ON NUMERICAL APPROXIMATION OF A SYSTEM OF HAMILTON–JACOBI–BELLMAN EQUATIONS ARISING IN INNOVATION DYNAMICS

ĽUBOMÍR BAŇAS, HERBERT DAWID, TSIRY A. RANDRIANASOLO, JOHANNES STORN, AND XINGANG WEN

Abstract. We consider a system of fully nonlinear partial differential equations (PDEs) which corresponds to the Hamilton–Jacobi–Bellman equations for the value functions of an optimal innovation investment problem of a monopoly firm. We study three different approaches for the numerical approximation of the considered problem: the collocation method, the finite difference method and the finite element method. We discuss implementation issues for the three considered schemes and perform numerical studies for different model parameters to assess their performance.

1. Introduction

We study the system of Hamilton–Jacobi–Bellman (HJB) equations

$$rV_1(x) = \sup_I \mathcal{L}_1(V_1, V_2, x, I) + f_1(x) \qquad x \in \mathbb{R}, \tag{1.1}$$

$$rV_2(x,y) = \mathcal{L}_2(V_2, x, y) + f_2(x) \qquad x \in \mathbb{R},\ y \geqslant 0, \tag{1.2}$$

where $r > 0$ is a constant (so-called discount rate) and the operators $\mathcal{L}_1$, $\mathcal{L}_2$ are defined as

$$\mathcal{L}_1(V_1, V_2, x, I) := b_1(x,I)\partial_x V_1(x) - p_0(x)V_1(x) + p_2(I)(V_2(x,0) - V_1(x)), \tag{1.3}$$

$$\begin{aligned}\mathcal{L}_2(V_2, x, y) :=& \tfrac{1}{2}\sigma^2(y)\partial^2_{yy}V_2(x,y) + b_2(x,y)\partial_x V_2(x,y) + \mu(y)\partial_y V_2(x,y)\\ &- p_0(x)V_2(x,y)\end{aligned} \tag{1.4}$$

The terms f_1 and f_2 in (1.1)–(1.2) and b_1, b_2, p_0, p_2, σ, and μ in (1.3)–(1.4) a real-valued continuous functions specified in Table 2.2 below.

The HJB system (1.1)–(1.2), which can be classified as a fully-nonlinear degenerate second order partial differential equation in non-divergence form, was derived in [10] to study optimal innovation investment strategies of a monopoly firm under financial constraints. The solutions V_1 and V_2 to (1.1)–(1.2) correspond to the value functions of different investment modes of the monopoly firm considered on an infinite time horizon and the control variable I in (1.1) is the innovation investment (i.e. the amount of innovation activity the firm performs). Given the liquidity $x \equiv x(t)$ and the innovation investment $I \equiv I(t)$ at time $t \geqslant 0$, the monopoly firm may enter three different modes of production $m = 0, 1, 2$ with certain probability rates which are characterized by a Markov process $m := \{m_t \in \{0,1,2\} : t \geqslant 0\}$ with $m_0 = 1$ and for every $t \geqslant 0$

$$\mathbb{P}[m_{t+\mathrm{d}t} = j | m_t = i] = \begin{cases} p_0(x(t))\mathrm{d}t, & (i,j) = (1,0) \text{ or } (i,j) = (2,0),\\ p_2(I(t))\mathrm{d}t, & (i,j) = (1,2),\\ 0, & \text{for any other } (i,j),\ i \neq j, \end{cases} \tag{1.5}$$

where $p_0 \geqslant 0$ is the bankruptcy rate and $p_2 \geqslant 0$ is the innovation rate. For every state $i \in \{0,1,2\}$, it holds that $\sum_{j=0}^{2} \mathbb{P}[m_{t+\mathrm{d}t} = j | m_t = i] = 1$.

(Ľ. Baňas, T.A. Randrianasolo, J. Storn) Faculty of mathematics, Bielefeld university, Universitätsstrasse 25, DE-33615 Bielefeld

(H. Dawid, X. Wen) Faculty of Business Administration and Economics, Universitätsstrasse 25, DE-33615 Bielefeld

Date: August 28, 2020.

This research was supported by the German Research Foundation as part of the Collaborative Research Center SFB 1283.

The possible scenarios given by the above Markov process can be summarized as follows. The mode $m = 1$ corresponds to the pre-innovation mode where the firm is selling products on the old market and is developing a new product with a suitable amount of innovation investment I which allows it to leave the old market ($m = 1$) and enter the new market ($m = 2$) with a probability rate $p_2 \equiv p_2(I)$. Finally, depending on its liquidity the firm may go bankrupt with a probability rate $p_0 \equiv p_0(x)$ and exit the market, that is the firm leaves the modes $m = 1$ or $m = 2$ and enters the mode $m = 0$, where value is constant, that is $V_0 = 0$. The objective of the firm is to maximize the accumulated dividend f_m, $m = 1, 2$ over time, which yields the system (1.1)–(1.2) for the modes $m = 1$, $m = 2$, respectively. For details on the derivation of the HJB equations (1.1)–(1.2) we refer to [10].

The solutions of equations (1.1)–(1.2) do in general not exist in the classical sense but can be defined in the sense of viscosity solutions, see, for instance [8, 14, 22, 25]. The existence of the unique viscosity solution a general HJB equation written in the form

$$F(\mathrm{D}^2 V, \nabla V, V, x) = 0 \qquad x \in \mathbb{R}^d, \tag{1.6}$$

is guaranteed provided that the functional F is continuous in all variables and satisfies a monotonicity condition, that is, it holds for $A, A' \in \mathbb{R}^{d\times d}$, $r, s \in \mathbb{R}^d$, $x \in \mathbb{R}^d$, $d \geqslant 1$ that

$$F(A, p, q, x) \leqslant F(A', p, q', x), \tag{1.7}$$

whenever $A - A'$ is negative definite and $q \geqslant q'$, cf. [18].

Equation (1.2) corresponds to (1.6) with with $d = 2$ and F given by

$$F(A, p, q, x, y) = \tfrac{1}{2} \begin{pmatrix} 0 & 0 \\ 0 & \sigma^2(y) \end{pmatrix} A + (b_2(x, y), \mu(y))\cdot p - (p_0(x) + r)q + f_2(x)\,,$$

which is monotone since $\sigma^2 \geqslant 0$ and $(p_0 + r) > 0$. The continuity of F is a consequence of the continuity of the functions σ, μ, and f_m, b_m, $m = 1, 2$. Hence, we deduce that (1.2) admits a unique viscosity solution V_2.

Equation (1.1) corresponds to (1.6) with $d = 1$ and F given by

$$F(0, p, q, x) = \sup_I \Big\{ b_1(x, I)p - (p_0(x) + p_2(I) + r)q + p_2(I)V_2(x, 0) \Big\} + f_1(x)\,,$$

which is monotone because $r > 0$ and $p_0, p_2 \geqslant 0$. Nevertheless, the continuity of F is not obvious since the control I may become discontinuous, as indicated by the numerical experiments, see Figure 4.3.

From the theoretical as well as computational point of view, the condition (1.7) is crucial since it guaranties the uniqueness of the viscosity solution and the convergence (in the viscosity sense) of numerical schemes [1], in addition the conditions ensures the discrete maximum principle holds for the numerical solution. There exist a number of approaches how to preserve the monotonicity property (1.7) in the numerical approximation, for a recent overview we refer to [21]. Monotone finite difference schemes can be obtain by an appropriate choice of the finite difference stencil, for instance by upwind finite difference schemes [24, 26] or by adopting a semi-Lagrangian approach [7, 11, 12]. Much fewer results are available on monotone finite element (FEM) based discretization methods [17, 15, 23, 21], where in particular [17] is the first to show convergence of FEM discretizations of HJB equations of the type (1.1). We also mention the collocation method which is popular choice in applications [10, 27] because its capability to solve HJB equations in higher dimensions, cf. [6]. So far, very few results exist on adaptive numerical approximation of HJB equations. We mention the work [27] which employs and adaptive collocation approach and [19] which applies the least-squares finite element method (LSFEM) to a linear second order problem that satisfies the so-called Cordes condition. We note that the Cordes conditions is not satisfied for the equations in (1.1)–(1.2). In this work we apply an adaptive LSFEM method to solve the first order nonlinear HJB equation (1.1).

We consider the discretization of (1.1) and (1.2) by three numerical methods: the Chebyshev collocation method, the finite difference method, and the (adaptive) finite element method. We note that the equation (1.2) can be solved independently of (1.1). Hence we first determine the numerical approximation of the value function V_2 in mode $m = 2$. This numerical approximation

is then used to compute the approximation of the value function V_1 in mode $m = 1$. The discrete nonlinear system which results from the discretization of (1.1) is solved using a iterative procedure, so-called policy iteration [2, 16]. Along with several other implementation details, we discuss, in particular, relevant modifications of the policy iteration which are required to ensure the convergence of the iterations for each of the considered numerical approaches.

The paper is organized as follows. In Section 2 we give details on the parameters and functions in (1.1) and (1.2), and discuss analytical properties which are useful to deduce boundary conditions that are required in the considered numerical approximations. In Sections 3–5 we give details on the respective numerical schemes. We summarize the advantage and disadvantage of the three numerical approaches in Section 6.

2. Preliminaries

2.1. **Setting of the parameters.** We fix the constants, $\nu_1, \nu_2, \widetilde{\alpha}_n, \bar{\alpha}_n, \eta, \delta, \sigma, r, \gamma_1, \xi$ according to Table 2.1. Then, we base our numerical experiment upon three scenarios: scenario 1 with ($\alpha_o = 0.8$, $\gamma_2 = 0.05$), scenario 2 with ($\alpha_o = 0.8$, $\gamma_2 = 0.005$), and scenario 3 with ($\alpha_o = 1.0$, $\gamma_2 = 0.05$), where α_o is the demand for the old market and γ_2 is the bankruptcy parameter. The functions which appear in (1.3) and (1.4) are defined in Table 2.2.

Table 2.1. Parameters of the monopoly model.

Parm.	Value	Description	Parm.	Value	Description
ν_2	0.2	Post-innovation dividend rate	δ	1.55	Adjustment speed of Y_s
ν_1	0.0	Pre-innovation dividend rate	σ	0.1	Volatility of Y_s
$\widetilde{\alpha}_n$	0.8	Demand for the new market	r	0.02	Discount rate
$\bar{\alpha}_n$	0.6		γ_1	0.1	Innovation parameter
η	0.5	Horiz. diff. old/new	ξ	0.025	Investment cost parameter

Table 2.2. Coefficients of the HJBs.

Func	Expression	Description
$p_0(x)$	$\gamma_2 \max\{0, -x\}$	Bankruptcy rate
$p_2(I)$	$\gamma_1 I$	Innovation rate
$f_m(x)$	$\nu_m \max\{0, x\}$	Pre/post-innovation dividends
π_1	$\frac{1}{4}\alpha_o^2$	Pre-innovation profit
$\pi_2(y)$	$\dfrac{(\bar{\alpha}_n + y)^2 + \alpha_o^2 - 2\eta(\bar{\alpha}_n + y)\alpha_o}{4 - 4\eta^2}$	Post-innovation profit
$b_1(x, I)$	$\pi_1 - \frac{1}{2}\xi I^2 + rx - f_1(x)$	Pre-innovation liquidity trend
$b_2(x, y)$	$\pi_2(y) + rx - f_2(x)$	Post-innovation liquidity trend
$\mu(y)$	$\delta(\widetilde{\alpha}_n - y)$	Demand trend for the new market
$\sigma(y)$	σy	Volatility of the new market

2.2. **Boundary conditions.** All numerical schemes considered in this paper can only be applied on bounded domains. Hence, we restrict the computations on a sufficiently large finite domains and deduce suitable boundary conditions from the considerations below.

The value function satisfies $V_1(x) \to 0$ as $x \to -\infty$. Moreover, for sufficiently large x, the solution can be characterized by the following result from [10, Proposition 1].

Proposition 2.1. *Assume that $r > \nu_1$. For every $x, y \geqslant 0$ the value in mode $m = 2$ is given by*

$$V_2(x, y) = x + c + \frac{\delta\widetilde{\alpha}_n + (\bar{\alpha}_n - \alpha_o\eta)}{2(r + \delta)(1 - \eta^2)}y + \frac{1}{4(1 - \eta^2)(r + 2\delta - \sigma^2)}y^2, \tag{2.1}$$

with $c := \dfrac{\delta^2\widetilde{\alpha}_n^2 + \delta\widetilde{\alpha}_n(\bar{\alpha}_n - \alpha_o\eta)}{2r(r+\delta)(1-\eta^2)} + \dfrac{\bar{\alpha}_n^2 + \alpha_o^2 - 2\eta\alpha_o\bar{\alpha}_n}{4r(1-\eta^2)}$.

Furthermore, it holds for every $x \geqslant \tilde{e} := \max\left\{\dfrac{2\xi|I^{\mathrm{nc}}|^2 - \alpha_o^2}{4(r-\nu_1)}, 0\right\}$ *with* $I^{\mathrm{nc}} := \sqrt{\frac{r^2}{\gamma_1^2} + \frac{2rc}{\xi} - \frac{\alpha_o^2}{2\xi}} - \frac{r}{\gamma_1}$

(i) The optimal control is constant over time, that is $I = I^{\mathrm{nc}}$*;*

(ii) The value in mode $m = 1$ *is given by*

$$V_1(x) = x + c + \frac{\xi}{\gamma_1} I^{\mathrm{nc}}. \tag{2.2}$$

2.3. **Rescaling transformation.** A modification of the differential equations on an infinite domain (1.1)–(1.2) results in problems on finite domains. This allows for the application of numerical schemes like the collocation method in Section 3 and the finite differences method in Section 4. In the manner of [10] we introduce the function

$$z : \mathbb{R} \to (0,1), \quad x \mapsto z(x) = \frac{1}{1+\exp(-x/2)}, \tag{2.3}$$

which scales $\mathbb{R}$ into $(0,1)$. The transformation (2.3) enables the application of the numerical methods on the finite domain $(0,1)$ where grid points can be constructed and boundary conditions can be assigned at 0 or 1 rather than $-\infty$ and $+\infty$. The inverse transformation of (2.3) reads

$$x : (0,1) \to \mathbb{R}, \quad z \mapsto x(z) = 2\ln\left(\frac{z}{1-z}\right). \tag{2.4}$$

We set $U(z) := V_1(x(z))$ and $V(z,y) := V_2(x(z),y)$. The inverse transformation (2.4) implies that $\partial_x V_1 = \frac{1}{2}z(1-z)\partial_z U$ and as a result the equations in (1.1)–(1.2) are rescaled into

$$rU(z) = \sup_I L_1\big(U, V, z, I\big) + f_1(x(z)), \quad z \in (0,1); \tag{2.5}$$

$$rV(z,y) = L_2\big(V, z, y\big) + f_2(x(z)), \qquad z \in (0,1),\ y \geqslant 0; \tag{2.6}$$

where

$$\begin{aligned} L_1(U,V,x,I) :=& \tfrac{1}{2}z(1-z)b_1(x(z),I)\partial_x U(z) - p_0(x(z))U(x(z)) \\ &+ p_2(I)(V(x(z),0) - U(x(z))), \end{aligned} \tag{2.7}$$

$$\begin{aligned} L_2(V,z,y) :=& \tfrac{1}{2}\sigma^2(y)\partial_{yy}^2 V_2(x,y) + \tfrac{1}{2}z(1-z)b_2(x(z),y)\partial_x V(x(z),y) + \mu(y)\partial_y V(x(z),y) \\ &- p_0(x(z))V(x(z),y), \end{aligned} \tag{2.8}$$

with the boundary conditions $U(0) = 0$ and $U(z) := V_1(x(z))$ with $x(z) \geqslant \tilde{e}$ is given by (2.2).

3. Discretization by collocation method

In this section we describe the Chebyshev collocation method to compute an approximate solution for (1.1) and (1.2). The method seeks the approximation $V_1(x)$ and $V_2(x,y)$ in the form of finite series of Chebyshev polynomials with unknown coefficients. The finite series are then substituted into (1.1) and (1.2) and the coefficients are determined by requiring that the differential equations are satisfied at some finite number of collocations nodes. The collocation method can only be applied on a finite domain and thus, for convenience, the scaling functions (2.3) and (2.4) are used.

3.1. **Discretization of the value in post-innovation mode,** $m = 2$**.** In a given state space $[\underline{z}, \bar{z}] \times [\underline{y}, \bar{y}]$, we first construct a sparse grid of collocation nodes $\mathcal{N} := \mathcal{N}_z \times \mathcal{N}_y$, where $\mathcal{N}_z := \{z^i\}_{i=1,\dots,n_z}$ and $\mathcal{N}_y := \{y^j\}_{j=1,\dots,n_y}$, z^i and y^j are defined as

$$z^i = \frac{\bar{z}+\underline{z}}{2} + \frac{\bar{z}-\underline{z}}{2}\cos\left(\frac{(n_z - i + 0.5)\,\pi}{n_z}\right), \tag{3.1}$$

$$y^j = \frac{\bar{y}+\underline{y}}{2} + \frac{\bar{y}-\underline{y}}{2}\cos\left(\frac{(n_y - j + 0.5)\,\pi}{n_y}\right). \tag{3.2}$$

We construct a set of basis functions $\{\phi_{i,j}(z,y)\}_{\{i=1,\dots,n_z\}\times\{j=1,\dots,n_y\}}$ corresponding to our Chebyshev sparse grid such that

$$\phi_{i,j}(z,y) = T_{i-1}\left(\frac{2\,(z-\underline{z})}{\bar{z}-\underline{z}} - 1\right) T_{j-1}\left(\frac{2\,(y-\underline{y})}{\bar{y}-\underline{y}} - 1\right), \tag{3.3}$$

where $T_k : t \mapsto T_k(t)$ is the Chebyshev polynomial of degree k defined for any $t \in [0,1]$. For the given state space of $(0, 0.5] \times [0, \bar{y}]$, our calculation is carried out in the space of $[\underline{z}, 0.5] \times [0, \bar{y}]$: $\underline{y} = 0$ represents that $y = 0$ at the moment the mode switch from $m = 1$ to $m = 2$, and $\bar{z} = 0.5$ corresponds to an upper boundary of $x = 0$. In order to make sure the calculated value function is continuous at $x = 0$, we specify further that

$$z^i = \begin{cases} \dfrac{0.5+\underline{z}}{2} + \dfrac{0.5-\underline{z}}{2}\cos\left(\dfrac{(n_z - i + 0.5)\,\pi}{n_z}\right) & 1 \leqslant i \leqslant n_z - 1, \\ 0.5 & i = n_z. \end{cases} \tag{3.4}$$

The expansion of V_2 as a Chebyshev series is given by

$$\widehat{V}_2(z,y) := \sum_{i=1}^{n_z}\sum_{j=1}^{n_y} v_{i,j}\phi_{i,j}(z,y) = v\phi(z,y),\ z \in (0, 0.5], \tag{3.5}$$

where $\bar{v} := \{\bar{v}_k : k = 1,\dots,n_z n_y\}_k$ and $\phi := \{\phi_k : k = 1,\dots,n_z n_y\}$ such that $\bar{v}_k := v_{i,j}$ and $\phi_k := \phi_{i,j}(z,y)$ with $k = (i-1)n_z + j$ for $i \in \{1,\dots,n_z\}$ and $j \in \{1,\dots,n_y\}$. The solution $V(z,y)$ of (2.6) is substituted by the expression (3.5). Thus the approximation of the value function V is equivalent to determine the components $\bar{v}_k$ of the vector $\bar{v}$ such that (2.6) holds on the collocation nodes $\{z^i, y^j\}$ with $i,j \in \{1,\dots,n_z\} \times \{1,\dots,n_y - 1\}$. The approximation of V_2 is therefore defined by

$$\bar{V}_2(z^i, y^j) := \begin{cases} \widehat{V}_2(z^i, y^j), & z^i \in (0, 0.5), \\ V_2(x(z^i), y^j), & z^i \in [0.5, 1], \end{cases} \tag{3.6}$$

where $x : z \mapsto x(z)$ is given by (2.4) and $V_2 : (x,y) \mapsto V_2(x,y)$ is given by the formula (2.1). Note that we impose $\widehat{V}_2(0.5, y^j) = V_2(x(0.5), y^j)$ to enforce the continuity of (3.6) at $z = 0.5$.

In total there are $n_z n_y$ number of nodes, implying $n_z n_y$ number of equations. Furthermore, for $i \in \{1,\dots,n_z\}$ and $j \in \{1,\dots,n_y - 1\}$ we introduce four $n_z(n_y-1) \times n_z n_y$ matrices $\mathbf{B}$, $\mathbf{B}^y$, $\mathbf{B}^{yy}$, and $\mathbf{B}^z$ with entries

$$\mathbf{B}_{s,k} = \phi_k(z^i, y^j),\ \mathbf{B}^y_{s,k} = \partial_y\phi_k(z^i, y^j),\ \mathbf{B}^{yy}_{s,k} = \partial^2_{yy}\phi_k(z^i, y^j),\ \mathbf{B}^z_{s,k} = \partial_z\phi_k(z^i, y^j), \tag{3.7}$$

where $s = (i-1)n_z + j$ denotes node s. These four matrices capture the values of all base functions and their partial derivatives at the nodes in $\mathcal{N}$ that are not on the boundary of $z^{n_z} = 0.5$. For each node $\{z^i, y^j\}$ with $i \in \{1,\dots,n_z - 1\}$ and $j \in \{1,\dots,n_y\}$, we define the five column vectors $\bar{\sigma} := (\bar{\sigma}_s)_s$, $\bar{\mu} := (\bar{\mu}_s)_s$, $\bar{b}_2 := (\bar{b}_{2,s})_s$, $\bar{p}_0 := (\bar{p}_{0,s})_s$, and $\bar{f}_2 := (\bar{f}_{2,s})_s$ as follows:

$$\bar{\mu}_s = \mu(y^j),\ \bar{\sigma}_s = \sigma(y^j),\ \bar{b}_{2,s} = \tfrac{1}{2}z^i(1-z^i)b_2(x(z^i), y^j),\ \bar{f}_{2,s} = f_2(x(z^i)),\ \bar{p}_{0,s} = p_0(x(z^i)),$$

and $s = (i-1)n_z + j \in \{1,\dots,(n_z-1)n_y\}$. Thus the nodes s in $\mathcal{N}$, $\bar{v}$ should enforce the identity

$$r\mathbf{B}\bar{v} = \bar{\sigma}^\top\mathbf{B}^{yy}\bar{v} + \bar{\mu}^\top\mathbf{B}^y\bar{v} + \bar{b}_2^\top\mathbf{B}^z\bar{v} - \bar{p}_0^\top\mathbf{B}\bar{v} + \bar{f}_2. \tag{3.8}$$

For the other n_y nodes, it holds that

$$\bar{v}^\top\phi(0.5, y^j) = \bar{V}_2(0, y^j),\ j \in \{1,\dots,n_y\}. \tag{3.9}$$

3.2. **Discretization of the value in pre-innovation mode, $m = 1$.** In contrast to mode $m = 2$, the value function V_1 in mode $m = 1$ is only defined on the domain of $\mathbb{R}$. The control $I = \arg\max_{\widetilde{I}} \mathcal{L}_1(V_1, V_2, x, \widetilde{I})$ is captured by

$$I(x) = \frac{\gamma_1(V_2(x,0) - V_1(x))}{\xi\partial_x V_1(x)}. \tag{3.10}$$

Similarly as for the mode $m = 2$, we employ the transformation from the state space of x to the state space of $z \in (0, 1)$ according to $x(z)$. After the transformation (2.4), the optimal control in (3.10) can be rewritten as

$$I(x(z)) = \frac{2\gamma_1(V_2(x(z), 0) - V_1(x(z)))}{\xi z(1 - z)\partial_z V_1(x(z))}. \tag{3.11}$$

We proceed with the same collocation method as in mode $m = 2$, but on the one dimensional state space of $z \in (0, 1)$. However, there exists a kink for $V_1(x(z))$ at $z = 0.5$ and a jump in the control function $I(x(z))$ because of different HJB expressions and the difference in $\partial_x V_1(x(z))$ for positive and negative x. Thus, we define the approximation of V_1 as follow

$$\bar{V}_1(z^i) := \begin{cases} \widehat{V}_1(z^i), & z^i \in (0, 0.5), \\ \check{V}_1(z^i), & z^i \in [0.5, 1), \end{cases} \tag{3.12}$$

where $x : z \mapsto x(z)$ is given by (2.4). Note that we impose $\widehat{V}_1(0.5) = \check{V}_1(0.5)$ to enforce the continuity of (3.12) at $z = 0.5$. Moreover, with the collocation method, it is of crucial importance to distinguish whether $\lim_{t\to\infty} X_t^{0,x,I} < 0$ given that $x = 0^+$. From an economic perspective,

- if $\lim_{t\to\infty} X_t^{0,x,I} < 0$, the option control at $t = 0$ already takes into account of the future negative liquidity, implying it is more reasonable to calculate $\widehat{V}_1$ first;
- if $\lim_{t\to\infty} X_t^{0,x,I} \geqslant 0$ for a given $x = 0^+$, then for any positive initial state x, $X_t^{0,x,I}$ never converges to a negative value as $t \to \infty$, and we can calculate for $\check{V}_1$ first.

In [10], it has been shown that whether a slightly positive initial liquidity converges to 0 or a negative value in the long run $t \to \infty$ depends on the parameter values.

The following technical explanation illustrates the situation where $\check{V}_1$ is calculated first. The situation to calculate $\widehat{V}_1$ first is analogous. To ensure that the estimated value function would be continuous at $x = 0$, that is $z = 0.5$, we further specify that

$$z^i = \begin{cases} 0.5 & i = 1, \\ \dfrac{\bar{z} + 0.5}{2} + \dfrac{\bar{z} - 0.5}{2} \cos\left(\dfrac{(n_z - i + 0.5)\,\pi}{n_z}\right) & 1 < i \leqslant n_z. \end{cases}$$

The expansion of V_1 as a Chebyshev series is given by

$$\check{V}_1(z) := \sum_{i=1}^{n_z} u_i \phi_{i,1}(z, 0) = \bar{u}^\top \phi(z, 0), \ z \in (0.5, 1), \tag{3.13}$$

where $\bar{u} := \{u_i : i = 1, \ldots, n_z\}$ is a column vector. The solution $U(z)$ of (2.5) is substituted by the expression (3.13). Thus the approximation of the value function U is equivalent to determine the components u_i of the vector $\bar{u}$ such that (2.5) holds on the collocation nodes $\{z^i : i = 2, \ldots, n_z\}$. We still need to specify an equation for the node $z^1 = 0.5$ corresponding to $x = 0$. When the liquidity $x = 0$ is a steady state, it can be calculated from $b_1(0, I) = 0$ that the optimal control is $I(0) = \alpha_o/\sqrt{2\xi}$, and from (3.11) it follows that

$$\check{V}_1(0.5) = \frac{\gamma_1 \alpha_o \bar{V}_2(0, 0)}{r\sqrt{2\xi} + \gamma_1 \alpha_o}. \tag{3.14}$$

We use an iterative algorithm to calculate the value function. In particular, we consider a sequence of vectors $\bar{u}^{(n-1)}$, with $n \in \mathbb{N}^*$ as the indicator for the iteration. In the stage $n \in \mathbb{N}^*$, we calculate the discrete optimal control $\bar{I}^{(n-1)} := \{I_i^{(n-1)} : i = 1, \ldots, n_z\}$ by

$$\bar{I}^{(n-1)} = \frac{\gamma_1\left(\mathbf{B}\bar{v}_{\cdot,1} - \mathbf{B}\bar{u}^{(n-1)}\right)}{\xi \bar{\lambda}^\top \mathbf{B}^z \bar{u}^{(n-1)}}, \tag{3.15}$$

where the column vectors $\bar{v}_{\cdot,1} := \{v_{i,1} : i = 1, \ldots, n_z\}$ and $\bar{\lambda} := \{\frac{1}{2} z^i(1 - z^i) : z^i \in \mathcal{N}_z, i = 1, \ldots, n_z\}$ and the matrices $\mathbf{B}$ and $\mathbf{B}^z$ are such that

$$\mathbf{B}_{i,k} = \phi_k(z^i, 0), \ \mathbf{B}^z_{i,k} = \partial_z \phi_k(z^i, 0), \ i \in \{1, \ldots, n_z\}.$$

Overall, the numerical details can be summarized as follows:

Algorithm 3.1 (Collocation method).
We start with an initial guess for $\bar{u}^{(0)}$ *and the control* $\bar{I}^{(0)}$. *Then, for every* $n \in \mathbb{N}^*$, *we proceed along the following two steps iteration:*

Policy evaluation: *Find* $\bar{u}^{(n)} = \{u_i^{(n)} : i = 1, \dots, n_z\}$ *such that*

$$\begin{cases} r\mathbf{B}\bar{u}^{(n)} = \bar{b}_1^\top(I^{(n-1)})\mathbf{B}^z\bar{u}^{(n)} - (\bar{p}_0 + \bar{p}_2(I^{(n-1)}))^\top\mathbf{B}\bar{u}^{(n)} + \bar{p}_2^\top(I^{(n-1)})\mathbf{B}\bar{v}_{\cdot,1} + \bar{f}_1, \\ \bar{u}_1^{(n)} = \dfrac{\gamma_1\alpha_o\bar{V}_2(0,0)}{r\sqrt{2\xi} + \gamma_1\alpha_o}. \end{cases} \tag{3.16}$$

Policy improvement: *Update the optimal control by (cf.* (3.15)*)*

$$\bar{I}^{(n)} = \frac{\gamma_1\big(\mathbf{B}\bar{v}_{\cdot,1} - \mathbf{B}\bar{u}^{(n)}\big)}{\xi\bar{\lambda}^\top\mathbf{B}^z\bar{u}^{(n)}}.$$

Stopping criterion: Repeat the loop until the ℓ^∞ *norm* $\left|\mathbf{B}\left(\bar{u}^{(n)} - \bar{u}^{(n-1)}\right)\right|_\infty \leqslant \epsilon$.

The numerical calculation of $\widehat{V}_1(z)$ with $z \in (0, 0.5]$ is similar to the calculation of $\breve{V}_1(z)$.

4. Discretization by finite differences

This section is devoted to the description of the finite difference approximation of (1.1). For the sake of brevity, we do not consider the finite difference approximation of V_2 since the results of the finite difference approximation are very similar to those obtained by the finite element method in Section 5.1 below.

It is more convenient to consider the discretization of the rescaled equations (2.5) since it allows us to apply the method on a finite domain. Furthermore, we observe that by Proposition 2.1, the solution V_1 of (1.1) is known explicitly for every $x \geqslant \tilde{e}$. Hence, it is enough to solve (2.5) and (2.6) in $(0, \tilde{z})$, where $\tilde{z} = 1/(1 + \exp(-\lambda\tilde{e}))$. We introduce an equidistant grid with mesh sizes $\Delta z := \tilde{z}/M$ where

$$z_i = i\Delta z \quad \text{for} \quad i = 0, 1, \dots, M\,. \tag{4.1}$$

We recall that the mesh grid has been rescaled by (2.3). The variable $x_i = x(z_i)$ are redistributed according to (2.4), see Fig. 4.1 where we display the distribution of the grid point x_i.

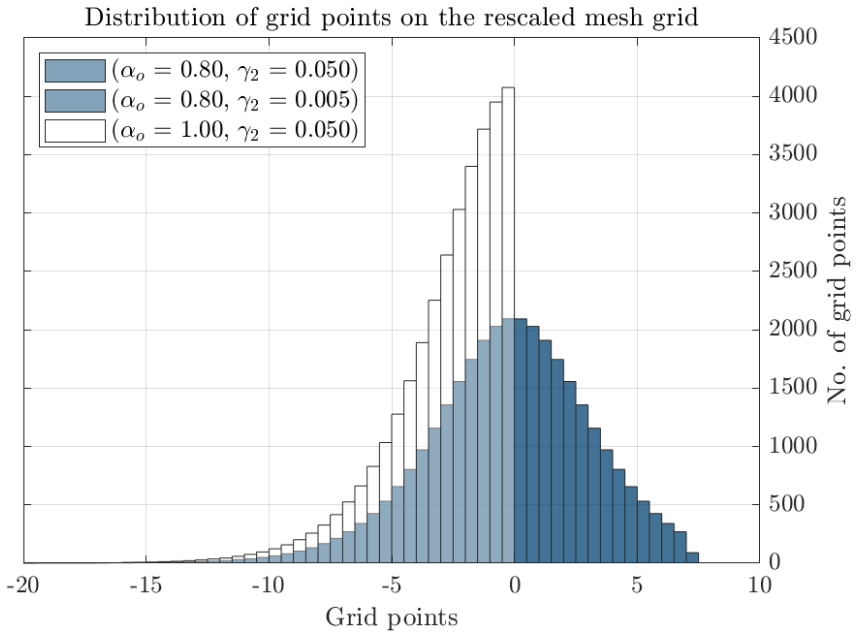


Figure 4.1. Distribution of the 2^{15} grid points on the rescaled domain.

We denote $b_1^i(I) := b_1(x(z_i), I)$, $p_0^i := p_0(x(z_i))$, $p_2 := p_2(I)$, $f_1^i := f_1(x(z_i))$, $\lambda^i := \frac{1}{2}z_i(1-z_i)$, and for $i = 0, \dots, M$ we denote by V_1^i the finite difference approximation of the value function V_1 on the grid (4.1), that is $V_1^i \approx V_1(x_i)$.

We approximate $\partial_z V_1$ by the backward, centered, or forward differences that are denoted by D_z^-, D_z^0, and D_z^+, respectively, and defined by

$$\mathrm{D}_z^- V_1^i := \frac{V_1^i - V_1^{i-1}}{\Delta z}, \quad \mathrm{D}_z^0 V_1^i := \frac{V_1^{i+1} - V_1^{i-1}}{2\Delta z}, \quad \mathrm{D}_z^+ V_1^i := \frac{V_1^{i+1} - V_1^i}{\Delta z}. \tag{4.2}$$

We also introduce the upwind finite difference approximation of $\partial_z V_1(x(z_i)) \approx \mathrm{D}_z^{\pm} V_1^i$ with

$$\mathrm{D}_z^{\pm} V_1^i := \begin{cases} \mathrm{D}_z^+ V_1^i, & \text{if } b_1^i(I) > 0, \\ \mathrm{D}_z^- V_1^i, & \text{if } b_1^i(I) \leqslant 0. \end{cases} \tag{4.3}$$

4.1. **Discretization of the value in mode $m = 1$.** We assume that the numerical approximation of the value function at $m = 2$ at $y = 0$ is known and denote its values on the finite difference grid (4.1) by $V_2^{i,0} \approx V_2(z_i, 0)$, $i = 0, \dots, M$ and set the vector $\overline{V}_2 := (V_2^0, \dots, V_2^M)$. The approximation of the control at the grid points is denoted by I^i, $i = 0, \dots, M$ and $\overline{I} := (I^0, \dots, I^M)$.

To obtain the finite difference approximation of V_1 we replace the derivatives in (2.5) by the corresponding difference approximations (4.3) which yields the following discrete system for all nodes $i = 1, \dots, M-1$

$$r V_1^i = \sup_{\overline{I}} L_1^M(\bar{V}_1, \bar{V}_2, z_i, \overline{I}) + f_1^i, \tag{4.4}$$

where $\bar{V}_1 := \{V_1^0, \dots, V_1^M\}$ and

$$L_1^M(\bar{V}_1, \bar{V}_2, z_i, \overline{I}) := \lambda_i b_1^i(I^i) \mathrm{D}_z^{\pm} V_1^i - p_0^i V_1^i + p_2(I^i)(V_2^{i,0} - V_1^i) \tag{4.5}$$

with V_1^0 and V_1^M given as boundary conditions, see Section 2.2. Similar to (3.10) or (3.11), the discrete optimal control I_h can be calculated explicitly. In the expression (3.11), we approximate $\partial_z V_1$ by the centered finite difference which gives

$$I^i = \frac{\gamma_1 \left(V_2^{i,0} - V_1^i\right)}{\xi \lambda^i \mathrm{D}_z^0 V_1^i}. \tag{4.6}$$

Analogically as in the case of the collocation method, we solve the nonlinear problem (4.4) with the *policy iteration* iterative procedure [2, 16]. Each iteration of the algorithm consists of a *policy improvement* step, where the control (also called policy or investment) $\bar{I}^{(n-1)}$ is improved by a previous value $\bar{V}_1^{(n)}$ of a *policy evaluation* step. Altogether, upwind scheme and policy iteration give the following algorithm:

Algorithm 4.1 (Finite difference)**.**
We start with an initial guess $\bar{I}^{(0)} := I^{\mathrm{nc}}$ for the control. Then, for every $n \in \mathbb{N}^*$, we proceed along the following two steps iteration:

Policy evaluation: *Find $\bar{V}_1^{(n)} := \left\{V_1^{i,(n)}, i = 1, \dots, M-1\right\}$, such that*

$$r V_1^{i,(n)} = L_1^M(\bar{V}_1^{(n)}, \bar{V}_2, z_i, \bar{I}^{(n-1)}) + f_1^i. \tag{4.7}$$

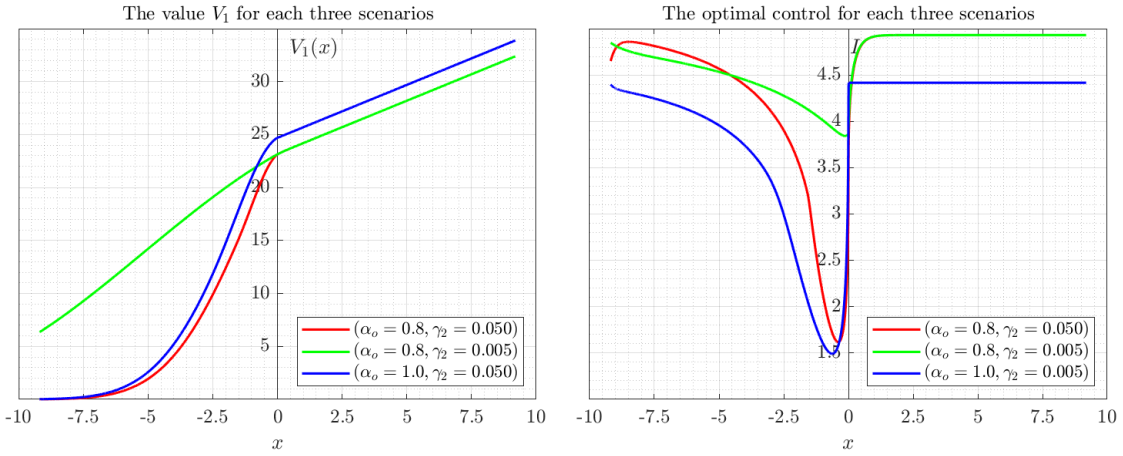
Policy improvement: *Update the control $\bar{I}^{(n)} := \left\{I^{i,(n)}, i = 1, \dots, M-1\right\}$ with*

$$I^{i,(n)} = \frac{\gamma_1 \left(V_2^{i,0} - V_1^{i,(n)}\right)}{\xi \lambda^i \mathrm{D}_z^0 V_1^{i,(n)}}. \tag{4.8}$$

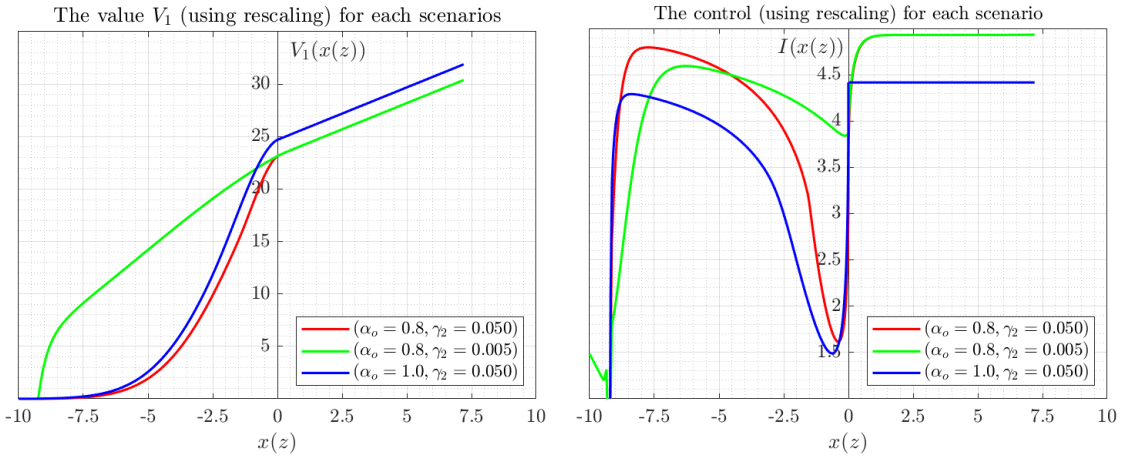
Stopping criterion: Repeat the loop until

$$\frac{\|\bar{I}^{(n)} - \bar{I}^{(n-1)}\|_\infty}{\|\bar{I}^{(n)}\|_\infty} \leqslant \epsilon.$$

4.2. **Numerical experiments.** In our numerical experiments we fix the tolerance of the stopping criteria to $\epsilon = 10^{-6}$. As it has been already discussed in Section 2.2, we have $V_1(x(z)) \to 0$ when $z \to 0$ while $V_1(x(\tilde{z}))$ is given by (2.2).

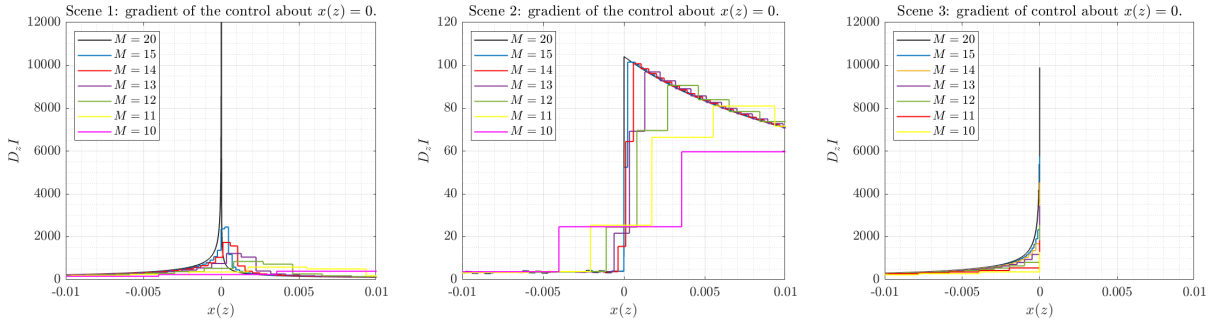


(a) V_1 computed before rescaling. (b) I computed before rescaling.



(c) V_1 computed after rescaling. (d) I computed after rescaling.

FIGURE 4.2. The value V_1 and the optimal control I computed with and without rescaling. Without rescaling, we used the solution computed by collocation method as the left boundary condition. The value V_2 was computed separately by collocation method with 10^4 grid points.



(a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

FIGURE 4.3. Gradient of the control close to $x(z) = 0$ and for each scenario. The number of grid points is $2^M, M = 20, 15, 14, 13, 12, 11, 10$.

4.2.1. *Effect of the domain and the bondary conditions.* We do not see any notable difference between the two solutions computed with and without rescaling as long as x and $x(z)$ stay far away from the left border. This is expected since in Figs. 4.2a–4.2b we used the solution

TABLE 4.3. Numerical results.

(a) Scenario 1 ($\alpha_o = 0.8$, $\gamma_2 = 0.050$) rescaled.

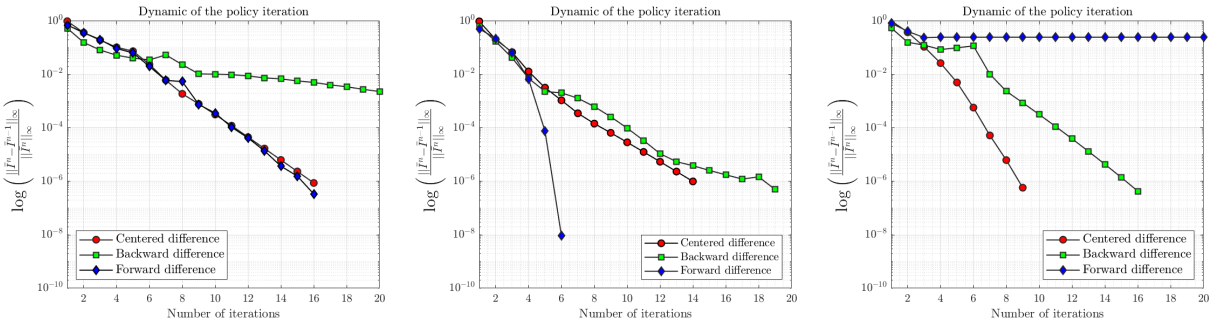
No. of grid points	Mesh size Δz	No. of iter.	ℓ^∞ error in V_1	ℓ^∞ error in I	ℓ^2 error in V_1	ℓ^2 error in I
2^{15}	2.97×10^{-5}	19	1.03×10^{-5}	1.24×10^{-2}	4.80×10^{-6}	1.06×10^{-4}
2^{14}	5.94×10^{-5}	20	2.11×10^{-5}	1.81×10^{-2}	9.90×10^{-6}	2.22×10^{-4}
2^{13}	1.19×10^{-4}	19	4.25×10^{-5}	2.61×10^{-2}	2.00×10^{-5}	4.40×10^{-4}
2^{12}	2.38×10^{-4}	18	8.51×10^{-5}	3.71×10^{-2}	4.00×10^{-5}	8.51×10^{-4}
2^{11}	4.75×10^{-4}	16	1.70×10^{-4}	5.20×10^{-2}	8.01×10^{-5}	1.64×10^{-3}
2^{10}	9.50×10^{-4}	16	3.41×10^{-4}	7.23×10^{-2}	1.60×10^{-4}	3.15×10^{-3}

(b) Scenario 2 ($\alpha_o = 0.8$, $\gamma_2 = 0.005$) rescaled.

No. of grid points	Mesh size Δz	No. of iter.	ℓ^∞ error in V_1	ℓ^∞ error in I	ℓ^2 error in V_1	ℓ^2 error in I
2^{15}	2.97×10^{-5}	14	6.26×10^{-6}	3.06×10^{-4}	1.68×10^{-6}	1.66×10^{-5}
2^{14}	5.94×10^{-5}	14	1.27×10^{-5}	7.07×10^{-4}	3.41×10^{-6}	3.37×10^{-5}
2^{13}	1.19×10^{-4}	13	2.59×10^{-5}	1.47×10^{-3}	6.90×10^{-6}	6.79×10^{-5}
2^{12}	2.38×10^{-4}	12	5.22×10^{-5}	2.90×10^{-3}	1.39×10^{-5}	1.36×10^{-4}
2^{11}	4.75×10^{-4}	14	1.06×10^{-4}	5.53×10^{-3}	2.80×10^{-5}	2.73×10^{-4}
2^{10}	9.50×10^{-4}	16	2.12×10^{-4}	1.01×10^{-2}	5.62×10^{-5}	5.48×10^{-4}

(c) Scenario 3 ($\alpha_o = 1.0$, $\gamma_2 = 0.050$) rescaled.

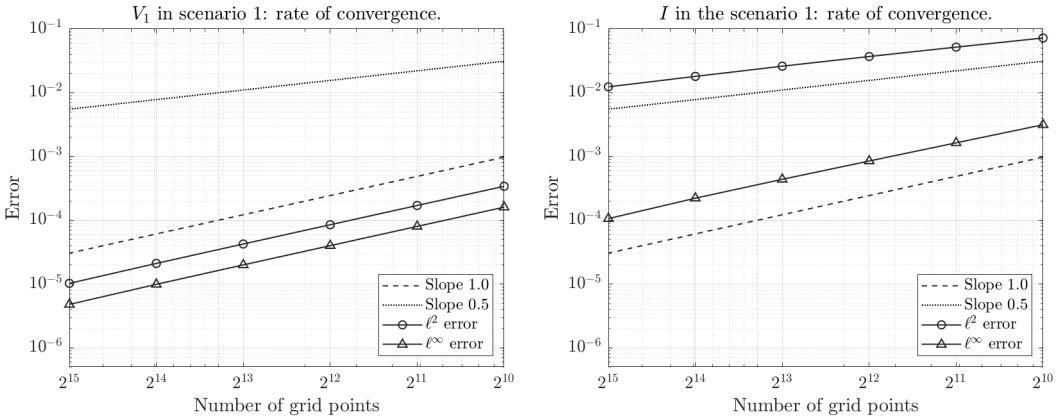
No. of grid points	Mesh size Δz	No. of iter.	ℓ^∞ error in V_1	ℓ^∞ error in I	ℓ^2 error in V_1	ℓ^2 error in I
2^{15}	1.53×10^{-5}	13	6.86×10^{-6}	4.69×10^{-3}	5.79×10^{-6}	1.19×10^{-4}
2^{14}	3.05×10^{-5}	14	1.39×10^{-5}	6.93×10^{-3}	1.18×10^{-5}	2.24×10^{-4}
2^{13}	6.10×10^{-5}	14	2.79×10^{-5}	9.88×10^{-3}	2.37×10^{-5}	4.13×10^{-4}
2^{12}	1.22×10^{-4}	12	5.59×10^{-5}	1.37×10^{-2}	4.75×10^{-5}	7.50×10^{-4}
2^{11}	2.44×10^{-4}	9	1.12×10^{-4}	1.87×10^{-2}	9.51×10^{-5}	1.34×10^{-3}
2^{10}	4.88×10^{-4}	9	2.23×10^{-4}	2.48×10^{-2}	1.90×10^{-4}	2.37×10^{-3}



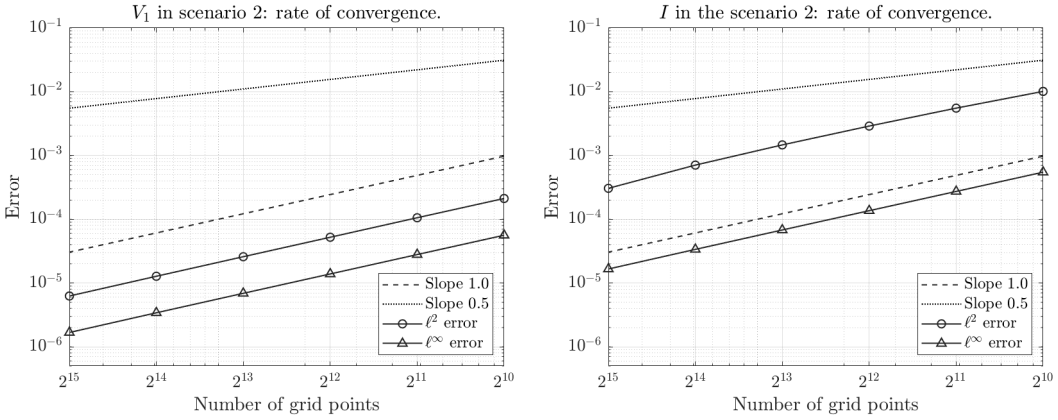
(a) Scenario 1. (b) Scenario 2 (c) Scenario 3

FIGURE 4.4. Convergence of the policy iteration.

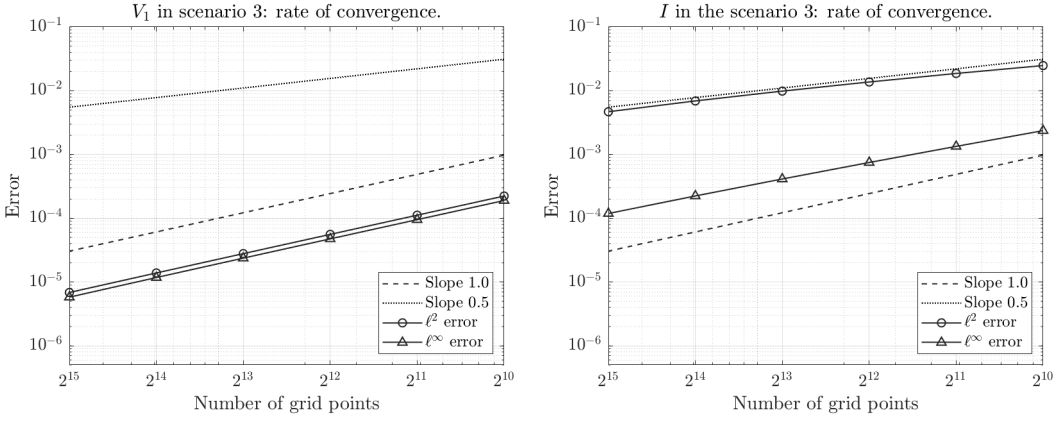
computed by collocation method as left boundary condition while in Figs. 4.2c–4.2d we used the condition that V_1 vanishes at infinity, that is $V_1(x(z)) \to 0$ when $z \to 0$. Moreover, the value V_2



(a) V_1 with $\alpha_o = 0.8$ and $\gamma_2 = 0.050$. (b) I with $\alpha_o = 0.8$ and $\gamma_2 = 0.050$



(c) V_1 with $\alpha_o = 0.8$ and $\gamma_2 = 0.005$. (d) I with $\alpha_o = 0.8$ and $\gamma_2 = 0.005$.



(e) V_1 with $\alpha_o = 1.0$ and $\gamma_2 = 0.050$. (f) I with $\alpha_o = 1.0$ and $\gamma_2 = 0.050$.

FIGURE 4.5. Rate of convergence

has been computed only for $x \in (-9.19, 9.19)$ and was extended by zero to the entire domain $\mathbb{R}$. For instance for $x < -9$, we can clearly see that both solutions do not agree.

There is a further explanation of the behavior at the left border of the domain. The original equation (1.1) is defined on $\mathbb{R}$. By solving (1.1) on the finite domain $(-9.19, 9.19)$ we have made an approximation which induces a boundary layer at the region close to the left border, see Figs. 4.2a–4.2b. Rescaling (1.1) into (2.5) does not solve this issue completely since we have also have to make a cutoff of the domain to solve the rescaled equation (2.5). Indeed, the transformation (2.4) is not defined neither at $z = 0$ nor at $z = 1$. Therefore, we must solve (2.5) on $(\texttt{eps}, \tilde{z})$, where $\texttt{eps} = 10^{-16}$ and $\tilde{z}$ is a point such that $V_1(x(\tilde{z}))$ is known for every $z \geqslant \tilde{z}$,

see Section 2.2. This explains the boundary layer in Figs. 4.2c–4.2d. Similar observations was observed by [1] for different problem defined on an unbounded domain.

4.2.2. *Approximation of the control.* In our numerical experiments, we used the centered finite differences to approximate the derivative in (4.6) and in Algorithm 4.1. Of course, other choices are possible namely the backward differences D_z^-, the forward differences D_z^+, and the upwind differences $\mathrm{D}_z^\pm$, see (4.2) and (4.3). Under different configuration of the couple of parameters (α_o, γ_2), Fig. 4.4 illustrates the sensitivity of the policy iteration according to the choice of the finite differences scheme used in the policy improvement step of Algorithm 4.1. The policy iteration converges with the centered finite differences in every configuration of the model. It converges even quicker with the forward finite differences for large γ_2, see Fig. 4.4b and with the backward finite differences for large α_o, see Fig. 4.4c. Even though the combination of the policy iteration and upwind differences is more natural since it can be deduced from the optimality condition of (4.4), the scheme diverges in every scenario.

We also observe that the number of iterations changes with the number of grid points, see Table 4.3. It is also sensitive to the parameters α_o and γ_2. Larger γ_2, see Table 4.3a, require less iterations and larger α_o, see Table 4.3c, require more.

4.2.3. *Rate of convergence.* Since no closed form or analytical solution is known, we determine the rate of convergence by using a reference solution computed with a large number of grid points $M = 2^{20}$. Fig. 4.5 illustrates the relative ℓ^∞ and the ℓ^2 errors and indicate convergence with order 1, despite the ℓ^2 error for the approximation of the optimal control in the scenarios 1 and 3 converging with reduced rate 1/2.

5. Discretization by finite elements

In this section we introduce finite element schemes for the approximation of (1.1) and (1.2). We note that, opposed to the collocation and the finite difference methods, we do not employ the transformation (2.3) but instead work on domains of finite size, that is we consider (1.1) for $x \in (-20, 20)$ or $x \in (-10, 10)$ and (1.2) for $(x, y) \in (-20, 0) \times (0, 20)$ and impose suitable conditions along the boundary of the respective domains. This choice is motivated by two reasons. Firstly we want to examine the effect of the (artificial) boundary conditions on the numerical solution and secondly we want to demonstrate the effect of adaptive mesh refinement which can be done more naturally on the unrescaled domain.

Throughout the section we use the following notation. By $L^2(\Omega)$ we denote the standard Lebesgue space of square integrable functions on Ω. For $u, v \in L^2(\Omega)$ we denote the L^2-inner product by $(u, v) := \int_\Omega u(x)v(x)\,\mathrm{d}x$ with the corresponding L^2-norm $\|u\|_{L^2(\Omega)} := (u, u)^{1/2}$.

5.1. **Discretization of the value in mode $m = 2$.** We note that for $x \geqslant 0$ by Proposition 2.1 the solution of the HJB equation (1.2) is given explicitly by the formula in (2.1). Hence, similarly as in Section 3, we only consider numerical approximation of V_2 for $x < 0$ and impose a Dirichlet boundary condition at $x = 0$ via (2.1). In addition, in order to solve V_1 we only need to obtain the solution V_2 at $y = 0$. Therefore, to obtain a finite element approximation of V_2 we solve the HJB equation (1.2) on the bounded domain $\Omega^- = (-20, 0) \times (0, 20)$. The boundary of Ω^- is denoted as $\partial\Omega^-$ and the boundary at $x = 0$ is denoted as $\Gamma_D := \partial\Omega^- \cap (0, y)$, $y \in [0, 20]$. We remark that by using the transformation from Section 2 it is possible to compute numerical solution on an unbounded domain $(-\infty, 0) \times (0, 20)$, but we are not going to pursue this approach here.

For simplicity we construct the finite element scheme using uniform triangulations $\mathcal{T}_h$ of Ω^-. For a fixed mesh size parameter $h > 0$ the triangulation $\mathcal{T}_h$ is obtained by partitioning Ω^- into uniform squares with side h and each square is then sub-divided into four equally sized right-angled triangles. Given a triangulation $\mathcal{T}_h$ of Ω^- we denote by $S^1(\mathcal{T}_h)$ the standard finite element space of continuous functions which are piece-wise linear with respect to the triangulation $\mathcal{T}_h$. The set of nodes $\{\mathbf{x}_\ell \equiv (x_\ell, y_\ell)\}_{\ell=1}^L$ of $\mathcal{T}_h$ is denoted as $\mathcal{N}_h$. We also introduce a finite element space of functions that satisfy the homogeneous Dirichlet boundary condition on Γ_D as $\mathbb{V}_h = \{\phi_h \in S^1(\mathcal{T}_h),\ \phi_h|\Gamma_D = 0\}$.

The standard nodal interpolation operator is denoted as $\mathcal{I}_h : C(\overline{\Omega}^-) \to S^1(\mathcal{T}_h)$, where for $v \in C(\overline{\Omega}^-)$ the interpolant $\mathcal{I}_h v \in S^1(\mathcal{T}_h)$ is defined via the relation $\mathcal{I}_h v(\mathbf{x}_\ell) = v(\mathbf{x}_\ell)$, $\forall \mathbf{x}_\ell \in \mathcal{N}_h$. For $u, v \in C(\overline{\Omega}^-)$ define the the discrete L^2-inner product for $\Omega^- \subseteq \mathbb{R}^2$ as

$$(u,v)_h = \int_{\Omega^-} \mathcal{I}_h(uv)(x)\mathrm{d}x = \sum_{T\in\mathcal{T}_h} \frac{|T|}{3} \sum_{i=1}^{3} u(\mathbf{v}_{T,i})v(\mathbf{v}_{T,i})\,, \tag{5.1}$$

where $\mathbf{v}_{T,1}, \mathbf{v}_{T,2}, \mathbf{v}_{T,3} \in \mathcal{N}_h$ denote the vertices of the triangle $T \in \mathcal{T}_h$.

We note that the considered finite element space of piecewise linear continuous functions only satisfies $S^1(\mathcal{T}_h) \subset H^1(\Omega^-)$, that is the second order derivatives are not defined for $v_h \in S^1(\mathcal{T}_h)$. For $v_h \in S^1(\mathcal{T}_h)$ we introduce the discrete second order operator $\delta^2_{xx,h} : S^1(\mathcal{T}_h) \to S^1(\mathcal{T}_h)$, $\delta^2_{xx,h} \approx \partial^2_{xx}$ as

$$(-\delta^2_{xx,h} v_h, \phi_h)_h = (\partial_x v_h, \partial_x \phi_h) \qquad \forall \phi_h \in S^1(\mathcal{T}_h)\,, \tag{5.2}$$

and similarly we define $\delta^2_{yy,h} \approx \partial^2_{yy}$. Consequently the discrete Laplace operator $\Delta_h := \delta^2_{xx,h} + \delta^2_{yy,h} : S^1(\mathcal{T}_h) \to S^1(\mathcal{T}_h)$ satisfies

$$(-\Delta_h v_h, \phi_h)_h = (\nabla v_h, \nabla \phi_h) \qquad \forall \phi_h \in S^1(\mathcal{T}_h)\,. \tag{5.3}$$

Let $\{\phi_\ell\}_{\ell=1}^L$ be the set of nodal basis functions of $S^1(\mathcal{T}_h)$, i.e. $S^1(\mathcal{T}_h) = \mathrm{span}\{\phi_1, \dots, \phi_L\}$ and $v_h \in S^1(\mathcal{T}_h)$ can be expressed as $v_h(x) = \sum_{\ell=1}^L v_\ell \phi_\ell(x)$ with coefficients $v_\ell \in \mathbb{R}$. By the definition of the discrete inner product (5.1) and the fact that $\phi_\ell(\mathbf{x}_m) = \delta_{\ell m}$ for $\mathbf{x}_m \in \mathcal{N}_h$, we deduce from (5.2) that

$$\partial^2_{xx} v_h(\mathbf{x}_\ell) = 3|\mathrm{supp}(\phi_\ell)|^{-1}(\partial_x v_h, \partial_x \phi_\ell) \quad \forall \mathbf{x}_\ell \in \mathcal{N}_h,$$

as well as

$$\Delta_h v_h(\mathbf{x}_\ell) = 3|\mathrm{supp}(\phi_\ell)|^{-1}(\nabla v_h, \nabla \phi_\ell) \quad \forall \mathbf{x}_\ell \in \mathcal{N}_h,$$

cf. [17, 13, 21].

The finite element approximation of the continuous problem is then constructed as follows. The finite element solution $V_{2,h} \in S^1(\mathcal{T}_h)$ satisfies the Dirichlet boundary condition at Γ_D, that is $V_{2,h}(\mathbf{x}_\ell) = V_2(0, y_\ell)$ for $\mathbf{x}_\ell = (0, y_\ell) \in \mathcal{N}_h \cap \Gamma_D$ (where $V_2(0,\cdot)$ is given by (2.1)) and solves the discrete syslem

$$\Big(-\tfrac{1}{2}\sigma^2 \delta^2_{yy,h} V_{2,h} + \mathbf{b}\cdot\nabla V_{2,h} + (r+p_0)V_{2,h}, \phi_h\Big)_h - \Big(h_{\mathtt{stab}} \Delta_h V_{2,h}, \phi_h\Big)_h = (f_2, \phi_h)_h \;\forall \phi_h \in \mathbb{V}_h\,. \tag{5.4}$$

where $h_{\mathtt{stab}}(\mathbf{x}_\ell) = h|\mathbf{b}(\mathbf{x}_\ell)|$, with $\mathbf{b} = -(b_2, \mu)$.

Equivalently we may rewrite (5.4) as

$$r\big(V_{2,h}, \phi_h\big)_h = \Big(\mathcal{L}_{2,h}(V_{2,h}) + f_2, \phi_h\Big)_h + \Big(h_{\mathtt{stab}} \Delta_h V_{2,h}, \phi_h\Big)_h \qquad \forall \phi_h \in \mathbb{V}_h\,, \tag{5.5}$$

with $\mathcal{L}_{2,h}(V_{2,h}) := \frac{1}{2}\sigma^2(y)\delta^2_{yy,h} V_{2,h}(x,y) - \mathbf{b}(x,y)\cdot\nabla V_{2,h}(x,y) - p_0(x)V_{2,h}(x,y)$ which is a discrete counterpart of (1.4) with an additional artificial diffusion term $h_{\mathtt{stab}}\Delta_h V_{2,h}$.

The artificial diffusion term guarantees the monotonicity and convergence of the finite element approximation, cf. [17] and [21, Section 3.5]. It is well know that the nodal basis functions of $S^1(\mathcal{T}_h)$ satisfy

$$(\nabla \phi_i, \nabla \phi_j) \leqslant 0 \quad \text{for } i \neq j\,, \tag{5.6}$$

for weakly acute triangulation $\mathcal{T}_h$. Since we employ uniform right-angled triangulations (consequently $\mathcal{T}_h$ are weakly acute) in can be shown that the artificial diffusion term $-h_{\mathtt{stab}}\Delta_h V_{2,h}$ ensures the monotonicity of the finite element scheme (5.4), [21, Lemma 3.42]; a generalization for adaptive meshes which satisfy the property (5.6) is possible by a suitable choice of the stabilization parameter $h_{\mathtt{stab}}$, for more details see [17, Section 8] and [21]. By the monotonicity of the numerical approximation standard arguments imply that the finite element solution $V_{2,h}$ of (5.4) converges to the (unique) viscosity solution of (1.2) (considered on Ω^-), cf. [17]. On noting the representation $V_{2,h}(x) = \sum_{\ell=1}^L v_{2,\ell}\phi_\ell(x)$ we observe that (5.4) is equivalent to a system of linear equations for the coefficients $\{v_{2,\ell}\}_{\ell=1}^L$. Due to the monotonicity of the numerical approximation

the associated matrix is a M-matrix, which also implies a maximum principle for the numerical approximation (5.4), see for instance [21, Corollary 3.43].

Surprisingly, the numerical scheme (5.4) also works for $h_{\texttt{stab}} \equiv 0$ and appears to provide a numerical solution which is monotone (without oscillations).

5.2. **Discretization of the value in mode $m = 1$.** In this subsection we investigate the approximation of V_1 by a continuous and piece-wise affine function $V_{1,h} \in \mathcal{T}$, where the underlying triangulation $\mathcal{T}$ of the domain $\Omega \subset \mathbb{R}$ results from an adaptive mesh refinement strategy, that is, the computation starts with a coarse initial grid $\mathcal{T}_0$ and than performs local mesh refinements. This strategy is motivated by the fact that the solution V_1 and the control I^{opt} are rather smooth despite some singularities close to $x = 0$ and close to the boundary due to the "artificial" boundary layer, cf. Section 4.2.1. Indeed, it is known that adaptive schemes lead to optimal convergence rates for certain model problems, see [9] for an overview. We propose the following adaptive numerical scheme sharing some similarities with the policy iteration in Algorithm 4.1. We stabilize the policy improvement step by using the L^2-projection Π_0 onto piece-wise constant functions $\mathbb{P}_0(\mathcal{T}) := \{v \in L^\infty(\Omega) \mid v|_T \in \mathbb{R} \text{ for all } T \in \mathcal{T}\}$ and a damped steps size.

Algorithm 5.1 (Adaptive Scheme)**.**
Outer loop: *Given some initial triangulation $\mathcal{T}_0$, the initial guess $I_0 := I^{\text{nc}}$ for the optimal control, and an approximation of V_2, we perform for each $\ell \in \mathbb{N}$ the following steps.*

Inner loop: *Set $I^{(0)} := I_\ell$ and solve for each $n \in \mathbb{N}$ the following problems.*

Policy evaluation: *Compute an approximation $V_{1,h}^{(n)} \in S^1(\mathcal{T}_\ell)$ of the solution to the scalar convection problem*

$$-b_1(I^{(n-1)})\partial_x V_1^{(n)} + (p_0 + p_2(I^{(n-1)}) + r)V_1^{(n)} = f_1 + p_2(I^{(n-1)})V_2(\cdot, 0) \quad \text{in } \Omega. \tag{5.7}$$

Policy improvement: *Update the optimal control by (cf. (3.10))*

$$I^{(n)} := \Pi_0 \left(I^{(n-1)} + 0.5 \left(\frac{\gamma_1(V_2(\cdot,0) - V_{1,h}^{(n)})}{\xi \partial_x V_{1,h}^{(n)}} - I^{(n-1)} \right) \right). \tag{5.8}$$

Stopping criterion: Repeat the loop until the relative change of the control $I^{(n)}$ is below some given threshold ϵ_ℓ depending on $\text{ndof} := \dim S^1(\mathcal{T}_\ell)$*, that is,*

$$\frac{\|I^{(n)} - I^{(n-1)}\|_{L^2(\Omega)}}{\|I^{(n)}\|_{L^2(\Omega)}} \leqslant \epsilon_\ell \qquad \text{with } \epsilon_\ell := \text{ndof}^{-1}. \tag{5.9}$$

Set the function $I_{\ell+1} := I^{(n)}$ and compute refinement indicators. These refinement indicators drive an adaptive mesh refinement resulting in a new triangulation $\mathcal{T}_{\ell+1}$.

We approximate the solution $V_1^{(n)}$ to (5.7) by the least-squares finite element method (LSFEM). This method approximates the solution to a partial differential equation by the minimization of some (artificial) energy. In case of (5.7) this minimization reads

$$V_{1,h}^{(n)} = \operatorname*{arg\,min}_{W_h \in S^1(\mathcal{T})} \|-b_1(I^{(n-1)})\partial_x W_h + (p_0 + p_2(I^{(n-1)}) + r)W_h - f_1 - p_2(I^{(n-1)})V_2(\cdot,0)\|_{L^2(\Omega)}. \tag{5.10}$$

The LSFEM is a popular numerical scheme in fluid dynamics and elasticity. It is competitive with the SUPG and superior to the Galerkin finite element method for advection-diffusion equations [3, 4]. Although there exist no theoretical results for the LSFEM and general HJB equations, a first investigation in [19] shows convergence for linear elliptic Dirichlet boundary value problems in non-divergence form with coefficients satisfying the Cordes condition. We choose the LSFEM for our computations due to the following reasons. It seems that we do not have to enforce boundary conditions. This is beneficial, since the left boundary data is without a rescaling of the domain unknown. Moreover, the LSFEM is easy to implement, results in a linear system of equations with symmetric positive definite matrix, and provides a built-in error control. Let us explain the latter two advantages in more detail. The minimization in (5.10) is equivalent to the

minimization of the error $W_h - V_1^{(n)}$ over all $W_h \in S^1(\mathcal{T})$ with respect to the semi-norm

$$\|\cdot\|_{a,I^{(n-1)}} := \|-b_1(I^{(n-1)})\partial_x \cdot + (p_0 + p_2(I^{(n-1)}) + r)\cdot\|_{L^2(\Omega)} \quad \text{in } H^1(\Omega).$$

If there exists a unique solution $V_1^{(n)} \in H^1(\Omega)$ to (5.7), the semi-norm is indeed a norm. This implies the well-posedness of the minimization problem in (5.10) and the resulting algebraic problem involves a symmetric positive definite matrix. Hence we can use efficient solvers for the corresponding linear system of equations. Furthermore, the evaluation of the residual on each $T \in \mathcal{T}$ leads to the computable refinement indicators

$$\begin{aligned}&\|-b_1(I^{(n-1)})\partial_x V_{1,h}^{(n)} + (p_0 + p_2(I^{(n-1)}) + r)V_{1,h}^{(n)} - f_1 - p_2(I^{(n-1)})V_2(\cdot,0)\|_{L^2(T)}\\ &\qquad = \|-b_1(I^{(n-1)})\partial_x(V_{1,h}^{(n)} - V_1^{(n)}) + (p_0 + p_2(I^{(n-1)}) + r)(V_{1,h}^{(n)} - V_1^{(n)})\|_{L^2(T)}.\end{aligned}$$

We use this built-in error control to drive adaptive mesh refinements with Dörfler marking strategy and bulk parameter $\Theta = 0.3$.

We terminate our computations if the number of iterations in an inner loop exceeds fifteen or the number of degrees of freedom ndof $= \dim S^1(\mathcal{T})$ exceeds half a million. All experiments utilize the parameters in Tables 2.1–2.2 and approximations of V_2 computed separately. The initial grid $\mathcal{T}_0$ is a uniform partition of the domain Ω into 10 subintervals in all experiments. We refine the mesh either adaptively or uniformly. The computations utilize the open source tool for solving partial differential equations FEniCS [20].

5.2.1. *Numerical experiment scenario 1 ($\alpha_o = 0.8$, $\gamma_2 = 0.05$).* The parameters in this experiment read $\alpha_o = 0.8$ and $\gamma_2 = 0.05$. We use the same approximation of V_2 as in the computation in Section 4. This approximation results from the collocation method in Section 3. The underlying domain reads $\Omega = (-9.19, 9.19)$. Our numerical experiments show a boundary layer close the left boundary in $x = -9.19$. This layer leads to additional difficulties in the convergence of the inner loop. To neglect this artificial effect, we modify the stopping criterion in (5.9). Rather than comparing the iteratively computed controls on the entire domain Ω, we restrict the integration to the subdomain $\Omega_{\rm red} := (-5, 9.19)$. More precisely, the stopping criterion reads

$$\frac{\|I^{(n)} - I^{(n-1)}\|_{L^2(\Omega_{\rm red})}}{\|I^{(n)}\|_{L^2(\Omega_{\rm red})}} \leqslant \epsilon_\ell \qquad \text{with} \qquad \epsilon_\ell := \text{ndof}^{-1}. \tag{5.11}$$

Figure 5.6 displays the convergence history plot, where the finite difference solution with $M = 2^{20}$ grid points from Section 4 serves as reference solution. Notice that the top left plot in Figure 5.6 shows that the behavior of the control close to the left boundary $x = -9.19$ differs significantly compared to the computation with the finite difference method on the rescaled domain. However, the convergence history plots on the right hand side of Figure 5.6 indicates that this phenomenon does not have a decisive impact on the control I and the value function V_1 in the domain $\Omega_{\rm red}$. Indeed, the plot shows that the L^2 and L^∞ error for the value function $V_{1,h}$ converges with the rate $\mathcal{O}(\text{ndof}^{-1})$ with adaptive and uniform mesh refinements. The error in the approximation of the control seems to be slightly worse than $\mathcal{O}(\text{ndof}^{-1})$ for the L^2 error and slightly worse than $\mathcal{O}(\text{ndof}^{-1/2})$ for the L^∞ error. As ndof approaches 10^4, we observe difficulties. This indicates that more accurate computations might require some stabilization as for example in the scheme in Section 5.1.

5.2.2. *Numerical experiment for scenario 2 ($\alpha_o = 0.8$, $\gamma_2 = 0.005$).* If we use the approximation V_2 from the collocation method in this scenario, the computed controls with the finite difference method and the LSFEM seem to differ slightly. A more accurate approximation with the finite element scheme from Section 5.1 overcomes that difficulty and results in similar approximations of V_1, despite the (artificial) boundary layer close to the left boundary of the underlying domain $\Omega = \Omega_{20} = (-20, 20)$ (cf. Figure 5.7). The adaptive scheme seems to be very stable, that is, the inner loop always terminates after maximal 5 iterations. With uniformly refined meshes the computations fails after ndof exceeds 81921.

Figure 5.7 displays the convergence history plot of the computed value function $V_{1,h}$ and control I_h, where we use the finite difference solution with $M = 2^{20}$ grid points and the same

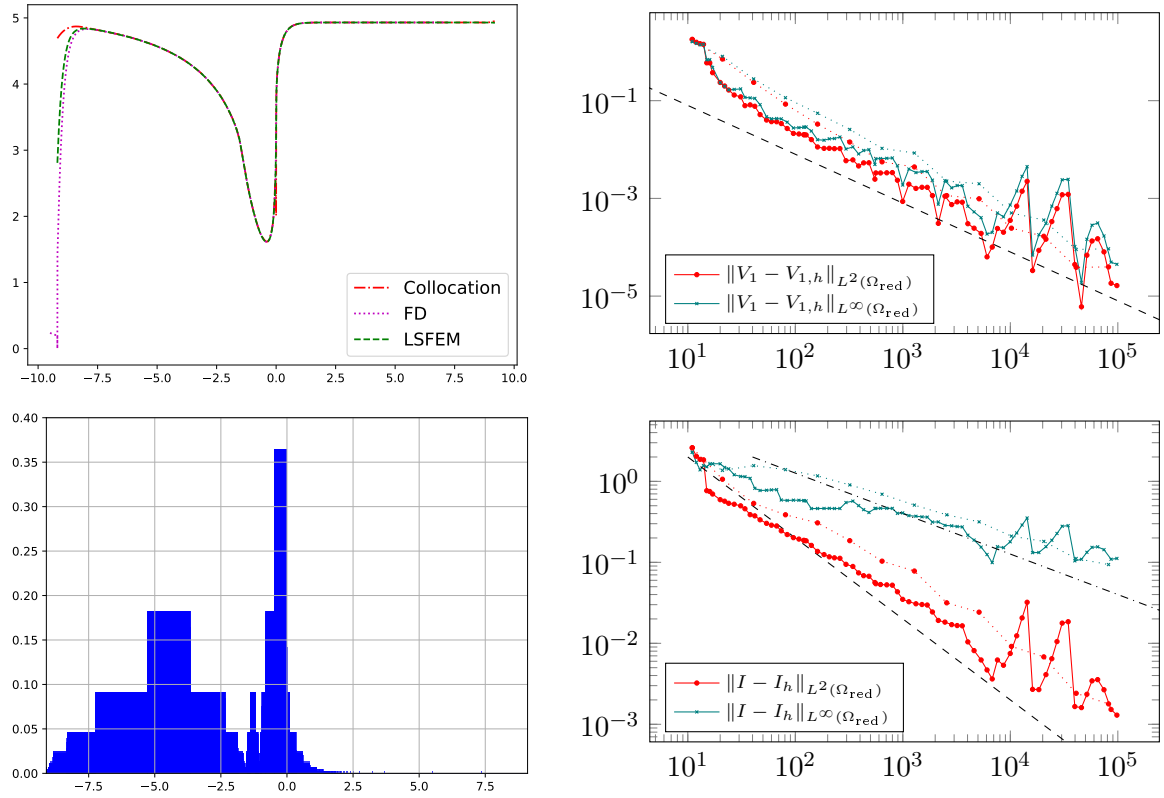


FIGURE 5.6. Computations for Scenario 1. Top left: Computed controls I_h with the collocation method, the finite difference (FD) method, and the LSFEM. Top and bottom right: Solid lines represent the error of the adaptive scheme, dotted lines represent the error on uniformly refined meshes. Dashed lines - - - indicate the rate $\mathcal{O}(\mathrm{ndof}^{-1})$ and the dash-dotted line -·-·- indicates the rate $\mathcal{O}(\mathrm{ndof}^{-1/2})$. Bottom left: Density of vertices in the finest adaptively refined mesh $\mathcal{T}$.

approximation of V_2 serves as reference solution. The figure displays the error on the domain $\Omega_{10} = (-10, 10)$. We observe that the L^2 and L^∞ error of the control I_h behave approximately like ndof^{-1}. This rate is achieved with adaptive and uniform mesh refinements. However, the uniform mesh refinement results in a pre-asymptotic regime with reduce speed of convergence. The L^2 and L^∞ errors for the approximated value function V_1 seem to converge with a rate of almost ndof^{-2}. In linear model problems such fast rates can be explained by the duality arguments of Aubin and Nitzsche, see for example [5]. After ndof exceeds 5000 the errors $\|V_1 - V_{1,h}\|_{L^2(\Omega_{10})}$ and $\|V_1 - V_{1,h}\|_{L^\infty(\Omega_{10})}$ remain almost constant. This might be caused by the fact that the seemingly beneficial convergence properties of the LSFEM scheme lead to a much better approximation (with respect to the L^2 and L^∞ norm) of the solution V_1 than the finite difference scheme with significantly more degrees of freedom.

In addition Figure 5.7 visualizes the control I_h computed with the collocation method, the finite difference method, and the LSFEM with underlying domain Ω_{10} and V_2 computed on the domain $(-10, 0) \times (0, 20)$ as well as with underlying domain Ω_{20} and V_2 computed on the domain $(-20, 0) \times (0, 20)$. The plot on the bottom left of Figure 5.7 shows that the adaptive mesh refinement focuses on the domain close to 0 and between -17 and -8.

5.2.3. *Numerical experiment scenario 3 ($\alpha_o = 1$, $\gamma_2 = 0.05$).* In this computation we use an approximation of V_2 with the collocation method on the domain $\Omega = (-9.19, 9.19)$. We compare our results to the reference solution resulting from the finite difference scheme with $M = 2^{20}$ grid points an the same approximation of V_2. In our experiments the inner loop for the uniformly refined meshes does not terminate as ndof exceeds 20000, the adaptively refined schemes seems to be stable in the sense that the inner loop terminates after one or two iterations.

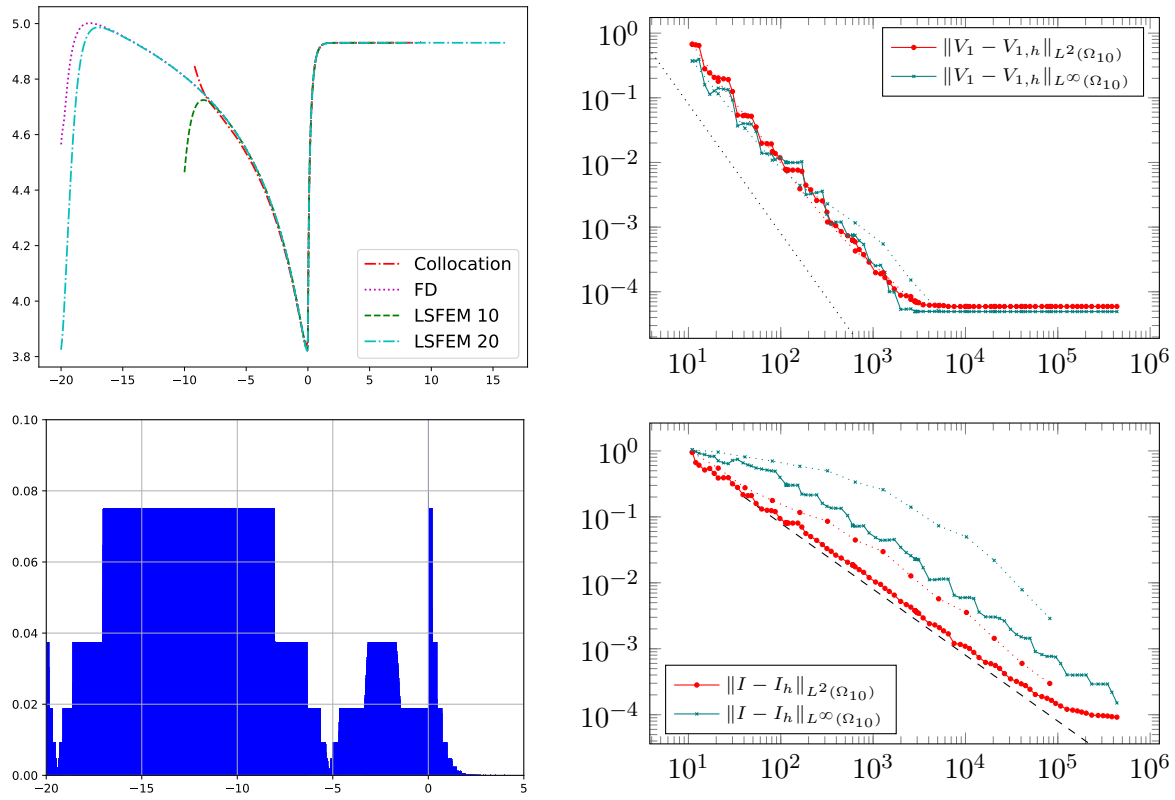


FIGURE 5.7. Computations for scenario 2. Top left: Computed controls I_h with the collocation method, the finite difference (FD) method, and the LSFEM (LSFEM 10 is computed on Ω_{10} and LSFEM 20 on Ω_{20}). Top and bottom right: Solid lines represent the error of the adaptive scheme, dotted lines represent the error on uniformly refined meshes. The dashed lines - - - indicates the rate $\mathcal{O}(\text{ndof}^{-1})$, the dotted line ····· indicates the rate $\mathcal{O}(\text{ndof}^{-2})$. Bottom left: Density of vertices in the finest adaptively refined mesh $\mathcal{T}$.

Figure 5.8 visualizes the approximations of the control I. Again, there is some boundary layer close to the left boundary $x = -9.19$ of the domain Ω. Therefore, the convergence history of our approximation $V_{1,h}$ displays the error on the reduced domain $\Omega_{\text{red}} = (-5, 9.19)$. As in the previous experiment the L^2 and L^∞ error of the approximated value function $V_{1,h}$ seems to converge with the rate ndof^{-2} and so outperforms the approximation with the finite difference scheme. The L^2 error of the control seems to converge approximately with the rate ndof^{-1} and the L^∞ error converges, unlike in the previous experiment, with the rate $\text{ndof}^{-1/2}$. Similar to the previous experiment, the adaptive scheme seems to be more stable than the uniform mesh refinement in the sense that the inner loop in the uniform scheme does not terminate as ndof exceeds 20481, the inner loop in the adaptive scheme requires less than two iterations in each iteration.

6. Conclusion

We conclude by noting that, for the considered parameters, the computed numerical approximations were similar for all three numerical schemes. Nevertheless, we made some interesting observations (in particular in the solution for $m = 1$) which we summarize below.

Boundary/transformation effects: There were no notable differences between the computed approximation of V_1 between the different numerical schemes. Compared to other methods, the adaptive LSFEM methods exhibited slightly faster convergence behavior, nevertheless due to the fact the V_1 is rather smooth the differences were not significant. For x close to the origin, the computed optimal controls also exhibited very good agreement. Nevertheless, near the left

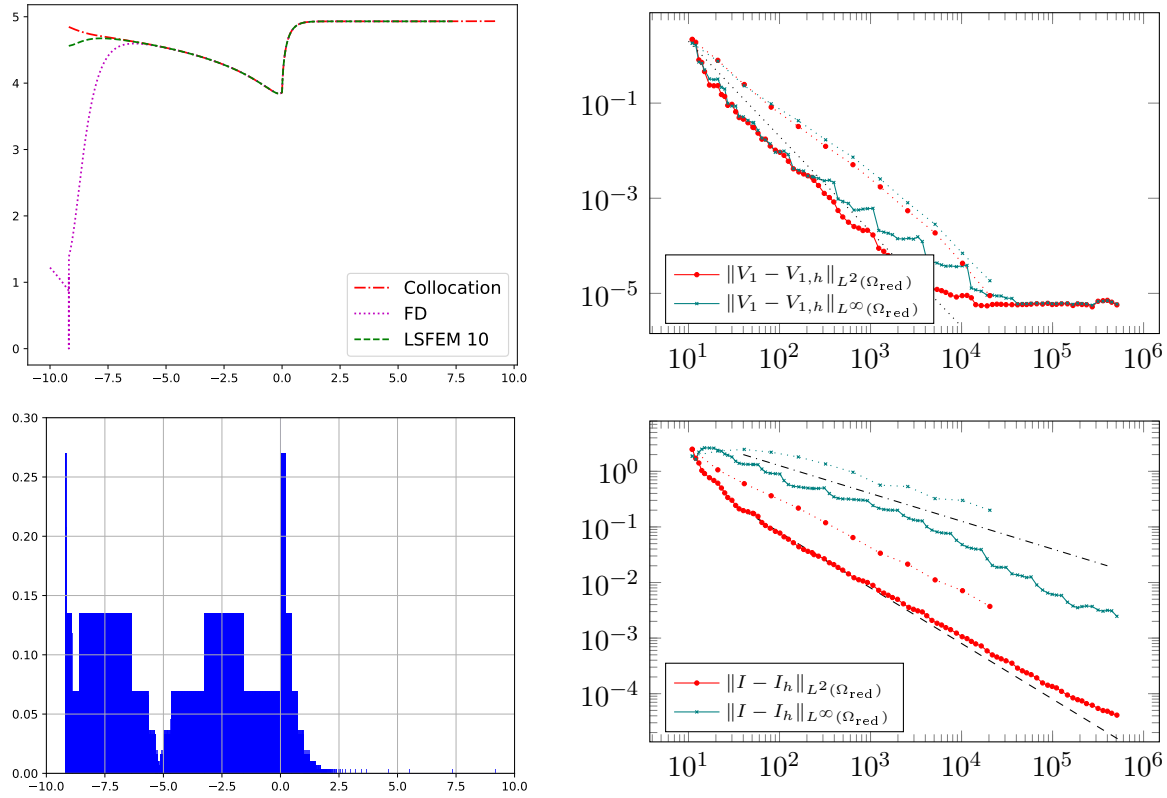


FIGURE 5.8. Computations for scenario 3. Top left: Computed controls I_h with the collocation method, the finite difference (FD) method, and the LSFEM. Top and bottom right: Solid lines represent the error of the adaptive scheme, dotted lines represent the error on uniformly refined meshes. The dashed line ····· indicates the rate $\mathcal{O}(\mathrm{ndof}^{-1})$, the dotted line ····· indicates the rate $\mathcal{O}(\mathrm{ndof}^{-2})$. Bottom left: Density of vertices in the finest adaptively refined mesh $\mathcal{T}$.

boundary the three methods give very different numerical solutions. This can be explained by the following consideration. The collocation method and the finite difference method are applied on the infinite domain $(-\infty, \infty)$ via the coordinate transformation on $(0, 1)$. The uniform discretization on $(0, 1)$ can be interpreted as a discretization of the original problem with increasing mesh size for growing $|x|$, that is the approximation inevitably becomes inaccurate for large $|x|$. The finite element method is applied on a bounded domain, where the solution near the boundary is affected by the artificial boundary condition. We observe that the effect of the boundary can be mitigated by increasing the size of the computational domain, which is particularly useful in combination with adaptive mesh refinement which potentially reduces the overall computational cost without the loss of accuracy on larger domains.

Convergence of the policy iteration: It was necessary to modify the policy iteration for each of the considered numerical schemes in order to ensure convergence of nonlinear iterations. It appears that, the sensitivity of the iterative scheme is due to the fact that the optimal control can become discontinuous at $x = 0$, as indicated by the numerical experiments. Nevertheless, a rigorous analysis of the convergence behavior of the policy iteration scheme for the present problem is still missing.

Approximation of V_2: The numerical solution of V_2 enters as a parameter in the numerical approximation of V_1, consequently the quality of approximation of V_1 will have influence on the numerical approximation of V_1. Due to the lack of artificial diffusion, we observed that the numerical approximation V_2 computed by collocation method contained minor oscillations. This effect can be attributed to the lack of numerical stabilization in the collocation scheme. A close inspection of the related numerical result reveals that these oscillations transfer to the

approximation of V_1. While the effect was minor and did not have notable impact on the accuracy of numerical solution in the present setup, the result suggest that the use of monotone schemes for V_2, such as the stabilized FEM method provides a more reliable numerical algorithm.

References

[1] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, 4(3):271–283, 1991.

[2] R. Bellman. Functional equations in the theory of dynamic programming. V. Positivity and quasi-linearity. *Proc. Nat. Acad. Sci. U.S.A.*, 41:743–746, 1955.

[3] P. B. Bochev and J. Choi. A comparative study of least-squares, SUPG and Galerkin methods for convection problems. *Int. J. Comput. Fluid Dyn.*, 15(2):127–146, 2001.

[4] P. B. Bochev and J. Choi. Improved least-squares error estimates for scalar hyperbolic problems. *Comput. Methods Appl. Math.*, 1(2):115–124, 2001.

[5] D. Braess. *Finite elements*. Cambridge University Press, Cambridge, third edition, 2007. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker.

[6] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numer.*, 13:147–269, 2004.

[7] F. Camilli and M. Falcone. An approximation scheme for the optimal control of diffusion processes. *RAIRO Modél. Math. Anal. Numér.*, 29(1):97–122, 1995.

[8] R. Carmona. *Lectures on BSDEs, stochastic control, and stochastic differential games with financial applications*, volume 1 of *Financial Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2016.

[9] C. Carstensen, M. Feischl, M. Page, and D. Praetorius. Axioms of adaptivity. *Comput. Math. Appl.*, 67(6):1195–1253, 2014.

[10] H. Dawid and X. Wen. Innovation investment with financial constraint. preprint, November 2019.

[11] K. Debrabant and E. R. Jakobsen. Semi-Lagrangian schemes for linear and fully non-linear diffusion equations. *Math. Comp.*, 82(283):1433–1462, 2013.

[12] M. Falcone. A numerical approach to the infinite horizon problem of deterministic control theory. *Appl. Math. Optim.*, 15(1):1–13, 1987.

[13] X. Feng, R. Glowinski, and M. Neilan. Recent developments in numerical methods for fully nonlinear second order partial differential equations. *SIAM Rev.*, 55(2):205–267, 2013.

[14] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*, volume 25 of *Stochastic Modelling and Applied Probability*. Springer, New York, second edition, 2006.

[15] D. Gallistl and E. Süli. Mixed finite element approximation of the Hamilton-Jacobi-Bellman equation with Cordes coefficients. *SIAM J. Numer. Anal.*, 57(2):592–614, 2019.

[16] R. A. Howard. *Dynamic programming and Markov processes*. The Technology Press of M.I.T., Cambridge, Mass.; John Wiley & Sons, Inc., New York-London, 1960.

[17] M. Jensen and I. Smears. On the convergence of finite element methods for Hamilton-Jacobi-Bellman equations. *SIAM J. Numer. Anal.*, 51(1):137–162, 2013.

[18] R. Jensen. The maximum principle for viscosity solutions of fully nonlinear second order partial differential equations. *Arch. Rational Mech. Anal.*, 101(1):1–27, 1988.

[19] O. Lakkis and A. Mousavi. A least-squares galerkin approach to gradient and hessian recovery for nondivergence-form elliptic equations. *arXiv preprint arXiv:1909.00491*, 2019.

[20] A. Logg, K.-A. Mardal, and G. N. Wells, editors. *Automated solution of differential equations by the finite element method*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer, Heidelberg, 2012. The FEniCS book.

[21] M. Neilan, A. J. Salgado, and W. Zhang. Numerical analysis of strongly nonlinear PDEs. *Acta Numer.*, 26:137–303, 2017.

[22] H. Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, Berlin, 2009.

[23] A. J. Salgado and W. Zhang. Finite element approximation of the Isaacs equation. *ESAIM Math. Model. Numer. Anal.*, 53(2):351–374, 2019.

[24] B. Sun and B.-Z. Guo. Convergence of an upwind finite-difference scheme for Hamilton-Jacobi-Bellman equation in optimal control. *IEEE Trans. Automat. Control*, 60(11):3012–3017, 2015.

[25] N. Touzi. *Optimal stochastic control, stochastic target problems, and backward SDE*, volume 29 of *Fields Institute Monographs*. Springer, New York; Fields Institute for Research in Mathematical Sciences, Toronto, ON, 2013. With Chapter 13 by Angès Tourin.

[26] S. Wang, F. Gao, and K. L. Teo. An upwind finite-difference method for the approximation of viscosity solutions to Hamilton-Jacobi-Bellman equations. *IMA J. Math. Control Inform.*, 17(2):167–178, 2000.

[27] X. Warin. Adaptive sparse grids for time dependent hamilton-jacobi-bellman equations in stochastic control. *arXiv preprint arXiv:1408.4267*, 2014.