Les Cahiers du GERAD ISSN: 0711–2440

Dynamic portfolio choices by simulation-and-regression: Revisiting the issue of value function vs portfolio weight recursions

M. Denault, J.-G. Simonato

G–2015–75

August 2015

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Dynamic portfolio choices by simulation-and-regression: Revisiting the issue of value function vs portfolio weight recursions

Michel Denault [a]

Jean-Guy Simonato [b]

[a] *GERAD & Department of Decision Sciences, HEC Montréal, Montréal (Québec) Canada, H3T 2A7*

[b] *Department of Finance, HEC Montréal, Montréal (Québec) Canada, H3T 2A7*

michel.denault@hec.ca
jean-guy.simonato@hec.ca

August 2015

Les Cahiers du GERAD

G–2015–75

Abstract: Simulation-and-regression methods have been recently proposed to solve multi-period, dynamic portfolio choice problems. In the constant relative risk aversion (CRRA) framework, the "value function recursion vs portfolio weight recursion" issue was previously examined in van Binsbergen and Brandt (2007) and Garlappi and Skoulakis (2009). We revisit this issue in the context of an alternative simulation-and-regression algorithmic approach which does not rely on Taylor series approximations of the value function. We find that, in this context, the portfolio weight recursion variant of the algorithm provides very precise results, is more reliable, and should be preferred to the value function recursion variant, especially for problems with long maturities and large risk-aversion levels.

Acknowledgments: The first author acknowledges financial support from NSERC.

1 Introduction

Dynamic, incomplete-market portfolio optimization is a challenging task for which there is no analytical solution in most cases. Papers examining such problems have thus extensively relied on numerical approaches to solve the associated dynamic programs. However, many of these numerical approaches are limited in scope because they rely on a normality assumption of the asset returns, and/or on lattice approaches for which it is difficult to handle many assets and state-variables (see for example Brennan et al. (1997), Balduzzi and Lynch (1999), Wang and Judd (2000), Dammon et al. (2001)). In order to avoid these limitations, methods using regressions, and often simulations, have been recently proposed to solve multi-period portfolio problems. Inspired by earlier work for the pricing of American options by Carriere (1996), Tsitsiklis and Van Roy (2001) and Longstaff and Schwartz (2001), these methods can use realistic distributions and time series processes for the returns, and can handle more assets and state-variables than pure lattice based approaches. Brandt et al. (2005), van Binsbergen and Brandt (2007), Garlappi and Skoulakis (2010) are examples where such approaches are applied to a variety of portfolio optimization problems. These algorithms use Taylor series approximations of the value function at every time point, and a combination of least-squares regressions and Monte Carlo simulations (or quadratures) to obtain the required conditional expected values entering these approximate value functions.

For these algorithms and for an investor with a constant relative risk aversion (CRRA) utility function, two variants are possible to perform the computations associated with the backward recursions of the dynamic program: recursions on portfolio weights, and recursions on value functions. van Binsbergen and Brandt (2007) argue that recursing on portfolio weights leads to superior results. Garlappi and Skoulakis (2009) argue, on the contrary, that with the use of an economically-motivated transformation of the value function, recursing on the value function can produce more accurate results than what is claimed in van Binsbergen and Brandt (2007). In both studies, a Taylor series approximation of the value function is used as one of the main ingredient of the numerical approaches.

Although we take a portfolio optimization angle in this paper, the issue of recursion on "regressed values" vs recursion on "realized value" is much broader in scope, and in financial engineering in particular, spans the ubiquitous least-squares Monte Carlo techniques for option pricing (see Glasserman (2004)). In fact, even when an endogenous state variable forces the discretization of the state space, the issue still applies locally in time, that is, until the start of the next period.

In this paper, we revisit this "value function vs portfolio weight recursion" issue, but for a simulation-and-regression approach that does not rely on a Taylor series of the value function and which is invariant to the transformation proposed by Garlappi and Skoulakis (2009). We rely on an approach developed independently by Koijen, Nijman, Werker (2010) and by the authors and Delage (Delage, Denault, Simonato (2014)), that bypasses the need of a Taylor series by regressing on decision variables as well as state variables. The two papers develop approaches that are similar but not identical.[1] The resulting regression surfaces can be optimized easily over a continuous domain in the dimensions of the decision variables to find optimal policies. In opposition to approaches that optimize on finely discretized values of the decision variables, relatively few points are required to obtain good precision. Such an approach is interesting since it provides an alternative simulation-and-regression algorithm that avoid the overhead work associated with the first four derivatives of the value function. Unlike the Taylor series based approach, the algorithm can also handle function that are non differentiable everywhere. Finally, the approach allows a clearer interpretation of both recursion possibilities. Note that although approximate dynamic programming methods that regress on decisions are well known (often under the name "Q-learning"), their application to portfolio problems has been extremely limited; see the literature review of Delage, Denault and Simonato (2014), and the paper by Errais and Sadowsky (2008).

In the framework of this simulation-and-regression algorithm, the distinction between value function (VFR) and portfolio weights recursion (PWR) is quite easy to make. At each point in time and for a simulation path, the regression surface that we design is optimized to find an optimal policy (this ensures non-anticipativity). The *value* associated to this optimal policy is simply that on the regression surface for

[1] Hereafter when we refer to "our approach", we mean the approach from Delage, Denault, Simonato (2014).

value function recursions; it is therefore some sort of average. For portfolio weights recursions, the optimal policy is combined to a specific simulated path and a value is computed, which represents what really happens when *that* policy is applied on *that* path. This distinction between the two recursion possibilities is similar to the one made in the American option pricing literature by simulation-and-regression where the value function recursions correspond to the "regression surface value" approach advocated by Tsitsiklis and Van Roy (2001), while the portfolio weights recursions correspond to the "realized values" approach advocated by Longstaff and Schwartz (2001).[2]

Using a CRRA example based on the examples of van Binsbergen and Brandt (2007) and Garlappi and Skoulakis (2009), we examine the relative merits of VFR and PWR. Taking into account the precision and dispersion of the answers given by both variants, we find that the value function recursion variant is fairly precise for short maturities and small value of the risk aversion coefficient, but breaks down for longer maturities and larger risk aversion levels. The portfolio weights recursion variant however provides results that remain very close to the benchmark results computed with a Gauss-Hermite quadrature approach. Our results are thus similar to those obtained in van Binsbergen and Brandt (2007) who find that portfolio weight recursions are preferable to value function recursions in the context of their Taylor series based algorithms.

The rest of the paper is organized as follows. Section 2 describes the dynamics of the returns and portfolio problem examined in this study. Section 3 presents the simulation-and-regression algorithm in a simple example, and in both Value Function Recursion (VFR) and Portfolio Weigths Recursion (PWR) variants. Section 4 provides the full algorithm while in Section 5 we analyze the biases of each variant. Section 6 examines in detail our numerical results, while Section 7 concludes.

2 The portfolio problem

In order to examine the portfolio weight vs value function recursion issue, we use a problem similar to the one examined in van Binsbergen and Brandt (2007) and Garlappi and Skoulakis (2009) with a portfolio consisting of one risk-free asset with gross return R_f, and a single risky asset with a simple return in excess of the risk-free rate defined as r_t for the time period between $t-1$ and t. The log excess return $R_t = \ln(r_t + 1)$ is predictable with a dynamics given by the following restricted vector autoregressive process:

$$R_{t+1} = a_r + b_r d_t + e_{r,t+1} \tag{1}$$

with

$$d_{t+1} = a_d + b_d d_t + e_{d,t+1} \tag{2}$$

where d_t is the log dividend yield for the time period between $t-1$ and t, and where a_r, b_r, a_d, and b_d are constant parameters, with Gaussian error terms $e_{r,t}$ and $e_{d,t}$ and a constant covariance matrix Σ. In this model, the stock return inherits a predictability property through the AR(1) structure of the dividend yield. Such a model has been used in the multiperiod portfolio choice literature in Barberis (2000), Campbell and Viceira (1999), and Lynch (2001), and of course van Binsbergen, Brandt (2007) and Garlappi, Skoulakis (2009). As discussed in these papers, this dynamics for the return is interesting since it induces substantial differences between the single period and multiperiod portfolio choices.

The investor is endowed with wealth W_0 at time $t = 0$ and wishes to maximize the expected utility of his terminal wealth at date T. There are no frictions: no transaction costs nor taxes, no intermediate consumption. The investor can trade the risky asset and the risk-free security at times $t = 0, 1, ..., T-1$. With these assumptions the investor's problem can be expressed as

$$V_0(W_0, d_0) = \max_{\{x_t\}_{t=0}^{T-1}} \mathbb{E}_0[u(W_T)] \tag{3}$$

with the following constraints for all t :

$$W_{t+1} = W_t \cdot (x_t r_{t+1} + R_f) \tag{4}$$

[2] See Stentoft (2014) for a thorough comparison of both approaches for American option pricing.

where $\mathbb{E}_0$ denotes the expectation conditional on the information available at time $t = 0$, which is, in this problem, the initial wealth W_0, and the initial dividend yield value d_0. Here, $u(W_T) = W_T^{1-\gamma}/(1-\gamma)$, $\gamma > 1$ is the CRRA utility function.

In the present context, as it is well known from the portfolio literature (see Brandt et al. (2005) for example), the homotheticity of the CRRA utility function allows to rewrite the problem in the form of a Bellman equation which is independent of the wealth level. As shown in Appendix A, the following Bellman equation can be obtained (note that wealth is no more an argument, hence the change in notation):

$$v_t(d_t) = \max_{x_t} \mathbb{E}_t \left[(x_t r_{t+1} + R_f)^{1-\gamma} \; v_{t+1}(d_{t+1}) \right] \tag{5}$$

where $v_T(d_T) = 1/(1-\gamma)$. Here, at each time point t, $v_t(d_t)$ can be interpreted as the maximum utility of wealth at T for a dollar of initial wealth at t, conditional on the dividend yield value d_t.

Our simulation-and-regression algorithm uses the above formulation and, in a nutshell, works as follows. At each time step, a simulated sample for the values appearing inside of the conditional expectation on the right hand side of the Bellman equation (5) is generated. Using this sample as a dependent variable, a regression on a basis formed with portfolio weights and dividend yields is used to obtain a conditional expected value function which is continuous in the portfolio weights. This expected value function can then be optimized for each sample path of the simulated dividend yield to obtain the optimal portfolios. With the results from these optimizations, it is then possible to perform the recursions with the optimal portfolio weights or with points on the value function.

It is important to note that this simulation-and-regression method is invariant to the transformation of the value function proposed in Garlappi and Skoulakis (2009). With their method and the one used in van Binsbergen and Brandt (2007), a certainty equivalent transformation linearizing the value function provides more precise function approximations; this is linked to their use of Taylor series. For our algorithm, using their suggested transformation has no impact on the computed sample of dependant variables. More specifically, they propose using

$$v_{t+1}(d_{t+1}) = \frac{1}{1-\gamma} J_{t+1}(d_{t+1})^{1-\gamma}$$

to obtain the following Bellman equation:

$$\frac{1}{1-\gamma} J_t(d_t)^{1-\gamma} = \max_{x_t} \mathbb{E}_t \left[(x_t r_{t+1} + R_f)^{1-\gamma} \; \frac{1}{1-\gamma} J_{t+1}(d_{t+1})^{1-\gamma} \right].$$

The simulation-and-regression algorithm used here generates a sample of values for the terms inside the expected value operator with a grid of portfolio weights. Given the same paths of returns and dividends, using the above formulation for the Bellman equation or the one given by (5) will yield the same sample and will not generate any differences in results.

The next section illustrates the algorithm for a simple example and in both the VFR and the PWR variants.

3 Value function recursion vs portfolio weights recursion: An illustrative example

We introduce our simulation-and-regression algorithm with a simple two-period example. Both Value Function Recursion and Portfolio Weights Recursion variants are illustrated in turn. The example is small enough to be displayed in full, but it also conveys the complete mechanics of the approach. The general algorithm is provided in the following section.

The investor, endowed with current wealth $W_0 = 1$ at time $t = 0$, wants to maximize the utility of his wealth at time $t = 2$, with an optimal asset allocation at time $t = 0$ and $t = 1$, without shortsales or borrowing. In a first step, the algorithm requires the computation of some basic quantities that will be used

later in the recursive steps. These quantities are: i) a grid of portfolio weights; ii) a set of simulated sample paths of returns and dividend yields (the state variable). Denote the grid of portfolio weights as

$$x = \left[\begin{array}{ccc} x^1 & x^2 & x^3 \end{array} \right]^\intercal .$$

For illustration purposes, only two sample paths of returns are simulated at each period. These sample paths can be obtained with equations (1) and (2) and are denoted as:

path	$t=0$	$t=1$	$t=2$
a	d_0	$r_1^{(a)}$ $d_1^{(a)}$	$r_2^{(a)}$ $d_2^{(a)}$
b	d_0	$r_1^{(b)}$ $d_1^{(b)}$	$r_2^{(b)}$ $d_2^{(b)}$

.

We note here that the realization of d_0 is observed at $t = 0$ and is thus identical for each sample path. There is no need for r_0.

The second step, which does the dynamic programming procedure itself, can be implemented as a VFR or as a PWR. The next subsections describe the two variants.

Remarks about the notation: Throughout the paper, upper indices in parentheses refer to simulations (e.g. $d^{(j)}$), plain upper indices refer to grid values (e.g. x^i), and lower indices refer to time. The only exceptions to these rules are for the statistical parameters (a_r, b_r, $e_{r,t+1}$, a_d, b_d, $e_{d,t+1}$, and $\beta_1, \ldots, \beta_6$). In a few occasions where we need to square variable x, we write $(x)^2$. The overhead tilde, bar and double bar refer to respectively the quadrature benchmark algorithm and the VFR and PWR variants (e.g. $\tilde{v}_t$, $\bar{v}_t$, $\bar{\bar{v}}_t$).

3.1 Value function recursions

This section presents the VFR variant. The goal of the recursive process is to solve the Bellman equation (5) for all time periods. We start using the VFR-specific notation $\bar{v}_t(\cdot)$ instead of the generic $v_t(\cdot)$ of equation (5). This recursion is naturally done backwards through time; in our example, since for $T = 2$, $\bar{v}_2$ is already known, the real work starts with $\bar{v}_1$.

In our simulation-and-regression approach, the first recursion task is to simulate a sample of the values inside the expectation on the right-hand side of the Bellman equation. To do this, starting at $t = 1$, one period before maturity, we generate a sample of portfolio returns at $t = 2$. For example, with the weight x^1 and path a of the simulated return we have the portfolio return

$$\rho\left(x^1, r_2^{(a)}\right) \triangleq x^1 r_2^{(a)} + R_f. \tag{6}$$

Using all the possible combinations of simulated returns at $T = 2$ and portfolio weights on the grid, the end-of-period sample of portfolio returns at $t = 2$ is:

$$\left[\begin{array}{ccccc} \rho\left(x^1, r_2^{(a)}\right) & \rho\left(x^2, r_2^{(a)}\right) & \rho\left(x^3, r_2^{(a)}\right) & \rho\left(x^1, r_2^{(b)}\right) & \rho\left(x^2, r_2^{(b)}\right) & \rho\left(x^3, r_2^{(b)}\right) \end{array} \right].$$

The sample utility associated to decision x^i at time $t = 1$, simulation path (j), and end-of-period utility $\bar{v}_2^{(j)}$, is then

$$U_1(x^i, r_2^{(j)}, \bar{v}_2^{(j)}) = \left(\rho\left(x^i, r_2^{(j)}\right)\right)^{1-\gamma} \bar{v}_2^{(j)} = \left(x^i r_2^{(j)} + R_f \right)^{1-\gamma} \bar{v}_2^{(j)} \tag{7}$$

where we introduce a convenient function $U_1(\cdot,\cdot,\cdot)$ and notation $\bar{v}_t^{(j)}$ which is a shortcut for $\bar{v}_t(d_t^{(j)})$.

Using the terminal value function levels $\bar{v}_2^{(a)} = \bar{v}_2^{(b)} = 1/(1-\gamma)$, a linear regression system can then be formed:

$$\begin{bmatrix} U_1\left(x^1, r_2^{(a)}, \bar{v}_2^{(a)}\right) \\ U_1\left(x^2, r_2^{(a)}, \bar{v}_2^{(a)}\right) \\ U_1\left(x^3, r_2^{(a)}, \bar{v}_2^{(a)}\right) \\ U_1\left(x^1, r_2^{(b)}, \bar{v}_2^{(b)}\right) \\ U_1\left(x^2, r_2^{(b)}, \bar{v}_2^{(b)}\right) \\ U_1\left(x^3, r_2^{(b)}, \bar{v}_2^{(b)}\right) \end{bmatrix} = \begin{bmatrix} 1 & x^1 & \left(x^1\right)^2 & d_1^{(a)} & \left(d_1^{(a)}\right)^2 & x^1 d_1^{(a)} \\ 1 & x^2 & \left(x^2\right)^2 & d_1^{(a)} & \left(d_1^{(a)}\right)^2 & x^2 d_1^{(a)} \\ 1 & x^3 & \left(x^3\right)^2 & d_1^{(a)} & \left(d_1^{(a)}\right)^2 & x^3 d_1^{(a)} \\ 1 & x^1 & \left(x^1\right)^2 & d_1^{(b)} & \left(d_1^{(b)}\right)^2 & x^1 d_1^{(b)} \\ 1 & x^2 & \left(x^2\right)^2 & d_1^{(b)} & \left(d_1^{(b)}\right)^2 & x^2 d_1^{(b)} \\ 1 & x^3 & \left(x^3\right)^2 & d_1^{(b)} & \left(d_1^{(b)}\right)^2 & x^3 d_1^{(b)} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix}, \tag{8}$$

where the lines of the independent variable matrix are formed with a constant, powers up to 2 of the portfolio weights, powers up to 2 of the dividend yield at $t = 1$ (the state variable), and the product of the weights and the dividend yield. In this example, we use powers up to two for illustrative purposes, but nothing precludes using other bases. The ε_i are random errors with zero expected value, and β_1 to β_6 are unknown coefficients, whose value can be estimated with an ordinary least square regression; let us call the estimated coefficients $\bar{\beta}_1, \ldots, \bar{\beta}_6$. Using these estimated coefficients, the approximate expected utility, conditional on simulation (a) of the state variable at $t = 1$, can be written as a continuous function of the portfolio weights. This function can in turn be maximized to find the optimal portfolio as:

$$\bar{x}_1^{(a)} = \underset{0 \le x \le 1}{\arg\max}\ \bar{\beta}_1 + \bar{\beta}_2 x + \bar{\beta}_3 (x)^2 + \bar{\beta}_4 d_1^{(a)} + \bar{\beta}_5 \left(d_1^{(a)}\right)^2 + \bar{\beta}_6 x d_1^{(a)}. \tag{9}$$

Notice here that a constraint on the portfolio weight accounts for the restriction on shortsales. The constrained optimum of this same function, denoted $\bar{v}_1^{(a)}$,

$$\bar{v}_1^{(a)} = \max_{0 \le x \le 1}\ \bar{\beta}_1 + \bar{\beta}_2 x + \bar{\beta}_3 (x)^2 + \bar{\beta}_4 d_1^{(a)} + \bar{\beta}_5 \left(d_1^{(a)}\right)^2 + \bar{\beta}_6 x d_1^{(a)}.$$

represents an estimate of a point on the value function i.e. an estimate of the maximum expected utility at $t = 2$ of one dollar of wealth invested at $t = 1$, conditional on a dividend yield $d_1^{(a)}$. This value will be referred to as a "regression surface value".

Similarly, the approximate expected utility, conditional on simulation (b) of our state variable, can be written as a continuous function of the portfolio weights, which can be used to find the optimal portfolio as:

$$\bar{x}_1^{(b)} = \underset{0 \le x \le 1}{\arg\max}\ \ \bar{\beta}_1 + \bar{\beta}_2 x + \bar{\beta}_3 (x)^2 + \bar{\beta}_4 d_1^{(b)} + \bar{\beta}_5 \left(d_1^{(b)}\right)^2 + \bar{\beta}_6 x d_1^{(b)}. \tag{10}$$

The regression surface value $\bar{v}_1^{(b)}$ represents an estimate of the maximum expected utility at $t = 2$ of one dollar of wealth invested at $t = 1$, conditional on a dividend yield value $d_1^{(b)}$.

Hence, to summarize, for each sample path, solving a maximization problem provides an optimal portfolio weight, and an estimate of a point on the value function.

At this point, we are ready to move (backward in time) to $t = 0$. The optimal weight at $t = 0$ can be computed using a similar procedure. We first compute a sample of portfolio returns (recall (6)) at $t = 1$ using all combinations of simulated returns and portfolio weights. The sample is

$$\left[\ \rho\left(x^1, r_1^{(a)}\right)\quad \rho\left(x^2, r_1^{(a)}\right)\quad \rho\left(x^3, r_1^{(a)}\right)\quad \rho\left(x^1, r_1^{(b)}\right)\quad \rho\left(x^2, r_1^{(b)}\right)\quad \rho\left(x^3, r_1^{(b)}\right)\ \right].$$

Using these and the previously computed values $\bar{v}_1^{(a)}$ and $\bar{v}_1^{(b)}$, we can generate a sample of utility values associated to decision x^i at time $t = 0$, simulation path (j), and $\bar{v}_1^{(j)}$

$$U_0(x^i, r_1^{(j)}, \bar{v}_1^{(j)}) = \left(\rho\left(x^i, r_1^{(j)}\right)\right)^{1-\gamma} \bar{v}_1^{(j)}. \tag{11}$$

These points are used as dependent variables in the linear system:

$$\begin{bmatrix} U_0\left(x^1, r_1^{(a)}, \bar{v}_1^{(a)}\right) \\ U_0\left(x^2, r_1^{(a)}, \bar{v}_1^{(a)}\right) \\ U_0\left(x^3, r_1^{(a)}, \bar{v}_1^{(a)}\right) \\ U_0\left(x^1, r_1^{(b)}, \bar{v}_1^{(b)}\right) \\ U_0\left(x^2, r_1^{(b)}, \bar{v}_1^{(b)}\right) \\ U_0\left(x^3, r_1^{(b)}, \bar{v}_1^{(b)}\right) \end{bmatrix} = \begin{bmatrix} 1 & x^1 & \left(x^1\right)^2 \\ 1 & x^2 & \left(x^2\right)^2 \\ 1 & x^3 & \left(x^3\right)^2 \\ 1 & x^1 & \left(x^1\right)^2 \\ 1 & x^2 & \left(x^2\right)^2 \\ 1 & x^3 & \left(x^3\right)^2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix}. \tag{12}$$

We note here that, unlike the earlier regressions, the dividend yield need not be included in the dependent variables, since all sample paths at $t = 1$ have been obtained from the same dividend yield. Hence, only one optimization is required since only one value of the state variable (the known value) is relevant at this point. The optimal portfolio allocation is then:

$$\bar{x}_0 = \underset{0 \le x \le 1}{\arg\max} \ \bar{\beta}_1 + \bar{\beta}_2 x + \bar{\beta}_3 (x)^2$$

with a value function level $\bar{v}_0$ which can be obtained from the regression surface value at $\bar{x}_0$.

Note that at this point in time, there is only one estimated point on the value function, i.e. $\bar{v}_0$ is not dependent on a path, as in the other time steps. This is due to the uniqueness of d_0 (no path dependence), but also to the VFR variant.

3.2 Portfolio weight recursions

The PWR is an alternative to the VFR presented in the previous section. It also uses the grid of portfolio weights, the simulated returns and the dividend yields described at the beginning of Section 3. For the recursive work, at $t = T - 1 = 1$, the computations are also almost identical to those of the VFR variant i.e. the linear system given by equation (8) is formed and the optimal weights are computed. Although for $t = T - 1$ the optimal weights are the same under both the VFR and PWR variants, we already introduce a specific notation for PWR weights, $\bar{\bar{x}}_1^{(a)}$ and $\bar{\bar{x}}_1^{(b)}$ that will be useful shortly.

There is one new computation to perform before the $t = 1$ step is done. The values, denoted $\bar{\bar{v}}_1^{(a)}$ and $\bar{\bar{v}}_1^{(b)}$ associated to the optimal decisions $\bar{\bar{x}}_1^{(a)}$ and $\bar{\bar{x}}_1^{(b)}$ are *not* taken from the regression surface as in VFR, but computed as "realized" utility values for each path. For path (a) (and similarly for path (b)),

$$\begin{aligned} \bar{\bar{v}}_1^{(a)} &= u\left(\bar{\bar{x}}_1^{(a)} r_2^{(a)} + R_f\right), \\ &= \frac{\left(\bar{\bar{x}}_1^{(a)} r_2^{(a)} + R_f\right)^{1-\gamma}}{1-\gamma}. \end{aligned}$$

Note that conveniently, with the same previous final utility as before $\bar{\bar{v}}_2^{(a)} = \bar{\bar{v}}_2^{(b)} = \frac{1}{1-\gamma}$, the computation of $\bar{\bar{v}}_1^{(a)}$ (and similarly for (b)) can be performed based on the next-time-step $\bar{\bar{v}}_2^{(a)}$

$$\begin{aligned} \bar{\bar{v}}_1^{(a)} &= \left(\bar{\bar{x}}_1^{(a)} r_2^{(a)} + R_f\right)^{1-\gamma} \bar{\bar{v}}_2^{(a)}, \\ &= U_1\left(\bar{\bar{x}}_1^{(a)}, r_2^{(a)}, \bar{\bar{v}}_2^{(a)}\right) \end{aligned}$$

where we reuse the U_1 operator introduced in equation (7).

The quantity $\bar{\bar{v}}_1^{(a)}$ can be interpreted as a sample realization of the conditional expected maximum utility value at $t = 2$ since it simultaneously uses the optimal portfolio weight $\bar{\bar{x}}_1^{(a)}$, and the realized sample return along the path, $r_2^{(a)}$. Such a quantity, labelled here as a "realized value", is different from the "regression

surface value" $\bar{v}_1^{(a)}$ introduced earlier with the VFR variant. While the regression surface value uses all the information available at t to obtain the expected value, the "realized" value" uses an additional piece of information given by the realized sample value $r_2^{(a)}$. This use of "future" information is not anticipative: the optimal choice does not rely on this information, only the value associated to the optimal choice, *after* it has been made. This distinction is the same as the one discussed by Longstaff and Schwartz (2001) in the context of option pricing; there the authors argue in favor of "realized values", in opposition to Tsitsiklis and Van Roy (1999) who used "regression surface values".

We can now tackle the period that starts at $t = 0$. The sample of returns

$$\left[\begin{array}{ccccccc} \rho\left(x^1, r_1^{(a)}\right) & \rho\left(x^2, r_1^{(a)}\right) & \rho\left(x^3, r_1^{(a)}\right) & \rho\left(x^1, r_1^{(b)}\right) & \rho\left(x^2, r_1^{(b)}\right) & \rho\left(x^3, r_1^{(b)}\right) \end{array} \right]$$

is the same as in VFR. The sample of utility values will be different however, as it relies on $\bar{\bar{v}}_1^{(j)}$. The sample of utility values associated to decisions x^i at time $t = 0$, simulation paths (j), and $\bar{\bar{v}}_1^{(j)}$ is derived from

$$\begin{aligned} U_0(x^i, r_1^{(j)}, \bar{\bar{v}}_1^{(j)}) &= \left(\rho\left(x^i, r_1^{(j)}\right)\right)^{1-\gamma} \bar{\bar{v}}_1^{(j)}, \\ &= \frac{\left(1 \times \left(x^i r_1^{(j)} + R_f\right) \times \left(\bar{\bar{x}}_1^{(j)} r_2^{(j)} + R_f\right)\right)^{1-\gamma}}{1-\gamma}. \end{aligned} \tag{13}$$

Compare this with equation (11) in the VFR variant. The second expression on the right-hand side is not attractive numerically, but conceptually explicit: it is the utility derived from a dollar at $t = 0$, which grows in the first and second periods according to the returns associated to path (j), when invested in portfolios determined by the grid point x^i (for the first period) and path-j optimal portfolio $\bar{\bar{x}}_1^{(j)}$ (for the second period). Note how the independence of wealth (brought by the CRRA utility) is important here: path-j optimal portfolio $\bar{\bar{x}}_1^{(j)}$ is optimal irrespective of the $t = 0$ decision (which does influence the wealth at $t = 1$).

The complete sample appears as the left-hand side of the linear system

$$\left[\begin{array}{c} U_0\left(x^1, r_1^{(a)}, \bar{\bar{v}}_1^{(a)}\right) \\ U_0\left(x^2, r_1^{(a)}, \bar{\bar{v}}_1^{(a)}\right) \\ U_0\left(x^3, r_1^{(a)}, \bar{\bar{v}}_1^{(a)}\right) \\ U_0\left(x^1, r_1^{(b)}, \bar{\bar{v}}_1^{(b)}\right) \\ U_0\left(x^2, r_1^{(b)}, \bar{\bar{v}}_1^{(b)}\right) \\ U_0\left(x^3, r_1^{(b)}, \bar{\bar{v}}_1^{(b)}\right) \end{array} \right] = \left[\begin{array}{ccc} 1 & x^1 & \left(x^1\right)^2 \\ 1 & x^2 & \left(x^2\right)^2 \\ 1 & x^3 & \left(x^3\right)^2 \\ 1 & x^1 & \left(x^1\right)^2 \\ 1 & x^2 & \left(x^2\right)^2 \\ 1 & x^3 & \left(x^3\right)^2 \end{array} \right] \left[\begin{array}{c} \beta_1 \\ \beta_2 \\ \beta_3 \end{array} \right] + \left[\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{array} \right].$$

(Let us note again that we do not regress on the dividend because we are at time $t = 0$). This system can be solved by least-squares, providing a regression surface identified by the parameters $\bar{\bar{\beta}}_1, \bar{\bar{\beta}}_2, \bar{\bar{\beta}}_3$, and which is optimized to find $\bar{\bar{x}}_0$

$$\bar{\bar{x}}_0 = \underset{0 \le x \le 1}{\arg\max}\ \bar{\bar{\beta}}_1 + \bar{\bar{\beta}}_2 x + \bar{\bar{\beta}}_3 (x)^2.$$

Since we assume the same value of the state variable at $t = 0$ for both paths, there is, as in VFR, only one optimization to do, which yields $\bar{\bar{x}}_0$. The realized values at $t = 0$ can finally be computed,

$$\bar{\bar{v}}_0^{(a)} = U_0\left(\bar{\bar{x}}_0, r_1^{(a)}, \bar{\bar{v}}_1^{(a)}\right) \qquad \text{and} \qquad \bar{\bar{v}}_0^{(b)} = U_0\left(\bar{\bar{x}}_0, r_1^{(b)}, \bar{\bar{v}}_1^{(b)}\right). \tag{14}$$

It is important to note that, unlike the values obtained at $t = 0$ with VFR, the PWR function values $\bar{\bar{v}}_0^{(a)}$ and $\bar{\bar{v}}_0^{(b)}$are not independent of the sample path with which they are computed. To remove this dependency, we can compute the average of the $\bar{\bar{v}}_0^{(i)}$ over all n_r sample paths i.e.

$$\bar{\bar{v}}_0 = \frac{1}{n_r} \sum_{j=1}^{n_r} \bar{\bar{v}}_0^{(j)}.$$

This last quantity can be used as an estimate of a point on the value function at $t = 0$.

It is important to note that PWR works *only* because the CRRA utility brings along an independence of the optimal policy to the wealth level. With other utility functions (where wealth matters), the PWR variant of our algorithm does not apply, only the VFR variant. Thus the choice of CRRA utility is not trivial nor is it specific to this paper; Garlappi and Skoulakis (2009) is also written exclusively on a CRRA case.

The next section gives the algorithm in its general form for the two variants.

4 Simulation-and-regression algorithms descriptions

This section provides a complete description of the portfolio optimization algorithm in both its "recursion on value function" and its "recursion on weights" variants.

4.1 Simulation-and-regression algorithms descriptions

A first set of operations, common to both variants, does the preliminary work. A second set of operations does the recursive work. The algorithm applies to the following setup:

- The portfolio problem is that of Section 2; we want to solve the Bellman equation (5).
- The notation for sample utility values U_t, introduced in Section 3, generalizes naturally to:

$$U_t\left(x, r_{t+1}^{(j)}, v_{t+1}^{(j)}\right) \quad \triangleq \quad \left(x\, r_{t+1}^{(j)} + R_f\right)^{1-\gamma} v_{t+1}^{(j)}.$$

  It represents a utility value associated to decision x at time t, simulation path (j), and a generic value $v_{t+1}^{(j)}$ taken to represent either $\bar{v}_{t+1}^{(j)}$ or $\bar{\bar{v}}_{t+1}^{(j)}$, as specified in the algorithm below. Recall that the "bar" and "double bar" versions are used for the Value Function Recursion and the Portfolio Weight Recursion, respectively. For notational simplicity, explicit references to the maturity time T are avoided, but U_t should always be understood as a utility generated by managing the portfolio from time t to time T.
- The basis used for the regression surfaces may for example contain quadratic and linear monomials for the dividend and the risky asset weight, the cross-product of the dividend and the risk asset weight, and a constant; the size of the basis is then $n_b = 6$. If monomials of order 3 and 4 are added (for the dividend and the risky asset weight), then $n_b = 10$.

Below, steps that are specific to the Value Function Recursion (VFR) and Portfolio Weights Recursion (PWR) are identified as such (following the the VFR (or PWR) steps describes the VFR (or PWR) variant).

I- Preliminary operations: Weight grid formation and Monte Carlo simulations.

1) Form portfolio weights grid x^i Compute a portfolio weight grid of n_x values x^i for $i = 1$ to n_x. The grid contains the weights associated to the risky asset; the corresponding weights for the risk-free asset are implicit. The grid can be built to satisfy upper and lower bounds on the risky asset, as well as bounds on the risk-free asset.

2) Simulate paths of returns $r_t^{(j)}$ and dividends $d_t^{(j)}$ Simulate n_r paths of returns and dividends, $r_t^{(j)}$ and $d_t^{(j)}$ for $j = 1, \ldots, n_r$ and $t = 1, \ldots, T$.

II- Recursion operations, backward through time.

Begin Time loop, from $t = T - 1$ to 0 :

1) Generate a sample of $n_x \times n_r$ returns for the period between t and $t+1$, using the simulated returns and the discretized portfolio weights:

$$x^i r_{t+1}^{(j)} + R_f \qquad \text{for } i = 1 \text{ to } n_x \text{ and } j = 1 \text{ to } n_r.$$

2) Generate a sample of $n_x \times n_r$ values

(VFR): $U_t(x^i, r_{t+1}^{(j)}, \bar{v}_{t+1}^{(j)})$ for $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_r$. *For the special case of* $t = T-1$, let $\bar{v}_T^{(j)} = \frac{1}{1-\gamma}$.

(PWR): $U_t(x^i, r_{t+1}^{(j)}, \bar{\bar{v}}_{t+1}^{(j)})$ for $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_r$. *For the special case of* $t = T-1$, let $\bar{\bar{v}}_T^{(j)} = \frac{1}{1-\gamma}$.

3) Construct $n_x \times n_r$ basis vectors $B_{i,j}$, each of length n_b. Each vector $\mathbf{B}_{i,j}$ is associated to a pair $\left(x^i, d_t^{(j)}\right)$ for $i = 1$ to n_x and $j = 1$ to n_r.

4) Regress the dependent values found in step 2 on the independent basis vectors $\mathbf{B}_{i,j}$ to obtain an n_b-sized vector of estimated regression coefficients (one per basis).

(VFR): Call the coefficients $\bar{\boldsymbol{\beta}}_t$ and the corresponding regression surface $L_t^{\bar{\boldsymbol{\beta}}_t}(x, d)$.

(PWR): Call the coefficients $\bar{\bar{\boldsymbol{\beta}}}_t$ and the corresponding regression surface $L_t^{\bar{\bar{\boldsymbol{\beta}}}_t}(x, d)$.

5) Optimize the regression surfaces $L_t^{\bar{\boldsymbol{\beta}}_t}(x, d)$ (for VFR) or $L_t^{\bar{\bar{\boldsymbol{\beta}}}_t}(x, d)$ (for PWR) with respect to the portfolio weight x for each of the $j = 1, \ldots, n_r$ paths, i.e. :

$$\max_{x_{\min} \le x \le x_{\max}} L_t^{\bar{\boldsymbol{\beta}}_t}\left(x; d_t^{(j)}\right) \qquad \text{or} \qquad \max_{x_{\min} \le x \le x_{\max}} L_t^{\bar{\bar{\boldsymbol{\beta}}}_t}\left(x; d_t^{(j)}\right).$$

Under VFR, call the solution $\bar{x}_t^{(j)}$ and call the value of the objective $\bar{v}_t^{(j)}$; under PWR, call the solution $\bar{\bar{x}}_t^{(j)}$ (and there is no use for the value of the objective).

6) (PWR) Compute the n_r values $\bar{\bar{v}}_t^{(j)}$ for the optimal weights $\bar{\bar{x}}_t^{(j)}$:

$$\begin{aligned} \bar{\bar{v}}_t^{(j)} &= U_t(\bar{\bar{x}}_t^{(j)}, r_{t+1}^{(j)}, \bar{\bar{v}}_{t+1}^{(j)}), \\ &= \left(\bar{\bar{x}}_t^{(j)} r_{t+1}^{(j)} + R_f\right)^{1-\gamma} \bar{\bar{v}}_{t+1}^{(j)}. \end{aligned}$$

End **Time loop.**

Notes on the algorithm:

- The regression surface in step 4 is an approximate representation of the expected utility when taking position x at time t, conditional on the dividend information available at that time.
- At the completion of the algorithm, the optimal policy is defined by the set of regression parameters $\boldsymbol{\beta}_t$, which describe the surface to optimize over the decision variable.
- In step 5, provided all paths were started with the same dividend, there is only one optimization to do for $t = 0$, the solution being the same for all paths j. The expected utility at time T, when following the optimal policy, is given by $\bar{v}_0 = \bar{v}_0^{(j)}$ (i.e. for any j).
- In step 6, at $t = 0$, $\bar{\bar{v}}_0^{(j)}$ is not the same for all paths, unlike in the VFR case. The average over j is thus used as an estimate of the value function level at $t = 0$: $\bar{\bar{v}}_0 = \frac{1}{n_r} \sum_{j=1}^{n_r} \bar{\bar{v}}_0^{(j)}$.

4.2 Forward-pass values

The above algorithms each yield a policy as well as, a value associated to time 0 and the exogenous state at time 0, to which we now refer as the "backward-pass" or "first-pass" value, denoted $\bar{v}_0$ and $\bar{\bar{v}}_0$ for respectively VFR and PWR. It is also often called the "in-sample" value.

The backward-pass value is prone to biases, as discussed in the next section, and to find what a policy can achieve in an out-of-sample setting, we must run this policy on a set of independent paths of the exogenous state variable(s), and average the results.

That is, after establishing an optimal policy with a first sample of returns and dividends, forget the first sample, generate a second, independent sample of size n, apply the optimal policy from time $t = 0$ through $t = T - 1$ for each path in the sample, compute the utilities of the final wealths, and average those final utilities (over all paths).

Note that for the two simulation-regression variants, "applying the optimal policy" means performing an optimization (that of step 5 of the algorithm on page 9) for each point in time and each path. For the quadrature approach, the optimal policy already exists for each point of a grid of dividend yields; here "applying the optimal policy" for a dividend that is not on the grid means interpolating between divident yield values that are on the grid. In this paper, we used linear interpolation.

More formally, for path j, if x_t^* represents the optimal weight under one of the three policies given the information of path j until time t, then the final wealth is

$$W_T^{(j)} = \prod_{t=0}^{T-1} \left(x_t^* r_{t+1}^{(j)} + R_f \right),$$

and the forward-pass value (for that policy) is

$$v_{\text{fwd}} = \frac{1}{n} \sum_{j=1}^{n} u\left(W_T^{(j)} \right)$$

We call this average the "forward" or "second-pass" value; it is also an "out-of-sample" value, in opposition to the "in-sample" value. The value derived through this forward pass often has better properties than the backward-pass value. We note that while the idea of a forward-pass value almost goes without saying in some application areas, it is not necessarily so in option pricing, most likely for historical reasons.

We choose not to add to the already cluttered notation, and always refer to the forward-pass values as v_{fwd}; the underlying policy will be clear from the context or unimportant.

Note that a forward pass can equally well be applied to an approach that is not based on simulations, as illustrated with the quadrature approach that we use as benchmark, see Section 6.

The backward-pass and forward-pass values are central to the bias and numerical analyses of the next two sections. We start with an analysis of the biases, next.

5 Bias analysis

It is possible to identify two biases for our various estimates of the value at time 0: values associated to VFR or PWR, and backward passes or forward passes. Except where explicitely noted, in this section we refer to the backward-pass values, in opposition to the forward-pass values. What's more, the analysis applies equally to both VFR and PWR valued until Section 5.2, where we make a distinction. The arguments are extensions of those used in American option pricing; see for example Glasserman (2004), chapter 8.

5.1 High biases, low biases

We derive two biases, one high and one low, but establish beforehand some notation.

Notation. Let us recall our Bellman equation (5)

$$v_t\left(d_t\right) = \max_{x_t} \mathbb{E}_t\left[\left(x_t r_{t+1} + R_f\right)^{1-\gamma}\ v_{t+1}\left(d_{t+1}\right)\right]$$

and introduce the following notation for the imbedded conditional expected value:

$$L_t(x,d) \triangleq \mathbb{E}_t\left[\left(x_t r_{t+1} + R_f\right)^{1-\gamma}\ v_{t+1}\left(d_{t+1}\right)\right],$$

so that $v_t\left(d_t\right) = \max_{x_t} L_t(x,d)$.

We need to clearly identify the statistical estimates and do so with a hat (e.g. $\hat{v}_t$), the underlying variant (VFR or PWR) being irrelevant or clear from the context. This is a slight abuse of notation, as in earlier sections, estimates did not have hats to avoid cumbersome cases like $\hat{\bar{v}}_t$ and $\hat{\bar{\bar{v}}}_t$.

Also, the expression $L_t^{\boldsymbol{\beta}_t}(x,d)$ denotes a *parametric approximation* of $L_t(x,d)$ as a sum of $\boldsymbol{\beta}_t$-weighted bases, while the *estimated* parametric approximation is denoted $L_t^{\hat{\boldsymbol{\beta}}_t}(x,d)$, where $\hat{\boldsymbol{\beta}}_t$ is determined by least squares. In other words, in the case of $L_t^{\boldsymbol{\beta}_t}(x,d)$, $\boldsymbol{\beta}_t$ is undetermined, while in the case of $L_t^{\hat{\boldsymbol{\beta}}_t}(x,d)$, it relies implicitly on a statistical sample. From this, we naturally introduce $\hat{v}_t\left(d_t\right)$ to denote an estimated value:

$$\hat{v}_t\left(d_t\right) = \max_{x_t}\ L_t^{\hat{\beta}_t}(x,d) \tag{15}$$

It is implied, though not fully explicit in the notation, that $L_t^{\boldsymbol{\beta}_t}(x,d)$ relies on values v_{t+1} at $t+1$ that are of (undetermined) parametric origin, and $L_t^{\hat{\boldsymbol{\beta}}_t}(x,d)$ relies on values $\hat{v}_{t+1}$ at $t+1$ that are of *estimated,* parametric origin.

High bias. Since $\hat{v}_t\left(d_t\right)$ is an estimated value, relying on some sample, we may wonder about the expected value of this estimate. Let us first assume that there exists a β^* such that $L_t(x,d)$ can be perfectly represented by the parametric form: $L_t(x,d) = L_t^{\beta_t^*}(x,d)$; this assumption is relieved later.

We find that under Jensen's inequality,

$$\begin{aligned}\mathbb{E}_t\left[\hat{v}_t(d_t)\right] &= \mathbb{E}_t\left[\max_{x_t}\ L_t^{\hat{\beta}_t}(x,d)\right] \\ &\geq \max_{x_t} \mathbb{E}_t\left[L_t^{\hat{\beta}_t}(x,d)\right]\end{aligned}$$

Now if

$$\mathbb{E}_t\left[L_t^{\hat{\beta}_t}(x,d)\right] = L_t^{\beta_t^*}(x,d) \tag{16}$$

then the estimate $\hat{v}_t\left(d_t\right)$ is high-biased. It may or may not be that (16) holds for finite samples. In our numerical tests, a high bias is quite clear for the VFR variant.

This high bias can be removed, as suggested in the option pricing context by Broadie and Glasserman (1997, 2004), by the forward pass discussed in Section 4.2. The resulting sample mean utility is an unbiased estimate (of the utility this policy can generate).

Low bias. The above assumption about the parametric approximation, $L_t(x,d) = L_t^{\beta_t^*}(x,d)$, is usually too strong. As a result, the maximization in (15) is performed on an inaccurate objective function, so that $\hat{v}_t\left(d_t\right)$ is inaccurate; then at time step $t-1$, the policy implicit in the β_{t-1} will also be inaccurate, as it relies on (inaccurate) $\hat{v}_t\left(d_t\right)$. Inaccurate policies can only be suboptimal, inducing a low-bias in the estimated values.

Overall bias. The overall bias in the estimated values would depend on the use or not of a forward pass with an independent sample, for high-bias removal. Without a forward pass, the bias is uncertain and depends on the relative magnitudes of the high and low bias. With a forward pass, we expect a low-biased estimate. Of course, identifying (here, two) sources of biases is only a preliminary analysis; there could be other sources, and the magnitudes of each is uncharted. However, our numerical results tend to support this basic analysis, see Section 6.

5.2 Bias back-propagation

Given the recursive nature of dynamic programming, it is worthwhile to consider the propagation of the biases backward through the time steps. As mentionned, $L_t^{\hat{\boldsymbol{\beta}}_t}(x,d)$ and thus $\hat{v}_t\,(d_t)$ rely on values $\hat{v}_{t+1}\,(d_{t+1})$ that were themselves estimated. Here we need to distinguish between the VFR and PWR variants of the algorithm. For the VFR variant, any *high* bias in $\hat{v}_{t+1}$ propagates back as a *high* bias to $\hat{v}_t$ as long as $(x_t r_{t+1} + R_f)^{1-\gamma}$, which multiplies $\hat{v}_{t+1}$ in the Bellman equation, is not negative too often nor too much, which is quite intuitively the case. Also, any error in the values $\hat{v}_{t+1}$ can turn into a suboptimal policy at t, back-propagating low biases as well, so VFR propagates both biases backward.

The analysis for the PWR variant is different. The use of policy-based utilities (step 6 of the algorithm) means that the high bias does not back-propagate through the recursion; the suboptimality bias (low bias) may still propagate. The rate at which propagating biases will increase with the time steps is unknown.

This analysis of the biases, opens the door to interesting observations on the numerical results, which we examine next.

6 Numerical results

We examine here numerical results obtained with our simulation-and-regression approach, in both its VFR and PWR variants. The results are compared to those obtained with a benchmark Gauss-Hermite quadrature algorithm. We present the data in Section 6.1, and the details of the algorithms in Section 6.2. We analyze our numerical results in the following sections. Section 6.3 discusses backward-pass policies at time 0, while 6.4 discusses the same policies at other times in points; backward-pass values results are also briefly presented. The important Section 6.5 analyzes the backward-pass and forward-pass values, transformed in certainty equivalents.

6.1 Experimental design: Data

We consider a portfolio with two securities, one risky and one risk-free. The risky asset data is taken from the US Index Database 2014 from the Center for Research in Security Prices (CRSP). We use the value-weighted stock index formed with stocks from the NYSE, AMEX and NASDAQ markets. The log-dividend yield is defined as the standardized value of $\ln(1 + D_t/M_t)$ where D_t is the sum of the dividend levels for the 12 months prior to t and M_t is the market value of the securities in the index.[3] The risk-free rate is the 30-day bill return taken from the US Treasuries Database 2014 from CRSP. Using monthly data from March 1928 to December 2013, we obtain the following parameter estimates for our model (1)–(2) of Section 2, with t-statistics in parenthesis:

$$\begin{aligned} R_{t+1} &= \underset{(1.3998)}{0.0024} + \underset{(1.9668)}{0.0033}\, d_t + e_{r,t+1}, \\ d_{t+1} &= \underset{(-0.2516)}{-0.0015} + \underset{(165.42)}{0.9819}\, d_t + e_{d,t+1}, \\ cov\,(e_{r,t}\ e_{d,t}) &= \begin{bmatrix} 0.0030 & -0.0090 \\ -0.0090 & 0.0366 \end{bmatrix}. \end{aligned}$$

In all cases, we used $R_f = 1.0025$.

6.2 Experimental design: Algorithms

To establish a benchmark policy and compute backward-pass values, we use standard dynamic programming with Gauss-Hermite quadrature integration for the expected values. The algorithm, which is provided in

[3] Even for a model of monthly returns, it is this standardized, annual yield that has been used as predictor in the literature (see Section 2).

Appendix B, uses $n_q = 12$ quadrature points, a grid of dividend yields with $n_d = 200$ points, linear interpolations between points on the dividend grid, and a golden section search algorithm for the optimizations with minimum and maximum values of 0.0 and 1.0 for the portfolio weights (a "no short sale" constraint).

For the regression-and-simulation approach (both variants), we use a Latin Hypercube sampling simulation approach for the correlated Gaussian random shocks.[4] An equally-spaced grid of $n_x = 51$ portfolio weights with minimum and maximum values of 0.0 and 1.0 is used, and the optimizations are performed with a golden section search algorithm. The number of paths used to derive a policy (the backward pass) is $n_r = 100,000$ for all results except the graphical results provided in Figures 1 to 4 (see Section 6.4) which were obtained with $n_r = 5000$.

The out-of-sample test of Section 6.5 was done with a sample of size $n = 1,000,000$, independent of the policy-building sample, and with the same paths for the three policies to be tested. The forward pass is much lighter numerically than the backward pass, so using a large number of paths causes no difficulty.

To illustrate the memory treatment and computing times, consider that in a run-of-the-mill test, that of Section 6.3, there are 5,100,000 simulated dependent variable values and 5,100,000 vectors of independent variables entering each call to a regression procedure. This can create computer memory problems. To avoid them, the matrices of the normal equation associated with each regression are built using summations. More specifically, for the generic linear system $y = x^\top \beta + e$ where y and e are scalars and x, and β are $k \times 1$ vectors, given $n_r n_x$ observations of y and x, the ordinary least squares estimate of the coefficients is computed with

$$\widehat{\beta} = \left(\sum_{i=1}^{n_r n_x} x_i x_i^\top\right)^{-1} \left(\sum_{i=1}^{n_r n_x} x_i y_i\right).$$

Computing times are quite reasonable, and either variant (VFR and PWR) requires a similar effort. For example, for a portfolio choice problem with 100,000 sample paths, $n_x = 51$ portfolio weights, bases up to power 4 and a maturity of 24 months (24 periods), the computing time is just above 30 seconds, of which about one second is spent on optimizations (the vast majority of the effort is spent on the regressions). Although the number of constrained optimizations is quite large, this task accounts for little because of the simplicity of the problem examined here. We used a single cpu on a standard laptop computer (CRRA problems do not parallelize as well as problems where wealth matters). The main code is Matlab-based, with crucial procedures coded in C.

Note that a cruder discretization of portfolio weights can be used with little impact on the quality of the solutions and a good impact on the computation times; with $n_x = 11$, the problem just mentionned runs in 11 seconds, or in 5 seconds with $n_x = 11$ and $n_r = 50,000$.

In comparison, the benchmark quadrature procedure for the same problem, with $n_q = 12$ nodes and $n_d = 200$ dividend levels, runs in 5 seconds.

We believe it is fair to say that the computational effort of quadrature approaches does not scale as well with the dimension of the problem, as simulation-based approaches do. The problem considered in this paper, which is one-dimensional, of course does not display this comparative advantage of simulation-and-regression approaches; see however Delage, Denault and Simonato (2014).

6.3 Results: Optimal weights at $t = 0$

Table 1 presents optimal portfolio weights and (backward-pass) value function levels at $t = 0$ for different risk aversion and maturities, for the benchmark, VFR and PWR algorithms. Three different levels of the initial dividend yield d_0 are examined (low, mean and high) corresponding to minus one, zero, and one unconditional standard deviation $\left(\frac{\Sigma_{2,2}}{1-(b_d)^2}\right)^{\frac{1}{2}}$ from the unconditional mean $\frac{a_d}{1-b_d}$. (Recall that the series itself was standardized to a mean of 0 and a standard deviation of 1).

[4] See Section 4.4 of Glasserman (2004) for a description of this approach.

Table 1: Optimal portfolio weights and backward-pass value function levels

	Low d_0			Mean d_0			High d_0		
	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$
					$T = 24$				
					Gauss-Hermite quadrature				
$\tilde{x}_0$	0.0289	0.0155	0.0106	0.2835	0.1449	0.0973	0.5422	0.2765	0.1856
$\tilde{v}_0$	-0.1941	-0.0638	-0.0304	-0.1849	-0.0603	-0.0286	-0.1667	-0.0536	-0.0253
					Value function recursion				
$\bar{x}_0$	0.0284	0.0118	0.0049	0.2848	0.1425	0.0893	0.5448	0.2786	0.1818
	(4e-04)	(4e-04)	(1e-03)	(1e-03)	(1e-03)	(1e-03)	(1e-03)	(2e-03)	(3e-03)
$\bar{v}_0$	-0.1939	-0.0636	-0.0302	-0.1847	-0.0600	-0.0284	-0.1666	-0.0532	-0.0250
	(8e-05)	(3e-05)	(5e-05)	(1e-04)	(8e-05)	(8e-05)	(1e-04)	(1e-04)	(1e-04)
					Portfolio weight recursion				
$\bar{\bar{x}}_0$	0.0293	0.0163	0.0137	0.2842	0.1452	0.0968	0.5413	0.2737	0.1847
	(2e-03)	(1e-03)	(1e-03)	(4e-03)	(2e-03)	(2e-03)	(5e-03)	(5e-03)	(3e-03)
$\bar{\bar{v}}_0$	-0.1941	-0.0638	-0.0304	-0.1853	-0.0605	-0.0288	-0.1682	-0.0542	-0.0257
	(7e-05)	(3e-05)	(1e-05)	(1e-04)	(5e-05)	(3e-05)	(2e-04)	(1e-04)	(5e-05)
					$T = 60$				
					Gauss-Hermite quadrature				
$\tilde{x}_0$	0.0622	0.0353	0.0246	0.3404	0.1792	0.1216	0.6249	0.3269	0.2212
$\tilde{v}_0$	-0.1259	-0.0261	-0.0079	-0.1114	-0.0225	-0.0068	-0.0894	-0.0174	-0.0051
					Value function recursion				
$\bar{x}_0$	0.0630	0.0280	0.0479	0.3498	0.1849	0.1824	0.6427	0.3665	0.5501
	(7e-04)	(7e-04)	(1e-01)	(2e-03)	(2e-03)	(1e-01)	(2e-03)	(4e-03)	(3e-01)
$\bar{v}_0$	-0.1249	-0.0251	-0.0069	-0.1100	-0.0212	-0.0054	-0.0884	-0.0156	-0.0030
	(1e-04)	(7e-05)	(7e-04)	(2e-04)	(1e-04)	(7e-04)	(2e-04)	(1e-04)	(1e-03)
					Portfolio weight recursion				
$\bar{\bar{x}}_0$	0.0621	0.0359	0.0242	0.3398	0.1777	0.1171	0.6233	0.3244	0.2161
	(4e-03)	(3e-03)	(6e-03)	(8e-03)	(7e-03)	(6e-03)	(1e-02)	(1e-02)	(1e-02)
$\bar{\bar{v}}_0$	-0.1259	-0.0261	-0.0080	-0.1117	-0.0227	-0.0070	-0.0906	-0.0177	-0.0054
	(1e-04)	(5e-05)	(2e-05)	(2e-04)	(8e-05)	(3e-05)	(3e-04)	(1e-04)	(5e-05)
					$T = 120$				
					Gauss-Hermite quadrature				
$\tilde{x}_0$	0.1078	0.0645	0.0456	0.4007	0.2185	0.1500	0.6981	0.3761	0.2570
$\tilde{v}_0$	-0.0550	-0.0051	-0.0007	-0.0445	-0.0040	-0.0005	-0.0320	-0.0026	-0.0004
					Value function recursion				
$\bar{x}_0$	0.1285	0.5880	1.0000	0.4490	1.0000	1.0000	0.7678	1.0000	1.0000
	(1e-03)	(3e-01)	(1e-16)	(2e-03)	(1e-16)	(1e-16)	(2e-03)	(1e-16)	(1e-16)
$\bar{v}_0$	-0.0521	-0.0023	958.7535	-0.0414	0.0049	168.9935	-0.0294	0.0070	57.5491
	(2e-04)	(6e-04)	(6e+02)	(9e-05)	(2e-03)	(9e+01)	(1e-04)	(7e-04)	(4e+01)
					Portfolio weight recursion				
$\bar{\bar{x}}_0$	0.1075	0.0644	0.0330	0.4116	0.2178	0.1376	0.6987	0.3807	0.2447
	(1e-02)	(2e-02)	(2e-02)	(1e-02)	(3e-02)	(3e-02)	(2e-02)	(3e-02)	(3e-02)
$\bar{\bar{v}}_0$	-0.0551	-0.0052	-0.0008	-0.0448	-0.0040	-0.0006	-0.0324	-0.0027	-0.0004
	(2e-04)	(5e-05)	(8e-06)	(2e-04)	(4e-05)	(1e-05)	(2e-04)	(5e-05)	(1e-05)

In the table, $\tilde{x}_0$ and $\tilde{v}_0$ are the portfolio weights and value function levels at $t = 0$ for the Gauss-Hermite benchmark values, obtained using the algorithm described in Appendix B with $n_q = 12$, $n_d = 200$ and linear interpolations. The values $\bar{x}_0$ and $\bar{v}_0$ are the averages of 20 estimated optimal portfolio weights and value function levels at $t = 0$ for the value function recursion variant, with standard deviations in parenthesis below. The values $\bar{\bar{x}}_0$ and $\bar{\bar{v}}_0$ correspond to $\bar{x}_0$ and $\bar{v}_0$ but for the portfolio weight recursion variant. For the simulation methods, $n_r = 100,000$ sample paths, and an equally spaced grid of $n_x = 51$ portfolio weights with minimum and maximum values of 0.0 and 1.0 were used. The values for d_0 are respectively -1.093906 (low), -0.082528 (mean) and 0.928851 (high). $R_f = 1.0025$.

For both variants of the regression-and-simulation approach, the numbers presented in the table are averages of 20 independent runs of the algorithms, each with 100,000 sample paths; this allows us to provide a standard deviation (in parentheses) for the optimal weights and value function levels.

Note first that although backward-pass value function levels are provided, we mostly analyse portfolio weights in this section, and defer the analysis of the values to Section 6.5.

A first thing to notice from Table 1 is that, for problems with short maturities and low risk aversions, both the VFR and the PWR variants give average portfolio weights and value function levels that are very close to the Gauss-Hermite benchmarks. However, for combinations of longer maturities and higher risk aversions, the results from the VFR show large departures from the benchmark values, while the PWR results remain close. It is important to notice that both variants share the same set of independent variables at each time step. Only the dependant variables, which can be interpreted as a random sample of value function levels, differ. Hence the relative deterioration in the precision can only be attributed to the deterioration in the quality of the sample obtained by each method. For higher values of the risk aversion coefficient, the value function is curvier and more difficult to approximate with a regression surface. Now recall that for the VFR variant, the sample is formed using regression surface values and the portfolio weight computed with this surface. The approximation quality of this surface is thus directly impacting the computations with this variant. As discussed in Section 5.2, these approximation errors also accumulate. This is not the case for the PWR variant which uses the regression surface only indirectly, through the computed weights, when creating the sample of dependant variable. Furthermore, because the approximated portfolio weights are bounded by the constraints, the approximation errors are also bounded.

A second point to notice from Table 1 is that the standard deviation of the estimated portfolio weights is much larger for the PWR variant. For example, for a maturity of 60 months, the standard deviation of the optimal weights computed with the PWR can be up to 5 times the one obtained with the VFR. Again, because both algorithmic variants use the same sets of independent variables, the source of these discrepancies should be linked to the dependent variables in the regressions. It is natural to think that the PWR could lead to larger variances of simulated dependent variables, since a simulated observation generated by this variant depends on all the returns along a sample paths (as shown in equation (13)), unlike the VFR variant, which only depends on the next-period return along a path. This larger variance of the dependent variables for the PRW variant leads to regression residuals with larger standard errors. As it is well known from the linear regression framework, the precision of the regression coefficients are functions of the residuals variance. Hence, the larger variability found for the portfolio weights under PWR can be directly linked to the larger variability of the dependent variable.

6.4 Results: Optimal weights and backward-pass value functions at various times

The previous table examined results at the initial time point ($t = 0$). For a better understanding of the performance of each variant, it is interesting to examine some of the results obtained at different time points within the recursions. For this purpose, Figures 1 to 4 plot the optimal risky asset allocations and backward-pass value function levels at different time points, as a function of the dividend yield for each method used in this study. While Figures 1 and 2 look at a case with a small risk aversion coefficient $\gamma = 5$, Figures 3 and 4 look at the large risk aversion case $\gamma = 15$. The figures plot the risky asset allocation and value function levels at $t = \{1, 10, 50, 59\}$ for the case $T = 60$ and $d_0 = 1$. For each simulation method, the plotted values are averages of 100 independent replications. Also plotted in each graph are the 5% and 95% percentiles obtained from the 100 replications, and the Gauss-Hermite benchmark values. For the simulation methods, each of the 100 independent replication uses $5,000$ sample paths to build the policies. In each graph, the dividend yield span is determined by the conditional expected values of the dividend yield at t, plus or minus two conditional standard deviations.

In the low-aversion case (Figures 1 and 2), the benchmark values are within the 5% and 95% bounds of the simulation-and-regression value for all cases except for the $t = 10$, VFR case.

As noticed earlier, the dispersion of the computed quantities with the VFR variant is smaller than the dispersion of the PWR as t approaches the current period. We also notice that the bounds are typically tighter

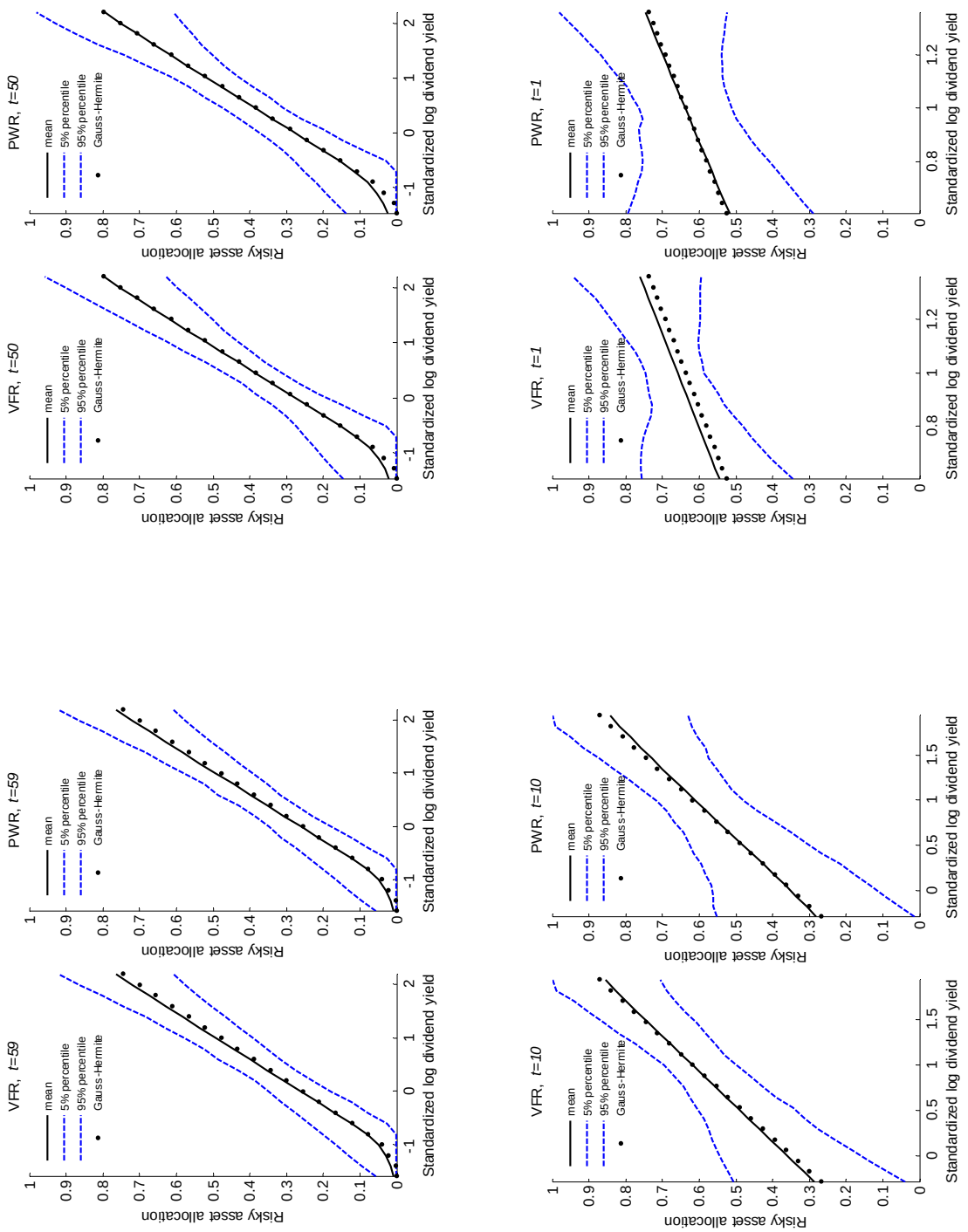

Figure 1: Risky asset allocation for $\gamma = 5$, $T = 60$

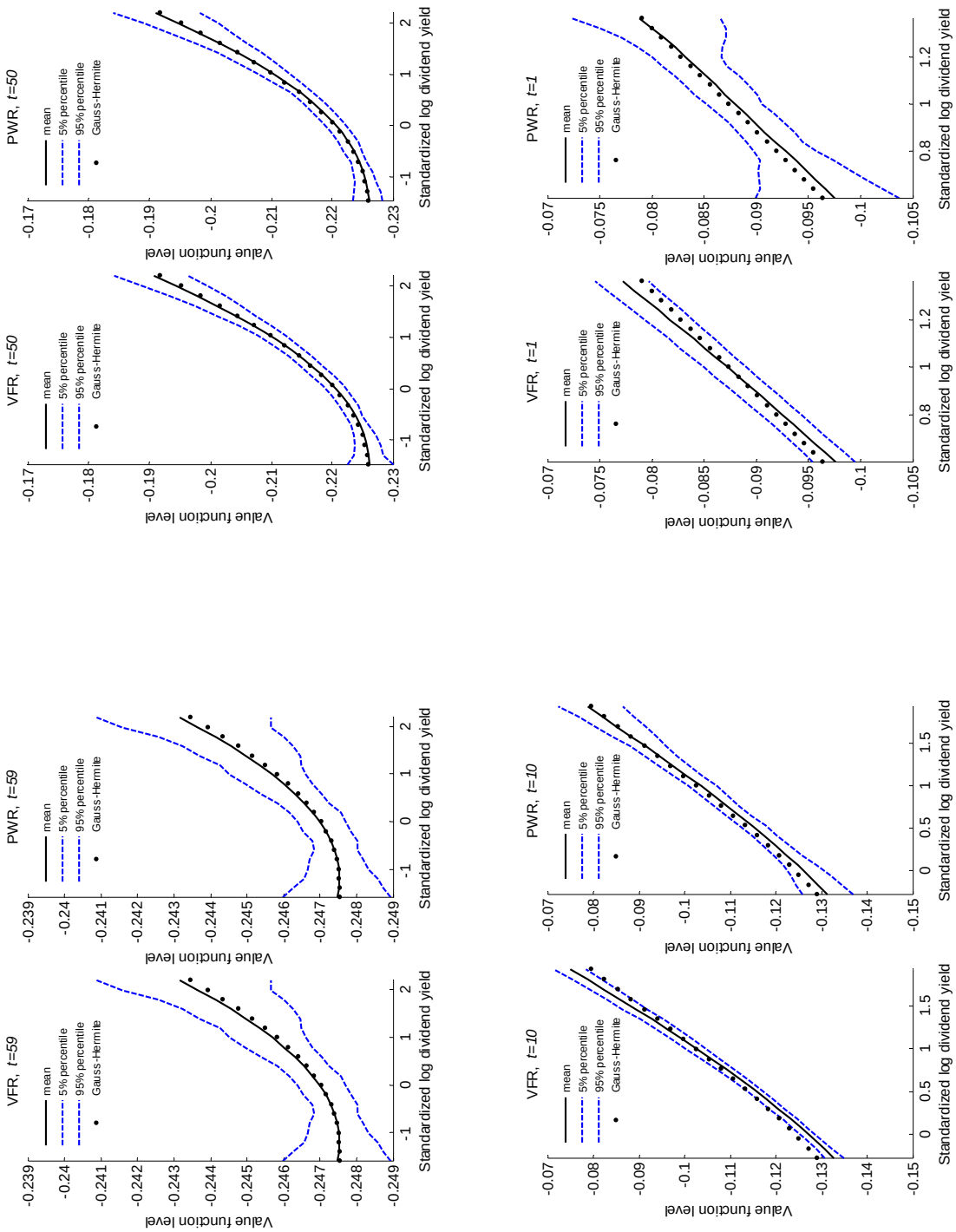

Figure 2: Backward-pass value function levels for $\gamma = 5$, $T = 60$

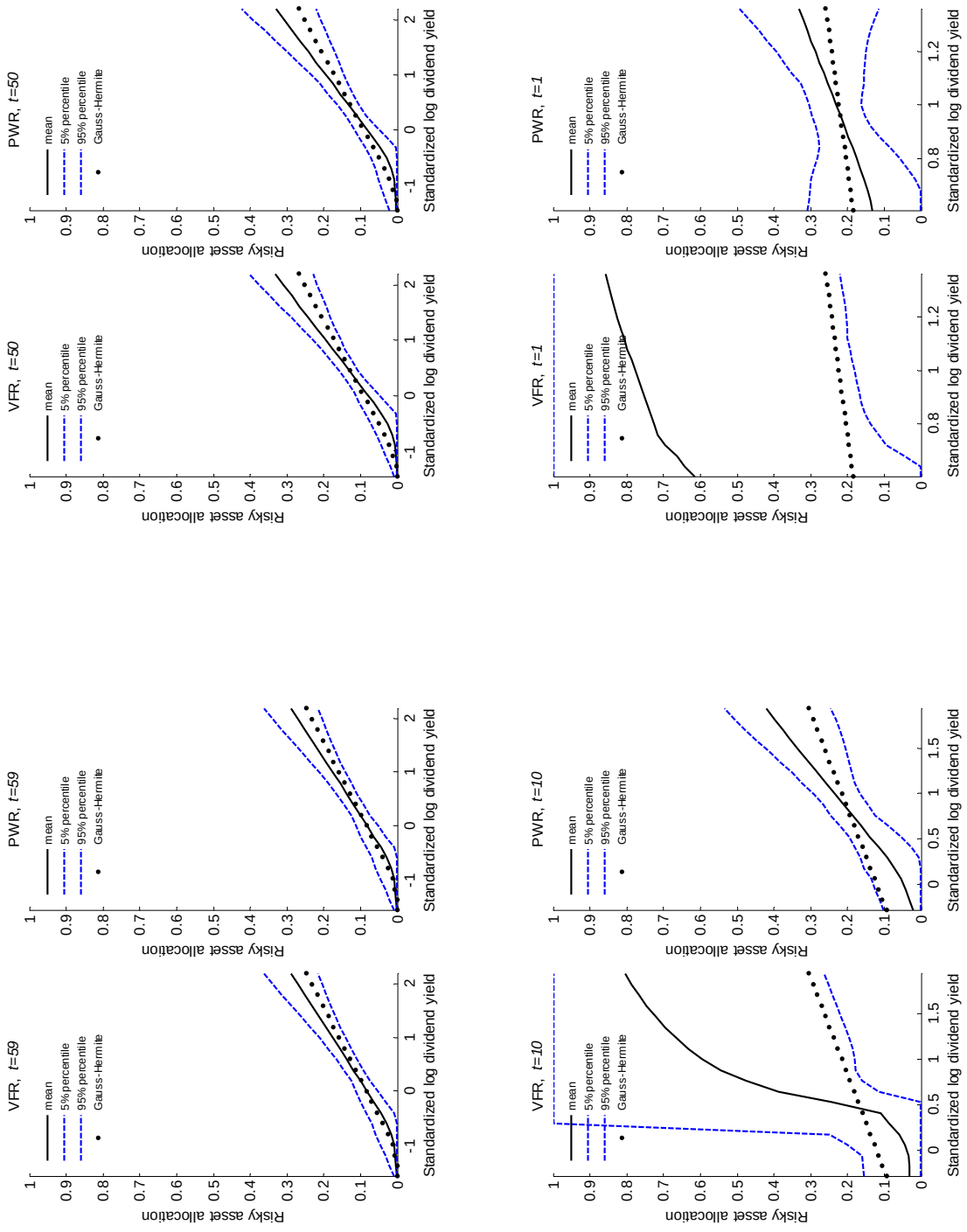

Figure 3: Risky asset allocation for $\gamma = 15$, $T = 60$

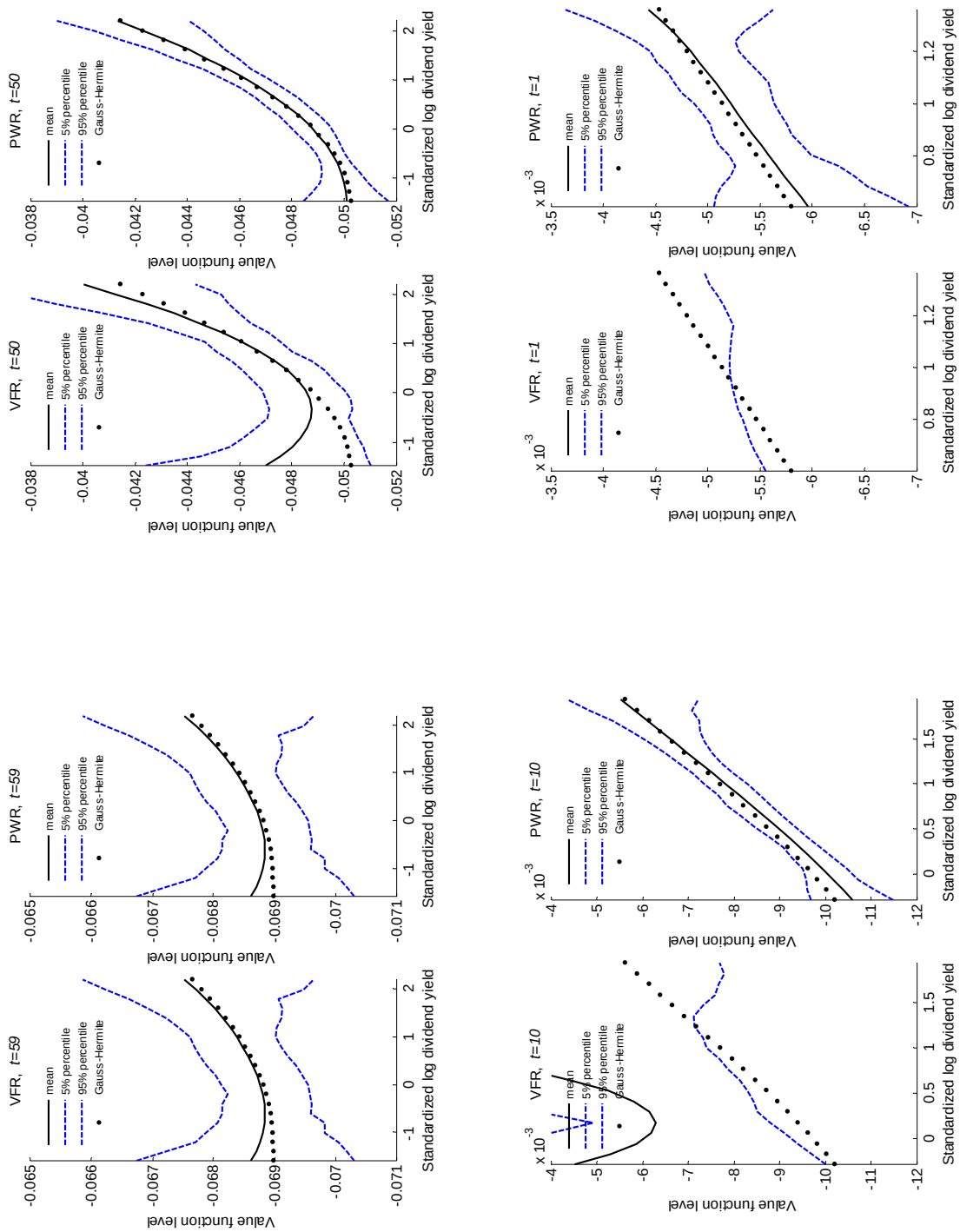

Figure 4: Backward-pass value function levels for $\gamma = 15$, $T = 60$

around the conditional expected dividend yield values (the center of the horizontal axis). This phenomenon is caused by the greater number of dividend yield paths which are close the conditional expected value, resulting in a smaller sample variability around these.

Considering the high-aversion case (Figures 3 and 4), a first difference is the clear deterioration of the results computed with the VFR variant. As seen in Figure 4, while both variants start with value function levels that are close to those of the benchmark at $t = 59$, we observe discrepancies at $t = 50$ for the VFR variant. As we approach the initial period, the average values for the VFR get further away from the benchmark as the errors accumulate.[5] For the PWR approach, the benchmark values remain close to the benchmarks, with reasonable percentile bounds. We notice that for the risky asset allocations, the average allocations are slightly overestimated for low dividend values and underestimated for higher values. However, near the conditional expected dividend value, the estimated proportions are close to those of the benchmark. Again, this can be explained by the greater presence of dividend yield paths that are close to this conditional expected dividend yield value.

6.5 Backward-pass vs forward-pass values: An out-of-sample test

The real test of a policy is an out-of-sample test, which corresponds to computing forward-pass values as discussed in Section 4.2.

In this section we compare VFR and PWR policies to the benchmark policy, both in terms of the backward-pass (i.e. in-sample) values and the forward-pass (i.e. out-of-sample) values. While value functions could be compared in themselves, we prefer to compare their more intuitive certainty equivalent returns, computed as follows.

Certainty equivalent returns *for the backward-pass values* are the inverse utility function applied to the backward-pass values, on an annualized basis:

$$\left[u^{-1}\left(\bar{v}_0^{(j)}\right)\right]^{12/T} - 1$$

for VFR; recall that $\bar{v}_0^{(j)}$ is the same for all j's. For PWR, we apply the same expression but with the average over j of all $\bar{v}_0^{(j)}$ replacing $\bar{v}_0^{(j)}$.

Certainty equivalent returns for the *forward-pass values* v_{fwd} of Section 4.2 are computed as

$$\left[u^{-1}\left(v_{\text{fwd}}\right)\right]^{12/T} - 1.$$

Of course, each of the three policies (quadrature and two variants of simulation-and-regression) yields a different v_{fwd}, so we can get three forward-pass certainty equivalents.

Table 2 displays the backward-pass and forward-pass certainty equivalent values for the benchmark, VFR and PWR variants, for a variety of parameter settings. As with the results of Table 1, three different levels of the initial dividend yield d_0 are examined for three levels of risk aversion and three time horizons.

Our analysis of the results goes as follows; we first comment on the methods without distinguishing between backward-pass and forward-pass values, then distinguish between the two afterwards. Note that in all cases the backward-pass and forward-pass *benchmark* values are very close (less than one basis point) so either can serve as target when we do not care to make the distinction; when we refer to a "better" or "worse" result, it is always with respect to this (very short) interval.

[5] For the VFR approach at $t = 1$, the average value function levels are located outside the graph since a unique scale is used for the vertical axis to facilitate comparisons.

Table 2: Annualized certainty equivalents, an out-of-sample study

	Low d_0			Mean d_0			High d_0		
	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$
					$T = 24$				
GHQ-F	0.03215	0.03132	0.03103	0.03839	0.03449	0.03316	0.05193	0.04136	0.03776
GHQ-B	0.03216	0.03132	0.03103	0.03840	0.03450	0.03316	0.05195	0.04137	0.03777
VFR-F	0.03215	0.03130	0.03100	0.03837	0.03446	0.03306	0.05192	0.04132	0.03755
VFR-B	0.03224	0.03149	0.03124	0.03858	0.03473	0.03354	0.05198	0.04178	0.03846
PWR-F	0.03215	0.03130	0.03098	0.03837	0.03447	0.03308	0.05191	0.04132	0.03762
PWR-B	0.03218	0.03130	0.03099	0.03813	0.03429	0.03293	0.05075	0.04079	0.03729
					$T = 60$				
GHQ-F	0.03483	0.03272	0.03198	0.04117	0.03604	0.03423	0.05263	0.04200	0.03826
GHQ-B	0.03488	0.03275	0.03200	0.04126	0.03609	0.03427	0.05278	0.04210	0.03833
VFR-F	0.03481	0.03261	0.03075	0.04115	0.03581	0.02769	0.05260	0.04154	0.03637
VFR-B	0.03538	0.03359	0.03422	0.04180	0.03736	0.04066	0.05330	0.04450	0.04173
PWR-F	0.03481	0.03264	0.03175	0.04114	0.03590	0.03371	0.05258	0.04180	0.03747
PWR-B	0.03493	0.03269	0.03181	0.04104	0.03578	0.03391	0.05215	0.04158	0.03752
					$T = 120$				
GHQ-F	0.03851	0.03474	0.03337	0.04400	0.03773	0.03542	0.05266	0.04244	0.03865
GHQ-B	0.03857	0.03477	0.03339	0.04408	0.03777	0.03545	0.05276	0.04248	0.03867
VFR-F	0.03846	0.02681	-0.04625	0.04393	0.02000	-0.02687	0.05259	0.02667	-0.01651
VFR-B	0.03992	0.04374	NaN	0.04599	NaN	NaN	0.05507	NaN	NaN
PWR-F	0.03846	0.03447	0.03243	0.04391	0.03733	0.03398	0.05252	0.04198	0.03659
PWR-B	0.03849	0.03453	0.03295	0.04390	0.03750	0.03484	0.05262	0.04236	0.03768

This table displays the annualized certainty equivalent returns computed with the benchmark Gauss-Hermite quadrature approach (GHQ), and the value function recursion (VFR) and portfolio weights recursion (PWR) simulation-and-regression variants. The algorithm settings (number of quadratures notes, discretization, values of d_0 etc.) are the same as in Table 1; the only new value is the number of paths in the forward simulation, which is $1,000,000$. Details on the underlying asset model are in Section 6.1, while details on the computations of the policies and of the certainty equivalents are provided in Sections 6.2 and 6.5. GHQ-B, VFR-B and PWR-B refer to the certainty equivalent returns obtained from the utility computed after the standard, policy-building, backward pass. GHQ-F, VFR-F and PWR-F refer to the C.E. returns obtained from the (average) utility computed with the out-of-sample test. Details are provided in Section 6.5. The parameters d_0, γ and T refer respectively to the initial dividend, the risk aversion parameter, and the number of months (and rebalancing periods) covered by the optimization.

First, and irrespective of the simulation-and-regression algorithm, a few stylised facts are observed:

- C.E. values decrease when risk aversion increases.
- C.E. values increase (on an annualized basis) when the number of months increases.
- C.E. values increase when the initial dividend rate increases, and this improvement is especially strong in the initial periods. Under high d_0 for example, the values hardly increase at all when passing from the $T = 24$ results to the $T = 120$ results; clearly, there is little room for improvement as "later" months are added, because the initial months are so profitable.
- C.E. values obtained by simulation-and-regression are closer to the benchmark C.E. when there are fewer periods.
- C.E. values obtained by simulation-and-regression are closer to the benchmark C.E. for lower risk aversion (lower values of γ).
- The initial dividend yield has little impact on the quality of the C.E. values obtained by simulation-and-regression (with respect to the benchmark).

- Small deviations from the true optimal policies do not lead to significant deviation in the corresponding utilities (or certainty equivalent): compare the deviations in weights ($\bar{x}_0$ and $\bar{\bar{x}}_0$ vs $\tilde{x}_0$) in Table 1 with the deviations in certainty equivalents of Table 2. This was pointed out in Cochrane (1989) and van Binsbergen and Brandt (2007) and is quite plain here as well. In many cases, deviations of one percent in optimal weight translate in a deviation of only one basis point in the certainty equivalent.

In fact, only the VFR variant ever incurs serious problems, and then only for the highest number of periods ($T = 120$) and high risk aversion ($\gamma = 10$ and $\gamma = 15$). We consider these values (and underlying policies) as unreliable and ignore them in the comments below. In all other cases of VFR, and in all cases of PWR, the results are good or very good.

Comparing backward-pass and forward-pass values provides a fertile ground for various observations. We refer the reader to Section 5 and note that the certainty equivalent transformation being monotone increasing, a high bias in the value function values remains a high bias in certainty equivalent values.

Given that the forward pass should remove the high bias in both VFR and PWR, we expect the backward-pass values to be higher than their respective forward-pass values. This is indeed observed for all VFR cases. The PWR results are a mixed bag, with a third of the cases showing a forward-pass value that is (mildly) higher that the backward-pass value; the values are so close however (for PWR) that it is difficult to distinguish "bias" from plain simulation-induced variability.

Compare now the VFR and PWR certainty equivalent values to the benchmark values (and in particular as T grows):

- The VFR backward-pass results are above the benchmark values, and this overshoot grows steadily with T, from a few basis points at $T = 24$ to twenty basis points and more at $T = 120$.
- The VFR forward-pass results are at or below the benchmark values, often just below. (Ignoring the unreliable cases.)
- The PWR results, both backward- and forward-pass, are almost always very close to the benchmark results, within a few basis points at most (the exception being the three extreme cases with $T = 120$, $\gamma = 15$). In consequence, any attempt at bias identification is rather perillous.

These observations support our backpropagation analysis (Section 5.2) in that we might expect the VFR backward-pass results to deteriorate as T grows, the PWR backward-pass results less so. (Recall that unless the backpropagation rates are known, no definite claims can be made)

What's more, and although we have no theoretical argument to sustain this, our numerical results indicate that the high bias of the VFR variant (present in the backward-pass) is larger than its low bias, which is itself quite small. For the PWR variant, our numerical tests cannot support any such statement.

In conclusion, our numerical results seem to indicate the following: VFR works well, but PWR works even better, and for VFR, forward passes are useful to remove a high bias that is more damageable (in terms of reaching the benchmark target) than the low bias that remains. For extreme cases like $\gamma = 15$, VFR can collapse if there are too many periods whereas PWR is remarkably stable.

7 Conclusion

In dynamic portfolio optimization, the issue of whether it is best to do value function recursion or portfolio weights recursion is considered for the simulation-and-regression algorithm of Delage, Denault, Simonato (2014). The issue itself was already taken up by van Binsbergen and Brandt (2007) and Garlappi and Skoulakis (2009) for their Taylor series-based algorithms. In option pricing, the issue has also been debated, in Tsitsiklis, Van Roy (2001), Longstaff, Schwartz (2001), and more recently in Stentoft (2014).

While van Binsbergen and Brandt (2007) argue in favor of portfolio weights recursion for their algorithm, Garlappi and Skoulakis (2009) note that a transformation of the value function brings accurate results when value function recursions are used. We stress that our algorithmic approach is fundamentally different and

as such our results cannot confirm nor infirm the conclusions of these two papers. Still, what we do find, is that with our algorithm, portfolio weights recursion works very well for all maturities and risk aversion coefficients, unlike the value function recursions which also work well but only with shorter maturities and lower risk aversions.

The issue of recursions on regressed values vs realized values applies to a much broader set of problems than the portfolio optimization problem we study here. We believe that we bring one more brick to the understanding of this issue.

A Bellman equation for the CRRA case

In the case of CRRA utility functions, the Bellman equation simplifies to a value function that *does not depend on wealth.* This standard result can be found for example in Brandt et al. (2005), and is provided here for completeness.

Using the law of iterated expectation and the definition of the CRRA utility function, the maximum expected utility of wealth at T given by equation (3) can be rewritten as:

$$V_0\left(W_0, d_0\right) = \max_{x_0} \mathbb{E}_0 \left[\max_{\{x_s\}_{s=1}^{T-1}} \mathbb{E}_1 \left\{ \frac{W_T^{1-\gamma}}{1-\gamma} \right\} \right].$$

Using the budget constraint (4), the above equation becomes:

$$V_0\left(W_0, d_0\right) = \max_{x_0} \mathbb{E}_0 \left[\max_{\{x_s\}_{s=1}^{T-1}} \mathbb{E}_1 \left\{ \frac{\left(W_0 \prod_{s=0}^{T-1} \left(x_s r_{s+1} + R_f\right)\right)^{1-\gamma}}{1-\gamma} \right\} \right],$$

which can be rewritten as:

$$V_0\left(W_0, d_0\right) = \max_{x_0} \mathbb{E}_0 \left[\left(W_0\left(x_0 r_1 + R_f\right)\right)^{1-\gamma} \times \max_{\{x_s\}_{s=1}^{T-1}} \mathbb{E}_1 \left\{ \frac{\prod_{s=1}^{T-1} \left(x_s r_{s+1} + R_f\right)^{1-\gamma}}{1-\gamma} \right\} \right].$$

Because of the homotheticity in wealth of the utility function, we can fix $W_0 = 1$ to obtain a Bellmann equation in function v_0 that depends only on past information, not wealth:

$$v_0\left(d_0\right) = \max_{x_0} \mathbb{E}_0 \left[\left(x_0 r_1 + R_f\right)^{1-\gamma} v_1\left(d_1\right) \right]$$

and $V_0\left(1, d_0\right) = v_0\left(d_0\right)$ with $v_T\left(d_T\right) = 1/\left(1-\gamma\right)$.

B Benchmark algorithm, Gauss-Hermite integration

To simplify the expressions, let us denote the restricted VAR system given in equations (1) and (2) as

$$\mathbf{y}_{t+1} = \boldsymbol{\mu}_t\left(d_t\right) + \mathbf{e}_{t+1}$$

where

$$\mathbf{y}_{t+1} = \begin{bmatrix} R_{t+1} \\ d_{t+1} \end{bmatrix}, \quad \boldsymbol{\mu}_t\left(d_t\right) = \begin{bmatrix} a_r + b_r d_t \\ a_d + b_d d_t \end{bmatrix}, \quad \mathbf{e}_t = \begin{bmatrix} e_{r,t+1} \\ e_{d,t+1} \end{bmatrix}.$$

The 2×2 covariance matrix of the error term vector, introduced in Section 2, is $\boldsymbol{\Sigma} = \mathbb{E}\left(\mathbf{e}_t \mathbf{e}_t^\top\right)$. In order to solve the portfolio problem, the conditional expected value appearing in the Bellman equation (5) is rewritten generically here as:

$$\mathbb{E}\Big[g\left(\mathbf{y}_{t+1}\right)\Big] = \int_{\mathbb{R}^2} g\left(\mathbf{y}_{t+1}\right) f\left(\mathbf{y}_{t+1}\right) d\mathbf{y}_{t+1}$$

where $f(\mathbf{y}_{t+1})$ is the bivariate normal density associated with vector $\mathbf{y}_{t+1}$, and where $g(\mathbf{y}_{t+1}) = (x_t r_{t+1} + R_f)^{1-\gamma}\ v_{t+1}(d_{t+1})$. Substituting the expression for the bivariate normal density, and using the following change of variable

$$\mathbf{y}_{t+1} = \sqrt{2}\boldsymbol{\Omega}\mathbf{z}_{t+1} + \boldsymbol{\mu}_t(d_t)$$

where we use the Choleski factorization $\boldsymbol{\Sigma} = \boldsymbol{\Omega}\boldsymbol{\Omega}^{\intercal}$, yields the following expression which is computable with Gauss-Hermite integration:

$$\mathbb{E}\Big[g(\mathbf{y}_{t+1})\Big] = \frac{1}{\pi}\int_{\mathbb{R}^2} g\left(\sqrt{2}\boldsymbol{\Omega}\mathbf{z}_{t+1} + \boldsymbol{\mu}_t(d_t)\right)\exp\left(-\mathbf{z}_{t+1}^{\intercal}\mathbf{z}_{t+1}\right)d\mathbf{z}_{t+1}.$$

Using this formulation for the conditional expected values, the recursive integration algorithm by Gauss-Hermite quadrature is then based on the following steps.

I- Preliminary operations: Dividend yield grid and quadrature nodes.

1) Compute quadrature nodes and weights

- Compute n_q univariate quadrature nodes $\mathbf{Z}$ and quadrature weights $\boldsymbol{\omega}$ (using any standard software).
- Form the $n_q^2 \times 2$ matrix $\widehat{\mathbf{Z}}$ of multivariate quadrature nodes, which is simply all possible pairs made with the univariate nodes. Do similarly with the weights to form the $n_q^2 \times 2$ matrix $\widehat{\boldsymbol{\omega}}$ of multivariate quadrature weights.
- Compute transformed quadrature nodes $\widetilde{\mathbf{Z}} = \sqrt{2}\widehat{\mathbf{Z}}\boldsymbol{\Omega}^{\top}$, and the $n_q^2 \times 1$ weights vector $\widetilde{\boldsymbol{\omega}}$, defined element-by-element as $\widetilde{\omega}_j = \widehat{\omega}_{j,1} \times \widehat{\omega}_{j,2}$.

2) Form dividend yield grid $\boldsymbol{\delta_{i,t}}$ The grid is uniformly spaced, has n_d points for each time step, but is different for each time step.

- Determine $\delta_{\min,t}$ and $\delta_{\max,t}$, the lower and upper values of the dividend yield grid at t. These values are determined from $t = 1$ to T with

$$\delta_{\min,t} = \mathbb{E}_0[d_t] - 5 \times \sqrt{\mathbb{V}_0[d_t]} \quad \text{and} \quad \delta_{\max,t} = \mathbb{E}_0[d_t] + 5 \times \sqrt{\mathbb{V}_0[d_t]}$$

  where $\mathbb{E}_0[d_t]$ and $\mathbb{V}_0[d_t]$ are, respectively, the conditional expected value and variance at $t = 0$ of d_t.

- Compute the dividend yield grid values for $t = 1$ to T with

$$\delta_{i,t} = \delta_{\min,t} + (i-1)\frac{(\delta_{\max,t} - \delta_{\min,t})}{(n_d - 1)} \quad \text{for } i = 1 \text{ to } n_d.$$

II- Recursion operations, backward through time.

Begin Time loop, from $t = T - 1$ to 0:

Begin Grid loop (dividend yield loop), for $i = 1$ to n_d:

1. Compute the n_q^2 returns and dividends

$$\widetilde{R}_{j,t+1} = a_r + b_r \delta_{i,t} + \widetilde{Z}_{j,1}, \qquad \widetilde{d}_{j,t+1} = a_d + b_d \delta_{i,t} + \widetilde{Z}_{j,2}.$$

2. If $t = T - 1$, $\widetilde{v}_{t+1}\left(\widetilde{d}_{j,t+1}\right) = \frac{1}{1-\gamma}$. This is the terminal condition.
   If $t < T - 1$, obtain values $\widetilde{v}_{t+1}\left(\widetilde{d}_{j,t+1}\right)$ by interpolation with the set of values

$$\left\{\delta_{i,t+1},\ \widetilde{v}_{t+1}\left(\delta_{i,t+1}\right)\right\}_{i=1}^{n_d}.$$

3. Compute the optimal portfolio weight and value function associated to $\delta_{i,t}$:

$$\widetilde{v}_t\left(\delta_{i,t}\right) \ = \ \max_{0 \le x \le 1} \ \frac{1}{\pi} \sum_{j=1}^{n_q^2} \left(x\left(e^{\widetilde{R}_{j,t+1}} - 1\right) + R_f\right)^{1-\gamma} \widetilde{v}_{t+1}\left(\widetilde{d}_{j,t+1}\right) \widetilde{\omega}_j \ .$$

End **Grid loop.**

End **Time loop.**

References

[1] Balduzzi, P., Lynch, A., 1999, Transaction costs and predictability: Some utility cost calculations, Journal of Financial Economics 52, 47–78.

[2] Barberis, N., 2000, Investing for the long-run when returns are predictable, Journal of Finance 55, 225–264.

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

[4] Brennan, M., Schwartz, E., Lagnado, R., 1997, Strategic asset allocation, Journal of Economic Dynamics and Control 21, 1377–1403.

[5] Broadie, M., Glasserman, P., 1997, Pricing American-style securities using simulation, Journal of Economic Dynamics and Control 21, 1323–1352.

[6] Broadie, M., Glasserman, P., 2004, A stochastic mesh method for pricing high-dimensional American options, Journal of Computational Finance 7(4), 35–72.

[7] Carriere, J.F., 1996, Valuation of the early-exercise price for options using simulations and nonparametric regression, Insurance: Mathematics and Economics 19, 19–30.

[8] Campbell, J., Viceira, L., 1999, Consumption and portfolio decisions when expected returns are time-varying, Quarterly Journal of Economics 114, 433–495.

[9] Cochrane, J., 1989, The sensitivity of tests of the intertemporal allocation of consumption to near-rational alternatives, American Economic Review 79, 319–337.

[10] CRSP, Center for Research on Security Prices, The University of Chicago Booth School of Business.

[11] Dammon, R., Spatt, C., Zhang, H., 2000, Optimal consumption and investment with capital gains taxes, Review of Financial Studies 14, 583–616.

[12] Delage, E., Denault, M., Simonato, J.-G., 2014, A simulation-and-regression approach for dynamic programming, and its application to portfolio choice, Les Cahiers du GERAD G–2014–42, technical report.

[13] Errais, E., Sadowsky, J., 2008, Valuing pilot projects in a learning by investing framework: An approximate dynamic programming approach, Computers and Operations Research 35, 90–112.

[14] Garlappi, L., Skoulakis, G., 2009, Numerical solutions to dynamic portfolio problems: The case for value function iteration using Taylor expansion, Computational Economics 33, 193–207.

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

[16] Glasserman, P., 2004, Monte Carlo Methods in Financial Engineering. Springer, New York.

[17] Koijen, R., Nijman, T., Werker, B., 2010, When can life cycle investors benefit from time-varying bond risk premia?, Review of Financial Studies 23, 741–780.

[18] Lynch, A., 2001, Portfolio choice and equity characteristics: Characterizing the hedging demands induced by return predictability, Journal of Financial Economics 62, 67–130.

[19] Longstaff, F., Schwartz, E., 2001, Valuing American options by simulations: A simple least squares approach, Review of Financial Studies 14, 113–148.

[20] Stentoft, L., 2014, Value function approximation or stopping time approximation: A comparison of two recent numerical methods for American option pricing using simulation and regression, Journal of Computational Finance 18(1), 65–120.

[21] Tsitsiklis, J., Van Roy, B., 2001, Regression methods for pricing complex American-style options, IEEE Transactions on Neural Networks 12, 694–703.

[22] van Binsbergen, J., Brandt, M., 2007, Solving dynamic portfolio choice problems by recursing on optimized portfolio weights or on the value function? Computational Economics 29, 355–367.

[23] Wang, S.-P., Judd, K.L., 2000, Solving a savings allocation problem by numerical dynamic programming with shape-preserving interpolation, Computers and Operations Research 27, 399–408.