Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

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

Preprint BUW-IMACM 17/06

Igor Kossaczký, Matthias Ehrhardt and Michael Günther

# A new convergent explicit Tree-Grid Method for HJB equations in one space dimension

September 28, 2017

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

# A new convergent explicit Tree-Grid Method for HJB equations in one space dimension

Igor Kossaczký[*], Matthias Ehrhardt and Michael Günther

*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 introduce a new unconditionally convergent explicit Tree-Grid Method for solving stochastic control problems with one space and one time dimension or equivalently, the corresponding Hamilton-Jacobi-Bellman equation. We prove the convergence of the method and outline the relationships to other numerical methods. The case of vanishing diffusion is treated by introducing an artificial diffusion term. We illustrate the superiority of our method to the standardly used implicit finite difference method on two numerical examples from finance.

**AMS subject classifications**: 65M75, 65C40

**Key words**: Tree-Grid Method, Hamilton-Jacobi-Bellman equation, Stochastic control problem.

## 1. Introduction

In this work we are interested in solving stochastic control problems (SCP) numerically. Such problems can be represented by so called Hamilton-Jacobi-Bellman (HJB) equations, and arise in many applications in physics, economics, or finance. This article is divided into five main sections. In this first introductory section, we define the stochastic control problem and HJB equation and discuss the most widely used numerical methods. In the Section 2, we derive the new Tree-Grid Method –the main result of this paper. The convergence of this method is proven in Section 3. In Section 4, we test the performance of the method on two problems from finance. We compare the results with the ones from the standardly used implicit finite-difference method. Finally, Section 5 presents the conclusion.

---

[*]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)

## 1.1. Problem formulation

We are concerned with searching for the value function $V(s,t)$ of the following *general stochastic control problem* (SCP):

$$V(s,t) = \max_{\theta(s,t) \in \bar{\Theta}} \mathbb{E}\left( \int_t^T \exp\left( \int_t^k r(S_l, l, \theta(S_l, l))dl \right) f(S_k, k, \theta(S_k, k))dk \right.$$
$$\left. + \exp\left( \int_t^T r(S_k, k, \theta(S_k, k))dk \right) V_T(S_T) \Big| S_t = s \right), \tag{1.1}$$

$$dS_t = \mu(S_t, t, \theta(S_t, t))dt + \sigma(S_t, t, \theta(S_t, t))dW_t, \tag{1.2}$$

$$0 < t < T, \quad s \in \mathbb{R}$$

where $s$ is state variable and $t$ is time. Here, $\bar{\Theta}$ is space of all *suitable* control functions from $\mathbb{R} \times [0, T]$ to a set $\Theta$. For our purpose, we will suppose $\Theta$ to be discrete. If this is not the case, we can easily achieve this property by its discretization. We also suppose that the functions $r, f, \mu, \sigma, V_T$ are chosen *suitably*. For example, we demand Lipschitz continuity of $\mu, \sigma$:

$$\exists K > 0 : \forall t \in [0, T], \theta \in \Theta, s_1, s_2 \in \mathbb{R} :$$
$$|\mu(s_1, t, \theta) - \mu(s_2, t, \theta)| + |\sigma(s_1, t, \theta) - \sigma(s_2, t, \theta)| \le K|s_1 - s_2|$$

For a detailed analysis of suitability of coefficient and control functions we reffer to some classic stochastic control literature e.g. [14, 20]. Now following Bellman's principle, the *dynamic programming equation* holds:

$$V(s, t_j) = \max_{\theta(s,t) \in \bar{\Theta}_{t_j}} \mathbb{E}\left( \int_{t_j}^{t_{j+1}} \exp\left( \int_{t_j}^k r(S_l, l, \theta(S_l, l))dl \right) f(S_k, k, \theta(S_k, k))dk \right.$$
$$\left. + \exp\left( \int_{t_j}^{t_{j+1}} r(S_k, k, \theta(S_k, k))dk \right) V(S_{t_{j+1}}, t_{j+1}) \Big| S_{t_j} = s \right), \tag{1.3}$$

where $0 \le t_j < t_{j+1} \le T$ are some time-points and $\bar{\Theta}_{t_j}$ is a set of control functions from $\bar{\Theta}$ restricted to the $\mathbb{R} \times [t_j, t_{j+1})$ domain. Using this equation (1.3), it can be shown [20, 21], that solving the SCP (1.1),(1.2) is equivalent to solving the so-called Hamilton-Jacobi-Bellman (HJB) equation:

$$\frac{\partial V}{\partial t} + \max_{\theta \in \Theta}\left( \frac{\sigma(\cdot)^2}{2} \frac{\partial^2 V}{\partial s^2} + \mu(\cdot)\frac{\partial V}{\partial s} + r(\cdot)V + f(\cdot) \right) = 0, \tag{1.4}$$

$$V(s, T) = V_T(s), \tag{1.5}$$

$$0 < t < T, \quad s \in \mathbb{R}$$

where $\sigma(\cdot)$, $\mu(\cdot)$, $r(\cdot)$, $f(\cdot)$ are functions of $s, t, \theta$. We should note that the maximum operator in (1.1) and (1.4) can be replaced by minimum, (supremum, infimum) operator

and the whole following analysis will hold analogously. Another possible generalization is the use of both infimum and supremum in so called stochastic differential games and corresponding Hamilton-Jacobi-Bellman-Isaac equation [16]. Use of even more general operators is analyzed for example in [17]. However, in this work we will restrict ourselves to the case of minimum and maximum operators. Let us note that this covers also the case of omitting the operator completely, as this can be seen as maximum through an one-element set. In that case, a relationship between (1.1),(1.2) and (1.4),(1.5) is established by the classical Feynman-Kac formula. In financial mathematics, this relationship is represented by a connection between the option pricing problem and the Black-Scholes equation [5].

### 1.2. Overview of the numerical methods

Numerical methods used to solve the above SCP can be divided into two classes, based on the formulation they are exploiting. The main idea of methods based on the partial differential equation (PDE) approach is to solve the PDE (1.4),(1.5) with numerical methods as for example finite differences. Here the implicit finite difference methods (FDM) using policy iteration were shown to be successful e.g. in the work of Forsyth and Vetzal [9]. An alternative approach to policy iteration used in this method is the piecewise constant policy timestepping (PCPT) scheme used for example in [8]. A modification of this scheme leading to experimentally faster algorithms was proposed by the authors in [12]. For studying the convergence of solutions of these methods to viscosity solutions the classical theory of Barles and Souganidis [3] is used. An alternative approach based on the transformation of the PDE was proposed by Kilianová and Ševčovič in [11].

Whereas methods based on the PDE approach are typically implicit, methods based on the original problem formulation (1.1),(1.2) are mostly explicit. Apparently the most famous of these approaches are the methods based on Markov chain approximations of the stochastic differential equation (SDE) (1.2) presented in the book of Kushner and Dupuis [14]. We should note, that in this setting probabilistic techniques are used to prove convergence. In finance, binomial and trinomial tree methods [1,2] are widely used. These methods often present another viewpoint on the Markov chain approximation, and are equivalent up to some order to explicit FDMs (see for example [1]). These methods are known to suffer from instability if a certain condition on stepsizes is not met. This is also the reason why implicit methods are used more often. In tree and Markov chain methods, fulfilling this condition is achieved by a problem-specific construction of the grid or lattice. And finally, also based on the original problem formulation, there are forward shooting grid (FSG) methods [10], combining the tree lattice of binomial or trinomial method and the grid of the FDM or Markov chain approximation method and that are used typically in path-dependent option pricing [4,10], but can be easily implemented also for SCPs. These methods however may suffer from convergence problems as outlined by Forsyth, Vetzal and Zvan in [7]. In this article we will also briefly sketch the main reasons of these convergence issues of FSG schemes.

Studying these methods, it's easy to realize that they often present different viewpoints on similar approaches. Also, the method presented in this paper exhibits certain similarities

with these established methods, which we will outline later.

## 2. The Tree-Grid Method

In this section, we will derive the new Tree-Grid Method.

### 2.1. The basic idea

In our proposed method we compute the approximation of the solution on a rectangular domain $[s_L, s_R] \times [0, T]$ with some grid as in PDE-based schemes. The gridpoints are denoted as $[s_i, t_j]$, $i \in \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots, M\}$, $k < l \Rightarrow S_k < s_l, t_k < t_l$, $t_1 = 0, t_M = T$, $s_1 = s_L, s_N = s_R$. For step-sizes we use following notation: $\Delta_i s = s_{i+1} - s_i$, $\Delta_j t = t_{j+1} - t_j$. We should point out that the grid defined in such manner is very general, in contrast to grids or lattices used for Markov chain approximations or tree methods. Later, we will show that in our new explicit method no additional restrictions on the grid are required, in contrast to standard explicit methods. This gives us a lot of freedom to choose the discretization not only according to problem coefficients, but also according to the terminal condition (1.5), being an important advantage of implicit methods.

The numerical approximation of $V(s_i, t_j)$ will be denoted as $v(s_i, t_j)$ or simply as $v_i^j$. We define a terminal condition $v_i^M = V_T(s_i)$ and some suitable boundary conditions. In this paper we suppose that the solution on the intervals $[-\infty, s_1] \times [0, T]$ resp. $[s_N, \infty] \times [0, T]$ can by approximated with known functions $BC_L(s, t)$ resp. $BC_R(s, t)$. This also covers the case of Dirichlet boundary conditions, where $BC_L$ and $BC_R$ are constant in $s$. In case of Neumann boundary conditions, $BC_L$ and $BC_R$ can be set to linear functions with a prescribed slope, fulfilling $BC_L(s_2, t_j) = v(s_2, t_j)$, $BC_R(s_{N-1}, t_j) = v(s_{N-1}, t_j)$. Also generalization of the method to other cases of boundary conditions should be straightforward.

The main idea of this scheme follows the same principle as most numerical methods for this kind of problems: we will start in the last time layer $t_M = T$ and then subsequently compute values in the previous time layers $t_j$. To intuitively derive the method we will use the original problem formulation (1.1),(1.2) and the dynamic programming equation (1.3). To prove the convergence we will however regard the scheme as an approximation of the PDE-problem (1.4),(1.5).

Let us assume, we are at time $t_j$, $S_{t_j} = s_i$ and we want to compute an approximation of the current value of the value function $v_i^j$. Also assume that we already somehow know the values $v_l^{j+1}$, $\forall l = 1, 2, \ldots, N$ from the previous time layer $t_{j+1}$. Now we can compute the approximate probability distribution of $S_{t_{j+1}}$ using the SDE (1.2) for the stochastic process:

$$\begin{aligned}
\bar{S}_{t_{j+1}} &= S_{t_j} + \mu(S_{t_j}, t_j, \theta(S_{t_j}, t_j))\Delta_j t + \sigma(S_{t_j}, t_j, \theta(S_{t_j}, t_j))\Delta_j W \\
&= s_i + \mu(s_i, t_j, \theta)\Delta_j t + \sigma(s_i, t_j, \theta)\Delta_j W.
\end{aligned} \tag{2.1}$$

Here, and also later if misunderstanding is not possible, we abbreviate $\theta(s_i, t_j)$ simply as $\theta$ and $\Delta_j W$ is normally distributed random variable (RV) with mean 0 and variance $\Delta_j t$.

Using this approximation, we want to compute $v_i^j$, the approximation of $V(s_i, t_j)$, by using again some discrete approximation of the dynamic programming equation (1.3). However, as we only know approximations of $V(s, t_{j+1})$ in discrete points $s_i$, a continuous RV $\bar{S}_{t_{j+1}}$ is not suitable and should be replaced by a discrete one. This is the main idea of our method as well as of FSG methods, Tree methods, and Markov chain approximation methods.

**Problem:** Discrete random variable (RV) with values from $\{s_1, s_2, \ldots, s_N\}$ suitably approximating normally distributed $\bar{S}_{t_{j+1}}$ from (2.1) should be found.

## 2.2. Excursion: FSG method

The FSG approach [4] to this problem is to approximate $\bar{S}_{t_{j+1}}$ with a RV $\tilde{S}_{t_{j+1}}$ that attains a finite number (typically 2, $\tilde{s}_+$ and $\tilde{s}_-$) of values with in-forward given probabilities ($p_+$ and $p_-$). These probabilities with corresponding values arise typically from a binomial tree model, therefore the first two moments of $\bar{S}_{t_{j+1}}$ and of $\tilde{S}_{t_{j+1}}$ are matching, what is a desirable property, as $\bar{S}_{t_{j+1}}$ is normally distributed and therefore fully characterized by its first two moments. However as the values $\tilde{s}_+$ and $\tilde{s}_-$ of $\tilde{S}_{t_{j+1}}$ typically do not coincide with the gridpoints from the set $\{s_1, s_2, \ldots, s_N\}$, approximations $v_+^{j+1}, v_-^{j+1}$, of $V(\tilde{s}_+, t_{j+1}), V(\tilde{s}_-, t_{j+1})$ are not known. Therefore, these values are computed by some interpolation formula from known values $v_1^{j+1}, v_2^{j+1}, \ldots, v_N^{j+1}$: $v_\pm^{j+1} = \sum_{k_\pm=1}^{K_\pm} \alpha_{k_\pm}^\pm v_{i_{k_\pm}}^{j+1}$. After that a discrete version of dynamic programming equation (1.3) will be used, and at this step we will be interested only in an approximation of expected value $\mathbb{E}(V(S_{t_{j+1}} t_{j+1}))$, that will be computed as

$$\mathbb{E}(V(S_{t_{j+1}}, t_{j+1})) \approx \mathbb{E}(V(\bar{S}_{t_{j+1}}, t_{j+1})) \approx \mathbb{E}(V(\tilde{S}_{t_{j+1}}, t_{j+1})) \tag{2.2}$$

$$\approx p_+ v_+^{j+1} + p_- v_-^{j+1} = \sum_{k_+=1}^{K_+} p_+ \alpha_{k_+}^+ v_{i_{k_+}}^{j+1} + \sum_{k_-=1}^{K_-} p_- \alpha_{k_-}^- v_{i_{k_-}}^{j+1}, \tag{2.3}$$

where the first approximation in (2.2) is with respect to time, second is with respect to space and the last approximation, leading to formula (2.3), is an interpolation with the known values in the grid points. However, the final approximation (2.3) can be again interpreted as an expected value $\mathbb{E}(V(\tilde{S}'_{t_{j+1}}, t_{j+1}))$ where $\tilde{S}'_{t_{j+1}}$ is a discrete RV taking values $s_{i_{k_\pm}}$ with "probabilities" $p_\pm \alpha_{k_\pm}^\pm$. However in contrast to $\tilde{S}_{t_{j+1}}$, the moments of $\tilde{S}'_{t_{j+1}}$ will most probably not match with the moments of $\bar{S}_{t_{j+1}}$. This can be interpreted in such manner, that using this approach we solve a SCP driven by an SDE different from (1.2). Moreover, depending on the interpolation formula the "probabilities" may not sum up to 1 and even not be non-negative anymore (not the case of constant or linear interpolation), what may lead to instability of the whole scheme. This possible defect of the method is analogous to the instability of explicit FDM schemes if the timestep-spacestep condition is not met: both defects harm the monotonicity of the schemes. This makes such methods unsuitable for searching viscosity solution and possibly even unstable.

We should note, that the above analysis was done for a FSG method based on $\tilde{S}_{t_{j+1}}$ attaining two values, but the case of more values is completely analogical. Of course interpolation used in this method may be in many cases "good enough", meaning that the moment matching is done in the limit case, and the whole method may converge to the solution. However this is not automatic, and it is problem-specific. An analysis of when FSG is successful and when not for the problem of path-dependent option pricing can be found in [7].

## 2.3. The basic Tree-Grid Method

In our new scheme we will also approximate $\bar{S}_{t_{j+1}}$ from (2.1) with a discrete RV $\tilde{S}_{t_{j+1}}$. However, because of the problems with standard FSG schemes, we choose a different approach to construct this RV. In order to avoid interpolation $\tilde{S}_{t_{j+1}}$ will attain only values from the set $\{s_1, s_2, \ldots, s_N\}$. Exceptions arising close to a boundary will be discussed later. In this paper, we will derive a scheme where $\tilde{S}_{t_{j+1}}$ attains three possible values $s_-, s_o, s_+ \in \{s_1, s_2, \ldots, s_N\}$, $s_- < s_o < s_+$, with corresponding probabilities $p_-, p_o, p_+$. Of course, because of (2.1), these values will depend on the current state $s_i$, time $t_j$ and control $\theta$ and should be denoted as $s_-(s_i, t_j, \theta), s_o(s_i, t_j, \theta), s_+(s_i, t_j, \theta), p_-(s_i, t_j, \theta), p_o(s_i, t_j, \theta), p_+(s_i, t_j, \theta)$, however for simplicity we prefer the shorter form. We will try to choose the values in such manner, that the following conditions are satisfied:

$$p_-, p_o, p_+ \geq 0, \tag{2.4}$$

$$p_- + p_o + p_+ = 1, \tag{2.5}$$

$$p_- s_- + p_o s_o + p_+ s_+ = E, \tag{2.6}$$

$$p_- s_-^2 + p_o s_o^2 + p_+ s_+^2 = Var + E^2, \tag{2.7}$$

where

$$E := \mathbb{E}(\bar{S}_{t_{j+1}}) = s_i + \mu(s_i, t_j, \theta)\Delta_j t, \tag{2.8}$$

$$Var := Var(\bar{S}_{t_{j+1}}) = \sigma(s_i, t_j, \theta)^2 \Delta_j t. \tag{2.9}$$

The first two conditions (2.4),(2.5) state that $p_-, p_o, p_+$ can be interpreted as probabilities and, as we will see later, they also ensure stability and monotonicity of the scheme. The following two conditions (2.6),(2.7) ensure that the first two moments of the RVs $\tilde{S}_{t_{j+1}}$ and $\bar{S}_{t_{j+1}}$ are matching, and as shown later, together with (2.5) also ensure the consistency of the scheme with the PDE (1.4). Solving equations (2.5)-(2.7) we get

$$p_- = \frac{(s_o - E)(s_+ - E) + Var}{(s_- - s_o)(s_- - s_+)} \tag{2.10}$$

$$p_o = \frac{(s_- - E)(s_+ - E) + Var}{(s_o - s_-)(s_o - s_+)} \tag{2.11}$$

$$p_+ = \frac{(s_- - E)(s_o - E) + Var}{(s_+ - s_-)(s_+ - s_o)} \tag{2.12}$$

The question remains, if we can choose $s_-, s_o, s_+$ in such manner, that the non-negativity (2.4) is also fulfilled.

Let us suppose without loss of generality that $\mu(s_i, t_j, \theta) \geq 0 \Rightarrow E \geq s_i$. If $E < s_-$ or $E > s_+$ then $p_o < 0$. If $Var > 0$, also $E = s_-$ or $E = s_+$ lead to $p_o < 0$. Therefore, we will choose $s_-, s_+$ so that $s_- < E < s_+$. As we will see later, for unconditional consistency of the scheme, it is necessary that one of $s_-$, $s_o$, or $s_+$ equals to $s_i$. As $s_i \leq E < s_+$ we can't choose $s_+ = s_i$. Analogously in case of negative drift $\mu(s_i, t_j, \theta) \leq 0$, we would not be able to choose $s_- = s_i$. To make a suitable choice in both cases we will choose $s_o = s_i$. Now, (case of a positive drift), $p_+ \geq 0$ automatically (in case of a negative drift it would be $p_- \geq 0$). The denominator of $p_-$ is positive and hence we want a positive numerator. The denominator of $p_o$ is negative therefore we want a negative numerator.

Since $s_-$ only appears in the numerator of $p_o$ and not in the numerator of $p_-$, for any choice of $s_+$ we can choose $s_-$ small enough such that $p_o \geq 0$ for any choice of $s_+$. Choices of $s_-$ behind the boundary ($s_- < s_1$) are also possible as a special case and will be explained later. Therefore the only question remains, if we can choose such $s_+$, that the numerator of $p_-$ will be positive. We will use abbreviations $\Delta_{i+}s = s_+ - s_i = s_+ - s_o$ and $\Delta_{i-}s = s_i - s_- = s_o - s_-$. Moreover, we will use abbreviations $\mu := \mu(s_i, t_j, \theta)$ and $\sigma = \sigma(s_i, t_j, \theta)$ if confusion is not possible. For the condition on numerator of $p_-$ from equation (2.10) holds

$$(s_o - E)(s_+ - E) + Var \geq 0 \Leftrightarrow -\mu\Delta_j t(\Delta_{i+}s - \mu\Delta_j t) + \sigma^2 \Delta_j t \geq 0$$
$$\Leftrightarrow \Delta_{i+}s \leq \mu\Delta_j t + \sigma^2/\mu. \tag{2.13}$$

It should hold also $s_+ > E \Rightarrow \Delta_{i+}s > \mu\Delta_j t$. Combining this with (2.13), we get the following condition: $\mu\Delta_j t < \Delta_{i+}s \leq \mu\Delta_j t + \sigma^2/\mu$. A sufficient condition, under which $s_+$ leading to fulfillment of this inequality can be found, is

$$\Delta s \leq \sigma(s_i, t_j, \theta)^2/\mu(s_i, t_j, \theta), \tag{2.14}$$

where $\Delta s = \max_{i \in \{1,2,\ldots,N-1\}} \Delta_i s$. This seems a good result, however as we will see later, the convergence of this method depends on the distances $\Delta_{i+}s$, $\Delta_{i-}s$. Unfortunately, we do not have any bound on $\Delta_{i-}s$ now, we just know that it can be chosen to ensure that $p_o$ will be positive. Therefore we will try another approach: we will try to minimize the distances $\Delta_{i+}s$, $\Delta_{i-}s$ while keeping $p_o$ positive. Then we will check, if also $p_-$ is positive.

**Problem:**

$$\min_{s_+, s_- \in \{s_1, s_2, \ldots, s_n\}} \min(s_+ - s_i, s_i - s_-) \tag{2.15}$$
$$|(s_- - E)(s_+ - E)| \geq Var \tag{2.16}$$

Solving this problem may not be trivial in general, however an exact solution is also not needed. If we suppose $E$ to be "close enough" to $s_i$ then some "close to optimal" (and possibly also optimal) solution will be:

$$s_- = \left\lfloor E - \sqrt{Var} \right\rfloor_s = \left\lfloor s_i + \mu\Delta_j t - \sqrt{Var} \right\rfloor_s \tag{2.17}$$
$$s_+ = \left\lceil E + \sqrt{Var} \right\rceil_s = \left\lceil s_i + \mu\Delta_j t + \sqrt{Var} \right\rceil_s \tag{2.18}$$

where $\lceil \rceil_s$ denotes rounding to nearest greater element from $s_1, s_2, \ldots, s_N$, and $\lfloor \rfloor_s$ denotes rounding to nearest smaller element from $s_1, s_2, \ldots, s_N$. If such element does not exist, $\lceil x \rceil_s$ and $\lfloor x \rfloor_s$ will return just $x$. This corresponds to the boundary cases where $x < s_1$ or $x > s_N$ and will be discussed later. As we will see later, on an equidistant or "locally equidistant" grid it may be advantageous to choose $s_+$ and $s_-$ symmetric around $s_i$. Therefore we propose here also another choices of $s_+$, $s_-$, that will ensure this symmetry, fulfill condition (2.16), however possibly lead to a greater value of the minimized expression (2.15):

$$s_- = \left\lfloor s_i - \sqrt{(\mu \Delta_j t)^2 + Var} \right\rfloor_s, \tag{2.19}$$

$$s_+ = \left\lceil s_i + \sqrt{(\mu \Delta_j t)^2 + Var} \right\rceil_s. \tag{2.20}$$

Now, for (2.17),(2.18) hold the estimates

$$s_- \geq s_i + \mu \Delta_j t - \sqrt{Var} - \Delta s, \tag{2.21}$$

$$s_+ \leq s_i + \mu \Delta_j t + \sqrt{Var} + \Delta s, \tag{2.22}$$

and for (2.19),(2.20) hold estimates

$$s_- \geq s_i - \sqrt{(\mu \Delta_j t)^2 + Var} - \Delta s \geq s_i - |\mu| \Delta_j t - \sqrt{Var} - \Delta s, \tag{2.23}$$

$$s_+ \leq s_i + \sqrt{(\mu \Delta_j t)^2 + Var} + \Delta s \leq s_i + |\mu| \Delta_j t + \sqrt{Var} + \Delta s. \tag{2.24}$$

Let us now check if $p_-$ is non-negative, that means, if its numerator is non-negative. Substituting (2.18) or (2.20) into it, and further supposing $\mu \geq 0$, we get

$$(s_o - E)(s_+ - E) + Var = -\mu \Delta_j t (s_+ - s_o - \mu \Delta_j t) + Var \tag{2.25}$$

$$\geq -\mu \Delta_j t \left( \sqrt{Var} + \Delta s \right) + Var = -\mu \Delta_j t \left( \sigma \sqrt{\Delta_j t} + \Delta s \right) + \sigma^2 \Delta_j t. \tag{2.26}$$

This is greater than 0 if $\Delta s \leq \sigma^2/\mu - \sigma \sqrt{\Delta_j t}$. A completely analogous analysis can be done for the case of negative drift $\mu < 0 \Rightarrow E < x_i$. Joining both cases into one condition, we get that if $s_o = s_i$ and (2.17), (2.18) or (2.19), (2.20) holds, then

$$\Delta s \leq \frac{\sigma(s_i, t_j, \theta)^2}{|\mu(s_i, t_j, \theta)|} - \sigma(s_i, t_j, \theta)\sqrt{\Delta_j t} \tag{2.27}$$

is a sufficient condition for non-negativity of $p_-, p_o, p_+$ defined in (2.10)-(2.12). Last question is, if $s_- < s_o < s_+$ holds. This may be a problem in (2.17) (in case of a positive drift) or in (2.18) (in case of a negative drift). However it is easy to check that the condition (2.27) is sufficient for this inequality to be fulfilled. The condition (2.27) is quite weak for $\sigma$ large enough. However for problems with vanishing $\sigma$ it may be hard or even impossible to fulfill. In the next section we will describe how to tackle this problem.

## 2.4. The Tree-Grid Method with artificial diffusion

Let us now examine the case that condition (2.27) is not fulfilled. This can only happen if the variance $Var$ defined in (2.9) is not large enough to compensate the negative part in (2.25). We solve this problem by redefining the variance $Var$ in such manner that we add to the variance (2.9) some additional positive term of higher order in $\Delta_j t$:

$$Var := \sigma(s_i, t_j, \theta)^2 \Delta_j t + a(s_i, t_j, \theta)^2 (\Delta_j t)^2. \tag{2.28}$$

Here $a(s_i, t_j, \theta)^2 (\Delta_j t)^2$ is the so-called artificial diffusion term, and if large enough, use of this new modified variance (2.28) should lead to positive weights. Moreover, as the whole term should be vanishing with $\Delta_j t \to 0$ and the true variance term $\sigma \Delta_t$ should dominate. For this, we need however $a(s_i, t_j, \theta)$ (later denoted also simply as $a$) to be bounded. Now assuming positive $\mu$, we will repeat the analysis (2.25),(2.26) of the numerator of $p_-$ with the new $Var$ (results for $p_o$ and $p_+$ still hold).

$$(s_o - E)(s_+ - E) + Var \geq -\mu \Delta_j t \left( \sqrt{Var} + \Delta s \right) + Var \tag{2.29}$$

$$= -\mu \Delta_j t \left( \sqrt{\sigma^2 \Delta_j t + a^2 (\Delta_j t)^2} + \Delta s \right) + \sigma^2 \Delta_j t + a^2 (\Delta_j t)^2$$

$$\geq -\mu \Delta_j t \left( \sigma \sqrt{\Delta_j t} + a \Delta_j t + \Delta s \right) + \sigma^2 \Delta_j t + a^2 (\Delta_j t)^2. \tag{2.30}$$

In the last step, we decided that $a$ will be positive, so that we can do the above estimation. Now in order to introduce as small artificial diffusion as needed, but still having (2.30) non-negative, we will choose $a$ as the root of $-\mu \Delta_j t \left( \sigma \sqrt{\Delta_j t} + a \Delta_j t + \Delta s \right) + \sigma^2 \Delta_j t + a^2 (\Delta_j t)^2 = 0$. Moreover to ensure positivity of $a$, we will choose the larger of both roots:

$$a = \frac{\mu \Delta_j t + \sqrt{\mu^2 (\Delta_j t)^2 - 4|\mu| \Delta_j t \left( \sigma^2 / |\mu| - \sigma \sqrt{\Delta_j t} - \Delta s \right)}}{2 \Delta_j t} \tag{2.31}$$

We should note, that we substituted a non-negative $\mu$ with $|\mu|$ so that (2.31) holds also as result of fully analogous analysis for negative $\mu$.

If $\left( \sigma^2 / \mu - \sigma \sqrt{\Delta_j t} - \Delta s \right) \geq 0$, then condition (2.27) is fulfilled and we can switch to $a = 0$. Let us now examine the case that $\left( \sigma^2 / \mu - \sigma \sqrt{\Delta_j t} - \Delta s \right) < 0$; in that case, the whole discriminant and therefore also $a$ is positive. That means, the numerator of $p_-$ as well as $p_-$ itself is positive and we found positive probabilities $p_-, p_o, p_+$. Now we will try to find an upper bound on $a$, satisfying convergence of new variance to the true variance

from the problem setting. Following (2.31),

$$
\begin{aligned}
a &\leq \frac{\mu\Delta_j t + \sqrt{\mu^2(\Delta_j t)^2 - 4|\mu|\Delta_j t \min_{\sigma\in\mathbb{R}^+}\left(\sigma^2/|\mu| - \sigma\sqrt{\Delta_j t} - \Delta s\right)}}{2\Delta_j t} \\
&= \frac{\mu\Delta_j t + \sqrt{2\mu^2(\Delta_j t)^2 + 4|\mu|\Delta_j t\Delta s}}{2\Delta_j t} \\
&\leq \frac{\mu\Delta_j t + \sqrt{2\mu^2(\Delta_j t)^2} + \sqrt{4|\mu|\Delta_j t\Delta s}}{2\Delta_j t} \\
&= \frac{1}{\Delta_j t}\left(\frac{(1+\sqrt{2})|\mu|}{2}\Delta_j t + 2\sqrt{|\mu|}\sqrt{\Delta_j t\Delta s}\right).
\end{aligned} \tag{2.32}
$$

Let us define the abbreviations:

$$
m_1 = (1+\sqrt{2})|\mu|/2, \quad m_2 = 2\sqrt{|\mu|}. \tag{2.33}
$$

Following the estimation, for the whole artificial diffusion term holds:

$$
\begin{aligned}
a^2(\Delta_j t)^2 &= (m_1\Delta_j t + m_2\sqrt{\Delta_j t\Delta s})^2 \\
&= m_1^2(\Delta_j t)^2 + 2m_1 m_2\Delta_j t\sqrt{\Delta_j t\Delta s} + m_2^2\Delta_j t\Delta s \\
&\leq m_1^2(\Delta_j t)^2 + m_1 m_2\Delta_j t(\Delta_j t + \Delta s) + m_2^2\Delta_j t\Delta s \\
&= m_1(m_1 + m_2)(\Delta_j t)^2 + m_2(m_1 + m_2)\Delta_j t\Delta s \\
&= \mathcal{O}(\Delta t(\Delta t + \Delta s)),
\end{aligned} \tag{2.34}
$$

where $\Delta t = \max_{j\in\{1,2,\ldots,M-1\}}\Delta_j t$. Together with (2.28) we get the estimate

$$
Var = \mathcal{O}(\Delta t). \tag{2.35}
$$

We will use this estimation of the artificial diffusion term later to prove consistency.

## 2.5. The final algorithm

In the following algorithm, we are interested in the values $v_i^{j+1}$ corresponding to the states $s_-$, $s_o$ and $s_+$. Therefore, we define the following function:

$$
\begin{aligned}
\text{if} \quad &s \in \{s_1, s_2, \ldots, s_N\} : v^{j+1}(s) = v_k^{j+1} \text{ so that } s = s_k \\
&\text{else if} \quad s < s_1 : v^{j+1}(s) = BC_L(s, t_{j+1}) \\
&\text{else if} \quad s > s_N : v^{j+1}(s) = BC_R(s, t_{j+1}).
\end{aligned} \tag{2.36}
$$

Moreover, we define the short notation $v_-^{j+1} = v^{j+1}(s_-)$, $v_o^{j+1} = v^{j+1}(s_o)$, $v_+^{j+1} = v^{j+1}(s_+)$ and $f_i^j(\theta) = f(s_i, t_j, \theta(s_i, t_j))$, $r_i^j(\theta) = r(s_i, t_j, \theta(s_i, t_j))$. Now, assuming $S_{t_j} = s_i$ in order to discretize the equation (1.3) we use the following approximations:

- $\int_{t_j}^{t_{j+1}} \exp\left(\int_{t_j}^{k} r(S_l, l, \theta(S_l, l))dl\right) f(S_k, k, \theta(S_k, k))dk \approx f_i^j(\theta)\Delta_j t$

- $\exp\left(\int_{t_j}^{t_{j+1}} r(S_k, k, \theta(S_k, k))dk\right) \approx 1 + r_i^j(\theta)\Delta_j t$

- $\mathbb{E}\left(V(S_{t_{j+1}}, t_{j+1})\Big|S_{t_j} = s_i\right) \approx p_- v_-^{j+1} + p_o v_o^{j+1} + p_+ v_+^{j+1}$

Then the discretized version of the dynamic programming equation (1.3) for $i = 2, 3, \ldots, N-1$ reads

$$v_i^j = \max_{\theta \in \Theta}\left(f_i^j(\theta)\Delta_j t + (1 + r_i^j(\theta)\Delta_j t)\left(p_- v_-^{j+1} + p_o v_o^{j+1} + p_+ v_+^{j+1}\right)\right). \tag{2.37}$$

or

$$v_i^j = \max_{\theta \in \Theta} w_i^j(\theta), \tag{2.38}$$

$$w_i^j(\theta) = f_i^j(\theta)\Delta_j t + (1 + r_i^j(\theta)\Delta_j t)\left(p_- v_-^{j+1} + p_o v_o^{j+1} + p_+ v_+^{j+1}\right). \tag{2.39}$$

We should note, that unicity of the maximum in (2.38) is not needed. For $i = 1$ and $i = N$ we employ the boundary conditions:

$$v_1^j = BC_L(s_1, t_j), \quad v_N^j = BC_R(s_N, t_j). \tag{2.40}$$

Finally we can summarize the whole algorithm of the Tree-Grid Method for solving the SCP (1.1),(1.2) (and the HJB equation (1.4),(1.5)):

**Algorithm 2.1** (The Tree-Grid method).
1: *Set $v_i^M = V_T(s_i)$ for $i = 1, 2, \ldots, N$*
2: **for** $j = M-1, M-2, \ldots, 1$ **do**
3:     *Compute $v_1^j$, $v_N^j$ according to (2.40)*
4:     **for** $i = 2, 3, \ldots, N-1$ **do**
5:         **for** $\theta \in \Theta$ **do**
6:             *Compute E according to (2.8)*
7:             **if** *Condition (2.27) holds* **then**
8:                 *Compute Var according to (2.9)*
9:             **else**
10:                *Compute a according to (2.31)*
11:                *Compute Var according to (2.28)*
12:            **end if**
13:            *Set $s_o = s_i$ and compute $s_-, s_+$ using (2.17)-(2.18) or (2.19)-(2.20)*
14:            *Compute $p_-, p_o, p_+$ using (2.10)-(2.12)*
15:            *Using (2.36) and (2.39) compute $w_i^j(\theta)$*
16:        **end for**
17:        *Compute $v_i^j$ according to (2.38)*
18:    **end for**
19: **end for**

## 2.6. Relationship to other numerical methods

In this section we outline (very informally) the interesting relationships between our new method and standard approaches, as well as point out the most relevant differences.

**Forward shooting grid methods**. We already discussed FSG methods in Section 2.2 in order to motivate the Tree-Grid approach. We can see the Tree-Grid Method in it's simplest version (no artificial diffusion) as modification of the FSG method with 3 "branches" instead of 2, non-constant probabilities, and, most importantly with no need to perform any interpolation. This differences however make the method convergent in contrast to the (general) FSG.
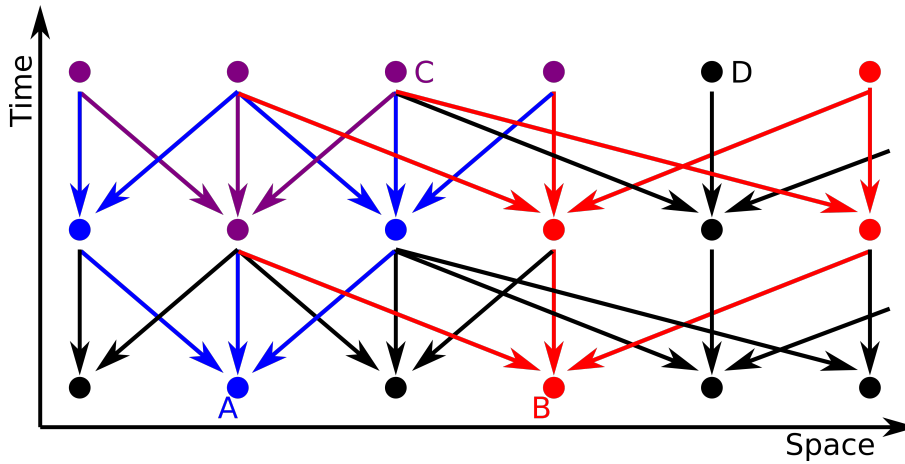


Figure 1: Illustration of the flexibility of the Tree-Grid structure. The node A represents a node with lower volatility, whereas node B exhibits higher volatility. Blue nodes are passing its values to node A and red nodes are passing its values to node B. Purple nodes have impact on both A and B. Node C passes its value to four different nodes on previous time level whereas node D passes its value only to one node in previous time level.

**Tree methods**. The Tree-Grid Method has many similarities with binomial and especially trinomial tree methods mostly used in option pricing. Both approaches are based on approximating the continuum of possible outcomes after 1 time-step by a RV gaining only 3 (in case of Tree-Grid and trinomial tree methods) values, graphically often represented by three new "branches" of the "tree". However, in contrast to the trinomial tree method where the branches are growing only from nodes on the tree, in Tree-Grid Method, 3 branches are "planted" in each gridpoint of an arbitrary grid. Therefore, we also chose the name "Tree-Grid" method. In trinomial tree methods, we get the value of the value function (e.g. option price) only in 1 space point in the first time layer, whereas in Tree-Grid Method, we get the value of value function on a whole set of space points. One may correctly comment, that also trinomial tree method works on some lattice that can be easily extended so that more values are computed in the first time-layer. This is true, however this lattice is constructed depending on the problem, space-steps are already determined by timestep (can't be chosen according to terminal condition) and the time-step size also can't be determined for each layer arbitrarily. On the other hand the tree grid method works on

an arbitrary grid. Besides artificial diffusion, one of the most obvious technical differences is the following: in trinomial tree methods from each node grow 3 branches, and each ("inner") node also passes its value into 3 earlier nodes. However in tree grid method only the first statement holds: each node may pass it's value to different number of earlier nodes –depending on the problem and on the grid. The "tree structure" in Tree-Grid Method is then much more flexible. This flexibility is illustrated in Figure 1.

**Finite difference methods**. FDMs are used to solve the HJB equation (1.4),(1.5) instead of the original SCP. However, in the next section, we will also prove the convergence of the approximation computed by the Tree-Grid Method to the solution of the HJB equation, despite the fact that the derivation of the method was based on the original formulation (1.1),(1.2). This motivates us to try to look at the method through a finite difference perspective. As $s_o = s_i$ (and therefore $v_o^{j+1} = v_i^{j+1}$) it can be shown:

$$
\begin{aligned}
&p_- v_-^{j+1} + p_o v_o^{j+1} + p_+ v_+^{j+1} \\
&= v_o^{j+1} + \mu \Delta_j t D_1 v_i^{j+1} + 1/2 \left( Var + (\mu \Delta_j t)^2 \right) D_2 v_i^{j+1},
\end{aligned}
\tag{2.41}
$$

where $D_1$ and $D_2$ denote standard finite difference approximations of first and second derivative on nonuniform grids:

$$
D_1 v_i^{j+1} = \left( \frac{s_+ - s_i}{s_+ - s_-} \right) \frac{v_i^{j+1} - v_-^{j+1}}{s_i - s_-} + \left( \frac{s_i - s_-}{s_+ - s_-} \right) \frac{v_+^{j+1} - v_i^{j+1}}{s_+ - s_-},
\tag{2.42}
$$

$$
D_2 v_i^{j+1} = \left( \frac{v_+^{j+1} - v_i^{j+1}}{s_+ - s_i} - \frac{v_i^{j+1} - v_-^{j+1}}{s_i - s_-} \right) \bigg/ \left( \frac{s_+ - s_-}{2} \right).
\tag{2.43}
$$

Now if we substitute (2.41) into (2.37) and suppose that no artificial diffusion is used, and for the discount rate holds $r_i^j(\theta) = 0$, the only difference between the Tree-Grid scheme and an explicit finite difference approximation with an "wide stencil" on the nodes $s_-, s_i, s_+$ is the term $1/2(\mu \Delta_j t)^2 D_2 v_i^{j+1}$. This term can be interpreted as some "inherent artificial diffusion" that comes into the scheme directly from numerical modeling, and is also making the scheme more stable than explicit FDMs (it in fact makes the derivation of condition (2.27) possible). Therefore, even in this simplest case this scheme can't be viewed as just FDM on specifically chosen nodes, although the similarity is clear. Let us note that such "inherent artificial diffusion term" is also present in tree methods (therefore they are not equivalent to FDM as sometimes stated in literature, only equivalent up to certain order), but not present in Markov chain approximation methods from [14].

**Markov chain approximation methods**. The basic idea of Markov chain approximation methods is to construct a Markov chain (in discrete time) approximating a Markov process (1.2) (in continuous time) and then using this chain to find an approximation of the solution to the SCP -an idea very similar to Tree methods, however in literature used in more general frameworks as tree-methods are used mostly only in option pricing. From this viewpoint Tree-Grid Method can be also seen as Markov chain approximation method as by constructing variable $\tilde{S}_{t_{j+1}}$ gaining values $s_-, s_o, s_+$ with probabilities $p_-, p_o, p_+$. In

Section 2.3 we in fact construct a Markov chain approximating (1.2). However, in standard Markov chain methods presented in [14], (as well as in tree methods), the grid can't be chosen arbitrarily, is problem-dependent and the space step is determined by the time step. Moreover, these methods can be often linked to explicit FDM and (as stated earlier) also do not possess the inherent artificial diffusion term.

## 3. Convergence of the Tree-Grid Method

In the previous section we directly discretized the SDE (1.2) to find the approximation of the solution of the SCP. However, in order to show the convergence of this approximation to the viscosity solution as the stepsizes tend to zero we will look at the above algorithm as on a method for solving the HJB equation (1.4),(1.5). At first, we will present the required definitions and the convergence theorem for general nonlinear problems from [3].

### 3.1. The convergence theory

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 [6]) of the equation $FV(x) = 0$, and denote this solution simply by $V(x)$. To find some approximation of the viscosity solution we define a **discrete approximation scheme**

$$Gv(x) = G\big(v(x), v(x + b_1 h), v(x + b_2 h), \ldots, v(x + b_n h)\big), \tag{3.1}$$

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

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

$$X_h = \{x_i \in \mathbb{R}^K | i = 1, 2, \ldots, N_h\}, \tag{3.2}$$

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

**Definition 3.1** (Numerical scheme). *The system of equations $Gv(x) = 0$ with $x \in X_h$ depending on a parameter $h \in \mathbb{R}^+$ is called numerical scheme.*

The 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 denote an approximation of the solution of $FV(x) = 0$ computed by solving the 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 3.2** (Monotonicity). *A discrete approximation scheme*
$Gv(x) = G\big(v(x), v(x + b_1 h), v(x + b_2 h), \ldots, v(x + b_n h)\big)$ *is monotone, if the function $G$ is non-increasing in $v(x + b_i h)$ for $b_i \neq 0$, $i = 1, \ldots, n$.*

**Definition 3.3** (Consistency)**.** *The scheme* $Gv(x) = G(v(x), v(x + b_1 h), v(x + b_2 h), \ldots, v(x + b_n h))$ *is a consistent approximation of* $FV(x)$, *if* $\lim_{h \to 0} |F\phi(x) - G\phi(x)| = 0$, *for any* $C^\infty$- *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. In literature, often $C^2$-smooth test functions are used. However, as shown for example in [15], this leads to an equivalent definition.

**Definition 3.4** (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 following Theorem of Barles and Souganidis in [3] is the key for proving convergence of a numerical scheme approximating a nonlinear PDE:

**Theorem 3.1** (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 above mentioned strong uniqueness property [3] is a property of the problem and not of the numerical scheme. Therefore, we will simply assume that our problems possess this property without actually proving it.

## 3.2. Consistency of the scheme

For the purposes of following analysis we rewrite (2.37) it in the form $Gv(s_i, t_j) = 0$:

$$
\begin{aligned}
Gv(s_i, t_j) =& G(v(s_i, t_j), v(s_-, t_{j+1}), v(s_o, t_{j+1}), v(s_+, t_{j+1})) \\
=& \frac{1}{\Delta_j t} \Big( v_i^j - \max_{\theta \in \Theta} \Big( f_i^j(\theta) \Delta_j t + (1 + r_i^j(\theta) \Delta_j t) \\
& \cdot \Big( p_- v_-^{j+1} + p_o v_o^{j+1} + p_+ v_+^{j+1} \Big) \Big) \Big) = 0.
\end{aligned}
\tag{3.3}
$$

Using theory from the section 3.1, our goal is to show that equation (3.3) is a monotone, consistent, and stable approximation of the nonlinear differential operator $F$ defined by the PDE (1.4):

$$
FV(s, t) = -\frac{\partial V}{\partial t} - \max_{\theta \in \Theta} \left( \frac{\sigma(\cdot)^2}{2} \frac{\partial^2 V}{\partial s^2} + \mu() \frac{\partial V}{\partial s} + r(\cdot)V + f(\cdot) \right).
\tag{3.4}
$$

Let us note that we multiplied both sides of (1.4) with $-1$ so that the operator $F$ is elliptic as in the theory of Barles and Souganidis [3]. The variable $x$ from section 3.1 is here represented by a 2-dimensional vector $[s, t]$. Let us recall $\Delta s = \max_{i \in \{1, 2, \ldots, N-1\}} \Delta_i s$ and $\Delta t = \max_{j \in \{1, 2, \ldots, M-1\}} \Delta_j t$. Then the stepsize parameter $h$ from the section 3.1 is in our case defined as $h = \min(\Delta s, \Delta t)$.

At first we prove the consistency of the scheme (3.3) with the HJB equation (1.4). We define $\Delta_{i-}s = s_- - s_i$, $\Delta_{i0}s = s_o - s_i$, $\Delta_{i+}s = s_+ - s_i$ and rewrite (2.10)-(2.12) equivalently as

$$p_- = \frac{(\Delta_{i0}s - \mu\Delta_j t)(\Delta_{i+}s - \mu\Delta_j t) + Var}{(\Delta_{i-}s - \Delta_{i0}s)(\Delta_{i-}s - \Delta_{i+}s)}, \tag{3.5}$$

$$p_o = \frac{(\Delta_{i-}s - \mu\Delta_j t)(\Delta_{i+}s - \mu\Delta_j t) + Var}{(\Delta_{i0}s - \Delta_{i-}s)(\Delta_{i0}s - \Delta_{i+}s)}, \tag{3.6}$$

$$p_+ = \frac{(\Delta_{i-}s - \mu\Delta_j t)(\Delta_{i0}s - \mu\Delta_j t) + Var}{(\Delta_{i+}s - \Delta_{i-}s)(\Delta_{i+}s - \Delta_{i0}s)}. \tag{3.7}$$

Now we can see that $p_-, p_o, p_+$ is also a solution of the system of equations

$$p_- + p_o + p_+ = 1, \tag{3.8}$$

$$p_-\Delta_{i-}s + p_o\Delta_{i0}s + p_+\Delta_{i+}s = \mu\Delta_j t, \tag{3.9}$$

$$p_-(\Delta_{i-}s)^2 + p_o(\Delta_{i0}s)^2 + p_+(\Delta_{i+}s)^2 = Var + (\mu\Delta_j t)^2. \tag{3.10}$$

Now a lemma about a remainder-terms that will be used in our consistency proof follows.

**Lemma 3.1** (Rest terms). *Let $\Delta_{i0}s = 0$ ($s_o = s_i$). Then for $s_-, s_+$ computed according to* (2.17),(2.18) *or* (2.19),(2.20) *we have*

$$R_3 := p_-(\Delta_{i-}s)^3 + p_o(\Delta_{i0}s)^3 + p_+(\Delta_{i+}s)^3 = \mathcal{O}\left(\Delta t\left(\sqrt{\Delta t} + \Delta s\right)\right), \tag{3.11}$$

*and for $s_-, s_+$ computed according to* (2.19),(2.20) *on equidistant grid, we have*

$$R_3 = \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right). \tag{3.12}$$

*Moreover, for $s_-, s_+$ computed either by formulas* (2.17),(2.18), *or by* (2.19),(2.20) *it holds*

$$R_4^b := b_-p_-(\Delta_{i-}s)^4 + b_0 p_o(\Delta_{i0}s)^4 + b_+p_+(\Delta_{i+}s)^4$$
$$= \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right), \tag{3.13}$$

$$R_2^b := b_-p_-(\Delta_{i-}s)^2 + b_0 p_o(\Delta_{i0}s)^2 + b_+p_+(\Delta_{i+}s)^2 = \mathcal{O}(\Delta t), \tag{3.14}$$

*where $b_-, b_0, b_+ \in \mathbb{R}$ are arbitrary constants.*

*Proof.* From (3.5)-(3.7) follows:

$$R_3 = \Delta_{i-}s\Delta_{i0}s\Delta_{i+}s - \mu\Delta_j t\Delta_{i0}s\Delta_{i+}s - \mu\Delta_j t\Delta_{i-}s\Delta_{i+}s - \mu\Delta_j t\Delta_{i-}s\Delta_{i0}s$$
$$+ Var\Delta_{i+}s + Var\Delta_{i0}s + Var\Delta_{i-}s$$
$$+ \mu^2(\Delta_j t)^2\Delta_{i+}s + \mu^2(\Delta_j t)^2\Delta_{i0}s + \mu^2(\Delta_j t)^2\Delta_{i-}s. \tag{3.15}$$

According to (2.35) $Var = \mathcal{O}(\Delta t)$. Substituting this into (2.17),(2.18) or (2.19),(2.20) using the inequality $s_i < s_o = s_i < s_+$ and using definitions of $\Delta_{i-}s, \Delta_{i+}s$, we get

$$\Delta_{i+}s = \mathcal{O}(\sqrt{\Delta t} + \Delta s), \quad \Delta_{i-}s = \mathcal{O}(\sqrt{\Delta t} + \Delta s). \tag{3.16}$$

As $\Delta_{i0}s = 0$ for $s_o = s_i$, by using formulas (2.17),(2.18) we get

$$\begin{aligned}
R_3 = &-\mu\Delta_j t \Delta_{i-}s\Delta_{i+}s + Var\Delta_{i+}s + Var\Delta_{i-}s \\
&+ \mu^2(\Delta_j t)^2\Delta_{i+}s + \mu^2(\Delta_j t)^2\Delta_{i-}s = \mathcal{O}\left(\Delta t\left(\sqrt{\Delta t} + \Delta s\right)\right),
\end{aligned} \tag{3.17}$$

where we used (2.35), (3.16). By using formulas (2.19),(2.20) on an equidistant, it holds $\Delta_{i-}s = -\Delta_{i+}s$ and therefore

$$R_3 = -\mu\Delta_j t \Delta_{i-}s\Delta_{i+}s = \mu\Delta_j t(\Delta_{i+}s)^2 = \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right). \tag{3.18}$$

Let us define $R_4 := p_-(\Delta_{i-}s)^4 + p_o(\Delta_{i0}s)^4 + p_+(\Delta_{i+}s)^4$. Now by using either (2.17),(2.18) or (2.19),(2.20) we can prove that $R_4 = \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right)$ in the same manner as in the case of $R_3$. Now it holds

$$\begin{aligned}
\mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right) = \min(b_-, b_0, b_+)R_4 &\leq R_4^b \\
&\leq \max(b_-, b_0, b_+)R_4 = \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right),
\end{aligned}$$

and therefore also $R_4^b = \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right)$. Finally, following (3.10), (2.35) it holds

$$\begin{aligned}
\mathcal{O}\left(\Delta t\right) = \min(b_-, b_0, b_+)\left(Var + (\mu\Delta_j t)^2\right) &\leq R_2^b \\
&\leq \max(b_-, b_0, b_+)\left(Var + (\mu\Delta_j t)^2\right) = \mathcal{O}\left(\Delta t\right)
\end{aligned}$$

and therefore $R_2^b = \mathcal{O}\left(\Delta t\right)$.

The lemma establishing the consistency of our scheme follows.

**Lemma 3.2** (Consistency)**.** *If the parameters $p_-$, $p_o$, $p_+$ in the scheme* (3.3) *satisfy the conditions* (2.5)-(2.7)*, $s_o = s_i$ and $s_-, s_+$ are computed according to* (2.17),(2.18) *then the scheme* (3.3) *is consistent with the PDE* (3.4)*.*

*Proof.* Let $\phi : \mathbb{R} \times [0, T] \to \mathbb{R}$ be a $C^\infty$-smooth function. Let us define $\phi_i^j = \phi(s_i, t_j)$, $\phi_-^j = \phi(s_-, t_j)$, $\phi_0^j = \phi(s_o, t_j)$, $\phi_+^j = \phi(s_+, t_j)$. Now it holds

$$\begin{aligned}
\phi_*^{j+1} = &\phi_i^j + \frac{\partial\phi_i^j}{\partial s}\Delta_{i*}s + \frac{\partial\phi_i^j}{\partial t}\Delta_j t + \frac{1}{2}\frac{\partial^2\phi_i^j}{\partial s^2}(\Delta_{i*}s)^2 + \frac{\partial^2\phi_i^j}{\partial s\partial t}\Delta_{i*}s\Delta_j t \\
&+ \frac{1}{6}\frac{\partial^3\phi_i^j}{\partial s^3}(\Delta_{i*}s)^3 + \frac{1}{2}\frac{\partial^3\phi_i^j}{\partial s^2\partial t}(\Delta_{i*}s)^2\Delta_j t \\
&+ \frac{1}{24}\frac{\partial^4\phi_{i+\delta*}^{j+\epsilon*}}{\partial s^4}(\Delta_{i*}s)^4 + \frac{1}{6}\frac{\partial^4\phi_{i+\delta*}^{j+\epsilon*}}{\partial s^3\partial t}(\Delta_{i*}s)^3\Delta_j t + \mathcal{O}\left(\left(\Delta_j t\right)^2\right),
\end{aligned} \tag{3.19}$$

where the index $*$ should be substituted by either $-$ or $0$ or $+$, and $\phi_{i+\delta*}^{j+\epsilon*} = \phi(s_i+\delta_*\Delta_{i*}s, t_j+\epsilon_*\Delta_j t)$, and $\delta_*, \epsilon_* \in [0,1]$. Now, using (3.8)-(3.10) and the definitions from Lemma 3.1 we get:

$$
\begin{aligned}
p_-\phi_-^{j+1} + p_o\phi_0^{j+1} + p_+\phi_+^{j+1} &= \phi_i^j + \frac{\partial \phi_i^j}{\partial s}\mu\Delta_j t + \frac{\partial \phi_i^j}{\partial t}\Delta_j t \\
&+ \frac{1}{2}\frac{\partial^2 \phi_i^j}{\partial s^2}\left(Var + (\mu\Delta_j t)^2\right) + \frac{\partial^2 \phi_i^j}{\partial s\partial t}\mu(\Delta_j t)^2 + \frac{1}{6}\frac{\partial^3 \phi_i^j}{\partial s^3}R_3 \\
&+ \frac{1}{2}\frac{\partial^3 \phi_i^j}{\partial s^2\partial t}\left(Var + (\mu\Delta_j t)^2\right)\Delta_j t + R_4^b + R_2^{b'}\Delta_j t + \mathcal{O}\left((\Delta_j t)^2\right),
\end{aligned}
\tag{3.20}
$$

where

$$
b_* = \frac{1}{24}\frac{\partial^4 \phi_{i+\delta*}^{j+\epsilon*}}{\partial s^4}, \quad b_*' = \frac{1}{6}\frac{\partial^4 \phi_{i+\delta*}^{j+\epsilon*}}{\partial s^3\partial t}\Delta_{i*}s, \quad \text{for } * = -, 0, +.
\tag{3.21}
$$

Now, using Lemma 3.1, and definition of $Var$ (2.28) we can rewrite (3.20) as

$$
\begin{aligned}
p_-\phi_-^{j+1} + p_o\phi_0^{j+1} + p_+\phi_+^{j+1} &= \phi_i^j + \frac{\partial \phi_i^j}{\partial s}\mu\Delta_j t + \frac{\partial \phi_i^j}{\partial t}\Delta_j t \\
&+ \frac{1}{2}\frac{\partial^2 \phi_i^j}{\partial s^2}\left(\sigma^2\Delta_j t + (a^2+\mu^2)(\Delta_j t)^2\right) + \frac{\partial^2 \phi_i^j}{\partial s\partial t}\mu(\Delta_j t)^2 + \frac{1}{6}\frac{\partial^3 \phi_i^j}{\partial s^3}R_3 \\
&+ \frac{1}{2}\frac{\partial^3 \phi_i^j}{\partial s^2\partial t}\left(\sigma^2\Delta_j t + (a^2+\mu^2)(\Delta_j t)^2\right)\Delta_j t + \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right) \\
&= \phi_i^j + \frac{\partial \phi_i^j}{\partial s}\mu\Delta_j t + \frac{\partial \phi_i^j}{\partial t}\Delta_j t + \frac{1}{2}\frac{\partial^2 \phi_i^j}{\partial s^2}\sigma^2\Delta_j t \\
&+ \frac{1}{6}\frac{\partial^3 \phi_i^j}{\partial s^3}R_3 + R_a + \mathcal{O}\left(\Delta t\left(\Delta t + (\Delta s)^2\right)\right),
\end{aligned}
\tag{3.22}
$$

where

$$
R_a = \frac{1}{2}\frac{\partial^2 \phi_i^j}{\partial s^2}a^2(\Delta_j t)^2 + \frac{1}{2}\frac{\partial^3 \phi_i^j}{\partial s^2\partial t}a^2(\Delta_j t)^2,
\tag{3.23}
$$

and according to (2.34) $R_a = \mathcal{O}\left(\Delta t\left(\Delta t + \Delta s\right)\right)$ if $a > 0$ and $R_a = 0$ if $a = 0$ (artificial diffusion not needed). Now, substituting (3.22) to $G\phi(s_i, t_j)$ (defined according to (3.3))

we get

$$
\begin{aligned}
G\phi(s_i, t_j) &= \frac{1}{\Delta_j t}\Big(\phi_i^j - \max_{\theta\in\Theta}\Big(f_i^j(\theta)\Delta_j t + (1 + r_i^j(\theta)\Delta_j t)\Big(\phi_i^j + \frac{\partial\phi_i^j}{\partial s}\mu\Delta_j t \\
&\quad + \frac{\partial\phi_i^j}{\partial t}\Delta_j t + \frac{1}{2}\frac{\partial^2\phi_i^j}{\partial s^2}\sigma^2\Delta_j t + \frac{1}{6}\frac{\partial^3\phi_i^j}{\partial s^3}R_3 + R_a + \mathscr{O}\big(\Delta t(\Delta t + (\Delta s)^2)\big)\Big)\Big)\Big) \\
&= -\frac{\partial\phi_i^j}{\partial t} - \max_{\theta\in\Theta}\Big(f_i^j(\theta) + r_i^j(\theta)\phi_i^j + \frac{\partial\phi_i^j}{\partial s}\mu + \frac{1}{2}\frac{\partial^2\phi_i^j}{\partial s^2}\sigma^2 \\
&\quad + \frac{1}{6}\frac{\partial^3\phi_i^j}{\partial s^3}\Big(r_i^j(\theta)R_3 + \frac{R_3}{\Delta_j t}\Big) + R_a + \mathscr{O}\big(\Delta t + (\Delta s)^2\big)\Big) \\
&= -\frac{\partial\phi_i^j}{\partial t} - \max_{\theta\in\Theta}\Big(\frac{\sigma^2}{2}\frac{\partial^2\phi_i^j}{\partial s^2} + \mu\frac{\partial\phi_i^j}{\partial s} + r_i^j(\theta)\phi_i^j + f_i^j(\theta)\Big) + R, \qquad (3.24)
\end{aligned}
$$

where, according to the estimation of $R_3$ in Lemma 3.1 and according to estimation of $R_a$ $R = \mathscr{O}\big(\sqrt{(\Delta t)} + \Delta s\big)$ if formulas (2.17), (2.18) were used and $R = \mathscr{O}\big(\Delta t + \Delta s\big)$ if formulas (2.19), (2.20) were used, and artificial diffusion is present ($a > 0$) and $R = \mathscr{O}\big(\Delta t + (\Delta s)^2\big)$ if formulas (2.19), (2.20) were used, and artificial diffusion is not present ($a = 0$).

According to (3.4), we have

$$
F\phi(s_i, t_j) = -\frac{\partial\phi_i^j}{\partial t} - \max_{\theta\in\Theta}\Big(\frac{\sigma^2}{2}\frac{\partial^2\phi_i^j}{\partial s^2} + \mu\frac{\partial\phi_i^j}{\partial s} + r_i^j(\theta)\phi_i^j + f_i^j(\theta)\Big). \qquad (3.25)
$$

Comparing (3.24) and (3.25) we see that $|F\phi(s_i, t_j) - G\phi(s_i, t_j)|$ is of order $\mathscr{O}\big(\sqrt{(\Delta t)} + \Delta s\big)$ resp. $\mathscr{O}(\Delta t + \Delta s)$ resp. $\mathscr{O}\big(\Delta t + (\Delta s)^2\big)$ and therefore vanishing with $h = \min(\Delta s, \Delta t) \to 0$.

**Remark 3.1** (Order of consistency). *The original paper of Barles and Souganidis [3] does not define the order of convergence to the viscosity solution, it just presents a theory for convergence. Therefore, the impact of the order of the scheme (or "consistency order") on the convergence rate is not clear. Moreover, as shown in [13] the maximal order of a monotone scheme is 2. However, in the work of Wang and Forsyth [19] it is experimentally shown that a higher order of the scheme leads to faster convergence for a particular problem. Therefore following the proof of the previous Lemma it may be advantageous to use formulas (2.19), (2.20) leading to an order of the scheme $\mathscr{O}(\Delta t + \Delta s)$ or even $\mathscr{O}\big(\Delta t + (\Delta s)^2\big)$ (if artificial diffusion is not needed) rather than formulas (2.17), (2.18) leading to an order of the scheme $\mathscr{O}\big(\sqrt{(\Delta t)} + \Delta s\big)$. On the other hand, formulas (2.17), (2.18) may lead to smaller space-steps $\Delta_{i-}, \Delta_{i+}$ which may theoretically also lead to a higher convergence rate. Therefore, the optimal choice between formulas (2.17), (2.18) and (2.19), (2.20) needs deeper examination.*

### 3.3. Monotonicity, stability, convergence

Next, we will prove the monotonicity and stability of the method. Together with the already proven consistency we get the convergence result for the method. The lemma

establishing the monotonicity property of our method follows.

**Lemma 3.3** (Monotonicity). *If the parameters $p_-$, $p_o$, $p_+$ in the scheme* (3.3) *satisfy the condition* (2.4) *and if $1 + r_i^j(\theta)\Delta_j t \geq 0$ for all $\theta \in \Theta$ then this scheme is monotone.*

*Proof.* In our case, monotonicity means that (3.3) is non-increasing in $v_-^{j+1}$, $v_o^{j+1}$, $v_+^{j+1}$. This follows directly from the non-negativity of the $p_-$, $p_o$, $p_+$ -condition (2.4).

**Remark 3.2.** *Even if $1 + r_i^j(\theta)\Delta_j t < 0$ for some $\theta$, we can get a monotone scheme if we substitute $1 + r_i^j(\theta)\Delta_j t$ by $1/(1 - r_i^j(\theta)\Delta_j t)$ in* (3.3) *for these parameters $\theta$. Note that this change does not harm consistency, nor stability of the scheme.*

The next step is to prove the stability of the method. At first we will pose some conditions on the problem that will be needed for the stability proof.

**Condition 3.1** (Stability condition on the problem). *We suppose that:*

1. *There exist constants $C_f, C_r$, $\forall s, t, \theta : |f(s, t, \theta)| < C_f$, $|r(s, t, \theta)| < C_r$.*

2. *There exist constant $C_L$, $|BC_L(s, t)| < C_L$, $\forall t$ and for all possible values $s < s_1$ of the variables $s_-, s_o, s_+$ for any grid.*

3. *There exist constant $C_R$, $|BC_R(s, t)| < C_R$, $\forall t$ and for all possible values $s > s_N$ of the variables $s_-, s_o, s_+$ for any grid.*

The first condition simply establish boundedness of functions $r(\cdot), f(\cdot)$, and the second and third condition establish boundedness of the values that can flow into the model from behind the boundary. Checking if this condition is fulfilled is in most cases trivial. Now we state an inequality that will be used for the stability proof.

**Lemma 3.4** (Inequality recurrence).

$$x_j \leq a_j x_{j+1} + b_j, j < M \Rightarrow x_j \leq \left(\prod_{k=j}^{M-1} a_k\right) x_M + \sum_{k=j}^{M-1}\left(\left(\prod_{l=j}^{k-1} a_l\right) b_k\right)$$

*Proof.* Proof can be done by induction, we let the details as exercise for the reader.

Lemma about the stability of the method follows.

**Lemma 3.5** (Stability). *If the parameters $p_-$, $p_o$, $p_+$ in the scheme* (3.3) *satisfy the conditions* (2.4), (2.5), *and the problem satisfies condition 3.1 then this scheme is stable.*

*Proof.* Let us define

$$C_j = \max\left(C_L, C_R, \max_{i\in\{1,2,\dots,N\}}\left(v_i^j\right)\right)$$

Then, it holds:

$$
\begin{aligned}
C_j ={}& \max\Big(C_L, C_R, \max_{i\in\{1,2,\dots,N\}} \big(f_i^j(\theta)\Delta_j t + (1+r_i^j(\theta)\Delta_j t) \\
& \cdot \big(p_- v_-^{j+1} + p_o v_o^{j+1} + p_+ v_+^{j+1}\big)\big)\Big) \\
\leq{}& \max\Big(C_L, C_R, \max_{i\in\{1,2,\dots,N\}} \big(C_f\Delta_j t + (1+C_r\Delta_j t) \\
& \cdot \max\big(C_L, C_R, \max_{i\in\{1,2,\dots,N\}}\big(v_i^{j+1}\big)\big)\big)\Big) \\
={}& C_f\Delta_j t + (1+C_r\Delta_j t)C_{j+1} \\
\leq{}& C_f\Delta_j t + \exp(C_r\Delta_j t)C_{j+1}.
\end{aligned}
$$

Using Lemma 3.4 we get

$$
\begin{aligned}
C_j ={}& \left(\prod_{k=j}^{M-1}\exp(C_r\Delta_k t)\right)C_M + \sum_{k=j}^{M-1}\left(\left(\prod_{l=j}^{k-1}\exp(C_r\Delta_l t)\right)C_f\Delta_k t\right) \\
={}& \exp\big(C_r(t_M - t_j)\big)C_M + \sum_{k=j}^{M-1}\big(\exp\big(C_r(t_k - t_j)\big)C_f\Delta_k t\big) \\
\leq{}& \exp(C_r T)C_M + \exp(C_r T)\sum_{k=j}^{M-1}\big(C_f\Delta_k t\big) \\
={}& \exp(C_r T)\big(C_M + C_f T\big) =: C.
\end{aligned}
$$

Let $v$ be the vector of all values $v_i^j$. Now it holds

$$
\|v\|_\infty = \max_{j\in\{1,2,\dots,M\}}\max_{i\in\{1,2,\dots,N\}}|v_i^j| \leq \max_{j\in\{1,2,\dots,M\}}C_j \leq C.
$$

As this estimation is independent of the grid spacing, the scheme is stable.

**Theorem 3.2** (Convergence of Tree-Grid Method)**.** *The approximation computed by the implicit Tree-Grid Method defined by Algorithm 2.1 for solving the SCP* (1.1),(1.2) *and the corresponding HJB equation* (1.4),(1.5) *satisfying the strong uniqueness property (see [3]) and stability conditions defined in Condition 3.1 converges to the viscosity solution solution of this SCP (and HJB equation).*

*Proof.* The proof follows from Property 3.1 and Lemmas 3.2, 3.3, 3.5.

**Remark 3.3.** *Actually the approximation converges to solution of the HJB equation restricted to the domain $[s_L, s_R]\times[0, T]$ with boundary conditions defined by functions $BC_L(s,t)$, $BC_R(s,t)$. We silently assumed that these boundary functions are chosen consistently with the viscosity solution - an assumption that is frequently done in literature, as choosing boundary conditions is problem specific.*

## 4. Numerical examples

In this section we will compare the performance of the Tree-Grid Method with the classic implicit finite difference method for the HJB equation presented in [8] on two examples from finance. In the first example no artificial diffusion is needed in contrast to the second example. We should note, that in order to validate the Tree-Grid method, we tested it also with uncontrolled Black-Scholes equation, which posses explicit solution. The method was convergent, however standard FDMs provided better results in this case. The numerical methods were implemented in Matlab and tested on Intel Core i7-4770 CPU 3.40GHz computer with 8 GB RAM.

### 4.1. The uncertain volatility model

Our first example of the usefulness of Tree-Grid Method is the problem of option pricing under the uncertain volatility model. This setting is similar to the famous Black-Scholes model, the only difference is that the volatility is uncertain, only known to lie in some interval. Using this model, we can compute maximal (best case) and minimal (worst case) option prices. Here we present the results for the best case option price $V$ that can be, according to [9], computed using the HJB equation

$$\frac{\partial V}{\partial t} + \max_{\theta \in \Theta}\Big(\frac{\theta^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + S\frac{\partial V}{\partial S} - rV\Big) = 0. \tag{4.1}$$

Here, $t$ represents time, $S$ is asset price, $\theta$ (control variable) is volatility, $\Theta = \{\sigma_{min}, \sigma_{max}\}$ are the minimal and maximal values of volatility and $r$ is the risk-free interest rate. For comparison reasons we used parameters from [9]: $r = 0.04$ and $\Theta = \{0.3, 0.45\}$.

**Computational domain:** The maturity of the option will be six months ($T = 0.5$), the space domain will be restricted to $S \in [0, 500]$. The grid will be uniformly spaced in time, and non-uniformly in space (nodes will be more dense near to "edges" of terminal condition and less dense near to boundaries of the computational domain)

**Terminal and boundary conditions:** Terminal and boundary conditions will be also set as in [9]. We will use a butterfly-spread payoff around 100 as the terminal condition:

$$V(T, S) = V_T(S) = \begin{cases} S - 95 & \text{if } 95 < x \leq 100 \\ = 105 - S & \text{if } 100 < S \leq 105 \, , \\ = 0 & \text{else.} \end{cases}$$

and the Dirichlet boundary conditions:

$$V(S_{min}, t) = BC_L(S) = 0, \quad V(S_{max}, t) = BC_R(S) = 0,$$
$$[S_{min}, S_{max}] = [0, 500].$$

**Numerical results:** Now we will present results of numerical solutions of the option pricing problem in uncertain volatility model computed on grids with different levels of

refinement. With $A^k$, let us denote the approximation computed on the $k$-th refinement level. $N_t^k$ will denote number of time-nodes on the $k - th$ refinement level, and $N_s^k$ will denote number of space-nodes on the $k - th$ refinement level. The error of the approximation on the $k$-th space- and time-refinement level is denoted as $Err\ A^k$ and estimated by the formula

$$Err\ A^k = \|A^k - A^{ref}\|_1, \tag{4.2}$$

where $A^{ref}$ denotes a reference solution. The experimental order of convergence on the $k$-th space- and time-refinement level is denoted denoted as $EOC\ A^k$ and computed using the formula

$$EOC\ A^k = \frac{\log(ErrA^k) - \log(ErrA^{k-1})}{\log(h_{k-1}) - \log(h_k)}. \tag{4.3}$$

As reference solution we will use an approximation computed on a grid with 25601 time-nodes and 50689 space nodes using the classic implicit method with maximal use of central differences [19]. The Tables 1 and 2 present the results for approximations computed with fixed ratio between time-step size and space-step size ($\Delta t = c \cdot \Delta s$) and with fixed ratio between time-step size and square of space-step size ($\Delta t = c \cdot (\Delta s)^2$). Figure 2 illustrates the results.

Table 1: Uncertain volatility model, $\Delta t = c \cdot \Delta s$. Error, experimental order of convergence and computational time of the approximation $A^{k,k}$ for the classic implicit and the Tree-Grid methods, for different numbers of nodes.

| k | $N_t^k$ | $N_s^k$ | Classic Implicit | | | Tree-Grid | | |
|---|---------|---------|------------------|------|----------|-----------|-------|--------|
| | | | Err | EOC | Time | Err | EOC | Time |
| 1 | 51 | 100 | 6.03E-005 | - | 0.0398 | 4.29E-005 | - | 0.0031 |
| 2 | 101 | 199 | 1.50E-005 | 2.01 | 0.0854 | 2.09E-006 | 4.36 | 0.0033 |
| 3 | 201 | 397 | 4.07E-006 | 1.88 | 0.1990 | 2.19E-006 | -0.07 | 0.0068 |
| 4 | 401 | 793 | 1.10E-006 | 1.88 | 0.5138 | 4.56E-007 | 2.26 | 0.0157 |
| 5 | 801 | 1585 | 2.89E-007 | 1.93 | 1.4758 | 1.77E-007 | 1.36 | 0.0386 |
| 6 | 1601 | 3169 | 7.11E-008 | 2.02 | 5.3508 | 2.43E-008 | 2.87 | 0.1057 |
| 7 | 3201 | 6337 | 1.60E-008 | 2.15 | 19.7059 | 1.32E-008 | 0.87 | 0.3350 |
| 8 | 6401 | 12673 | 2.99E-009 | 2.42 | 78.0347 | 2.30E-009 | 2.53 | 1.1445 |
| 9 | 12801 | 25345 | 3.36E-010 | 3.16 | 312.6208 | 2.88E-010 | 3.00 | 4.3767 |

From Tables 1 and 2 and Figure 2, it is clear that the Tree-Grid Method was not only significantly faster than the classic implicit FDM, but also its error was slightly smaller. Therefore the Tree-Grid method is clearly superior for this model. We should note, that in this example the condition (2.27) was always met and therefore no artificial diffusion was needed.

## 4.2. The passport option pricing problem

In this second example we will test the Tree-Grid Method and the classic implicit method on the HJB equation for passport option pricing. Passport options are contracts that allow

Table 2: Uncertain volatility model, $\Delta t = c \cdot (\Delta s)^2$. Error, experimental order of convergence and computational time of the approximation $A^{k,k}$ for the classic implicit and the Tree-Grid methods, for different numbers of nodes.

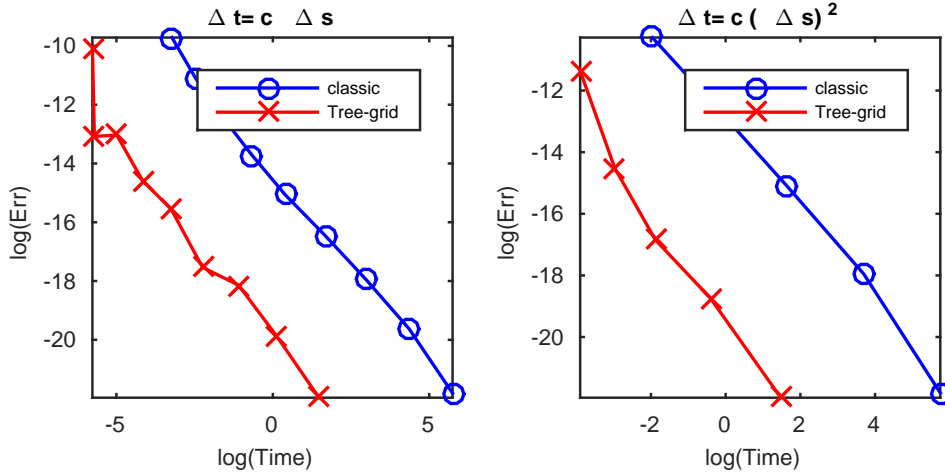|   |       |        | Classic Implicit |      |          | Tree-Grid |      |        |
|---|-------|--------|------------------|------|----------|-----------|------|--------|
| k | $N_t^k$ | $N_s^k$ | Err            | EOC  | Time     | Err       | EOC  | Time   |
| 1 | 51    | 1585   | 3.43E-005        | -    | 0.1406   | 1.16E-005 | -    | 0.0204 |
| 2 | 201   | 3169   | 3.63E-006        | 3.24 | 0.7650   | 4.95E-007 | 4.55 | 0.0515 |
| 3 | 801   | 6337   | 2.82E-007        | 3.69 | 5.0952   | 4.84E-008 | 3.35 | 0.1524 |
| 4 | 3201  | 12673  | 1.59E-008        | 4.15 | 40.3774  | 7.27E-009 | 2.74 | 0.6655 |
| 5 | 12801 | 25345  | 3.36E-010        | 5.57 | 312.6208 | 2.88E-010 | 4.66 | 4.3767 |



Figure 2: Uncertain volatility model. Comparison of natural logarithm of estimated absolute error of the approximation of solution against natural logarithm of computational time (in seconds) for the classic implicit, and Tree-Grid Method. (Illustration of results from Tables 1, 2)

the buyer to run trading account for a certain amount of time. After the maturity, the buyer of this contract can keep the profit, however the potential loss will be covered by the seller. Here we will examine the case in which the buyer is allowed to invest in one particular asset only. The price depends on buyer's wealth $W$, current asset price $S$ and time to maturity $t$. According to [19], [18], the HJB equation for the current price of the contract can be simplified to the form

$$\frac{\partial V}{\partial t} + \max_{|\theta| \leq 1} \Big( \frac{\sigma^2}{2}(x-\theta)^2 \frac{\partial^2 V}{\partial x^2}$$
$$+ \Big( (r - r_c - \gamma)\theta - (r - r_t - \gamma)x \Big) \frac{\partial V}{\partial x} - \gamma V \Big) = 0 \qquad (4.4)$$

Here, $t$ is time, $x = W/S$ and $V$ is the option price divided by $S$. By $r$, we denote the risk-free interest rate, $\gamma$ is the dividend rate, $r_c$ is the cost of carry rate, $r_t$ is the interest rate for the trading account and $\sigma$ is the volatility. The number of shares that the investor holds (control variable) is denoted by $\theta$, and it does not have to be an integer. In this case

the seller of the option requires the constraint $|\theta| \leq 1$. For comparison reasons we used the same parameter values as in [19]: $r = 0.08, \gamma = 0.03, r_c = 0.12, r_t = 0.05, \sigma = 0.2$.

**Computational domain:** The maturity of the option will be one year ($T = 1$), the space domain will be restricted to $[-3, 4]$. The grid will be uniformly spaced in time, and non-uniformly in space (nodes will be more dense near to zero and less dense near to the boundaries of the computational domain)

**Terminal and boundary conditions:** As terminal condition we will use the "capped" payoff:

$$V(T, x) = V_T(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ = x & \text{if } 0 < x \leq 1 \,, \\ = 1 & \text{if } x > 1 \end{cases}$$

and the Dirichlet boundary conditions:

$$V(x_{min}, t) = BC_L(x) = 0, \quad V(x_{max}, t) = BC_R(x) = 1,$$
$$[x_{min}, x_{max}] = [-3, 4].$$

**Numerical results:** In this part we use the same definitions of $A^k$, $Err$, $EOC$ and as in previous numerical model.

Table 3: Passport option pricing, $\Delta t = c \cdot \Delta s$. Error, experimental order of convergence and computational time of the approximation $A^{k,k}$ for the classic implicit and the Tree-Grid methods, for different numbers of nodes.

| k | $N_t^k$ | $N_s^k$ | Classic Implicit | | | Tree-Grid | | |
|---|---|---|---|---|---|---|---|---|
| | | | Err | EOC | Time | Err | EOC | Time |
| 1 | 51 | 24 | 3.42E-007 | - | 0.0301 | 1.65E-005 | - | 0.0037 |
| 2 | 101 | 47 | 1.03E-006 | -1.59 | 0.0569 | 1.64E-006 | 3.33 | 0.0051 |
| 3 | 201 | 93 | 3.41E-007 | 1.59 | 0.1205 | 1.00E-006 | 0.71 | 0.0089 |
| 4 | 401 | 185 | 6.26E-007 | -0.88 | 0.2825 | 2.73E-006 | -1.45 | 0.0177 |
| 5 | 801 | 369 | 4.32E-007 | 0.53 | 0.8047 | 2.04E-006 | 0.42 | 0.0393 |
| 6 | 1601 | 737 | 1.89E-007 | 1.19 | 2.0047 | 9.29E-007 | 1.13 | 0.0983 |
| 7 | 3201 | 1473 | 8.75E-008 | 1.11 | 5.8793 | 3.99E-007 | 1.22 | 0.2735 |
| 8 | 6401 | 2945 | 3.76E-008 | 1.22 | 21.2013 | 1.62E-007 | 1.30 | 0.8108 |
| 9 | 12801 | 5889 | 1.44E-008 | 1.38 | 77.8576 | 6.23E-008 | 1.38 | 2.8490 |
| 10 | 25601 | 11777 | 4.48E-009 | 1.68 | 312.0041 | 2.15E-008 | 1.53 | 11.2965 |
| 11 | 51201 | 23553 | 7.83E-010 | 2.52 | 1080.4906 | 6.01E-009 | 1.84 | 38.0828 |

As reference solution we will use the approximation computed on a grid with 102401 time-nodes and 47105 space nodes using again the classic implicit method with maximal use of central differences. As in the previous numerical example, the Tables 3 and 4 and Figure 3 illustrate the results. However, as the diffusion is vanishing in this case, artificial diffusion was needed in Tree-Grid Method in contrast to the uncertain volatility model. The error was smaller in the case of classic implicit FDM, however, taking into account the low

Table 4: Passport option pricing, $\Delta t = c \cdot (\Delta s)^2$. Error, experimental order of convergence and computational time of the approximation $A^{k,k}$ for the classic implicit and the Tree-Grid methods, for different numbers of nodes.

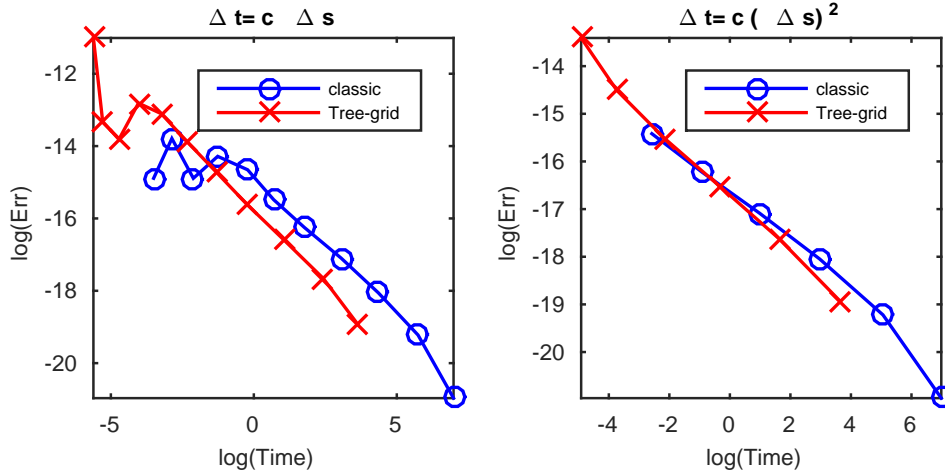| | | | Classic Implicit | | | Tree-Grid | | |
|---|---|---|---|---|---|---|---|---|
| k | $N_t^k$ | $N_s^k$ | Err | EOC | Time | Err | EOC | Time |
| 1 | 51 | 737 | 2.01E-007 | - | 0.0759 | 1.50E-006 | - | 0.0073 |
| 2 | 201 | 1473 | 9.01E-008 | 1.16 | 0.3969 | 5.00E-007 | 1.59 | 0.0239 |
| 3 | 801 | 2945 | 3.81E-008 | 1.24 | 2.6518 | 1.83E-007 | 1.45 | 0.1148 |
| 4 | 3201 | 5889 | 1.45E-008 | 1.39 | 19.4999 | 6.65E-008 | 1.46 | 0.7204 |
| 5 | 12801 | 11777 | 4.50E-009 | 1.69 | 157.2967 | 2.22E-008 | 1.58 | 5.0818 |
| 6 | 51201 | 23553 | 7.83E-010 | 2.52 | 1080.4906 | 6.01E-009 | 1.89 | 38.0828 |



Figure 3: Passport option pricing, "capped" payoff. Comparison of natural logarithm of estimated absolute error of the approximation of solution against natural logarithm of computational time (in seconds) for the classic implicit, and Tree-Grid Method. (Illustration of results from Tables 3, 4)

computational time of the Tree-Grid Method, the Tree-Grid Method was still superior on some grids. Moreover one should note that both examples were done with a reference solution computed with implicit FDM, so the implicit method was favored. This larger error of Tree-Grid can be probably explained by the additional artificial diffusion term needed to stabilize the scheme. The growth of experimental order of convergence with grid refinement results from using solution computed on finest grid as reference solution. For solutions computed on fine grid, the error computed using such reference solution is smaller than in case of using the exact solution as a reference solution. However this difference is not so huge in case of the rough grid solution for which the finest grid solution is a good approximation of exact solution.

## 5. Conclusion

In this paper we introduced a new unconditionally convergent explicit Tree-Grid Method for solving SCP or the related HJB equation with one space and one time dimension. The main idea of this method is to approximate the controlled SDE with an discrete controlled process attaining only values from the grid. However we demand that the moments of the increment of the SDE and of the discrete process are matching. As examined, this condition might be hard to fulfill on general grids if the diffusion in the SDE is vanishing. Therefore we introduced an artificial diffusion term and made the necessary estimations to ensure the consistency with the original problem. Using theory from Barles and Souganidis [3], we have proven unconditional convergence of the method for problems satisfying a set of natural conditions. The method posses many features that are specific to other numerical approaches: Tree methods (especially the trinomial tree method), FSG methods, FDM, Markov chain approximation methods. We discussed these similarities, as well as the most important differences of the method.

One of the most important advantages of the Tree-Grid Method (in contrast to Markov chain approximation methods, explicit FDM or Tree methods) is the freedom by grid construction. Therefore, one may easily construct the grid with respect to terminal condition and change the steps-sizes arbitrarily. The explicitness of the method is a clear advantage over the standardly used implicit FDM for HJB equation. It makes this method faster and also suitable for parallelization. However, unlike explicit FDM or FSG methods, the Tree-Grid Method still remains convergent. We verified these advantages also with numerical simulation of the models from finance.

Taking into account analytical as well as numerical results, it is easy to see that the Tree-Grid Method may be the method of choice for a large range of SCPs with one space dimension. An interesting question is, what is an optimal ratio between time-step and space-step. Another important questions for future research are, how the Tree-Grid Method can be generalized to more space dimensions and how to set boundary conditions in case of more complex problems.

## References

[1] Jaemin Ahn and Minsu Song. Convergence of the trinomial tree method for pricing European/American options. *Applied Mathematics and Computation*, 189(1):575–582, 2007.

[2] Marco Avellaneda, Arnon Levy, and Antonio Parás. Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2(2):73–88, 1995.

[3] Guy Barles and Panagiotis E Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. In *Asymptotic Analysis*, number 4, pages 2347–2349, 1991.

[4] Jérôme Barraquand and Thierry Pudet. Pricing of American path-dependent contingent claims. *Mathematical Finance*, 6(1):17–51, 1996.

[5] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[6] Michael G Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order Partial Differential Equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

[7] P A Forsyth, K R Vetzal, and R Zvan. Convergence of numerical methods for valuing path-dependent options using interpolation. *Review of Derivatives Research*, 5(3):273–314, 2002.

[8] Peter A Forsyth and George Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11(2):1, 2007.

[9] Peter A Forsyth and Kenneth R Vetzal. Numerical methods for nonlinear PDEs in finance. In *Handbook of Computational Finance*, pages 503–528. Springer, 2012.

[10] John C Hull and Alan D White. Efficient procedures for valuing European and American path-dependent options. *The Journal of Derivatives*, 1(1):21–31, 1993.

[11] Soňa Kilianová and Daniel Ševčovič. A transformation method for solving the Hamilton-Jacobi-Bellman equation for a constrained dynamic stochastic optimal allocation problem. *The ANZIAM Journal*, 55(01):14–38, 2013.

[12] Igor Kossaczký, Matthias Ehrhardt, and Michael Günther. Modifications of the PCPT method for HJB equations. In *Application of Mathematics in Technical and Natural sciences: 8th International Conference for Promoting the Application of Mathematics in Technical and Natural Sciences-AMiTaNS'16*, volume 1773, page 030002. AIP Publishing, 2016.

[13] Igor Kossaczký, Matthias Ehrhardt, and Michael Günther. On the non-existence of higher order monotone approximation schemes for HJB equations. *Applied Mathematics Letters*, 52:53–57, 2016.

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

[15] Pierre-Louis Lions. Optimal control of diffusion processes and Hamilton–Jacobi–Bellman equations part 2: viscosity solutions and uniqueness. *Communications in Partial Differential Equations*, 8(11):1229–1276, 1983.

[16] Sure Mataramvura and Bernt Øksendal. Risk minimizing portfolios and HJBI equations for stochastic differential games. *Stochastics An International Journal of Probability and Stochastic Processes*, 80(4):317–337, 2008.

[17] QS Song. Convergence of Markov chain approximation on generalized HJB equation and its applications. *Automatica*, 44(3):761–766, 2008.

[18] Jürgen Topper. A finite element implementation of passport options. Master's thesis, 2003.

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

[20] Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.

[21] Dariusz Zawisza. Smooth solutions to discounted reward control problems with unbounded discount rate and financial applications. *arXiv preprint arXiv:1602.00899*, 2016.