Solution methods for DSGE models in continuous time: Application to a climate-economy model*

Stan Olijslagers

October 12, 2020

Abstract

We consider two solution methods to solve dynamic stochastic general equilibrium models in continuous time with Epstein-Zin preferences. The specific setting that we analyze is a stochastic endowment economy with disaster risk. We allow for an optimal policy setting, where the agent optimizes the value function over a set of controls. The first method directly solves the Hamilton-Jacobi-Bellman equation using a finite difference scheme for PDEs. To reduce the curse of dimensionality, sparse grid methods are used. The second method is more common in finance applications and makes use of simulation and regression techniques (Least Squares Monte Carlo). This method uses the Bellman equation and the conditional expectations are approximated using a regression. It does not suffer from the curse of dimensionality and is therefore applicable to multi-dimensional problems. Least Squares Monte Carlo techniques are not a common solution method in macro-models, but we show that this method is very suitable to solve DSGE models. As a numerical example, we solve a 7-dimensional model in which the economy is subject to climate disasters and the representative agent can decide on abatement spending. Outcome variables of interest are the social cost of carbon, optimal abatement policy, risk-free rate and risk premium. The simulation and regression method outperforms the finite difference method in this 7-dimensional setting.

1 Introduction

This paper considers two solution methods to solve multi-dimensional continuous-time DSGE models. The setting that we study is a stochastic endowment economy with disaster risk, in which the representative agent has Epstein-Zin preferences. We also show how to calculate the equilibrium asset prices in this setting. Specifically, we derive the endogenous risk-free rate, risk-premium and consumption-wealth ratios of

*I am grateful for financial support from pension administrator and wealth manager MN.

the asset pricing model. We use this specific setting as an illustration for our solution methods, but the solution methods are applicable to a more general class of models.

The numerical example that we consider is a stochastic climate model in which the economy is subject to climate disasters. Outcome variables of interest are the optimal abatement policy, the expected trajectories of the climate variables over time and the social cost of carbon (monetized welfare loss of one ton of carbon emissions).

The first method that we consider is the finite difference method using the sparse grid combination technique. Finite difference methods have recently been a popular method to solve continuous-time DSGE models. The value function of the model must satisfy the Hamilton-Jacobi-Bellman (HJB) equation. This equation is a controlled partial differential equation. The finite difference method is a standard method to solve partial differential equations. One of the first applications in economics is Candler (1999). More recently, Achdou, Han, Lasry, Lions, and Moll (2017) use the finite difference method to solve a heterogeneous agent model and to back out the wealth distribution. Barnett, Brock, and Hansen (2020) solve a climate-economy model with ambiguity aversion using a finite difference scheme.

One extension to the standard finite difference method is the use of sparse grids. A good overview of different sparse grid solution methods is Pflüger (2010). Griebel (1998) proposes a finite difference algorithm with adaptive sparse grids. Ruttscheidt (2018) uses a similar algorithm to solve heterogeneous agent models. Brumm and Scheidegger (2017) apply adaptive sparse grid methods to discrete time problems using value function iteration. We do not consider adaptive sparse grids, but rather apply the sparse grid combination technique (Griebel, Schneider, & Zenger, 1990). The idea behind the combination technique is to solve the problem using several smaller regular grids, and combine the solutions of the smaller grids to obtain a single solution on a sparse grid. The main advantage of this technique compared to adaptive sparse grids is its simple implementation. Furthermore, the sub-problems can easily be solved in parallel which speeds up the computation time. The sparse grid combination method has to our knowledge not been applied to macro models yet. Another extension of this paper is that we show that the finite difference method is also applicable to the more general Epstein-Zin setting.

In addition to the finite difference method, we also consider the Least Squares Monte Carlo (LSMC) method. LSMC methods are based on a simulation and regression approach and were originally proposed to calculate the price of american options (Longstaff & Schwartz, 2001; Tsitsiklis & Van Roy, 2001). The idea is that in a dynamic programming setting, the conditional expectation can be approximated using regression. In the initial algorithms, the conditional expectation $E_t[V_{t+\delta_t}]$ is approximated by regressing the value function at time $t+\delta_t$ on basis functions of the state variables at time t, which we define as the regress now approach. The approach that we use is the regress later approach. The idea is to regress the value function at time $t+\delta_t$ on the basis functions of the state variables at time $t+\delta_t$, and then use the conditional expectations of the basis functions to obtain an approximation of the conditional expectation of the value function. This method is used by for example Glasserman and Yu (2004) and Jain and Oosterlee (2015) to price options.

Most methods rely on forward simulation and then solve the recursive program-

ming problem backwards. If we consider an optimal control problem, forward simulation is not possible, since the control is unknown. One way to solve this problem is to use a random control variable in the simulation, and then optimize over the control variable when solving the model backwards. This is for example applied in Andreasson and Shevchenko (2019). However, it is not necessary for the approximation of the conditional expectations to actually simulate the process of the state variables forward. Using a regress later approach, it is more efficient to specify at every time point a support region for the state variables and draw random state variables from this support region. This idea is also used in Balata and Palczewski (2017) and Ikefuji, Laeven, Magnus, and Muris (2020). The conditional expectations of the basis functions in the regress later approach are a function of the control variables. The optimal control variables are then the controls that maximize the conditional expectation of the value function in the next period. A key new insight of this paper is that in continuous time, one can directly use the first order conditions from the HJB-equation to find the optimal policy, which is faster than performing grid search or some other maximization algorithm. Furthermore, we extend the LSMC method to recursive Epstein-Zin preferences.

LSMC methods are widely used in option pricing, but are relatively new in macro-models. A literature where these methods are used more often including an optimal control setting are portfolio choice models (Brandt, Goyal, Santa-Clara, & Stroud, 2005; Koijen, Nijman, & Werker, 2010). One closely related example of an application in a macro model is Ikefuji et al. (2020), who look at a stochastic climate-economy model and solve the model using simulation and regression techniques.

2 The general problem

We first specify a general problem, then the solution methods are discussed. After that we look at the accuracy and performance of the solution methods for several numerical examples.

The general setting that we consider in this paper is the setting of a stochastic Lucas-tree economy with disasters in continuous time. There is a single representative agent that maximizes utility of consumption. The agent receives a stochastic exogenous endowment stream. We assume that the representative agent has Epstein-Zin

utility[1]:

$$V_t = \max_{u_t} E_t\Big[\int_t^{\infty} f(C_s, V_s)ds\Big]$$

where

$$f(C,V) = \frac{\beta}{1-1/\epsilon}\frac{C^{1-1/\epsilon} - \big((1-\gamma)V\big)^{\frac{1}{\zeta}}}{\big((1-\gamma)V\big)^{\frac{1}{\zeta}-1}} \quad \text{for } \epsilon \neq 1 \tag{1}$$

$$\text{with } \zeta = \frac{1-\gamma}{1-1/\epsilon}.$$

Define X_t as the d_X-dimensional vector with state variables: $X_t = [X_{1,t} \ ... \ X_{d_X,t}]'$. u_t is the d_u-dimensional vector of control variables: $u_t = [u_{1,t} \ ... \ u_{d_u,t}]'$. Endowment Y_t follows a geometric Brownian motion with additional jump processes. The drift, volatility and arrival rate distributions can depend on the state variables X_t and the controls u_t:

$$dY_t = \mu(X_t, u_t, t)Y_t dt + \sigma(X_t, u_t, t)Y_t dZ_t^Y + \sum_{m=1}^{M} J_{m,t}Y_{t^-}\, dN_{m,t}. \tag{2}$$

Z_t^Y is a standard Brownian motion. $N_{m,t}$ is a Poisson process with a state- and control-dependent arrival rate $\lambda_{m,t} = \lambda_m(X_t, u_t, t)$. The multiple Poisson processes are assumed to be independent. Y_{t^-} denotes aggregate endowment just before a jump ($Y_{t^-} = \lim_{h \# 0} Y_{t-h}$). When a jump of Poisson process $N_{m,t}$ arrives at time t, the jump size is controlled by the random variable $J_{m,t}$. $J_{m,t}$ can be seen as the percentage change of Y_t after a jump. We assume that the distributions of the jump sizes are time invariant, so we drop the time-index of J_m from now on. The dynamics of the vector of state variables is given by:

$$dX_t = \mu_X(X_t, u_t, t)dt + \sigma_X(X_t, u_t, t)dZ_t^X. \tag{3}$$

Here Z_t^X is an n-dimensional standard Brownian motion, $\mu_X(X_t, u_t, t)$ is a d_X-dimensional drift vector and $\sigma_X(X_t, u_t, t)$ is a $d_X \times d_X$ matrix of volatilities. We assume that σ_X is a diagonal matrix, i.e. we assume that the Brownian motions of the different state variables are not correlated.

A fraction of the endowment can be spent on costly control variables. The cost function can be time and state dependent. Therefore, $C_t = Y_t(1 - c(X_t, u_t, t))$. Define the consumption-endowment ratio by $\xi_t = \xi(X_t, u_t, t) = (1 - c(X_t, u_t, t))$. One can for example consider that it is possible to invest part of the endowment to increase the drift of the endowment or decrease the volatility of the endowment. Another example is investing part of the endowment to avert catostrophes. Martin and Pindyck (2015) consider a model with multiple catastrophes and look at optimal investment to reduce the disaster probabilities. Barro (2015) considers a model with climate disasters with

[1]Note that V_t is not directly the continuous time counterpart of the Epstein-Zin preferences, but a transformed ordinally equivalent utility process. This transformation is performed to make sure that the variance multiplier that belongs to f is 0. This transformation makes the utility function more tractable. For more details, see Duffie and Epstein (1992b).

a constant probability and environmental investment to decrease the probability of a climate disaster. The setting that we will consider as a numerical example is the setting of climate disasters where the intensity of climate disasters are a function of temperature. The control variable in this case is carbon emissions abatement.

We use this specific model to illustrate the solution methods. However, the solution methods are also applicable to several extensions. It is possible to also allow for jumps in the state variables. Furthermore, correlation between Brownian motions can be introduced. For the Least-Squares Monte Carlo method, this is straightforward to implement. For the finite difference method, this is slightly more involved. We will come back to this point later. Another possible extension is to step away from the Lucas-tree model and to consider a capital and/or labour based model with a production function and investment.

2.1 HJB-equation

To keep notation simple, we often drop the explicit dependence of variables on the state and control variables. For example, we use μ_t instead of $\mu(X_t, u_t, t)$. We use notation V_X, V_{XX} for the derivatives of the value function with respect to the vector of state-variables: $V_X = \begin{bmatrix} \frac{\partial V_t}{\partial X_{1,t}} & ... & \frac{\partial V_t}{\partial X_{d_X,t}} \end{bmatrix}$ and

$$V_{XX} = \begin{bmatrix} \frac{\partial^2 V_t}{(\partial X_{1,t})^2} & \cdots & \frac{\partial^2 V_t}{\partial X_{d_X,t} \partial X_{1,t}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V_t}{\partial X_{1,t} \partial X_{d_X,t}} & \cdots & \frac{\partial^2 V_t}{(\partial X_{d_X,t})^2} \end{bmatrix}. \tag{4}$$

We obtain the following HJB-equation that the value-function $V(Y_t, X_t, t)$ must satisfy:

$$\begin{aligned} 0 = \max_{u_t} \Big\{ & f(C_t, V_t) + V_Y \mu_t Y_t + \frac{1}{2} V_{YY} \sigma_t^2 Y_t^2 + \frac{\partial V}{\partial t} + V_X \mu_X \\ & + \frac{1}{2} tr(V_{XX} \sigma_X \sigma_X^0) + \sum_{m=1}^{M} \lambda_{m,t} E\Big[V\Big(Y_t \ \ (1 + J_m), X_t, t \Big) - V(Y_t \ \ , X_t, t) \Big] \Big\}. \end{aligned} \tag{5}$$

Now conjecture: $V(Y_t, X_t, t) = \frac{g(X_t,t) Y_t^{1-\gamma}}{1 \ \ \gamma}$. We can calculate the derivatives and we can substitute the conjecture of V_t into $f(C, V)$. Substituting our conjecture into $f(C, V)$ gives:

$$\begin{aligned} f(C_t, V_t) &= \frac{\beta}{1 - 1/\epsilon} \frac{\Big(Y_t \xi_t \Big)^{1 \ \ 1/\epsilon} - \Big(g_t Y_t^{1 \ \ \gamma} \Big)^{\frac{1}{\zeta}}}{\Big(g_t Y_t^{1 \ \ \gamma} \Big)^{\frac{1}{\zeta} \ \ 1}} \\ &= \frac{\beta}{1 - 1/\epsilon} \Big(g_t^{1 \ \ \frac{1}{\zeta}} \xi_t^{1 \ \ 1/\epsilon} Y_t^{1 \ \ \gamma} - g_t Y_t^{1 \ \ \gamma} \Big) \\ &= \beta \zeta \Big(g_t^{\ \frac{1}{\zeta}} \xi_t^{1 \ \ 1/\epsilon} - 1 \Big) V_t. \end{aligned} \tag{6}$$

The partial derivatives of V_t are given by:

$$\begin{aligned} V_Y &= g_t Y_t^{\ \gamma}, \quad V_{YY} = -\gamma g_t Y_t^{\ \gamma\ 1} \\ V_X &= \frac{g_X Y_t^{1\ \gamma}}{1-\gamma}, \quad V_{XX} = \frac{g_{XX} Y_t^{1\ \gamma}}{1-\gamma} \\ \frac{\partial V_t}{\partial t} &= \frac{\frac{\partial g_t}{\partial t} Y_t^{1\ \gamma}}{1-\gamma}. \end{aligned} \tag{7}$$

Here g_X denotes the row vector with partial derivatives to each of the state variables, similar to V_X and g_{XX} the matrix of second derivatives, similar to V_{XX}. Note that for $\gamma > 1$, V_t is negative. Substituting the derivatives and dividing by V_t gives the reduced HJB-equation:

$$\begin{aligned} 0 = \min_{u_t} \Big\{ \Big(\beta\zeta \Big(g_t^{\ 1/\zeta} \xi_t^{1\ \ 1/\epsilon} - 1 \Big) + (1-\gamma) \Big(\mu_t - \frac{1}{2}\gamma\sigma_t^2 + \sum_{m=1}^{M} \lambda_{m,t} E\Big[\frac{(1+J_m)^{1\ \gamma} - 1}{1-\gamma} \Big] \Big) \Big) g_t \\ + \frac{\partial g_t}{\partial t} + g_X \mu_X + \frac{1}{2} tr\Big(g_{XX} \sigma_X \sigma_X^{\emptyset} \Big) \Big\}. \end{aligned} \tag{8}$$

Note that since we assume that V_t is negative, dividing by V_t implies that the maximization problem becomes a minimization problem. Our goal is to find the function g_t that solves this partial differential equation. Given g_t, the optimal policy can be obtained using the first order condition(s). Denote the optimal policy by u_t. In special cases, it is possible to find a closed form expression for u_t as a function of g_t, g_X, g_{XX}, X_t, t. Otherwise, u_t is implicitly defined by the first order condition(s).

2.2 Asset prices

Once we know the function g_t, it is possible to derive the endogenous risk-free rate, risk premium and wealth-consumption ratio. Let π_t be the pricing kernel of the economy. First, define by B_t the risk free asset that pays continuous interest r_t. Furthermore, let S_t be the price of the asset that gives a claim on the consumption stream. It therefore pays continuous dividends equal to C_t. More formally: $S_t = E_t\Big[\int_t^{\mathbf{1}} \frac{\pi_s}{\pi_t} C_s ds \Big]$. The risk premium is defined by the difference between the return on S_t (including dividend payments) and the return on the risk-free asset. The consumption-wealth ratio k_t in the model is equal to $\frac{C_t}{S_t}$. The interest rate r_t, risk premium rp_t and consumption-wealth ratio k_t can be expressed as a function of the parameters, g_t and the consumption-endowment ratio ξ_t:

$$
\begin{aligned}
r_t = & \underbrace{\beta + \frac{\mu_t}{\epsilon} - \Big(1+\frac{1}{\epsilon}\Big)\frac{\gamma}{2}\sigma_t^2}_{\text{Standard interest rate}} \underbrace{- \frac{1}{2}\frac{1}{\zeta}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X \sigma_X^{\emptyset}\Big)}_{\text{Diffusion risk of the state variables}} \\
& + \underbrace{1/\epsilon\Big(\mu_\xi - \frac{1}{2}(1+1/\epsilon)tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\Big) - 1/\epsilon(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big)}_{\text{Drift and diffusion risk of the consumption-endowment ratio}} \\
& \underbrace{- \Big(\gamma - \frac{1}{\epsilon}\Big)\sum_{m=1}^{M}\lambda_{m,t}E\Big[\frac{(1+J_m)^{1\ -\gamma} - 1}{1-\gamma}\Big] - \sum_{m=1}^{M}\lambda_{m,t}E\Big[(1+J_m)^{\ -\gamma} - 1\Big]}_{\text{Jump risk}} \\
rp_t = & \underbrace{\gamma\sigma_t^2}_{\text{Standard diffusion risk}} \underbrace{+ \sum_{m=1}^{M}\lambda_{m,t}E\Big[J_m + (1+J_m)^{\ -\gamma} - (1+J_m)^{1\ -\gamma}\Big]}_{\text{Jump risk}} \\
& + \underbrace{\frac{1}{\zeta}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X\sigma_X^{\emptyset}\Big)}_{\text{Diffusion risk of the state variables.}} + \underbrace{\frac{1}{\epsilon^2}tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big) + \frac{1}{\epsilon}\Big(\frac{2}{\zeta} - 1\Big)tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big)}_{\text{Diffusion risk of the consumption-endowment ratio}} \\
k_t = & \beta g_t^{\ \frac{1}{\zeta}}\xi_t^{1\ \ 1/\epsilon}.
\end{aligned}
\tag{9}
$$

The derivations are given in appendix A.

3 Solution method 1: Finite Difference

The reduced HJB-equation is a partial differential equation. The finite difference method numerically solves this partial differential equation. We refer to Lapeyre, Sulem, and Talay (2005), Ch. 7 and 8 or Thomas (2013), for proofs and more extensive results on finite differences. First, we specify a bounded domain on which g_t is defined. Even though several state variables might have an unbounded domain, we still have to cut of the domain at some point. The cut off point should be far enough away of the area of interest. Therefore we first specify for the vector of state variables a vector with the minimum and maximum values: $\underline{X} = [\underline{X_1}\ \ ...\ \ \underline{X_{d_X}}]^{\emptyset}$ and $\overline{X} = [\overline{X_1}\ \ ...\ \ \overline{X_{d_X}}]^{\emptyset}$ respectively. We also cut off the time dimension at a point $\overline{T}$ far enough in the future.

We can write the reduced HJB equation as follows:

$$
\begin{aligned}
0 &= \min_{u_t} \Big\{ R(X_t, g_t, u_t, t) g_t + \frac{\partial g_t}{\partial t} + D_t g_t \Big\} \\
&\text{where} \quad D_t g_t = \sum_{d=1}^{d_X} (\mu_X)_d \frac{\partial g_t}{\partial X_{d,t}} + \frac{1}{2} \sum_{d=1}^{d_X} (\sigma_X \sigma_X^{\mathbb{0}})_{d,d} \frac{\partial^2 g_t}{(\partial X_{d,t})^2} \\
&\text{and} \quad R(X_t, g_t, u_t, t) = \beta\zeta \Big(g_t^{\ 1/\zeta} \xi_t^{1\ \ 1/\epsilon} - 1 \Big) \\
&+ (1-\gamma) \Big(\mu_t - \frac{1}{2}\gamma\sigma_t^2 + \sum_{m=1}^{M} \lambda_{m,t} E\Big[\frac{(1+J_m)^{1\ \ \gamma} - 1}{1-\gamma} \Big] \Big).
\end{aligned}
\tag{10}
$$

D_t is a difference operator that depends on X_t, u_t and t. Here $(\mu_X)_d$ denotes the d-th element of the vector μ_X. Similar notation is used for the volatility matrix. Denote by e_d a d_X dimensional vector with zeros except for the d-th element, which is a one. A natural way to approximate the first and second derivative with respect to $X_{d,t}$ is to use the central finite difference approximations. Let δ_d be the finite difference step in dimension d. The derivatives can then be approximated by:

$$
\begin{aligned}
\frac{\partial g_t}{\partial X_{d,t}} &\approx \frac{g(X_t + \delta_d e_d, t) - g(X_t - \delta_d e_d, t)}{2\delta_d} \equiv \partial_d g(X_t, t), \\
\frac{\partial^2 g_t}{(\partial X_{d,t})^2} &\approx \frac{g(X_t + \delta_d e_d, t) - 2g(X_t) + g(X_t - \delta_d e_d, t)}{\delta_d^2} \equiv \partial_{dd} g(X_t, t).
\end{aligned}
\tag{11}
$$

Finite difference schemes are unfortunately not always stable and might therefore not converge to the right solution. A textbook example is the one-dimensional advection equation $\frac{\partial g(x,t)}{\partial t} + a\frac{\partial g(x,t)}{\partial x} = 0$ where $a > 0$ with $g(x, 0)$ given. The finite difference approximation of $a\frac{\partial g(x,t)}{\partial x}$ with a central scheme with step size δ can then be written as:

$$
a\frac{\partial g(x,t)}{\partial x} \approx \frac{a}{2\delta} g(x+\delta, t) - \frac{a}{2\delta} g(x-\delta, t). \tag{12}
$$

This scheme turns out to be unstable when central differences are used. To obtain a stable scheme, the scheme must be monotone. An important condition for monotonicity in this case is that the coefficients on $g(x + \delta, t)$ and $g(x - \delta, t)$ must both be non-positive and the coefficients on $g(x, t)$ must be non-negative. This is the case when a backward-difference approximation is used. See Candler (1999) for more details on this example.

In order to obtain a stable scheme, one can use the so-called upwind method. The upwind scheme either uses a forward or backward first difference approximation, depending on whether the drift is positive or negative. Formally, we define the upwind

differential operator ∂_i^u as follows[2]:

$$\begin{aligned}
\partial_d^+ g(X_t,t) &\equiv \frac{g(X_t+\delta_d e_d,t)-g(X_t,t)}{\delta_d} \\
\partial_d\, g(X_t,t) &\equiv \frac{g(X_t,t)-g(X_t-\delta_d e_d,t)}{\delta_d} \\
\frac{\partial g_t}{\partial X_{d,t}} &\approx \partial_d^+ g(X_t,t)\mathbb{1}_{(\mu_X)_d\ \ 0} + \partial_d\, g(X_t,t)\mathbb{1}_{(\mu_X)_d<0} \equiv \partial_d^u g(X_t,t).
\end{aligned} \tag{13}$$

Note that for the second derivative approximation, we can just use the central difference scheme. The second central derivative scheme is monotone since the volatility is always non-negative: $(\sigma_X \sigma_X^\emptyset)_{d,d} \geq 0$. Due to the assumption that correlations between state variables is zero, we only have to consider $\frac{\partial^2 g_t}{(\partial X_{d,t})^2}$. If we would consider non-zero correlations, we would have to calculate $\frac{\partial^2 g_t}{\partial X_{i,t}\partial X_{j,t}}$ as well. The covariances can be negative and therefore it is not straightforward to have a monotone scheme using these cross correlations. A way to deal with this monotonicity is described in Ma and Forsyth (2017). We do not consider this extension.

Let us now define an operator $D_{i,j,t}$ on the grid points, that approximates the exact operator D_t. To keep the notation clear, we consider for now a 2-dimensional problem. The extension to a d_X dimensional problem is straightforward. Assume that the number of grid points per dimension equals N_d. This gives $\delta_d = \frac{\overline{X_d}\ \ \underline{X_d}}{N_d\ \ 1}$. Then we obtain the following grid points:

$$\begin{aligned}
x_i^1 &= \underline{X_1} + \delta_1(i-1), \quad i \in \{1,...,N_1\}, \\
x_j^2 &= \underline{X_2} + \delta_2(j-1), \quad j \in \{1,...,N_2\}.
\end{aligned} \tag{14}$$

Define for each grid point the following functions:

$$\begin{aligned}
&g_{i,j,t} \equiv g(x_i^1,x_j^2,t), \quad u_{i,j,t} = u\ (x_i^1,x_j^2,g_{i,j,t},t)\ \ \mu_{Xi,j,t} = \mu_X(x_i^1,x_j^2,u_{i,j,t},t), \\
&\mu^+_{Xi,j,t} = \max\Big(0,\mu_{Xi,j,t}\Big), \quad \mu_{Xi,j,t} = -\min\Big(0,\mu_{Xi,j,t}\Big), \quad \sigma_{Xi,j,t} = \sigma_X(x_i^1,x_j^2,u_{i,j,t},t).
\end{aligned} \tag{15}$$

Using an upwind approximation for the first derivative and a central approximation for the second derivative we obtain for the interior points of the grid the following

[2]Note that in the example of the advection equation, the backward difference approximation is stable when the drift a is positive. In our definition of the upwind scheme, we use a forward difference approximation when the drift is positive. The reason for this difference is that HJB-equations are solved backwards in time given a terminal condition, where the advection equation is solved forward in time given an initial condition.

operator $D_{i,j,t}$:

$$D_{i,j,t}g_{i,j,t} = \left(-\sum_{d=1}^{2} \frac{|(\mu_{Xi,j,t})_d|}{\delta_d} - \sum_{d=1}^{2} \frac{\left(\sigma_{Xi,j,t}\sigma^{0}_{Xi,j,t}\right)_{d,d}}{\delta_d^2} \right) g_{i,j,t}$$
$$+ \left(\frac{(\mu_{Xi,j,t})_1}{\delta_1} + \frac{1}{2}\frac{\left(\sigma_{Xi,j,t}\sigma^{0}_{Xi,j,t}\right)_{1,1}}{\delta_1^2} \right) g_{i\ \ 1,j,t} + \left(\frac{(\mu_{Xi,j,t})_2}{\delta_2} + \frac{1}{2}\frac{\left(\sigma_{Xi,j,t}\sigma^{0}_{Xi,j,t}\right)_{2,2}}{\delta_2^2} \right) g_{i,j\ \ 1,t}$$
$$+ \left(\frac{(\mu^{+}_{Xi,j,t})_1}{\delta_1} + \frac{1}{2}\frac{\left(\sigma_{Xi,j,t}\sigma^{0}_{Xi,j,t}\right)_{1,1}}{\delta_1^2} \right) g_{i+1,j,t} + \left(\frac{(\mu^{+}_{Xi,j,t})_2}{\delta_2} + \frac{1}{2}\frac{\left(\sigma_{Xi,j,t}\sigma^{0}_{Xi,j,t}\right)_{2,2}}{\delta_2^2} \right) g_{i,j+1,t}. \tag{16}$$

3.1 Boundary conditions

At the boundaries of the grid it is not possible to apply the central scheme for the second derivative since there is only one neighbour point available. Furthermore, the backward (forward) differences at the left (right) boundaries are also not possible anymore. We have to specify some kind of boundary condition such that we are able to deal with these issues. It is often the case that the state variables are somehow mean-reverting. When this is the case, the drift of the state variable will be positive at the left boundary and negative at the right boundary if the boundaries are chosen far enough to the left and right. The upwind scheme will then use the forward (backward) difference at the left (right) boundary and therefore there are no problems with the first difference at the boundaries. In this case, we can assume as a boundary condition that the second derivative vanishes at the boundary to handle the problem with the central difference scheme for the second derivative. Every boundary condition will introduce an error at the boundary, but this error will be small for points far enough away from the boundaries.

One way to implement the boundary conditions is to introduce so called 'ghost' points. Consider the point at the boundary $g_{1,j,t}$ where $j \in \{2, .., N_2 - 1\}$, i.e. x_j^2 is not a boundary point but x_1^1 is at the boundary. In this case we can define the 'ghost' point $g_{0,j,t}$. The second difference approximation in dimension 1 becomes:

$$\partial_{11} g_{1,j,t} = \frac{1}{2}\frac{\left(\sigma_{X1,j,t}\sigma^{0}_{X1,j,t}\right)_{1,1}}{\delta_1^2}\Big(g_{0,j,t} - 2g_{1,j,t} + g_{2,j,t}\Big). \tag{17}$$

Setting this equation equal to 0 implies that $g_{0,j,t} = 2g_{1,j,t} - g_{2,j,t}$. Given $g_{0,j,t}$, we can just apply (16). For different boundary points, we can construct 'ghost' points in a similar way.

When the state variables are not mean-reverting and the upwind scheme might use the backward (forward) difference at the left (right) boundary, the boundary conditions specified above might not be stable. One can instead assume that the first derivative vanishes at the boundary. Again, a 'ghost' point can be introduced. For

the point $g_{1,j,t}$ we will now obtain the 'ghost' point $g_{0,j,t} = g_{1,j,t}$. The reason that we do not always use the condition on the first derivative, is because the condition specified on the second derivative will lead to a smaller error.

3.2 Setting up the scheme

It is possible to perform all the operations directly for an entire vector of grid points. Define $R_{i,j,t} = R(x_i^1, x_j^2, g_{i,j,t}, u_{i,j,t}, t)$. We define the vector g_t^δ and the vector R_t^δ:

$$g_t^\delta \equiv \begin{bmatrix} g_{1,1,t} \\ g_{2,1,t} \\ \vdots \\ g_{N_1,1,t} \\ g_{1,2,t} \\ \vdots \\ g_{N_1,N_2,t} \end{bmatrix}, \quad R_t^\delta \equiv \begin{bmatrix} R_{1,1,t} \\ R_{2,1,t} \\ \vdots \\ R_{N_1,1,t} \\ R_{1,2,t} \\ \vdots \\ R_{N_1,N_2,t} \end{bmatrix}. \tag{18}$$

So these two vectors evaluate the functions $g(X_t, t)$ and $R(X_t, g_t, u_t, t)$ at all grid points of the state variables. The superscript δ on g_t^δ and R_t^δ indicates that these are vectors of the functions g_t and R_t evaluated at the different grid points. Using this notation we can distinguish the vectors from the original functions $g_t = g(X_t, t)$ and $R_t = R(X_t, g_t, u_t, t)$. g_t^δ and R_t^δ are column vectors of length $N_1 N_2$.

Note that equation (16) describes the finite difference operation $D_{i,j,t}$ for a single point $g_{i,j,t}$. It is now possible to construct a matrix D_t^δ such that $D_t^\delta g_t^\delta$ performs the finite difference operation directly for the entire vector g_t^δ. The definition of the $N_1N_2 \times N_1N_2$ matrix D_t^δ is given in appendix B. Note that D_t^δ is a very sparse matrix. Lastly, denote by $I_{N_1N_2}$ the $N_1N_2 \times N_1N_2$ identity matrix. Similar to our state space, we also discretize the time-space. Assume the time step equals δ_t, this gives the grid of time points: $[t_0 = 0, t_1 = \delta_t, ..., t_{N_t} = T]$. We will now propose two finite difference schemes: the explicit and the semi-implicit scheme. Let us start with the explicit scheme. A possible discrete approximation of the reduced HJB-equation (10) at time t_{i+1}, $i \in \{0, ..., N_t - 1\}$ is the following:

$$0 = \frac{g_{t_{i+1}}^\delta - g_{t_i}^\delta}{\delta_t} + \Big(D_{t_{i+1}}^\delta + diag(R_{t_{i+1}}^\delta)\Big) g_{t_{i+1}}^\delta. \tag{19}$$

Here $\frac{g_{t_{i+1}}^\delta \; g_{t_i}^\delta}{\delta_t}$ is a backward approximation of the time derivative for the vector $g_{t_{i+1}}^\delta$. $D_{t_{i+1}}^\delta g_{t_{i+1}}^\delta$ equals a vector that approximates the true difference operator $D_{t_{i+1}} g_{t_{i+1}}$. Lastly, $diag(R_{t_{i+1}}^\delta) g_{t_{i+1}}^\delta$ is the matrix-vector multiplication that calculates $R_t g_t$ at every grid point. Rewriting this equation gives the explicit scheme:

$$\begin{aligned} g_{t_i}^\delta &= A_{t_{i+1}}^E g_{t_{i+1}}^\delta, \\ A_{t_{i+1}}^E &\equiv I_{N_1N_2} + \delta_t \Big(D_{t_{i+1}}^\delta + diag(R_{t_{i+1}}^\delta)\Big). \end{aligned} \tag{20}$$

Similarly, we can define the semi-implicit scheme as:

$$\begin{aligned} 0 &= \frac{g^{\delta}_{t_{i+1}} - g^{\delta}_{t_i}}{\delta_t} + \Big(D^{\delta}_{t_{i+1}} + diag(R^{\delta}_{t_{i+1}})\Big) g^{\delta}_{t_i}, \\ &\implies g^{\delta}_{t_i} = A^{I}_{t_{i+1}} g^{\delta}_{t_{i+1}}, \\ A^{I}_{t_{i+1}} &\equiv \left(I_{N_1 N_2} - \delta_t \Big(D^{\delta}_{t_{i+1}} + diag(R^{\delta}_{t_{i+1}})\Big) \right)^{1}. \end{aligned} \tag{21}$$

The explicit scheme is the most straightforward scheme, but has the issue that the stability of the scheme depends on the time step. In some cases, a very small time step is required to obtain a stable solution. The semi-implicit scheme is generally more stable but either requires to invert a matrix or solve a linear system, where the explicit scheme merely requires a matrix vector multiplication. Note that the scheme is called semi-implicit since the vector $R^{\delta}_{t_{i+1}}$ still depends on $g^{\delta}_{t_{i+1}}$. A fully implicit scheme would solve the following equation for $g^{\delta}_{t_i}$:

$$0 = \frac{g^{\delta}_{t_{i+1}} - g^{\delta}_{t_i}}{\delta_t} + \Big(D^{\delta}_{t_i} + diag(R^{\delta}_{t_i})\Big) g^{\delta}_{t_i}. \tag{22}$$

A fully implicit scheme is often unconditionally stable. This implies that the stability conditions are met for an arbitrarily large time step. We will discuss stability later in more detail. Since R^{δ}_t depends on the unknown function g^{δ}_t in a non-linear way, a non-linear solver must be used to obtain g^{δ}_t. However, this is inefficient. This is the reason why we do not consider the fully implicit scheme. Let us now describe the full algorithm.

The algorithm

Step 1: Start with an initial guess for the g^{δ}_T .
Start one step before terminal time $t_{N_t\ 1}$. Backwards in time, for every time step t_i, $i = N_t - 1, ..., 1$ we perform the following steps.
Step 2: $g^{\delta}_{t_{i+1}}$ is obtained from the previous iteration. Calculate the optimal policy $u_{t_{i+1}}$ using either a closed form or implicit expression that follows from the first order conditions. This requires as input $g^{\delta}_{t_{i+1}}$ and its derivatives. The derivatives can be calculated using central finite differences.
Step 3: Use $u_{t_{i+1}}$ to calculate $\mu_{X t_{i+1}}$ and $\sigma_{X t_{i+1}}$.
Step 4: Construct $D^{\delta}_{t_{i+1}}$ (see appendix B).
Step 5: Use $g^{\delta}_{t_{i+1}}$ and $u_{t_{i+1}}$ to calculate $R^{\delta}_{t_{i+1}}$.
Step 6: Given $g^{\delta}_{t_{i+1}}$, $D^{\delta}_{t_{i+1}}$ and $R^{\delta}_{t_{i+1}}$, we can calculate A_t and either use the explicit scheme (20) or the semi-implicit scheme (21) to obtain $g^{\delta}_{t_i}$.
Optional: Step 7. To obtain the risk-free rate and the risk premium at period t_i, ξ_{t_i} and its derivatives can be calculated using (central) finite differences.
Repeat steps 2-7 until g^{δ}_0 is obtained. Given g^{δ}_0, calculate the optimal policy u_0 once more.

The matrix A_t (both in the explicit and implicit case) is a large sparse matrix. Several programming languages (e.g. Matlab and Python) have efficient procedures to set up the matrix A_t, by using the fact that only a few diagonal arrays are non-zero. In the case of the explicit scheme, the sparsity of the matrix speeds up the matrix-vector multiplication $A_t^E g_t^\delta$. In the case of the semi-implicit scheme, calculating the matrix A_t^I requires a matrix inversion. Actually inverting the matrix every time step is very inefficient. Instead, one can find $g_{t_i}^\delta$ by solving the following linear system.

$$\left(I_{N_1 N_2} - \delta_t \Big(L_{t_{i+1}}^\delta + diag(R_{t_{i+1}}^\delta)\Big)\right) g_{t_i}^\delta = g_{t_{i+1}}^\delta \tag{23}$$

There are several efficient routines to efficiently solve this linear system. We use the biconjugate gradient stabilized method, which is especially convenient to solve linear system with a large and sparse matrix.

If the optimal policy u_t has a closed form expression, one can easily calculate the optimal policy given g_t and its derivatives. If this is not the case, a non-linear equation has to be solved. Solving such an equation can be slow. To speed up the computation, it is important to start with a good initial guess. A good guess is the optimal policy in the previous period. So the guess for u_{t_i} is $u_{t_{i+1}}$. Furthermore, it pays off to supply the analytical Jacobian of the first order conditions to the solver. We use a trust-region algorithm with analytical Jacobian to calculate the optimal policy function.

As mentioned before, finite difference schemes can be unstable. Furthermore, oscillations might occur, which especially in the Epstein-Zin setting can cause problems since it may lead to complex numbers. Consistency, convergence and stability are discussed in more detail in appendix C. Until now we have assumed that the time derivative $\frac{\partial g_t}{\partial t} \neq 0$. In the case of an infinite horizon problem the time derivative in not equal to zero when there is explicit time dependence. Also in finite horizon problems the time derivative is non-zero. However, many economic models have an infinite time horizon and do not have explicit time dependence. In this case $\frac{\partial g_t}{\partial t} = 0$. In a discrete time framework, one can then solve for the value function by starting with an initial guess and iterating over the Bellman equation until the value function converged. We can use a similar algorithm for the finite difference approach. This approach is described in appendix D and turns out to be very similar to the finite difference scheme that we already proposed.

3.3 Sparse grids: Combination method

A problem with the finite difference approach is that it suffers from the curse of dimensionality. The size of the matrix A_t and of the vector g_t^δ grows exponentially with the number of dimensions. One solution to make sure the computational effort grows less fast with the number of dimensions is to use sparse grids. For a full grid with equal amount of points in each dimension, the total number of points on the d-dimensional grid equals N^d. So doubling the number of points in each dimension leads to 2^d times as much points. For a sparse grid with N points at the boundary

in each dimension, the total number of points is of the order $\mathcal{O}(N(\log N)^{d\ 1})$. As an example, doubling the number of points at the boundary of a sparse grid in 3 dimensions from 128 to 256 leads to 2.3 times as many points, instead of 8 times as many points for the full grid. Of course, less points on the grid also implies that the approximation will be less accurate. The accuracy of the solution depends on the smoothness of the function that is approximated, together with the task (e.g. interpolation, finite difference). Overall it turns out that sparse grids do not give up too much accuracy, while they are not subject to the curse of dimensionality.

A possible extension of regular sparse grids are spatially adaptive sparse grids. Based on some condition, an algorithm puts more grid points at the places where the error of the approximation is the largest. This is useful if the function that is approximated is very non-linear in some area's. However, there are computational costs related to the adaptivity. Griebel (1998) proposes an algorithm to apply finite differences on adaptive sparse grids. For more details on (adaptive) sparse grids, see for example Pflüger (2010) or Zumbusch (2000).

We choose to apply the so called combination method (Griebel et al., 1990). Instead of directly solving the problem on a sparse grid, one can combine solution on smaller full grids to generate a sparse grid solution. For linear interpolation tasks, it turns out that using the combination method or a direct sparse grid method give equivalent results. For non-linear tasks like solving a PDE combining full grids might not lead to exactly the same answer as the corresponding sparse grid. Nevertheless, the combination method has been widely applied. The first advantage is that one still solves full-grid problems. Finite difference operations on full grid problems are more straightforward to implement and can also be coded more efficient. Second, solving several sub-problems and combining them later is ideal for parallelization. Compared to direct sparse grid methods, the number of evaluation points will be slightly larger but the efficiency due to the well-behaved full grids and the possibility to parallelize outweigh this disadvantage. Spatial adaptivity is not possible with the combination method.

Let us again start with a 2-dimensional problem. Remember that N_d denotes the number of points and in dimension d. $\overline{X_d}$ and $\underline{X_d}$ are respectively the maximum and minimum values in both dimensions and $\delta_d = \frac{\overline{X_d}\ \underline{X_d}}{N_d\ 1}$ is the step size. Define the so called level per dimension by l_d. At the first level in dimension d ($l_d = 1$), we assume that the number of points in that dimension equals 3. When we increase the level by 1, we assume that the step size in that dimension is divided by two. This leads to the following relation between the number of points and the level: $N_d = 2^{l_d} + 1$. In two dimensions, we can therefore define a grid by the level $l = [l_1\ \ l_2]^{\emptyset}$. This is graphically illustrated in figure 1.

The combination method combines the finite difference solutions on several of the sub-grids. Define by $g_0^{\delta,l}$ the finite difference solution at $t = 0$ on a full grid with level l. Combining grids is not straightforward since all grids have different grid points. Assume we are interested in the value of g at a point $x = [x^1\ \ x^2]$. The full grid with level l gives as output the value of g at the grid points $x_i^1 = \underline{X_1} + \delta_1(i-1),\ \ i \in \{1, ..., 2^{l_1}+1\}$ and $x_i^2 = \underline{X_2} + \delta_2(i-1),\ \ i \in \{1, ..., 2^{l_2}+1\}$. To obtain the value of g at

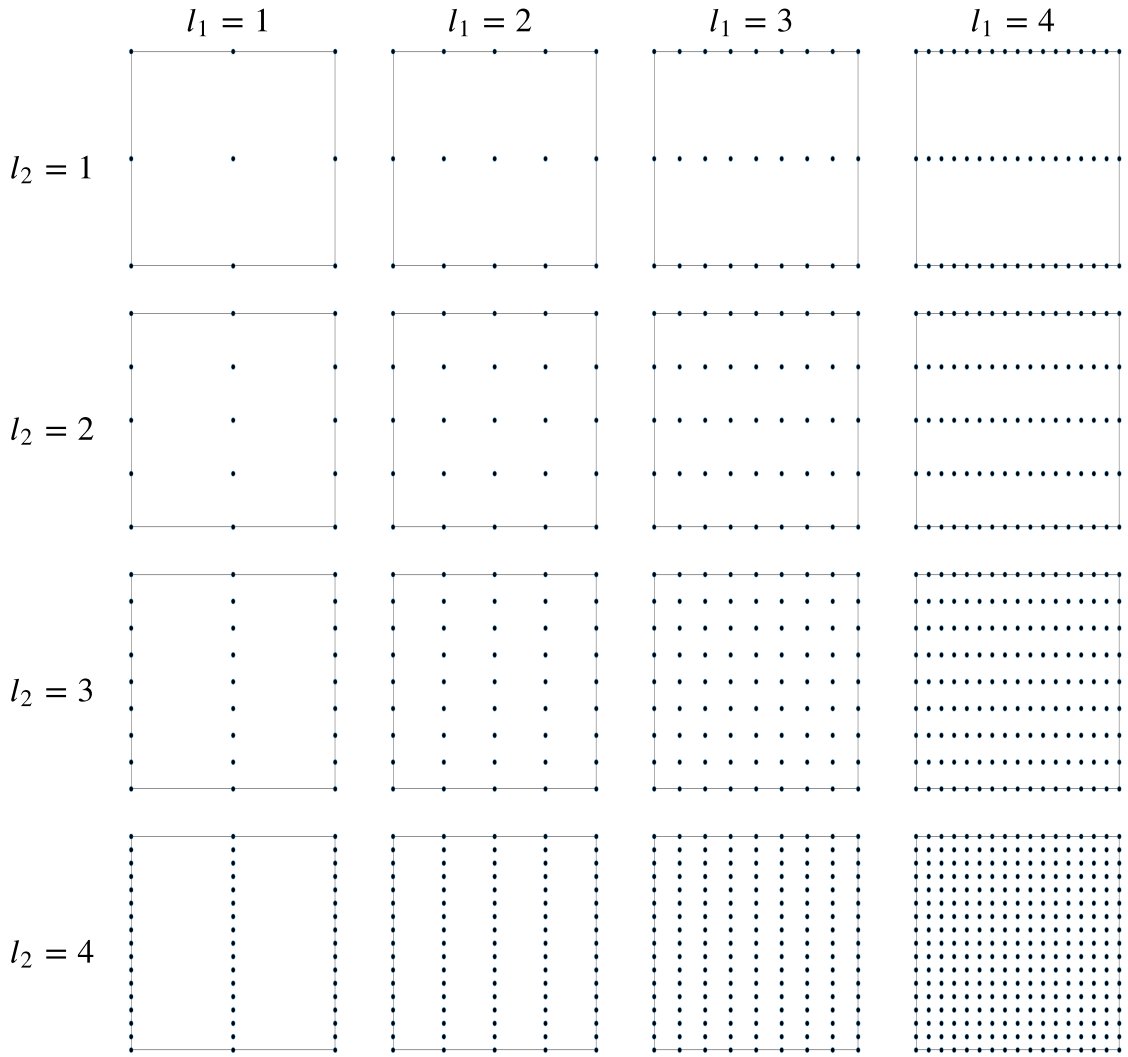

Figure 1: Full grids for several level combinations in two dimensions.

the point x, we use linear interpolation. Define the approximation of g at the point x using a level l full grid by $g^{\delta,l}(x,0)$. Lastly, denote by L the level of the sparse grid. Then the sparse grid combination solution at point x becomes:

$$g^{\delta}_{SG}(x,0) = \sum_{l_1+l_2=L+1} g^{\delta,l}(x,0) - \sum_{l_1+l_2=L} g^{\delta,l}(x,0). \tag{24}$$

Figure 2 shows graphically how the full sub-grids are combined. For a d-dimensional problem, the intuition of the combination method is similar to the 2-dimensional problem, but the formulas are slightly different. Let d_X be the number of state variables. Then the sparse grid solution becomes:

$$g^{\delta,L}_{SG}(x,0) = \sum_{k=0}^{d_X \quad 1} (-1)^k \binom{d_X - 1}{k} \sum_{l_1+\ldots+l_{d_X}=L+(d_X \quad 1) \quad k} g^{\delta,l}(x,0). \tag{25}$$

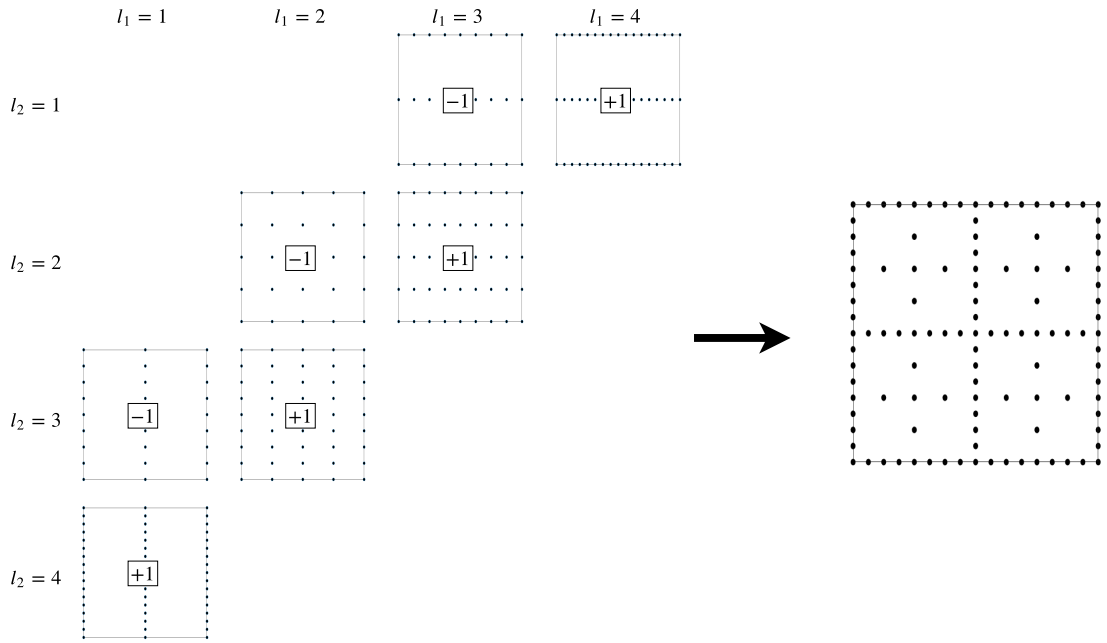

Figure 2: The combination method to obtain a sparse grid with level $L = 4$ in two dimensions.

4 Solution method 2: Least Squares Monte Carlo

The Least Squares Monte Carlo (LSMC) methods are originally proposed to calculate the value of American options. The methods use dynamic programming which makes it suitable for the type of problem that we consider. Traditional value function iteration can become extremely slow for high-dimensional problems. The LSMC methods use a simulation and regression approach to estimate the conditional expectations and solve the dynamic programming method. They do not suffer from the curse of dimensionality.

Since we look at recursive preferences, we can not use the standard dynamic programming equation. We use the discrete time Epstein-Zin recursion. Assume that the time step equals δ_t. The standard discrete time Epstein-Zin recursion equals:

$$U_t = \max_{u_t} \left((1 - e^{\ \beta\delta_t}) C_t^{1\ \ 1/\epsilon} + e^{\ \beta\delta_t} E_t[U_{t+\delta_t}^{1\ \ \gamma}]^{1/\zeta} \right)^{\frac{1}{1-1/\epsilon}}. \tag{26}$$

Instead of using the standard discrete time recursion of the Epstein-Zin preferences, we again use an ordinally equivalent version to be in line with the stochastic differential utility setting. We define $V_t = \frac{U_t^{1-\gamma}}{1\ \ \gamma}$. We have derived that $V_t = \frac{g(X_t,t)Y_t^{1-\gamma}}{1\ \ \gamma}$.

We again assume that $\gamma > 1$. After substitution of U_t and rearranging we obtain the following recursion:

$$g(X_t,t)Y_t^{1\ \ \gamma} = \min_{u_t}\left((1-e^{\ \beta\delta_t})C_t^{1\ \ 1/\epsilon} + e^{\ \beta\delta_t}\Big(E_t\big[g(X_{t+\delta_t},t+\delta_t)Y_{t+\delta_t}^{1\ \ \gamma}\big]\Big)^{1/\zeta}\right)^{\zeta}. \quad (27)$$

Note that the maximization has become a minimization problem, since $\gamma > 1$. Dividing both sides by $Y_t^{1\ \ \gamma}$ and rearranging gives:

$$g(X_t,t) = \min_{u_t}\left((1-e^{\ \beta\delta_t})\xi_t^{1\ \ 1/\epsilon} + e^{\ \beta\delta_t}\Big(E_t\Big[g(X_{t+\delta_t},t+\delta_t)\frac{Y_{t+\delta_t}^{1\ \ \gamma}}{Y_t^{1\ \ \gamma}}\Big]\Big)^{1/\zeta}\right)^{\zeta}. \quad (28)$$

For small time steps δ_t, it is reasonable to assume that $E_t\Big[g(X_{t+\delta_t},t+\delta_t)\frac{Y_{t+\delta_t}^{1-\gamma}}{Y_t^{1-\gamma}}\Big] \approx E_t[g(X_{t+\delta_t},t+\delta_t)]E_t\Big[\frac{Y_{t+\delta_t}^{1-\gamma}}{Y_t^{1-\gamma}}\Big]$. If we consider the Ito formula for Y_t and $g(X_t)$, the quadratic covariation between $g(X_t)$ and Y_t is equal to 0. The idea behind this is that there is no correlation between the shocks that drive Y_t and X_t. The correlation occurs through the drift μ, standard deviation σ and arrival rate λ. However, it takes some time for a shock in X_t to have an effect on Y_t through μ, σ and λ.

Applying Ito calculus, we can derive a closed form expression for $\frac{Y_{t+\delta_t}^{1-\gamma}}{Y_t^{1-\gamma}}$:

$$\begin{aligned}\frac{Y_{t+\delta_t}^{1\ \ \gamma}}{Y_t^{1\ \ \gamma}} = \exp\Big((1-\gamma)\int_t^{t+\delta_t}(\mu(X_s,u_s,s) - \sigma(X_s,u_s,s)^2/2)ds \\ + (1-\gamma)\int_t^{t+\delta_t}\sigma(X_s,u_s,s)dZ_s^c + (1-\gamma)\sum_{m=1}^{M}\sum_{i=1}^{N_{m,t+\delta_t}-N_{m,t}}\log(1+J_{m,i})\Big).\end{aligned} \quad (29)$$

If $N_{m,t+\delta_t} - N_{m,t} \geq 2$, i.e. when there are more than two jump processes of type m over the period $[t, t+\delta_t]$, then $J_{m,i},\ \ i = 1, ..., N_{m,t+\delta_t} - N_{m,t}$ are independent realizations of J_m. Then depending on the processes $\mu(X_t,u_t,t), \sigma(X_t,u_t,t)$ and $\lambda(X_t,u_t,t)$ we are either able to calculate the conditional expectation for the ratio in closed form or we can approximate it. Let us now describe the full algorithm.

The algorithm

Initialization: First we determine some algorithm parameters. We must cut-off the infinite horizon problem at some time T in the far future. We must make sure that T is far enough away, such that choosing a larger T does not change the solution anymore. We then discretize the problem in the time dimension. The second choice is therefore the time step δ_t. Define the grid of time points as: $[t_0 = 0,\ t_1 = \delta_t,\ ...\ , t_N = T]$. Next we define a plausible range for the state variables X_t at every time point t. Since we do not know the optimal policy u_t yet, we do not know exactly what the distribution of the state variables is. However, it is often still possible to obtain some (wide) range for the state variables. Denote the vector with upper bounds of X_{t_i} at period t_i by $\overline{X_{t_i}}$ and the vector of lower bounds by $\underline{X_{t_i}}$.

Step 1: Simulate K trajectories of the state variables at time T. The easiest way to do this is to simulate draws from a uniform distribution with boundaries $\underline{X_T}$ and $\overline{X_T}$. This gives the trajectories $X_T(k)$, $k = 1, ..., K$. If it is very clear that it is more plausible for X_T to be in the middle of the interval, one could for example also use a beta distribution or some other distribution to obtain trajectories for the state variables.

Step 2: Guess function $g(X_T(k), T)$ at terminal time T. A good approximation at terminal time implies that the solution will converge using a shorter time horizon.

Then start at one step before terminal time $t_{N\ 1}$. Backwards in time, for every time step t_i, $i = N - 1, ..., 1$ we perform the following steps.

Step 3: Approximating g as a function of the state variables. To approximate $g\big(X_{t_{i+1}}, t_{i+1}\big)$, choose a vector of N_b basis functions:

$$B\big(X_t\big) = [B_1\big(X_t\big) \quad ... \quad B_{N_b}\big(X_t\big)]. \tag{30}$$

Then using the sample of K trajectories, regress $g\big(X_{t_{i+1}}(k), t_{i+1}\big)$ on $B\big(X_{t_{i+1}}(k)\big)$ to obtain an N_b dimensional column vector of coefficients $\alpha_{t_{i+1}}$.

Step 4: Simulation. Simulate K trajectories of the state variables at time t_i, using a uniform distribution with boundaries $\underline{X_{t_i}}$ and $\overline{X_{t_i}}$.

Step 5: Calculating the optimal policy. To calculate u_{t_i}, we use the first order condition, which needs as input the derivatives of g. To calculate these derivatives, we use the following approximation of $g(X_{t_i}, t_i)$:

$$\hat{g}(X_{t_i}(k), t_i) = \alpha^{\emptyset}_{t_{i+1}} B(X_{t_i}(k)) = \sum_{j=1}^{N_b} (\alpha_{t_{i+1}})_j B_j(X_{t_i}(k)). \tag{31}$$

The vector of first derivatives with respect to the state variables can then be calculated as[3]:

$$\hat{g}_X(k) = \frac{\partial \hat{g}(X_t, t_i)}{\partial X_t}\Big|_{X_{t_i}(k)} = \sum_{j=1}^{n_b} (\alpha_{t_{i+1}})_j \frac{\partial B_j(X_t)}{\partial X_t}\Big|_{X_{t_i}(k)}. \tag{32}$$

Similarly we can find the second derivative $\hat{g}_{XX}$. Using these derivatives and the first order condition, we can calculate u_{t_i}. If u_{t_i} does not have a closed form expression, a non-linear equation has to be solved. It pays off to supply the analytical Jacobian of the first order conditions to the solver. We use a trust-region algorithm with analytical Jacobian to calculate the optimal policy function.

[3]Similar to Jain and Oosterlee (2015) we assume that $\frac{\partial \alpha_{t_i}}{\partial X_{t_i}} = 0$. This makes it much simpler to calculate the derivatives and this derivative is generally very close to zero. In a follow up paper, Jain, Leitao, and Oosterlee (2019) relax this assumption and they show how to calculate the derivatives recursively. We do not consider this extension.

Step 6: Approximating the conditional expectation. We want to approximate for each (d_X-dimensional) grid point $X_{t_i}(k)$ the conditional expectation $E_{t_i}\Big[g(X_{t_{i+1}}, t_{i+1})|X_{t_i}(k)\Big]$, $k = 1, ..., K$. We use the following relation:

$$E_{t_i}\Big[\hat{g}\big(X_{t_{i+1}}, t_{i+1}\big)|X_{t_i}(k)\Big] = \alpha^{0}_{t_{i+1}} E_{t_i}\Big[B\big(X_{t_{i+1}}\big)|X_{t_i}(k)\Big]. \tag{33}$$

Here $E_{t_i}\Big[B\big(X_{t_{i+1}}\big)|X_{t_i}(k)\Big]$ is known in closed form or has an analytic approximation. Note that this conditional approximation also depends on u_{t_i}.

Step 7: In this step we use the recursion to calculate the function g at time t_i:

$$g(X_{t_i}(k), t_i) = \left((1-e^{\ \beta\delta_t})\xi_t^{1\ 1/\epsilon} + e^{\ \beta\delta_t}\left(E_{t_i}\Big[\hat{g}(X_{t_{i+1}}, t_{i+1})|X_{t_i}(k)\Big] E_{t_i}\Big[\frac{Y^{1\ \gamma}_{t_{i+1}}}{Y^{1\ \gamma}_{t_i}}|X_{t_i}(k)\Big]\right)^{1/\zeta}\right)^{\zeta}. \tag{34}$$

Optional: Step 8. To calculate the risk-free rate and the risk premium, we have to calculate the derivatives of ξ_{t_i}. If u_{t_i} has a closed form expression in terms of g_{t_i} and its derivatives, it might be possible to express the derivatives of ξ_{t_i} as a function of the derivatives of g_{t_i}. If this is not possible, we can calculate the derivatives numerically. Note that we can calculate g_{t_i} and its derivatives at any state-space point X_{t_i} and time point t_i using $\hat{g}(x, t_i) = \alpha^{0}_{t_i} B(x)$. Therefore we can also calculate $\xi_{t_i} = \xi(X_{t_i}, u_{t_i}, t_i)$ at any state-space and time point t_i. An easy way to calculate the derivatives of ξ_{t_i} numerically is to use a finite difference approximation.

After iterating until $i = 1$, we obtain an estimate of g at time t_0. Calculate u_0 once more to obtain the optimal policy at time 0.

5 Results

In this section we present several numerical results. All computations are performed on a laptop with 7-th generation intel i7 processor, 4 cores and 16GB RAM. The first example that we consider is the asset pricing model of Wachter (2013).

5.1 Time-varying jump risk

Wachter (2013) considers a jump risk model where the arrival rate is time-varying. The model can be solved analytically when it is assumed that $\epsilon = 1$, but in this example we deviate from this assumption. Furthermore, we use a different jump size distribution for tractability. Except from that, the model is identical to the model developed in Wachter (2013). The growth rate μ and volatility σ of the endowment process are constant. There is a single Poisson process with arrival rate λ_t, which is

Parameter	Value
γ	4
ϵ	1.5
β	0.02
μ	0.025
σ	0.03
$\overline{\lambda}$	0.035
κ	0.08
σ_λ	0.07
α_J	6.5

Table 1: Parameters

time-varying. The jump size $1+J$ follows a power distribution with parameter α_J and pdf $f(x) = \alpha_J x^{\alpha_J \ 1}, \ \ 0 < x < 1$. This gives: $E[J] = \frac{1}{\alpha_J+1}$ and $E\Big[(1+J)^n\Big] = \frac{\alpha_J}{\alpha_J+n}$.

The model has a single state variable, namely the arrival rate λ_t. λ_t follows a Cox-Ingersoll-Ross (CIR) process.

$$\lambda_t = \kappa_\lambda(\overline{\lambda} - \lambda_t)dt + \sigma_\lambda\sqrt{\lambda_t}dZ_t^\lambda \tag{35}$$

The calibration is given in table 1. This problem does not have any control variables and $\xi_t = 1$ which implies that endowment and consumption are equal in equilibrium: $Y_t = C_t$.

5.1.1 Finite Difference Method

This problem is an infinite horizon problem without explicit time dependency, which implies that the time derivative is equal to zero. Furthermore, the problem is one-dimensional which implies we do not have to use the sparse grid method. We use an equally spaced one-dimensional grid to discretize the problem. The only state variable is the arrival rate: $X_t = \lambda_t$. The following parameters are used: $[\underline{X}, \overline{X}] = [0, 1]$, $\Delta = 50$, $crit = 10^{\ -6}$ and we vary the number of points. The right boundary is chosen such that increasing the boundary does not change the results anymore. As initial guess, we choose g_T^δ such that g_T^δ solves $R(X_t, g_t, t) = 0$ at the grid points. This initial guess is chosen since it makes sure that the finite difference scheme satisfies the stability conditions.

Table 2 illustrates the results for $\lambda_t = \overline{\lambda}$. Since the problem is only one-dimensional, the finite-difference scheme is very fast. When λ_t is at its average value, the model generates a risk-free rate of 0.9% and a risk premium of 2.8%. The consumption-wealth ratio is around 1.7%. The risk-free rate is close to the historical observed real risk-free rate. The risk-premium is bit lower than the historical equity premium, which implies that this model does not fully solve the equity premium puzzle (Mehra & Prescott, 1985). Dimson, Marsh, and Staunton (2011) estimate the world-wide historical equity premium to be around 4.5%, which shows that the model implied equity premium is on the low side. One way to obtain a more realistic equity premium

# points	$k = \frac{C}{S}$	g	$\frac{\partial g}{\partial \lambda_t}$	Interest rate	Risk premium	time (s)
201	1.724%	0.263	3.410	0.922%	2.849%	0.03
2001	1.708%	0.242	3.053	0.931%	2.831%	0.04
20001	1.707%	0.240	3.021	0.932%	2.830%	0.13
200001	1.706%	0.240	3.018	0.932%	2.829%	1.49

Table 2: Wealth-Consumption ratio, error of the HJB-equation and computation time for different number of points N ($\lambda_t = \overline{\lambda}$).

is to introduce leverage, as Wachter (2013) does. However, since it is not our main purpose to solve this puzzle, we will not consider this extension.

5.1.2 Least Squares Monte Carlo Method

We compare the LSMC with the finite difference method. λ_t follows a CIR-process. Shao (2012) shows that this implies that the increments of the arrival rate follow a non-central χ^2 distribution:

$$\begin{aligned} &\lambda_{t_{i+1}}|\lambda_{t_i}(k) \sim a_2\chi^2_{a_1}(a_3\lambda_{t_i}(k)) \quad \text{where} \\ &a_1 = \frac{4\kappa_\lambda\overline{\lambda}}{\sigma_\lambda^2}, \quad a_2 = \frac{\sigma_\lambda^2(1-e^{-\kappa_\lambda\delta_t})}{4\kappa} \quad \text{and} \quad a_3 = \frac{4\kappa_\lambda e^{-\kappa_\lambda dt}}{\sigma_\lambda^2(1-e^{-\kappa_\lambda\delta_t})}. \end{aligned} \tag{36}$$

As basis function we use standard polynomials: $B_j\big(\lambda_t\big) = \lambda_t^j$. Therefore, $E_{t_i}[B(\lambda_{t_{i+1}})|\lambda_{t_i}(k)]$ is just a vector of moments of the non-central χ^2-distribution, which are known in closed form. Also $E_{t_i}\Big[\frac{Y_{t_{i+1}}^{1-\gamma}}{Y_{t_i}^{1-\gamma}}|\lambda_{t_i}(k)\Big]$ can be calculated in closed form

$$\begin{aligned} &E_{t_i}\Big[\frac{Y_{t_{i+1}}^{1-\gamma}}{Y_{t_i}^{1-\gamma}}|\lambda_{t_i}(k)\Big] \\ &= E_{t_i}\Big[\exp\Big((1-\gamma)(\mu-\sigma^2/2)\delta_t + (1-\gamma)\sigma(Z^c_{t_{i+1}} - Z^c_{t_i}) + (1-\gamma)\sum_{j=1}^{N_{t_{i+1}}-N_{t_i}}\log(1+J_j)\Big)\Big] \\ &= \exp\Big((1-\gamma)(\mu-\sigma^2/2)\delta_t + \frac{1}{2}(1-\gamma)^2\sigma^2\delta_t + E_{t_i}\Big[\int_{t_i}^{t_{i+1}}\lambda_s ds|\lambda_{t_i}(k)\Big]E[(1+J)^{1-\gamma}-1]\Big) \end{aligned} \tag{37}$$

We approximate $E_{t_i}\Big[\int_{t_i}^{t_{i+1}}\lambda_s ds|\lambda_{t_i}(k)\Big]$ by $\lambda_{t_i}(k)\delta_t$, which is a good approximation for small time steps.

We have to choose several algorithmic parameters. First, we fix T at 500 years. For larger T the outcomes do not change anymore. When there is no jump risk, the problem can be solved in closed form. We use the g that belongs to the problem without jump risk as guess for g at terminal time for all trajectories. The number of trajectories equals: $K = 1000$. Increasing K also doens't change the results. Since there is no explicit time dependence, we choose $\overline{X_{t_i}} = 0.1$ and $\underline{X_{t_i}} = 0$ for any t_i.

Table 3 shows the results for the LSMC method. In the first step, we start with a fast run. After that run we change several algorithmic parameters to see how that

N_b	δ_t	$k = \frac{C}{S}$	g	$\frac{\partial g}{\partial \lambda_t}$	r	rp	time (s)
3	1	1.657%	0.184	2.066	0.966%	2.762%	0.2
6	1	1.698%	0.229	2.872	0.934%	2.826%	0.6
9	1	1.704%	0.236	3.004	0.929%	2.836%	1.0
9	0.1	1.705%	0.238	3.001	0.932%	2.829%	8.6
9	0.01	1.706%	0.239	3.001	0.933%	2.828%	88.9

Table 3: Wealth-Consumption ratio, error of the HJB-equation and computation time for different algorithmic parameters.

influences the estimate. The first run uses 3 basis functions (up to quadratic) and has a time step δ_t of 1 year. The algorithm is fast but is also not very accurate. Increasing the number of basis function to 9 improves the estimate a lot. To verify how important the discretization error is, we then decrease the time step to δ_t to 0.1. This is a slight improvement. This run takes about 9 seconds. In the last step we again decrease the time step ($\delta_t = 0.01$). The run now takes around 90 seconds. There is only modest improvement now. If we compare the results of the last run to the results of the finite difference method, the two are very close.

The computation time depends a lot on the discount rate β. In this example we use a discount rate $\beta = 2\%$. However, if we would choose e.g. $\beta = 5\%$, the problem will converge in much shorter time period and we could choose T smaller.

5.1.3 Graphical results

We have now presented the results at one specific point, namely $\lambda_t = \overline{\lambda}$. In this section we present graphically the results for an entire grid of values for λ_t. The results are shown in figure 3. The consumption-wealth ratio is not very responsive to changes in the arrival rate. However, the risk-free rate and the equity premium do react a lot. The risk-free rate even becomes negative if the arrival rate is large enough. The risk premium also increases a lot for large values of λ_t.

Concluding, the LSMC provides accurate solutions in a reasonable time for this example, but in a one-dimensional setting the finite method is much faster. It is clear that this model is very non-linear, since relatively many points are needed for the finite difference method to converge. Also for the LSMC method, three basis functions are not enough to capture the curvature of the function g. This example illustrates the capability of these methods to solve non-linear problems.

5.2 Multidimensional climate problem

The second problem we consider is a multidimensional climate problem in the spirit of Olijslagers and van Wijnbergen (2019). However, instead of having a deterministic climate model we assume that the climate model is stochastic. We take our climate model from Aengenheyster, Feng, Van Der Ploeg, and Dijkstra (2018), who develop a stochastic climate model. In their paper the stochastic climate model is used to

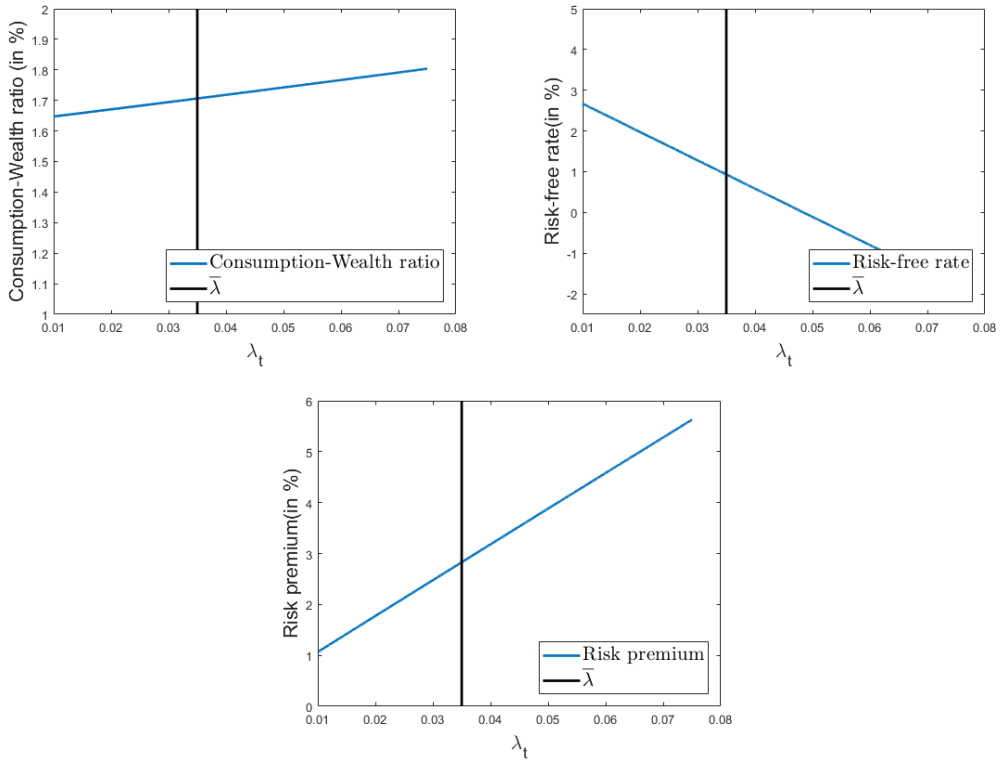

Figure 3: The consumption-wealth ratio, risk-free rate and the risk premium as a function of the state variable λ_t.

answer questions like when and how strict should carbon emissions be reduced if the 2 degrees target should be met with 67% probability in 2100. We integrate the stochastic model in a consumption based asset pricing model with an exogenous emissions path to calculate the social cost of carbon.

The model setup is as follows. $\mu_t = \mu$ and $\sigma_t = \sigma$ are constant. There are two Poisson processes: $N_{1,t}$ and $N_{2,t}$. $N_{1,t}$ reflects economic disasters as in the example of the previous subsection. In contrast to the previous model, we now assume that the arrival rate $\lambda_{1,t} = \lambda_1$ is constant. J_1 has again a power distribution with parameter α_{J_1}. The second type of disasters are climate disasters. The arrival rate of disasters is assumed to increase with the temperature: $\lambda_{2,t} = \lambda_2 T_t$. J_2 also follows a power distribution with parameter α_{J_2}.

Carbon emissions E_t (in GtC) are an exogenous function of time and calibrated to match the business as usual scenario of the DICE model (Nordhaus, 2017): $E_t = 1450.84 \exp(-4.933e^{\ 0.0075t} - 0.02t)$.

The vector of state variables equals $X_t = [M_{P,t} \ \ M_{1,t} \ \ M_{2,t} \ \ M_{3,t} \ \ T_{0,t} \ \ T_{1,t} \ \ T_{2,t}]^{\emptyset}$. $M_t = M_{P,t} + M_{1,t} + M_{2,t} + M_{3,t}$ is the atmospheric carbon concentration and $T_t = T_{0,t} + T_{1,t} + T_{2,t}$ is the temperature anomaly. The idea is to model artificial carbon and temperature boxes in order to mimic the dynamics of the complex climate system. Atmospheric carbon concentration is measured in GtC and temperature in ^oC. Define radiative forcing at time t as $F_t = A \log\left(\frac{M_t}{M_{pre}}\right)$. The drift and volatility vectors are

Par.	Value	Par.	Value	Par.	Value	Par.	Value
γ	4	α_{J_2}	35	τ_3	4.304	τ_{b_1}	1.427062
ϵ	1.5	C_0	80×10^{12}	σ_{M^2}	0.65	τ_{b_2}	8.021185
β	0.02	a_0	0.2173	M_{pre}	592.25	σ_{T^0}	0.015
μ	0.025	a_1	0.2240	A	7.92	σ_{T^2}	0.13
σ	0.03	a_2	0.2824	b_0	0.001152		
λ_1	0.035	a_3	0.2763	b_1	0.109680		
α_{J_1}	6.5	τ_1	394.4	b_2	0.033611		
λ_2	0.015	τ_2	36.54	τ_{b_0}	400		

Table 4: Calibration

defined as:

$$\mu_X = \begin{bmatrix} a_0 E_t \\ a_1 E_t - \frac{1}{\tau_1} M_{1,t} \\ a_2 E_t - \frac{1}{\tau_2} M_{2,t} \\ a_3 E_t - \frac{1}{\tau_3} M_{3,t} \\ b_0 F_t - \frac{1}{\tau_{b_0}} T_{0,t} \\ b_1 F_t - \frac{1}{\tau_{b_1}} T_{1,t} \\ b_2 F_t - \frac{1}{\tau_{b_2}} T_{2,t} \end{bmatrix}, \quad \sigma_X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{M_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{T_0} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{T_2} T_t \end{bmatrix}. \tag{38}$$

Note that this implies that there is one permanent carbon box $M_{P,t}$ that does never decay. The coefficients a_0, a_1, a_2 and a_3 sum up to 1, such that a percentage a_i of one unit of emissions ends up in the artificial carbon box $M_{i,t}$. The calibration of the model is given in table 4.

One of the variables of interest for a climate model is the social cost of carbon, which is the welfare loss of emitting one unit of carbon emissions in terms of consumption units at time t. In this model it is not straightforward to define the social cost of carbon, since we have four artificial carbon boxes. One unit of emissions will lead to an increase of a_0 units of $M_{P,t}$, a_1 units of $M_{1,t}$, a_2 units of $M_{2,t}$ and a_3 units of $M_{3,t}$. The social cost of carbon then becomes:

$$SCC_t = -\frac{a_0 \frac{\partial V_t}{\partial M_{P,t}} + a_1 \frac{\partial V_t}{\partial M_{1,t}} + a_2 \frac{\partial V_t}{\partial M_{2,t}} + a_3 \frac{\partial V_t}{\partial M_{3,t}}}{f_C(C_t, V_t)}. \tag{39}$$

Note that the social cost of carbon scales with the endowment or consumption level, since damages are a fraction of endowment. Therefore the social cost of carbon will grow over time. The social cost of carbon in this paper is measured in \$ per ton carbon. To get the social cost of carbon per ton CO_2, the social cost of carbon must be divided by 3.67. As initial consumption level, we use $Y_0 = C_0 = 80$ trillion US \$ where the base year is chosen to be 2015. This is a proxy for world consumption using purchasing power parity (instead of exchange rates) in 2015. The vector of initial points at the base year 2015 is given by: $X_0 = [680.2 \;\; 83.2 \;\; 56.9 \;\; 11.1 \;\; 0.11 \;\; 0.41 \;\; 0.62]^{\emptyset}$.

L	N	δ_t	$k = \frac{C}{S}$	g	SCC (\$)	r	rp	time (s)
1	1	1	1.600%	0.134	242.2	1.058%	2.506%	2.8
2	8	1	1.601%	0.135	250.0	1.058%	2.506%	8.6
3	36	1	1.603%	0.136	253.0	1.058%	2.506%	57.4
4	120	1	1.603%	0.137	253.4	1.058%	2.506%	287.7
5	330	1	1.603%	0.137	253.0	1.058%	2.506%	1440.1
1	1	0.1	1.600%	0.134	242.3	1.058%	2.506%	17.9
2	8	0.1	1.602%	0.135	250.2	1.058%	2.506%	71.4
3	36	0.1	1.603%	0.136	253.2	1.058%	2.506%	501.6

Table 5: Consumption-wealth ratio, g, social cost of carbon, interest rate, risk premium and computation time for different algorithmic parameters.

5.2.1 Finite Difference Method

The problem that we have to solve is a 7-dimensional problem with explicit time dependence. We choose $\overline{X} = [3000\ \ 2000\ \ 250\ \ 50\ \ 8\ \ 3\ \ 10]^{\emptyset}$ and $\underline{X} = [600\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^{\emptyset}$. With these bounds, except for $M_{P,t}$, all state variables on the left of the grid have a positive drift and on the right of the drift have a negative drift. Therefore, we use as boundary condition that the second derivative vanishes, except for $M_{P,t}$. The drift of $M_{P,t}$ is always positive. Therefore we can again set the second derivative to zero at the left boundary. At the right boundary of $M_{P,t}$, we assume that the first derivative is equal to zero.

To get a good starting guess, we first solve the problem with the finite difference method at $T = 250$ with $\Delta = 50$ and $crit = 10^{\ 6}$ assuming that there are no emissions: $E_{250} = 0$. The time derivative of g_t comes from carbon emissions and therefore $\frac{\partial g_t}{\partial t} = 0$ in this setup. We use the outcome as our initial guess. Given our initial guess, we solve the actual problem with explicit time dependence starting from $T = 250$ and working backwards in time. A more 'stupid' initial guess would also work, as long as the stability conditions are met using that guess. The algorithm might take longer to converge, so T might have to be chosen larger in that case.

We use the sparse grid combination method and try different combinations of L and δ_t. Define by N the number of sub-problems that are solved, so N depends on the level of the sparse grid L. We use the semi-implicit method in all runs. The explicit method is not much faster and the semi-implicit method is more stable. One of the outputs of the finite difference method is the first derivative of g_t with respect to the vector of state variables X_t. This derivative is used to calculate the social cost of carbon. The results are given in table 5.

If we look at the table, it is clear that the run on a single grid ($L = 1$) and $\delta_t = 1$ already gives a reasonable approximation of the outcomes. If we increase the level L to 2 or 3, we see that the social cost of carbon becomes approximately 250\$. Increasing L even further does not change the outcomes much, and also choosing a smaller time step has little effect.

5.2.2 Least Squares Monte Carlo Method

We use again standard power functions as basis functions. However, in the multidimensional setting we also consider cross terms. Define in this case the highest order of the basis functions by L. Then if $L = 1$, we only consider the vector of linear first order basis functions:

$$B(X_t) = [1 \;\; M_{P,t} \;\; M_{1,t} \;\; M_{2,t} \;\; M_{3,t} \;\; T_{0,t} \;\; T_{1,t} \;\; T_{2,t}]^{\emptyset}. \tag{40}$$

When $L = 2$, we use basis functions up to order two, for example $(M_{P,t})^2$ and the cross-term $M_{P,t}M_{1,t}$. And for order $L = 3$ we consider e.g. $M_{P,t}^3$, $M_{P,t}^2 M_{1,t}$ and $M_{P,t}M_{1,t}M_{2,t}$, so all possible basis functions up to third order. This can be extended to any higher order. We define the number of basis functions again by N_b.

For this setup, there are no closed form expectations of the basis functions available. Using the Euler method, we can still find closed form approximate conditional expectations. The conditional expectation for the linear basis functions are straightforward: $E[X_{t_{i+1}}|X_{t_i}] = X_{t_i} + \mu_X(X_{t_i})\delta_t$. For the higher order and cross terms, we can use Ito-calculus and then use the Euler method approximation. For example:

$$\begin{aligned} dM_{2,t}^2 &= 2M_{2,t}dM_{2,t} + d[M_2, M_2]_t = \Big(2M_{2,t}(a_2E_t - \frac{1}{\tau_2}M_{2,t}) + \sigma_{M_2}^2\Big)dt + 2M_{2,t}\sigma_{M_2}dZ_t^{M^2} \\ E[M_{2,t_{i+1}}^2|M_{2,t_i}] &\approx M_{2,t_i}^2 + \Big(2M_{2,t_i}(a_2E_{t_i} - \frac{1}{\tau_2}M_{2,t_i}) + \sigma_{M_2}^2\Big)\delta_t. \end{aligned} \tag{41}$$

The conditional expectation for Y_t becomes:

$$\begin{aligned} E\Big[\frac{Y_{t_{i+1}}^{1\;\;\gamma}}{Y_{t_i}^{1\;\;\gamma}}|X_{t_i}(k)\Big] &= \exp\Big((1-\gamma)(\mu - \sigma^2/2)\delta_t + \frac{1}{2}(1-\gamma)^2\sigma^2\delta_t + \lambda_1\delta_t(E[(1+J_1)^{1\;\;\gamma}] - 1) \\ &+ \lambda_2(T_{0,t_i}(k) + T_{1,t_i}(k) + T_{2,t_i}(k))\delta_t(E[(1+J_2)^{1\;\;\gamma}] - 1)\Big). \end{aligned} \tag{42}$$

The algorithmic parameters that we use are $T = 500$ and $K = 1000$. To obtain the time-dependent boundaries, we first calculate the expected path of the state variables. This can be easily found by setting $\sigma_X = 0$ and solving the system forward using a standard differential equation solver. Around the expected path, we then create an interval for the simulation. Define by X_t^{av} the expected value of the vector of state variables at time t. Then at time t_i, we define the boundaries as: $\overline{X_{t_i}} = X_{t_i}^{av} + [150 \;\; 75 \;\; 37.5 \;\; 7.5 \;\; 1.5 \;\; 0.75 \;\; 1.5]^{\emptyset}$ and $\underline{X_{t_i}} = X_{t_i}^{av} - [100 \;\; 50 \;\; 25 \;\; 5 \;\; 1 \;\; 0.5 \;\; 1]^{\emptyset}$. If the lower bound becomes negative, we set it equal to zero.

Table 6 shows the results for different combinations of L and δ_t. We start with $\delta_t = 1$. The first run with only linear basis functions already gives reasonable results. This implies that the function g is not very non-linear in the state variables. Increasing L to 2 improves the results, but when L is increased to 3 or 4 the results become unstable. The SCC explodes. The reason for this is that the conditional expectations of the basis functions are an approximation that depends on the time step, in contrast to the time-varying jump risk example where the conditional expectations could be calculated in closed form. Especially for the higher order basis functions,

L	N_b	δ_t	$k = \frac{C}{S}$	g	SCC (\$)	r	rp	time (s)
1	8	1	1.606%	0.139	259.2	1.058%	2.506%	0.3
2	36	1	1.606%	0.139	249.5	1.058%	2.506%	1.1
3	120	1	1.373%	0.034	16602.7	1.055%	2.511%	4.7
4	330	1	1.803%	0.394	1127.8	1.058%	2.506%	19.0
1	8	0.1	1.603%	0.137	261.0	1,058%	2.506%	2.7
2	36	0.1	1.604%	0.137	251.4	1.058%	2.506%	10.3
3	120	0.1	1.604%	0.137	250.4	1.058%	2.506%	49.7
4	330	0.1	1.604%	0.137	250.4	1.058%	2.506%	224.2
1	8	0.01	1.603%	0.136	261.1	1.058%	2.506%	27.3
2	36	0.01	1.603%	0.137	251.6	1.058%	2.506%	101.7
3	120	0.01	1.603%	0.137	250.6	1.058%	2.506%	496.1
4	330	0.01	1.603%	0.137	250.6	1.058%	2.506%	2333.7

Table 6: Consumption-wealth ratio, g, social cost of carbon, interest rate, risk premium and computation time for different algorithmic parameters.

this approximation seems to be not very accurate. If we decrease the time step δ_t to 0.1 or 0.01, the results also become stable for $L = 3$ and $L = 4$.

5.2.3 Graphical results

In this section, we present some graphical results of the evolution of the problem over time. Since the problem is stochastic, we do not know the exact future path of the state variables. The output of the solution methods gives the value function and therefore the social cost of carbon at any time t for any combination of the state variables. The results are graphically presented for the expected path of the state variables. Figure 4 shows the climate variables, social cost of carbon, consumption-wealth ratio, risk-free rate and risk premium. Emissions are exogenous and are modeled to peak at the end of the century. Both the carbon concentration and the temperature keep increasing over time. The social cost of carbon at 2015 is approximately 250\$ and grows over time, since endowment also grows over time. The risk-free rate slightly declines over time and the risk premium increases due to higher climate risk, but both effects are quantitatively small.

In terms of computation time, the LSMC method is more efficient. However, both methods are able to solve this multidimensional problem in reasonable computation time.

5.3 Multidimensional climate problem with control variable

In the last example we consider a very similar model as in the previous example, but we add a control variable. In the previous example emissions were entirely exogenous. In this example, the agent can reduce emissions using abatement policy. Assume that $\tilde{E}_t = E_t(1 - u_t)$ are actual carbon emissions where E_t are business as usual

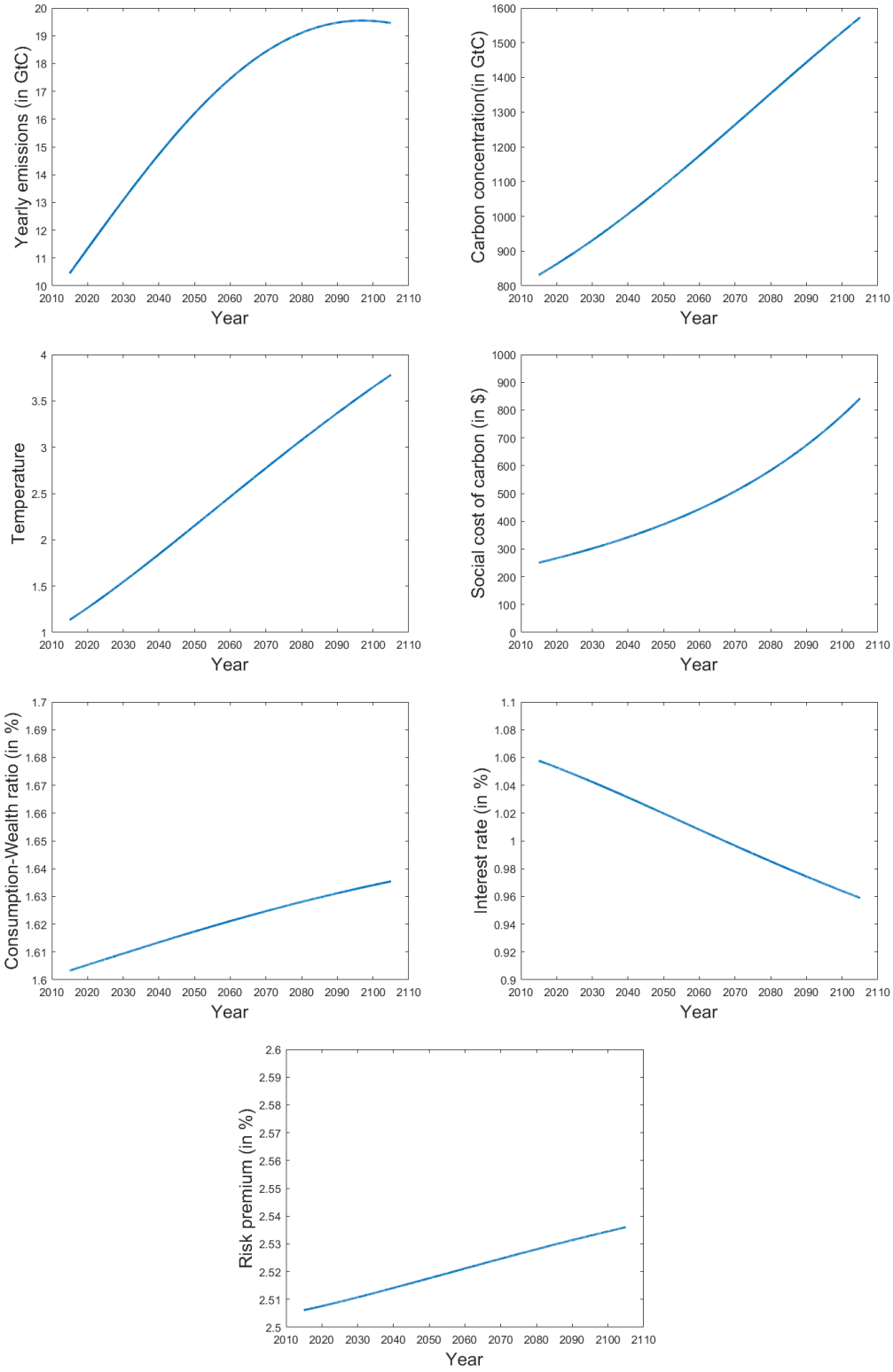

Figure 4: The expected paths of emissions, carbon concentration, temperature, social cost of carbon, consumption-wealth ratio, risk-free rate and the risk premium over time.

carbon emissions. u_t is then the emissions control rate at time t. In this example we assume that endowment Y_t can be spent on consumption C_t and abatement A_t. Abatement costs are proportional to output: $A_t = c_{1,t} u_t^{c_2} Y_t$ where $c_2 > 1$ is used to capture that the marginal cost of abatement increases in the emissions control rate and c_t^1 is declining over time to take into account technological process in abatement technologies. This yields $C_t = Y_t - A_t = (1 - c_{1,t} u_t^{c_2}) Y_t = \xi_t Y_t$ where $\xi_t = 1 - c_{1,t} u_t^{c_2}$. We assume that $u_t \leq 1$, so that it is not possible to take carbon out of atmosphere. At best, it is possible to reduce carbon emissions to zero.

The set of state variables and initial conditions are the same as in the previous example. The only difference is that now in the drift μ_X of the state variables, business as usual emissions E_t are replaced by controlled emissions $\tilde{E}_t$. The calibration is also identical, but we have two additional abatement parameters now. These are calibrated to match the abatement function of the dice model (Nordhaus, 2017): $c_{1,t} = 0.074 \exp(-0.019t)$ and $c_2 = 2.8$. As initial endowment level, we take $Y_0 = 80$ trillion US \$. The HJB-equation corresponding to this problem is:

$$0 = \min_{u_t} \Big\{ \Big(\beta\zeta \Big(g_t^{\ \frac{1}{\zeta}} \xi_t^{1\ \ 1/\epsilon} - 1 \Big) + (1-\gamma) \Big(\mu - \frac{1}{2}\gamma\sigma^2 + \lambda_1 E\Big[\frac{(1+J_1)^{1\ \ \gamma} - 1}{1-\gamma} \Big] \\ + \lambda_2 (T_{0,t} + T_{1,t} + T_{2,t}) E\Big[\frac{(1+J_2)^{1\ \ \gamma} - 1}{1-\gamma} \Big] \Big) \Big) g_t + \frac{\partial g_t}{\partial t} + g_X \mu_X + \frac{1}{2} tr\Big(g_{XX} \sigma_X \sigma_X^{\emptyset} \Big) \Big\}. \tag{43}$$

Taking the first derivative, we obtain the first order condition:

$$\begin{aligned} 0 = &-\beta(1-\gamma) g_t^{\ 1\ \ \frac{1}{\zeta}} \xi_t^{\ 1/\epsilon} c_t^1 c_2 u_t^{c_2\ \ 1} \\ &- a_0 \frac{\partial g_t}{\partial M_{P,t}} E_t - a_1 \frac{\partial g_t}{\partial M_{1,t}} E_t - a_2 \frac{\partial g_t}{\partial M_{2,t}} E_t - a_3 \frac{\partial g_t}{\partial M_{3,t}} E_t. \end{aligned} \tag{44}$$

This first order condition can not be solved in closed-form for u_t and therefore the optimal policy u_t is implicitly defined by the first order condition.

5.3.1 Finite difference

If we consider the finite difference algorithm for this problem, we use exactly the same algorithmic parameters and boundary conditions. The only difference is that we calculate optimal policy at every time period and grid point. So again we first solve the problem with $E_{250} = 0$ and then solve the problem with explicit time dependence backward in time. The results of the finite difference methods are presented in table 7. Compared to the previous problem, this time we need a higher level L for the problem to converge. $L = 3$ already gives relatively accurate results. The time step again does not play an important role, $\delta_t = 1$ seems to be accurate enough. It is also clear that solving for the optimal control variable increases the computation time quite a lot, especially for the higher sparse grid levels.

5.3.2 Least squares monte carlo method

The LSMC method cannot directly be applied to this new setup, since in this case we cannot solve the system of expected state variables forward. We again assume

L	N	δ_t	$k = \frac{C}{S}$	g	SCC (\$)	u	r	rp	time (s)
1	1	1	1.584%	0.125	290.0	38.9%	1.045%	2.506%	11.7
2	8	1	1.586%	0.126	312.2	40.5%	1.043%	2.506%	38.7
3	36	1	1.587%	0.127	325.7	41.5%	1.042%	2.506%	303.4
4	120	1	1.587%	0.127	332.6	42.0%	1.042%	2.506%	2153.4
5	330	1	1.588%	0.128	333.5	42.0%	1.042%	2.506%	12373.5
1	1	0.1	1.584%	0.125	290.0	38.9%	1.045%	2.506%	90.3
2	8	0.1	1.586%	0.126	312.1	40.5%	1.043%	2.506%	344.6
3	36	0.1	1.587%	0.127	325.6	41.5%	1.042%	2.506%	3108.6

Table 7: Consumption-wealth ratio, g, social cost of carbon, optimal abatement policy, interest rate, risk premium and computation time for different algorithmic parameters.

$T = 500$ but in this case choose $K = 2500$. The reason that we increase K is because the problem is more non-linear compared to the problem without control variable. The non-linearity is due to the cap on the emissions control rate. As lower bound we take the following bounds for any period in time: $\underline{X} = [600 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\emptyset}$. To obtain an upper bound, we first assume a constant policy rule of $u = 0.5$. We then find the expected path of state variables X_t^{av} given $u = 0.5$ by solving the system forward with $\sigma_X = 0$. At time t_i, we define the upper boundary as: $\overline{X_{t_i}} = X_{t_i}^{av} + [150 \quad 75 \quad 37.5 \quad 7.5 \quad 1.5 \quad 0.75 \quad 1.5]^{\emptyset}$. We tried different choices for u and this does not seem to matter much, as long as the resulting interval turns out to be roughly around the optimal expected path of state variables. In general, the results seem to be very robust against differences in the simulation intervals.

The outcomes and computation times are given in table 8. Taking $L = 2$ and $\delta_t = 1$ provides reasonable results in a short computation time. In this case, the problem does not even converge for $L > 2$ and $\delta_t = 1$, due to the approximation error of the conditional expectations. Decreasing δ_t to 0.1 and choosing $L = 3$ gives accurate results in a few minutes. Increasing L or decreasing δ_t even further does not change the results much.

5.3.3 Graphical results

Again, we present results for the expected path of state variables over time. Since we do not know the optimal emissions control rate before solving the problem, we first solve the problem backward and then iterate forward to obtain the expected path of state variables. The graphs are presented in figure 5. The optimal emissions control rate equals roughly 40% in 2015 and increases over time. The reason of this increase over time is that the abatement costs are assumed to decrease over time. In the optimal policy scenario the emissions control rate reaches 100% between 2065 and 2070. The temperature level peaks at 1.7 degrees, which is in line with the Paris climate agreement.

The social cost of carbon at 2015 is approximately 330\$, which is larger than in the business as usual scenario. If we look at the consumption-endowment ratio, we see

L	N_b	δ_t	$k = \frac{C}{S}$	g	SCC (\$)	u	r	rp	time (s)
1	8	1	1.590%	0.129	347.4	43.0%	1.041%	2.506%	4.4
2	36	1	1.591%	0.130	326.5	41.5%	1.042%	2.506%	8.6
3	120	1	-	-	-	-	-	-	-
4	330	1	-	-	-	-	-	-	-
1	8	0.1	1.586%	0.127	350.7	43.2%	1.041%	2.506%	39.4
2	36	0.1	1.588%	0.128	329.4	41.7%	1.042%	2.506%	77.5
3	120	0.1	1.588%	0.128	327.8	41.6%	1.042%	2.506%	217.8
4	330	0.1	1.588%	0.128	327.7	41.6%	1.042%	2.506%	832.4
1	8	0.01	1.586%	0.127	351.0	43.2%	1.041%	2.506%	387.4
2	36	0.01	1.587%	0.127	329.7	41.7%	1.042%	2.506%	774.8
3	120	0.01	1.587%	0.127	328.1	41.6%	1.042%	2.506%	2237.2
4	330	0.01	1.587%	0.127	328.0	41.6%	1.042%	2.506%	8156.5

Table 8: Consumption-wealth ratio, g, social cost of carbon, optimal abatement policy, interest rate, risk premium and computation time for different algorithmic parameters.

that the slope of ξ_t changes at the point where the emissions control rate reaches its upper bound of 100%. Up to that moment, the amount of endowment that is spent on abatement is increasing. Although abatement becomes cheaper, the emissions control rate increases fast enough to make the total amount increasing. However, when the upper bound is reached, the amount spent on abatement decreases over time. At the moment that u_t reaches its upper bound there is a jump in the interest rate, since the growth rate of the endowment-consumption ratio (μ_ξ) jumps. However, quantitatively the effect of climate change on the risk-free rate and risk premium are not very large.

Comparing the two methods in this case, it is clear that the LSMC method outperforms the finite difference methods. The LSMC method converges faster and the computation time increases not as fast when the accuracy of the solution method is increased. The main reason is that the number of grid points increases much faster for the finite difference method than for the LSMC method. One of the most time-consuming parts of the algorithm is to find the optimal policy at every grid points. This leads to very large computation times for more accurate runs of the finite difference method. For multi-dimensional problems with optimal policy, it is therefore more efficient to use the LSMC method.

6 Conclusion

We have developed and compared two solution methods that are suitable to solve multi-dimensional DSGE models. The finite difference method has recently been popular to solve macro-models. We extend this method to the setting with Epstein-Zin preferences and show how to easily apply the finite difference method on sparse grids

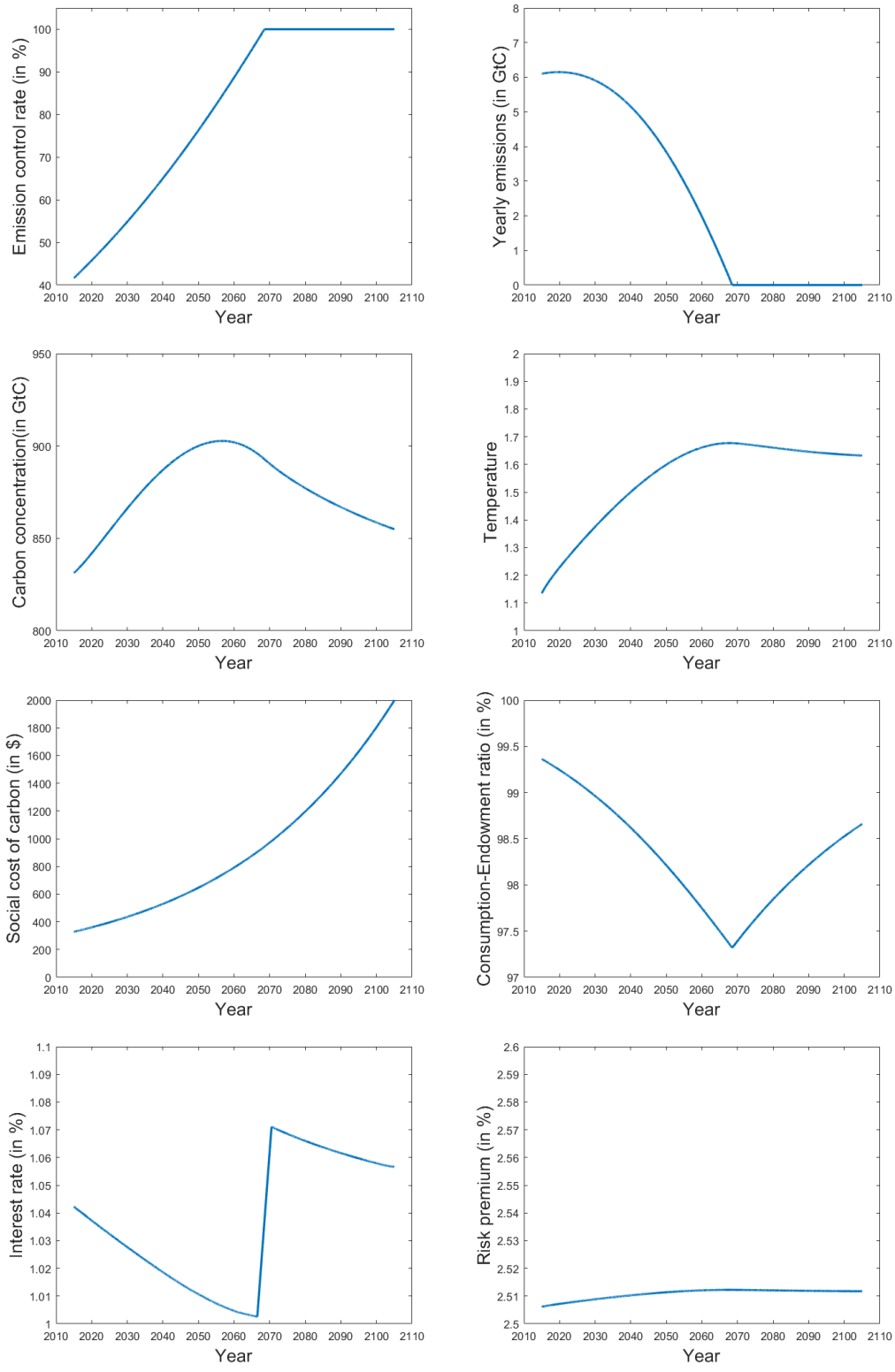

Figure 5: The expected paths of emissions, carbon concentration, temperature, social cost of carbon, consumption-endowment ratio, risk-free rate and the risk premium over time.

using the combination method. This method is capable of solving the 7-dimensional optimal control example of an economic model with a climate change externality. However, the computation time increases substantially for denser grids, which might be problematic if one wants to solve high-dimensional problems that are very non-linear.

The second method that we have discussed is the Least Squares Monte Carlo method that up to now mostly applied to option pricing and portfolio choice models. We show that this method is very suitable to solve optimal control DSGE models with multiple dimensions. The method is able to solve the example problem within a few minutes. It does not suffer from the curse of dimensionality. The conditional expectations are straightforward to calculate for any set of basis functions by applying Ito calculus. This makes the method flexible and makes it also able to solve non-linear problems. The order of basis functions can be increased to capture the non-linearities.

In this paper only three numerical examples have been considered. It would be interesting to see how these methods perform on problems with different characteristics. As mentioned before, another topic for further research is to see how the methods perform if there is correlation between the state variables.

References

Achdou, Y., Han, J., Lasry, J.-M., Lions, P.-L., & Moll, B. (2017). *Income and wealth distribution in macroeconomics: A continuous-time approach* (Tech. Rep.). National Bureau of Economic Research.

Aengenheyster, M., Feng, Q. Y., Van Der Ploeg, F., & Dijkstra, H. A. (2018). The point of no return for climate action. *Earth System Dynamics*, *9*(3).

Andreasson, J., & Shevchenko, P. V. (2019). Bias-corrected least-squares monte carlo for utility based optimal stochastic control problems. *Available at SSRN 2985828*.

Balata, A., & Palczewski, J. (2017). Regress-later monte carlo for optimal control of markov processes. *arXiv preprint arXiv:1712.09705*.

Barnett, M., Brock, W., & Hansen, L. P. (2020, 02). Pricing Uncertainty Induced by Climate Change. *The Review of Financial Studies*, *33*(3), 1024-1066. Retrieved from `https://doi.org/10.1093/rfs/hhz144` doi: 10.1093/rfs/hhz144

Barro, R. J. (2015). Environmental protection, rare disasters and discount rates. *Economica*, *82*(325), 1–23.

Brandt, M. W., Goyal, A., Santa-Clara, P., & Stroud, J. R. (2005). A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *The Review of Financial Studies*, *18*(3), 831–873.

Brumm, J., & Scheidegger, S. (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica*, *85*(5), 1575–1612.

Candler, G. V. (1999). Finite-difference methods for continuous-time dynamic programming. In *Computational methods for the study of dynamic economies.* Oxford University Press.

Dimson, E., Marsh, P., & Staunton, M. (2011). Equity premia around the world. *Available at SSRN 1940165*.

Duffie, D., & Epstein, L. G. (1992a). Asset pricing with stochastic differential utility. *The Review of Financial Studies*, *5*(3), 411–436.

Duffie, D., & Epstein, L. G. (1992b). Stochastic differential utility. *Econometrica: Journal of the Econometric Society*, 353–394.

Glasserman, P., & Yu, B. (2004). Simulation for american options: Regression now or regression later? In *Monte carlo and quasi-monte carlo methods 2002* (pp. 213–226). Springer.

Griebel, M. (1998). Adaptive sparse grid multilevel methods for elliptic pdes based on finite differences. *Computing*, *61*(2), 151–179.

Griebel, M., Schneider, M., & Zenger, C. (1990). A combination technique for the solution of sparse grid problems.

Harwood, R. C. (2017). Steady and stable: Numerical investigations of nonlinear partial differential equations. In *A primer for undergraduate research* (pp. 265–303). Springer.

Ikefuji, M., Laeven, R. J., Magnus, J. R., & Muris, C. (2020). Expected utility and catastrophic risk in a stochastic economy–climate model. *Journal of Econometrics*, *214*(1), 110–129.

Jain, S., Leitao, Á., & Oosterlee, C. W. (2019). Rolling adjoints: fast greeks along monte carlo scenarios for early-exercise options. *Journal of Computational Science*, *33*, 95–112.

Jain, S., & Oosterlee, C. W. (2015). The stochastic grid bundling method: Efficient pricing of bermudan options and their greeks. *Applied Mathematics and Computation*, *269*, 412–431.

Koijen, R. S., Nijman, T. E., & Werker, B. J. (2010). When can life cycle investors benefit from time-varying bond risk premia? *The review of nancial studies*, *23*(2), 741–780.

Lapeyre, B., Sulem, A., & Talay, D. (2005). Simulation of financial models: Mathematical foundations and applications. *to appear*.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing american options by simulation: a simple least-squares approach. *The review of nancial studies*, *14*(1), 113–147.

Ma, K., & Forsyth, P. (2017). An unconditionally monotone numerical scheme for the two-factor uncertain volatility model. *IMA Journal of Numerical Analysis*, *37*(2), 905–944.

Martin, I. W., & Pindyck, R. S. (2015). Averting catastrophes: The strange economics of scylla and charybdis. *American Economic Review*, *105*(10), 2947–85.

Mehra, R., & Prescott, E. C. (1985). The equity premium: A puzzle. *Journal of monetary Economics*, *15*(2), 145–161.

Nordhaus, W. D. (2017). Revisiting the social cost of carbon. *Proceedings of the National Academy of Sciences*, *114*(7), 1518–1523.

Olijslagers, S., & van Wijnbergen, S. (2019). Discounting the future: on climate change, ambiguity aversion and epstein-zin preferences.

Pflüger, D. M. (2010). *Spatially adaptive sparse grids for high-dimensional problems* (Unpublished doctoral dissertation). Technische Universität München.

Ruttscheidt, S. (2018). *Adaptive sparse grids for solving continuous time heterogeneous agent models* (Master thesis). Universität Bonn, Bonn.

Shao, A. (2012). *A fast and exact simulation for cir process* (Unpublished doctoral dissertation). University of Florida Gainesville, FL.

Thomas, J. W. (2013). *Numerical partial di erential equations: nite di erence methods* (Vol. 22). Springer Science & Business Media.

Tsai, J., & Wachter, J. A. (2018). Pricing long-lived securities in dynamic endowment economies. *Journal of Economic Theory*, *177*, 848–878.

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

Wachter, J. A. (2013). Can time-varying risk of rare disasters explain aggregate stock market volatility? *The Journal of Finance*, *68*(3), 987–1035.

Zumbusch, G. W. (2000). A sparse grid pde solver; discretization, adaptivity, software design and parallelization. In *Advances in software tools for scienti c computing* (pp. 133–177). Springer.

A Asset pricing

A.1 Wealth-consumption ratio

S_t equals the total wealth of the representative agent. At the optimum, the following condition is satisfied: $f_C = V_S$ (see for example Tsai and Wachter (2018)). Now define the consumption-wealth ratio by $k_t = \frac{C_t}{S_t}$. Using the chain rule, this implies that $V_S = V_C k_t$). The optimality condition then implies that: $k_t = \frac{f_C}{V_C}$. We can calculate f_C:

$$f_C(C,V) = \frac{\beta C^{\ 1/\epsilon}}{\big((1-\gamma)V\big)^{\frac{1}{\zeta}\ \ 1}}. \tag{45}$$

Substituting in $f_c, V_t = g_t \frac{Y_t^{1-\gamma}}{1\ \ \gamma} = g_t \frac{(\frac{C_t}{\xi_t})^{1-\gamma}}{1\ \ \gamma}$ and $V_C = g_t \xi_t^{\gamma\ \ 1} C_t^{\ \gamma}$ gives:

$$k_t = \beta g_t^{\ \frac{1}{\zeta}} \xi_t^{1\ \ 1/\epsilon}. \tag{46}$$

A.2 Pricing kernel

Duffie and Epstein (1992a) derive that the pricing kernel with stochastic differential utility equals $\pi_t = \exp\Big\{\int_0^t f_V(C_s,V_s)ds\Big\} f_C(C_t,V_t)$.

We will start with deriving the explicit stochastic differential equation of the pricing kernel. First we calculate the derivatives of $f(C,V)$ with respect to V (the derivative with respect to C is given in the previous subsection).

$$f_V(C,V) = \beta\zeta\Big\{\Big(1-\frac{1}{\zeta}\Big)\Big((1-\gamma)V\Big)^{\ \frac{1}{\zeta}} C^{1\ \ 1/\epsilon} - 1\Big\} \tag{47}$$

Substituting $V_t = g_t \frac{Y_t^{1-\gamma}}{1-\gamma}$ and $C_t = \xi_t Y_t$ into $f_C(C,V)$ and $f_V(C,V)$ we obtain:

$$f_C(C_t, V_t) = \beta \xi_t^{1/\epsilon} g_t^{1-\frac{1}{\zeta}} Y_t^{\gamma} \tag{48}$$

$$f_V(C_t, V_t) = \beta\zeta\Big\{g_t^{\frac{1}{\zeta}} \xi_t^{1-1/\epsilon}\Big(1-\frac{1}{\zeta}\Big) - 1\Big\} \tag{49}$$

This gives:

$$\pi_t = \exp\left(\int_0^t \beta\zeta\Big\{g_s^{\frac{1}{\zeta}} \xi_s^{1-1/\epsilon}\Big(1-\frac{1}{\zeta}\Big) - 1\Big\} ds\right) \beta \xi_t^{1/\epsilon} g_t^{1-\frac{1}{\zeta}} Y_t^{\gamma}. \tag{50}$$

Write as a differential equation:

$$\frac{d\pi_t}{\pi_t} = \beta\zeta\Big\{g_t^{\frac{1}{\zeta}} \xi_t^{1-1/\epsilon}\Big(1-\frac{1}{\zeta}\Big) - 1\Big\} dt + \frac{dY_t^{\gamma}}{Y_t^{\gamma}} + \frac{dg_t^{1-1/\zeta}}{g_t^{1-1/\zeta}} + \frac{d\xi_t^{1/\epsilon}}{\xi_t^{1/\epsilon}} + \frac{d[g_t^{1-1/\zeta}, \xi_t^{1/\epsilon}]}{g_t^{1-1/\zeta} \xi_t^{1/\epsilon}}. \tag{51}$$

Apply Ito's lemma to Y_t^{γ} gives:

$$dY_t^{\gamma} = -\gamma\Big(\mu_t - \frac{1}{2}(\gamma+1)\sigma_t^2\Big) Y_t^{\gamma} dt - \gamma\sigma_t Y_t^{\gamma} dZ_t^c + \sum_{m=1}^{M} \Big((1+J_m)^{\gamma} - 1\Big) Y_t^{\gamma} dN_{m,t}. \tag{52}$$

We can also apply Ito's lemma to $g_t^{1-1/\zeta}$:

$$\begin{aligned} dg_t &= \Big(\frac{\partial g_t}{\partial t} + g_X \mu_X + \frac{1}{2} tr(g_{XX}\sigma_X\sigma_X^{\intercal})\Big) dt + g_X \sigma_X dZ_t^X \\ dg_t^{1-1/\zeta} &= (1-1/\zeta)\left(\frac{\frac{\partial g_t}{\partial t}}{g_t} + \frac{g_X}{g_t}\mu_X + \frac{1}{2} tr\Big(\frac{g_{XX}}{g_t}\sigma_X\sigma_X^{\intercal}\Big)\right. \\ &\quad \left. - \frac{1}{2}\frac{1}{\zeta} tr\Big(\frac{g_X^{\intercal} g_X}{g_t^2}\sigma_X\sigma_X^{\intercal}\Big)\right) g_t^{1-1/\zeta} dt + (1-1/\zeta)\frac{g_X}{g_t} g_t^{1-1/\zeta} \sigma_X dZ_t^X. \end{aligned} \tag{53}$$

And lastly, applying Ito's lemma to $\xi_t^{1/\epsilon}$ gives:

$$\begin{aligned} d\xi_t &= \Big(\frac{\partial \xi_t}{\partial t} + \xi_X \mu_X + \frac{1}{2} tr(\xi_{XX}\sigma_X\sigma_X^{\intercal})\Big) dt + \xi_X \sigma_X dZ_t^X \\ d\xi_t^{1/\epsilon} &= -1/\epsilon\left(\frac{\frac{\partial \xi_t}{\partial t}}{\xi_t} + \frac{\xi_X}{\xi_t}\mu_X + \frac{1}{2} tr\Big(\frac{\xi_{XX}}{\xi_t}\sigma_X\sigma_X^{\intercal}\Big)\right. \\ &\quad \left. - \frac{1}{2}(1+1/\epsilon) tr\Big(\frac{\xi_X^{\intercal}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\intercal}\Big)\right) \xi_t^{1/\epsilon} dt - 1/\epsilon \frac{\xi_X}{\xi_t} \xi_t^{1/\epsilon} \sigma_X dZ_t^X. \end{aligned} \tag{54}$$

The quadratic covariation between $g_t^{1-1/\zeta}$ and $\xi_t^{1/\epsilon}$ equals:

$$\frac{d[g_t^{1-1/\zeta}, \xi_t^{1/\epsilon}]}{g_t^{1-1/\zeta} \xi_t^{1/\epsilon}} = -1/\epsilon(1-1/\zeta) tr\Big(\frac{g_X^{\intercal}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\intercal}\Big) dt \tag{55}$$

Define $\mu_g = \frac{\frac{\partial g_t}{\partial t}}{g_t} + \frac{g_X}{g_t}\mu_X + \frac{1}{2}tr\Big(\frac{g_{XX}}{g_t}\sigma_X\sigma_X^\emptyset\Big)$ and $\mu_\xi = \frac{\frac{\partial \xi_t}{\partial t}}{\xi_t} + \frac{\xi_X}{\xi_t}\mu_X + \frac{1}{2}tr\Big(\frac{\xi_{XX}}{\xi_t}\sigma_X\sigma_X^\emptyset\Big)$. Combining everything gives:

$$\begin{aligned}\frac{d\pi_t}{\pi_t} =&\Big\{\beta\zeta\Big(g_t^{\ \frac{1}{\zeta}}\xi_t^{1\ \ 1/\epsilon}\big(1-\frac{1}{\zeta}\big)-1\Big)-\gamma\Big(\mu_t-\frac{1}{2}(\gamma+1)\sigma_t^2\Big)\\&-1/\epsilon\Big(\mu_\xi-\frac{1}{2}(1+1/\epsilon)tr\Big(\frac{\xi_X^\emptyset\xi_X}{\xi_t^2}\sigma_X\sigma_X^\emptyset\Big)\Big)-1/\epsilon(1-1/\zeta)tr\Big(\frac{g_X^\emptyset\xi_X}{g_t\xi_t}\sigma_X\sigma_X^\emptyset\Big)\\&+(1-1/\zeta)\Big(\mu_g-\frac{1}{2}\frac{1}{\zeta}tr\Big(\frac{g_X^\emptyset g_X}{g_t^2}\sigma_X\sigma_X^\emptyset\Big)\Big)\Big\}dt\\&-\gamma\sigma_t dZ_t^c+\Big((1-1/\zeta)\frac{g_X}{g_t}\sigma_X-1/\epsilon\frac{\xi_X}{\xi_t}\sigma_X\Big)dZ_t^X+\sum_{m=1}^{M}\Big((1+J_m)^{\ \gamma}-1\Big)dN_{m,t}\end{aligned} \tag{56}$$

We can now substitute the HJB equation into the pricing kernel. Note that we can rewrite the HJB-equation as:

$$\mu_g = -\beta\zeta\Big(g_t^{\frac{-1}{\zeta}}\xi_t^{1\ \ 1/\epsilon}-1\Big)-(1-\gamma)\Big(\mu_t-\frac{1}{2}\gamma\sigma_t^2+\sum_{m=1}^{M}\lambda_{m,t}E\Big[\frac{(1+J_m)^{1\ \gamma}-1}{1-\gamma}\Big]\Big). \tag{57}$$

Substituting this gives:

$$\begin{aligned}\frac{d\pi_t}{\pi_t} =&\Big\{-\beta-\frac{\mu_t}{\epsilon}+\Big(1+\frac{1}{\epsilon}\Big)\frac{\gamma}{2}\sigma_t^2+\Big(\gamma-\frac{1}{\epsilon}\Big)\sum_{m=1}^{M}\lambda_t^m E\Big[\frac{(1+J^m)^{1\ \gamma}-1}{1-\gamma}\Big]\\&-1/\epsilon\Big(\mu_\xi-\frac{1}{2}(1+1/\epsilon)tr\Big(\frac{\xi_X^\emptyset\xi_X}{\xi_t^2}\sigma_X\sigma_X^\emptyset\Big)\Big)-1/\epsilon(1-1/\zeta)tr\Big(\frac{g_X^\emptyset\xi_X}{g_t\xi_t}\sigma_X\sigma_X^\emptyset\Big)\\&-\frac{1}{2}\frac{1}{\zeta}(1-1/\zeta)tr\Big(\frac{g_X^\emptyset g_X}{g_t^2}\sigma_X\sigma_X^\emptyset\Big)\Big\}dt\\&-\gamma\sigma_t dZ_t^c+\Big((1-1/\zeta)\frac{g_X}{g_t}\sigma_X-1/\epsilon\frac{\xi_X}{\xi_t}\sigma_X\Big)dZ_t^X+\sum_{m=1}^{M}\Big((1+J_m)^{\ \gamma}-1\Big)dN_{m,t}.\end{aligned} \tag{58}$$

Therefore π_t is of the form: $\frac{d\pi_t}{\pi_{t-}} = \mu_\pi dt - \gamma\sigma_t dZ_t^c + \sigma_\pi dZ_t^X + \sum_{m=1}^{M}\Big((1+J_m)^{\ \gamma}-1\Big)dN_{m,t}$.

A.3 Interest rate

Let B_t be the price of the risk-free asset with continuous return r_t: $dB_t = r_t B_t dt$. Using a no-arbitrage argument, $\pi_t B_t$ must be a martingale:

$$\frac{d\pi_t B_t}{\pi_t\ B_t} = (r_t+\mu_\pi)dt - \gamma\sigma_t dZ_t^c + \sigma_\pi dZ_t^X + \sum_{m=1}^{M}\Big((1+J_m)^{\ \gamma}-1\Big)dN_{m,t}. \tag{59}$$

This is a martingale if $r_t = -\mu_\pi - \sum_{m=1}^{M} \lambda_{m,t} E\Big[(1+J_m)^{\ -\gamma} - 1\Big]$. Therefore the interest rate equals:

$$\begin{aligned} r_t = \beta + \frac{\mu_t}{\epsilon} - \Big(1+\frac{1}{\epsilon}\Big)\frac{\gamma}{2}\sigma_t^2 - \frac{1}{2}\frac{1}{\zeta}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X \sigma_X^{\emptyset}\Big) \\ + 1/\epsilon\Big(\mu_\xi - \frac{1}{2}(1+1/\epsilon)tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\Big) - 1/\epsilon(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big) \\ - \Big(\gamma - \frac{1}{\epsilon}\Big)\sum_{m=1}^{M}\lambda_{m,t}E\Big[\frac{(1+J_m)^{1\ -\gamma} - 1}{1-\gamma}\Big] - \sum_{m=1}^{M}\lambda_{m,t}E\Big[(1+J_m)^{\ -\gamma} - 1\Big]. \end{aligned} \tag{60}$$

A.4 Risk premium

Let S_t be the ex-dividend price of the stock that pays dividend at a rate C_t and denote by S_t^d the cum-dividend price. Then by no arbitrage, $\pi_t S_t^d$ must be a martingale. Furthermore, we use the relationship $S_t = \frac{\xi_t}{k(X_t,t)}Y_t = \frac{\xi_t^{1/\epsilon} g_t^{1/\zeta} Y_t}{\beta}$. The dynamics of S_t^d are given by:

$$dS_t^d = dS_t + C_t dt = S_t\frac{dY_t}{Y_t} + S_t\frac{dg_t^{1/\zeta}}{g_t^{1/\zeta}} + S_t\frac{d\xi_t^{1/\epsilon}}{\xi_t^{1/\epsilon}} + S_t\frac{d[g_t^{1/\zeta}, \xi_t^{1/\epsilon}]}{g_t^{1/\zeta}\xi_t^{1/\epsilon}} + \beta\xi_t^{1\ \ 1/\epsilon} g_t^{\ 1/\zeta} S_t dt \tag{61}$$

Now we calculate $dg_t^{1/\zeta}$ and $d\xi_t^{1/\epsilon}$ and the quadratic covariation:

$$\begin{aligned} \frac{dg_t^{1/\zeta}}{g_t^{1/\zeta}} &= 1/\zeta\left(\mu_g + \frac{1}{2}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\right)dt + 1/\zeta\frac{g_X}{g_t}\sigma_X dZ_t^X \\ \frac{d\xi_t^{1/\epsilon}}{\xi_t^{1/\epsilon}} &= 1/\epsilon\left(\mu_\xi + \frac{1}{2}(1/\epsilon - 1)tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\right)dt + 1/\epsilon\frac{\xi_X}{\xi_t}\sigma_X dZ_t^X \\ \frac{d[g_t^{1/\zeta}, \xi_t^{1/\epsilon}]}{g_t^{1/\zeta}\xi_t^{1/\epsilon}} &= \frac{1}{\epsilon}\frac{1}{\zeta}tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big). \end{aligned} \tag{62}$$

Substituting this into (61) yields:

$$\begin{aligned} \frac{dS_t^d}{S_t} = \Bigg\{\mu_t + 1/\zeta\left(\mu_g + \frac{1}{2}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\right) + 1/\epsilon\left(\mu_\xi + \frac{1}{2}(1/\epsilon - 1)tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\right) \\ + \frac{1}{\epsilon}\frac{1}{\zeta}tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big)\Bigg\}dt + \Big(1/\zeta\frac{g_X}{g_t}\sigma_X + 1/\epsilon\frac{\xi_X}{\xi_t}\sigma_X\Big)dZ_t^X + \sigma_t dZ_t^c + \sum_{m=1}^{M} J_m dN_{m,t} \end{aligned} \tag{63}$$

Substituting μ_g (57) and taking everything together gives:

$$\begin{aligned}\mu_S = \beta + \frac{\mu_t}{\epsilon} - \frac{1}{2}(1/\epsilon - 1)\gamma\sigma_t^2 + (1/\epsilon - 1)\sum_{m=1}^{M}\lambda_{m,t}E\Big[\frac{(1+J_m)^{1\ \ \gamma} - 1}{1-\gamma}\Big] \\ + \frac{1}{2}\frac{1}{\zeta}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X\sigma_X^{\emptyset}\Big) + 1/\epsilon\Big(\mu_\xi + \frac{1}{2}(1/\epsilon - 1)tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big)\Big) \\ + \frac{1}{\epsilon}\frac{1}{\zeta}tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big).\end{aligned} \tag{64}$$

From this equation we can calculate the risk premium rp_t:

$$\begin{aligned}rp_t = \mu_S + \sum_{m=1}^{M}\lambda_{m,t}E[J_m] - r_t = \gamma\sigma_t^2 + \sum_{m=1}^{M}\lambda_{m,t}E\Big[J_m + (1+J_m)^{\ \ \gamma} - (1+J_m)^{1\ \ \gamma}\Big] \\ + \frac{1}{\zeta}(1/\zeta - 1)tr\Big(\frac{g_X^{\emptyset} g_X}{g_t^2}\sigma_X\sigma_X^{\emptyset}\Big) + \frac{1}{\epsilon^2}tr\Big(\frac{\xi_X^{\emptyset}\xi_X}{\xi_t^2}\sigma_X\sigma_X^{\emptyset}\Big) + \frac{1}{\epsilon}\Big(\frac{2}{\zeta} - 1\Big)tr\Big(\frac{g_X^{\emptyset}\xi_X}{g_t\xi_t}\sigma_X\sigma_X^{\emptyset}\Big).\end{aligned} \tag{65}$$

B Definition of finite difference matrix

We first define the backward, central and forward parameters α for the interior grid points.

$$\begin{aligned}&\alpha^C_{i,j,t} = -\sum_{d=1}^{2}\frac{|(\mu_{Xi,j,t})_d|}{\delta_d} - \sum_{d=1}^{2}\frac{\Big(\sigma_{Xi,j,t}\sigma^{\emptyset}_{Xi,j,t}\Big)_{d,d}}{\delta_d^2}, \quad i \in \{2,...,N_1-1\},\ j \in \{2,...,N_2-1\} \\ &\alpha^B_{i,j,t,d} = \frac{(\mu_{Xi,j,t})_d}{\delta_d} + \frac{1}{2}\frac{\Big(\sigma_{Xi,j,t}\sigma^{\emptyset}_{Xi,j,t}\Big)_{d,d}}{\delta_d^2} \quad \alpha^F_{i,j,t,d} = \frac{(\mu^+_{Xi,j,t})_d}{\delta_d} + \frac{1}{2}\frac{\Big(\sigma_{Xi,j,t}\sigma^{\emptyset}_{Xi,j,t}\Big)_{d,d}}{\delta_d^2} \\ &d=1,\ i \in \{2,...,N_1-1\},\ j \in \{1,...,N_2\}, \quad d=2,\ i \in \{1,...,N_1\},\ j \in \{2,...,N_2-1\}\end{aligned} \tag{66}$$

In this example, assume that $(\mu_{Xi,j,t})_1 > 0$ when $i = 1$, $(\mu_{Xi,j,t})_2 > 0$ when $j = 1$, $(\mu_{Xi,j,t})_1 < 0$ when $i = N_1$ and $(\mu_{Xi,j,t})_2 < 0$ $j = N_2$. This implies that at all left boundaries, the forward difference is used and at all right boundaries, the backward difference is always used in the upwind scheme. At the boundary points we then assume that the second derivative vanishes. The α parameters at the boundaries then become.

$$\begin{aligned}&\alpha^C_{i,j,t} = -\sum_{d=1}^{2}\frac{|(\mu_{Xi,j,t})_d|}{\delta_d}, \quad (i,j) \in \{(1,1),(1,N_2),(N_1,1),(N_1,N_2)\} \\ &\alpha^C_{i,j,t} = -\sum_{d=1}^{2}\frac{|(\mu_{Xi,j,t})_d|}{\delta_d} - \frac{\Big(\sigma_{Xi,j,t}\sigma^{\emptyset}_{Xi,j,t}\Big)_{2,2}}{\delta_2^2}, \quad i \in \{1,N_1\},\ j \in \{2,...,N_2-1\} \\ &\alpha^C_{i,j,t} = -\sum_{d=1}^{2}\frac{|(\mu_{Xi,j,t})_d|}{\delta_d} - \frac{\Big(\sigma_{Xi,j,t}\sigma^{\emptyset}_{Xi,j,t}\Big)_{1,1}}{\delta_1^2}, \quad i \in \{2,..,N_1-1\},\ j \in \{1,N_2\}\end{aligned} \tag{67}$$

$$
\begin{aligned}
&\alpha^B_{1,j,t,1} = 0, \quad j \in \{1,...,N_2\}, \quad \alpha^B_{i,1,t,2} = 0, \quad i \in \{1,...,N_1\} \\
&\alpha^B_{N_1,j,t,1} = \frac{(\mu_{XN_1,j,t})_1}{\delta_1}, \quad j \in \{1,...,N_2\}, \quad \alpha^B_{i,N_2,t,2} = \frac{(\mu_{Xi,N_2,t})_2}{\delta_2}, \quad i \in \{1,...,N_1\} \\
&\alpha^F_{1,j,t,1} = \frac{(\mu^+_{X1,j,t})_1}{\delta_1}, \quad j \in \{1,...,N_2\}, \quad \alpha^F_{i,1,t,2} = \frac{(\mu^+_{Xi,1,t})_2}{\delta_2}, \quad i \in \{1,...,N_1\} \\
&\alpha^F_{N_1,j,t,1} = 0, \quad j \in \{1,...,N_2\}, \quad \alpha^F_{i,N_2,t,2} = 0, \quad i \in \{1,...,N_1\}
\end{aligned}
\tag{68}
$$

Putting everything together, we can construct the matrix D^δ_t.

$$
D^C_{j,t} = \begin{bmatrix}
\alpha^C_{1,j,t} & \alpha^F_{1,j,t,1} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
\alpha^B_{2,j,t,1} & \alpha^C_{2,j,t} & \alpha^F_{2,j,t,1} & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & \alpha^B_{3,j,t,1} & \alpha^C_{3,j,t} & \alpha^F_{3,j,t,1} & \dots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \dots & \alpha^B_{N_1\ 2,j,t,1} & \alpha^C_{N_1\ 2,j,t} & \alpha^F_{N_2\ 2,j,t,1} & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & \alpha^B_{N_1\ 1,j,t,1} & \alpha^C_{N_1\ 1,j,t} & \alpha^F_{N_1\ 1,j,t,1} \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & \alpha^B_{N_1,j,t,q} & \alpha^C_{N_1,j,t}
\end{bmatrix}
$$

$$
D^B_{j,t} = \begin{bmatrix}
\alpha^B_{1,j,t,2} & 0 & \dots & 0 & 0 \\
0 & \alpha^B_{2,j,t,2} & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & \alpha^B_{N_1\ 1,j,t,2} & 0 \\
0 & 0 & \dots & 0 & \alpha^B_{N_1,j,t,2}
\end{bmatrix}
$$

$$
D^F_{j,t} = \begin{bmatrix}
\alpha^F_{1,j,t,2} & 0 & \dots & 0 & 0 \\
0 & \alpha^F_{2,j,t,2} & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & \alpha^F_{N_1\ 1,j,t,2} & 0 \\
0 & 0 & \dots & 0 & \alpha^F_{N_1,j,t,2}
\end{bmatrix}
$$

$$
D^\delta_t = \begin{bmatrix}
D^C_{1,t} & D^F_{1,t} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
D^B_{2,t} & D^C_{2,t} & D^F_{2,t} & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & D^B_{3,t} & D^C_{3,t} & D^F_{3,t} & \dots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \dots & D^B_{N_2\ 2,t} & D^C_{N_2\ 2,t} & D^F_{N_2\ 2,t} & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & D^B_{N_2\ 1,t} & D^C_{N_2\ 1,t} & D^F_{N_2\ 1,t} \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & D^C_{N_2,t} & D^F_{N_2,t}
\end{bmatrix}
\tag{69}
$$

C Consistency, convergence and stability

A proper finite difference scheme should satisfy the following three properties: consistency, convergence and stability. We will briefly cover these three properties, see Thomas (2013) for more details. Let us start with consistency. Denote the true solution at the grid points at time t_i by g_{t_i} and the approximate solution by $g^\delta_{t_i}$. A

scheme $g^{\delta}_{t_i} = Ag^{\delta}_{t_{i+1}}$ is consistent with respect to the norm $||.||$ if g_{t_i} satisfies:

$$g_{t_i} = Ag_{t_{i+1}} + \delta_t \tau_{t_{i+1}}, \quad \text{and } ||\tau_{t_i}|| \to 0 \text{ as } \delta_t, \delta_1, \delta_2 \to 0 \tag{70}$$

τ_{t_i} is the error vector of the finite difference approximation. We did not yet explicitly specify the norm, but one can use for example the absolute sum ($||.||_1$), the Euclidean norm ($||.||_2$) or the max absolute value ($||.||_1$). If the function g is sufficiently smooth (bounded and existing derivatives), then consistency is satisfied. One can for example verify consistency by writing out the error terms of the Taylor approximations of the finite difference schemes.

Convergence en stability are less straightforward to show. We now first consider stability. We use the following definition for stability. A finite difference scheme is said to be stable with respect to the norm $||.||$ if there exist positive constants $\Delta_1, \Delta_2, \Delta_t$ and a non-negative constant K such that:

$$||g^{\delta}_0|| \leq K||g^{\delta}_T||, \quad \text{for } 0 < \delta_1 \leq \Delta_1,\ 0 < \delta_2 \leq \Delta_2,\ 0 < \delta_t \leq \Delta_t. \tag{71}$$

The stability condition implies that when you let the time steps go from $\Delta_1, \Delta_2, \Delta_t$ towards zero, the norm of the solution remains bounded. It turns out that when considering a consistent linear scheme of the form $g^{\delta}_{t_i} = Ag^{\delta}_{t_{i+1}}$, the scheme is convergent if and only if it is stable. This is the so called Lax Equivalence theorem. So the only thing that we have to verify is the stability of the scheme.

Let us first consider a simplified example of the PDE that we are trying to solve. Assume that $R = -r$ where $r > 0$ is the interest rate. Furthermore, assume that $\mu_X = [rX_{1,t} \ \ rX_{2,t}]$ and $\sigma_X = \begin{bmatrix} \sigma_1 X_{1,t} & 0 \\ 0 & \sigma_2 X_{2,t} \end{bmatrix}$, so there is no optimal control. Then the PDE becomes:

$$0 = \frac{\partial g_t}{\partial t} + rX_{1,t}\frac{\partial g_t}{\partial X_{1,t}} + rX_{2,t}\frac{\partial g_t}{\partial X_{2,t}} + \frac{1}{2}(\sigma_1 X_{1,t})^2 \frac{\partial^2 g_t}{(\partial X_{1,t})^2} + \frac{1}{2}(\sigma_1 X_{2,t})^2 \frac{\partial^2 g_t}{(\partial X_{2,t})^2} - rg_t. \tag{72}$$

This is the two-dimensional Black-Scholes PDE when the assets are uncorrelated. Note that the drifts are always positive, so the upwind scheme implies that only forward differences are used. Since the drift is always positive in this case, we assume that the first derivative vanishes at the boundary. Every European style two-asset option satisfies this equation. We can again write the scheme as $g^{\delta}_{t_i} = Ag^{\delta}_{t_{i+1}}$ where A is independent of time since μ_X, σ_X and R are independent of time.

Due to the upwind scheme, the matrix D^{δ} satisfies the following properties:

$$\begin{aligned} (D^{\delta})_{i,i} < 0, \quad (D^{\delta})_{i,j} \geq 0 \ j \neq i, \\ \sum_j (D^{\delta})_{i,j} = 0 \quad \forall i \end{aligned} \tag{73}$$

.

Then it can be shown that $||A^I||_1 \leq \frac{1}{1+r\delta_t}$ (see Lapeyre et al. (2005), Theorem 7.2.3). Since we can write $||g^{\delta}_0||_1 = ||(A^I)^{N_t} g^{\delta}_T||_1 \leq \left(||A^I||_1\right)^{N_t} ||g^{\delta}_T||_1 \leq ||g^{\delta}_T||_1$,

the implicit scheme is stable for any time step δ_t. The $||.||_1$ for a matrix corresponds to the maximum absolute row sum of a matrix. Note that the implicit scheme in this case is fully implicit, since R does not depend on g_t.

Similarly, $||A^E||_1 \leq \frac{1}{1+r\delta_t}$ when the additional condition $\delta_t \leq \frac{1}{j(D^\delta)_{i,i}j} \quad \forall i$ is satisfied. So given a small enough time step, the explicit scheme is stable as well. However, this condition can be quite restrictive in some cases. It might lead to an unnecessary small time step which will make the algorithm slower compared to the implicit method which is stable for any time step.

But in the general case there is no guarantee that $R(X_t, g_t, u_t, t)$ is negative for any X_t. Now consider again a simplified problem where $R(X_t, u_t)$ does not depend on time or on g_t, but possibly on the states X_t and the control u_t and $R(X_t, u_t)$ can be both negative and positive. Furthermore, assume that μ_X and σ_X are independent of t such that the matrix A does not depend on time. Define the spectral radius as: $\rho(A) \equiv \max\{|\lambda| : \lambda \text{ is an eigenvalue of A}\}$, i.e. the maximum absolute eigenvalue. A necessary condition for stability is then that the spectral radius must be smaller or equal to one: $\rho(A) \leq 1$. The idea behind this condition is the following. We can write: $||g_0^\delta|| = ||A^{N_t} g_T^\delta||$. Since $||A^n|| \geq \rho(A)^n \quad \forall n$, $\rho(A) \leq 1$ is a necessary condition for stability. So if we want the norm of the matrix A to the power n to be bounded for large n, the spectral radius must be less or equal to one. This condition is also a sufficient condition when the matrix A is symmetric. When A is not symmetric (which is often the case), in practice this condition turns out to still be useful.

Consider now the general problem where A_t depends on time since μ_X, σ_X and $R(X_t, g_t, u_t, t)$ might depend on time. In this case we can write $||g_0^\delta|| = ||A_{t_1}...A_{t_N} g_T^\delta|| \leq ||A_{t_1}...A_{t_N}||\ ||g_T^\delta||$. We therefore must find a bound on the product of a sequence of matrices, which is not straightforward. We can use the following inequality: $||A_{t_1}...A_{t_N}|| \leq ||A_{t_1}||...||A_{t_N}||$. This implies that if can establish that the norm the matrix A_t is less than or equal to one for every t, the scheme is stable. Since $||A_t|| \geq \rho(A_t) \quad \forall t$, it is again useful to verify whether the spectral radius of A_t is bounded by one for each t. It might take some time to calculate the spectral radius of the matrix A_t, especially when A_t is very large. Therefore, one should only calculate the spectral radius when there are serious concerns about stability. It could be useful to calculate the spectral radius at least for the first time step (A_{t_N}) or every n-th iteration.

Lastly, we discuss oscillatory behavior. Even when the scheme is stable, there might be oscillations if the time step is too large. In our specific setting, the function $R(X_t, g_t, u_t, t)$ contains a term $g_t^{\frac{-1}{\zeta}}$. When g_t becomes negative, this term might lead to complex numbers which is unsatisfactory. The scheme is free of oscillatory behavior when all eigenvalues of the matrix A are non-negative (Harwood, 2017). Note that this is a sufficient condition and that the condition is not necessary.

Summarizing, it is not straightforward to derive precise stability conditions for the general problem. In the simpler setting, it is shown that the implicit upwind scheme is unconditionally stable. Although we cannot guarantee that this will also lead to a stable scheme in the more general setting, in practice the implicit upwind scheme is often stable. It can still be useful to verify, e.g. in the first step, whether

the eigenvalues of the matrix A are between 0 and 1 (up to numerical precision). If this is not the case, this indicates an unstable algorithm. It is expensive in terms of computation time to do this often but if there are concerns this gives at least some guidance in whether the scheme is stable or not.

D Finite difference without time derivative

The problem that we are solving becomes:

$$\begin{aligned} 0 &= \min_{u_t} \Big\{ R(X_t, g_t, u_t) g_t + Dg_t \Big\} \\ \text{where } \; Dg_t &= \sum_{d=1}^{d_X} (\mu_X)_d \frac{\partial g_t}{\partial X_{d,t}} + \frac{1}{2} \sum_{d=1}^{d_X} (\sigma_X \sigma_X^{\emptyset})_{d,d} \frac{\partial^2 g_t}{(\partial X_{d,t})^2}. \end{aligned} \tag{74}$$

Then our finite difference solution would solve the following equation:

$$0 = \left(D^{\delta}\big(u\ (g^{\delta})\big) + diag\Big(R^{\delta}\big(g^{\delta}, u\ (g^{\delta})\big)\Big) \right) g^{\delta}. \tag{75}$$

Note that the matrix D^{δ} and the vector R^{δ} are independent of time. One possibility is to solve this equation directly using a non-linear solver. Another way to solve this equation is to use an iterative approach and add a 'false transient' with step size Δ to (75). Define $D_n^{\delta} = D^{\delta}\big(u\ (g_n^{\delta})\big)$ and $R_n^{\delta} = R^{\delta}(g_n^{\delta}, u\ (g_n^{\delta})\Big)$. This gives the following equation:

$$\frac{g_{n+1}^{\delta} - g_n^{\delta}}{\Delta} = \Big(D_n^{\delta} + diag(R_n^{\delta}) \Big) g_{n+1}^{\delta}. \tag{76}$$

We can use a similar algorithm as for the problem with time derivative to solve for g^{δ}.

The algorithm

Step 1: Start with an initial guess for the g_0^{δ} .

Starting with $n = 0$, repeat the following steps until $|g_{n+1}^{\delta} - g_n^{\delta}| < crit$ where $crit$ is some small number.

Step 2: g_n^{δ} is obtained from the previous iteration. Calculate the optimal policy $u\ (g_n^{\delta})$ using either a closed form or implicit expression that follows from the first order conditions. This requires as input g_n^{δ} and its derivatives. The derivatives can easily be calculated using central finite differences.

Step 3: Use u_n to calculate μ_X and σ_X.

Step 4: Construct D_n^{δ} (see appendix B).

Step 5: Use g_n^{δ} and u_n to calculate R_n^{δ}.

Step 6: Given g_n^{δ}, D_n^{δ} and R_n^{δ}, we can use a semi-implicit scheme to obtain g_{n+1}^{δ}:

$$g_{n+1}^{\delta} = A_n^I g_n^{\delta} = \left(I_{N_1 N_2} - \Delta \Big(D_n^{\delta} + diag(R_n^{\delta}) \Big) \right)^{1} g_n^{\delta}.$$

Optional: Step 7. To obtain the risk-free rate and the risk premium, ξ and its derivatives can be calculated using (central) finite differences. Result: After convergence,

we obtain the function g^δ and the optimal policy u (g^δ).

The idea of the 'false transient' is to use an iterative method to solve (75). Δ in this case can be seen as a stability parameter in contrast to an actual time step. However, the algorithm is basically identical to the algorithm for the problem with time derivative. A small Δ restricts the updated g^δ_{n+1} to be close to the previous step g^δ_n. This leads to a stable scheme, but it might take a long time before the algorithm converges. A larger Δ will lead to faster convergence, but for a too large Δ the matrix A^I_n might not satisfy the stability conditions discussed in the previous section. In contrast to the problem with time derivative, the accuracy of the scheme does not depend on Δ. Since the accuracy does not depend on the time step, it is efficient to choose Δ quite large but making sure that the stability conditions are met. The semi-implicit scheme is stable for much larger Δ compared to the explicit scheme and therefore we do not consider the explicit scheme in this case.