Numerical Approximation of a System of Hamilton–Jacobi–Bellman Equations Arising in Innovation Dynamics

L'ubomír Baňas1 · Herbert Dawid2 · Tsiry Avisoa Randrianasolo1 · Johannes Storn1 · Xingang Wen2

Received: 12 August 2021 / Revised: 5 May 2022 / Accepted: 11 May 2022 / Published online: 4 July 2022 © The Author(s) 2022

Abstract
We consider a system of fully nonlinear partial differential equations that corresponds to the Hamilton–Jacobi–Bellman equations for the value functions of an optimal innovation investment problem of a monopoly firm facing bankruptcy risk. We compare several algorithms for the numerical solution of the considered problem: the collocation method, the finite difference method, WENO method and the adaptive finite element method. We discuss implementation issues for the considered schemes and perform numerical studies for different model parameters to assess their performance.

Keywords Hamilton–Jacobi–Bellman equations · Innovation dynamics · Stochastic games · Finite difference approximations · Collocation method · Stabilized finite element method · Least-squares finite element method · Adaptive mesh refinement

1 Introduction

We study the system of Hamilton–Jacobi–Bellman (HJB) equations
\[ rV_1(x) = \sup_I \mathcal{L}_1(V_1, V_2, x, I) + f_1(x) \quad x \in \mathbb{R}, \]
\[ rV_2(x, y) = \mathcal{L}_2(V_2, x, y) + f_2(x) \quad x \in \mathbb{R}, \ y \geq 0, \]
where \( r > 0 \) is a constant (the so-called discount rate) and the operators \( \mathcal{L}_1, \mathcal{L}_2 \) are defined as
\[ \mathcal{L}_1(V_1, V_2, x, I) := b_1(x, I) \partial_x V_1(x) - p_0(x)V_1(x) + p_2(I)(V_2(x, 0) - V_1(x)). \]
\[
L_2(V_2, x, y) := \frac{1}{2} \sigma^2(y) \partial_{yy}^2 V_2(x, y) + b_2(x, y) \partial_y V_2(x, y) + \mu(y) \partial_y V_2(x, y) - p_0(x) V_2(x, y).
\] 

The terms \( f_1 \) and \( f_2 \) in (1.1)–(1.2) as well as \( b_1, b_2, p_0, p_2, \sigma, \) and \( \mu \) in (1.3)–(1.4) are real-valued continuous functions specified in Table 2 below.

The HJB system (1.1)–(1.2), which can be classified as a fully-nonlinear degenerate second-order partial differential equation (PDE) in non-divergence form, was derived in [9] to study optimal innovation investment strategies of a monopoly firm under financial constraints, i.e., that is facing a bankruptcy risk. 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 \( x = x(t) \) and the innovation investment \( I \equiv I(t) \) at time \( t \geq 0 \), the monopoly firm may enter three different modes of production \( m = 0, 1, 2 \) with certain probability rates that are characterized by a Markov process \( m := \{ m_t \in [0, 1, 2] : t \geq 0 \} \) with \( m_0 = 1 \), and for every \( t \geq 0 \)

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

where \( p_0 \geq 0 \) is the bankruptcy rate and \( p_2 \geq 0 \) is the innovation rate. For every mode \( i \in \{0, 1, 2\} \), it holds that \( \sum_{j=0}^2 \mathbb{P}[m_{t+dt} = j | m_t = i] = 1 \).

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 an old product on an established market and is developing a new product with a suitable amount of investment \( I \). The latter decreases firm’s liquidity, but allows it to enter the new market and to move from selling only the old product \((m = 1)\) to selling both the old and the new product on the 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 respective mode \( m = 1 \) or \( m = 2 \) and enters the mode \( m = 0 \), where the value \( V_0 = 0 \) is constant. 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 Eqs. (1.1–1.2) we refer to [9].

In general, the solutions to (1.1)–(1.2) do not exist in the classical sense but can be defined in the sense of viscosity solutions, cf., [14, 28, 33]. The existence of the unique viscosity solution of a general HJB equation written in the form

\[
F(D^2 V, \nabla V, V, x) = 0 \quad x \in \mathbb{R}^d,
\]

is guaranteed provided that the functional \( F \) is semicontinuous and satisfies a monotonicity condition, i.e., it holds for \( A, A' \in \mathbb{R}^{d \times d}, A \neq A', p \in \mathbb{R}^d, x \in \mathbb{R}^d, d \geq 1 \) that

\[
F(A, p, q, x) \leq F(A', p, q', x),
\]

whenever \( A - A' \) is negative semidefinite and \( q \geq q' \), see [18] and [33, Chapter 5].

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

\[
F(A, p, q, (x, y)) = \frac{1}{2} \begin{pmatrix}
0 & 0 \\
0 & \sigma^2(y)
\end{pmatrix} : A + (b_2(x, y), \mu(y)) \cdot p - (p_0(x) + r)q + f_2(x),
\]
where for two $d \times d$ matrices $A := \{a_{i,j}\}_{i,j}, B := \{b_{i,j}\}_{i,j}$ we denote $A : B := \sum_{i,j}^d a_{i,j} b_{i,j}$.

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

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

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

which is monotone since $r > 0$ and $p_0, p_2 \geq 0$. Formally, one can equivalently formulate the Hamiltonian using $I^{opt} := \arg\max_I \{ b_1(x, I) p - (p_0(x) + p_2(I) + r) q + p_2(I) V_2(x, 0) \}$ as

$$F(0, p, q, x) = \left\{ b_1(x, I^{opt}) p - (p_0(x) + p_2(I^{opt}) + r) q + p_2(I^{opt}) V_2(x, 0) \right\} + f_1(x).$$

For the considered model parameters (cf., Table 2) $F$ is continuous, except for $p = 0$; the singular case $p = 0$ can be dealt with by imposing an additional condition $I(x) \leq I_{max}$, $x \in \mathbb{R}$ for some $0 < I_{max} < \infty$.

Apart from theoretical considerations, the monotonicity condition (1.7) is also required for the convergence of monotone and consistent numerical schemes, cf., [1] and the recent overview paper [27]. The finite difference schemes are among the most established schemes for the approximation of HJB equations. Monotone finite difference schemes can be constructed by appropriately choosing the finite difference stencil, for instance by the upwind scheme as the references therein), where a monotone finite difference method for the solution of the HJB system is analyzed using the Barles–Souganidis framework. In contrast to the regime switching problem the HJB system (1.1) is decoupled but the respective equations are posed on different state spaces. Furthermore, whereas the generator of the regime switching process in [25] is constant over time, in our setting the switching rate between modes is a function of the state and of the control, in general.
In the present work we compare different discretization methods for the approximation of the HJB Eqs. (1.1)–(1.2). We note that the Eq. (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 $m = 1$. We consider a stabilized finite element method (FEM) for the approximation of the linear degenerate Eq. (1.2). Furthermore, we propose two adaptive finite element discretization strategies for the solution of the (nonlinear) HJB Eq. (1.1); the first approach employs a stabilized finite element approximation and the second one is based on an adaptive least-squares finite element method (LSFEM). The nonlinear algebraic system that results from the discretization of (1.1) is solved iteratively using the well-known policy iteration algorithm, see [16,Chapter 4] and [5]. We compare the proposed finite element methods with alternative discretization approaches: the Chebyshev collocation method and the finite difference method. We discuss relevant implementation details of the considered numerical schemes. In particular, the convergence of the policy iteration for the discrete approximation of (1.1) proves to be problematic in certain scenarios; we discuss suitable modification of the considered numerical algorithms which improve the convergence behavior of the policy iteration.

In addition to the above schemes that approximate the problem (1.1) directly, we propose a numerical approach where (1.1) is interpreted as a steady state of the corresponding time-dependent problem. The time-dependent problem is solved using the policy iteration where the subproblems are discretized in time using approach that combines a total-variation diminishing time discretization with a WENO spatial-discretization of the time dependent problem. The spatial discretisation is based on the upwind WENO5 scheme proposed in [23], [19]. The resulting finite-dimensional system of ordinary differential equations is approximated by a strong stability preserving time-discretization scheme, cf. [30], which is used to compute the steady state of the discrete problem.

The paper is organized as follows. In Sect. 2, we give details on the notation and the considered parameters in (1.1)–(1.2), and discuss analytical properties which are useful to deduce boundary conditions that are required in the considered numerical approximations. In Sect. 3, we discuss the discretization of (1.1)–(1.2) by the collocation method, Sect. 4 is dedicated to the finite difference approximation. The WENO scheme is introduced in Sect. 5. In Sect. 6, we present the stabilized FEM method and the LSFEM method as well as the corresponding adaptive mesh refinement algorithms. We discuss the performance of the considered numerical schemes in Sect. 7.

## 2 Preliminaries

### 2.1 Model Parameters

The functions in (1.1)–(1.4) are defined in Table 2 and the values of the constants $\nu_1, \nu_2, \tilde{\alpha}_n, \bar{\alpha}_n, \eta, \delta, \sigma, r, \gamma_I, \xi$ in these expressions are fixed throughout the paper, see Table 1. We compute the numerical solution for different values of the parameters $\alpha_o, \gamma_B$; we consider the following three Scenarios: Scenario 1 with $(\alpha_o = 0.8, \gamma_B = 0.05)$, Scenario 2 with $(\alpha_o = 0.8, \gamma_B = 0.005)$, and Scenario 3 with $(\alpha_o = 1.0, \gamma_B = 0.05)$; the parameters in the considered Scenarios can be interpreted as follows: $\alpha_o$ is the established market size and $\gamma_B$ is the bankruptcy parameter. Furthermore, the state variable $y \equiv y(t)$ is interpreted as the evolving size of the market demand for the new product.
Table 1 Parameters of the monopoly model

| Param. | Value | Description | Param. | Value | Description |
|--------|-------|-------------|--------|-------|-------------|
| \( v_2 \) | 0.2   | Post-innovation dividend rate | \( \delta \) | 1.55  | Adjustment speed of \( y \) |
| \( v_1 \) | 0.0   | Pre-innovation dividend rate  | \( \sigma \) | 0.1   | Volatility of \( y \) |
| \( \bar{\alpha}_n \) | 0.8   | Avg. long-run size of the new market | \( r \) | 0.02  | Discount rate |
| \( \bar{\alpha}_m \) | 0.6   | Minimal size of the new market | \( \gamma_I \) | 0.1   | Innovation parameter |
| \( \eta \) | 0.5   | Degree of horiz. product diff. | \( \xi \) | 0.025 | Investment cost parameter |

Table 2 Coefficients of the HJBs

| Func. | Expression | Description |
|-------|------------|-------------|
| \( p_0(x) \) | \( \gamma_B \max\{0, -x\} \) | Bankruptcy rate |
| \( p_2(I) \) | \( \gamma_I I \) | Innovation rate |
| \( f_m(x) \) | \( \nu_m \max\{0, x\} \) | Pre/post-innovation dividends |
| \( \pi_1 \) | \( \frac{1}{4} \alpha_o^2 \) | Pre-innovation profit |
| \( \pi_2(y) \) | \( \frac{(\bar{\alpha}_n + y)^2 + \alpha_o^2 - 2\eta(\bar{\alpha}_n + y)\alpha_o}{4 - 4\eta^2} \) | 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 |
| \( \mu(y) \) | \( \delta(\bar{\alpha}_n - y) \) | Demand trend for the new market |
| \( \sigma(y) \) | \( \sigma y \) | 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 sufficiently large finite domains and deduce suitable boundary conditions from the considerations below. Regarding (1.5), the probability of bankruptcy, i.e., \( V_1 = V_0 \) or \( V_2 = V_0 \) is increasing when \( x \to -\infty \), thus at the left end of the domain, we impose a zero Dirichlet boundary condition to \( V_1 \) while at the right end, the solution can be characterized by the following result from [9, Proposition 1].

Proposition 2.1 Assume that \( r > v_1 \). For every \( x, y \geq 0 \) the value in mode \( m = 2 \) is given by

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

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

Furthermore, it holds for every \( x \geq \bar{x} := \max \left\{ \frac{2\xi |I^\text{nc}|^2 - \alpha_o^2}{4(r - v_1)}, 0 \right\} \) with \( I^\text{nc} := \sqrt{\frac{r^2}{\nu_I^2} + \frac{2rc}{\xi} - \frac{a_o^2}{2\xi} - \frac{r}{\nu_I}} \)

(i) The optimal control is constant over time, that is \( I = I^\text{nc} \);
(ii) The value in mode $m = 1$ is given by

$$V_1(x) = x + c - \frac{\xi}{\gamma^l} I^{nc}. \quad (2.2)$$

### 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 approximates $V_1(x)$ and $V_2(x, y)$ by 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 (3.1)–(3.2) are used.

#### 3.1 Rescaling Transformation

A modification of the differential Eqs. (1.1)–(1.2) on an infinite domain results into equivalent differential equations on finite domains. In the manner of [9], we introduce the function

$$z : \mathbb{R} \to (0, 1), \quad x \mapsto z(x) = \frac{1}{1 + \exp(-x/2)}, \quad (3.1)$$

which transforms $\mathbb{R}$ into $(0, 1)$. The inverse transformation of (3.1) reads

$$x : (0, 1) \to \mathbb{R}, \quad z \mapsto x(z) = 2 \ln \left( \frac{z}{1 - z} \right). \quad (3.2)$$

Thanks to the transformation (3.2), we can work on the bounded interval $(0, 1)$. However, the transformation (3.2) is still undefined at $z = 0$ and $z = 1$. Thus, we make an approximation of $(0, 1)$ by cutting off the left and right ends of the interval and work in $[\tilde{z}, \bar{z}]$. Analogously, we perform the computations for $y \in [\tilde{y}, \bar{y}]$ instead of $y \geq 0$.

We set $U(z) := V_1(x(z))$ and $V(z, y) := V_2(x(z), y)$. The inverse transformation (3.2) implies that $\partial_z V_1 = \frac{1}{2} z (1 - z) \partial_z U$ and as a result the differential equations in (1.1)–(1.2) are transformed into

$$rU(z) = \sup_I L_1(U, V, z, I) + f_1(x(z)) \quad z \in [\tilde{z}, \bar{z}]; \quad (3.3)$$

$$rV(z, y) = L_2(V, z, y) + f_2(x(z)) \quad z \in [\tilde{z}, \bar{z}], \quad y \in [\tilde{y}, \bar{y}]; \quad (3.4)$$

where

$$L_1(U, V, z, I) := \frac{1}{2} z (1 - z) b_1(x(z), I) \partial_z U(z) - p_0(x(z)) U(z) + p_2(I) (V(z, 0) - U(z)), \quad (3.5)$$
\[ L_2(V, z, y) := \frac{1}{2} \sigma^2(y) \partial^2_{yy} V_2(z, y) + \frac{1}{2} z(1 - z)b_2(x(z), y) \partial_z V(z, y) + \mu(y) \partial_y V(z, y) - p_0(x(z)) V(z, y). \] (3.6)

### 3.2 Discretization of the Value in Post–Innovation Mode, \( m = 2 \)

In a given state space \([\bar{z}, \bar{z}] \times [\bar{y}, \bar{y}]\), we first construct a grid of collocation nodes \( \mathcal{N} := \mathcal{N}_x \times \mathcal{N}_y \), where \( \mathcal{N}_x := \{z^i\}_{i=1, \ldots, n_z} \) and \( \mathcal{N}_y := \{y^j\}_{j=1, \ldots, n_y} \), \( z^i \) and \( y^j \) are defined as

\[
z^i = \frac{\bar{z} + z}{2} + \frac{z - \bar{z}}{2} \cos \left( \frac{(n_z - i + 0.5) \pi}{n_z} \right),
\] (3.7)

\[
y^j = \frac{\bar{y} + y}{2} + \frac{y - \bar{y}}{2} \cos \left( \frac{(n_y - j + 0.5) \pi}{n_y} \right). \] (3.8)

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

\[
\phi_{i, j}(z, y) = T_{i-1} \left( \frac{2(z - \bar{z})}{\bar{z} - \bar{z}} - 1 \right) T_{j-1} \left( \frac{2(y - \bar{y})}{\bar{y} - \bar{y}} - 1 \right), \] (3.9)

where \( T_k : t \mapsto T_k(t) \) is the Chebyshev polynomial of degree \( k \) defined for any \( t \in [0, 1] \). For all \( z \in [0.5, 1) \) and \( y \geq 0 \), the value function in mode \( m = 2 \) is given by (2.1). Our calculation is carried out only in the space \([\bar{z}, 0.5] \times [0, \bar{y}]\). In order to make sure the calculated value function is continuous at \( z = 0.5 \), that corresponds to \( x = 0 \), we specify further that

\[
z^i = \begin{cases} 
0.5 + \frac{z - \bar{z}}{2} + \frac{0.5 - \bar{z}}{2} \cos \left( \frac{(n_z - i + 0.5) \pi}{n_z} \right) & 1 \leq i \leq n_z - 1, \\
0.5 & i = n_z.
\end{cases} \] (3.10)

The expansion of \( V_2 \) as a Chebyshev series is given by

\[
\hat{V}_2(z, y) := \sum_{i=1}^{n_z} \sum_{j=1}^{n_y} v_{i, j} \phi_{i, j}(z, y) = \hat{v}^\top \phi(z, y), \ z \in [\bar{z}, 0.5],
\] (3.11)

where \( \hat{v} := \{\bar{v}_k : k = 1, \ldots, n_z n_y\} \) and \( \phi := \{\phi_k : k = 1, \ldots, n_z n_y\} \) such that \( \bar{v}_k := v_{i, j} \) and \( \phi_k(z, y) := \phi_{i, j}(z, y) \) with \( k = (i - 1)n_z + j \) for \( i \in \{1, \ldots, n_z\} \) and \( j \in \{1, \ldots, n_y\} \). The solution \( V(z, y) \) of (3.4) is substituted by the expression (3.11). Thus the approximation of the value function \( V \) is equivalent to determine the components \( \bar{v}_k \) of the vector \( \hat{v} \) such that (3.4) holds on the collocation nodes \( \{z^i, y^j\} \) with \( i, j \in \{1, \ldots, n_z - 1\} \times \{1, \ldots, n_y\} \).

The approximation of \( V_2 \) is therefore defined by

\[
\overline{V}_2(z, y) := \begin{cases} 
\hat{V}_2(z, y), & z \in [\bar{z}, 0.5], \\
V_2(x(z), y), & z \in [0.5, \bar{z}].
\end{cases} \] (3.12)

where \( x : z \mapsto x(z) \) is given by (3.2) and \( V_2 : (x, y) \mapsto V_2(x, y) \) is given by the formula (2.1). Furthermore, we impose \( \overline{V}_2(0.5, y^j) = V_2(x(0.5), y^j) \) to enforce the continuity of (3.12) 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 \( i \in \{1, \ldots, n_z - 1\} \) and \( j \in \{1, \ldots, n_y\} \) we introduce four \((n_z - 1)n_y \times n_z n_y\) matrices \( B_s, B^{yy}, B^{yy}, \) and \( B^s \) with entries

\[
B_{s,k} = \phi_k(z^i, y^j), \quad B^{yy}_{s,k} = \partial_{y} \phi_k(z^i, y^j), \quad B^{yy}_{s,k} = \partial_{yy}^2 \phi_k(z^i, y^j), \quad B^s_{s,k} = \partial_{z} \phi_k(z^i, y^j),
\]

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 \( N \) that are not on the boundary of \( z^n = 0.5 \). For each node \( \{z^i, y^j\} \) with \( i \in \{1, \ldots, n_z - 1\} \) and \( j \in \{1, \ldots, n_y\} \), we define the five column vectors

\[
\tilde{\sigma} := (\tilde{\sigma}_s), \quad \tilde{\mu} := (\tilde{\mu}_s), \quad \bar{b}_s := (\bar{b}_{2,s}), \quad \bar{p}_0 := (\bar{p}_{0,s}), \quad \bar{f}_2 := (\bar{f}_{2,s}).
\]

as follows:

\[
\begin{align*}
\tilde{\sigma}_s &= \mu(y^j), \\
\tilde{\sigma}_s &= \frac{1}{2} \sigma(y^j)^2, \\
\bar{b}_{2,s} &= \frac{1}{2} z^i (1 - z^i) b_2(x(z^i), y^j), \\
\bar{f}_{2,s} &= f_2(x(z^i)), \\
\bar{p}_{0,s} &= p_0(x(z^i)),
\end{align*}
\]

and \( s = (i - 1)n_z + j \in \{1, \ldots, (n_z - 1)n_y\} \). For the \( s \) nodes, the resulting linear system of equations reads

\[
r B \bar{v} = \tilde{\sigma}^T B^{yy} \bar{v} + \tilde{\mu}^T B^{yy} \bar{v} + \bar{b}_2^T \bar{B} \bar{v} - \bar{p}_{0}^T \bar{B} \bar{v} + \bar{f}_2.
\]

For the other \( n_y \) nodes, it reads

\[
\bar{v} \cdot \phi(0.5, y^j) = V_2(0, y^j), \quad j \in \{1, \ldots, n_y\}.
\]

### 3.3 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 a one-dimensional problem defined on \( \mathbb{R} \). The optimal control

\[
I^* = \arg \max_{\tilde{I}} \mathcal{L}_1(V_1, V_2, x, \tilde{I})
\]

can be expressed explicitly as

\[
I^*(x) = \frac{\gamma_I(V_2(x, 0) - V_1(x))}{\xi \partial_x V_1(x)}.
\]

Similarly as for the mode \( m = 2 \), we employ the transformation (3.2). After the transformation, the optimal control in (3.14) can be rewritten as

\[
\tilde{I}^*(z) = I^*(x(z)) = \frac{2 \gamma_I(V_2(x(z), 0) - V_1(x(z)))}{\xi z(1 - z) \partial_z V_1(x(z))},
\]

for all \( z \) where \( \partial_z V_1(x(z)) \) exists. We proceed with the same collocation method as in mode \( m = 2 \), but on the one dimensional state space of \( z \in [\bar{z}, \bar{z}] \). However, in mode \( m = 1 \) for some values of the model parameters there potentially exists a kink in \( V_1(x(z)) \) at \( z = 0.5 \) and consequently the control function \( \tilde{I}^*(z) \) is discontinuous at \( z = 0.5 \) \( (x = 0) \). In [9,Proposition 2 and Corollary 3] an analytical characterization of scenarios where such a jump in the optimal control function occurs is provided. In particular, it is shown that the following three cases can occur.

(i) For \( \gamma_I \in [\gamma_I, \gamma_I] \) with

\[
\gamma_I = \frac{r \alpha_o \sqrt{2 \xi}}{2 \gamma c - \alpha_o^2}, \quad \gamma_I = \frac{(r + \gamma \beta \gamma) \alpha_o \sqrt{2 \xi}}{2 \gamma c - \alpha_o^2}
\]
the state $x = 0$ is a (stable) fixed point under the dynamics with optimal investment in mode $m_1$, i.e. $b_1(0, I^*(0)) = 0$ and both intervals $(-\infty, 0]$ and $[0, \infty)$ are invariant under the dynamics $\dot{x} = b_1(x, I^*(x))$. For $\gamma_I \in (\gamma_f, \overline{\gamma}_f)$ the optimal control is discontinuous at $x = 0$ with $\lim_{x \to 0^-} I^*(x) < \lim_{x \to 0^+} I^*(x)$.

(ii) If $\gamma_I < \gamma_f$, then $b_1(0, I^*(0)) > 0$. Hence, the interval $[0, \infty)$ is invariant under the optimized state dynamics, but $(-\infty, 0]$ is not.

(iii) If $\gamma_I > \overline{\gamma}_f$, then $b_1(0, I^*(0)) < 0$. Hence, the interval $(-\infty, 0]$ is invariant under the optimized state dynamics, but $[0, \infty)$ is not.

In the case (i) the discontinuity of the control at $x = 0$ is guaranteed, whereas in cases (ii) and (iii) the theoretical results do not allow for a clear conclusion with respect to the continuity of optimal investment at $x = 0$. Due to the polynomial approximation of the value function in our collocation approach, by definition the approximate value function is smooth on the entire considered state space and a value function with a kink cannot be generated. Hence, based on the considerations above, we apply the collocation scheme separately on the intervals $(-\infty, 0]$ and $[0, +\infty)$. We define the approximation of $V_1$ as follows

$$V_1(z) := \begin{cases} \tilde{V}_1(z), & z \in [\bar{z}, 0.5], \\ \tilde{V}_1(z), & z \in (0.5, \bar{z}], \end{cases}$$

(3.16)

where $\tilde{V}_1$ and $\tilde{V}_1$ are respectively the approximation of $V_1$ by Chebyshev expansion in $[\bar{z}, 0.5]$ and $[0.5, \bar{z}]$, that is for $x \leq 0$ and $x \geq 0$. Depending on which of the three listed cases applies boundary conditions for one or both of the collocation schemes are inserted. In particular, in the case (i) the value at $z = 0.5$ is calculated as

$$V_1(x(0.5)) = V_1(0) = \int_0^\infty e^{-(r+\gamma_I I^*(0.5))t} \gamma_I I^*(0.5) V_2(0, 0) dt = \frac{\gamma_I \alpha_o V_2(0, 0)}{r \sqrt{2\xi} + \gamma_I \alpha_o},$$

(3.17)

and in both collocations, similar to the procedure in mode $m_2$, one node is set at $z = 0.5$ and the condition $\tilde{V}_1(x(0.5)) = V_1(x(0.5))$ respectively $\tilde{V}_1(x(0.5)) = V_1(x(0.5))$ is included. In case (ii) the collocation is first applied to $[0.5, \bar{z}]$ without any boundary condition, i.e. all $n_z$ nodes in the interior of the interval and all conditions determined by the HJB equation at the node. This yields $\tilde{V}_1(z)$. In a second step the collocation is applied to $[\bar{z}, 0.5]$ with the boundary condition $\tilde{V}_1(0.5) = \tilde{V}_1(0.5)$. Analogous for case (iii).

We illustrate the application of the collocation method for scenario (iii), the other scenarios are analogous. In the case (iii), the interval $[\bar{z}, 0.5]$ is invariant under the dynamics and the expansion of $V_1$ as a Chebyshev series on this interval is given by

$$\tilde{V}_1(z) := \sum_{i=1}^{n_z} u_i \phi_i(z, 0) = \tilde{u}^\top \phi(z, 0), \quad z \in [\bar{z}, 0.5],$$

(3.18)

where $\tilde{u} := \{u_i : i = 1, \ldots, n_z\}$ is a column vector of coefficients. The value function $U(z)$ in (3.3) is then substituted by the expression (3.18) and the approximation is found by determining the components $u_i$ of the vector $\tilde{u}$ such that (3.3) holds on the collocation nodes $\{z^i : i = 1, \ldots, n_z\}$. To solve the resulting nonlinear problem we use a policy iteration procedure, see for instance [2, 16]. In particular, we consider a sequence of vectors $\tilde{u}^{(n-1)}$, with $n \in \mathbb{N}$ as the iteration counter. By (3.15), we derive the discrete optimal control $I^{(n-1)} := \ldots$
\{I_i^{(n-1)} : i = 1, \ldots, n_z\} at the stage \(n \in \mathbb{N}\) of the iteration by

\[
\overline{T}^{(n-1)} = \frac{\gamma_I (\mathbb{B} \tilde{u}, - \mathbb{B} \tilde{u}^{(n-1)})}{\xi \bar{\lambda}^T \mathbb{B} \tilde{z} \tilde{u}^{(n-1)}},
\]

(3.19)

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

\[
\mathbb{B}_{i,k} = \phi_k(z^i, 0), \quad \mathbb{B}^z_{i,k} = \partial_z \phi_k(z^i, 0), \quad i \in \{1, \ldots, n_z\}.
\]

Overall, the numerical algorithm that couples the Chebyshev collocation method and the policy iteration can be summarized as follows:

**Algorithm 3.1** (Collocation method) **Start with an initial guess for \(\tilde{u}^{(0)}\) (see below) and calculate the control \(\overline{T}^{(0)}\) according to (3.19).** Then, for every \(n \in \mathbb{N}\), we proceed along the following two steps iteration:

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

\[
r \mathbb{B} u^{(n)} = \tilde{b}_1^T (\overline{T}^{(n-1)}) \mathbb{B} \tilde{z} \tilde{u}^{(n)} - (\tilde{p}_0 + \tilde{p}_2 (\overline{T}^{(n-1)}))^T \mathbb{B} u^{(n)} + \tilde{p}_2 (\overline{T}^{(n-1)}) \mathbb{B} \tilde{u}, + f_1.
\]

**Policy improvement:** Update the optimal control by (cf. (3.19))

\[
\overline{T}^{(n)} = \frac{\gamma_I (\mathbb{B} \tilde{u}, - \mathbb{B} \tilde{u}^{(n)})}{\xi \bar{\lambda}^T \mathbb{B} \tilde{z} \tilde{u}^{(n)}}.
\]

**Stopping criterion:** Repeat the loop until

\[
\left| \frac{r \mathbb{B} u^{(n)} - \tilde{b}_1^T (I^{(n)}) \mathbb{B} \tilde{z} \tilde{u}^{(n)} + (\tilde{p}_0 + \tilde{p}_2 (I^{(n)}))^T \mathbb{B} u^{(n)} - \tilde{p}_2 (I^{(n)}) \mathbb{B} \tilde{u}, - f_1}{\mathbb{B} \tilde{u}^{(n)}} \right|_\infty \leq \epsilon.
\]

In order to determine the initial guess for \(\tilde{u}^{(0)}\) we employ a continuation approach. For \(\gamma_I = 0\) the value function is linear and can be determined in closed form (see [9]). The corresponding coefficient vector \(\tilde{u}\) is found by interpolation of this function at the nodes \(\{z^i : i = 1, \ldots, n_z\}\). The parameter \(\gamma_I\) is then stepwise increased till its target value is reached, where in each step the coefficient vector found in the previous step is used as the initial guess \(\tilde{u}^{(0)}\). The numerical calculation of \(\tilde{V}_1(z)\) with \(z \in (0.5, \bar{z}]\) is similar to the calculation of \(\tilde{V}_1(z)\) except we further enforce \(\tilde{V}_1(0.5) = \tilde{V}_1(0.5)\) so that the continuity of (3.16) is preserved at \(z = 0.5\).

### 3.4 Numerical Results

To illustrate the results obtained using the collocation method, we show at the bottom part of Fig. 1 the value and feedback functions obtained for the Scenarios 1–3. The number of collocation nodes for these calculations are \(n_z = 40\) for the collocation on the domain \(z \in [0.5, \bar{z}]\) (i.e. \(x \geq 0\)) and \(n_z = 25\) on the domain \(z \in [\bar{z}, 0.5]\) (i.e. \(x \leq 0\)).

Note that in light of the considered parameter values we have \(\gamma_I \in (\gamma_I, \bar{\gamma}_I)\) in Scenario 1, whereas \(\gamma_I > \bar{\gamma}_I\) in Scenario 2 and \(\gamma_I < \bar{\gamma}_I\) in Scenario 3. Hence, in Scenario 1 \(x = 0\) is a stable fixed point of the state dynamics under optimal investment and we impose boundary conditions \(\tilde{V}_1(0.5) = \tilde{V}_1(0.5) = V_1(0)\), where \(V_1(0)\) is given by (3.17), for both the collocations run on \(z \in [\bar{z}, 0.5]\) and \(z \in [0.5, \bar{z}]\). In the other two scenarios no boundary conditions were imposed on the domain \(z \in [\bar{z}, 0.5]\) (Scenario 2) respectively on the domain \(z \in [0.5, \bar{z}]\) (Scenario 3) and, following our description above, the obtained values of \(\tilde{V}_1(0.5)\) respectively \(\tilde{V}_1(0.5)\) were used as the boundary conditions on the second domain. For Scenario 2 we have
calculated the value and control functions not only for our standard value $z = z(-10)$, but also for $z = z(-20)$. The reason for also considering a smaller lower bound of the domain is that $\hat{V}_1(z(-10))$ still has a substantial positive value, whereas it is close to zero at $z(-20)$. Comparing the dashed green lines ($z = z(-10)$) and the solid green lines ($z = z(-20)$) however shows that the value functions calculated under these two domain specifications are indistinguishable, and also the optimal control determined under $z = z(-10)$ is very close to that obtained for $z = z(-20)$. Although hard to discern from the figure in the right lower panel, we like to point out that not only for Scenario 1, where according to our theoretical results we know that a jump in the control occurs at $x = 0$, but also for Scenario 3 the calculated optimal investment function exhibits a jump at $x = 0$.

Considering still the lower panels of Fig. 1, we observe that the optimal investment function in Scenario 2 with $z = z(-20)$ crosses the line $I = 0$ at approximately $x = -14$. This indicates that for the calculated value functions $\hat{V}_2(z, 0) - \hat{V}_1(z) < 0$ holds for $z$ close to the lower bound $\bar{z}$. Since the value after successful innovation cannot be lower than that in mode $m_1$ before the innovation, this is a numerical artifact suggesting that the value of $\hat{V}_1$ is overestimated for $z$ close to $\bar{z}$. To address this issue, we show in the upper part of Fig. 1 results for Scenario 2 on different domains where the boundary condition $\hat{V}_1(z) = 0$ is imposed; for comparison the results computed with the finite difference method are shown in Fig. 3. For $z = z(-20)$ imposing the boundary condition yields an optimal investment function which for values of $z(-10) \leq z \leq 0.5$ closely resembles the one calculated without a boundary condition. Furthermore, the domain on which investment is positive is substantially larger if the boundary condition is imposed. If the boundary condition is however imposed at $z = z(-15)$ or $z = z(-10)$ the generated investment functions are more volatile and in

![Fig. 1](image-url)  
*Fig. 1* Numerical solutions for the collocation method. Top: Effect of the domain on $V_1$ (left) and $I$ (right) for Scenario 2 for lower bounds $\tilde{x} = -20, -15, -10$ with boundary condition $V_1(\tilde{x}) = 0$; the dashed green lines correspond to the solution computed without state space transformation for $\tilde{x} = -20$. Bottom: The numerical solutions $V_1$ (left) and $I$ (right) computed on $(-10, 10)$ for Scenario 1 (red), 2 (dashed green) and 3 (blue) and on $(-20, 20)$ for Scenario 2 (solid green).
particular for $z = z(-10)$ yield unsatisfactory results. This can be attributed to the fact that the distribution of the collocation points is getting coarser away from $x(z = 0.5) = 0$ due to the transformation (3.1), (3.2) which results in low accuracy of the scheme for large domains. Nevertheless, without the transformation the policy iteration does not converge, unless a good initial guess for the policy iteration is used (see next paragraph below). Furthermore, we observe in Fig. 1 that the results get worse if a homogeneous Dirichlet boundary condition is imposed since in Scenario 2 the actual value of $V_1(x)$ at $x = -10$ and $x = -15$ is still substantially larger than 0.

In order to improve the performance of the collocation method in Scenario 2, we have computed the solution with the collocation method for this case without applying the state space transformation (3.1), (3.2) (also imposing the boundary condition $V_1(-20) = 0$). To improve the convergence of the policy iteration we employ a preconditioning approach: we use the approximate value function calculated with the state space transformation as the initial guess ($u^0$ in Algorithm 3.1) in the policy iteration. The preconditioning step solves convergence issues and the computed solution agrees with the results of other methods. The computed value and control are depicted as the dashed green line in the top graphs in Fig. 1. Unlike in the case of the method with transformation, the computed control does not exhibit oscillation on $[-20, -10]$.

4 Discretization by the Finite Difference Method

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 Sect. 6.1 below.

The finite difference method requires the spatial domain to be of finite size. Thus, we consider (1.1) for $x \in (\bar{x}, \tilde{x})$ with $\bar{x} < 0$ and $\tilde{x} \geq \tilde{x}$, see Proposition 2.1. We impose the following Dirichlet boundary conditions

$$V_1(\bar{x}) = 0 \quad \text{and} \quad V_1(\tilde{x}) = \tilde{x} + c - \frac{\xi}{\gamma I} I^{\text{inc}}, \quad \tilde{x} \geq \tilde{x}. \quad (4.1)$$

We introduce an equidistant grid $\Pi h_1 := \{x_0, \ldots, x_{M-1}\}$ with mesh size $h := (\tilde{x} - \bar{x})/(M - 1)$, where

$$x_i = \bar{x} + ih \quad i = 0, 1, \ldots, M - 1. \quad (4.2)$$

4.1 Upwind Finite Difference Approximation

For $i = 0, \ldots, M - 1$, we denote $b_i^j(I) := b_1(x_i, I), p_0^i := p_0(x_i), p_2 := p_2(I)$, and $f_i^j := f_1(x_i)$. The finite difference approximation of the value function $V_1$ on the grid (4.2) is denoted by $V_1^i \approx V_1(x_i)$.

We approximate $\partial_x V_1$ by backward, central, or forward differences, which are denoted by $D^-x V_1$, $D^0x V_1$, and $D^+x V_1$, respectively, and defined by

$$D^-x V_1^i := \frac{V_1^i - V_1^{i-1}}{h}, \quad D^0x V_1^i := \frac{V_1^{i+1} - V_1^{i-1}}{2h}, \quad D^+x V_1^i := \frac{V_1^{i+1} - V_1^i}{h}. \quad (4.3)$$
We also introduce the upwind finite difference approximation $D^\pm_x V^i_1 \approx \partial_x V_1(x_i)$ defined by

$$
D^\pm_x V^i_1 := \begin{cases} 
D^+_x V^i_1, & \text{if } b^i_1(I) > 0, \\
D^-_x V^i_1, & \text{if } b^i_1(I) \leq 0.
\end{cases}
$$

Once a numerical approximation of the value function $V_2$ is computed we denote its values on the grid points at $y = 0$ by $V^i_2 \approx V_2(x_i, 0)$, $i = 0, \ldots, M - 1$ and set the vector $\vec{V}_2 := (V^1_2, \ldots, V^{M-2}_2)$.

To obtain the finite difference approximation of $V_1$, we replace the derivative in (1.1) by the upwind finite difference (4.4). We set $\vec{V}_1 := (V^1_1, \ldots, V^{M-2}_1)$ and by (4.1) impose

$$
V^0_1 = 0 \quad \text{and} \quad V^M_1 = \bar{x} + c - \frac{\xi}{\gamma_l} I^{nc}.
$$

To obtain the finite difference approximation of the optimal control $I^*$, we replace the derivative in (3.14) by the central finite difference $D^0_x V^i_1$, see (4.3). We set $\vec{I} := (I^1, \ldots, I^{M-2})$ with

$$
I^i = \frac{\gamma_l (V^i_2 - V^i_1)}{\xi D^0_x V^i_1}, \quad i = 1, \ldots, M - 2.
$$

Thus, to (1.1), we associate the following discrete system for all nodes $i = 1, \ldots, M - 2$

$$
r V^i_1 = L^M_1 (\vec{V}_1, \vec{V}_2, x_i, \vec{I}) + f^i_1,
$$

where

$$
L^M_1 (\vec{V}_1, \vec{V}_2, x_i, \vec{I}) := b^i_1(I^i) D^\pm_x V^i_1 - p^1_0 V^i_1 + p^2_0 (V^i_2 - V^i_1).
$$

In the discrete system (4.6), the optimal control $\vec{I}$ given by the formula (4.5) depends implicitly on $\vec{V}_1$ and the central difference of $\vec{V}_1$. Analogically to the collocation method, we use the policy iteration procedure to break the implicit character of the system (4.6). Altogether, the upwind scheme and the policy iteration give the following algorithm:

**Algorithm 4.1** (FDM) Set $\vec{I}^{(0)} := I^{nc}$ and for $n = 1, \ldots$ proceed as follows:

*Policy evaluation:* Find $\vec{V}^{(n)}_1 := (V^{1,(n)}, \ldots, V^{M-2,(n)})$, such that

$$
r V^{i,(n)}_1 = L^M_1 (\vec{V}^{(n)}_1, \vec{V}_2, x_i, \vec{I}^{(n-1)}) + f^i_1.
$$

*Policy improvement:* Update the control $\vec{I}^{(n)} := (I^{1,(n)}, \ldots, I^{M-2,(n)})$ with

$$
I^{i,(n)} = \frac{\gamma_l (V^{i,0}_2 - V^{i,(n)}_1)}{\xi D^0_x V^{i,(n)}_1}.
$$

*Stopping criterion:* Repeat the loop until

$$
\max_{i=1,\ldots,M-2} |I^{i,(n)} - I^{i,(n-1)}| \quad \text{and} \quad \max_{i=1,\ldots,M-2} |I^{i,(n)}| \leq 10^{-6}.
$$
4.1.1 Numerical Experiments

To obtain the numerical approximation of $V_1$, we use the values of the numerical approximation of $V_2$ at $y = 0$ computed by the finite element method with mesh size $h = 2^{-11}$ (see Sect. 6.1 below) in all numerical experiments in this section.

Effect of $V_2$ on the solution As mentioned above, we use a value function $V_2$ computed on a (fixed) coarse mesh, i.e. the approximation of $V_2$ is given on $M = 2^{11}$ uniformly distributed grid-points. To compute a value function $V_1$ on a finer mesh, we have to interpolate the available data for $V_2$ over the (finer) mesh on which $V_1$ is computed. Below we examine the effect of different interpolation methods on the numerical solution in Scenario 2. The value function $V_1$ is computed for $x \in (-20, 10)$ using Algorithm 4.1.

We compare the effect of linear interpolation and cubic interpolation (see [20]) of $V_2$ on the control. We observe in Fig. 2 (bottom) that algorithm with linearly interpolated $V_2$ produces a control with irregular oscillations while control obtained with cubic interpolation is smooth; note that in Fig. 2 (top) we display the central difference approximation of the gradient of $V_2$. Although, the different interpolation methods for $V_2$ do not influence the overall convergence of the policy algorithm, it is preferable to avoid the oscillations in the control caused by the linear interpolation of $V_2$. Therefore, we employ cubic interpolation of $V_2$ in the remaining numerical experiments.

Effect of the domain on the solution In this second batch of numerical experiments, we consider three domains $\Omega_{10} := (-10, 10)$, $\Omega_{15} := (-15, 10)$, and $\Omega_{20} := (-20, 10)$.
We made the following observations:

- **Scenario 2.** The solution obtained changes significantly with the domain, see Fig. 3. The reason is that for $x \geq -15$, the value function $V_1(x)$ is far from zero. Thus, if we impose a zero Dirichlet boundary condition at $x = -15$ or above, then the solution obtained deviates significantly from the exact solution.

- **Scenarios 1 and 3.** The solution obtained is not affected by the change of domain or boundary conditions. At $x = -7$, the value function is almost zero, see Fig. 3. Thus, imposing a zero Dirichlet boundary condition at $x = -10$ does not have a significant effect on the solution.

In general, we observe that smaller values of $\gamma_B$ require the use of larger domains.

**Discontinuity of the control** In [9, Proposition 2], the authors argue that the control is discontinuous at $x = 0$ for certain values of the model parameters, cf., also the discussion at the beginning of Sect. 3.3. Our numerical experiments confirm this statement. The numerical experiments for decreasing grids size in Fig. 4 confirm the theoretical prediction, i.e., the control exhibits a discontinuity at $x = 0$ in Scenarios 1 and 3, which is reflected by the growing values of the gradient of the control. In the case of discontinuity the control is not
Lipschitz continuous, which affects the convergence of the policy iteration algorithm; we discuss this issue in more detail in Sect. 4.1.2 below.

4.1.2 Effect of the Discretization of the Control

We examine the effect of different approximations of the optimal control \((3.14)\) on the convergence behavior of the policy iteration algorithm. The computations below were performed on a fixed grid with \(M = 2^{15}\) grid-points. In addition to the central difference approximation \((4.5)\), we consider the backward difference, forward difference and upwind approximation, i.e., the central difference \(D_0^x\) in the denominator of \((4.5)\) is replaced by \(D_-^x\), the forward difference \(D_+^x\), and the upwind difference \(D_\pm^x\), respectively (with the corresponding modification of the formula \((4.9)\) in the policy improvement step of Algorithm 4.1).

We also consider the following alternative to \((4.5)\)

\[
I^i = \frac{\gamma}{2} \left( \frac{V_{i+1,0}^i - V_i^i}{D_-^x V_i^i} + \frac{V_{2,0}^{i,0} - V_i^i}{D_+^x V_i^i} \right),
\]

which is an average of the respective formulas obtained by the forward and backward differences. The averaging has a regularizing effect in the case of discontinuous control (cf. Fig. 4) and improves the convergence behavior even when the control is not Lipschitz continuous.

The convergence plots of the policy iteration with the considered finite difference approximations of the control for Scenarios 1–3 are displayed in Fig. 5. The results can be summarized as follows:

- The upwind finite difference approximation (the results are not displayed in Fig. 5): this discretization yields control that is optimal on the discrete level; nevertheless, we observe that the policy iteration diverges with the upwind discretization.

![Fig. 4: Gradient of the control computed by Algorithm 4.1 for Scenarios 1, 2, and 3 (from left to right)](image)

![Fig. 5: Error evolution of the policy iteration for Scenarios 1, 2, and 3 (from left to right) for difference approximations of the control](image)
- The forward finite difference: the algorithm converges in Scenario 1 and 2 but diverges in Scenario 3.
- The backward finite difference: the algorithm converges in all scenarios, nevertheless one may expect similar problems with convergence as for the forward difference approximation in general (i.e., for different model parameters).
- The central and “averaged” finite differences: in both cases the algorithm converges in every scenario, the “averaged” difference approximation exhibits slightly better convergence behavior overall.

Finally, it is apparent from Fig. 5 that in Scenario 2 (i.e., when there is no discontinuity in the control) the approximation of the control has only minor influence on the convergence behavior of the policy iteration algorithm.

### 4.1.3 Experimental Rate of Convergence

Next, we study the rate of convergence of the finite difference numerical approximation; the numerical solution is computed on $\Omega_1$ in Scenarios 1, 3 and on $\Omega_2$ in Scenario 2. Since no analytical solution is known for the given problem, we determine the rate of convergence by using a reference solution computed on a fine grid.

We compute a sequence of approximations for $\kappa = 0, \ldots, 5$ on uniformly spaced grids with $M(\kappa) = 2^{1+\kappa}$ grid-points denoted as $\bar{V}_{1,M(\kappa)} := (V_{1,M(\kappa)}, \ldots, V_{1,M(\kappa)})$ and $\bar{T}_{M(\kappa)} = (I_{1,M(\kappa)}, \ldots, I_{M(\kappa)})$.

We approximate the exact solution $V_1, I$ by the respective reference solutions $\bar{V}_{1,M(\kappa)}, \bar{T}_{M(\kappa)}$ on the uniformly spaced grid with $M(\kappa) < M_{\text{ref}}$ grid-points. On a reference domain $\Omega_r = (-9, 9)$, the $L^\infty(\Omega_r)$-errors $\|V_1 - \bar{V}_{1,M(\kappa)}\|_{L^\infty(\Omega_r)}$ and $\|I - \bar{T}_{M(\kappa)}\|_{L^\infty(\Omega_r)}$ are computed from piecewise linear interpolation of the numerical solution on the respective finite difference grids.

The rate of convergence in the $L^\infty$-norm (analogically in the $L^2$-norm) is determined as

$$\frac{\ln (\|V_1 - \bar{V}_{1,M(\kappa-1)}\|_{L^\infty(\Omega_r)}) - \ln (\|V_1 - \bar{V}_{1,M(\kappa)}\|_{L^\infty(\Omega_r)})}{\ln (h(\kappa - 1)) - \ln (h(\kappa))}, \quad \kappa = 1, \ldots, 5.$$

The results are displayed in Table 3; we observe that the experimental convergence rate is of order about 1 for the variables $V_1$ and $I$. The convergence rate in the $L^\infty$-norm is reduced to about 1/2 in Scenarios 1 and 3, which reflects the fact that the control is discontinuous. In addition, in Table 4 we display the experimental convergence rates for the “averaged” discretization of the control (4.10) (which has the advantage of a slightly faster convergence of the policy iteration algorithm); the observed convergence rates are similar to those from Table 3, the accuracy of the approximation is slightly worse in Scenario 2.

### 5 Weighted Essentially Non-oscillatory (WENO) Finite–Difference Approximation

In this section we propose an algorithm for the computation of the value function that employs higher-order finite difference discretization in space. The scheme is similar to the policy iteration Algorithm 4.1 with the exception that the value function in the policy evaluation step is obtain as a steady state of a time dependent version of (1.1), i.e., the solution of (1.1) is obtained as $V_1(x) = \lim_{t \to \infty} w(t, x)$ where $w$ solves the transient PDE $\partial_t w + rw = \sup I \mathcal{L}_1(w, V_2, x, I) + f_1(x)$. 

 Springer
Table 3  Experimental rate of convergence of the finite difference method

| Scenario | κ | h(κ) | # iter. | Value function $V_1$ | Optimal control $I$ |
|----------|---|------|--------|----------------------|---------------------|
|          | $\kappa(h)$ |        |        | $L^\infty(\Omega_r)$-error | rate | $L^2(\Omega_r)$-error | rate |
| Scenario 1 | 0 | $1.96 \times 10^{-2}$ | 13 | $2.64 \times 10^{-2}$ | 1.01 | $2.35 \times 10^{-2}$ | 1.01 |
|          | 1 | $9.77 \times 10^{-3}$ | 12 | $1.31 \times 10^{-2}$ | 1.01 | $1.17 \times 10^{-2}$ | 1.01 |
|          | 2 | $4.88 \times 10^{-3}$ | 15 | $6.54 \times 10^{-3}$ | 1.01 | $5.83 \times 10^{-3}$ | 1.01 |
|          | 3 | $2.44 \times 10^{-3}$ | 17 | $3.25 \times 10^{-3}$ | 1.01 | $2.90 \times 10^{-3}$ | 1.01 |
|          | 4 | $1.22 \times 10^{-3}$ | 17 | $1.61 \times 10^{-3}$ | 1.01 | $1.43 \times 10^{-3}$ | 1.01 |
|          | 5 | $6.10 \times 10^{-4}$ | 18 | $7.91 \times 10^{-4}$ | 1.02 | $7.05 \times 10^{-4}$ | 1.02 |
| Scenario 2 | 0 | $2.93 \times 10^{-2}$ | 7 | $2.06 \times 10^{-3}$ | 1.01 | $2.55 \times 10^{-3}$ | 1.01 |
|          | 1 | $1.47 \times 10^{-2}$ | 7 | $1.02 \times 10^{-3}$ | 1.02 | $1.26 \times 10^{-3}$ | 1.01 |
|          | 2 | $7.33 \times 10^{-3}$ | 7 | $4.95 \times 10^{-4}$ | 1.04 | $6.25 \times 10^{-4}$ | 1.01 |
|          | 3 | $3.66 \times 10^{-3}$ | 6 | $2.45 \times 10^{-4}$ | 1.01 | $3.11 \times 10^{-4}$ | 1.01 |
|          | 4 | $1.83 \times 10^{-3}$ | 5 | $1.21 \times 10^{-4}$ | 1.02 | $1.54 \times 10^{-4}$ | 1.01 |
|          | 5 | $9.16 \times 10^{-4}$ | 5 | $5.93 \times 10^{-5}$ | 1.02 | $7.56 \times 10^{-5}$ | 1.02 |
| Scenario 3 | 0 | $1.96 \times 10^{-2}$ | 12 | $2.61 \times 10^{-2}$ | 1.02 | $2.76 \times 10^{-2}$ | 1.01 |
|          | 1 | $9.77 \times 10^{-3}$ | 13 | $1.30 \times 10^{-2}$ | 1.01 | $1.37 \times 10^{-2}$ | 1.01 |
|          | 2 | $4.88 \times 10^{-3}$ | 15 | $6.43 \times 10^{-3}$ | 1.01 | $6.77 \times 10^{-3}$ | 1.01 |
|          | 3 | $2.44 \times 10^{-3}$ | 15 | $3.19 \times 10^{-3}$ | 1.01 | $3.36 \times 10^{-3}$ | 1.01 |
|          | 4 | $1.22 \times 10^{-3}$ | 19 | $1.58 \times 10^{-3}$ | 1.02 | $1.66 \times 10^{-3}$ | 1.02 |
|          | 5 | $6.10 \times 10^{-4}$ | 20 | $7.75 \times 10^{-4}$ | 1.03 | $8.14 \times 10^{-4}$ | 1.03 |
Table 4  Experimental convergence rate for the “averaged” approximation of the control (formula (4.10))

| $\kappa$ | $h(\kappa)$ | # iter. | Value function $V_1$ | Optimal control $I$ |
|----------|-------------|---------|-----------------------|----------------------|
|          |             |         | $L^\infty(\Omega_r)$-error | rate | $L^2(\Omega_r)$-error | rate | $L^\infty(\Omega_r)$-error | rate | $L^2(\Omega_r)$-error | rate |
| Scenario 1 | 0 | $1.96 \times 10^{-2}$ | 14 | $2.47 \times 10^{-2}$ | 1.00 | $2.17 \times 10^{-2}$ | 1.00 | $6.30 \times 10^{-1}$ | 0.44 | $1.64 \times 10^{-1}$ | 0.97 |
|           | 1 | $9.77 \times 10^{-3}$ | 14 | $1.23 \times 10^{-2}$ | 1.00 | $1.08 \times 10^{-2}$ | 1.00 | $4.65 \times 10^{-1}$ | 0.46 | $8.37 \times 10^{-2}$ | 0.98 |
|           | 2 | $4.88 \times 10^{-3}$ | 15 | $6.16 \times 10^{-3}$ | 1.00 | $5.41 \times 10^{-3}$ | 1.00 | $3.39 \times 10^{-1}$ | 0.47 | $4.24 \times 10^{-2}$ | 0.99 |
|           | 3 | $2.44 \times 10^{-3}$ | 15 | $3.07 \times 10^{-3}$ | 1.01 | $2.70 \times 10^{-3}$ | 1.01 | $2.44 \times 10^{-1}$ | 0.48 | $2.14 \times 10^{-2}$ | 1.00 |
|           | 4 | $1.22 \times 10^{-3}$ | 16 | $1.52 \times 10^{-3}$ | 1.01 | $1.34 \times 10^{-3}$ | 1.01 | $1.75 \times 10^{-1}$ | 0.49 | $1.07 \times 10^{-2}$ | 1.01 |
|           | 5 | $6.10 \times 10^{-4}$ | 17 | $7.49 \times 10^{-4}$ | 1.02 | $6.58 \times 10^{-4}$ | 1.02 | $1.24 \times 10^{-1}$ | 0.49 | $5.34 \times 10^{-3}$ | 1.01 |
| Scenario 2 | 0 | $2.93 \times 10^{-2}$ | 8 | $4.42 \times 10^{-2}$ | 1.00 | $2.65 \times 10^{-3}$ | 1.00 | $1.30 \times 10^{-1}$ | 0.66 | $1.78 \times 10^{-2}$ | 0.96 |
|           | 1 | $1.47 \times 10^{-2}$ | 8 | $2.06 \times 10^{-2}$ | 1.00 | $1.29 \times 10^{-3}$ | 1.00 | $8.26 \times 10^{-2}$ | 0.89 | $8.63 \times 10^{-3}$ | 1.04 |
|           | 2 | $7.33 \times 10^{-3}$ | 7 | $9.54 \times 10^{-3}$ | 1.00 | $6.33 \times 10^{-4}$ | 1.00 | $4.45 \times 10^{-2}$ | 0.91 | $4.25 \times 10^{-3}$ | 1.02 |
|           | 3 | $3.66 \times 10^{-3}$ | 7 | $4.58 \times 10^{-3}$ | 1.00 | $3.13 \times 10^{-4}$ | 1.00 | $2.36 \times 10^{-2}$ | 1.02 | $2.07 \times 10^{-3}$ | 1.04 |
|           | 4 | $1.83 \times 10^{-3}$ | 6 | $2.22 \times 10^{-3}$ | 1.00 | $1.54 \times 10^{-4}$ | 1.00 | $1.17 \times 10^{-2}$ | 1.03 | $1.01 \times 10^{-3}$ | 1.03 |
|           | 5 | $9.16 \times 10^{-4}$ | 5 | $1.09 \times 10^{-3}$ | 1.00 | $7.57 \times 10^{-5}$ | 1.00 | $5.78 \times 10^{-3}$ | 1.03 | $1.01 \times 10^{-3}$ | 1.03 |
| Scenario 3 | 0 | $1.96 \times 10^{-2}$ | 12 | $2.59 \times 10^{-2}$ | 1.00 | $2.71 \times 10^{-2}$ | 1.00 | $7.58 \times 10^{-1}$ | 0.46 | $1.35 \times 10^{-1}$ | 0.94 |
|           | 1 | $9.77 \times 10^{-3}$ | 15 | $1.28 \times 10^{-2}$ | 1.00 | $1.35 \times 10^{-2}$ | 1.00 | $5.52 \times 10^{-1}$ | 0.47 | $7.00 \times 10^{-2}$ | 0.96 |
|           | 2 | $4.88 \times 10^{-3}$ | 18 | $6.40 \times 10^{-3}$ | 1.00 | $6.71 \times 10^{-3}$ | 1.00 | $3.92 \times 10^{-1}$ | 0.49 | $3.60 \times 10^{-2}$ | 0.96 |
|           | 3 | $2.44 \times 10^{-3}$ | 18 | $3.18 \times 10^{-3}$ | 1.00 | $3.33 \times 10^{-3}$ | 1.00 | $2.72 \times 10^{-1}$ | 0.53 | $1.84 \times 10^{-2}$ | 0.97 |
|           | 4 | $1.22 \times 10^{-3}$ | 21 | $1.57 \times 10^{-3}$ | 1.00 | $1.65 \times 10^{-3}$ | 1.00 | $1.82 \times 10^{-1}$ | 0.57 | $9.30 \times 10^{-3}$ | 0.98 |
|           | 5 | $6.10 \times 10^{-4}$ | 16 | $7.74 \times 10^{-4}$ | 1.00 | $8.11 \times 10^{-4}$ | 1.00 | $1.18 \times 10^{-1}$ | 0.63 | $4.64 \times 10^{-3}$ | 1.00 |
Given the control $I$ we consider the time-dependent counterpart of (1.1) as

$$
\begin{cases}
\partial_t u(t, x) = \Phi(u, I, x), & (t, x) \in (0, \infty) \times \mathbb{R}, \\
u(0, x) = g(x), & x \in \mathbb{R},
\end{cases}
$$

(5.1)

where

$$\Phi(u, I, x) := b_1(x, I)\partial_x u(t, x) - (p_0(x) + p_2(I) + r)u(t, x) + p_2(I)V_2(x, 0) + f_1(I)$$

(5.2)

**Spatial semi-discretization** We employ the notation from the previous section, in particular we denote $x_i = i h$, where $h$ is the mesh size. We perform the spatial semi-discretization of (5.1) on the grid (4.2). We use a WENO5 interpolation procedure that employs a stencil with 6 nodes [31],

$$\{x_{i-3}, \ldots, x_{i+2}\} \text{ or } \{x_{i-2}, \ldots, x_{i+3}\},$$

(5.3)

depending on the direction of the velocity, see (5.5). We assign the values to grid points near the boundary of $\Omega_r$; via the boundary condition given by

$$g(x) := \begin{cases}
c - \frac{\bar{c}}{\gamma} I_{nc}^0, & \text{if } x < 0, \\
x + c - \frac{\bar{c}}{\gamma} I_{nc}^0, & \text{if } x \geq 0.
\end{cases}
$$

(5.4)

The choice of the function (5.4) is motivated by the fact that the solution to (1.1) is asymptotically $0$ for $x \to -\infty$ and takes the form (2.2) when $x \to +\infty$.

Below, we describe the WENO spatial discretization in the case that $b_1^I(I) \geq 0$, which uses the stencil biased to the left, i.e. on the nodes $\{x_{i-3}, \ldots, x_{i+2}\}$; the case $b_1^I(I) < 0$ uses the the nodes $\{x_{i-2}, \ldots, x_{i+3}\}$ and is constructed analogously, cf., [31].

We denote the approximation at the nose $x_i$ as $u_i(t) \approx u(t, x_i)$ and define the following interpolated values

$$[u_i(t)]_1 = \frac{1}{h} \left(-\frac{1}{3}u_{i-1}(t) - \frac{1}{2}u_i(t) + u_{i-1}(t) - \frac{1}{6}u_{i+2}(t)\right),$$

$$[u_i(t)]_2 = \frac{1}{h} \left(-\frac{1}{6}u_{i-2}(t) - u_{i-1}(t) + \frac{1}{2}u_i(t) + \frac{1}{3}u_{i+1}(t)\right),$$

$$[u_i(t)]_3 = \frac{1}{h} \left(-\frac{1}{3}u_{i-3}(t) + \frac{3}{2}u_{i-2}(t) - 3u_{i-1}(t) + \frac{11}{6}u_i(t)\right).$$

We define the approximation of $\partial_x u(t, x_i)$ by

$$D^u_t u_i(t) = \omega_1[u_i(t)]_1 + \omega_2[u_i(t)]_2 + \omega_3[u_i(t)]_3,$$

(5.5)

where the weights $\omega_i$ are given by

$$\omega_i = \frac{\alpha_i}{\alpha_1 + \alpha_2 + \alpha_3}, \quad i = 1, 2, 3, \quad \text{with}$$

$$\alpha_i = \frac{d_i}{c_e + \beta_i}, \quad i = 1, 2, 3, \quad d_1 = 3/10, \quad d_2 = 3/5, \quad d_3 = 1/10.$$

(5.6)
The parameter $c_e$ in (5.6) is introduced to avoid a denominator with zero value, in the numerical experiments below we set $c_e = 10^{-6}$. The parameters $\beta_i$ in (5.6), the so-called smoothness indicators, are defined as

$$
\beta_1 = \frac{13}{12} (\tilde{u}^i - 2\tilde{u}^{i+1} + \tilde{u}^{i+2})^2 + \frac{1}{4} (3\tilde{u}^i - 4\tilde{u}^{i+1} + \tilde{u}^{i+2})^2,
$$

$$
\beta_2 = \frac{13}{12} (\tilde{u}^{i-1} - 2\tilde{u}^i + \tilde{u}^{i+1})^2 + \frac{1}{4} (\tilde{u}^{i-1} - 4\tilde{u}^{i+1})^2,
$$

$$
\beta_3 = \frac{13}{12} (\tilde{u}^{i-2} - 2\tilde{u}^{i-1} + \tilde{u}^i)^2 + \frac{1}{4} (\tilde{u}^{i-2} - 4\tilde{u}^{i-1} + 3\tilde{u}^i)^2,
$$

where $\tilde{u}^i = (u^i - u^{i-1})/h$.

Hence, the spatial semi-discretization based on the upwind WENO5 interpolation procedure leads to the following system of ordinary differential equations

$$
\begin{aligned}
\partial_t u_i(t) &= \Phi_i(u_i, I), \quad t \in (0, \infty), \\
u_i(0) &= g(x_i),
\end{aligned}
$$

(5.7)

where

$$
\Phi_i(u_i, I) := b_1(x_i, I) D_x u_i - (p_0(x_i) + p_2(I) + r) u_i + p_2(I) V_2(x_i, 0) + f_1(x_i).
$$

**Time-discretization**

Given a time step $\Delta t$ and a sequence of discrete time levels $t_\ell = \ell \Delta t$ we denote the fully discrete approximation at $(t_\ell, x_i)$ as $u_i^\ell \approx u_i(t_\ell))$. The time-discretization of (5.7) is based on the Heun method, which is an explicit one-step 2-stage Runge–Kutta method of order 2 (RK22). The choice of the time-discretization is motivated by the stability properties of the RK22 scheme [30]. Since the scheme is explicit we set $\Delta t = h/2$ to preserve the associated CFL condition.

The fully discrete scheme for the approximation of (5.1) then reads as follows: given an initial guess $\{u_i^0\}_{i=1}^{M-2}$ compute for $\ell = 1, \ldots$\n
$$
u_i^\ell = u_i^\ell - \Delta t \frac{(k_1 + k_2)}{2} i = 1, \ldots, M - 2,
$$

(5.8)

where

$$
k_1 = \Phi_i(u_i^{\ell-1}, I),
$$

$$
k_2 = \Phi_i(u_i^{\ell-1} + \Delta t k_1, I).
$$

We then obtain the following modification of the policy iteration algorithm:

**Algorithm 5.1** (WENO) Start with the initial guess $\overline{I}^{(0)} := I_{nc}$ and for $n = 1, \ldots$ iterate as follows:

**Policy evaluation:**

1. Set $u^0_i = V_{1,n-1}^{i}$ for $n > 1$ and $V_1^{i,(0)} = g(x_i)$ for $i = -3, \ldots, M + 1$.
2. For $\ell = 1, \ldots$ compute $\{u^\ell_i\}_{i=1}^{M-2}$ using (5.8) with $I \equiv \overline{I}^{(n-1)}$ until $\max_{i=1,\ldots,M-2} \mid u_i^\ell - u_i^{\ell-1} \mid \leq 10^{-6}$.
3. Set $V_i^{n,(n)} = u_i^\ell, i = 1, \ldots, M - 2$ and proceed to the policy improvement step.

**Policy improvement:** Update the control $\overline{I}^{(n)} := (I_1^{1,(n)}, \ldots, I_{M-2}^{1,(n)})$ with

$$
\overline{I}_i^{1,(n)} = \frac{\gamma_I}{\xi} \frac{(V_2^{i,0} - V_1^{i,(n)})}{D_0 V_1^{i,(n)}}.
$$
Comparison of the solutions obtained using the upwind finite difference scheme and the WENO scheme, i.e., Algorithms 4.1 and 5.1 for $M = 2^{11}$, Scenario 1.

Table 5: Number of time-marching iteration (WENO5 + RK22) required at each policy iteration

| Policy iter. | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | … | 443 |
|--------------|----|----|----|----|----|----|----|----|----|----|---|----|
| # time-teps  | 5144 | 5023 | 4990 | 3271 | 2323 | 1174 | 1483 | 942 | 12 | 2  | … | 2  |

**Stopping criterion:** *Repeat the loop until*

$$\max_{i=1,\ldots,M-2} \frac{|I^{i,(n)} - I^{i,(n-1)}|}{\max_{i=1,\ldots,M-2} |I^{i,(n)}|} \leq 10^{-6}.$$ 

**Numerical results** We discuss the results for Scenario 1, the behavior of Algorithm 5.1 in the remaining scenarios was similar. We observe very good agreement between the solutions computed with the upwind finite difference scheme and the WENO scheme, see Fig. 6. In Fig. 7 we display the error between the solution of the WENO scheme for decreasing mesh size and the solution of the upwind finite difference scheme on a very fine mesh with $M = 2^{20}$ grid points; we observe a linear convergence for the approximation of $V_1$ and convergence of order $\frac{1}{2}$ for the control in the maximum norm. Algorithm 5.1 requires about 30 – 40 time more policy iterations than Algorithm 4.1, in Scenario 1 443 policy iterations were required to achieve the convergence criterion. As expected, the number of time-steps required to reach the steady-state in each policy iteration, decreases in later states of the policy iteration algorithm, only two time-steps per policy iteration are required after the 10th policy iteration, see Tables 3 and 4. The advantage of the time-stepping approach in Algorithm 5.1 is that it does not require the solution of a linear system of equations, nevertheless in the present setting Algorithm 4.1 was considerably faster in terms of CPU time due to the lower policy iteration count.

To reduce the number of policy iterations for the WENO scheme we employ a preconditioning approach: we use the control computed by Algorithm 4.1 as an initial guess for Algorithm 5.1. We obtained a reduction in the number of policy iterations as well as in the total number of time-steps, see Table 6.
54

Fig. 7 Scenario 1: difference between the solution of the upwind finite difference scheme with $M = 2^{20}$ and the solution of the WENO scheme with $M = 2^{11}, 2^{12}, 2^{13}$ (left $V_1$ and right $I$).

Table 6 Number of time-marching iteration (WENO5 + RK22) required at each policy iteration with control preconditioning

| Policy iter. | 1    | 2    | 3     | 4     | 5     | ... | 318 |
|--------------|------|------|-------|-------|-------|-----|-----|
| # time-steps | 3007 | 994  | 782   | 42    | 2     | ... | 2   |

6 Discretization by the Stabilized Finite Element Method

In this section we introduce stabilized finite element (FEM) schemes for the approximation of (1.1)–(1.2). We note that, in contrast to the collocation method, we do not employ the transformation (3.1) and, similarly as in the previous section, we compute the numerical solution on domains of finite size. 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.

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

6.1 Discretization of the Value in Mode $m = 2$

We note that for $x \geq 0$ by Proposition 2.1 the solution of the HJB Eq. (1.2) is given explicitly by (2.1). Hence, similarly as in Sect. 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 Eq. (1.2) on the bounded domain $\Omega^- = (-20, 0) \times (0, 20)$. The boundary of $\Omega^-$ is denoted as $\partial \Omega^- \cap (0, y)$, $y \in [0, 20]$

For simplicity we construct the finite element scheme using uniform triangulations $T_h$ of $\Omega^-$. For a fixed mesh size parameter $h > 0$ the triangulation $T_h$ is obtained by partitioning $\Omega^-$ into uniform squares with side $h$ and each square is then sub-divided into four equally sized right-angled triangles. Given a triangulation $T_h$ of $\Omega^-$ we denote by $V_h(T_h)$ the standard finite element space of continuous functions which are piecewise linear with...
respect to the triangulation \( T_h \). The set of nodes \( \{ x_\ell = (x_\ell, y_\ell) \}_{\ell=1}^L \) of \( T_h \) is denoted as \( \mathcal{N}_h \). Note that \( \mathcal{V}_h(T_h) = \text{span}\{\phi_\ell, \ \ell = 1, \ldots, L\} \) where \( \{\phi_\ell\}_{\ell=1}^L \) are the standard nodal basis functions, s.t. \( \phi_\ell(x_m) = \delta_m\ell \) for all \( x_m \in \mathcal{N}_h \). We also introduce a finite element space of functions that satisfy the homogeneous Dirichlet boundary condition on \( \Gamma_D \) as \( \mathcal{V}_{h,0} = \{\phi_h \in \mathcal{V}_h(T_h), \ \phi_h|_{\Gamma_D} = 0\} \).

The standard nodal interpolation operator is denoted as \( I_h : C(\Omega^-) \to \mathcal{V}_h(T_h) \), where for \( v \in C(\Omega^-) \) the interpolant \( I_h v \in \mathcal{V}_h(T_h) \) is defined via the relation \( I_h v(x_\ell) = v(x_\ell) \), \( \forall x_\ell \in \mathcal{N}_h \), i.e., \( I_h v(x) = \sum_{\ell=1}^L v(x_\ell) \phi_\ell(x) \) for \( x \in \Omega^- \). For \( u, v \in C(\Omega^-) \) define the discrete \( L^2 \)-inner product for \( \Omega^- \subseteq \mathbb{R}^2 \) as

\[
(u, v)_h = \int_{\Omega^-} I_h(uv)(x) \, dx = \sum_{T \in T_h} \frac{|T|}{3} \sum_{i=1}^3 u(T,i)v(T,i),
\]

where \( T, 1, T, 2, T, 3 \in \mathcal{N}_h \) denote the vertices of the triangle \( T \in T_h \).

The finite element space of piecewise linear continuous functions only satisfies \( \mathcal{V}_h(T_h) \subset H^1(\Omega^-) \), i.e., the second-order derivatives are not defined for \( v_h \in \mathcal{V}_h(T_h) \). For \( v_h \in \mathcal{V}_h(T_h) \) we introduce the discrete second-order operator \( \delta_{xx,h} : \mathcal{V}_h(T_h) \to \mathcal{V}_h(T_h) \), \( \delta_{xx,h} \approx \delta_{xx} \) as

\[
(-\delta_{xx,h} v_h, \phi_h) = (\partial_{xx} v_h, \partial_{xx} \phi_h) \quad \forall \phi_h \in \mathcal{V}_h(T_h),
\]

and similarly we define \( \delta_{yy,h} \approx \delta_{yy} \). Consequently the discrete Laplace operator \( \Delta_h := \delta_{xx,h} + \delta_{yy,h} : \mathcal{V}_h(T_h) \to \mathcal{V}_h(T_h) \) satisfies

\[
(-\Delta_h v_h, \phi_h) = (\nabla v_h, \nabla \phi_h) \quad \forall \phi_h \in \mathcal{V}_h(T_h).
\]

Let \( \{\phi_j\}_{j=1}^L \) be the set of nodal basis functions of \( \mathcal{V}_h(T_h) \), i.e., \( \mathcal{V}_h(T_h) = \text{span}\{\phi_1, \ldots, \phi_L\} \). Hence, any \( v_h \in \mathcal{V}_h(T_h) \) 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 (6.1) and the fact that \( \phi_\ell(x_m) = \delta_m\ell \) for \( x_m \in \mathcal{N}_h \), we deduce from (6.2) that

\[
\partial_{xx} v_h(x_\ell) = 3|\text{supp}(\phi_\ell)|^{-1}(\partial_{xx} v_h, \partial_{xx} \phi_\ell) \quad \forall x_\ell \in \mathcal{N}_h,
\]

where \( \text{supp}(\phi_\ell) = \{x \in \Omega^-, \ \phi_\ell(x) \neq 0\} \). Analogically (6.3) implies that

\[
\Delta_h v_h(x_\ell) = 3|\text{supp}(\phi_\ell)|^{-1}(\nabla v_h, \nabla \phi_\ell) \quad \forall x_\ell \in \mathcal{N}_h,
\]

cf. [13, 17, 27].

The finite element approximation of the continuous problem is then constructed as follows. The finite element solution \( V_{2,h} \in \mathcal{V}_h(T_h) \) satisfies the Dirichlet boundary condition at \( \Gamma_D \), that is \( V_{2,h}(x_\ell) = V_2(0, y_\ell) \) for \( x_\ell = (0, y_\ell) \in \mathcal{N}_h \cap \Gamma_D \) (where \( V_2(0, \cdot) \) is given by (2.1)) and solves the discrete system

\[
\left( -\frac{1}{2}\sigma^2, \delta_{yy}, h V_{2,h} + b \cdot \nabla V_{2,h} + (r + p_0)V_{2,h}, \phi_h \right)_h = \left( h_{\text{stab}} \Delta_h V_{2,h}, \phi_h \right)_h = (f_2, \phi_h) \quad \forall \phi_h \in \mathcal{V}_{h,0},
\]

where \( h_{\text{stab}}(x_\ell) = h|b(x_\ell)|, \) with \( b = -(b_2, \mu) \).

On setting \( L_{2,h}(V_{2,h}) := \frac{1}{2}\sigma^2(y) \delta_{yy}, h V_{2,h} - b \cdot \nabla V_{2,h} - p_0 V_{2,h} \) we observe that (6.4) is equivalent to

\[
r(V_{2,h}, \phi_h)_h = \left( L_{2,h}(V_{2,h}) + f_2, \phi_h \right)_h + \left( h_{\text{stab}} \Delta_h V_{2,h}, \phi_h \right)_h \quad \forall \phi_h \in \mathcal{V}_h,
\]
which is a discrete counterpart of (1.4) with an additional artificial diffusion term \( h_{\text{stab}} \Delta_h V_{2,h} \).

The artificial diffusion term guarantees the monotonicity and convergence of the finite element approximation, cf. [17] and [27,Sect. 3.5]. It is well known that the nodal basis functions of \( \nabla_h (T_h) \) satisfy

\[
(\nabla \phi_i, \nabla \phi_j) \leq 0 \text{ for } i \neq j,
\]

for weakly acute triangulation \( T_h \). Since we employ uniform right-angled triangulations (consequently \( T_h \) are weakly acute) it can be shown that the artificial diffusion term \(-h_{\text{stab}} \Delta_h V_{2,h}\) ensures the monotonicity of the finite element scheme (6.4), [27,Lemma 3.42]. The monotonicity of the finite element approximation for adaptive meshes can be ensured by a suitable choice of the stabilization parameter \( h_{\text{stab}} \), for more details see [17,Sect. 8] and [27]. By the monotonicity of the numerical approximation standard arguments imply that the finite element solution \( V_{2,h} \) of (6.4) converges to the (unique) viscosity solution of (1.2) (considered on \( \Omega^- \), cf. [17]. Furthermore, on noting the representation \( V_{2,h} \equiv V_{2,h}(x,y) = \sum_{\ell=1}^{L} v_{2,\ell} \phi_{\ell}(x,y) \) we observe that (6.4) is equivalent to a system of linear equations for the coefficients \( \{ v_{2,\ell} \}_{\ell=1}^{L} \). Hence, owing to the monotonicity of the numerical approximation the associated matrix is a \( M \)-matrix, which implies that the numerical approximation (6.4) satisfies a discrete maximum principle, see for instance [27,Corollary 3.43].

We observed that, in the present setting, it is also possible to use the numerical scheme (6.4) without the stabilization (i.e., with \( h_{\text{stab}} \equiv 0 \)) and the resulting numerical solution (not displayed) is very similar to those obtained by the stabilized scheme. In general the use of suitable numerical stabilization is advisable to ensure the reliability of numerical approximation.

The graph of the finite element solution \( V_{2,h} \) for Scenarios 1–3 (see Sect. 2) computed on a mesh with mesh size \( h = 20 \times 2^{-11} \) is displayed in Fig. 8.

6.2 Discretization of the Value in Mode \( m = 1 \)

In this section we propose two adaptive finite element algorithms for the numerical solution of (1.1). The first algorithm uses a stabilized finite element method along with a heuristic mesh refinement strategy, the second adaptive finite element algorithm is based on a least-squares approach.

Since the finite element method requires the spatial domain to be of finite size we consider (1.1) for \( x \in \Omega = (-10, 10) \). If not mentioned otherwise we impose the following Dirichlet boundary condition

\[
V_1(-10) = 0 \text{ and } V_1(10) = 10 + c - \frac{\xi}{\gamma_1} I^{nc};
\]
the first condition is motivated by heuristic observations and the second condition is derived from (2.2). We partition \( \Omega \) into possibly non-equidistant subintervals, i.e., we denote \( T_\ell = (x_{\ell-1}, x_\ell) \), \( \ell = 1, \ldots, L \) and \( T_h = \bigcup_{\ell=1}^L T_\ell \). The partition \( T_h \) is determined by an adaptive refinement strategy, that is, the computation starts with a coarse initial grid and then performs local mesh refinements and coarsening using suitable criteria which will be discussed below. The space of piecewise linear globally continuous functions on \( T_h \) is denoted as \( \mathbb{V}_h \equiv \mathbb{V}_h(T_h) = \text{span}\{\phi_\ell, \; \ell = 1, \ldots, L\} \) and \( \mathbb{V}_h, 0 := \{\phi \in \mathbb{V}_h; \; \phi(-10) = \phi(10) = 0\} \). The value function \( V_1 \) is then approximated by a continuous and piecewise linear function \( V_{1,h} \in \mathbb{V}_h \).

We propose the following adaptive numerical algorithm which combines the policy iteration algorithm (cf., Algorithm 4.1) with iterative mesh refinement. The algorithm requires the values of \( V_2(x, 0), x \in \Omega \); we construct an approximation \( V_{2,h} \approx V_2(\cdot, 0) \) as follows. For \( x \leq 0 \) we obtain the (piecewise linear) approximation \( V_{2,h} \in \mathbb{V}_h \equiv \mathbb{V}_h(T_h) \) using the stabilized finite element method from Sect. 6.1 on a uniform mesh with mesh size \( \bar{h} = 20 \times 2^{-11} \), see Fig. 8; for \( x \geq 0 \) the approximation \( V_{2,h} \) is obtained from formula (2.1) by piecewise linear interpolation over a uniform mesh with the same mesh size.

**Algorithm 6.1** (Adaptive Scheme) **Initialization:** Given some initial triangulation \( T_{h,0} \) and the approximation \( V_{2,h} \in \mathbb{V}_h \) of \( V_2(\cdot, 0) \), set \( I_{h,0} := I^{nc} \) and iterate for \( k = 0, 1, \ldots \):

1. **Policy iteration:**
   - Set \( I_h^{(0)} = I_{h,k} \in \mathbb{V}_h(T_{h,k}) \) and iterate for \( n = 1, 2, \ldots \):
     1A) **Policy evaluation:** Compute the approximation \( V_{1,h}^{(n)} \in \mathbb{V}_h(T_{h,k}) \) of the solution \( V_1^{(n)} \) of the linear problem
     \[-b_1(I_h^{(n-1)})\partial_x V_1^{(n)} + (p_0 + p_2(I_h^{(n-1)}) + r) V_1^{(n)} = f_1 + p_2(I_h^{(n-1)}) V_{2,h} \text{ in } \Omega. \quad (6.7)\]
     1B) **Policy improvement:** For \( P_h : L^2(\Omega) \to \mathbb{V}_h (\text{cf., (6.11) below}) \) update the control as
     \[ I_h^{(n)} = \frac{V_1}{\xi} P_h \left( \frac{V_{2,h} - V_{1,h}^{(n)}}{\partial_x V_{1,h}^{(n)}} \right). \quad (6.8)\]
     1C) **Stopping criterion:** Set \( n \to n + 1 \) and proceed with step 1A) unless the relative change of the control \( I_h^{(n)} \) is below the prescribed tolerance (see, (6.9), (6.12), respectively).

2. **Mesh refinement:** Set \( V_{1,h}^{k+1} := V_{1,h}^{(n)}, I_{h,k+1} := I_h^{(n)}, \) and construct a new mesh \( T_{h,k+1} \) by local mesh refinement/coarsening that is performed on the basis of a suitable error indicator \( \eta_h \equiv \eta_h(V_1^{k}, I_{h,k}) \).

3. **Stopping criterion:** Set \( k \to k + 1 \) and proceed to step (1) unless the mesh \( T_{h,k} \) (or the error indicator \( \eta_h \)) satisfies a prescribed stopping criterion.

We note that the use of a suitable operator \( P_h : L^2(\Omega) \to \mathbb{V}_h \) in step 1B) of the above algorithm is necessary because for \( V_{1,h} \in \mathbb{V}_h \) the derivative \( \partial_x V_{1,h} \) is a piecewise constant function, i.e., \( \partial_x V_{1,h} \notin \mathbb{V}_h \). Possible choices of the operator are given implicitly by (6.11) below.

### 6.2.1 Stabilized Finite Element Approximation with Heuristic Mesh Refinement

In this section we introduce a stabilized finite element method with heuristic mesh refinement strategy for the (linear) subproblem (6.7). The method employs the following stopping
criterion in step (1C) of Algorithm 6.1:

$$\frac{\|f_h^{(n)} - f_h^{(n-1)}\|_{L^\infty(\Omega)}}{\|f_h^{(n)}\|_{L^\infty(\Omega)}} \leq \epsilon.$$  

(6.9)

In all considered cases the prescribed tolerance $\epsilon = 10^{-6}$ was typically reached after 5 to 15 iterations; the iteration count is higher in the initial iterations and decreases once a good approximation of the control is available.

To simplify the notation we denote throughout this section $T_h \equiv \Gamma_h^{(n)}, \forall h \equiv V_h(\Gamma_h^{(n)}), V_{1,h} \equiv V_{1,h}^{(n)}$ and $I_h \equiv I_h^{(n-1)}$. We note that the discrete inner product for $\Omega \subseteq \mathbb{R}^1$ takes the form

$$(u, v)_h = \int_{\Omega} I_h(uv)(x)dx = \sum_{\ell=1}^{L} \frac{(x_{\ell} - x_{\ell-1})}{2}(u(x_{\ell-1})v(u(x_{\ell-1}) + u(x_{\ell})v(u(x_{\ell}))).$$

We employ a 1d counterpart of the stabilized finite element method from Sect. 6.1, i.e., given the control $I_h$ the finite element approximation $V_{1,h} \in \mathcal{V}_{h}$ that satisfies the Dirichlet boundary condition (6.6) and solves

$$\begin{align*}
&\left(-b_1(I_h)\partial_x V_{1,h} + (p_0(I_h) + p_2 + r)V_{1,h}, \phi_h\right) \\
&-\left(h_{\text{stab}} \delta_{xx,h}^2 V_{1,h}, \phi_h\right)_h = \left(p_2(I_h)V_{2,h} + f_1, \phi_h\right) \quad \forall \phi_h \in \mathcal{V}_{h,0},
\end{align*}$$

(6.10)

where the discrete second-order term $\delta_{xx,h}^2 V_{1,h}$ is a 1d analogue of (6.2). The artificial diffusion parameter in the above scheme is defined as

$$h_{\text{stab}} = \sum_{\ell=1}^{L} h_{\text{stab}}(x_{\ell})\phi_{\ell}(x) = \alpha_{\text{stab}} \sum_{\ell=1}^{L} \frac{|\text{supp}(\phi_{\ell})|}{4} |b_1(x_{\ell}, I_h(x_{\ell}))| \phi_{\ell}(x),$$

with $\text{supp}(\phi_{\ell}) = \{x \in \Omega, \phi_{\ell}(x) \neq 0\}$, and $\alpha_{\text{stab}} = 1$, unless mentioned otherwise. We note that, in general, the above choice of the stabilization parameter $h_{\text{stab}}$ does not guarantee the $M$-matrix property of the corresponding linear system of equations (i.e., the monotonicity of the numerical approximation). However, in the current setting the $M$-matrix property was only violated for very few rows of the matrix (e.g. rows that correspond to the elements with large gradient in the discrete control) and this effect had negligible influence on the numerical approximation. The monotonicity of the numerical approximation can be obtained by choosing a large value of the stabilization parameter $\alpha_{\text{stab}}$, more sophisticated monotonicity preserving stabilization methods are also available, see for instance in [17].

The discrete control in (6.8) of Algorithm 6.1 reads

$$I_h = \sum_{\ell=1}^{L} I_{\ell} \phi_{\ell}(x) = \frac{\gamma_1}{\xi} P_h \left(\frac{V_{2,h} - V_{1,h}^{(n)}}{\partial_x V_{1,h}^{(n)}}\right) \quad \text{with} \quad I_{\ell} = \frac{\gamma_1}{\xi} \frac{V_{2,h}(x_{\ell}) - V_{1,h}(x_{\ell})}{1/2(\partial_x V_{1,h}|_{T_\ell} + \partial_x V_{1,h}|_{T_{\ell+1}})}.$$  

(6.11)

An alternative choice of the approximation of the optimal control, which is an analogue of the "averaged" finite difference approximation from Sect. 4.1.2, is given as

$$I_{\ell} = \frac{\gamma_1}{\xi} \left(V_{2,h}(x_{\ell}) - V_{1,h}(x_{\ell})\right) \frac{1}{2} \left(\frac{1}{\partial_x V_{1,h}|_{T_\ell}} + \frac{1}{\partial_x V_{1,h}|_{T_{\ell+1}}}\right).$$
Fig. 9 Plot of the $2|\text{supp}(\phi_\ell)|^{-1} = 2(h_i + h_{i+1})^{-1}$ (left), discrete control (middle) and the gradient of the discrete control (right) for Scenario 1

Similarly as in the case of the finite difference approximation, we observe that the above approximation requires similar iteration counts for the convergence as (6.11).

Furthermore, we note that the policy iteration does not converge without any "averaging" of the (piecewise constant) derivative $\partial_x V_1,h$, e.g., for $I_\ell = \gamma_1 \xi V_2,h(x_\ell) - V_1,h(x_\ell)$, or for the choice $I_\ell \equiv \tilde{I}_\ell$, with $\tilde{I}_\ell$ defined below.

For the adaptive mesh refinement we employ the following local error indicator $\eta_T := \eta_h|T, T \in T_h$

$$\eta_T = \alpha_\eta \| I_h - \tilde{I}_h \|_{L^2(T)}^2,$$

with a scaling factor $\alpha_\eta > 0$; the error indicator involves a modified control $\tilde{I}_h = \sum_{\ell=1}^L I_\ell \phi_\ell(x)$ where

$$\tilde{I}_\ell = \gamma_1 \frac{V_2,h(x_\ell) - V_1,h(x_\ell)}{\xi}$$

On meshes with uniform mesh size $h = \frac{1}{2} |\text{supp}(\phi_\ell)|$ the denominator in the above expression becomes $\frac{1}{2} (\partial_x V_1,h|_{T_\ell} + \partial_x V_1,h|_{T_{\ell+1}}) + \frac{|\text{supp}(\phi_\ell)|}{4} \delta_{x,h}^2 V_1,h(x_\ell) \equiv \partial_x V_1,h|_{T_\ell}$. Hence, the error indicator $\eta_h$ measures the difference between discrete controls obtained by two alternative discrete formulas which both approximate the continuous expression of the optimal control (3.14). Due to the sensitivity of the control with respect to the discretization, the above heuristic mesh refinement strategy produces meshes with good approximation properties, see Table 7 below. We observe that the choice of the scaling factor $\alpha_\eta = 2.5$ improves the efficiency of the mesh refinement strategy for the considered simulation parameters.

In Fig. 9 we display the results for Scenario 1 computed on an adaptive mesh with 10945 degrees of freedom. The displayed results indicate growth of the gradient of the control at $x = 0$, which indicates a discontinuity that is also captured by the mesh refinement algorithm (the tolerance for the refinement was $\eta_h = \sum_T \eta_T \leq 2 \times 10^{-5}$). Results for Scenarios 2–3 are not displayed, we observed no significant differences to the results obtained by the finite difference method.

We remark that the scheme (6.10) can be modified by replacing the $L^2$-inner product with the discrete inner product Sect. 6.10; on an equidistant mesh with mesh size $h$ it holds that $\text{supp}(\phi_\ell) = 2h$ and the resulting finite element scheme is equivalent to the upwind finite difference approximation (4.6). Nevertheless, this generalization of the finite difference
scheme turned out to be unsuitable for the use in conjunction with adaptive mesh refinement as the policy iteration in Algorithm 6.1 did not converge.

### 6.2.2 Adaptive Least–Squares Finite Element Approximation

In this section we discuss the least-squares finite element method (LSFEM) for the approximation of (6.7). We employ the following mesh dependent stopping criterion in step (1C) of Algorithm 6.1: set $\text{ndof} := \dim V_h(T_{h,k})$, $\epsilon_k := \text{ndof}^{-1}$ (ndof := dim $V_h(T_{h,k})$), i.e.,

$$\frac{\| I_h^{(n)} - I_h^{(n-1)} \|_{L^2(\Omega)}}{\| I_h^{(n)} \|_{L^2(\Omega)}} \leq \epsilon_k.$$  \hfill (6.12)

As in the previous section we describe the application of the method in the $n$-th iteration of Algorithm 6.1 and denote $T_h = T_h^{(n)}$, $V_h = V_h(T_h^{(n)})$, and $I_h = I_h^{(n-1)}$. The LSFEM method consists of finding the approximation $V_{1,h} \approx V_1^{(n)}$ by the minimization of the functional

$$V_{1,h} = \arg \min_{W_h \in V_h} \| -b_1(I_h) \partial_x W_h + (p_0 + p_2(I_h) + r) W_h - f_1 - p_2(I_h) V_2(\cdot, 0) \|_{L^2(\Omega)}^2.$$ \hfill (6.13)

Notice that we do not impose any boundary conditions, since they are in general unknown (in particular on the negative axis). The above least squares functional provides the following straightforward way for the construction of mesh refinement indicators. The minimization in (6.13) is equivalent to the minimization of the error $W_h - V_1^{(n)}$ over all $W_h \in V_h$ with respect to the semi-norm

$$\| \cdot \|_{a,I_h} := \| -b_1(I_h) \partial_x \cdot + (p_0 + p_2(I_h) + r) \cdot \|_{L^2(\Omega)}.$$ \hfill (6.14)

To minimize the error $\| V_{1,h} - V_1^{(n)} \|_{a,I_h}$, we want to refine the mesh in regions where this error is large. This motivates the evaluation of the residual on each $T \in T_h$, leading to the refinement indicators

$$\eta_{h,T} := \| -b_1(I_h) \partial_x V_{1,h} + (p_0 + p_2(I_h) + r) V_{1,h} - f_1 - p_2(I_h) V_2(\cdot, 0) \|_{L^2(T)}$$

$$= \| -b_1(I_h) \partial_x (V_{1,h} - V_1^{(n)}) + (p_0 + p_2(I_h) + r)(V_{1,h} - V_1^{(n)}) \|_{L^2(T)}.$$\hfill (6.14)

We use this built-in error control to drive adaptive mesh refinements with Dörfler marking strategy (cf. [26]) and bulk parameter $\Theta = 0.3$.

A downside of the LSFEM is that it is not a monotone scheme, in general. Thus the solution $V_{1,h}$ might oscillate, which results in difficulties in the approximation of the control $I_h$. To stabilize the policy improvement in Algorithm 6.1, our numerical experiments suggested a damping, that is, given some damping parameter $\rho \in (0, 1)$ and the operator $P_h$ in (6.11), we replace the policy improvement (6.8) by the damped version

$$I_h^{(n)} := I_h^{(n-1)} + \rho \left( \frac{\gamma_1}{\xi} P_h \left( \frac{V_{2,h} - V_{1,h}^{(n)}}{\partial_x V_{1,h}^{(n)}} \right) - I_h^{(n-1)} \right).$$ \hfill (6.14)

Throughout all numerical experiments, we use the damping $\rho = 1/2$. Nevertheless, there are still tiny oscillations of the control near the origin $x = 0$, causing difficulties in attaining the stopping criterion (6.12). To circumvent this difficulty, we had to replace the $L^\infty$ norm by the $L^2$ norm.
We terminate our computations if the number $k$ of policy iterations in Algorithm 6.1 exceeds fifteen or the number of degrees of freedom $\text{ndof} = \dim V_h$ exceeds half a million. All experiments utilize the parameters in Tables 1, 2 and the approximations of $V_2$ already used in Sects. 4 and 6.2.1. The initial grid $T_0$ is a uniform partition of the domain $\Omega$ into 10 subintervals. We refine the mesh either adaptively or uniformly. The computations utilize the open source tool for solving partial differential equations FEniCS [24].

6.3 Numerical Experiments

In this section we examine the performance of the respective adaptive finite element schemes proposed in Sect. 6.2.1, 6.2.2 for the computation of $V_1$ and the control $I$. The approximation of $V_2$ is constructed using the (uniform) finite element method from Sect. 6.1 as discussed in Sect. 6.2.1. We compare our results with the finite difference scheme introduced in Sect. 4 which also provides the reference solution computed on a grid with $M = 2^{20} \approx 10^6$ grid-points. Table 7 displays the resulting errors on two different underlying grids (the number of degrees of freedom equals roughly 1000 and 15000). To exclude the artificial behavior of the approximations near the left boundary, the errors have been computed on the reduced intervals $\Omega_1 := [-7, 10]$ in Scenario 1 and 3 and $\Omega_1 := [-17, 20]$ in Scenario 2.

It turns out that the adaptive schemes allow for significantly better approximations compared to the (uniformly refined) finite differences scheme, in particular on the coarse grid. This observation underlines the need of adaptive schemes in higher dimensions, where we have to solve such models on coarse grids due to limitations in our computational resources. We do not experience difficulties with the adaptive stabilized FEM in the sense that the policy iterations converge after a finite number of iterations.

The adaptive LSFEM performs well in Scenario 2 and 3, but struggles in Scenario 1: the policy iterates $I_h^{(n)}$ seem to oscillate between several states (cf. Fig. 10) and so do not converge after $\text{ndof}$ exceeds $10^4$. As discussed previously, the failure of convergence of the algorithm is related to the discontinuity in the exact control $I$ (cf. Fig. 9) at the origin $x = 0$, which induces oscillation in the computed approximation $V_{1,h}$ and in the control $I_h$. We overcome the failure of the adaptive LSFEM in Scenario 1 by a domain decomposition approach similar to the one used in the collocation method. More precisely, we use the scheme to solve the problem on the domains $\Omega_l = (-10, 0)$ and $\Omega_r = (0, 10)$ separately with given boundary data for $V_{1,h}$ in the origin $x = 0$. With this splitting, the results obtained by adaptive LSFEM are comparable to those obtained by the FDM and stabilized FEM in Scenario 1, cf. the results in Table 7.

In Fig. 11 we display the mesh size distribution for adaptive meshes with $\text{ndof} \approx 1000$ for the considered adaptive finite element methods. We observe that, up to the boundary, the meshes exhibit similar structure; in particular both algorithms exhibit similar behavior near $x = 0$. The smaller mesh size near the boundary for the stabilized FEM method is due to the boundary effects and has minor effect on the overall accuracy. The efficiency of the scheme can be increased by a suitable modification of the refinement indicator near the boundary, but we do not pursue this approach for simplicity.

7 Conclusion

In general, the computed numerical approximations were similar for all considered numerical schemes in all the considered scenarios, at least for certain choices of simulation parameters.
Table 7  Errors of the stabilized adaptive finite element scheme, the finite differences scheme, and the adaptive least-squares scheme

| Problem | Method     | $\| V_1 - V_{1,h} \|_{L^2(\Omega_e)}$ | $\| I - I_h \|_{L^2(\Omega_e)}$ | $\| V_1 - V_{1,h} \|_{L^\infty(\Omega_e)}$ | $\| I - I_h \|_{L^\infty(\Omega_e)}$ |
|---------|------------|-------------------------------------|--------------------------------|---------------------------------|--------------------------------|
| Scen. 1 | Stab. FEM  | 1.95E-02                            | 5.10E-02                        | 1.16E-02                        | 6.56E-02                        |
|         | ndof ≈ 1000| FDM                                 | 2.35E-02                        | 9.82E-02                        | 2.64E-02                        | 5.57E-01                        |
|         |            | LSFEM                               | 3.25E-01                        | 6.52E-01                        | 1.88E-01                        | 4.15E-01                        |
| Scen. 1 | Stab. FEM  | 2.10E-02                            | 4.20E-02                        | 1.22E-02                        | 2.27E-02                        |
|         | ndof ≈ 15000| FDM                                 | 1.41E-03                        | 6.67E-03                        | 1.58E-03                        | 1.44E-01                        |
|         |            | LSFEM (Split)                       | 2.12E-02                        | 3.97E-02                        | 1.23E-02                        | 8.87E-02                        |
| Scen. 2 | Stab. FEM  | 1.40E-03                            | 4.11E-03                        | 7.92E-04                        | 1.31E-02                        |
|         | ndof ≈ 1000| FDM                                 | 2.54E-03                        | 2.11E-02                        | 2.06E-03                        | 1.14E-01                        |
|         |            | LSFEM                               | 1.93E-04                        | 1.31E-02                        | 2.07E-04                        | 3.69E-02                        |
| Scen. 2 | Stab. FEM  | 1.80E-04                            | 7.14E-04                        | 1.04E-04                        | 8.78E-03                        |
|         | ndof ≈ 15000| FDM                                 | 2.02E-04                        | 1.80E-03                        | 1.60E-04                        | 1.36E-02                        |
|         |            | LSFEM                               | 8.08E-06                        | 1.17E-03                        | 2.99E-05                        | 9.51E-03                        |
| Scen. 3 | Stab. FEM  | 1.15E-02                            | 1.79E-02                        | 1.13E-02                        | 1.87E-01                        |
|         | ndof ≈ 1000| FDM                                 | 2.75E-02                        | 1.01E-01                        | 2.61E-02                        | 7.95E-01                        |
|         |            | LSFEM                               | 3.03E-04                        | 4.57E-02                        | 4.94E-04                        | 2.65E-01                        |
| Scen. 3 | Stab. FEM  | 6.96E-04                            | 1.25E-03                        | 7.64E-04                        | 1.57E-02                        |
|         | ndof ≈ 15000| FDM                                 | 1.63E-03                        | 7.24E-03                        | 1.55E-03                        | 1.89E-01                        |
|         |            | LSFEM                               | 3.35E-05                        | 2.23E-03                        | 3.38E-05                        | 4.53E-02                        |
Nevertheless, we made some interesting observations (in particular in the solution for \( m = 1 \)) that are summarized below. We believe that these observations can be used as a benchmark for the evaluation of the robustness of numerical schemes for Hamilton–Jacobi–Bellman equations.

**Convergence of the policy iteration** The convergence of the policy iteration proved to be the most challenging aspect of numerical approximation of (1.1). The non-monotone numerical schemes, i.e., the collocation and LSFEM methods, required a splitting/domain decomposition approach in order to obtain the convergence of the policy iteration algorithm in Scenario 1, where the control is discontinuous at \( x = 0 \). In addition, without a good initial guess, the collocation method required the use of the transformation (3.1), (3.2) to ensure convergence. The solution computed with the state state transformation can be used as a preconditioner to obtain an accurate solution on larger domains without transformation. The divergence can be explained by the fact that the policy iteration can be interpreted as a nonlinear fixed-point iteration (cf. [5]) and, in the case of discontinuous control, the corresponding nonlinearity is not Lipschitz continuous. Nevertheless, a rigorous convergence analysis of the policy iteration in the present setting is still open. We note that the application of the splitting/domain decomposition approach may be difficult for more general problems.

![Fig. 10](image1.png) **Fig. 10** Zoom at the approximations of the optimal control in Scenario 1 with the FD scheme, the LSFEM with splitting (blue solid line, solutions are graphically indistinguishable), and several (non-convergent) iterates of the policy iteration in the LSFEM without splitting (dashed lines)

![Fig. 11](image2.png) **Fig. 11** Plot of the inverse mesh size \( 2|\text{supp}(\phi_i)|^{-1} = 2(h_i + h_{i+1})^{-1} \) in the \( \log_2 \) scale for Scenario 1–3 (from left to right) for stabilized FEM (solid line) and LSFEM (dotted line)
(e.g., in higher dimensions) where the location of the discontinuity is not known a priori. The monotone schemes proved to be more robust in this respect, in particular when a suitable discrete approximation of the optimal control is employed. For instance in the case of the finite element approximation in Scenario 1, a suitable nodal interpolation of the (piecewise constant) gradient of the numerical solution in the formula for the discrete optimal control was essential in order to ensure robust convergence of the nonlinear solver. We observed that the number of policy iterations is independent of the mesh size, nevertheless the convergence of the policy iteration can be affected by ”too harsh” mesh refinement, i.e., on meshes with too sharp differences in the sizes of the neighboring elements. On the other hand, suitable mesh refinement may improve the convergence of the policy iteration, in particular in the case of LSFEM method. We note that an additional benefit of the mesh refinement algorithm is that the iterations counts decrease in the final stages of the mesh refinement loop, i.e., once the mesh is close to the optimal one. This can be attributed to the fact that the final refinement steps typically only introduce minor changes in the control. In the case of the WENO scheme, the convergence of the policy iteration combined with the time-stepping algorithm was robust, but the algorithm required a high number of iterations to reach the prescribed tolerances.

**WENO scheme** The algorithm produced very similar results as the monotone finite difference and finite element schemes. The disadvantage of the method is the high iteration count of the policy iteration, nevertheless compared to the LSFEM and the collocation method, the convergence of the policy iteration was robust. Suitable preconditioning strategies can improve the efficiency of the algorithm.

**Boundary/transformation effects** Except for the collocation method, we observed no major differences between the computed approximation of $V_1$ by the respective numerical schemes. On larger domains the collocation method becomes inaccurate due to the transformation (3.1), (3.2) which is required for the convergence of the policy iteration. The transformation causes the collocation points to be concentrated near $x = 0$ and the approximation becomes increasingly inaccurate on large domains; a more advanced adaptive collocation strategies could improve the accuracy of the scheme, cf. [4, 35]. Without the transformation the collocation method produces accurate results on large domains as well, but it requires preconditioning to ensure the convergence of the policy iteration algorithm. Compared to other methods, the adaptive LSFEM methods exhibited slightly faster convergence behavior w.r.t. the mesh size, nevertheless due to the fact that the value function $V_1$ is smooth (i.e., the solution does not exhibit any shocks) the differences were not significant. For $x$ close to the origin, the computed optimal controls also exhibited good agreement for all methods, in particular when adaptivity is employed. Nevertheless, near the left boundary the three methods produce very different numerical results. This can be explained by the different approximation of the infinite spatial domain as well as the different choice of the (artificial) boundary conditions. With the exception of the collocation method, all considered numerical schemes exhibit similar behavior for increasing sizes of the (bounded) spatial domain, i.e., the effects of the artificial boundary conditions can be mitigated by the choice of a sufficiently large spatial domain for numerical computations.

**Approximation of $V_2$** The numerical solution of $V_2$ enters as a parameter in the numerical approximation of $V_1$ and hence the quality of the approximation of $V_2$ has some influence on the numerical approximation of $V_1$. 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 as well as the aforementioned domain transformation.
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 results suggest that the use of monotone schemes for $V_2$, such as the stabilized FEM method in Sect. 6.1 produces a more reliable numerical algorithm overall.

**Funding** Open Access funding enabled and organized by Projekt DEAL. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)-SFB 1283/2 2021-317210226.

**Data Availability** Enquiries about data availability should be directed to the authors.

**Competing interests** The authors have not disclosed any competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**References**

1. Barles, G., Souganidis, P.E.: Convergence of approximation schemes for fully nonlinear second order equations. Asymptotic Anal. 4(3), 271–283 (1991)
2. Bellman, R.: 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. Bokanowski, O., Falcone, M., Ferretti, R., Grüne, L., Kalise, D., Zidani, H.: Value iteration convergence of $\epsilon$-monotone schemes for stationary Hamilton-Jacobi equations. Discrete Contin. Dyn. Syst. 35(9), 4041–4070 (2015)
4. Bokanowski, O., Garcke, J., Griebel, M., Klompmaker, I.: An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. J. Sci. Comput. 55(3), 575–605 (2013)
5. Bokanowski, O., Maroso, S., Zidani, H.: Some convergence results for Howard’s algorithm. SIAM J. Numer. Anal. 47(4), 3001–3026 (2009)
6. Bungartz, H.-J., Griebel, M.: Sparse grids. Acta Numer. 13, 147–269 (2004)
7. Camilli, F., Falcone, M.: An approximation scheme for the optimal control of diffusion processes. RAIRO Modél. Math. Anal. Numér. 29(1), 97–122 (1995)
8. Carlini, E., Ferretti, R., Russo, G.: A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations. SIAM J. Sci. Comput. 27(3), 1071–1091 (2005)
9. Dawid, H., Wen, X.: Product innovation investment under technological uncertainty and financial constraints. SFB1283 Preprint 20101 https://www.sfb1283.uni-bielefeld.de/preprints/sfb20101.pdf, (2020)
10. Debrabant, K., Jakobsen, E.R.: Semi-Lagrangian schemes for linear and fully non-linear diffusion equations. Math. Comp. 82(283), 1433–1462 (2013)
11. Falcone, M.: A numerical approach to the infinite horizon problem of deterministic control theory. Appl. Math. Optim. 15(1), 1–13 (1987)
12. Falcone, M., Kalise, D.: A high-order semi-lagrangian/finite volume scheme for hamilton-jacobi-isaacs equations. In: System Modeling and Optimization: CSMO 2013. volume 443 of IFIP Advances in Information and Communication Technology, pp. 105–117. Springer, Berlin, Heidelberg (2013)
13. Feng, X., Glowinski, R., Neilan, M.: Recent developments in numerical methods for fully nonlinear second order partial differential equations. SIAM Rev. 55(2), 205–267 (2013)
14. Fleming, W.H., Soner, H.M.: Controlled Markov processes and viscosity solutions. volume 25 of Stochastic Modelling and Applied Probability, 2nd edn. Springer, New York (2006)
15. Gallistl, D., Süli, E.: Mixed finite element approximation of the Hamilton-Jacobi-Bellman equation with Cordes coefficients. SIAM J. Numer. Anal. 57(2), 592–614 (2019)
16. Howard, R.A.: Dynamic programming and Markov processes. The Technology Press of M.I.T., Cambridge, Mass.; John Wiley & Sons, Inc., New York-London, (1960)
17. Jensen, M., Smears, I.: On the convergence of finite element methods for Hamilton-Jacobi-Bellman equations. SIAM J. Numer. Anal. 51(1), 137–162 (2013)
18. Jensen, R.: The maximum principle for viscosity solutions of fully nonlinear second order partial differential equations. Arch. Rational Mech. Anal. 101(1), 1–27 (1988)
19. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. 126(1), 202–228 (1996)
20. Keys, R.G.: Cubic convolution interpolation for digital image processing. IEEE Trans. Acoust. Speech Signal Process. 29(6), 1153–1160 (1981)
21. Lakkis, O., Mousavi, A.: A least-squares galerkin approach to gradient and hessian recovery for nondivergence-form elliptic equations. IMA Journal Numerical Analysis 09, drab034 (2021)
22. Li, F., Shu, C.-W.: Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. Appl. Math. Lett. 18(11), 1204–1209 (2005)
23. Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. 115(1), 200–212 (1994)
24. Logg, A., Mardal, K.-A., Wells, G.N.: editors Automated solution of differential equations by the finite element method, volume 84 of Lecture Notes in Computational Science and Engineering. Springer, Heidelberg, The FEniCS book (2012)
25. Ma, J., Ma, J.: Finite difference methods for the Hamilton-Jacobi-Bellman equations arising in regime switching utility maximization. J. Sci. Comput. 85(3), 55, 27 (2020)
26. Morin, P., Siebert, K., Veeser, A.: A basic convergence result for conforming adaptive finite elements. Math. Models Methods Appl. Sci. 18(5), 707–737 (2008)
27. Neilan, M., Salgado, A.J., Zhang, W.: Numerical analysis of strongly nonlinear PDEs. Acta Numer 26, 137–303 (2017)
28. Pham, H.: Continuous-time stochastic control and optimization with financial applications. Stochastic Modelling and Applied Probability, vol. 61. Springer-Verlag, Berlin (2009)
29. Salgado, A.J., Zhang, W.: Finite element approximation of the Isaacs equation. ESAIM Math. Model. Numer. Anal. 53(2), 351–374 (2019)
30. Shu, C.-W.: Total-variation-diminishing time discretizations. SIAM J. Sci. Statist. Comput. 9(6), 1073–1084 (1988)
31. Shu, C.-W.: High order weighted essentially nonoscillatory schemes for convection dominated problems. SIAM Rev. 51(1), 82–126 (2009)
32. Sun, B., Guo, B.-Z.: Convergence of an upwind finite-difference scheme for Hamilton-Jacobi-Bellman equation in optimal control. IEEE Trans. Automat. Control 60(11), 3012–3017 (2015)
33. Touzi, N.: 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
34. Wang, S., Gao, F., Teo, K.L.: 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)
35. Warin, X.: Adaptive sparse grids for time dependent Hamilton–Jacobi–Bellman equations in stochastic control. arXiv preprint arXiv:1408.4267, (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.