of problem can be solved with backward recursion of the Bellman equation

$$\begin{aligned} V_t(x) &= \sup_{\pi_t}\left\{R_t(x, \pi_t) + \mathbb{E}\left[\beta V_{t+1}(X_{t+1}^\pi) \;\middle|\; X_t^\pi = x; \pi_t\right]\right\}, t = N-1, ..., 0, \\ V_N(x) &= R_N(x). \end{aligned} \tag{3}$$

The solution of such a problem is often not possible to find analytically and numerical methods are required. As the number of state variables, stochastic processes, or control variables increases, the numerical solution quickly becomes very computationally expensive. In the standard case, where state is not affected by control, the idea behind utilitising the LSMC method is to approximate the conditional expectation in Equation (3)

$$\Phi_t(X_t) = \mathbb{E}\left[\beta V_{t+1}(X_{t+1})|X_t\right], \tag{4}$$

by a regression scheme with independent variables X_t, and response variable $\beta V_{t+1}(X_{t+1})$. The approximation of the function is then denoted as $\widehat{\Phi}_t$. However, if the state is affected by control, then control randomization is required and the conditional expectation

$$\Phi_t(X_t^\pi, \pi_t) = \mathbb{E}\left[\beta V_{t+1}(X_{t+1}^\pi)|X_t^\pi; \pi_t\right], \tag{5}$$

is estimated by regression on X_t^π and randomised π_t (Kharroubi et al., 2014).

For ease of notation the superscript π on the state variable is now dropped.

3 Transformation of utility

One of the difficulties with LSMC is to select correct basis functions for regression estimate of conditional expectation $\Phi_t(X_t, \pi_t)$. Commonly used basis functions include polynomials such as Chebyshev, Hermite, Laguerre and Legendre (Table 1) or one can use Appell polynomials as the generalized form of many standard polynomials (Novikov and Shiryaev, 2005). Unless the function that is being approximated is convex/concave or smooth, such as piecewise linear, the basis functions might only work locally. Increasingly complex (e.g. higher order) basis functions can be used with increasing number of sample paths to improve the accuracy, but this comes with a computational cost. However, when reward functions $R_t(x, \cdot)$ in (3) are based on the standard utility functions, such as Constant Relative Risk Aversion (CRRA) $U(x) = x^\gamma/\gamma$ or Hyperbolic Absolute Risk Aversion (HARA) $U(x) = (x-a)^\gamma/\gamma$, the basis functions do not produce accurate solution unless constrained locally. A model based on utility functions will have a value function with a similar shape, hence the same problems will arise when fitting the basis functions to $V_{t+1}(X_{t+1})$ as if fitting it to a utility function. A more effective solution is to perform a transformation based on the utility function, and account for the re-transformation bias.

Table 1: Definition of common polynomials used as basis functions up to the nth order.

Polynomial	$f_0(x)$	$f_1(x)$	$f_n(x)$
Chebyshev	1	x	$2xf_{n-1}(x) - f_{n-2}(x)$
Hermite	1	x	$(-1)^n e^{-x^2} \frac{d^n}{dx^n}(e^{-x^2})$
Laguerre	1	$1-x$	$\frac{(2(n-1)+1-x)f_{n-1}(x)-(n-1)f_{n-2}(x)}{n}$
Legendre	1	x	$\frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)n$

Define a utility function $U : \mathbb{R} \to \mathbb{R}$ to be an increasing, monotonic and concave function. Consider a stochastic control problem (3) where the reward is based on this utility function. Then a value function $V_t(x)$ equals the utility function at time T and will have a similar shape for $t < N$. Such a function is difficult to fit with linear regression due to the extreme curvature of common utility functions. For example, consider a CRRA utility function $U(x) = x^\gamma/\gamma$ with a risk aversion parameter $\gamma < 0$. The first problem arises when fitting the regression to low values of x, since when $x \to 0$ then $U(x) \to -\infty$ and therefore no intercept exists. This could be avoided by using fractional polynomials in the basis functions, which are polynomials with fractional exponents. These polynomials (or independent variables with negative exponents) tend to approximate the utility function shape better. Unfortunately, fitting such a model is non-trivial when the utility function is not CRRA or the utility function exerts any kind of piecewise behaviour in relation to the state variables. If a transformation is applied to either decrease the non-linearity in the utility function or to deal with non-normality and heteroskedastic residuals, then Jensen's inequality results in an incorrect projection when the regression is transformed back since $\mathbb{E}[U(\cdot)] \leq U(\mathbb{E}[\cdot])$ due to U being concave. The solution will therefore be biased. In addition to this, any control variables that relate to disturbance terms (such as allocation between risky and risk-free assets in wealth allocation problems) will be biased because risk is underestimated.

To improve the approximation of the value function with the least-squares regression we propose that the regression is performed on the *transformed* value function and

adjusted for the inverse transformation bias with a bias correction function to account for Jensens's inequality. The value function transformed using the inverse of the utility function will have less non-linearity and will allow for an intercept, hence it will have better fit with linear regression (although non-linear independent variables might still be required). Specifically, we proceed as follows.

Define a transformation $H^{-1} : \mathbb{R} \to \mathbb{R}$ and the inverse ('re-transformation') $H : \mathbb{R} \to \mathbb{R}$ such that $H^{-1}(H(x)) = x$. It is implied that state variables still depend on control. Let $\mathbf{L}(X_t, \pi_t)$ be a vector of basis functions and $\mathbf{\Lambda}_t$ the corresponding regression coefficients vector, such that

$$\mathbb{E}\left[H^{-1}(\beta V_{t+1}(X_{t+1}))|X_t; \pi_t\right] = \mathbf{\Lambda}_t' \mathbf{L}(X_t, \pi_t). \tag{6}$$

If M independent Markovian paths of state and control variables are simulated, one can consider the ordinary linear regression

$$\begin{aligned} &H^{-1}(\beta V_{t+1}(X_{t+1}^m)) = \mathbf{\Lambda}_t' \mathbf{L}(X_t^m, \pi_t) + \epsilon_t^m, \\ &\epsilon_t^m \overset{iid}{\sim} F_t(\cdot), \;\; \mathbb{E}[\epsilon_t^m] = 0, \;\; \text{var}[\epsilon_t^m] = \sigma_t^2, \;\; m = 1, ..., M \end{aligned} \tag{7}$$

to estimate the regression coefficients as

$$\widehat{\mathbf{\Lambda}}_t = \arg\min_{\mathbf{\Lambda}_t} \sum_{m=1}^{M} \left[H^{-1}(\beta V_{t+1}(X_{t+1}^m)) - \mathbf{\Lambda}_t' \mathbf{L}(X_t^m, \pi_t^m)\right]^2. \tag{8}$$

It is well known that the estimator $\widehat{\mathbf{\Lambda}}_t$ is the best linear unbiased estimator which is also consistent and asymptotically normally distributed. If the disturbances ϵ_t^m are normally distributed, then this estimator is the maximum likelihood estimator and asymptotically efficient. Moreover, if disturbances ϵ_t^m are heteroscedastic (have different variance), this estimator remains unbiased, consistent, and asymptotically normally distributed but no longer efficient; see for example (Greene, 2008, chapter 8).

Our objective is to estimate $\Phi_t(X_t^\pi, \pi_t) = \mathbb{E}\left[\beta V_{t+1}(X_{t+1}^\pi)|X_t^\pi; \pi_t\right]$ that can be expressed as

$$\Phi_t(X_t, \pi_t) := H^B(\mathbf{\Lambda}_t' \mathbf{L}(X_t, \pi_t)) = \int H(\mathbf{\Lambda}_t' \mathbf{L}(X_t, \pi_t) + \epsilon_t) dF_t(\epsilon_t), \tag{9}$$

where $F_t(\epsilon_t)$ is the distribution of disturbance term ϵ_t. Obviously, in general, naive estimation

$$\widehat{H}^B(\widehat{\mathbf{\Lambda}}_t' \mathbf{L}(X_t, \pi_t)) = H(\widehat{\mathbf{\Lambda}}_t' \mathbf{L}(X_t, \pi_t)) \tag{10}$$

will be neither unbiased nor consistent (even if we know the true parameters $\mathbf{\Lambda}_t$) unless the transformation is linear.

If a specific distribution is assumed for ϵ_t, then the integration in (9) can be performed (in closed form for some cases). Otherwise, the empirical distribution of residuals

$$\widehat{\epsilon}_t^m = H^{-1}(\beta V_{t+1}(X_{t+1}^m)) - \widehat{\mathbf{\Lambda}}_t' \mathbf{L}(X_t^m, \pi_t^m), \tag{11}$$

can be used to perform the required integration as proposed in (Duan, 1983) leading to the following estimate.

Smearing Estimate:

$$\begin{aligned} \widehat{H}^B(\widehat{\mathbf{\Lambda}}_t' \mathbf{L}(X_t, \pi_t)) &= \int H(\widehat{\mathbf{\Lambda}}_t' \mathbf{L}(X_t, \pi_t) + \epsilon_t) d\widehat{F}_M(\epsilon_t) \\ &= \frac{1}{M} \sum_{m=1}^{M} H(\widehat{\mathbf{\Lambda}}_t' \mathbf{L}(X_t, \pi_t) + \widehat{\epsilon}_t^m), \end{aligned} \tag{12}$$

where $\widehat{F}_M(\epsilon_t)$ is the empirical distribution function of the estimated residuals (see Appendix A for details).

If heteroskedasticity is present in the regression with respect to state and control variables, a method that accounts for the heteroskedasticity is required. In this case the conditional variance can be modelled as a function of covariates,

$$\text{var}[\epsilon_t|X_t, \pi_t] = [\Omega(\boldsymbol{\mathcal{L}}_t'\mathbf{C}(X_t, \pi_t))]^2, \tag{13}$$

where $\Omega(\cdot)$ is some positive function, $\boldsymbol{\mathcal{L}}_t$ is the vector of coefficients and $\mathbf{C}(X_t, \pi_t)$ is a vector of basis functions. There are various standard ways to find estimates $\widehat{\boldsymbol{\mathcal{L}}}_t$, the one we use in this paper is based on squared residuals of the ordinary least squares method as outlined in Appendix B. Then, one can use the Smearing Estimate with Controlled Heteroskedasticity proposed in (Zhou et al., 2008) and defined as follows.

Smearing Estimate with Controlled Heteroskedasticity:

$$\widehat{H}^B(\widehat{\boldsymbol{\Lambda}}_t'\mathbf{L}(X_t, \pi_t)) = \frac{1}{M}\sum_{m=1}^{M} H\left(\widehat{\boldsymbol{\Lambda}}_t'\mathbf{L}(X_t, \pi_t) + \Omega(\widehat{\boldsymbol{\mathcal{L}}}_t'\mathbf{C}(X_t, \pi_t))\frac{\widehat{\epsilon}_t^m}{\Omega(\widehat{\boldsymbol{\mathcal{L}}}_t'\mathbf{C}(X_t^m, \pi_t^m))}\right). \tag{14}$$

Here, it is also common to replace $\widehat{\boldsymbol{\Lambda}}_t$ with the weighted least squares estimator that can be found after estimation of $\Omega(\cdot)$, see for example two-step procedure in (Greene, 2008, chapter 8).

It should be noted that an alternative would be to use Generalised Linear Models, where no transformation of the value function is required and which allow for heteroskedasticity through a link function. However, Generalised Linear Models are reported to be quite imprecise when the error distribution assumptions are inaccurate (Baser, 2007), or if the distribution family is misspecified. For a more flexible approach with fewer restrictions we prefer to use Smearing Estimate if no control of disturbance terms is required, and Smearing Estimate with Controlled Heteroskedasticity if control variables are related to disturbance terms in the model. These methods also have the additional advantage that the utility function does not need to be differentiable or continuous, as long as a transformation that roughly represents the shape of the value function can be found.

4 LSMC algorithm

In this section we describe the LSMC algorithms for the exogenous state and the endogenous state with control randomisation. In addition, we use an example with Bermudan options to benchmark the re-transformation method with bias correction against the standard LSMC method.

4.1 Basic algorithm with exogenous state

The basic exogenous state LSMC commonly used in research literature, presented below in Algorithm 1, is based on two parts: a forward simulation and a backward solution with optimal control. First, random state paths $X_t^m, m = 1, ..., M$, are generated which are affected by random disturbances, following the state evolution in Equation (1) for $t = 0, ..., N$. The problem is then solved in a dynamic programming fashion, where the

reward function is first evaluated at time $t = N$. Then, starting at $t = N - 1$ we find the optimal control for each t by regressing the value function at $t + 1$ on the state variables at t. Once a decision has been made, the value function is updated with the outcome and calculations are repeated for $t - 1$ until we find value function at 0.

Algorithm 1 LSMC for exogenous state

[Forward simulation]
for $t = 0$ **to** N **do**
  for $m = 1$ **to** M **do**
    [Simulate random path]
    if $t = 0$ **then**
      $X_t^m := S_0$
    else
      $X_t^m := \mathcal{T}_t(X_{t-1}^m, z_t)$
    end if
  end for
end for

[Backward solution]
for $t = N$ **to** 0 **do**
  if $t = N$ **then**
    $\widehat{V}_t(\boldsymbol{X}_t) := R_N(\boldsymbol{X}_t)$
  else if $t < N$ **then**
    [Regress transformed value function on state variables]
    $\widehat{\boldsymbol{\Lambda}}_t := \arg\min_{\boldsymbol{\Lambda}_t} \sum_{m=1}^{M} \left[\boldsymbol{\Lambda}_t' \mathbf{L}(X_t^m) - H^{-1}(\beta \widehat{V}_{t+1}(X_{t+1}^m))\right]^2$
    Find bias corrected transformation $H^B(\widehat{\boldsymbol{\Lambda}}_t' \mathbf{L}(X_t))$
    [Approximate conditional expectation]
    $\widehat{\Phi}_t(X_t) := H^B(\widehat{\boldsymbol{\Lambda}}_t' \mathbf{L}(X_t))$
    for $m = 1$ **to** M **do**
      [Optimal control]
      $\pi_t^*(X_t^m) := \arg\sup_{\pi_t \in A} \left\{R_t(X_t^m, \pi_t) + \widehat{\Phi}_t(X_t^m)\right\}$
      $\widehat{V}_t(X_t^m) := R_t(X_t^m, \pi_t^*(X_t^m)) + \beta \widehat{V}_{t+1}(X_{t+1}^m)$
    end for
  end if
end for

To illustrate the LSMC algorithm, we start with a basic exogenous state version of LSMC applied to pricing standard Bermudan options in the same manner as in Longstaff and Schwartz (2001). These options can only be exercised at pre-specified dates and at maturity. Although an option that can be exercised prior to maturity does not have a truly exogenous state (as the state changes if the option is exercised), it can still be written in such a way that it can be presented in the current framework and the exogenous state algorithm can be used. Since Algorithm 1 shows the general algorithm for the exogenous state, some minor changes are required in order to use it for Bermudan options and will be discussed further down.

Since utility functions are not used in the pricing, no transformation of the value function is required (hence the value function estimate will not suffer from a re-transformation bias other than possible bias from regression errors). Consider a put option as in the original paper (Longstaff and Schwartz, 2001). Let $t = 0, 1, ..., N$ correspond to the pre-specified equidistant exercise dates obtained by dividing the option maturity T into N steps of length $\delta t = T/N$. The option underlying asset price S_t evolves (under the

so-called risk-neutral process appropriate for valuation of option fair value) as

$$S_{t+1} = S_t e^{(r-\sigma^2/2)\delta t + \sigma Z_t \sqrt{\delta t}}, \quad Z_t \overset{i.i.d}{\sim} \mathcal{N}(0,1), \tag{15}$$

where r is risk free interest rate, σ is volatility and $\mathcal{N}(0,1)$ is the standard normal distribution. The discounting factor for each time period is $\beta = e^{-r\delta t}$. The control variable takes on two values, $\pi_t \in \{0,1\}$, which represent continuation if $\pi_t = 0$ or immediate exercising if $\pi_t = 1$. The state variable consists of the current asset price and an absorbing state, $X_t = \{S_t, \Delta\}$, where Δ indicates that the option has already been exercised. The transition probabilities for the absorbing state are

$$\Pr[X_{t+1} = \Delta | X_t = \Delta] = \Pr[X_{t+1} = \Delta | X_t = S_t, \pi_t = 1] = 1, \tag{16}$$

while the transition probability $\Pr[X_{t+1} \in dS_{t+1} | X_t = S_t, \pi_t = 0]$ corresponds to the process for the asset price. Any other transitions cannot occur, such as moving to the absorbing state if the option is not exercised, hence the remaining transition probabilities are zero.

The terminal reward depends on the moneyness of the option at expiration assuming it has not been exercised,

$$R_N(X_N) = \begin{cases} \max(0, K - X_N), & \text{if } X_N \neq \Delta, \\ 0, & \text{if } X_N = \Delta, \end{cases} \tag{17}$$

where K is the option strike price. The reward function at time t

$$R_t(X_t, \pi_t) = \begin{cases} \max(0, K - X_t), & \text{if } X_t \neq \Delta \text{ and } \pi_t = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

only provides reward (a payoff) if the option has not been exercised earlier and the decision is to exercise the option immediately. The solution of the problem starts by evaluating the payoff at time N. The cash flow at this point is stored as $V_N(X_N)$ and at each previous time the decision to exercise the option or wait is determined by comparing the immediate payoff $\max(0, K - X_t)$ with the continuation value $\mathbb{E}[e^{-r\delta t} V_{t+1}(X_{t+1}) | X_t]$. The continuation value is the discounted expected value if the option is not exercised at time t. The estimation of the continuation value is done by regressing the realised cash flows of $V_{t+1}(X_{t+1})$ at time $t+1$ discounted to t (if not exercised) on a vector $\mathbf{L}(X_t)$ of basis functions of the state variable

$$\mathbb{E}[e^{-r\delta t} V_{t+1}(X_{t+1}) | X_t] = \mathbf{\Lambda}_t' \mathbf{L}(X_t), \tag{19}$$

in order to approximate the conditional expectation $\Phi_t(X_t) = \mathbb{E}\left[\beta V_{t+1}(X_{t+1}) | X_t\right]$. Note that the state variable, if not exercised, is exogenous and not affected by any control, hence the transition function in Equation (1) is simplified and depends only on the previous state and the outcome of the disturbance term(s). The optimal control can then be written as

$$\pi_t^*(X_t) := \arg\max_{\pi_t} \left\{ R_t(X_t, \pi_t) + (1 - \pi_t) \Phi_t(X_t) \right\}, \tag{20}$$

since if the option is now exercised the continuation value will not be received, which is reflected by $(1 - \pi_t)$ in front of the conditional expectation approximation, and is the same as all future rewards are zero as the state will transition to the absorbing state

Δ. Equation (20) therefore replaces line 18 in Algorithm 1. The effect of the decision is recorded in the realised value of

$$\widehat{V}_t(X_t) = R_t(X_t, \pi_t^*(X_t)) + (1 - \pi_t^*(X_t))e^{-r\delta t}\widehat{V}_{t+1}(X_{t+1}), \tag{21}$$

which replaces line 19. If the option is exercised, the realised value equals the reward for the current period, and if it is not exercised it equals the present value of future rewards. The full objective function for the Bermudan option problem, which originally is an optimal stopping problem, then leads to the same optimal stochastic control problem as Equation (2).

As a numerical example, consider the Bermudan option when $S_0 = 36$, $K = 40$, $r = 0.06$, $\sigma = 0.2$, $T = 1$ and $N = 12$ (for results, see Table 2). The basis functions are based on ordinary polynomials up to the fourth order of the state variable, and the price is benchmarked against the Binomial Tree method. First, the problem is solved using the standard LSMC (column $V^{(0)}$). By using a log transformation of the value function, i.e. $H(x) = e^x$, the heteroskedasticity in the residual errors is reduced and we get a better regression estimation, although the bias still remains when re-transformed (column $V^{(1)}$ where naive estimation (10) is used). Finally, we correct for the bias after re-transformation using Smearing Estimate (12). This log transformation with the bias correction results in a more accurate price even with fewer sample paths (column $V^{(2)}$).

Table 2: Price and standard error of Bermudan option estimated using standard LSMC ($V^{(0)}$), LSMC with log transformation of the value function without bias correction ($V^{(1)}$) and LSMC with log transformation of value function and bias correction ($V^{(2)}$) using Smearing Estimate. The results are based on M sample paths, 20 independent repetitions (iterations), and the basis functions are ordinary polynomials up to the 4th order. The 'exact' price obtained by the finite difference method is \$4.3862.

M	$V^{(0)}$	$V^{(1)}$	$V^{(2)}$
1,000	4.4984 (0.032)	4.4336 (0.038)	4.4054 (0.039)
10,000	4.4616 (0.007)	4.4161 (0.007)	4.3962 (0.008)
100,000	4.4457 (0.003)	4.4048 (0.004)	4.3857 (0.004)

4.2 Endogenous state and random control

Algorithm 1 presented in previous section is the very basic case where optimal decisions do not really affect the evolution of the state variable, with the exception of reaching the absorbing (exercised) state. To extend it to the case with an endogenous state, we adopt the discretised version of the control randomisation technique and LSMC algorithm with realised values from Kharroubi et al. (2014), which is the only theoretically justified LSMC algorithm with endogenous state variables. The algorithm is also based on forward simulation and backward solution with optimal control, with the main difference that random state paths in the forward simulation are affected by both a random control and random disturbance. The regression to estimate the conditional expectation then includes the random control in the basis functions. Kharroubi et al. (2014) present two alternative versions of the control randomisation algorithm: the one that uses the regression surface to update the value function,

$$\widehat{V}_t(X_t) = R_t(X_t, \pi_t^*(X_t)) + \widehat{\Phi}_t(X_t, \pi_t^*(X_t)), \tag{22}$$

and another one that uses the realised value function,

$$\widehat{V}_t(X_t) = R_t(X_t, \pi_t^*(X_t)) + \beta \widehat{V}_{t+1}(X_{t+1}). \tag{23}$$

The difference is that the first method is a value function iteration (VFI), while the second is a policy function iteration (PFI). The PFI requires a recalculation of the sample paths for $t+1$ to T after each iteration backwards, as the optimal control affects the future state variables hence changes the simulated paths. This effect is already estimated in the VFI version with $\Phi_t(X_t, \pi_t)$, hence no recalculation is necessary. Longstaff and Schwartz (2001) argue in favour of realised value function (23), while Tsitsiklis and Van Roy (2001) use the regression surface (22). Denault et al. (2017) notice no difference between the two, but only use a very local regression which does not include the endogenous state variable. Due to the recalculation requirement with PFI the computational complexity grows quadratically compared with linear growth for VFI. However, PFI tends to accumulate much less regression errors over time and from experience this method is much more suitable for problems prone to regression errors when the number of time periods increases.

4.2.1 Forward simulation

The forward simulation in the case of endogenous state with control randomisation is more delicate than in the case of exogenous state variables, and deserves a special discussion below. The objective of the forward simulation is to generate enough information (sample paths) such that the conditional expectation of state and control variables can be estimated.

The algorithm is intended to be used with a known starting state x_0 where each simulated path X_t^m, $t = 0, ..., N$, is subject to the random control $\widetilde{\pi}_t^m$ and disturbances in the diffusion process, given by the evolution in Equation (1). However, if the optimal control π_t^* tends to take on either a high or low value in the control domain $\mathcal{A}$ this will cause a problem. On the one hand, if random control is uniformly distributed then the simulated paths will end up in a sub-domain very different from the one if optimal control is used at each time step. On the other hand, if the typical range of the optimal control is known, a distribution to reflect this could be used to better simulate the randomised control where more paths would end up in the same sub-domain as if the optimal control was applied. While the former would lead to difficulties in the regression estimation due to lack of state sample paths in sections of the state domain, the latter would do the same due to lack of control samples outside the expected range (hence lack samples in the full control domain $\mathcal{A}$).

A better approach would be to simulate state and control for the full domain to ensure a better fit for the regression. This can be achieved by using a random state each time t that is independent of decisions and disturbance for $0, ..., t-1$, as the simulated paths are recalculated after each time period anyway. Denote $\widetilde{X}_t$ as the state variable implied by transition function $\mathcal{T}_{t-1}(X_{t-1}, \pi_{t-1}, z_t)$, where X_t is an independent random sample from the state variable domain at each time step. If the state variable X_t would be simulated using the transition function as a path for the full period $t = 0, ..., T$, then $\widetilde{X}_{t+1} = X_{t+1}$ would hold. The logical steps of this procedure are summarized by Algorithm 2 below. Each X_t is simulated independently of the previous state, which allows the algorithm to spread samples over the full domain each time period to avoid the pitfalls described. This will explore the space better and the reason for this will become apparent in Algorithm

3. In the algorithm, *Rand* corresponds to random sampling from some distribution that could be designed for the specific problem. These can be e.g. uniform distributions for X_t and π_t, while distribution for z_t is model specific.

Algorithm 2 Forward simulation

for $t = 0$ **to** $N - 1$ **do**
  for $m = 1$ **to** M **do**
    [Simulate random samples]
    $X_t^m := Rand \in \mathcal{X}$ ▷ State
    $\widetilde{\pi}_t^m := Rand \in \mathcal{A}$ ▷ Control
    $z_{t+1}^m := Rand \in \mathcal{Z}$ ▷ Disturbance
    [Compute the state variable after control]
    $\widetilde{X}_{t+1}^m := \mathcal{T}_t(X_t^m, \widetilde{\pi}_t^m, z_{t+1}^m)$ ▷ Evolution of state
  end for
end for

4.2.2 Backward optimisation

After the forward simulation step is completed, the problem is now solved with the backward induction (Algorithm 3), similarly to Algorithm 1. The conditional expectation $\Phi_t(\cdot)$ of the value function at time $t + 1$ is estimated with a regression function, and the optimal decision is found by maximising the sum of the reward function and the approximated value function. Once the optimal decision has been found, the sample paths $t + 1, ..., N$ are recalculated with the new optimal control and the corresponding value function for the realisation of the paths is stored to be used in the next iteration.

Algorithm 3 Backward solution (Realised value)

for $t = N$ **to** 0 **do**
  if $t = N$ **then**
    $\widehat{V}_t(\widetilde{\boldsymbol{X}}_t) := R_N(\widetilde{\boldsymbol{X}}_t)$
  else if $t < N$ **then**
    [Regression of transformed value function]
    $\widehat{\boldsymbol{\Lambda}}_t := \arg\min_{\boldsymbol{\Lambda}_t} \sum_{m=1}^{M} \left[\boldsymbol{\Lambda}_t' \mathbf{L}(X_t^m, \widetilde{\pi}_t) - H^{-1}(\beta \widehat{V}_{t+1}(\widetilde{X}_{t+1}^m))\right]^2$
    Find bias corrected transformation $H^B(\widehat{\boldsymbol{\Lambda}}_t' \mathbf{L}(X_t, \widetilde{\pi}_t))$
    [Approximate conditional expectation]
    $\widehat{\Phi}_t(X_t, \widetilde{\pi}_t) := H^B(\widehat{\boldsymbol{\Lambda}}_t' \mathbf{L}(X_t, \widetilde{\pi}_t))$
    for $m = 1$ **to** M **do**
      $\widehat{X}_t^m := \widetilde{X}_t^m$
      [Optimal control]
      $\pi_t^*(\widehat{X}_t^m) := \arg\sup_{\pi_t \in A} \left\{ R_t(\widehat{X}_t^m, \pi_t) + \widehat{\Phi}_t(\widehat{X}_t^m, \pi_t) \right\}$
      [Update value function with optimal paths]
      $\widehat{V}_t(\widehat{X}_t^m) := R_t(\widehat{X}_t^m, \pi_t^*(\widehat{X}_t^m))$
      $\widehat{X}_{t+1}^m := \mathcal{T}_t(\widehat{X}_t^m, \pi_t^*(\widehat{X}_t^m), z_t^m)$
      for $t_j = t + 1$ **to** $N - 1$ **do**
        $\widehat{V}_t(\widehat{X}_t^m) := \widehat{V}_t(\widehat{X}_t^m) + \beta^{t_j - t} R_{t_j}(\widehat{X}_{t_j}^m, \pi_{t_j}^*(\widehat{X}_{t_j}^m))$
        $\widehat{X}_{t_j+1}^m := \mathcal{T}_t(\widehat{X}_{t_j}^m, \pi_{t_j}^*(\widehat{X}_{t_j}^m), z_{t_j}^m)$
      end for
      $\widehat{V}_t(\widehat{X}_t^m) := \widehat{V}_t(\widehat{X}_t^m) + \beta^{N-t} R_N(\widehat{X}_N^m)$
    end for
  end if
end for

Note that at terminal time $t = N$, where no decision is allowed, $\widetilde{X}_N = X_N$ holds true, which is why $\widetilde{X}_N$ is used on line 3 in Algorithm 3 (hence no need to simulate X_N which is reflected in Algorithm 2 as it stops at $N-1$). Furthermore, on line 9 and 10 the state after control $\widetilde{X}_t$ is used, in order to estimate the value function for the current period t on line 14. This way π_t^* has been found and the value function for $\widetilde{X}_t$ rather than X_t has already been prepared so it can be used directly for the next iteration.

The loop starting at line 13 is crucial for multi-period stochastic control problems with utility functions (unless the basis functions are correct for all periods). It updates the forward simulation at each backward step with the optimal decision, similar to Longstaff and Schwartz (2001). It therefore uses the realised value function rather than the regression surface (VFI and PFI methods in Equation (22) and (23)) and takes advantage of the tower property of conditional expectations[3]. This step helps significantly with the accuracy of the approximation as the time horizon extends, and avoids (or at least reduces) the risk of the solution blowing up by limiting the accumulation of regression errors. Compare this with Algorithm 4, which is the equivalent of Algorithm 3 but based on the regression surface (VFI approach in Equation (22)) rather than realised values (PFI approach in Equation (23)). The algorithm does not update the forward simulation at each pass, hence is faster but might pile regression errors in the value function. However, in Algorithm 3, the function for the optimal control $\pi_t^*(\widehat{X}_t)$ is solved each time period during the forward loop on line 13 that makes the algorithm more computationally expensive. By storing the optimal control for each state sample, rather than just the regression coefficients, the optimal control can instead be interpolated for each t. This significantly speeds up the solution, especially as the number of dimensions/controls increases.

Algorithm 4 Backward solution (Regression surface)

1: **for** $t = T$ **to** 0 **do**
2: **if** $t = T$ **then**
3: $\widehat{V}_t(\widetilde{\boldsymbol{X}}_t) := R_N(\widetilde{\boldsymbol{X}}_t)$
4: **else if** $t < T$ **then**
[Regression of transformed value function]
5: $\widehat{\boldsymbol{\Lambda}}_t := \arg\min_{\boldsymbol{\Lambda}_t} \sum_{m=1}^{M} \left[\boldsymbol{\Lambda}_t' \mathbf{L}(X_t^m, \widetilde{\pi}_t) - H^{-1}(\beta \widehat{V}_{t+1}(\widetilde{X}_{t+1}^m))\right]^2$
6: Find bias corrected transformation $H^B(\widehat{\boldsymbol{\Lambda}}_t' \mathbf{L}(X_t, \widetilde{\pi}_t))$
[Approximate conditional expectation]
7: $\widehat{\Phi}_t(X_t, \widetilde{\pi}_t) := H^B(\widehat{\boldsymbol{\Lambda}}_t' \mathbf{L}(X_t, \widetilde{\pi}_t))$
8: **for** $m = 1$ **to** M **do**
9: $\widehat{X}_t^m := \widetilde{X}_t^m$
[Optimal control]
10: $\pi_t^*(\widehat{X}_t^m) := \arg\sup_{\pi_t \in A} \left\{ R_t(\widehat{X}_t^m, \pi_t) + \widehat{\Phi}_t(\widehat{X}_t^m, \pi_t) \right\}$
11: $\widehat{V}_t(\widehat{X}_t^m) := R_t(\widehat{X}_t^m, \pi_t^*(\widehat{X}_t^m)) + \widehat{\Phi}_t(\widehat{X}_t^m, \pi_t^*(\widehat{X}_t^m))$
12: **end for**
13: **end if**
14: **end for**

If a fixed starting point X_0 is desired as in the original algorithm, rather than a range of potential starting points, then line 9 would be replaced at $t = 0$ with $\widehat{X}_0^m := X_0$.

[3]The tower property states that when conditioning twice, with respect to nested σ-algebras, the smaller amount of information always prevails such that $\mathbb{E}[\mathbb{E}[Z|\mathcal{F}_{t+1}]|\mathcal{F}_t] = \mathbb{E}[Z|\mathcal{F}_t]$

4.3 Upper and lower bounds

The value function from Algorithm 3 is already a lower bound (up to the Monte Carlo error), as replacing the supremum in Equation (3) with the estimated optimal control yields a lower bound by definition of the supremum (Aïd et al., 2014). Similarly, an approximate upper bound can be found by using the optimal control estimate π_t^* from Algorithm 3, but on line 11 in Algorithm 4 where the realised value function is replaced by the regression surface. Given that estimator $\hat{\Phi}_t(X_t, \pi_t)$ of the approximated conditional expectation in Equation (5) is unbiased, this results in an upper bound up to the Monte Carlo and regression error (Aïd et al., 2014).

Even if the transformation method minimises the regression error, the error will always be present. Given a concave utility function these errors will be biased downwards. Since this upper bound is based on the regression surface, rather than realised values, it will accumulate significant regression bias over time and often leads to lower value than the lower bound. An alternative approach is to use an upper bound based on the expected change of the stochastic variables, as this will always be equal or larger according to Jensen's inequality given a concave utility function. As the lower and upper bound now will suffer equally from the regression bias it represents a more realistic upper bound of the solution. This only holds true for a concave utility function – if the function is convex the inequality changes direction.

5 Accuracy of solution

In this section we examine the impacts of Smearing Estimate (12), Smearing Estimate with Controlled Heteroskedasticity (14) and naive estimate without transformation bias correction (10) on the accuracy of the LSMC numerical solution. Two basic models based on CRRA utility are considered: optimal consumption, and optimal consumption and risky asset allocation. The models have closed form solutions presented in Appendix C which are used for benchmarking of our numerical methods. The chosen model parameter values correspond to the ones causing problems with numerical solutions in Denault et al. (2017) in the case of high risk aversion and volatility.

In both examples, we use 10 000 sample paths where state and control were sampled from uniform distributions, and the disturbance term from a normal distribution. The basis functions are ordinary polynomials up to the 4th order in both transformed state and control variables, including mixed terms. The transformation used is based on the exponent and a CRRA utility function, such that

$$H^{-1}(x) = \ln\left[(\gamma x)^{1/\gamma}\right]. \tag{24}$$

Note that the examples below do not include the standard LSMC case, and use re-sampling in the forward simulation (Algorithm 2). The reason is simply because the standard LSMC method is not stable and the solution either blows up, or optimal control equals full or zero consumption due to bad regression fit if re-sampling in the forward simulation step is not used.

5.1 Consumption model

Consider a typical simple model where the agent receives utility by consuming a proportion $\pi_t := \alpha_t \in [0, 1]$ of the endogenous state variable wealth X_t each period $t = 0, ..., 9$,

hence terminal time $T = N = 9$ resulting in 10 evaluations. The utility at time t is $R_t(X_t, \pi_t) = (\alpha_t X_t)^\gamma/\gamma$ and utility of wealth at terminal time is $R_N(X_N) = (\alpha_N X_N)^\gamma/\gamma$ with risk aversion $\gamma = -10$. Wealth change between periods is based on a stochastic return $Z \sim \mathcal{N}(\mu, \sigma^2)$ with drift $\mu = 0.1$ and standard deviation $\sigma = 0.2$, such that the transition to the wealth at $t+1$ is

$$X_{t+1} = \mathcal{T}_t(X_t, \pi_t, Z_{t+1}) := X_t(1 - \alpha_t)e^{Z_{t+1}}. \tag{25}$$

The closed-form solution for optimal consumption is then

$$\alpha_t = \begin{cases} 1, & \text{if } t = N, \\ (1 + (e^{\gamma\mu + \gamma^2\sigma^2/2}\alpha_{t+1}^{\gamma-1})^{\frac{1}{1-\gamma}})^{-1}, & \text{otherwise,} \end{cases} \tag{26}$$

and the value function is

$$V_t(X_t^\pi) = \frac{(X_t^\pi)^\gamma}{\gamma}(\alpha_t)^{\gamma-1}, \tag{27}$$

as derived in Appendix C. The problem does not have any control variables for allocation of wealth into the risky and riskless assets and is wealth independent, hence no heteroskedasticity will exist and Smearing Estimate (12) is accurate enough. Figure 1 shows the optimal consumption each time period for the different methods described in previous sections. A clear bias can be identified for the transformation without consideration to the re-transformation bias, indicating it will not give an accurate solution, while the two other methods based on Smearing Estimate (12) and Smearing Estimate with Controlled Heteroscedastisity (14) are very close to the true optimal value.

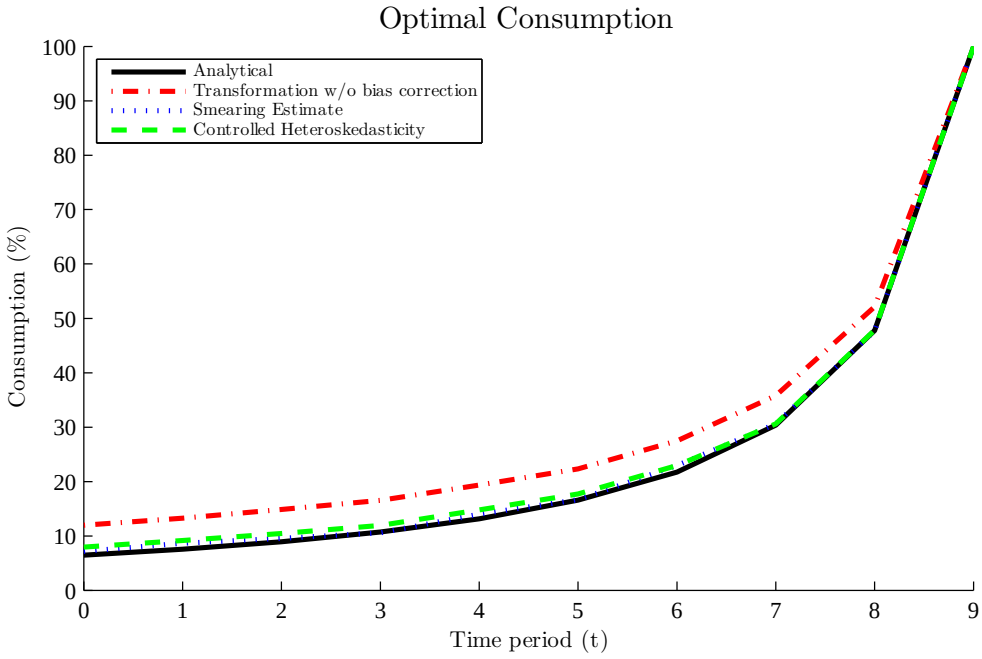


Figure 1: Optimal consumption α_t as a percentage proportion of wealth for four different solution methods.

5.2 Consumption and investment model

The second model we consider is based on the first, but extended with an additional control variable for risky asset allocation. The agent now consumes a proportion of wealth α_t and chooses to allocate a proportion $\delta_t \in [0, 1]$ of remaining wealth into a risky asset with stochastic return Z_t and the rest into a risk free asset with deterministic return $r = 0.03$. Hence, the decision variables are $\pi_t = (\alpha_t, \delta_t)$ and the transition function is

$$X_{t+1} = \mathcal{T}_t(X_t, \pi_t, Z_{t+1}) := X_t(1 - \alpha_t)e^{\delta_t Z_{t+1} + (1-\delta_t)r}. \tag{28}$$

This transition function is approximately the same as the correct transition function for the specified allocation problem when returns are small. The closed-form solution (see Appendix C) gives optimal consumption

$$\alpha_t = \begin{cases} 1, & \text{if } t = N, \\ (1 + (e^{\gamma\delta_t\mu + \gamma^2\delta_t^2\sigma^2/2 + (1-\delta_t)\gamma r}\alpha_{t+1}^{\gamma-1})^{\frac{1}{1-\gamma}})^{-1}, & \text{otherwise,} \end{cases} \tag{29}$$

optimal risky allocation

$$\delta_t = \frac{r - \mu}{\gamma\sigma^2}, \tag{30}$$

and the value function is the same as in (27).

These changes introduce heteroskedasticity with respect to the control variable by allowing control of the disturbance term, and the Smearing Estimate no longer gives a correct solution. Figure 2 shows the optimal consumption for each time period and different methods, and Figure 3 shows the optimal risky allocation. Since the effect of the disturbance control is not transformed back, the risk is underestimated and therefore suggests full risky investment allocation as risk does not increase with higher allocation. The Smearing Estimate then includes a slight bias with regards to optimal consumption due to a constant risk being transformed, which is higher than the true optimal consumption but still underestimated in regards to δ_t. The transformation without bias correction, however, now seems to be accurate, but does in fact include two biases that happen to cancel each other out in this example. As the risk is ignored, the risk from the transformation without bias correction is underestimated and full risky investment is suggested. This in turn affects the expected capital growth, and together with the bias in the consumption decision the overall bias is almost cancelled out. Controlled Heteroskedasticity, on the other hand, considers the effect of disturbance terms once re-transformed and leads to an unbiased solution (but still subject to noise, regression and numerical errors).

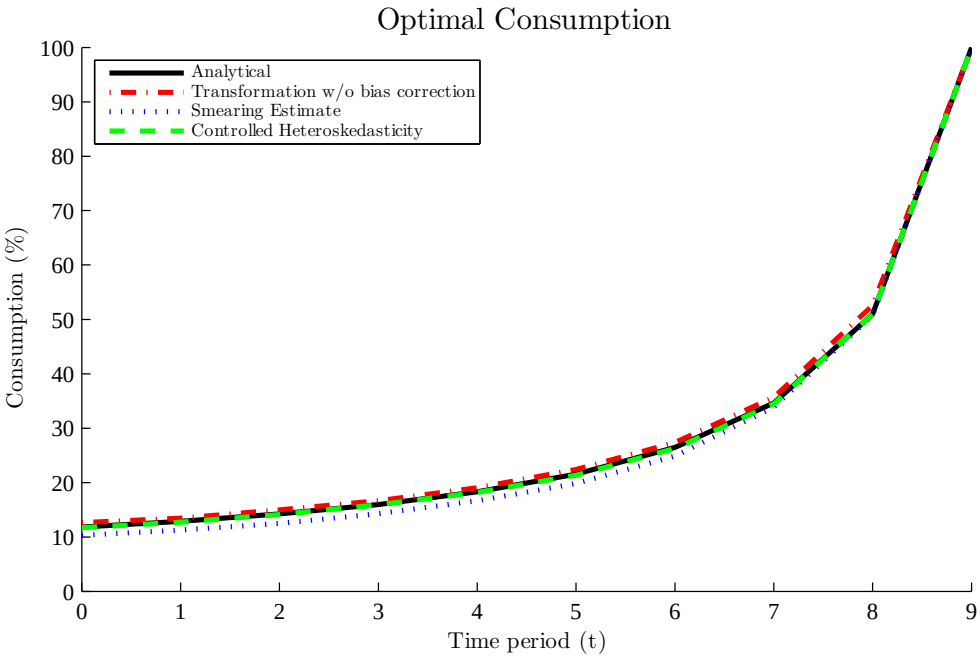


Figure 2: Optimal consumption α_t as a percentage proportion of wealth when the model allows risky investments, for four different solution methods.

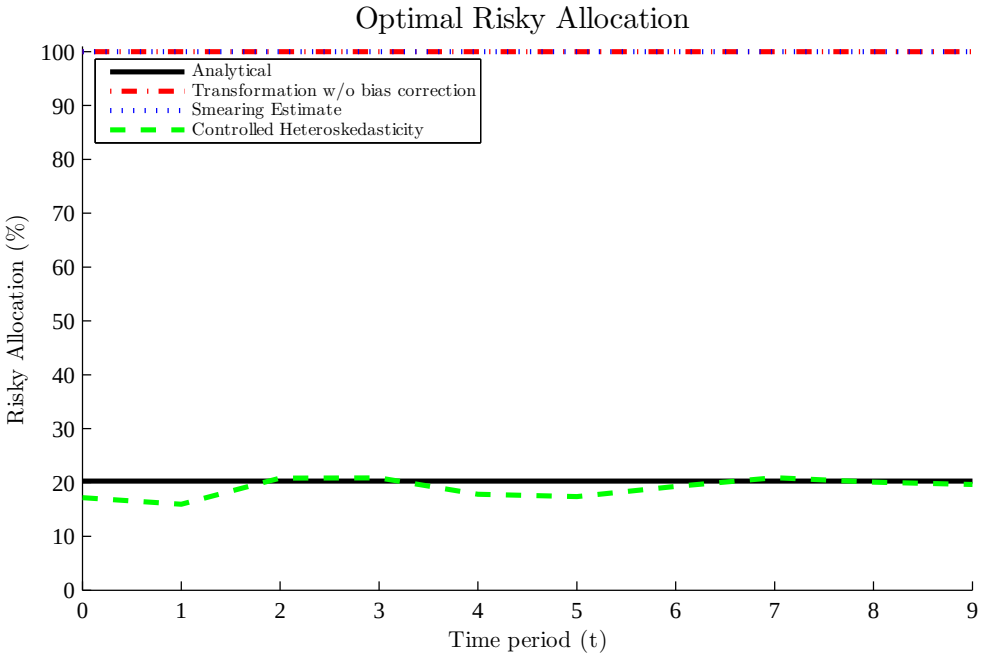


Figure 3: Optimal allocation of risky assets δ_t for four different solution methods.

5.3 Bounded solutions

Even though the accuracy of the estimated optimal control variables compared with the true optimal control is satisfactory, we performed a further analysis to see whether the

calculated upper and lower bounds of the LSMC value function spans over the exact value function. The bounds are estimated as described in Section 4.3, in the case of Model 2 with Controlled Heteroskedasticity (14), considered in Section 5.2. The problem is solved with 20 independent iterations where each iteration involves $M = 10000$ independent sample paths. In order to estimate the solution and standard errors, the solutions of the 20 iterations are averaged and standard errors are calculated. Note that a utility function will always bias errors downwards, assuming it is concave, and the extreme curvature will quickly affect any bias. Therefore, the value function is compared on the transformed scale. Table 3 shows the lower bound Q_L (which is the value from using Algorithm 2 or 3), the upper bound Q_U from replacing the realised disturbance with the expectated value of the disturbance term. The absolute difference of the true optimal control parameters and the numerical solution for consumption, $|\Delta\alpha|$, and risky asset allocation, $|\Delta\delta|$, are shown as the average and the maximum difference. Regression is based on ordinary polynomials up to the n-th degree in both X_t and π_t, and results in the table are presented for $n = 2, 3, 4, 6, 8$.

Table 3: Bounded solutions and differences in control variables with different basis functions. The analytical solution of the problem is $H^{-1}(V_0) = 9119$.

nth degree	Q_{L}	Q_{U}	$\lvert\Delta\alpha\rvert_{\mathrm{avg}}$	$\lvert\Delta\alpha\rvert_{\max}$	$\lvert\Delta\delta\rvert_{\mathrm{avg}}$	$\lvert\Delta\delta\rvert_{\max}$
2	8735.9 (1.4)	8968.2 (2.0)	0.0011	0.0016	0.0285	0.0640
3	8742.9 (1.2)	9009.1 (6.4)	0.0011	0.0017	0.0225	0.0562
4	8745.7 (1.7)	9027.0 (4.6)	0.0010	0.0016	0.0215	0.0737
6	8742.5 (2.0)	9006.4 (7.4)	0.0012	0.0018	0.0255	0.0832
8	8739.3 (2.1)	9001.0 (7.2)	0.0014	0.0032	0.0235	0.0602

The difference between the bounds and the true value is due to the regression bias, and that an approximate model always will be suboptimal. As the upper bound given in Aïd et al. (2014) contains a significant amount of regression bias over time, which is further inflated by utility based objective functions, this value turns out to be *lower* than Q_{L} and has been omitted from the table. Using an upper bound based on the expected change of the stochastic variables will therefore reflect a more realistic upper bound, although it can still be less than the true solution due to regression and numerical errors or early stopping criteria in the optimisation of optimal control.

The difference between the approximation and the analytical solution equals to as little difference as 4% using the forth order polynomials. The optimal control tends to be close to the true optimal value. This can be seen when polynomials with a higher order than four are used, as the regression model now contains redundant predictors. The optimal control parameters are still within a valid range from their true value, even though the non-transformed value function starts to deviate quickly.

6 Conclusion

LSMC provides many advantages in dynamic programming. Firstly, it does not suffer from the 'curse of dimensionality' in the same way as other methods, and is therefore faster than numerical methods such as PDE or quadrature based ones. Secondly, it does not impose restrictions on the stochastic variable dynamics, hence, even an empirical

distribution is sufficient. Finally, it returns a parametric estimate in a feedback form of control which voids the need for a grid for control. However, there are many difficulties as well. It is an approximate method only and can be computationally intensive – especially in finding the optimal control variables for each sample. The basis functions are often difficult to find and highly problem-specific, and if they are not defined properly, substantial errors can pile up over multiple periods.

In this paper the LSMC method was applied on stochastic control problems characterised by utility functions. We found that standard LSMC does not work well for these problems and suggest to perform regression on transformed value function and then accounting for the re-transformation bias. The bias correction function can be constructed in various ways depending on the type of problem. The Smearing Estimate can improve the accuracy of simpler problems without heteroskedasticity and control of disturbance, while more complex problems require Smearing Estimate with Controlled Heteroskedasticity if the heteroskedasticity depends on the state or control variables. The latter requires performing two regressions, but since the computational burden is on the optimisation and not the regression the additional computational cost is minimal. We further observed that the standard forward simulation stage of LSMC should be modified to achieve accurate results. In particular, we suggest to re-sample state variables independently at each time step to achieve better exploration of the state space. This occurs when the sample paths are simulated with control randomisation and the control has a significant influence on the transition of the state variable, thus all sample paths tend to end up in a small sub-domain of the state after simulation. By re-sampling the state variables each time step, we can ensure that the samples exist in the full state variable domain.

The numerical examples presented in the paper are very basic, but serve the purpose of demonstrating the increased accuracy when re-transformation and bias correction are applied. The motivation for our work on improving LSMC with respect to difficult objective functions was to solve models in more realistic setups, such as the sequential retirement model with optimal annuitisation and flexible housing decisions considered in our work in progress Andreasson and Shevchenko (2017). Without the improvements presented in this paper, such model would have to rely on quadrature based solutions and would not be computationally feasible.

Acknowledgment

This research was supported by the CSIRO-Monash Superannuation Research Cluster, a collaboration among CSIRO, Monash University, Griffith University, the University of Western Australia, the University of Warwick, and stakeholders of the retirement system in the interest of better outcomes for all. Pavel Shevchenko acknowledges the support of Australian Research Council's Discovery Projects funding scheme (project number DP160103489) and travel support from the Institute of Statistical Mathematics, Japan.

References

Aïd, René, Luciano Campi, Nicolas Langrené, and Huyên Pham (2014), "A Probabilistic Numerical Method for Optimal Multiple Switching Problems in High Dimension." *SIAM Journal on Financial Mathematics*, 5, 191–231.

Andreasson, Johan G. and Pavel V. Shevchenko (2017), "Optimal Annuitisation, Housing Decisions and Means-Tested Public Pension in Retirement." *Preprint SSRN: 28075551*, URL `https://ssrn.com/abstract=2985830`.

Baser, O (2007), "Modeling Transformed Health Care Cost with Unknown Heteroskedasticity." *Applied Economics Research Bulletin*, 1, 1–6.

Brandt, Michael W., Amit Goyal, Pedro Santa-Clara, and Jonathan R. 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, Mark and Paul Glasserman (2004), "A stochastic mesh method for pricing high- dimensional American options." *Journal of Computational Finance*, 7, 35–72.

Chen, Wen, Nicolas Langrené, and Tanya Tarnopolskaya (2015), "Switching Surfaces for Optimal Natural Resource Extraction under Uncertainty." In *MODSIM2015, 21st International Congress on Modelling and Simulation* (T. Weber, M.J. McPhee, and R.S Anderssen, eds.), 1063–1069, Modelling and Simulation Society of Australia and New Zealand. URL `www.mssanz.org.au/modsim2015/E6/chen.pdf` (Accessed 2016-12-02).

Denault, Michel, Erick Delage, , and J-G Simonato (2017), "Dynamic portfolio choice: a simulation-and-regression approach." *Optimization and Engineering*, 9, 1–38.

Denault, Michel, J-G Simonato, and Lars Stentoft (2013), "A simulation-and-regression approach for stochastic dynamic programs with endogenous state variables." *Computers and Operations Research*, 40, 2760–2769.

Duan, Naihua (1983), "Smearing estimate: A Nonparametric retransformation method." *Journal of the American Statistical Association*, 78, 605–610.

Fabozzi, Frank J., Tommaso Paletta, and Radu Tunaru (2017), "An improved least squares Monte Carlo valuation method based on heteroscedasticity." *European Journal of Operational Research*, 263, 698–706.

Garlappi, Lorenzo and Georgios Skoulakis (2010), "Solving Consumption and Portfolio Choice Problems: The State Variable Decomposition Method." *Review of Financial Studies*, 23, 3346–3400.

Glasserman, Paul and Bin Yu (2004), "Simulation for American Options: Regression Now or Regression Later?" In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, 213–226, Springer Berlin Heidelberg, Berlin, Heidelberg, URL `http://link.springer.com/10.1007/978-3-642-18743-8{_}12`.

Greene, W.H. (2008), *Econometric Analysis*, 6th edition. Pearson Prentice Hall, New Jersey.

Harvey, A. (1976), "Estimating regression models with multiplicative heteroscedasticity." *Econometrica*, 44, 461–465.

Hoff, Peter D. and Xiaoyue Niu (2012), "A Covariance Regression Model." *Statistica Sinica*, 22, 1–31.

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

Kharroubi, Idris, Nicolas Langrené, and Huyên Pham (2015), "Discrete time approximation of fully nonlinear HJB equations via BSDEs with nonpositive jumps." *The Annals of Applied Probability*, 25, 2301–2338.

Langrene, N, Tanya Tarnopolskaya, W Chen, Z Zhu, and M Cooksey (2015), "New Regression Monte Carlo Methods for High-dimensional Real Options Problems in Minerals industry." *21st International Congress on Modelling and Simulation*, 1077–1083.

Longstaff, Francis A and Eduardo S Schwartz (2001), "Valuing American Options by Simulation: A Simple Least-Squares Approach." *Review of Financial Studies*, 14, 113–147.

Muller, Hans-Georg and Ulrich Stadtmuller (1987), "Estimation of Heteroscedasticity in Regression Analysis." *The Annals of Statistics*, 15, 610–625.

Nadarajah, Selvaprabu, François Margot, and Nicola Secomandi (2017), "Comparison of least squares Monte Carlo methods with applications to energy real options." *European Journal of Operational Research*, 256, 196–204, URL `http://dx.doi.org/10.1016/j.ejor.2016.06.020`.

Nadarajah, Selvaprabu and Nicola Secomandi (2017), "Relationship between least squares Monte Carlo and approximate linear programming." *Operations Research Letters*, 45, 409–414, URL `http://dx.doi.org/10.1016/j.orl.2017.05.010`.

Novikov, A. A. and A. N. Shiryaev (2005), "On an Effective Solution of the Optimal Stopping Problem for Random Walks." *Theory of Probability & Its Applications*, 49, 344–354.

Samuelson, P (1969), "Lifetime portfolio selection by dynamic stochastic programming." *The Review of Economics and Statistics*, 51, 239–246.

Smyth, Gordon K. (1989), "Generalized linear models with varying dispersion." *Journal of the Royal Statistical Society*, 51, 47–60.

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

Zhang, Rongju, Nicolas Langren, Yu Tian, Zili Zhu, Fima Klebaner, and Kais Hamza (2016a), "Dynamic Portfolio Optimisation with Intermediate Costs: A Least-Squares Monte Carlo Simulation Approach." SSRN Working Paper No 2696968.

Zhang, Rongju, Nicolas Langrené, Yu Tian, Zili Zhu, Fima Klebaner, and Kais Hamza (2016b), "Efficient Simulation Method for Dynamic Portfolio Selection with Transaction Cost, Liquidity Cost and Market Impact." arXiv Working Paper No 1610.07694.

Zhou, Xiao, Huazhen Lin, and Eric Johnson (2008), "Nonparametric Heteroscedastic Transformation Regression Models for Skewed Data with an Application to Health Care Costs." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 70, 1029–1047.

A Duan's smearing estimate

The results in this section are based on Duan (1983). Denote the non-transformed observations $Y_i, i = 1, ..., n$ and the transformed observations $\eta_i, i = 1, ..., n$ such that $\eta_i = g(Y_i)$, $Y_i = h(\eta_i)$, i.e. $h := g^{-1}$. Assume g (the transformation) and h (the re-transformation) are known monotonic and continuously differentiable functions, such as a CRRA utility function $h(x) = x^\gamma/\gamma,\ \gamma < 0$. Consider the linear regression carried out on the transformed observations

$$\eta_i = \boldsymbol{\beta}'\boldsymbol{X}_i + \epsilon_i, \quad \epsilon_i \overset{i.i.d}{\sim} F(\cdot), \quad \mathbb{E}[\epsilon_i] = 0, \quad \text{var}[\epsilon_i] = \sigma^2, \tag{31}$$

where $\boldsymbol{\beta}$ is the vector of coefficients, $\boldsymbol{X}_i$ is the vector of covariates and ϵ_i are the independent and identically distributed residuals from some zero mean distribution $F(\cdot)$ with finite variance. The error terms do not need to have a known distribution, although they are expected to have zero mean and constant variance. Now, if the re-transformation is applied to the prediction of the transformed variables we would get an incorrect estimate due to Jensen's inequality, because $\mathbb{E}[Y] \leq h(\mathbb{E}[\boldsymbol{\beta}'\boldsymbol{X} + \epsilon])$ if h is a concave function such as a utility function.

Smearing Estimate attempts to approximate the non-transformed expectation

$$\mathbb{E}[Y] = \mathbb{E}\left[h(\boldsymbol{\beta}'\boldsymbol{X} + \epsilon)\right] = \int h\left(\boldsymbol{\beta}'\boldsymbol{X} + \epsilon\right) dF(\epsilon) \tag{32}$$

after estimating the regression coefficients $\widehat{\boldsymbol{\beta}}$ using the empirical distribution function of the residuals $\widehat{\epsilon}_i = \eta_i - \widehat{\boldsymbol{\beta}}'\boldsymbol{X}_i$:

$$\widehat{F}_n(e) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\widehat{\epsilon}_i \leq e\}, \tag{33}$$

where $\mathbb{I}\{\cdot\}$ is the indicator symbol that equals 1 if the statement in brackets $\{\cdot\}$ is true and 0 otherwise. The estimated expectation of Y can then be found as

$$\widehat{\mathbb{E}}[Y] = \int h\left(\widehat{\boldsymbol{\beta}}'\boldsymbol{X} + \epsilon\right) d\widehat{F}_n(\epsilon) = \frac{1}{n}\sum_{i=1}^{n} h\left(\widehat{\boldsymbol{\beta}}'\boldsymbol{X} + \widehat{\epsilon}_i\right). \tag{34}$$

To illustrate, suppose we consider regression $\ln Y_i = \boldsymbol{\beta}'\boldsymbol{X}_i + \epsilon_i$ and we want to estimate $\mathbb{E}[Y^\gamma/\gamma]$, then the Smearing Estimate is

$$\frac{1}{n}\sum_{i=1}^{n} \frac{\left(e^{\widehat{\boldsymbol{\beta}}'\boldsymbol{X}+\widehat{\epsilon}_i}\right)^\gamma}{\gamma} = \frac{\left(e^{\widehat{\boldsymbol{\beta}}'\boldsymbol{X}}\right)^\gamma}{n\gamma}\sum_{i=1}^{n} e^{\widehat{\epsilon}_i\gamma}. \tag{35}$$

The Smearing Estimate works well for non-normal errors and can accommodate for heteroskedasticity, provided it is not related to a covariate.

B Controlled Heteroskedasticity

Consider a simple model with heteroskedasticity, such as $Y = \boldsymbol{\beta}'\boldsymbol{X} + \epsilon$ where $\boldsymbol{X}$ is a vector of covariates, $\boldsymbol{\beta}$ is a vector of regression coefficients, $\mathbb{E}[\epsilon] = 0$ and $\text{var}[\epsilon] = \sigma^2 c(\boldsymbol{X})$. There are various ways to estimate function $c(\boldsymbol{X})$ that is causing heteroskedasticity. In particular, we adopt a popular method from Harvey (1976) (also see Baser (2007), (Greene, 2008, chapter 8)). Assume $c(\boldsymbol{X}) = e^{\boldsymbol{\mathcal{L}}'\boldsymbol{X}}$ to avoid negative values, where $\boldsymbol{\mathcal{L}} = \mathcal{L}_0, \mathcal{L}_1, ..., \mathcal{L}_K$ is another vector of regression coefficients. Thus

$$\epsilon^2 = \sigma^2 c(\boldsymbol{X}) v = \sigma^2 e^{\boldsymbol{\mathcal{L}}'\boldsymbol{X}} v^2, \quad \mathbb{E}[v] = 0,\ \mathbb{E}[v^2] = 1 \tag{36}$$

and we can write

$$\ln(\epsilon^2) = a + \mathcal{L}_1 X_1 + ... + \mathcal{L}_K X_K + \ln v^2, \tag{37}$$

where $a = \ln(\sigma^2) + \mathcal{L}_0$. The parameter estimates are found by two-stage procedure. First, we find the ordinary least squares estimate $\widehat{\boldsymbol{\beta}}$ and calculate the observed residuals $\widehat{\epsilon} = Y - \widehat{\boldsymbol{\beta}}'\boldsymbol{X}$. Then we perform the ordinary linear regression (37) where unobserved ϵ are replaced with $\widehat{\epsilon}$ to estimate the variance function $\widehat{\sigma}^2 \exp(\widehat{\boldsymbol{\mathcal{L}}}'\boldsymbol{X})$. Finally, using estimated variance, $\boldsymbol{\beta}$ is approximated by the weighted least squares method. The process can be iterated to improve the estimates.

Other methods to estimate $c(X)$ include random effect representation (Hoff and Niu, 2012), kernel estimates (Muller and Stadtmuller, 1987) or via link functions (Smyth, 1989).

C Solution to multiperiod utility model

In this section we derive the analytical solution for optimal drawdown and risky asset allocation in the multiperiod utility model considered in Section 5.2. The objective is to maximize the expected value function

$$V_0(x) = \sup_{\pi} \mathbb{E}\left[\sum_{t=0}^{N-1} \frac{(\alpha_t X_t)^\gamma}{\gamma} \;\middle|\; X_0 = x; \pi\right]. \tag{38}$$

Let ξ_t represent the stochastic component in the transition function, such as $\xi_t = e^{Z_{t+1}}$ in the case of a single risky asset. This type of problem was originally solved in Samuelson (1969).

At the terminal time $t = N$, the value function is given by

$$V_N(X_N) = \frac{(\alpha_N X_N)^\gamma}{\gamma}. \tag{39}$$

It is optimal to consume all wealth as no utility is received from saving wealth, hence by intuition $\alpha_N = 1$. The risky asset allocation at this point has no impact.

At time $t = N - 1$, the value function is

$$\begin{aligned} V_{N-1}(X_{N-1}) &= \frac{(\alpha_{N-1} X_{N-1})^\gamma}{\gamma} + \mathbb{E}\left[V_N(X_N)\right] \\ &= \frac{(\alpha_{N-1} X_{N-1})^\gamma}{\gamma} + \frac{((1-\alpha_{N-1})X_{N-1})^\gamma \mathbb{E}\left[\xi_{N-1}^\gamma\right]}{\gamma}. \end{aligned} \tag{40}$$

To find the optimal drawdown, differentiate with respect to α_{N-1}

$$\frac{\partial V_{N-1}}{\partial \alpha_{N-1}} = X_{N-1}(\alpha_{N-1} X_{N-1})^{\gamma-1} - X_{N-1}((1-\alpha_{N-1})X_{N-1})^{\gamma-1}\mathbb{E}\left[\xi_{N-1}^\gamma\right], \tag{41}$$

set this equal to 0 and solve for α_{N-1}

$$\begin{aligned} & X_{N-1}(\alpha_{N-1} X_{N-1})^{\gamma-1} - X_{N-1}((1-\alpha_{N-1})X_{N-1})^{\gamma-1}\mathbb{E}\left[\xi_{N-1}^\gamma\right] = 0 \\ & \Rightarrow \alpha_{N-1} = (1-\alpha_{N-1})\mathbb{E}\left[\xi_{N-1}^\gamma\right]^{\frac{1}{\gamma-1}} \\ & \Rightarrow \alpha_{N-1} = (1 + \mathbb{E}\left[\xi_{N-1}^\gamma\right]^{\frac{1}{1-\gamma}})^{-1}. \end{aligned} \tag{42}$$

If the stochastic growth of wealth depends on a control variable, such as if $\xi_{N-1} = e^{\delta_{N-1} Z_N + (1-\delta_{N-1}) r_{N-1}}$ considered in the model in section 5.2, then the same steps are used to find the optimal risky asset allocation δ_{N-1}. In this case, assuming $Z_t \overset{i.i.d}{\sim} \mathcal{N}(\mu, \sigma^2)$,

$$\frac{\partial V_{N-1}}{\partial \delta_{N-1}} = \mathbb{E}\left[(Z_N - r)((1-\alpha_{N-1})X_{N-1})^\gamma \xi_{N-1}^\gamma\right] \tag{43}$$

$$\begin{aligned} & \Rightarrow \mathbb{E}\left[(Z_N - r)(e^{\delta_{N-1} Z_N + (1-\delta_{N-1}) r_{N-1}})^\gamma\right] = 0 \\ & \Rightarrow \delta_{N-1} = \frac{r - \mu}{\gamma \sigma^2}. \end{aligned} \tag{44}$$

Finally, use α_{N-1} to find the maximum of the value function V_{N-1}

$$\begin{aligned} V_{N-1}(X_{N-1}) &= \frac{(\alpha_{N-1}X_{N-1})^{\gamma}}{\gamma} + \frac{((1-\alpha_{N-1})X_{N-1})^{\gamma}\mathbb{E}\left[\xi_{N-1}^{\gamma}\right]}{\gamma} \\ &= \frac{(X_{N-1})^{\gamma}}{\gamma}((\alpha_{N-1})^{\gamma} + (1-\alpha_{N-1})^{\gamma})\mathbb{E}\left[\xi_{N-1}^{\gamma}\right] \\ &= \frac{(X_{N-1})^{\gamma}}{\gamma}(\alpha_{N-1})^{\gamma-1}, \end{aligned} \tag{45}$$

which will be used in the next iteration. By repeating these steps for $t = N-2, ..., 0$ a distinct pattern is found, where

$$\alpha_t = \begin{cases} 1, & \text{if } t = N, \\ (1 + (\mathbb{E}\left[\xi_t^{\gamma}\right]\alpha_{t+1}^{\gamma-1})^{\frac{1}{1-\gamma}})^{-1}, & \text{otherwise.} \end{cases} \tag{46}$$