A numerical approach for a constrained free-boundary Hamilton-Jacobi-Bellman equation arising from financial insurance

Huanqun Jiang*[1] and Nathan Gibson†[2]

[1,2]*Department of Mathematics, Oregon State University, Corvallis, OR, 97331*

Abstract

This paper describes a novel numerical method for the solution of a free-boundary Hamilton-Jacobi-Bellman (HJB) equation. The equation comes from an optimal dividend problem within financial insurance and has a classical solution under strict assumptions. However the numerical approach is significant since the problem can be classified as a constrained variational inequality with free boundary, for which numerical methods are few. We combine semi-smooth and projected Newton methods with a shooting-like method to treat the nonsmoothness, implicit constraints and free boundary of the problem, respectively. The algorithms and associated convergence analysis are discussed. In the end, we show that the local convergence rate for this method is at least superlinear. The contribution of this paper is a numerical approach that can be applied to more general free boundary problems with first or second derivative constraints for which analytical solutions are not known.

Keywords— Hamilton-Jacobi-Bellman equation, Free boundary, Semi-smooth projected Newton, Inequality state constraints, Stochastic control.
Mathematics Subject Classification: 97M30, 49M15, 93E20, 35R35, 65K15

1 Introduction

The necessary conditions for stochastic optimal control problems can be represented by (an often nonlinear) differential equation called the Hamilton-Jacobi-Bellman (HJB) equation. The equation is usually coupled to inequality constraints on the states, and can contain non-smooth terms. This classical problem is well surveyed in the book [5]. Analytical solutions are only known under certain strict assumptions. The uniqueness of the optimal solution can be verified by the arguments of viscosity solutions with the aid of probabilistic techniques. Also, there exists a numerical approach, called the Markov chain approximation method, that allows one to solve for these viscosity solutions

*jiangh@math.oregonstate.edu
†Corresponding author, gibsonn@math.oregonstate.edu

numerically ([12]). However, classical numerical methods have not been directly applied to the HJB equation described below.

The main problem investigated in this paper comes from the area of insurance dividend optimization. Here we consider the basic model where jumps are not allowed and the parameters are constants, due to which the problem has an analytical solution ([1, 7]). These solutions are derived by utilizing certain properties of the solution. Specifically, the implicit conditions of the solutions are imposed upon the HJB equation, which allows the non-smooth equations to be decoupled into smooth, linear equations, under strict assumptions. The purpose of this paper is to explore the applicability of a novel numerical approach to such problems, by discretizing the HJB equation directly, which can then be applied more generally to problems without known analytical solutions.

For details of the stochastic models and HJB equation derivation, we refer the readers to the papers mentioned above. The system formulation will be given below in Section 2. In brief, it is a second-order nonlinear ODE with a non-smooth term. There is a one-side boundary condition and two implicit inequality state constraints. The original system is defined on the entire positive x-axis. However, for computational purposes, we must truncate the domain to work on a finite computational window, which introduces a free boundary condition.

1.1 Background

To handle the non-smooth term, we will apply a semi-smooth Newton method. The book [16] surveys the semi-smooth Newton method for variational inequalities, but the application and effectiveness of this method in the current problem has not been investigated. Complicating the numerical approach are the other aspects particular to this problem, namely the one-sided free boundary and implicit conditions on the state variables.

To deal with implicit conditions in a function minimization problem, the paper [3] uses the so-called projected Newton method. A transfer method is devised to convert general linear constraints into a simple linear form. For our system, only after discretization of the spatial derivative in the constraints, can they be reformulated into a simple linear form. Others have extended projected Newton methods to solve non-smooth equations. The paper [15] is one of those which apply a projected Newton type method to solve a constrained non-smooth system. However, the free boundary has not been investigated in this context.

Here we propose a shooting-like method for the free boundary condition. The basic idea is that we have an initial guess of the end boundary value and shoot a candidate solution trajectory using the HJB equation. Then we adjust our state, and thus the target (end boundary value), according to the implicit state constraints and finally shoot a new solution trajectory. Our approach differs from traditional shooting in that the end boundary condition is enforced and the inequality constraints are checked, rather than enforcing an initial slope and checking the end boundary condition. This approach is similar to the "method of sweeps" used in [13] for the optimal control of population dynamics.

The paper outline looks as follows. In Section 2, we write down the explicit problem of interest. In the Section 3, the detailed motivation of shooting-like method is given and we initially provide a heuristic algorithm for this problem (we call it heuristic because the idea of projection and shooting methods are used, but it is not yet a formal systematic formulation). We then reformulate the system after discretization as a simple linearly constrained non-smooth problem through the combined projection. Along with these solutions, we also compare two different shooting strategies: shooting at end-boundary value and the first derivative of the end-boundary. In Section 4, with the established results in the literature, we prove the superlinear local convergence for the reformulated system in Section 3. In Section 5, several numerical experiments are executed which validate the convergence rates derived in Section 4.

2 HJB equation

The HJB equation that we will discuss comes from papers [1, 7]. The solution will be the optimal value function which maximizes the discounted dividend payments given the current amount of capital by an optimal strategy. The system is simplified to a constrained problem with a homogeneous Dirichlet one-sided boundary condition. The approach works as well for nonhomogeneous Neumann boundary conditions. Details can be found in [8]. The constrained system with Dirichlet one-sided boundary condition has the following form:

$$\begin{cases} -mv(x) + \frac{\delta^2}{2} v(x) + \mu v_x(x) + \frac{\sigma^2}{2} v_{xx}(x) + \sup_{0 \leqslant \xi \leqslant \epsilon} \xi (1 - v_x(x)) = 0, \\ v(0) = 0, \\ v'(x) \geqslant 0 \text{ for } x > 0, \\ v''(x) \leqslant 0 \text{ for } x > 0. \end{cases} \tag{2.1}$$

where we assume m, δ, μ, σ and $\epsilon > 0$. The analytical solution for this system [2] is given by

$$v(x) = \begin{cases} \frac{\epsilon \eta}{m - \frac{\delta^2}{2}} \frac{e^{\theta x} - e^{\phi x}}{\eta(e^{\theta \hat{x}} - e^{\epsilon \hat{x}}) - \theta e^{\theta \hat{x}} + \phi e^{\phi \hat{x}}} & \text{if } x \leqslant \hat{x}; \\ \frac{\epsilon}{m - \frac{\delta^2}{2}} \left(1 + \frac{e^{\eta (x - \hat{x})} (\theta e^{\theta \hat{x}} - \phi e^{\phi \hat{x}})}{\eta(e^{\theta \hat{x}} - e^{\phi \bar{x}}) - \theta e^{\theta_1 \hat{x}} + \phi_1 e^{\phi \hat{x}}} \right) & \text{if } x > \hat{x} \end{cases} \tag{2.2}$$

where

$$\eta = \frac{\epsilon - \mu - \sqrt{(\epsilon - \mu)^2 + 2\sigma^2 (m - \frac{\delta^2}{2})}}{\sigma^2}$$

$$\theta = \frac{-\mu + \sqrt{\mu^2 + 2\sigma^2 (m - \frac{\delta^2}{2})}}{\sigma^2}$$

$$\phi = \frac{-\mu - \sqrt{\mu^2 + 2\sigma^2 (m - \frac{\delta^2}{2})}}{\sigma^2}$$

$$\hat{x} = \frac{\log(\frac{\phi^2 - \eta \phi}{\theta^2 - \eta \theta})}{\theta - \phi}$$

Next, we will present a numerical approach to approximate these solutions for the Dirichlet one-sided boundary conditions; the numerical approach for the Neumann case is similar.

3 Methods

We apply forward finite differences for the first derivatives and centered differences for the second derivatives. We assume a uniform mesh for simplicity of exposition. On $x \in [0, B]$, let $h = B/N$

be the mesh size for some $N > 1$ and $x_i = ih$ be the nodes. Then the HJB equation becomes

$$-mv(x_i) + \frac{\delta^2}{2} v(x_i) + \mu \frac{v(x_{i+1}) - v(x_i)}{h} + \frac{\sigma^2}{2} \frac{v(x_{i+1}) - 2v(x_i) + v(x_{i-1})}{h^2} + \sup_{0 \leqslant \xi \leqslant \epsilon} \left(\xi (1 - \frac{v(x_{i+1}) - v(x_i)}{h}) \right) = 0, \quad i = 1..N-1.$$

The addition of two boundary contitions would close the system, but there is only a one-sided boundary condition. The other two conditions are inequality constraints which are implicit on v. It is this feature, and the non-smooth term, that make the problem interesting from a numerical perspective. The inequality constraints may be similarly discretized in a straight-forward manner

$$v_x(x_i) = \frac{v(x_{i+1}) - v(x_i)}{h} \geqslant 0, \quad i = 0..N-1$$
$$v_{xx}(x_i) = \frac{v(x_{i+1}) - 2v(x_i) + v(x_{i-1})}{h^2} \leqslant 0, \quad i = 1..N-1.$$

The above are then coupled with the initial condition $v(x_0) = 0$.

3.1 Motivation for the shooting-like method

To our best knowledge, no numerical methods have been implemented to solve the above discretized problem. Additionally, there are two inequality constraints here, which differs from examples in the literature.

Herein we propose the combination of a semi-smooth, projected Newton method and a shooting-like iterative method to solve this problem. We will call the following algorithm *semismooth projected Newton with shooting-like method*. As a matter of fact, the projection steps appearing in these Algorithms take sweeping-like operations. Basically, it looks forward in the x direction and adjusts the state values at x_k, x_{k+1} from the current value at grid x_{k-1} according to the inequality constraints.

The classical shooting method [6] chooses some value as the guess of system's initial derivative and then shoots the trajectory (solves the resulting well-posed problem) to the other end of the boundary. By comparing that simulation with the given boundary data, improvements can be made eventually leading to the correct initial derivative and therefore the correct numerical approximation. However, this idea does not work in our problem since there is no information on the other boundary. Further, for this problem, it turns out that the numerical solution is extremely sensitive to the intial boundary slope, and thus the problem of determining it is ill-conditioned.

In order to overcome this difficulty, we propose a backward shooting (guess the end value) and forward adjustment method (sweep) to approximate the solution. We note that a more formal version of the algorithm will be introduced later to facilitate the resulting analysis. The proposed approach can be decomposed into two stages. In the first stage, we make a guess of the end boundary and shoot the trajectory of the state, which is driven by the HJB equation. In the next step, we adjust the trajectory by imposing the inequality constraints, proceeding forward in x direction, similar to the method of sweeps [13]. This naturally updates the end boundary value and leads to an iterative method which continues until convergence.

Shooting at the end-boundary value

In the first trial, we shoot at the end-boundary value and consider the Dirichlet boundary condition. The algorithm for implementation is below and the analogous one for Neumann condition only needs a few modifications.

Algorithm 1 (Dirichlet Condition):

A **Semi-smooth Newton**:
The nonsmooth term in the HJB equation is $\sup_{0\leqslant\xi\leqslant\epsilon}\xi(1-\frac{v(x_{i+1})-v(x_i)}{h})$.
Algorithm:

(1) At the kth step, let $v^k=(v^k(x_i))_{i=1}^N$ be the vector of solution candidate, where $\{x_i\}$ is the even partition of [0,B] with step size h.

(2) Compute the nonsmooth term using the current state v^k. For the interval $[x_i,x_{i+1}]$, we could build the following equivalent matrix form for the nonsmooth term,

$$I_i^k[\frac{1}{h}\begin{bmatrix}0 & -\epsilon & \epsilon\end{bmatrix}\begin{bmatrix}v^k(x_{i-1})\\ v^k(x_i)\\ v^k(x_{i+1})\end{bmatrix}+\epsilon]$$

where the function I_i^k is defined as: $I_i^k=\begin{cases}1, & \text{if } \frac{v^k(x_{i+1})-v^k(x_i)}{h}>1\\ 0, & \text{if } \frac{v^k(x_{i+1})-v^k(x_i)}{h}\leqslant 1\end{cases}$

B **Discretization** on the nonsmooth term:
Based on the previous computation, the full discretization of HJB equation will be:

$$\begin{bmatrix}\frac{\sigma^2}{2h^2} & \frac{\delta^2}{2}-m-\frac{\mu}{h}-\frac{\sigma^2}{h^2} & \frac{\mu}{h}+\frac{\sigma^2}{2h^2}\end{bmatrix}\begin{bmatrix}v^k(x_{i-1})\\ v^k(x_i)\\ v^k(x_{i+1})\end{bmatrix}+[\frac{I_i^k}{h}\begin{bmatrix}0 & -\epsilon & \epsilon\end{bmatrix}\begin{bmatrix}v^k(x_{i-1})\\ v^k(x_i)\\ v^k(x_{i+1})\end{bmatrix}+\epsilon]=0.$$

where $i=2,\ldots,N-1$. After the combination, we have the full matrix form:

$$\begin{bmatrix}1 & 0 & \ldots & & \ldots & 0\\ K & M-L_2^k & C+L_2^k & & & 0\\ 0 & \ddots & \ddots & \ddots & 0 & 0\\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots\\ 0 & \ldots & & K & M-L_{N-1}^k & C+L_{N-1}^k\\ 0 & \ldots & 0 & & \ldots & 1\end{bmatrix}\begin{bmatrix}v^k(x_1)\\ v^k(x_2)\\ \vdots\\ v^k(x_{N-1})\\ v^k(x_N)\end{bmatrix}=\begin{bmatrix}0\\ I_2^k\epsilon\\ \vdots\\ \vdots\\ I_{N-1}^k\epsilon\\ v^{k-1}(x_N)\end{bmatrix}$$

where $K=\frac{\sigma^2}{2h^2}, M=\frac{\delta^2}{2}-m-\frac{\mu}{h}-\frac{\sigma^2}{h^2}, C=\frac{\mu}{h}+\frac{\sigma^2}{2h^2}, L_i^k=I_i^k\frac{\epsilon}{h}$ and $v^{k-1}(x_N)$ is the guess of the end-boundary condition for this iteration.
We denote the left-hand side of the above equation as $\bar{F}(v^k)$.Let

$$A_k=\begin{bmatrix}1 & 0 & \ldots & & \ldots & 0\\ K & M-L_2^k & C+L_2^k & & & 0\\ 0 & \ddots & \ddots & \ddots & 0 & 0\\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots\\ 0 & \ldots & & K & M-L_{N-1}^k & C+L_{N-1}^k\\ 0 & \ldots & 0 & & \ldots & 1\end{bmatrix}$$

Update the current candidate solution to the new candidate:

Algorithm:

(a) Compute the matrix A_k and take it as a gradient for semi-smooth Newton method at the kth step.

(b) Update the current solution in the following way:

$$\bar{v}^k = v^k - A_k^{-1}\bar{F}(v^k)$$

where $\bar{F}(v^k)$ is the evaluation of the matrix form of the HJB equation at the kth step.

C **Pointwise projection** based on the constraints on the solution (value function):

- The value function $v(x)$ should be increasing and concave for all $x > 0$.
- The algorithm for checking the constraint:

(1) Let the current state of the candidate solution be $\bar{v}^k$.

(2) Compute the quantities: $\gamma_i^k = \bar{v}^k(x_{i+1}) - \bar{v}^k(x_i)$ for $1 \leqslant i \leqslant N-1$.

(3) Update the candidate solutions as follows:

$$\tilde{v}^k(x_{i+1}) = \begin{cases} \bar{v}^k(x_i) & \text{if } \gamma_i^k < 0 \\ \bar{v}^k(x_{i+1}) & \text{otherwise.} \end{cases}$$

(4) Denote the candidate solutions achieved in previous step as $\tilde{v}^k$.

(5) Compute the quantities: $\tilde{\gamma}_i^k = \tilde{v}^k(x_{i-1}) - 2\tilde{v}^k(x_i) + \tilde{v}^k(x_{i+1})$ for $2 \leqslant i \leqslant N-1$.

(6) Update the candidate solution in the following way:
For the points: x_{i-1}, x_i, x_{i+1},

$$v^{k+1}(x_{i+1}) = \begin{cases} 2\tilde{v}^k(x_i) - \tilde{v}^k(x_{i-1}) & \text{if } \gamma_i^k > 0 \\ \tilde{v}^k(x_{i+1}) & \text{if } \gamma_i^k \leqslant 0 \end{cases}$$

D **Criterion** to check the convergence: We compute the residual of each iteration in L^2 norm. The program stops once the residual is below some fixed level.

Shooting at the end derivative

We adjust the method a little bit with guessing the end derivative. Its algorithm therefore is almost the same as the previous one. But for the purpose of clear demonstration, we repeat the previous one with a few changes.

A **Semi-smooth Newton**:
The nonsmooth term in the HJB equation is $\sup_{0 \leqslant \xi \leqslant \epsilon} \xi(1 - \frac{v(x_{i+1}) - v(x_i)}{h})$.
Algorithm:

(1) At the kth step, let $v^k = (v^k(x_i))_{i=1}^N$ be the vector of solution candidate, where $\{x_i\}$ is the even partition of [0,B] with step size h.

(2) Compute the nonsmooth term using the current state v^k. For the interval $[x_i, x_{i+1}]$, we could build the following equivalent matrix form for the nonsmooth term,

$$I_i^k[\frac{1}{h}\begin{bmatrix}0 & -\epsilon & \epsilon\end{bmatrix}\begin{bmatrix}v^k(x_{i-1})\\ v^k(x_i)\\ v^k(x_{i+1})\end{bmatrix}+\epsilon]$$

where the function I_i^k is defined as: $I_i^k = \begin{cases}1, & \text{if } \frac{v^k(x_{i+1})-v^k(x_i)}{h} > 1\\ 0, & \text{if } \frac{v^k(x_{i+1})-v^k(x_i)}{h} \leqslant 1\end{cases}$

B **Discretization** on the nonsmooth term:
Based on the previous computation, the full discretization of HJB equation will be:

$$\begin{bmatrix}\frac{\sigma^2}{2h^2} & \frac{\delta^2}{2}-m-\frac{\mu}{h}-\frac{\sigma^2}{h^2} & \frac{\mu}{h}+\frac{\sigma^2}{2h^2}\end{bmatrix}\begin{bmatrix}v^k(x_{i-1})\\ v^k(x_i)\\ v^k(x_{i+1})\end{bmatrix}+[\frac{I_i^k}{h}\begin{bmatrix}0 & -\epsilon & \epsilon\end{bmatrix}\begin{bmatrix}v^k(x_{i-1})\\ v^k(x_i)\\ v^k(x_{i+1})\end{bmatrix}+\epsilon]=0$$

After the combination, we have the full matrix form:

$$\begin{bmatrix}1 & 0 & \cdots & & \cdots & 0\\ K & M-L_2^k & C+L_2^k & & & 0\\ 0 & \ddots & \ddots & \ddots & 0 & 0\\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & & K & M-L_{N-1}^k & C+L_{N-1}^k\\ 0 & \cdots & 0 & \cdots & -1 & 1\end{bmatrix}\begin{bmatrix}v^k(x_1)\\ v^k(x_2)\\ \vdots\\ v^k(x_{N-1})\\ v^k(x_N)\end{bmatrix}=\begin{bmatrix}0\\ I_2^k\epsilon\\ \vdots\\ \vdots\\ I_{N-1}^k\epsilon\\ V^{k-1}\end{bmatrix}$$

where $V^{k-1}=v^{k-1}(x_N)-v^{k-1}(x_{N-1})$, $K=\frac{\sigma^2}{2h^2}$, $M=\frac{\delta^2}{2}-m-\frac{\mu}{h}-\frac{\sigma^2}{h^2}$, $C=\frac{\mu}{h}+\frac{\sigma^2}{2h^2}$, $L_i^k = I_i^k\frac{\epsilon}{h}$ and $v^{k-1}(x_N)$ is the guess of the end-boundary condition for this iteration.
Again, we keep the left-hand side of the above equation as $\bar{F}(v^k)$.
Let

$$A_k=\begin{bmatrix}1 & 0 & \cdots & & \cdots & 0\\ K & M-L_2^k & C+L_2^k & & & 0\\ 0 & \ddots & \ddots & \ddots & 0 & 0\\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & & K & M-L_{N-1}^k & C+L_{N-1}^k\\ 0 & \cdots & 0 & \cdots & -1 & 1\end{bmatrix}$$

Update the current candidate solution to the new candidate:

Algorithm:

(a) Compute the matrix A_k and take it as the gradient for semi-smooth Newton method at the kth step.

(b) Update the current solution in the following way:

$$\bar{v}^k = v^k - A_k^{-1}\bar{F}(v^k)$$

where $\bar{F}(v^k)$ is the evaluation of the matrix form of the HJB equation at the kth step.

C **Pointwise projection** based on the constraints on the solution(value function):

- The value function $v(x)$ should be increasing and concave for all $x > 0$.
- The algorithm for checking the constraint:

(1) Let the current state of the candidate solution be $\bar{v}^k$.

(2) Compute the quantities: $\gamma_i^k = \bar{v}^k(x_{i+1}) - \bar{v}^k(x_i)$ for $1 \leqslant i \leqslant N-1$.

(3) Update the candidate solutions as follows:

$$\tilde{v}^k(x_{i+1}) = \begin{cases} \bar{v}^k(x_i) & \text{if } T_i^k < 0 \\ \bar{v}^k(x_{i+1}) & \text{otherwise.} \end{cases}$$

(4) Denote the candidate solutions achieved in previous step as $\tilde{v}^k$.

(5) Compute the quantities: $\tilde{\gamma}_i^k = \tilde{v}^k(x_{i-1}) - 2\tilde{v}^k(x_i) + \tilde{v}^k(x_{i+1})$ for $2 \leqslant i \leqslant N-1$.

(6) Update the candidate solution in the following way:
For the points: x_{i-1}, x_i, x_{i+1},

$$v^{k+1}(x_{i+1}) = \begin{cases} 2\tilde{v}^k(x_i) - \tilde{v}^k(x_{i-1}) & \text{if } \tilde{T}_i^k > 0 \\ \tilde{v}^k(x_{i+1}) & \text{if } \tilde{T}_i^k \leqslant 0 \end{cases}$$

D **Criterion** to check the convergence: We compute the residual of each iteration in L^2 norm. The program stops once the residual is below some fixed level.

3.2 Algorithm for a combined projection

In the previous section, we offer the methodology without the proof of convergence. In this section, we present a nearly equivalent formulation amenable to a convergence proof. As expected, and verified by the numerical demonstrations, local superlinear convergence is shown.

Suppose that $w = v'$. Then the problem (2.1) becomes

$$-mv + \frac{\delta^2}{2}v + \mu w + \frac{\sigma^2}{2}w_x + \sup_{0 \leqslant \xi \leqslant \epsilon} \{\xi(1-w)\} = 0.$$

Here we want to keep the discretization form we have established in the previous algorithm. We take the following for w and w_x at the grid point x_k: $w(x_k) = \frac{v(x_{k+1}) - v(x_k)}{h}$ for $0 \leqslant k \leqslant N-1$ and $w_x(x_k) = \frac{w(x_k) - w(x_{k-1})}{h} = \frac{v(x_{k+1}) - 2v(x_k) + v(x_{k-1})}{h^2}$ for $1 \leqslant k \leqslant N-1$.

In order to reduce the number of variables, combine v and w into a single variable $u = \begin{pmatrix} v \\ w \end{pmatrix}$. Here all the vectors are in column form. Thereafter we can rewrite the system dependent on the given boundary condition.

Dirichlet condition

$$\begin{cases} -mv(x_k) + \frac{\delta^2}{2}v(x_k) + \mu w(x_k) + \frac{\sigma^2}{2}w_x(x_k) + \sup_{0 \leqslant \xi \leqslant \epsilon} \{\xi(1 - w(x_k))\} = 0 \\ v(x_0) = 0 \\ Pu \geqslant 0 \end{cases}$$

Let $\bar{N} = 2N - 1$. Then u is $\bar{N}$-dimensional. P is the $\bar{N} \times \bar{N}$ projection matrix. Specifically, in this problem, we can write down the system in the following matrix form: for $k \geqslant 2$

$$\begin{cases} \bar{T}^k u^k = B^k \\ P^k u^k \geqslant 0 \end{cases}$$

Here we shoot the end derivative of u and the matrices $\bar{T}$, P and L will be as follows:

$$\begin{bmatrix}
-1 & 1 & 0 & \dots & -h & & \dots & & & 0 \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
 & & -1 & 1 & 0 & \dots & -h & \dots & & \vdots \\
1 & 0 & \dots & & \dots & & \dots & & & 0 \\
 & -m & D & \dots & E & M_2^k & \dots & & & \vdots \\
 & & -m & D & \dots & E & M_3^k & \dots & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\
 & & & & -m & D & \dots & E & & M_{N-1}^k \\
0 & & \dots & & \dots & & \dots & -1 & & 1
\end{bmatrix}
\begin{bmatrix} v^k(x_1) \\ v^k(x_2) \\ \vdots \\ \vdots \\ v^k(x_N) \\ w^k(x_1) \\ \vdots \\ \vdots \\ w^k(x_{N-1}) \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ B_2^k \\ B_3^k \\ \vdots \\ B_{N-1}^k \\ W^{k-1} \end{bmatrix}$$

$$\begin{bmatrix}
1 & 0 & & & & & & & 0 \\
-1 & 1 & 0 & \dots & & & \dots & & \vdots \\
 & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\
 & & -1 & 1 & 0 & \dots & & & 0 \\
 & & & & 1 & & \dots & & 0 \\
 & & & & 1 & -1 & 0 & \dots & 0 \\
 & & & & & \ddots & \ddots & \ddots & 0 \\
 & & & & & & & 1 & -1
\end{bmatrix}
\begin{bmatrix} v^k(x_1) \\ v^k(x_2) \\ \vdots \\ v^k(x_N) \\ w^k(x_1) \\ w^k(x_2) \\ \vdots \\ w^k(x_{N-1}) \end{bmatrix}
\geqslant
\begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Here $W^{k-1} = w^{k-1}(x_{N-1}) - w^{k-1}(x_{N-2})$, $D = \frac{\delta^2}{2}$, $E = -\frac{\sigma^2}{2h}$ and for $2 \leqslant l \leqslant N-1$, $M_l^k = \begin{cases} \mu + \frac{\sigma^2}{2h} - \xi & \text{if } w(x_l) \leqslant 1 \\ \mu + \frac{\sigma^2}{2h} & \text{if } w(x_l) > 1 \end{cases}$ and $B_l^k = \begin{cases} -\xi & \text{if } w^k(x_l) \leqslant 1 \\ 0 & \text{if } w^k(x_l) > 1 \end{cases}$.

This problem has a general linear constraint, but it is more reasonable to talk about this problem as a simple non-negative constraint (which will be discussed in the next section). For that purpose, in the k-th step, we take $y^k = P^k u^k$ and $U^k = \bar{T}^k (P^k)^{-1}$. Then this problem will be transformed to:

$$\begin{cases} U^k y^k = B^k \\ y^k \geqslant 0 \end{cases}$$

which is a (nonlinear) problem with a simple non-negative constraint (note, U and B depend on y via the non-smooth term).

Now the program works as follows:

Algorithm 3:

A **Semi-smooth Newton:**
From the discretization form, U^k is semismooth.

(1) Suppose that y^k is the current candidate solution.

(2) Update the solution with semismooth Newton step:

$$\bar{y}^{k+1} = y^k + (U^k)^{-1}(B^k - U^k y^k)$$

B **Projection:**
The constraint requires all the components of y to be non-negative. Therefore we take projection step as follows: for $1 \leqslant l \leqslant M$

$$(y^{k+1})_l = \begin{cases} (\bar{y}^{k+1})_l & \text{if } (\bar{y}^{k+1})_l \geqslant 0 \\ 0 & \text{if } (\bar{y}^{k+1})_l < 0 \end{cases}$$

C **Transfer back:**
Update the candidate solution from the original problem: $u^{k+1} = P^{k+1} y^{k+1}$.

D **Criterion:**
Compute the residual L^2 norm in each iteration of y. Stops if the criterion is met and take part v of the ultimate iterate u as the closest numerical solution. Otherwise go back to step A.

4 Convergence analysis

For each iteration, the problem (2.1) after discretization is formulated in the following way:

$$\begin{cases} \bar{F}(v) = 0 \\ P_1 v \geq 0 \\ P_2 v \geq 0 \end{cases} \tag{4.1}$$

where $\bar{F}, P_1$ and P_2 are nonsingular matrices. The following theorems and definitions are stated from the indicated literature for ease of reference.

Theorem 4.1 (Local convergence of semi-smooth Newton, [14])**.** *Suppose that $v^* \in \mathbb{R}^n$, $\bar{F}(v^*) = 0$ and $\bar{F}$ is semi-smooth at v^*. The generalized derivative of $\bar{F}(v)$ is nonsingular at v^*. Choose the initial iterate v_0 sufficiently close to v^* so that there exists $\epsilon > 0$ such that $v_0 \in B(v^*, \epsilon)$. Then under the semi-smooth Newton iteration, the generated sequence $\{v_k\}$ converges to v^* superlinearly fast, i.e.,*

$$||v_{k+1} - v^*|| = o||v_k - v^*|| \text{ as } k \to \infty.$$

Theorem 4.2 (Local convergence of projected Newton, [11])**.** *Suppose that v^* satisfies $\bar{F}(v^*) = 0$ and all constraints, and that the derivative of $\bar{F}(v)$ is locally Lipschitz continuous around v^*. Let the initial iterate v_0 be close to v^* and both have the same active set of constraints (constraints for which the state value is on the boundary of the admissible interval), i.e., $A(v_0) = A(v^*)$. Then the iteration generated by the projected Newton method converges quadratically to v^*, i.e,*

$$||v_{k+1} - v^*|| \leqslant K||v_k - v^*||^2 \text{ for some } K > 0, \text{ and } k = 0, 1, \ldots$$

Definition 4.1. *Let $JF(x)$ be the classical Jacobian matrix of the partial derivatives for F at x and $D_f(x)$ be the collection of points where F is differentiable in the neighborhood of x. Then, $\partial F(x)$ is the generalized Jacobian of F at x given by ([4]),*

$$\partial F(x) = co \left\{ \lim_{\substack{x_i \to x \\ x_i \in D_f(x)}} JF(x_i) \right\}.$$

Definition 4.2. *F is semismooth at x ([16]) if F is locally Lipschitz continuous at x and for any $s \in \mathbb{R}^n$,*

$$\lim_{\substack{V \in \partial F(x+ts') \\ s' \to s, t \downarrow 0}} \left\{ V s' \right\} \quad \text{exists.}$$

In other words, $\partial F(x)$ is the convex hull of all the derivatives on the differential points around x.

Lemma 4.1. *Suppose that v^* is the solution to the problem (4.1). Then the function $\bar{F}(v)$ is semi-smooth around v^*.*

Proof. Suppose that the discretization number is N. So $v^* \in \mathbb{R}^N$. Let the direction vector be $s' = [s_i']_{i=1}^N$ and $t > 0$. The Jacobian of $\bar{F}(v)$ at $v^* + ts'$ is

$$A' = \begin{bmatrix} 1 & -1 & \dots & & \dots & 0 \\ K & M - L_2' & C + L_2' & & & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & & K & M - L_{N-1}' & C + L_{N-1}' \\ 0 & \dots & 0 & & \dots & 1 \end{bmatrix}$$

where

$$L_i' = \begin{cases} \frac{\epsilon}{h}, & \text{if } \frac{v^*(x_{i+1}) + ts_{i+1}' - v^*(x_i) - ts_i'}{h} > 1 \\ 0, & \text{if } \frac{v^*(x_{i+1}) + ts_{i+1}' - v^*(x_i) - ts_i'}{h} \leqslant 1 \end{cases}$$

Here let $s' \to s$. Then as $s_i' \to s_i$ and $s_{i+1}' \to s_{i+1}$, L_i' won't change any more since the relation between $v^*(x_{i+1}) + ts_{i+1}'$ and $v^*(x_i) + ts_i'$ become more apparent when s' gets closer to s. That is to say, the Jacobian matrix A' at $v^* + ts'$ stays the same when $s' \to s$. Therefore,

$$\lim_{\substack{A' \in \partial \bar{F}(v^*(x)+ts') \\ s' \to s, t \downarrow 0}} \left\{ A' s' \right\} \text{ exists}$$

So $\bar{F}(v)$ is semismooth at v^*. □

4.1 Semismooth equation with the simple box constraint

We keep the notations suggested in (Page 87 of [11]). $\mathcal{A}(v)$ and $\mathcal{I}(v)$ are the active and inactive sets at v respectively. $\mathcal{P}$ will denote the projection into constrained region. Here we review the convergence of semismooth Newton method with simple box constraint from the perspective of active-set strategy ([10]&[15]). Similar to the reduced Hessian introduced in (Page 89 of [11]), here define the reduced Jacobian matrix. To get a better view of solving problem (4.1), we consider the simplified problem discussed in [15]:

$$\begin{cases} \bar{F}(v) = 0 \\ b_1 \leqslant v \leqslant b_2 \end{cases} \tag{4.2}$$

where b_1, b_2 are N-dimensional vectors. This is called the problem of semi smooth equation with the box constraint.

The projection $\mathcal{P}$ and active set $\mathcal{A}(v)$ can be denoted as follows. Specifically, the projection step in this situation works in the following way:

$$\mathcal{P}((v)_i) = \begin{cases} (b_1)_i & \text{if } (v)_i \leqslant (b_1)_i \\ (b_2)_i & \text{if } (v)_i \geqslant (b_2)_i \\ (v)_i & \text{if } (b_1)_i < (v)_i < (b_2)_i \end{cases}$$

As for the active set $\mathcal{A}(v)$, $\mathcal{A}(v) = \{i = 1, 2, \ldots, N | (v)_i \geqslant (b_2)_i \text{ or } (v)_i \leqslant (b_1)\}$. Then the inactive set $\mathcal{I}(v)$ is given as: $\mathcal{I}(v) = \{1, 2, 3, \ldots, N\} \setminus \mathcal{A}(v)$. Then the projection on $\mathcal{A}(v)$ is defined as:

$$\mathcal{P}_{\mathcal{A}(v)}((u)_i) = \begin{cases} (u)_i & \text{if } i \in \mathcal{A}(v) \\ 0 & \text{if } i \notin \mathcal{A}(v) \end{cases}$$

Similarly, we can define the projection on the set $\mathcal{I}(v)$.

Definition 4.3. *Suppose that A is the Jacobian matrix to $\bar{F}$ at v. Then the reduced Jacobian R is given as follows:*

$$R_{ij} = \begin{cases} \delta_{ij} & \text{if } i \in \mathcal{A}(v) \text{ or } j \in \mathcal{A}(v) \\ A_{ij} & \text{otherwise} \end{cases}$$

Theorem 4.3 (Local convergence of semi-smooth Projected Newton)**.** *Suppose that the problem (4.1) assumes that $\bar{F}(v)$ is semi-smooth around v^*, $\bar{F}(v^*) = 0$ and generalized derivative $\partial \bar{F}$ is nonsingular at v^*. Let the initial iterate v_0 be close to v^* and $\mathcal{A}(v_0) = \mathcal{A}(v^*)$. Then the iterations generated by the Semi-smooth Projected Newton method for problem (4.2) converge superlinearly:*

$$||v_{k+1} - v^*|| = o||v_k - v^*|| \text{ as } k \to \infty$$

Proof. Consider the $(k+1)$-th iteration v_{k+1}. Assume $A_k \in \partial \bar{F}(v_k)$ and denote the reduced Jacobian matrix to A_k by R_k. In the problem (4.1), v is restricted by the conditions: $b_1 \leqslant v \leqslant b_2$. It seems that we project the iterate v_k to the region bounded by these conditions.

$$v_{k+1} = \mathcal{P}(v_k - R_k^{-1} \bar{F}(v_k)) \tag{4.3}$$

By the definition of semi-smoothness for $\bar{F}$ at v^*,

$$\bar{F}(v_k) = \bar{F}(v^*) + A_k(v_k - v^*) + E_k \tag{4.4}$$

where $E_k = o(||v_k - v^*||)$.
By the assumption that $\mathcal{A}(v^*) = \mathcal{A}(v_0)$, we can say that

$$\mathcal{A}(v^*) = \mathcal{A}(v_k) = \mathcal{A}(v_{k+1})$$

Alternatively, $\mathcal{I}(v_k) = \mathcal{I}(v^*)$. So,

$$\mathcal{P}_{\mathcal{I}(v_k)}(v_k - v^*) = v_k - v^* \text{ and } \mathcal{P}_{\mathcal{A}(v_k)}(v_k - v^*) = 0$$

Applying projection $\mathcal{P}_{\mathcal{I}(v_k)}$ on both sides of equation (4.4),

$$\begin{aligned} \mathcal{P}_{\mathcal{I}(v_k)}(\bar{F}(v_k)) &= \mathcal{P}_{\mathcal{I}(v_k)}(A_k(v_k - v^*)) + \mathcal{P}_{\mathcal{I}(v_k)}(E_k) \\ &= \mathcal{P}_{\mathcal{A}(v_k)}(v_k - v^*) + \mathcal{P}_{\mathcal{I}(v_k)}(A_k(v_k - v^*)) + \mathcal{P}_{\mathcal{I}(v_k)}(E_k) \\ &= R_k(v_k - v^*) + \mathcal{P}_{\mathcal{I}(v_k)}(E_k) \end{aligned}$$

Then after applying R_k^{-1},

$$\mathcal{P}_{\mathcal{I}(v_k)}(R_k^{-1}\bar{F}(v_k)) = (v_k - v^*) + \mathcal{P}_{\mathcal{I}(v_k)}(R_k^{-1}E_k)$$

Since $\partial\bar{F}(v^*)$ is nonsingular, then $||R_k^{-1}||$ is bounded.
Denote $\mathcal{P}_{\mathcal{I}(v_k)}(R_k^{-1}E_k)$ by J_k. Then,

$$||J_k|| = o||v_k - v^*|| \text{ as } k \to \infty$$

Applying $\mathcal{P}_{\mathcal{I}(v_k)}$ on both sides of equation (4.3),

$$\begin{aligned}
\mathcal{P}_{\mathcal{I}(v_k)}(v_{k+1}) &= \mathcal{P}_{\mathcal{I}(v_k)}\mathcal{P}(v_k - R_k^{-1}\bar{F}(v_k)) \\
&= \mathcal{P}\mathcal{P}_{\mathcal{I}(v_k)}(v_k - R_k^{-1}\bar{F}(v_k)) \\
&= \mathcal{P}(v_k - (v_k - v^*) - J_k) = \mathcal{P}(v^* - J_k)
\end{aligned}$$

Note that $\mathcal{P}_{\mathcal{A}(v_k)}(v_{k+1}) = \mathcal{P}_{\mathcal{A}(v^*)}(v^*)$.

$$\begin{aligned}
\mathcal{P}_{\mathcal{A}(v_k)}(v_{k+1} - v^*) + \mathcal{P}_{\mathcal{I}(v_k)}(v_{k+1} - v^*) &= \mathcal{P}(v^* - \mathcal{P}_{\mathcal{I}(v_k)}(v^*) - J_k) \\
\mathcal{P}(v_{k+1} - v^*) &= \mathcal{P}(v^* - \mathcal{P}_{\mathcal{I}(v_k)}(v^*) - J_k) \\
\mathcal{P}_{\mathcal{I}(v_k)}(v_{k+1} - v^*) &= \mathcal{P}_{\mathcal{I}(v_k)}(v^* - \mathcal{P}_{\mathcal{I}(v_k)}(v^*) - J_k) \\
\mathcal{P}_{\mathcal{I}(v_k)}(v_{k+1} - v^*) &= \mathcal{P}_{\mathcal{I}(v_k)}(-J_k) \\
\mathcal{P}(v_{k+1} - v^*) &= \mathcal{P}(-J_k)
\end{aligned}$$

where we use the fact that $\mathcal{P}_{\mathcal{A}(v_k)}(J_k) = 0$.
The above implies that $||v_{k+1} - v^*|| \leqslant ||J_k|| = o||v_k - v^*||$. This shows the local superlinear convergence. □

4.2 Semismooth equation with the general linear constraint

In the previous section, the problem of a semismooth equation under the simple box constraint has been discussed. Now we are going to look at this problem with a general linear constraint, which could be considered as the extension of that problem. Here is the formulation:

$$\begin{cases} \bar{F}(v) = 0 \\ b_1 \leqslant Pv \leqslant b_2 \end{cases} \tag{4.5}$$

where b_1, b_2 are still N-dimensional vectors and P is an $N \times N$ nonsingular matrix.

Recall that the projected Newton-like iteration for the previous problem (4.2) updates in the following way (**Algorithm 4**):

(1) Initialize the initial guess for the solution: v_0.

(2) For step k, compute the Jacobian matrix A_k for this problem (4.2) based on the current iterate v_k.

(3) Update v_k to v_{k+1} through Projected Newton-type method:

  (a) $v_{k+1}^* = v_k + A_k^{-1}(-\bar{F}^k(v_k))$

(b) $v_{k+1} = \mathcal{P}(v^*_{k+1})$

(4) If the stopping criterion is not satisfied, go back to step 2.

(5) If the criterion is met, stop.

However, the projection (b) in step 3 is not working in the problem (4.5). Therefore it needs some modification on the algorithm. To our best knowledge of the literature in this topic, the paper ([3]) mentioned the idea for general box constraint in the Newton-type optimization, but it didn't address the semismooth problem or solving nonlinear differential equations. The basic strategy to deal with that is through the change of variable.

Consider the new vector $z = Pv$. With a little adjustment, the problem (4.5) can be converted to the standard semismooth equation with simple box constraint (4.2). In fact, if we denote $\bar{G} = \bar{F}P^{-1}$,

$$\begin{cases} \bar{G}(z) = 0 \\ b_1 \leqslant z \leqslant b_2 \end{cases}$$

Then the projected Newton-type algorithm above would work well with this problem. Once the stop criterion is triggered, the new update z_{k+1} would be used to to recover the candidate solution by $v_{k+1} = P^{-1}z_{k+1}$ for the original problem (4.5).

Algorithm 5:

(1) Initialize v_0. Then transfer it to $z_0 = Pv_0$.

(2) Update z:

(a) $\bar{z}_{k+1} = u_k + PA_k^{-1}(-\bar{G}^k(z_k))$.

(b) $z_{k+1} = \mathcal{P}(\bar{z}_{k+1})$

(3) If the stopping criterion is not met, go back to step (2).

(4) If the stopping criterion is met, take z_{k+1} and transfer it back to $v_{k+1} = P^{-1}z_{k+1}$.

But in order to reach the superlinear convergence of this algorithm with general linear constraint, we need the following results:

Lemma 4.2. *Suppose that the true solution for the new problem (4.5) is z^*. Then the function $\bar{G}(z)$ is semismooth around the true solution z^*.*

Proof. In Lemma 4.1, we know that any element of the convex hull $\partial\bar{F}(v^*)$ is in the form of A' and it is semismooth. Since $\bar{G} = \bar{F}P^{-1}$, then any element of convex hull $\partial\bar{G}(z^*)$ will be in the form of $A'P^{-1}$.

It has been shown that $\lim_{\substack{A'\in\partial\bar{F}(z^*+ts') \\ s'\to s, t\downarrow 0}} \{A's'\}$ exists. It is not hard to verify that once the gradient direction s' has been chosen, the matrix $A'P^{-1}$ will stay the same at $z^* + ts'$. Then the limit will still exist:

$$\lim_{\substack{(A'P^{-1})\in\partial\bar{G}(z^*+ts') \\ s'\to s, t\downarrow 0}} \left\{(A'P^{-1})s'\right\} \quad \text{exists}$$

That is to say that the function $\bar{G}(z)$ is semismooth around z^*. □

After we establish the semismoothness of the function $\bar{G}(z)$ around the true solution, we can state the superlinear convergence of the new algorithm:

Theorem 4.4 (Local convergence of problem (4.5))**.** *Suppose that $\bar{G}(z)$ is semismooth around the true solution z^* of problem (4.5) and the generalized derivative $\partial\bar{G}$ is nonsingular around z^*. Then the iterations v_k generated with **Algorithm 5** will converge superlinearly.*

Proof. The result of Theorem 4.3 implies that:

$$||z_{k+1} - z^*|| = o||z_k - z^*|| \quad \text{as } k \to \infty$$

On the other hand, $\bar{G}(z^*) = 0 \to \bar{F}(P^{-1}z^*) = 0$. If we say $v^* = P^{-1}z^*$, then v^* is the true solution of $\bar{F}(v) = 0$. Note that the iterations of z_k is obtained after projection. We then have the following arguments.

Since $z_{k+1} = Pv_{k+1}, z_k = Pv_k, z^* = Pv^*$,

$$\begin{cases} ||z_{k+1} - z^*|| = ||P(v_{k+1} - v^*)|| \\ ||z_k - z^*|| = ||P(v_k - v^*)|| \end{cases}$$

As the matrix P is nonsingular, it is a bounded nontrivial linear transformation. Suppose v_k and v_{k+1} are not equal to v^*. Otherwise, the error would be zero and it is not necessary to continue the iterations afterwards. In particular we can find $\lambda_k, \beta_k > 0$ for $k \in \mathbb{Z}$ such that:

$$\beta_k||v_k - v^*|| \leqslant ||P(v_k - v^*)|| \leqslant \lambda_k||v_k - v^*||$$

On the other hand, we know that when k is large enough, $||z_{k+1} - z^*|| = o||z_k - z^*||$. Then there exist some $M > 0, \alpha > 1$ s.t $||z_{k+1} - z^*|| = M||z_k - z^*||^\alpha$. Then,

$$\begin{aligned} ||v_{k+1} - v^*|| &\leqslant \frac{||P(v_{k+1} - v^*)||}{\beta_{k+1}} = \frac{||z_{k+1} - z^*||}{\beta_{k+1}} \\ &= \frac{M||z_k - z^*||^\alpha}{\beta_{k+1}} \leqslant \frac{M\lambda_k^\alpha||v_k - v^*||^\alpha}{\beta_{k+1}} \end{aligned}$$

The above inequality implies that $||v_{k+1} - v^*|| = o||v_k - v^*||$ and the local convergence rate is superlinear. □

The convergence rate of the problem (4.5) turns out to be superlinear, which is within the expectation, since the transfer step changes the space by a little. However, the above result implies that the Algorithm 3 produces Newton's iterations with superlinear convergence. That is to say the solution of constrained HJB equation stated in the very beginning can be approximated by this algorithm with superlinear convergence rate. This will be stated as a Corollary.

Corollary 4.1. *Suppose that v^* is the true solution of the equation (2.1) with Dirichlet(Neumann) condition and inequality constraints. The Algorithm 3 will enforce the Newton's iteration to converge to v^* superlinearly fast.*

5 Numerical results

Since this problem essentially is free-boundary, two different shooting-like numerical methods have been tried here with insufficient boundary data. The basic idea is that we shoot out a point as the guess for the end boundary. Then we adjust the end-boundary point through the inequality constraints on both first and second derivatives. Once we update the end boundary, then we take Newton-type step to find a better candidate solution to the problem. After enough iterations, the equation and constraints will be simultaneously satisfied. The numerical results show that the accuracy and the shape of the solution will depend on the end-boundary guess. On the other hand, due to two different types of inequality constraints imposed on HJB equation (2.1), the system is converted to the one with two general linear constraints. In the section on general-linear constraints, the matrix P on the constraint is called the transfer acting on the domain. It essentially gives us the choice to take Newton steps either on the original space or the transfer space.

5.1 Shooting-like method with no transfer

Shooting at the end-boundary point: $v^0(x_N)$
Dirichlet conditions

1 Let the initial guess of the end-boundary condition $v^0(x_N)$ be small. Choose parameters as follows:

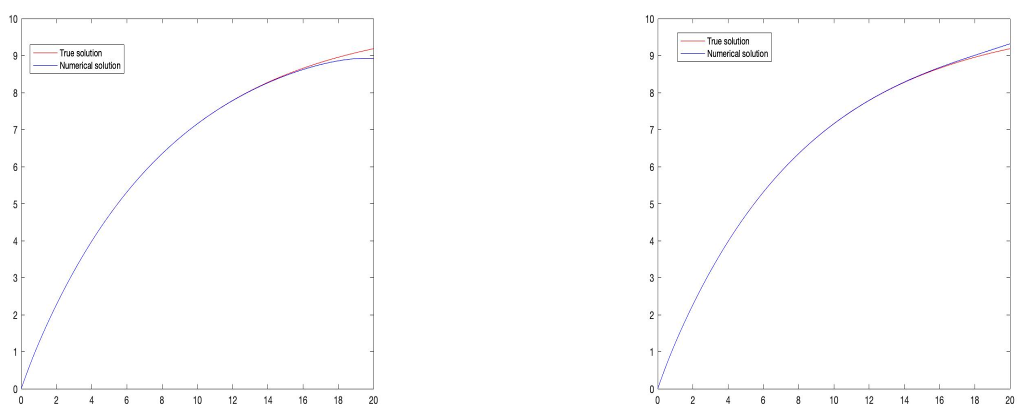
$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v^0(x_N) = 0, B = 20, h = 0.01$$

The true solutions shows that the end-boundary condition at x=20 is $v(20) = 9.19$ and the choice of end boundary $v^0(x_N) < v(20)$

2 Let the initial guess of the end-boundary condition $v^0(x_N)$ be large. Choose parameters as follows:

$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v^0(x_N) = 20, B = 20, h = 0.01$$

The true solutions shows that the end-boundary condition at x=20 is $v(20) = 9.19$ and the choice of end boundary $v^0(x_N) > v(20)$
The comparisons between numerical and true solutions are illustrated in Figure 1.



(a) Comparison between the true solution and numerical solution when $v^0(x_N) = 0$

(b) Comparison between the true solution and numerical solution when $v^0(x_N) = 20$

Figure 1: Figures for the numerical results in Dirichlet conditions (no transfer)

Neumann condition

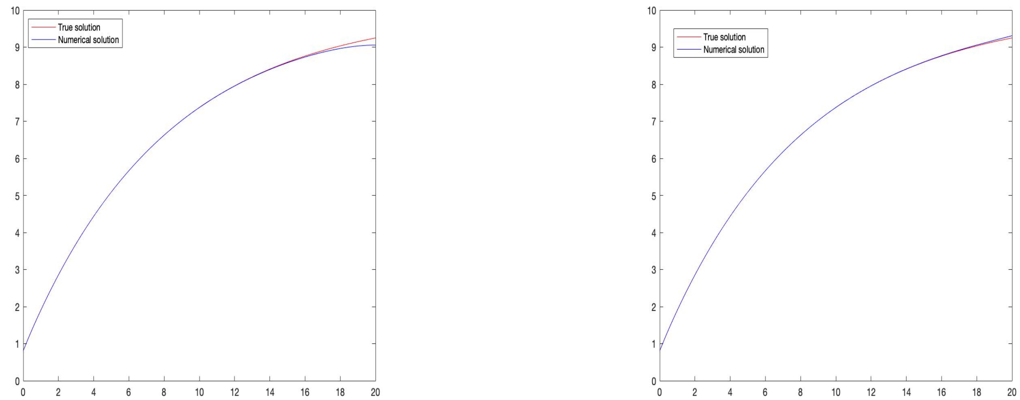
1 Let the initial guess of the end-boundary condition $v^0(x_N)$ be small.
Choose parameters as follows:

$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v^0(x_N) = 0, I = 20, h = 0.01$$

2 Let the initial guess of the end-boundary condition $v^0(x_N)$ be large.
Choose parameters as follows:

$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v^0(x_N) = 20, I = 20, h = 0.01$$

The comparisons between true and numerical solutions for both cases are shown in Figure 2.



(a) Comparison between the true solution and numerical solutions when $v^0(x_N) = 0$

(b) Comparison between the true solution and numerical solutions when $v^0(x_N) = 20$

Figure 2: Figures of numerical results in Neumann conditions (no transfer)

Shooting at the end derivative: $v_x^0(x_N)$
Dirichlet condition

1 Let the initial guess of the end-boundary condition $v_x^0(x_N)$ be small. Choose parameters as follows:

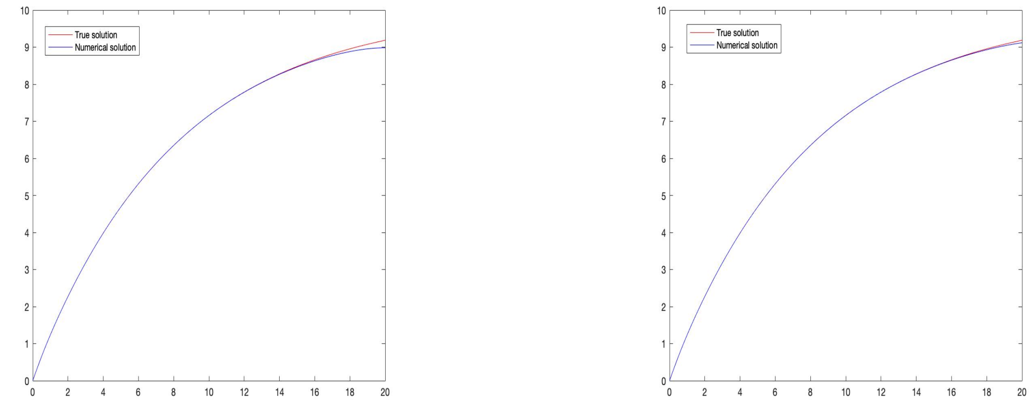
$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v_x^0(x_N) = 0, B = 20, h = 0.01$$

The true solutions shows that the end derivative at x=20 is $v_x(20) = 0.1024$ and the choice of the end boundary $v_x^0(x_N) < v_x(20)$

2 Let the initial guess of the end-boundary condition $v_x^0(x_N)$ be large. Choose parameters as follows:

$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v_x^0(x_N) = 1, I = 20, h = 0.01$$

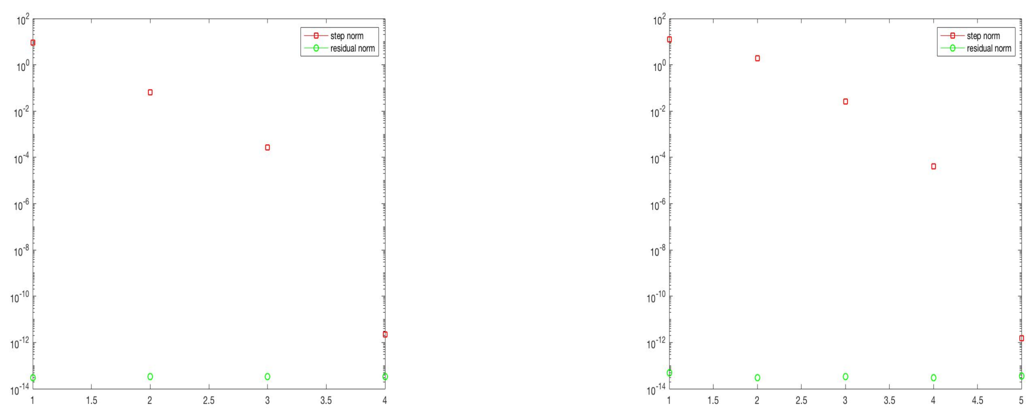
The comparisons between true and numerical solutions for both cases are shown in Figure 3. The results on the convergence are illustrated in a separate figure (Figure 4).



(a) the comparison between true and numerical solution when $v_x^0(x_N) = 0$

(b) the comparison between true and numerical solution when $v_x^0(x_N) = 1$.

Figure 3: Comparisons between true and numerical solutions in Dirichlet condition (no transfer)



(a) the convergences of step norm and residual norm when $v_x^0(x_N) = 0$

(b) the convergences of step norm and residual norm when $v_x^0(x_N) = 1$

Figure 4: Convergence test in Dirichlet condition (no transfer)

Neumann condition

1 Let the initial guess of the end-boundary condition $v_x^0(x_N)$ be small. Choose parameters as follows:

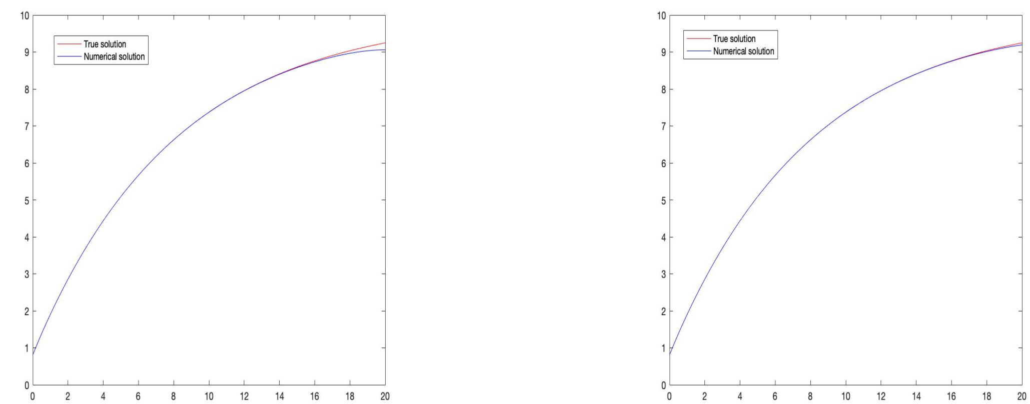
$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v_x^0(x_N) = 0, B = 20, h = 0.01$$

The true solutions shows that the end-boundary condition at x=20 is $v_x(20) = 0.1024$ and the choice of the end boundary $v_x(x_N) < v_x(20)$

2 Let the initial guess of the end-boundary condition $v_x(x_N)$ be large. Choose parameters as follows:

$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, v_x^0(x_N) = 1, I = 20, h = 0.01$$

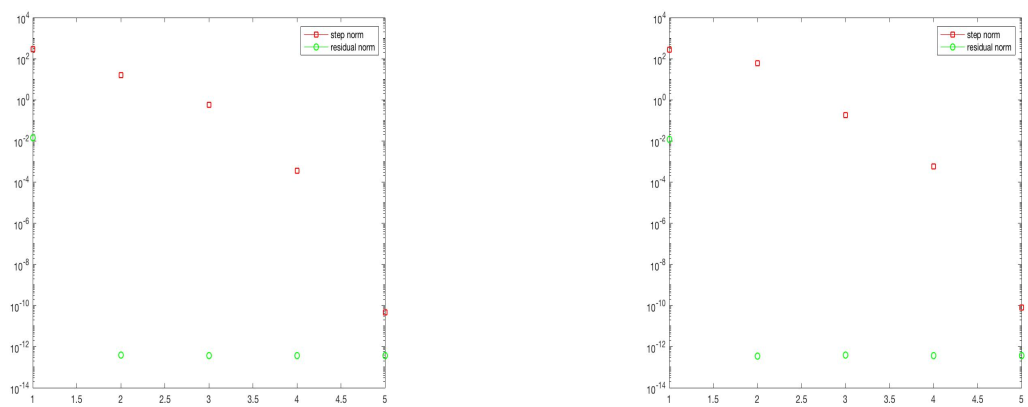
The comparisons between true and numerical solutions for both cases are shown in Figure 5. The results on the convergence are illustrated in separate figures (Figure 6).



(a) the comparison between true and numerical solution when $v_x^0(x_N) = 0$

(b) the comparison between true and numerical solution when $v_x^0(x_N) = 1$

Figure 5: Comparisons between true and numerical solutions in Neumann condition (no transfer)



(a) the convergences of step norm and residual norm when $v_x^0(x_N) = 0$

(b) the convergences of step norm and residual norm when $v_x^0(x_N) = 1$

Figure 6: Convergence test in Neumann condition (no transfer)

5.2 Shooting-like method with transfer

From the proof of Theorem 4.4, it seems that the convergences of simple box and general linear constraints imply each other. Here we will show that after taking the transfer from the original space to the transfer space, the Semi-smooth Newton's iterations in the transfer space still have the superlinear convergence rate. Here we follow the notations in Section 3.3.2 (Algorithm for a combined projection).

Shooting at the end derivative $u_x^0(x_{\bar{N}})$
Dirichlet condition

1 Let the initial guess of the end derivative $u_x^0(x_{\bar{N}})$ be small.
Choose the parameters as follows:

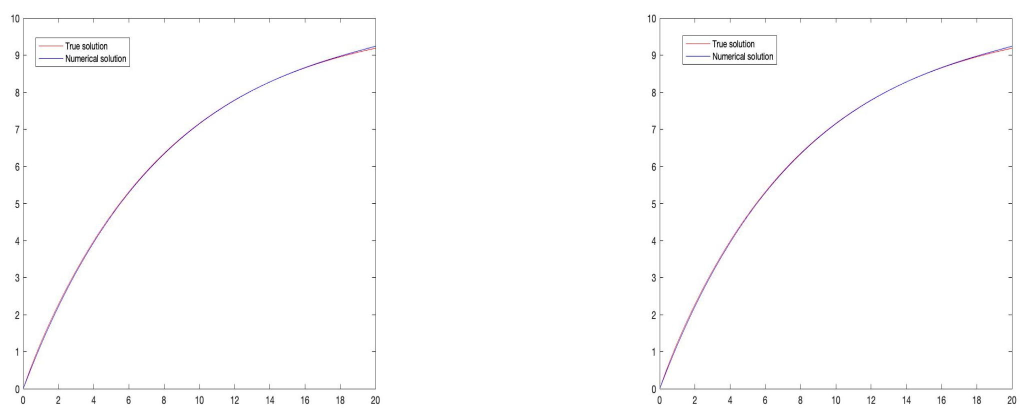
$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, u_x^0(x_{\bar{N}}) = 0, B = 20, h = 0.01$$

Comparison between numerical and true solutions are shown in Figure 7. The convergence of step norm and residual norm are shown in Figures 8.

2 Let the initial guess of the end derivative $u_x^0(x_{\bar{N}})$ be large.
Choose the parameters as follows:

$$m = 1, \delta = 1, \mu = 2, \sigma = 4, \epsilon = 5, u_x^0(x_{\bar{N}}) = 1, B = 20, h = 0.01$$

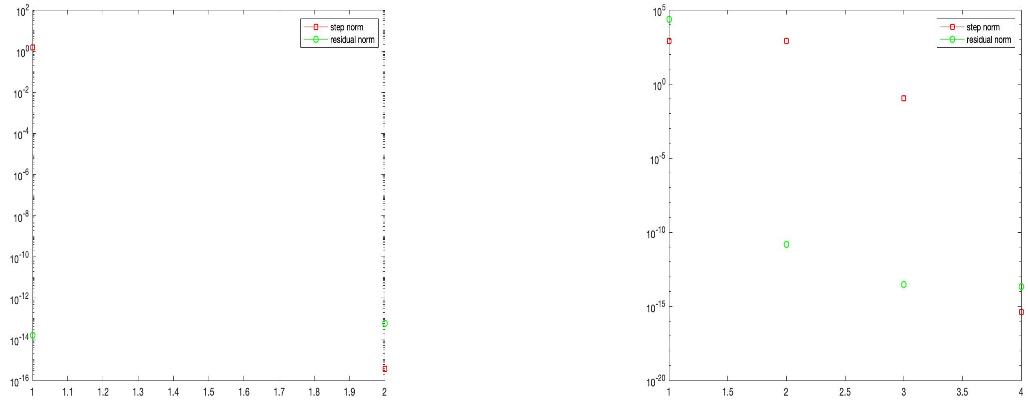
It turns out that the convergence rate is complicated in some cases, although the results presented here demonstrate the superlinear convergence rate. We believe it is for the reason that the initial guess is not close enough to the true solution for those tricky cases. In Figure 9, we will see all the iterations it takes to approach the true solution. The comparison between the true and numerical solutions are shown in Figure 7 and the convergences of step norm and residual norm are shown in Figure 8.



(a) the comparison between true and numerical solution when $u_x^0(x_{\bar{N}}) = 0$

(b) the comparison between true and numerical solution when $u_x^0(x_{\bar{N}}) = 1$

Figure 7: Comparisons between true and numerical solutions in Dirichlet condition(transfer)



(a) the convergences of step norm and residual norm when $u_x^0(x_{\bar{N}}) = 0$

(b) the convergences of step norm and residual norm when $u_x^0(x_{\bar{N}}) = 1$

Figure 8: Convergence test in Dirichlet condition(transfer)

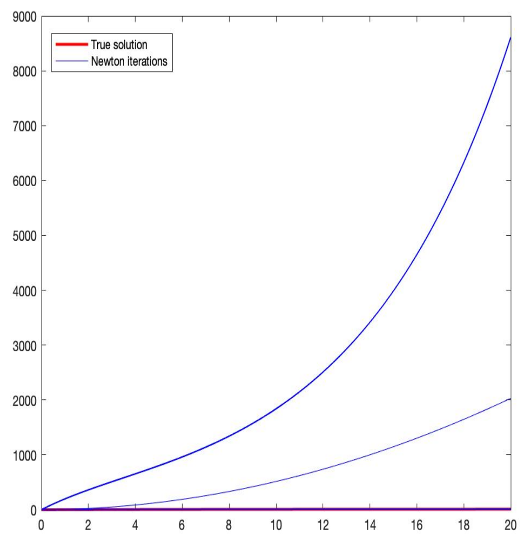


Figure 9: Newton iterations shooting large initial end derivative $u_x^0(x_{\bar{N}})$ in Dirichlet condition

6 Conclusion

This paper investigates a nonsmooth HJB equation with one-side free boundary and inequality constraints with a novel numerical approach. This approach combines the semi-smooth Projected Newton with the so-called shooting-like method in order to deal with the free boundary. It turns out that this method is efficient in solving this kind of problem with superlinear local convergence and can be applied to more general constrained variational inequalities including first-order and second-order constraints. However, there are many open questions left in this direction. It needs some rigorous proofs regarding the convergence of algorithms which are not involved with transfer space. In the experiments, it finds that "shooting at the end derivative" always converges faster than "shooting at the end boundary point" and the shooting-like method with no transfer still converges superlinearly fast. However, currently the proofs for them are not available. On the other hand, the proof demonstrated in Section 4 indicates that there is possibility to extend the general linear constraint to a few nonlinear cases. Above all, this paper suggests a numerical method to attempt some free boundary problems which have implicit state constraints like those on first and second derivatives.

At last, we need to point out the reason why we apply this method to a problem with an exact solution. Firstly, we have not seen any numerical method dealing with this type of constrained free-boundary nonlinear problem before. Secondly, it is easier for us to tell how well this method works after comparing its result with the exact solution. Future work is to expand the method to apply to two dimensional free boundary problems such as related to insurance modeling with stochastic prices [9].

7 Acknowledgement

We would like to thank first author's co-advisor Professor Edward C. Waymire for his help and comments, both of which greatly improved the quality of this paper. Both authors want to thank the funding support from Bonneville Power Administration on project TIP342.

References

[1] Søren Asmussen and Michael Taksar. Controlled diffusion models for optimal dividend pay-out. *Insurance: Mathematics and Economics*, 20(1):1–15, 1997.

[2] Søren Asmussen and Michael Taksar. Controlled diffusion models for optimal dividend pay-out. *Insurance: Mathematics and Economics*, 20(1):1–15, 1997.

[3] Dimitri P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization*, 20(2):221–246, 1982.

[4] Frank H. Clarke. *Optimization and nonsmooth analysis*, volume 5. SIAM, 1990.

[5] Wendell H. Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[6] Eugene Isaacson and Herbert Bishop Keller. *Analysis of numerical methods*. Courier Corporation, 2012.

[7] M. Jeanblanc-Picqu and A. N. Shiryaev. Optimization of the flow of dividends. *Russian Mathematical Surveys*, 50(2):257, 1995.

[8] Huanqun Jiang. *Stochastic and Numerical analysis on optimization problems*. Oregon State University, 2019.

[9] Huanqun Jiang and Nathan L. Gibson. A numerical approach for a constrained Hamiltion-Jacobi-Bellman equation arising from financial insurance. *Submitted*, 2019.

[10] C. T. Kelley and E. W. Sachs. Solution of optimal control problems by a pointwise projected Newton method. *SIAM Journal on Control and Optimization*, 33(6):1731–1757, 1995.

[11] Carl T. Kelley. *Iterative methods for optimization*, volume 18. SIAM, 1999.

[12] Harold Kushner and Paul G. Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2013.

[13] Suzanne Lenhart and John T Workman. *Optimal control applied to biological models*. CRC press, 2007.

[14] Liqun Qi and Jie Sun. A nonsmooth version of Newton's method. *Mathematical programming*, 58(1-3):353–367, 1993.

[15] Liqun Qi, XJ Tong, and D. H. Li. Active-set projected trust-region algorithm for box-constrained nonsmooth equations. *Journal of Optimization Theory and Applications*, 120(3):601–625, 2004.

[16] Michael Ulbrich. *Semismooth Newton methods for variational inequalities and constrained optimization problems in function spaces*, volume 11. SIAM, 2011.