Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics
(IMACM)

Preprint BUW-IMACM 16/13

Igor Kossaczký, Matthias Ehrhardt and Michael Günther

Piecewise fixed policy timestepping schemes for Hamilton-Jacobi-Bellman equations

July 22, 2016

http://www.math.uni-wuppertal.de

Piecewise fixed policy timestepping schemes for Hamilton-Jacobi-Bellman equations

Igor Kossaczký[a,*], Matthias Ehrhardt[a], Michael Günther[a]

[a] *Lehrstuhl für Angewandte Mathematik und Numerische Analysis, Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Gaußstrasse 20, 42119 Wuppertal, Germany*

Abstract

In this work we generalize the piecewise constant policy timestepping (PCPT) scheme for solving Hamilton-Jacobi-Bellman (HJB) equations to a class of piecewise fixed policy timestepping (PFPT) schemes. We show that any PFPT scheme by can be seen as an PCPT scheme for some related equations. Based on this, we establish a convergence result using the same results as in the paper of Forsyth and Labahn from 2007. We propose a new member of these class of PFPT schemes, the so called piecewise predicted policy timestepping (PPPT) scheme, that is in many cases significantly faster than PCPT scheme. We solve numerically a HJB equation resulting from a mean-variance optimal investment problem with this PPPT scheme and compare it with the solution provided by the classical implicit and by the PCPT scheme. Using the PPPT scheme, a significant speed-up by the same level of precision in contrast to the classical and the PCPT scheme is observed in this case.

Keywords: Hamilton-Jacobi-Bellman equation, PCPT scheme, PPPT scheme, PFPT schemes, Consistency, mean-variance optimal investment problem

*Corresponding author

Email addresses: kossaczky@math.uni-wuppertal.de (Igor Kossaczký), ehrhardt@math.uni-wuppertal.de (Matthias Ehrhardt), guenther@math.uni-wuppertal.de (Michael Günther)

URL: http://www-num.math.uni-wuppertal.de/en/amna/people/igor-kossaczky-msc.html (Igor Kossaczký), http://www-num.math.uni-wuppertal.de/~ehrhardt/ (Matthias Ehrhardt), http://www-num.math.uni-wuppertal.de/~guenther/ (Michael Günther)

1. Introduction

The Hamilton-Jacobi-Bellman (HJB) equation is a nonlinear partial differential equation (PDE), arising in many applications, particularly those dealing with some optimal control. As other nonlinear equations, HJB equations may also not have solution in a classical sense. An appropriate concept of solutions, the viscosity solutions, was proposed in 1992 by Crandall, Ishi and Lions [1]. Some basic facts on viscosity solutions can be found in [2]. As it is in general not possible to find viscosity solutions analytically, numerical schemes where developed for this purpose. Because of the special structure of viscosity solutions, this schemes should have besides the properties of stability and consistency also the monotonicity property, in order to converge (see [3]). For classical monotone finite difference schemes for solving the HJB equation we refer to [4], [5]. Explicit methods are not monotone, therefore implicit methods should be used. However, in order to solve the optimization problem in the HJB equation, policy iteration is needed in each time layer in an implicit method. Another possibility is to use piecewise constant policy timestepping (PCPT) schemes, presented for example in [6], [4], [7]. The idea of these schemes is solving several PDEs with a constant policy in each time layer, instead of using a policy iteration strategy. In our paper, we embedded this PCPT method to a larger class of piecewise fixed policy timestepping (PFPT) schemes. While the idea of solving several PDEs in each time layer remains, we don't restrict us to constant policies. We showed that any PFPT method can be seen as PCPT method for some different but closely related HJB equation. We used this equivalence to establish the convergence of PFPT methods, using results on convergence of the PCPT method from [6], [4]. As a new member of the class of PFPT schemes, we present here the so-called piecewise predicted policy timestepping (PPPT) scheme. The main idea of this method is to search for some prediction of optimal control on a coarse grid, and then use this prediction as a benchmark for searching an optimal control on a fine grid. This reduces the computational costs significantly, as we do not test as many controls in each time layer as in the case of the PCPT method.

Now, we will introduce the structure of this paper. In Section 2, we present the basic theory of Barles and Souganidis [3] on convergence of solutions of the numerical schemes for nonlinear systems to viscosity solution. In Section 3, we define the HJB equation and its discretization. In Section 4, we describe classical explicit and implicit schemes for solving HJB equations. In Section 5, we introduce the class of PFPT schemes, and examine its relationship to PCPT schemes. In Section 6, we prove the convergence of PFPT schemes, using the approach from [6], [4], and the equivalence between PCPT and PFPT methods. In Section 7 we introduce the PPPT scheme. As this scheme often leads to a significant speed-up in computational time, it can be seen as the main result of this paper. In Section 8, we compare approximation error and computational time of classical implicit, PCPT and PPPT method by solving mean-variance optimal investment problem presented in [7] and [8]. We show that a properly used PPPT method is always superior to PCPT and classical implicit method in this example, and by larger number of nodes, it is 4.7 time faster than the PCPT and 8 times faster than the classical implicit method, on the same level of accuracy. We also estimate the experimental order of convergence and test parallel computing implementations of our methods.

2. Convergence of numerical schemes for nonlinear systems

Let U denote some suitable function space. Let us define some **nonlinear differential operator F**

$$F : U \to \mathbb{R}, \quad V(x) \to FV(x).$$

We suppose there exists a **viscosity solution** (see [1]) of the equation $FV(x) = 0$, and denote this solution simply by $V(x)$. It may be hard, or even impossible to find the viscosity solution analytically, therefore we define the **discrete approximation scheme**

$$Gv(x) = G\left(v(x), v(x + b_1 h), v(x + b_2 h), \dots, v(x + b_n h)\right), \tag{1}$$

where $v(x)$, $x \in \mathbb{R}^K$ is defined as (possibly) multidimensional function with suitable properties, $b_i \in \mathbb{R}^K$, $i = 1, 2, ..., n$ and $h \in \mathbb{R}^+$.

Let us consider the system of sets called **discretized domains**

$$X_h = \{x_i \in \mathbb{R}^K | i = 1, 2, \dots N_h\}, \tag{2}$$

defined for different values of h, which is often referred as **step-size**.

Definition 1 (Numerical scheme)**.** *The system of equations $Gv(x) = 0$ with $x \in X_h$ depending on a parameter h is called numerical scheme.*

Our numerical scheme is well-defined, if it possess a unique solution. We will assume that this condition is met for any feasible h. By $v(x)$, we will understand an approximation of the solution of $FV(x) = 0$ computed by solving system of equations $Gv(x) = 0$, $x \in X_h$. In order to distinguish between approximations with different h, we will often denote $v(x)$ as $v_h(x)$.

Definition 2 (Monotonicity)**.** *A discrete approximation scheme $Gv(x) = G\big(v(x), v(x + b_1 h), v(x + b_2 h), \dots, v(x + b_n h)\big)$ is monotone, if the function G is non-increasing in $v(x + b_i h)$ for $b_i \neq 0$, $i = 1, \dots, n$.*

Definition 3 (Consistency)**.** *The scheme $G\phi(x) = G(\phi(x), \phi(x + b_1 h), \phi(x + b_2 h), \dots, \phi(x + b_n h))$ is a consistent approximation of $FV(x)$, in x, if $\lim_{h\to 0} |F\phi(x) - G\phi(x)| = 0$, for any smooth test function $\phi(x)$. We say it is consistent of order $n > 0$, if $|F\phi(x) - G\phi(x)| = \mathcal{O}(h^n)$ for any smooth test function $\phi(x)$.*

A scheme is consistent on a numerical domain, if it is consistent in all points of this numerical domain. In such case we will call the scheme consistent.

Definition 4 (Stability)**.** *The numerical scheme defined by the system of equation $Gv_h(x) = 0$, $x \in X_h$ with solution $v_h(x)$ is stable, if there exist some constant C so that $\|v_h(x)\|_\infty < C$, $\forall h > 0$.*

The next Theorem of Barles and Souganidis in [3] is the key for proving convergence of a numerical scheme approximating a nonlinear PDE:

Theorem 1 (Barles-Souganidis). *If the equation $FV(x) = 0$ satisfies the strong uniqueness property (see [3]) and if the numerical scheme $Gv_h(x) = 0$, $x \in X_h$ approximating equation $FV(x) = 0$ is monotone, consistent and stable, its solution $v_h(x)$ converges locally uniformly to the solution $V(x)$ of $FV(x) = 0$ with $h \to 0$.*

The strong uniqueness property [3] is a property of the problem and and not of the numerical scheme. Therefore, we will simply assume that our problems possess this property without actually proving it.

3. Discretising Hamilton Jacobi Bellman equations

Here, we will be concerned with the convergence of numerical schemes of a nonlinear PDE of the form $FV(x) = 0$ which is called Hamilton-Jacobi-Bellman equation.

Definition 5 (HJB equation). ***Hamilton-Jacobi-Bellman equation*** *is PDE of the form*

$$\frac{\partial V}{\partial t} - \max_{\theta \in \Theta} \left(\mathcal{L}_{s,t,\theta} V + d(s,t,\theta) \right) = 0 \tag{3}$$

with a family of ***elliptic*** *operators $\mathcal{L}_{s,t,\theta}$ defined as*

$$\mathcal{L}_{s,t,\theta} V = \sum_{i=1}^{K} \sum_{j=1}^{i} a_{i,j}(s,t,\theta) \frac{\partial^2 V}{\partial s_i \partial s_j} + \sum_{i=1}^{K} b_i(s,t,\theta) \frac{\partial V}{\partial s_i} + c(s,t,\theta) V,$$

where $s \in \mathbb{R}^K$, $t \in \mathbb{R}^+$ and Θ is called control set and its solution is a function $V(s,t) : S \times [0,T] \to \mathbb{R}$, $S \subset \mathbb{R}^K$, $T > 0$.

Remark 1. *Equations of the form* (3)*, with the maximum operator replaced by minimum, supremum or infimum operator are also called Hamilton-Jacobi-Bellman equations.*

In our settings, the control set Θ is defined as finite set $\Theta = \{\theta_1, \theta_2, \ldots, \theta_J\}$ and J represents the number of possible controls. Moreover, we suppose that all elements of Θ are real numbers and that it holds $\theta_1 < \theta_2 < \cdots < \theta_J$. If we have in some problem setting the control set as interval, we can discretize it to get Θ of the required form.

We see that the variable x is split into two variables **space** s and **time** t. Moreover, the solution $V(s,t)$ implies also optimal control function $\theta(s,t) = \arg\max_{\theta\in\Theta}\left(\mathcal{L}_{s,t,\theta}V + d(s,t,\theta)\right)$. For solving the HJB equation (3), an initial or terminal condition is necessary, and often also boundary conditions are supplied.

Now, let us suppose we want to solve the HJB equation on a rectangular domain $X = S \times [0,T]$, $S = [a_1,b_1] \times [a_1,b_1] \times \cdots \times [a_K,b_K]$.

To solve this HJB equation (3) we define the discretized domain $X_h = S_h \times T_h$ $S_h = S_{h,1} \times S_{h,2} \times \cdots \times S_{h,K}$, where $S_{h,i} = \{s_{i,j} \in [a_i,b_i] | j = 1,2,\dots,N_{h,i}\}$ and $T_h = \{t_i \in [0,T] | i = 1,2,\dots,M_h\}$. We suppose $s_{i,j} < s_{i,k}$ and $t_j < t_k$ for $j < k$, and $s_{i,j+1} - s_{i,j} < C_1 h$, $t_{j+1} - t_j < C_2 h$ $\forall i,h$ for some constants C_1, C_2. For simplicity we can write S_h as

$S_h = \{s_i | s_{\sum_{j=1}^{K} i_j N_j^{(j-1)}} = s_{1,i_1} \times s_{2,i_2}, \times \cdots \times s_{K,i_K}, s_{j,i_j} \in S_{h,j}, j = 1,2,\dots,K\}$. Then s_i's are defined for $i = 1,2,\dots,N_h$ with $N_h = \prod_{i=1}^{K} N_{h,i}$. We will call $S_h \times t_j$, $t_j \in T_h$ the **j-th time layer**, and $\Delta_j t = t_{j+1} - t_j$ the **j-th time-step**. Approximation of the solution of the HJB equation (3) $V(s,t)$ is denoted as $v(s,t)$. We will denote $v(s_i,t_j)$ as v_i^j, $\theta(s_i,t_j)$ as θ_i^j and $d(s_i,t_j,\theta_i^j)$ as $d_i^j(\theta)$.

At first, we will introduce the discrete scheme $L_{i,j,\theta}(v_1^j, v_2^j, \dots, v_{N_h}^j)$ approximating the elliptic operator $\mathcal{L}_{s,t,\theta}v(s_i,t^j)$:

$$L_{i,j,\theta}(v_1^j, v_2^j, \dots, v_{N_h}^j) = \sum_{k=1}^{N_h} a_k(i,j,\theta_i^j)v_k^j = \langle A_i^j(\theta), v^j \rangle. \tag{4}$$

Here $A_i^j(\theta)$ denotes N_h-dimensional vector $(a_1(i,j,\theta_i^j), a_2(i,j,\theta_i^j), \dots, a_{N_h}(i,j,\theta_i^j))$. For simplicity, we will often denote $L_{i,j,\theta}(v_1^j, v_2^j, \dots, v_{N_h}^j)$ as $L_{i,j,\theta}v^j$. Now, we will impose on the discretization of the elliptic operator two properties, that are essential for convergence of the schemes using this operator:

Property 1 (Consistency of elliptic operator approximation)**.** *We suppose, that* $L_{i,j,\theta}(v_1^j, v_2^j, \dots, v_{N_h}^j)$ *defined as* (4) *is consistent approximation of order* $k > 0$ *of the operator* $\mathcal{L}_{s_i,t_j,\theta}v(s_i,t_j)$*, that means* $|\mathcal{L}_{s_i,t_j,\theta}\phi_i^j - L_{i,j,\theta}(\phi_1^j, \phi_2^j, \dots, \phi_{N_h}^j)| = \mathcal{O}(h^k)$*, where* $\phi_i^j = \phi(s_i,t_j)$ *for any smooth test function* $\phi(s,t)$*, any control* θ

and for any $i = 1, 2, \ldots, N_h, j = 1, 2, \ldots, M_h$.

4. Classic implicit and explicit scheme

Finite difference schemes are widely used for solving HJB equations. We describe here the classical explicit and implicit scheme used for example in [4], [5].

We approximate the HJB equation (3) in points s_i, t_{j+1} with discrete scheme in the form $Gv(x) = 0$:

$$\frac{v_i^{j+1} - v_i^j}{\Delta_j t} - \max_{\theta_i^* \in \Theta} \left(L_{\theta,i,*}(v_1^*, v_2^*, \ldots, v_{N_h}^*) + d_i^*(\theta_i^j) \right) = 0 \tag{5}$$

For $*$, we set j in case of explicit scheme, and $j+1$ in case of implicit scheme. Multiplying (5) with $\Delta_j t$ and using (4) we get

$$v_i^{j+1} - v_i^j - \Delta_j t \left(\langle A_i^*(\hat{\theta}_i), v^* \rangle + d_i^* \right) = 0 \tag{6}$$

with the optimal control

$$\hat{\theta}_i = \arg \max_{\theta_i^* \in \Theta} \left(\langle A_i^*(\theta_i^*), v^* \rangle + d_i^*(\theta_i^*) \right) \quad i = 1, 2, \ldots, N_h, \tag{7}$$

Typically, values v_i^1 in the first time layer $X_{h,1}$ are known as initial condition.

4.1. Boundary conditions

We say, the node (s_i, t_j) is a boundary node, if for $s_i = s_{1,i_1} \times s_{2,i_2}, \times \cdots \times s_{K,i_K}$ exists such integer b, $1 \leq b \leq K$ that $i_b = 1$ or $i_b = N_{h,b}$. Often we do not have enough data to construct the scheme in such boundary node in the form of (6). Therefore, we set the value in the boundary node directly as **Dirichlet boundary condition**: $v_i^{j+1} = F_i^{j+1}$. This leads to an approximation scheme on the boundary which is also of the type $Gv(x) = 0$:

$$v_i^{j+1} - v_i^j - f_i^j = 0, \tag{8}$$

where $f_i^j = F_i^{j+1} - F_i^j$.

4.2. Matrix form of the equation

Joining equations for points on the boundary (8) together with equations for points in the inner domain (6), we can write the system of equations for one time layer in the matrix form:

$$v^{j+1} - v^j - \Delta_j t \left(A^*(\hat{\theta}) v^* + d^*(\hat{\theta}) + f^j \right) = 0, \tag{9}$$

where v^* is column vector with elements v_i^*, $A^*(\theta)$ is matrix with i-th row $A_i^*(\theta_i)$ if (6) holds in node (s_i, t_{j+1}) and zero-row if Dirichlet BC is set for this node, $d^*(\theta)$ is column vector with elements $d_i^*(\theta_i)$ for inner nodes and 0 for boundary nodes, and f^j is vector with elements f_i^j for boundary nodes and 0 for inner nodes. V The vector θ (resp. $\hat{\theta}$) is the so-called **control vector** and it's i-th element is control variable used in i-th node. Then, the solution of the explicit scheme $(* = j)$ in the $(j+1)$ time layer will be

$$v^{j+1} = (I + \Delta_j t A^{j+1}(\hat{\theta}))v^j + \Delta_j t d^j(\hat{\theta}) + f^j \tag{10}$$

and solution of explicit scheme $(* = j+1)$ will be

$$v^{j+1} = (I - \Delta_j t A^{j+1}(\hat{\theta}))^{-1}(v^j + \Delta_j t d^{j+1}(\hat{\theta}) + f^j) \tag{11}$$

where I is identity matrix.

We see, that for solving equations in one time layer we need only values from the previous time layer. This fact allow us to construct a simple algorithm to solve system of equations describing our numerical scheme:

1. Solution v^1 in nodes (s_i, t_1) is determined by initial condition.
2. for $j = 1$ to $M - 1$
   - Solve (10) , (7) $(* = j$, explicit scheme)
   - Or solve (11) , (7) $(* = j+1$, implicit scheme)

   end.

For implicit method, a policy iteration algorithm is needed to find optimal $\hat{\theta}$ (see [4]). As a policy, we understand here an N_h-dimensional control vector

$\bar{\theta}$, where the i-th element $\bar{\theta}_i$ is control that is used in node x_i of the current time-layer. The idea of policy iteration algorithm is to compute new (better) policies for the same time-layer in successive steps, until some level accuracy is achieved. Explicit schemes does not need a policy iteration, however, they are not monotone, which can harm the convergence of the method.

Let us note, that in each time layer, we need to solve an optimization problem (7), in order to find an optimal control vector $\hat{\theta}$. Elements of this control vector are from the finite control set Θ. We will search for the optimal control simply by trying all possible choices (brute force approach).

5. Piecewise fixed policy timestepping methods

To overcome the difficulty of policy iteration algorithm, piecewise constant policy timestepping (PCPT) method were developed (see [4], [7], [6]). Here, we will describe a new class of piecewise fixed policy timestepping (PFPT) methods. It will become clear that PCPT method is a member of the class of PFPT methods, but we will also show that PFPT methods can be seen as PCPT methods for a reformulated HJB equation. This equivalence will provide us with a convergence result for the whole class of PFPT method, since the convergence for PCPT is already proven.

Let $\{\theta^z(s,t), (s,t) \in S \times [0,T] | z = 1, 2, \dots, Z\}$ be a set of Z so-called control functions. Then, in j-th time layer we will solve Z PDEs with fixed N_h-dimensional control vectors $\bar{\theta}^{j,z}$ with i-th element defined as $\bar{\theta}^{j,z}_i = \theta^z(s_i, t_{j+1})$. The following algorithm makes this approach clear.

5.1. PFPT method:

1. INPUT: Initial condition v^1, Dirichlet BC's, Set of control functions $\{\theta^z(s,t), (s,t) \in S \times [0,T] | z = 1, 2, \dots, Z\}$
2. Solution v^1 in nodes (s_i, t_1) is determined by initial condition.
3. for $j = 1$ to $M - 1$
   (a) for $z = 1, 2, \dots, Z$:

- Define $\bar{\theta}^{j,z}$: $\bar{\theta}_i^{j,z} = \theta^z(s_i, t_{j+1})$ $(i \in \{1, 2, \dots, N_h\})$
- Find $v^{j+1,z}$ by solving

$$v^{j+1,z} - v^j - \Delta_j t \left(A^{j+1}(\bar{\theta}^{j,z}) v^{j+1,z} + d^{j+1}(\bar{\theta}^{j,z}) + f^j \right) = 0$$

(b) for $i = 1, 2, \dots, N_h$:

- $v_i^{j+1} = \max_{z \in \{1,2,\dots,Z\}} v_i^{j+1,z}$

end.

In node s_i, t_{j+1}, the PFPT scheme can be described by the equation

$$\frac{v_i^{j+1} - \max_{z \in \{1,2,\dots,Z\}} \left((I - \Delta_j t A^{j+1}(\bar{\theta}^{j,z}))^{-1} (v^j + \Delta_j t d^{j+1}(\bar{\theta}^{j,z}) + f^j) \right)_i}{\Delta_j t} = 0. \tag{12}$$

We see, that the scheme is in the form $Gv(x) = 0$.

As we want to solve a HJB equation with optimal control in each time-space point from the set Θ, we will demand that the control functions satisfy the following property:

Property 2. *We assume,* $\forall (s, t, z) \in S \times [0, T] \times \{1, 2, \dots, Z\}$*:* $\theta^z(s, t) \in \Theta$.

5.2. PCPT method

Now, if we set $Z = J$ and $\theta^z(s, t) \equiv \theta_z$ where $\theta_z \in \Theta$, then $\bar{\theta}^{j,z} = \theta_z \mathbf{1}$ where $\mathbf{1}$ is N_h-dimensional vector of ones, we get PCPT method from [4]. PCPT method is simply PFPT method using constant control functions with function values equal to all possible controls $\theta \in \Theta$.

Often, in many time layers we do not need to solve Z different PDEs, as more control vectors $\bar{\theta}^{j,z}$ will be equal to each other. This will be also the case of specific PFPT method presented later, and it's the main advantage in contrast to PCPT method.

5.3. PFPT method regarded as PCPT method for related HJB equation

We showed that the PCPT method can be defined as a PFPT method. However, it is also possible to show that the PFPT method can be seen as a PCPT method for another closely related HJB equation.

At first we need to define so-called **restricted control set**

$$\tilde{\Theta}(s,t) = \{\theta | \exists z \in \{1,2,\ldots,Z\} : \theta = \theta^z(s,t)\}, (s,t) \in S \times [0,T] \tag{13}$$

and **restricted HJB equation**:

$$\frac{\partial V}{\partial t} - \max_{\theta \in \tilde{\Theta}(s,t)} \left(\mathcal{L}_{s,t,\theta} V + d(s,t,\theta)\right) = 0 \tag{14}$$

From Property 2 follows, that $\forall (s,t) \in S \times [0,T]$: $\tilde{\Theta}(s,t) \subset \Theta$. The restricted HJB equation doesn't look like HJB equation according to Definition 5, because its set depends on space and time. However, by defining

$$\tilde{\mathcal{L}}_{s,t,z} V = \sum_{i=1}^{K} \sum_{j=1}^{i} \tilde{a}_{i,j}(s,t,z) \frac{\partial^2 V}{\partial s_i \partial s_j} + \sum_{i=1}^{K} \tilde{b}_i(s,t,z) \frac{\partial V}{\partial s_i} + \tilde{c}(s,t,z) V$$

$$\tilde{a}_{i,j}(s,t,z) = a_{i,j}(s,t,\theta^z(s,t)) \quad , \quad \tilde{b}_i(s,t,z) = b_i(s,t,\theta^z(s,t))$$
$$\tilde{c}(s,t,z) = c(s,t,\theta^z(s,t)) \quad , \quad \tilde{d}(s,t,z) = d(s,t,\theta^z(s,t))$$

we get $\tilde{\mathcal{L}}_{s,t,z} V = \mathcal{L}_{s,t,\theta^z(s,t)} V$ and we can rewrite (14) as

$$\frac{\partial V}{\partial t} - \max_{z \in \{1,2,\ldots,Z\}} \left(\tilde{\mathcal{L}}_{s,t,z} V + \tilde{d}(s,t,z)\right) = 0 \tag{15}$$

which is already HJB equation in its standard form. We will refer to equation (15) in this form as to **related HJB equation**. Finally, for clarity, we will often refer to HJB equation (3) as to **original HJB equation**

As $L_{i,j,\bar{\theta}_i^{j,z}}(v_1^j, v_2^j, \ldots, v_{N_h}^j)$ is consistent approximation of $\mathcal{L}_{s_i,t_j,\bar{\theta}_i^{j,z}} v_i^j$ it is also consistent approximation of $\tilde{\mathcal{L}}_{s_i,t_j,z} v_i^j$. However, by defining $\tilde{a}_k(i,j,z) = a_k(i,j,\bar{\theta}_i^{j,z})$ we can rewrite (4) into form where z will represent control parameter:

$$\begin{aligned} \tilde{L}_{i,j,z}(v_1^j, v_2^j, \ldots, v_{N_h}^j) &= \sum_{k=1}^{N_h} \tilde{a}_k(i,j,z) v_k^j \\ = \sum_{k=1}^{N_h} a_k(i,j,\bar{\theta}_i^{j,z}) v_k^j &= L_{i,j,\bar{\theta}_i^{j,z}}(v_1^j, v_2^j, \ldots, v_{N_h}^j) \end{aligned}$$

Then we can define $\tilde{A}_i^j(z)$ by $\langle \tilde{A}_i^j(z), v^j \rangle = \sum_{k=1}^{N_h} \tilde{a}_k(i,j,z) v_k^j$ and $\tilde{d}_i^j(z) = \tilde{d}(s_i, t_j, z)$. Now, we define $\tilde{A}^j(\bar{z})$ and vector $\tilde{d}^j(\bar{z})$ in the same manner as in section 4.2 ($\bar{z}$ is control vector with elements $\bar{z}_i \in \{1, 2, \ldots, Z\}$). Let us note that after such construction it holds $\tilde{A}^j(z\mathbf{1}) = A^j(\bar{\theta}^{j,z})$ and $\tilde{d}^j(z\mathbf{1}) = d^j(\bar{\theta}^{j,z})$. Now we constructed whole discretization of equation (15) and we can solve it with numerical methods. The next theorem shows that solving a HJB equation with PFPT method is identical to solving related HJB equation with PCPT method.

Theorem 2 (PCPT representation of PFPT method)**.** *PFPT scheme with set of control functions $\{\theta^z(s,t), (s,t) \in S \times [0,T] | z = 1, 2, \ldots, Z\}$ for solving HJB equation* (3) *on domain $S \times [0,T]$ is identical to PCPT method for solving related HJB equation* (15) *on the same domain.*

Proof. PCPT method is PFPT method with the set of all possible constant control function. Therefore, in our case the set of control function for solving (15) will be set of constant functions $\{f^z(s,t) = z | z = 1, 2, \ldots, Z\}$. Then, control vectors will be constant $z\mathbf{1}$, and following algorithm in section 5.1, the method will end up with approximation that will be solution of equations of the form

$$
\begin{aligned}
&v^{j+1,z} - v^j - \Delta_j t \left(\tilde{A}^{j+1}(z\mathbf{1}) v^{j+1,z} + \tilde{d}^{j+1}(z\mathbf{1}) + f^j \right) = 0, \\
&v_i^{j+1} = \max_{z \in \{1,2,\ldots,Z\}} v_i^{j+1,z}.
\end{aligned}
$$

However, as $\tilde{A}^j(z\mathbf{1}) = A^j(\bar{\theta}^{j,z})$ and $\tilde{d}^j(z\mathbf{1}) = d^j(\bar{\theta}^{j,z})$ the above equations are exactly identical to those from Section 5.1 with set of control functions $\theta^z(s,t)$. □

We proved that PFPT method applied on original HJB equation provide us with the same approximation of solution as PCPT method applied on related HJB equation. Let us for a while suppose that this is somehow "good" approximation of the solution of related HJB equation. Then, it is also good approximation of the solution of restricted HJB equation (since this is only related equation written in different way). Now, we are interested, if it is also "good" approximation of the original HJB equation. In order to be able to say this, we state a condition that should be fulfilled:

Property 3 (Feasibility). *Let* $\hat{\theta}(s,t) = \arg\max_{\theta\in\Theta}\left(\mathcal{L}_{s,t,\theta}V + d(s,t,\theta)\right)$, *where* V *is viscosity solution of the original HJB equation* (3). *We say that PFPT method defined by set of control functions* $\{\theta^z(s,t),(s,t)\in S\times[0,T]|z=1,2,\ldots,Z\}$ *is **feasible**, if* $\forall(s,t)\in S\times[0,T]$, $\hat{\theta}(t,s)\in\tilde{\Theta}(s,t)$, *where* $\tilde{\Theta}(s,t)$ *is restricted control set defined as* (13).

Theorem 3. *If PFPT method is feasible (Property 3) then solution of original HJB equation* (3) *is also a solution of the restricted HJB equation* (14).

Proof. If $\hat{\theta}(s,t) = \arg\max_{\theta\in\Theta}\left(\mathcal{L}_{s,t,\theta}V + d(s,t,\theta)\right)$ and $\hat{\theta}(t,s)\in\tilde{\Theta}(s,t)$, then also holds $\hat{\theta}(s,t) = \arg\max_{\theta\in\tilde{\Theta}(s,t)}\left(\mathcal{L}_{s,t,\theta}V + d(s,t,\theta)\right)$ because $\tilde{\Theta}(s,t)\subset\Theta$. The restricted HJB equation differs from the original HJB equation only by the set of controls, however, as the optimal controls for both are the same, solution of both equations is also the same. □

Following Theorem 3, if approximation computed with feasible PFPT method converges to solution of restricted (resp. related) HJB equation, then it also converges to solution of original HJB equation. In the next section we will examine convergence of PFPT method to the solution of restricted HJB equation.

6. Convergence of PFPT methods

In this section we will examine convergence approximation obtained by of PFPT method to the solution of restricted HJB equation. We will apply Theorem 1, therefore the scheme has to be monotone, stable and consistent. For classical and PCPT schemes, these properties are discussed for example in [4], for 1-dimensional case or in [9] for 2-dimensional case. Extension of monotonicity and stability proof to general case and PFPT schemes is straigtforward. For reader's convenience, we will show here these properties, using the same techniques as in [4]. The case of consistency is more tricky. In paper [4], a proof of consistency of PCPT method is stated to be non-trivial, and proof based on probabilistic methods from [6] is used. In our paper, we will refer to this approach employing the fact that any PFPT method can be seen as PCPT method for the related HJB equation (15).

6.1. Monotonicity

At first we will examine the monotonicity of the numerical scheme. We will present simple conditions that will ensure this property.

Property 4 (Positive coefficients condition). *The linear discrete approximation scheme $L_{i,j,\theta}(v_1^j, v_2^j, \ldots, v_N^j)$ defined in* (4) *satisfies the **positive coefficients condition** for the implicit scheme, if the following holds:*
$a_k(i,j,\theta_i^j) \geq 0$ *for* $k = 1, 2, \ldots, i-1, i+1, \ldots, N_h$.

Property 5 (Non-positive sum condition). *The linear discrete approximation scheme $L_{i,j,\theta}(v_1^j, v_2^j, \ldots, v_N^j)$ defined in* (4) *satisfies the **Non-positive sum condition** for the implicit scheme, if the following holds:*

$$\sum_{k=1}^{N_h} a_k(i,j,\theta_i^j) \leq 0.$$

Definition 6 (Z-matrix). *We say, that a matrix with non-positive off-diagonal elements is a Z-matrix.*

Definition 7 (M-matrix). *If A is a Z-matrix with all diagonal elements positive, and there exists a positive diagonal Matrix D, such that AD is strictly diagonally dominant, we say A is an M-matrix.*

Here we recall an important well-known property of M-matrices:

Remark 2. *Let A be a non-singular M-matrix. Then $A^{-1} \geq 0$ (All elements of A are non-negative).*

Lemma 1. *Let us suppose, that we have a PFPT scheme with set of control functions $\theta^z(s,t)$, satisfying the positive coefficients condition (Property 4) and non-positive sum condition (Property 5) for any $\theta^j = \bar{\theta}^{j,z}$, with Dirichlet BC's specified. Then, $I - \Delta_j t A^{j+1}(\bar{\theta}^{j,z})$ is an M-matrix for any control vector $\bar{\theta}^{j,z}$, $z \in \{1, 2, \ldots, Z\}$.*

Proof. Positive coefficients condition, together with the construction of Dirichlet boundary conditions, provides the Z-matrix property and a positive diagonal of the matrix $I - \Delta_j t A^{j+1}(\bar{\theta}^{j,z})$. Moreover, by employing the non-positive sum condition, matrix $I - \Delta_j t A^{j+1}(\bar{\theta}^{j,z})$ is also strictly diagonally dominant, and therefore satisfies the M-matrix property with $D = I$ (identity). $\square$

Theorem 4. *Let us suppose, that we have a PFPT scheme with set of control functions $\theta^z(s,t)$, satisfying the positive coefficients condition (Property 4) and the non-positive sum condition (Property 5) for any $\theta^j = \bar{\theta}^{j,z}$, with Dirichlet BC's specified. Then, this scheme is monotone.*

Proof. Let us recall, that the implicit PFPT scheme is defined as an equation of the form $Gv(x_i^{j+1}) = 0$ by (12). Now, as the positive coefficients condition is fulfilled, according to Lemma 1, $I - \Delta_j t A^{j+1}(\bar{\theta}^{j,z})$ is an M-matrix. Then, following Remark 2, $(I - \Delta_j t A^{j+1}(\bar{\theta}^{j,z}))^{-1}$ is non-negative, and as maximum is a non-decreasing function, whole left hand-side of equation (12) is non-increasing in v^j, which means, that the scheme is monotone. $\square$

6.2. Stability

Now, we will show that the conditions from the previous Section will ensure also the stability of the scheme.

Theorem 5. *Let us suppose, that we have a PFPT scheme with set of control functions $\theta^z(s,t)$, satisfying the positive coefficients condition (Property 4) and the non-positive sum condition (Property 5) for any $\theta^j = \bar{\theta}^{j,z}$, with Dirichlet BC's specified. Then, the scheme is stable.*

Proof. Let $\|v^{j+1}\|_\infty = v_p^{j+1}$. If is (s_p, t_{j+1}) is a boundary node, then from the construction of the scheme follows that

$$\|v^{j+1}\|_\infty = v_p^{j+1} = F_p^{j+1} \leq \max_i(|F_i^{j+1}|). \tag{16}$$

If $(p, j+1)$ is an inner domain node, $f_p^j = 0$ and the scheme (12) for $i = p$ can be written as

$$(I_p - \Delta_j t A_p^{j+1}(\bar{\theta}^{(j)}))v^{j+1} = v_p^j + \Delta_j t d_p^{j+1}(\bar{\theta}^{(j)}), \tag{17}$$

where $\bar{\theta}^{(j)}$ is the approximation of the optimal control vector, and I_p the p-th row of identity matrix. Now, from equation $\|v^{j+1}\| = v_p^{j+1}$, combined with the positive coefficients condition for $I - \Delta_j t A^{j+1}(\bar{\theta}^{(j)})$, follows, that

$$\begin{aligned} \sum_{k=1}^{N_h} (I_p - \Delta_j t A_p^{j+1}(\bar{\theta}^{(j)}))_k \|v^{j+1}\|_\infty &\leq v_p^j + \Delta_j t d_p^{j+1}(\bar{\theta}^{(j)}), \\ \left(1 - \Delta_j t \sum_{k=1}^{N_h} a_k(p, j, \bar{\theta}^{(j)})\right) \|v^{j+1}\|_\infty &\leq v_p^j + \Delta_j t d_p^{j+1}(\bar{\theta}^{(j)}). \end{aligned}$$

Now, using non-positive sum condition, we get

$$\begin{aligned} \|v^{j+1}\|_\infty &\leq v_p^j + \Delta_j t d_p^{j+1}(\bar{\theta}^{(j)}), \\ \|v^{j+1}\|_\infty &\leq \|v^j\|_\infty + \Delta_j t d_{max}, \end{aligned} \tag{18}$$

where $d_{max} = \max_{(s,t,\theta) \in S \times [0,T] \times \Theta} |d(s, t, \theta)|$. Now applying (18) recursively, we get

$$\|v^{j+1}\|_\infty \leq \|v^1\|_\infty + T d_{max}. \tag{19}$$

Combining (16) and (19), we end up with

$$\|v^{j+1}\|_\infty \leq \max\left(\max_i(|F_i^{j+1}|), \|v^1\|_\infty + T d_{max}\right).$$

□

Remark 3. *According to Theorem 2, the PFPT method is identical to the PCPT method for related HJB equation* (15)*. Therefore, if our PFPT method is monotone and stable, then also the PCPT method for related HJB equation is monotone and stable.*

6.3. Consistency and convergence

In [4], convergence of PCPT method is proved using results from [6]. We will reuse this approach and show convergence of PCPT method for related HJB equation (15). According to [4, 6], we do not need to show consistency of the whole PCPT scheme, which might be difficult, we just have to check if the following consistency requirement is fulfilled:

Definition 8 (Consistency requirement for PCPT method)**.** *We say that consistency requirement is fulfilled, if*

$$\lim_{h\to 0}\left|\left(\frac{\phi_i^{j+1}-\phi_i^j}{\Delta_j t}-(\tilde{L}_{i,j,z}(\phi_1^{j+1},\phi_2^{j+1},\dots,\phi_{N_h}^{j+1})+\tilde{d}(s_i,t_{j+1},z))\right)\right.$$
$$\left.-\left(\frac{\partial\phi_i^{j+1}}{\partial t}-(\tilde{\mathcal{L}}_{s_i,t_{j+1},z}\phi_i^{j+1}+\tilde{d}(s_i,t_{j+1},z))\right)\right|=0, \tag{20}$$

where $\phi_i^j=\phi(s_i,t_j)$, for any smooth test function $\phi(s,t):S\times[0,T]\to\mathbb{R}$ and any $z\in\{1,2,\dots,Z\}$.

Theorem 6. *If Property 1 holds, then consistency requirement from Definition 8 is met.*

Proof. According to definitions of $\tilde{\mathcal{L}}$ and $\tilde{L}$, it holds that $\tilde{\mathcal{L}}_{s,t,z}\phi_i^j=\mathcal{L}_{s,t,\bar{\theta}_i^{j,z}}\phi_i^j$ and $\tilde{L}_{i,j,z}(\phi_1^j,\phi_2^j,\dots,\phi_{N_h}^j)=L_{i,j,\bar{\theta}_i^{j,z}}(\phi_1^j,\phi_2^j,\dots,\phi_{N_h}^j)$. Property 1 states $|\mathcal{L}_{s_i,t_j,\theta}\phi_i^j-L_{i,j,\theta}(\phi_1^j,\phi_2^j,\dots,\phi_{N_h}^j)|=\mathcal{O}(h^k)$, $k>0$ for any θ, and therefore also specifically for $\theta=\bar{\theta}_i^{j,z}$. Moreover, as ϕ is smooth, it holds that $|(\phi_i^{j+1}-\phi_i^j)/\Delta_j t-\partial\phi_i^{j+1}/\partial t|=\mathcal{O}(\Delta_j t)=\mathcal{O}(h)$. Therefore (20) is smaller then $\mathcal{O}(h^k)+\mathcal{O}(h)$ and vanishes with h approaching 0. □

Theorem 7 (Convergence of PFPT method)**.** *If implicit PCPT scheme for the related HJB equation* (15) *satisfies strong uniqueness property (see [3]) and conditions of Theorems 4, 5, 6 then it is convergent.*

Proof. According to Theorem 4 Consistency requirement will be met an according to Remark 3 the scheme will be monotone and stable and therefore, as stated in [4], proof follows from results in [6]. □

We showed convergence of PCPT method for related HJB equation (15) under some conditions from previous theorems. As stated in Theorem 2, this method is identical to PFPT method for solving restricted HJB equation (14). According to Section 5.3 related HJB equation is only reformulated restricted HJB equation (14). Therefore, we can state, that PFPT method also converges to the solution of restricted HJB equation (under those same conditions). Moreover, according to Theorem 3, if the PFPT method is feasible (Property 3), it also converges to the solution of the original HJB equation.

7. Piecewise predicted policy timestepping method

In this section we will introduce a new member of the class of PFPT methods -so called piecewise predicted policy timestepping (PPPT) method. The idea of this method is quite simple: we solve at first the HJB equation with classical or PCPT method on a coarse grid. Then, we create the set of fixed control functions $\theta^z(s,t)$ by "intelligent" combining of the discrete approximations of optimal controls in different time layers computed on the coarse grid and adding "neighboring" fixed control functions. A detailed description of this algorithm clarifies this approach:

Construction of fixed policy functions for PPPT method:

1. Solve HJB PDE with classical or PCPT method on a coarse grid. By-product of this solution should be optimal control $\tilde{\theta}_i^j$ with $i \in \{1, 2, \ldots, \tilde{N}\}$, $j \in \{1, 2, \ldots, \tilde{M}\}$, where $\tilde{N}, \tilde{M}$ are dimensions of the coarse grid.
2. Define control indices $\tilde{z}_i^j$, such that $\tilde{\theta}_i^j = \theta_{\tilde{z}_i^j}$
3. Determine number of control functions $Z = \max\limits_{(i,j)\in\tilde{I}} \left|\tilde{z}_i^j - \tilde{z}_i^{j+1}\right|$ where $\tilde{I} = \{1, 2, \ldots, \tilde{N}\} \times \{1, 2, \ldots, \tilde{M} - 1\}$
4. Define $\tilde{M}$ 1-dimensional index functions in the layers of the coarse grid: for $j = 1, 2, \ldots, \tilde{M}$:
   - $\tilde{z}^j(s) = \tilde{z}_i^j$ where $i = \arg \min\limits_{k\in\{1,2,\ldots,\tilde{N}\}} \|s - s_k\|_\infty$
5. Define:
   $up(s,t) = \tilde{z}^j(s)$ where $j = \arg\min_{k\in\{1,2,\ldots,\tilde{M}\}, t\leq t_k} |t - t_k|$
   $down(s,t) = \tilde{z}^j(s)$ where $j = \arg\min_{k\in\{1,2,\ldots,\tilde{M}\}, t\geq t_k} |t - t_k|$
   Define $Z-2$ 2-dimensional index functions:
   for $z = 1, 2, \ldots, Z-2$:
   - $\tilde{z}^z(s,t) = round\left(\frac{z-1}{Z-3} up(s,t) + \frac{Z-2-z}{Z-3} down(s,t)\right)$

6. Determine neighbor index functions:

   $\tilde{z}^{Z-1}(s,t) = \min\left(\max_{z\in\{1,2,\ldots,Z-2\}} \tilde{z}^z(s,t), J\right)$

   $\tilde{z}^{Z}(s,t) = \max\left(\min_{z\in\{1,2,\ldots,Z-2\}} \tilde{z}^z(s,t), 1\right)$

7. Create control functions:

   for $z = 1, 2, \ldots, Z$:

   - $\theta^z(s,t) = \theta_{\tilde{z}^z(s,t)}$

   end.

After constructing these control functions, we can run PFPT method on finer grid, using in j-th time layer control vectors $\bar{\theta}^{j,z}$ with i-th element defined as $\bar{\theta}^{j,z}_i = \theta^z(s_i, t_j)$. Let us note that because of the specific construction of control functions, we may have in many time layers only a few different control vectors, that means, only a few different PDEs to solve, which makes the whole method significantly faster. That was also idea of this PPPT method: to create predictions of possible controls on coarse grid, and then use these predictions to reduce computational effort on fine grid. Therefore, this whole PPPT method can be seen as some kind predictor-corrector algorithm.

Question that remains is, if the method is feasible, that means, if the optimal control is in set $\tilde{\Theta}(s,t)$ that is implied by PPPT control functions. Then, the approximation computed with the PPPT method will converge not only to solution of restricted HJB equation, but also to solution of the original one. This question might be difficult to answer with certainty, despite the fact that the control functions are in PPPT method constructed in such way, that they should intuitively cover the optimal control in most nodes if the prediction isn't too bad. However, even if the optimal control is not covered by control functions in each node, the method is still convergent (converges to the solution of restricted HJB equation), and if control functions cover the optimal control in most nodes, this solution is probably still very good approximation of solution of the original HJB equation.

8. Numerical example

In this section, we will compare performance of PCPT and PPPT scheme on a numerical example. For comparison and verification reasons, we will take the whole example with boundary conditions, and up to small changes also with discretization, from [7]. For reader's convenience, we will repeat here the main characteristics of the problem.

8.1. Mean-variance optimal investment problem

We will start with a problem of dynamic investment allocation between stock and risk-free asset in an mean-variance framework. Let our stock follow a SDE of the form:

$$dS_t = (r + \xi\sigma)S_t dt + \sigma S_t dZ_t, \tag{21}$$

where S_t is stock price process, r is risk-free interest rate, σ is volatility, ξ is market price of risk and Z_t is a standard Brownian motion. Moreover, we will suppose that the investor contributes to portfolio at a constant rate π. Then, our task is to solve the following problem:

$$\max_{p\in P} \left(\mathbb{E}_{t=0}(W_T) - \lambda Var_{t=0}(W_T),\right) \tag{22}$$

$$dW_t = \left((r + p(W_t,t)\xi\sigma)W_t + \pi\right) dt + p(W_t,t)\sigma W_t dZ_t, \tag{23}$$

$$W_0 = K, \tag{24}$$

where λ is investors coefficient of risk aversion (or also Lagrange multiplier similar as in Markowitz model, see [10]), $p(W_t, t)$ is proportion of investors wealth invested in stock in time t for current wealth W_t, P is set of all admissible functions $p(W_t, t)$, K is some constant representing initial wealth, and T is final time.

For different λ, we expect different $\mathbb{E}_{t=0}W_T$. The set of all possible pairs $(\lambda, \mathbb{E}_{t=0}W_T)$ will be called efficient frontier. Note that often also set of pairs $(\mathbb{E}_{t=0}W_T, Var_{t=0}W_T)$ is referred as efficient frontier [10]. In [8], it is explained, how to compute pairs on this efficient frontier numerically. Main part of that problem is solving HJB equation in the following form:

$$\frac{\partial V}{\partial \tau} - \min_{p\in[p_{min},p_{max}]} \mathcal{L}_p V = 0, \tag{25}$$

$$\mathcal{L}_p V = \frac{1}{2}\sigma^2 p^2 W^2 \frac{\partial^2 V}{\partial W^2} + (\pi + (r + p\sigma\xi)W)\frac{\partial V}{\partial W}, \tag{26}$$

$$V(W,0) = \left(W - \frac{\gamma}{2}\right)^2, \tag{27}$$

where γ is a parameter set in advance, and dependent on the unknown pair $(\lambda, \mathbb{E}_{t=0}W_T)$. As solution, we will get beside the value function $V(W,t)$ also optimal control $p(W,t)$ which is optimal investment strategy for unknown value λ. This λ, also with $\mathbb{E}_{t=0}W_T$ can be computed afterwards, using the optimal investment strategy $p(W,t)$. For more details see [8]. Here, we will be concerned with solving HJB equation (25)-(27) numerically.

8.2. Discretization scheme

For our problem we will use an equidistant discretization of the domain $[0, W_{max}] \times [0,T]$, with M time-steps of size $\Delta\tau$ and N space-nodes, with distance between 2 neighboring nodes ΔW. Time derivative in (W_i, τ_j)-node $\partial v_i^j/\partial\tau$ will be discretized simply as $(v_i^j - v_i^{j-1})/\Delta\tau$ and elliptic operator $\mathcal{L}_p v_i^j$ will be discretized as

$$\begin{aligned} L_{W_i,\tau_j,p} v^j &= \frac{1}{2}\sigma^2 p^2 W_i^2 \frac{v_{i-1}^j - 2v_i^j + v_{i+1}^j}{(\Delta W)^2} \\ &+ (\pi + (r + p\sigma\xi)W_i)\, D_1(v,i,j,p), \end{aligned} \tag{28}$$

where

$$D_1(v,i,j,p) = \frac{v_{i+1}^j - v_{i-1}^j}{2\Delta W}, \tag{29}$$

if $|\,(\pi + (r + p\sigma\xi)W_i)\,| \le \sigma^2 p^2 W_i^2/(\Delta W)$, and

$$D_1(v,i,j,p) = \frac{v_{i+1}^j - v_i^j}{\Delta W} \qquad for \qquad \pi + (r + p\sigma\xi)W_i \ge 0,$$

$$D_1(v,i,j,p) = \frac{v_i^j - v_{i-1}^j}{\Delta W} \qquad for \qquad \pi + (r + p\sigma\xi)W_i < 0,$$

otherwise. As in [7], we use central differences to approximate also 1. order derivative in nodes where it does not harm monotonicity, what could lead to almost second order of convergence. This approach is well-described in [11]. As shown in [12], higher order of convergence is not feasible with monotone finite-difference schemes for HJB equations. Let us assume a positive money inflow rate π. Then, as the left boundary is in $W_0 = 0$, we need no boundary condition, because our discrete operator $L_{W_0,\tau_j,p}$ degenerates to

$$L_{W_0,\tau_j,p}v^j = \pi \frac{v_1^j - v_0^j}{\Delta W}, \tag{30}$$

so that we need no data from outside of the domain. In case of right boundary condition, we will use an approximation from [7] in form of Dirichlet boundary condition:

$$V(W_{max}, \tau) = \frac{1}{2}\alpha(\tau)W^2 + \beta(\tau)W + \delta(\tau), \tag{31}$$

where

$$\begin{aligned}
\alpha(\tau) &= \exp((a^2 + 2b)\tau), \\
\beta(\tau) &= -(\gamma + c)\exp(b\tau) + c\exp((a^2 + 2b)\tau), \\
\delta(\tau) &= -\frac{\pi(\gamma + c)}{b}(\exp(b\tau) - 1) + \frac{\pi c}{a^2 + b}(\exp((a^2 + 2b)\tau) - 1) + \frac{\gamma^2}{4}, \\
c &= 2\pi/(a^2 + b), \\
a &= \sigma p, \\
b &= r + p\sigma\xi.
\end{aligned}$$

Now, our goal is to use this discretization together with proposed BC's in an PFPT scheme. As the discretization satisfies the positive coefficients condition (Property 4) and the non-positive sum condition, it is monotone and stable. It is easy to show, that $L_{W_i,\tau_j,p}v^j$ is consistent approximation of $\mathcal{L}_p v(W_i, \tau_j)$ (Property 1), and therefore consistency requirement 8 is met. Therefore, under the strong uniqueness property (see [3]), our PPPT method will converge to solution of related (and restricted) HJB equation, due to Theorem 7. If we moreover assume that the PPPT method defined by control predictions computed on coarse grid is feasible, we can assume convergence to the viscosity solution of original HJB equation.

8.3. Numerical results

For solving Mean-variance optimal investment problem we implemented classical implicit method, PCPT method and PPPT method, all using discrete operator (28) and boundary conditions (30), (31). We implemented all methods in Matlab. For comparison reasons, we used the same parameter values as in [7]: $r = 0.03$, $\sigma = 0.15$, $\xi = 0.33$, $\pi = 0.1$, $\gamma = 14.47$. We implemented the scheme on the space domain $W \in [0, 5]$ and time domain $\tau \in [0, 20]$. We use time-step size $\Delta\tau = h_k$ and space-step size $\Delta W = 0.25h_k$. We used control set $P = [0, 1.5]$ equidistantly discretized on 31 different controls $0, 0.05, 0.1, \cdots, 1.5$.

In Table 1, we can see results of the numerical simulations. We tested PCPT method, classical implicit method and two PPPT methods with different approach to predictions. We runned the methods with $h_k = 2^{1-k}$, $k = 1, 2, \dots, 10$. To compute prediction of controls (control functions) for PPPT methods, we used PCPT scheme. In our first approach we have done predictions on grid with $h = 2^{-4}$. (that is even finer then some of the main algorithm grids) This PPPT method is dentoted as PPPT (1).

To verify our results, we checked the values of the solutions in $\tau = 20, W = 1$, which are also computed in [7]. This values are denoted as **Val**. We will estimate the error of the approximation **Err** with values from the final time layer A^k (computed with stepsize h_k). We also need exact solution, which is not known. Therefore, instead of the exact solution, we used an approximation of the final time layer A^{11} computed with the classical implicit method and $h = 2^{-10}$. The formula for estimating error is

$$Err \;\; A^k = \|A^k - A^{11}\|_2. \tag{32}$$

However, in [7] the error is estimated as difference of the values of the solutions in $\tau = 20, W = 1$ computed with stepsizes h_{k-1} and h_k. As this error is dependent only on the values Val, we will denote it as **Err(Val)**.

Experimental order of convergence **EOC** will be computed as

$$EOC \;\; A^k = \frac{\log(ErrA^{k-1}) - \log(ErrA^k)}{\log(h_{k-1}) - \log(h_k)}. \tag{33}$$

Table 1: Results of the methods

h_k	1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
Classical										
Err	3,03E+00	6,66E-01	1,51E-01	3,61E-02	8,58E-03	2,02E-03	4,48E-04	8,82E-05	1,26E-05	4,36E-07
EOC		2.185	2.146	2.058	2.075	2.085	2.175	2.344	2.809	4.852
Time	0.33	0.65	1.47	3.35	7.67	19.34	47.86	176.36	655.58	2570.57
Val	2.783	2.025	1.769	1.648	1.589	1.561	1.546	1.540	1.536	1.534
Err(Val)		0.7581	0.2558	0.1210	0.0586	0.0283	0.0144	0.0069	0.0034	0.0017
EOC(Val)			1.568	1.080	1.045	1.051	0.970	1.064	1.007	1.000
PCPT										
Err	3,37E+00	7,13E-01	1,58E-01	3,77E-02	8,91E-03	2,10E-03	4,64E-04	9,17E-05	1,32E-05	4,51E-07
EOC		2.239	2.171	2.071	2.081	2.087	2.175	2.340	2.798	4.871
Time	0.08	0.16	0.36	0.94	2.58	8.00	25.38	93.19	365.59	1475.30
Val	3.050	2.109	1.799	1.660	1.595	1.564	1.548	1.540	1.536	1.535
Err(Val)		0.9409	0.3092	0.1392	0.0654	0.0312	0.0158	0.0076	0.0038	0.0019
EOC(Val)			1.606	1.152	1.089	1.067	0.983	1.064	1.008	1.001
PPPT(1)										
Err	3,20E+00	6,85E-01	1,54E-01	3,64E-02	8,61E-03	2,09E-03	4,76E-04	9,88E-05	1,65E-05	1,38E-06
EOC		2.223	2.153	2.081	2.080	2.045	2.130	2.269	2.585	3.576
Time	2.71	2.72	2.74	2.81	3.05	3.78	6.61	17.55	61.57	238.09
Val	2.893	2.045	1.776	1.649	1.590	1.563	1.548	1.541	1.537	1.536
Err(Val)		0.8477	0.2691	0.1274	0.0591	0.0271	0.0144	0.0072	0.0036	0.0018
EOC(Val)			1.656	1.078	1.107	1.128	0.906	1.006	0.998	0.980
PPPT(2)										
Err	4,35E+00	9,92E-01	1,88E-01	4,20E-02	9,45E-03	2,16E-03	4,76E-04	9,28E-05	1,31E-05	4,43E-07
EOC		2.133	2.402	2.160	2.151	2.127	2.183	2.360	2.826	4.885
Time	0.06	0.10	0.20	0.40	0.88	2.35	6.53	21.48	77.60	310.84
Val	3.571	2.594	1.875	1.688	1.599	1.564	1.548	1.540	1.536	1.534
Err(Val)		0.9768	0.7193	0.1868	0.0883	0.0353	0.0159	0.0080	0.0039	0.0019
EOC(Val)			0.442	1.945	1.081	1.321	1.150	0.993	1.029	1.073

We will compute experimental order of convergence using the above formula also with error estimation Err(Val) and denote this experimental order of convergence as **EOC(Val)** Computational time of each method (dependent on computer), is in table denoted as **Time**. The time is in seconds and is just informative, as the value varies with each new run. In case of PPPT method, time needed to compute prediction of controls is added.

In Figure 1, we plotted the logarithm of computational time against the logarithm of the error, to see how much time we need for each method to get the same level of accuracy. We observe, that PPPT method is slower on low level of accuracy at first, what is caused by relatively high time-costs spent

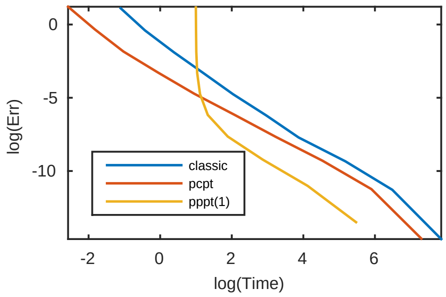

Figure 1: Comparison of classic, PCPT and PPPT (1) method with prediction grid step-size $h = 2^{-4}$

on computing prediction of control in contrast to fast low-accuracy PCPT and classical implicit method. For medium levels of accuracy that are more time-demanding for classical and PCPT method, the prediction already spares time and PPPT method is most effective. However, high levels of accuracy do not seem possible to be obtained with this particular PPPT, probably because of poor prediction (Feasibility condition not met in some nodes, causing error that can't be reduced).

The last analysis of results leads us an idea of running different PPPT method on each refinement level, so that the prediction grid will be adjusted to the desired level of accuracy. Therefore, now we will compute on each refinement level $k = 1, 2, \ldots, 10$ new prediction of controls with PCPT scheme with $h = 4h_k$. Results of these approach to PPPT scheme is in Table 1 denoted as PPPT (2) Figure 2 illustrates the dependence of computational time an accuracy. We see, that this PPPT method the is most efficient one all the time. For higher levels of accuracy it is 4.7 times faster then PCPT and 8 times faster classical implicit method, what is a significant speed-up.

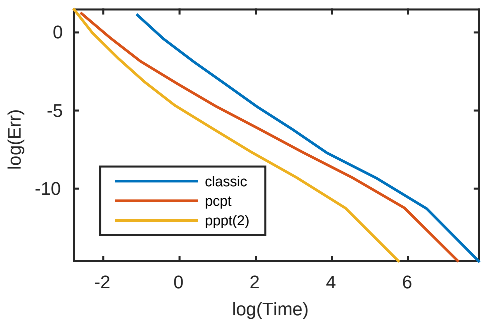

Figure 2: Comparison of classic, PCPT and PPPT (2) method with prediction grid step-size $h = 4h_k$

Table 1 shows the experimental second order of convergence EOC by using error estimation (32). As we used central differences as much as possible, order of approximation in space should be eventually close to 2. However, order of approximation in time is only 1, therefore the experimental order of convergence may imply that the space error is significantly dominating the time error. Higher experimental rates of convergence obtained for finer grid are biased because of using solution on fine grid instead of true analytical solution.

On the other hand, we achieved only experimental first order of convergence EOC(Val) computed by using approach from [7]. However, this estimation is done only using value in one specific node, therefore this may indicate slower convergence around this node. This experimental order as well as values in $\tau = 20, W = 1$ (Val) are very similar to results presented in [7], what verifies our implementation.

8.4. Parallelization

One of the biggest advantages of PCPT and also of PFPT methods in contrast to classical implicit methods, is that process solving n PDEs in each time layer can be, in contrast to policy iteration, parallelized. To test this possibility with our implementation of PCPT and PPPT method, we used Matlab parforroutine, to compute PDEs with fixed control in each time layer simultaneously on 4 cores. Moreover, as very time-consuming part of classical implicit method, we identified testing all controls to find the optimal policy iteration algorithm. This part is also suitable for parallelization, therefore we implemented simultaneous testing of the controls. Theoretically, these improvements should lead to great speed up, especially in case of PCPT and PPPT method, where parallelization can utilized most of the time. However, some speed-up is present only for very large number of space-nodes.

Table 2: Parallelized methods, computational time (2 time steps)

# Nodes:	5×10^1	5×10^2	5×10^3	5×10^4	5×10^5
Classic implicit					
-time	0.321	0.337	0.631	3.886	49.488
-% speed-up	-401	-171	13	48	46
PCPT					
-time	0.054	0.060	0.112	0.758	9.320
-% speed-up	-521	-175	18	49	47
PPPT					
-time	0,030	0.031	0.043	0.144	1.279
-% speed-up	-941	-647	-142	10	32

Table 2 summarizes the speedup of the method after using parallelization (in percents). Because of time reasons we computed with each method only 2 timesteps. By each method, first row states how much time did those 2 steps took, for parallelized method, and second row tells, how much faster (or slower) the method was than its non-parallelized version. We see that for smaller number of nodes speed up is actually slow down. Parallelization may be more effective using another more advanced techniques or programming languages. We added this part just to illustrate the possibilities of PFPT methods, as a proof of concept.

9. Conclusion

In this paper, we introduced a class of piecewise fixed policy timestepping schemes. The famous piecewise constant policy timestepping (PCPT) scheme is also member of this class of schemes. Moreover, we also showed that any PFPT scheme can be seen as PCPT scheme for an related (and restricted) HJB equation. This helped us by establishing convergence result, since the convergence of PCPT schemes is already shown in [4] using results from [6]. We reused these approaches to establish the convergence of PFPT scheme to the solution of the restricted HJB equation and defined a feasibility condition that should be fulfilled, so that the approximation obtained by using PFPT scheme converges also to the solution of the original HJB equation.

The main result of our paper is the introduction of a new member of the class of PFPT schemes: piecewise predicted policy timestepping (PPPT) scheme. In this method, we often need to solve only a few PDEs in many time layers in contrast to PCPT method, where the number of PDEs to be solved in each time layer corresponds to number of possible controls. Therefore, this method has good chances to be several times faster, depending on the problem. This reduction of computational effort was achieved by creating some prediction of control on an coarse grid. Feasibility condition may be difficult to check in this case, however, there are good indications that the method will end up with good approximation, even if this condition is not met.

To test out PPPT method and compare it with PCPT method and classical implicit method, we implemented all methods for HJB equation arising from a mean-variance optimal investment problem from [8] and [7]. We used the same parameters as in [7] and verified our implementation by comparing our results with the results of [7]. We estimated the error of the approximation and experimental order of convergence. For all three methods, we got experimental order of convergence between first and second order, which is also expected, as we used first order approximation of the first derivative and second order approximation of the second derivative. The PPPT method was fastest for some accuracy levels, but failed to deliver higher accuracy without better prediction. However, if we made the resolution of the prediction grid dependent on resolution of the main algorithm grid, PPPT method was clearly fastest. For higher levels of accuracy it was about 4.7 times faster then PCPT and 8 times faster then classical implicit method.

We should note that this speed-up is dependent on many characteristics of the particular problem. Higher number of control variables makes PCPT very time consuming and PPPT is significantly better. Also, if the optimal control doesn't change much in time, we will get only small number of control vectors in PPPT method, and less PDEs has to be solved in each time layer. Another question is how good is the control policy prediction, and how the control functions should be created from this prediction. Again, if the optimal control policy does not change much in time and space, then also predictions made on coarser grids may be suitable, as there are few important changes in control to be captured.

References

[1] M. G. Crandall, H. Ishii, P.-L. Lions, Users guide to viscosity solutions of second order partial differential equations, Bulletin of the American Mathematical Society 27 (1) (1992) 1–67.

[2] Q. Liu, X. Zhou, An introduction to viscosity solution theory, Introductory notes (2013).

[3] G. Barles, P. E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, in: Asymptotic Analysis, no. 4, 1991, pp. 2347–2349.

[4] P. A. Forsyth, G. Labahn, Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance, Journal of Computational Finance 11 (2) (2007) 1.

[5] P. A. Forsyth, K. R. Vetzal, Numerical methods for nonlinear PDEs in finance, in: Handbook of Computational Finance, Springer, 2012, pp. 503–528.

[6] N. Krylov, Approximating value functions for controlled degenerate diffusion processes by using piece-wise constant policies, Electronic Journal of Probability 4 (2) (1999) 1–19.

[7] C. Reisinger, P. Forsyth, Piecewise constant policy approximations to Hamilton-Jacobi-Bellman equations, Applied Numerical Mathematics 103 (2016) 27–47.

[8] J. Wang, P. Forsyth, Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous time mean variance asset allocation, Journal of Economic Dynamics and Control 34 (2) (2010) 207–230.

[9] K. Ma, P. Forsyth, An unconditionally monotone numerical scheme for the two factor uncertain volatility model, Submitted to IMA Journal of Numerical Analysis.

[10] R. C. Merton, et al., An analytic derivation of the efficient portfolio frontier, Journal of Financial and Quantitative Analysis 7 (4) (1972) 1851–1872.

[11] J. Wang, P. A. Forsyth, Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance, SIAM Journal on Numerical Analysis 46 (3) (2008) 1580–1601.

[12] I. Kossaczký, M. Ehrhardt, M. Günther, On the non-existence of higher order monotone approximation schemes for HJB equations, Applied Mathematics Letters 52 (2016) 53–57.