

Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

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

Preprint BUW-IMACM 18/02

Igor Kossaczky, Matthias Ehrhardt and Michael Günther

The Two-dimensional Tree-Grid Method

February 2018

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

The Two-dimensional Tree-Grid Method

Igor Kossaczky^{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 paper we introduce a novel two-dimensional Tree-Grid method for solving stochastic control problems with two space dimensions and one time dimension or equivalently, the corresponding Hamilton-Jacobi-Bellman equation. This new method can be regarded as a generalization of the Tree-Grid method for the stochastic control problems with one space dimension that was recently developed by the authors. This method overcomes two typical shortcomings: it is explicit and unconditionally stable. We prove the convergence of the method and exemplify it in the application on the two-factor uncertain volatility model.

Keywords: 2D Tree-Grid Method, Hamilton-Jacobi-Bellman equation, Stochastic control problem, two-factor uncertain volatility model

1. Introduction

Stochastic control problems (SCP) arise in many application where the stochastic (Itô) process is controlled with respect to time and state in order to optimize the expectation of some utility or cost function. The problem of
5 searching for an optimal control strategy can be often treated by solving the

*Corresponding author

Email addresses: kossaczky@math.uni-wuppertal.de (Igor Kossaczky),
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 Kossaczky),
<http://www-num.math.uni-wuppertal.de/~ehrhardt/> (Matthias Ehrhardt),
<http://www-num.math.uni-wuppertal.de/~guenther/> (Michael Günther)

underlying partial differential equation (PDE), the so-called Hamilton-Jacobi-Bellman (HJB) equation, cf. [1].

1.1. Overview of the numerical methods

10 Either way, solving the SCP directly or solving the HJB equation, numerical methods are often needed as closed form solution are rarely known. For the one-dimensional SCP (with one controlled stochastic process for one state variable) various numerical methods were developed. Based on the approximation of the time-continuous stochastic process with a discrete chain, Markov chain approximation methods are presented in [1]. Implicit finite difference methods (FDM) 15 for solving HJB equations were presented for example in [2]. The advantage of these methods in contrast to the explicit FDMs is their unconditional stability and monotonicity. In [3], a method based on a transformation of the HJB equation was developed. The advantage of using this approach is that we don't need to solve the optimization problem in each time-layer.

20 Recently a new explicit but still unconditionally stable and monotone scheme, the so-called *Tree-Grid method*, was developed by the authors [4]. This method combines the tree structure of the trinomial tree methods for option pricing, with the grid used typically in FDMs and can be regarded as some special explicit FDM or a Markov chain approximation. A useful modification of this method 25 was presented in [5]. As the explicitness and great flexibility of the method (unconditional stability with any grid spacing) exhibit clear advantages of this method, we are interested here in generalizing this method to two state-space dimensions (two controlled stochastic processes).

In two space dimensions, a generalization of the implicit unconditionally 30 stable method of Wang and Forsyth [6] was presented in [7] and later used in [8]. The main idea of this method is combining the wide and the fixed stencil depending on the correlation in the particular time-space node. Alternatively, explicit methods based on the ideas of Kushner and Dupuis [1] presented in the papers [9, 10, 11] can be used. These are wide stencil schemes stable under some 35 CFL condition. Moreover, in these methods a linear interpolation of the grid

values is needed. Our generalization of the Tree-Grid method presented in this paper also falls into this class of wide stencil schemes, however it overcomes two shortcomings: it is unconditionally stable for any grid and no interpolation of the grid values is needed. Furthermore, as this two-dimensional (2D) Tree-Grid
40 method is explicit, it is also suitable for parallelization.

1.2. Problem formulation

We are concerned with searching for the value function $V(x, y, t)$ of the following *2-dimensional general stochastic control problem* (SCP):

$$V(x, y, t) = \max_{\theta(x, y, t) \in \bar{\Theta}} \mathbb{E} \left(\int_t^T \exp \left(\int_t^k r(*_l) dl \right) f(*_k) dk + \exp \left(\int_t^T r(*_k) dk \right) V_T(X_T, Y_T) \Big| X_t = x, Y_t = y \right), \quad (1)$$

$$dX_t = \mu_x(*_t) dt + \sigma_x(*_t) dW_t^x, \quad (2)$$

$$dY_t = \mu_y(*_t) dt + \sigma_y(*_t) dW_t^y, \quad (3)$$

$$dW_t^x dW_t^y = \sigma_{xy}(*_t) / (\sigma_x(*_t) \sigma_y(*_t)) \quad (4)$$

$$\text{with } *_t = (X_t, Y_t, t, \theta(X_t, Y_t, t)), \quad 0 < t < T, \quad x, y \in \mathbb{R},$$

where x, y are the state variables and t is time. Here, $\bar{\Theta}$ denotes the space of all *suitable* control functions (see e.g. [1, 12]) from $\mathbb{R}^2 \times [0, T]$ to a set Θ . For our purpose, we will assume Θ to be discrete. If this is not the case, we can easily achieve this property by its discretization. Now following the Bellman's principle, the *dynamic programming equation* holds:

$$V(x, y, t_j) = \max_{\theta(x, y, t) \in \bar{\Theta}_{t_j}} \mathbb{E} \left(\int_{t_j}^{t_{j+1}} \exp \left(\int_{t_j}^k r(*_l) dl \right) f(*_k) dk + \exp \left(\int_{t_j}^{t_{j+1}} r(*_k) dk \right) V(X_{t_{j+1}}, Y_{t_{j+1}}, t_{j+1}) \Big| X_{t_j} = x, Y_{t_j} = y \right), \quad (5)$$

where $0 \leq t_j < t_{j+1} \leq T$ are some time-points and $\bar{\Theta}_{t_j}$ is a set of control functions from $\bar{\Theta}$ restricted to the $\mathbb{R}^2 \times [t_j, t_{j+1})$ domain. Using the dynamic programming equation (5), it can be shown [12], that solving the SCP (1)–(4)

is equivalent to solving the two-dimensional Hamilton-Jacobi-Bellman (HJB) equation:

$$\frac{\partial V}{\partial t} + \max_{\theta \in \Theta} (LV + r(\cdot)V + f(\cdot)) = 0, \quad (6)$$

$$LV = \frac{\sigma_x(\cdot)^2}{2} \frac{\partial^2 V}{\partial x^2} + \sigma_{xy}(\cdot) \frac{\partial^2 V}{\partial x \partial y} + \frac{\sigma_y(\cdot)^2}{2} \frac{\partial^2 V}{\partial y^2} + \mu_x(\cdot) \frac{\partial V}{\partial x} + \mu_y(\cdot) \frac{\partial V}{\partial y} \quad (7)$$

$$V(x, y, T) = V_T(x, y), \quad (8)$$

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

where $\sigma(\cdot)$, $\mu(\cdot)$, $r(\cdot)$, $f(\cdot)$ are functions of x, y, t, θ . We should note that the maximum operator in (1) and (6) can be replaced by a minimum operator and the whole following analysis will hold analogously. A possible generalization
45 of the SCP with corresponding HJB equation is a stochastic differential game with the corresponding Hamilton-Jacobi-Bellman-Isaac equation [13]. Further generalizations can be found in [14].

2. Construction of the 2D Tree-Grid algorithm

In this section we will derive the two-dimensional Tree-Grid algorithm for
50 solving the problem (1)–(4). We will use the ideas that were widely explained in [4], [5] and we refer interested readers to these papers. We will work on a three-dimensional rectangular domain with two space dimensions and one time dimension. For a fixed control θ , the candidate for a value in each node $V(x_i, y_j, t_k)$ will be computed from seven values from the next time layer t_{k+1} .
55 Figure 1 illustrates this approach. We will denote these seven values in this context simply as stencil located at (x_i, y_j, t_k) . The weights of these seven values can be interpreted as probabilities and therefore we demand, that the moments of such discrete random variable are matching with the moments of the increment of the two-dimensional stochastic process (2)–(4) with the fixed
60 control θ at least asymptotically. Then, the actual value $V(x_i, y_j, t_k)$ will be computed as a maximum of the candidates. For handling nodes close to the boundary we suppose that we know, how the solution behaves in the outer

neighbourhood of the boundaries, and that we can describe this behaviour with a boundary function $BC(x, y, t)$. The terminal condition in the time-layer $t_M = T$ reads $V(x, y, t_M) = V_T(x, y)$.

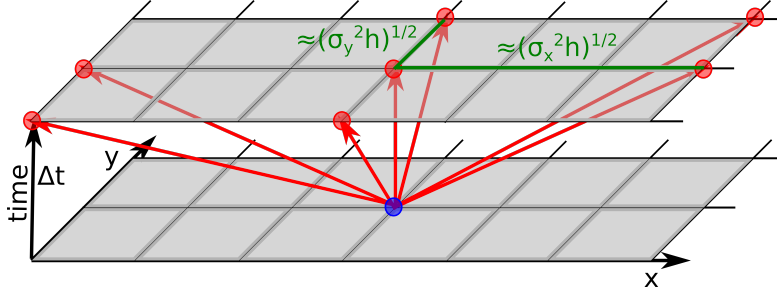


Figure 1: Illustration of the two-dimensional Tree-Grid structure. Only the red nodes in the later time layer impact the blue node. On the other hand, the red nodes can be interpreted as possible future states if we are in the blue node state. The figure illustrates a situation with positive correlation and a variance that is larger in the x -direction than in the y -direction. The stencil size in each direction is roughly proportional to the square root of the variance coefficient in that direction multiplied by the discretization parameter h .

2.1. Notation

At first, before discussing how to choose the stencil nodes around node (x_i, y_j, t_k) , and the proper weights, we present here the notation used in the sequel:

- x_1, x_2, \dots, x_{N_x} -space discretization in the first space dimension.
- y_1, y_2, \dots, y_{N_y} -space discretization in the second space dimension.
- t_1, t_2, \dots, t_M -time discretization .
- $\Delta_k t = t_{k+1} - t_k$ -(current) time-step. We will use equidistant time stepping, $\Delta_k t = \Delta t$ for all $k = 1, 2, \dots, M - 1$. A generalization to non-equidistant time stepping is straightforward.
- $\Delta_i x = x_{i+1} - x_i$, $\Delta_j y = y_{j+1} - y_j$ space-steps in the first and second dimension (possibly non-equidistant).
- $\Delta x = \max_i \Delta_i x$, $\Delta y = \max_i \Delta_i y$.

- $h = \max(K \max(\Delta x, \Delta y), \Delta t)$, where $K > 0$ is a parameter used for regulating the stencil size. In the non-equidistant case, Δt should be replaced by $\Delta_k t$.
- $b = h/\Delta t$. In the non-equidistant case, this should be replaced by $b = h/\Delta_k t$ in the following algorithm.
- (x_i, y_j, t_k) -the node for which the stencil is constructed.
- $E_* = \mu_*(x_i, y_j, t_k, \theta)\Delta t$ for $* = x, y$.
- $Var_* = (\sigma_*^2(x_i, y_j, t_k, \theta) + \mathcal{O}(h))\Delta t$ for $* = x, y$, will be determined later.
- $\rho(x_i, y_j, t_k, \theta) = \sigma_{xy}(x_i, y_j, t_k, \theta)/(\sigma_x(x_i, y_j, t_k, \theta)\sigma_y(x_i, y_j, t_k, \theta))$
- $\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) = \sigma_{xy}(x_i, y_j, t_k, \theta) + \mathcal{O}(\sqrt{h})$, will be determined later.
- $Cov = (\sigma_{xy}(x_i, y_j, t_k, \theta) + \mathcal{O}(\sqrt{h}))\Delta t$, will be determined later.
- $W_* = Var_* + E_*^2$ for $* = x, y$.
- $W_{xy} = Cov + E_x E_y$.
- $v_{i,j}^k$ -numerical approximation of $V(x_i, y_j, t_k)$

2.2. Choosing the stencil nodes

Next, we describe how to choose the stencil nodes around an arbitrary node (x_i, y_j, t_k) for a fixed control θ . The values in these nodes will impact the value in the node (x_i, y_j, t_k) .

If $E_x > 0$,

$$x_- = \left\lfloor x_i - \sqrt{2W_x b} \right\rfloor_x, \quad x_+ = \left\lceil \max \left(x_i + \sqrt{2W_x b}, x_i + (x_i - x_-) \right) \right\rceil_x, \quad (9)$$

else if $E_x < 0$,

$$x_+ = \left\lfloor x_i + \sqrt{2W_x b} \right\rfloor_x, \quad x_- = \left\lceil \min \left(x_i - \sqrt{2W_x b}, x_i - (x_+ - x_i) \right) \right\rceil_x, \quad (10)$$

else ($E_x = 0$),

$$x_+ = \left\lfloor x_i + \sqrt{2W_x b} \right\rfloor_x, \quad x_- = \left\lfloor x_i - \sqrt{2W_x b} \right\rfloor_x, \quad (11)$$

where $\lceil \cdot \rceil_x$ resp. $\lfloor \cdot \rfloor_x$ denotes rounding to the nearest greater resp. smaller element from x_1, x_2, \dots, x_{N_x} . If such element does not exist, $\lceil z \rceil_x$ resp. $\lfloor z \rfloor_x$ will return just z .

If $E_y > 0$,

$$y_- = \left\lfloor y_j - \sqrt{2W_y b} \right\rfloor_y, \quad y_+ = \left\lceil \max \left(y_j + \sqrt{2W_y b}, y_j + (y_j - y_-) \right) \right\rceil_y, \quad (12)$$

else if $E_y < 0$,

$$y_+ = \left\lceil y_j + \sqrt{2W_y b} \right\rceil_y, \quad y_- = \left\lfloor \min \left(y_j - \sqrt{2W_y b}, y_j - (y_+ - y_j) \right) \right\rfloor_y, \quad (13)$$

else ($E_y = 0$),

$$y_+ = \left\lceil y_j + \sqrt{2W_y b} \right\rceil_y, \quad y_- = \left\lfloor y_j - \sqrt{2W_y b} \right\rfloor_y, \quad (14)$$

100 where $\lceil \cdot \rceil_y, \lfloor \cdot \rfloor_y$ are defined analogously.

The following nodes with the respective weights (probabilities) will be used in the stencil located at (x_i, y_j, t_k) :

- node (x_i, y_j, t_{k+1}) with the probability p_o ,
- nodes (x_+, y_j, t_{k+1}) and (x_-, y_j, t_{k+1}) with the probabilities p_{x+} and p_{x-} ,
- 105 • nodes (x_i, y_+, t_{k+1}) and (x_i, y_-, t_{k+1}) with the probabilities p_{y+} and p_{y-} ,
- nodes (x_+, y_+, t_{k+1}) and (x_-, y_-, t_{k+1}) , both with the probability p_{xy} if W_{xy} is non-negative, and nodes (x_+, y_-, t_{k+1}) and (x_-, y_+, t_{k+1}) both with the probability p_{xy} if W_{xy} is negative. In the following algorithm we will focus on the case of non-negative W_{xy} , the case of a negative W_{xy} is
- 110 treated analogously.

Moreover, we define the difference operators

$$\Delta_+ x = x_+ - x_i, \quad \Delta_- x = x_i - x_-, \quad (15)$$

$$\Delta_+ y = y_+ - y_i, \quad \Delta_- y = y_i - y_-. \quad (16)$$

2.3. Choosing the stencil weights (probabilities)

To match the first two moments of the approximative increment of the stochastic process and of the increment of the discrete process defined by the “stencil nodes” and their probabilities and to ensure that the probabilities are positive and sum up to 1, we demand:

$$p_o + p_{x+} + p_{x-} + p_{y+} + p_{y-} + 2p_{xy} = 1, \quad (17)$$

$$(p_{x+} + p_{xy})\Delta_+x - (p_{x-} + p_{xy})\Delta_-x = E_x, \quad (18)$$

$$(p_{y+} + p_{xy})\Delta_+y - (p_{y-} + p_{xy})\Delta_-y = E_y, \quad (19)$$

$$(p_{x+} + p_{xy})(\Delta_+x)^2 + (p_{x-} + p_{xy})(\Delta_-x)^2 = W_x, \quad (20)$$

$$(p_{y+} + p_{xy})(\Delta_+y)^2 + (p_{y-} + p_{xy})(\Delta_-y)^2 = W_y, \quad (21)$$

$$p_{xy}(\Delta_+x\Delta_+y + \Delta_-x\Delta_-y) = W_{xy}. \quad (22)$$

$$p_o, p_{x+}, p_{x-}, p_{y+}, p_{y-}, p_{xy} \geq 0 \quad (23)$$

For negative W_{xy} , only the condition (22) changes to

$$p_{xy}(\Delta_+x\Delta_-y + \Delta_-x\Delta_+y) = |W_{xy}|. \quad (24)$$

The solution of the six equations (17)–(22) reads

$$p_o = \frac{\Delta_+x\Delta_-x\Delta_+y\Delta_-y - \Delta_+x\Delta_-xW_y - \Delta_+y\Delta_-yW_x}{\Delta_+x\Delta_-x\Delta_+y\Delta_-y} + \frac{E_y(\Delta_+y - \Delta_-y)}{\Delta_+y\Delta_-y} + \frac{E_x(\Delta_+x - \Delta_-x)}{\Delta_+x\Delta_-x} + \frac{2|W_{xy}|}{\Delta_c}, \quad (25)$$

$$p_{x+} = \frac{W_x + E_x\Delta_-x}{(\Delta_+x)^2 + \Delta_+x\Delta_-x} - \frac{|W_{xy}|}{\Delta_c}, \quad (26)$$

$$p_{x-} = \frac{W_x - E_x\Delta_+x}{(\Delta_-x)^2 + \Delta_+x\Delta_-x} - \frac{|W_{xy}|}{\Delta_c}, \quad (27)$$

$$p_{y+} = \frac{W_y + E_y\Delta_-y}{(\Delta_+y)^2 + \Delta_+y\Delta_-y} - \frac{|W_{xy}|}{\Delta_c}, \quad (28)$$

$$p_{y-} = \frac{W_y - E_y\Delta_+y}{(\Delta_-y)^2 + \Delta_+y\Delta_-y} - \frac{|W_{xy}|}{\Delta_c}, \quad (29)$$

$$p_{xy} = \frac{|W_{xy}|}{\Delta_c}, \quad (30)$$

where $\Delta_c = \Delta_+x\Delta_+y + \Delta_-x\Delta_-y$ for non-negative $(\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta)\Delta t + E_xE_y)$ and $\Delta_c = \Delta_+x\Delta_-y + \Delta_-x\Delta_+y$ for negative $(\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta)\Delta t + E_xE_y)$.

Following the construction of x_+, x_-, y_+, y_- , it is easy to check that p_o is
 115 always non-negative. The same holds for p_{xy} . To ensure also the non-negativity
 of p_{x+}, p_{x-}, p_{y+} and p_{y-} we have to properly choose the variables $Var_x, Var_y,$
 and Cov to get non-negative weights, while still remaining consistent with the
 original problem as $h \rightarrow 0$. This is done in the next Section 2.4.

2.4. Artificial diffusion and covariance adjustment

Let us assume $E_x \geq 0$. Now, the first fraction of p_{x+} is positive, however
 the first fraction of p_{x-} may be negative. We will set Var_x (and hence W_x) in
 such way, that it will be also positive. It holds

$$W_x - E_x \Delta_+ x > Var_x + E_x^2 - E_x(\sqrt{2b(Var_x + E_x^2)} + 2\Delta x). \quad (31)$$

The right-hand side of the inequality (31) is 0 for $Var_x = A_x$ and greater than
 0 for $Var_x > A_x$ with

$$A_x = 1/2 \left(|E_x| \sqrt{4b^2 E_x^2 + 16b \Delta x |E_x|} - (2b - 2)E_x^2 + 4\Delta x |E_x| \right). \quad (32)$$

We replaced here E_x with $|E_x|$ to cover also the analogous case $E_x < 0$ and
 possibly negative p_{x+} . Now, if we set

$$Var_x = \max(\sigma_x^2(x_i, y_j, t_k, \theta) \Delta t, A_x, E_x^2), \quad (33)$$

$$Var_y = \max(\sigma_y^2(x_i, y_j, t_k, \theta) \Delta t, A_y, E_y^2), \quad (34)$$

where A_y is defined analogously, the first fraction in p_{x+}, p_{x-}, p_{y+} and p_{y-}
 will be non-negative. The possible difference between Var_x resp. Var_y and
 the variances from the original problem $\sigma_x^2 \Delta t$ resp. $\sigma_y^2 \Delta t$ is called *artificial*
diffusion. Now, taking into account the correlation coefficient in the current
 node, following these definitions of variances Var_x, Var_y we introduce also the
 new covariance

$$\tilde{\sigma}_{xy} = \rho \frac{\sqrt{Var_x Var_y}}{\Delta t} \quad (35)$$

and

$$C_{xy} := \min \left(\frac{W_x + E_x \Delta_- x}{(\Delta_+ x)^2 + \Delta_+ x \Delta_- x} \Delta_c, \frac{W_x - E_x \Delta_+ x}{(\Delta_- x)^2 + \Delta_+ x \Delta_- x} \Delta_c, \right. \\ \left. \frac{W_y + E_y \Delta_- y}{(\Delta_+ y)^2 + \Delta_+ y \Delta_- y} \Delta_c, \frac{W_y - E_y \Delta_+ y}{(\Delta_- y)^2 + \Delta_+ y \Delta_- y} \Delta_c, \right. \\ \left. |\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t + E_x E_y| \right). \quad (36)$$

Now, we define Cov as

$$Cov = \begin{cases} C_{xy} - E_x E_y & \text{if } (\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t + E_x E_y) \geq 0, \\ -C_{xy} - E_x E_y & \text{if } (\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t + E_x E_y) < 0. \end{cases} \quad (37)$$

This covariance Cov is consistent with the variances Var_x , Var_y defined by (33), (34) as for $\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t + E_x E_y \geq 0$ it holds

$$-\sqrt{Var_x Var_y} \leq -E_x E_y \leq Cov \leq \tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t \leq \sqrt{Var_x Var_y}, \quad (38)$$

and for $\tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t + E_x E_y < 0$ it holds

$$-\sqrt{Var_x Var_y} \leq \tilde{\sigma}_{xy}(x_i, y_j, t_k, \theta) \Delta t \leq Cov \leq -E_x E_y \leq \sqrt{Var_x Var_y}. \quad (39)$$

Now it also holds $|W_{xy}| = C_{xy}$, which implies that p_{x+} , p_{x-} , p_{y+} and p_{y-} are all positive. It is easy to check that it holds

$$\frac{Var_x}{\Delta t} = \sigma_x^2(x_i, y_j, t_k, \theta) + \mathcal{O}(h), \quad \frac{Var_y}{\Delta t} = \sigma_y^2(x_i, y_j, t_k, \theta) + \mathcal{O}(h), \quad (40)$$

and

$$\frac{Var_x}{\Delta t} = \sigma_x^2(x_i, y_j, t_k, \theta), \quad \text{resp.} \quad \frac{Var_y}{\Delta t} = \sigma_y^2(x_i, y_j, t_k, \theta), \quad (41)$$

for $\sigma_x^2(x_i, y_j, t_k, \theta) \neq 0$ resp. $\sigma_y^2(x_i, y_j, t_k, \theta) \neq 0$ and h small enough. It holds

$$W_x + E_x \Delta_- x = \left(\sigma_x^2 + \mathcal{O}(\sqrt{h}) \right) \Delta t, \quad (42)$$

$$(\Delta_+ x)^2 + \Delta_+ x \Delta_- x = 4h\sigma_x^2 + \mathcal{O}(h^{3/2}), \quad (43)$$

$$\Delta_c = 4h\sigma_x\sigma_y + \mathcal{O}(h^{3/2}). \quad (44)$$

It follows that

$$\frac{W_x + E_x \Delta_- x}{(\Delta_+ x)^2 + \Delta_+ x \Delta_- x} \Delta_c = \left(\sigma_x\sigma_y + \mathcal{O}(\sqrt{h}) \right) \Delta t \quad (45)$$

and the same holds also for the second, third and fourth maximum candidate in (36). Moreover,

$$|\tilde{\sigma}_{xy}\Delta t + E_x E_y| = |\tilde{\sigma}_{xy}|\Delta t + \mathcal{O}((\Delta t)^2) = \left(|\sigma_{xy}| + \mathcal{O}(\sqrt{h})\right) \Delta t. \quad (46)$$

Therefore,

$$C_{xy} = \left(|\sigma_{xy}| + \mathcal{O}(\sqrt{h})\right) \Delta t, \quad \Rightarrow \quad \frac{Cov}{\Delta t} = \sigma_{xy} + \mathcal{O}(\sqrt{h}) \quad (47)$$

and it is easy to check that

$$\frac{Cov}{\Delta t} = \sigma_{xy} \quad (48)$$

120 for $\sigma_x^2 \neq 0$, $\sigma_y^2 \neq 0$, $|\sigma_{xy}| \neq \sigma_x \sigma_y$ and h small enough. Following (40),(47), the modified variances and the modified covariance are consistent with the variances and the covariance from the original problem. Moreover, for $\sigma_x, \sigma_y \neq 0$ and $|\sigma_{xy}| < \sigma_x \sigma_y$, the modified variances and the covariance will be equal to the original ones for h small enough.

125 2.5. Setting the parameter K and the stencil size reduction

In the formula for $h = \max(K \max(\Delta x, \Delta y), \Delta t)$, the part $K \max(\Delta x, \Delta y)$ is responsible for the consistency of the correlation in the numerical model with the correlation of the original problem. Here, the parameter $K > 0$ can be chosen arbitrarily, however for a large (relative to the problem parameters) K ,
 130 the stencil is large, what typically increases the error. On the other hand, for too small K , the correlation may start being exact (or exact enough) only for very fine grids, while not being sufficiently exact on the coarse grids, resulting in larger errors on these coarse grids. For $K = 0$ we can't guarantee the convergence of the correlation. However, the correlation in the numerical model
 135 may match with the correlation from the original problem even for small K or $K = 0$. This motivates the following *multiple K modification* of the Tree-Grid method;

For each control in each node:

1. set $l = 1$ and use $K = K_0$, $K_0 \geq 0$ to compute x_* , y_* , p_*

- 140 2. compute the correlation of the numerical model: $\tilde{\rho} = Cov/\sqrt{Var_x Var_y}$
3. **if** $\tilde{\rho} \neq \rho$: recompute x_*, y_*, p_* with $K = K_l$, $K_l > K_{l-1}$ and set $l := l + 1$
 else break.
4. **if** $l < l_{max}$: go to step 2,
 else break.

145 Using this modification, we will use a smaller stencil size as much as possible. This approach can be seen as some analogy to the approach of Ma and Forsyth [7], where a fixed (and thus small) stencil is used as much as possible. However, here we will not increase the convergence rate, but possibly reduce the error.

Another approach, *the non-constant K modification*, is to use a non-constant
150 $K = K(x, y, \theta)$. This K can be smaller in nodes with large volatilities σ_x, σ_y and larger in nodes with smaller σ_x, σ_y , regulating the stencil size to not explode in case of large volatilities.

Both modifications can be also combined and it is easy to check, that they do not harm the consistency. In our numerical simulation these modifications
155 didn't lead to a significant improvement and therefore we used simply a constant $K = 1/400$. However, a non-constant K modification may be useful for other stochastic control problems.

3. The 2D Tree-Grid Algorithm

Finally, we can use the stencil nodes and weights to construct the 2D Tree-
160 Grid algorithm.

We define the function v^{k+1} :

$$\begin{aligned} &\text{If } (x, y) \in \{x_1, x_2, \dots, x_{N_x}\} \times \{y_1, y_2, \dots, y_{N_y}\} : \\ &\quad v^{k+1}(x, y) = v_{i,j}^{k+1} \text{ so that } (x, y) = (x_i, y_j), \quad (49) \\ &\text{else: } v^{k+1}(x, y) = BC(x, y, t_{k+1}). \end{aligned}$$

For a given space-node (x_i, y_j) and a given control θ we define

$$v_o^{k+1} = v^{k+1}(x_i, y_j), \quad v_{x+}^{k+1} = v^{k+1}(x_+, y_j), \quad v_{x-}^{k+1} = v^{k+1}(x_-, y_j), \quad (50)$$

$$v_{y+}^{k+1} = v^{k+1}(x_i, y_+), \quad v_{y-}^{k+1} = v^{k+1}(x_i, y_-), \quad (51)$$

$$\text{If } W_{xy} \geq 0: \quad v_{xy+}^{k+1} = v^{k+1}(x_+, y_+), \quad v_{xy-}^{k+1} = v^{k+1}(x_-, y_-). \quad (52)$$

$$\text{If } W_{xy} < 0: \quad v_{xy+}^{k+1} = v^{k+1}(x_+, y_-), \quad v_{xy-}^{k+1} = v^{k+1}(x_-, y_+). \quad (53)$$

$$f_{i,j}^k(\theta) = f(x_i, y_j, t_k, \theta), \quad r_{i,j}^k(\theta) = r(x_i, y_j, t_k, \theta). \quad (54)$$

Now, the discretized version of the dynamic programming equation for

$i = 2, 3, \dots, N_x - 1, j = 2, 3, \dots, N_y - 1, k = 1, 2, \dots, M - 1$ reads

$$v_{i,j}^k = \max_{\theta \in \Theta} w_{i,j}^k(\theta) \quad (55)$$

$$\begin{aligned} w_{i,j}^k(\theta) = & f_{i,j}^k(\theta)\Delta t + (1 + r_{i,j}^k(\theta)\Delta t) \\ & \cdot \left(p_{x+}v_{x+}^{k+1} + p_{x-}v_{x-}^{k+1} + p_{y+}v_{y+}^{k+1} + p_{y-}v_{y-}^{k+1} \right. \\ & \left. + p_o v_o^{k+1} + p_{xy}(v_{xy+}^{k+1} + v_{xy-}^{k+1}) \right). \end{aligned} \quad (56)$$

For the boundary nodes (x_i, y_j) ($i \in \{1, N_x\}$ or $j \in \{1, N_y\}$) we employ the boundary condition:

$$v_{i,j}^k = BC(x_i, y_j, t_k). \quad (57)$$

The terminal condition is defined as

$$v_{i,j}^M = V_T(x_i, y_j). \quad (58)$$

Finally we can summarize the whole algorithm of the 2D Tree-Grid Method:

Algorithm 1 The 2D Tree-Grid method

```
1: Set  $v_{i,j}^M = V_T(x_i, y_j)$  for  $i = 1, 2, \dots, N_x, j = 1, 2, \dots, N_y$ ;  
2: for  $k = M - 1, M - 2, \dots, 1$  do  
3:   for  $i = 1, 2, 3, \dots, N_x$  do  
4:     for  $j = 1, 2, 3, \dots, N_y$  do  
5:       if  $i \in \{1, N_x\}$  or  $j \in \{1, N_y\}$  then  
6:         Compute  $v_{i,j}^k$  according to (57);  
7:       else  
8:         for  $\theta \in \Theta$  do  
9:           Determine  $E_x, E_y$  according to Section 2.1;  
10:          Compute  $A_x, A_y$  according to (32);  
11:          Compute  $Var_x, Var_y$  according to (33), (34);  
12:          Determine  $W_x, W_y$  according to Section 2.1;  
13:          Determine  $x_+, x_-, y_+, y_-$  according to (9)–(14);  
14:          Determine  $\Delta_+x, \Delta_-x, \Delta_+y, \Delta_-y$  according to (15)–(16);  
15:          Compute  $\tilde{\sigma}_{xy}$  according to (35);  
16:          Compute  $C_{xy}$  according to (36);  
17:          Compute  $Cov$  according to (37);  
18:          Determine  $W_{xy}$  according to Section 2.1;  
19:          Compute  $p_o, p_{x+}, p_{x-}, p_{y+}, p_{y-}, p_{xy}$  according to (25)–(30);  
20:          Using (49), compute all variables defined by (50)–(54);  
21:          Compute  $w_{i,j}^k(\theta)$  according to (56);  
22:        end for  
23:        Compute  $v_{i,j}^k$  according to (55);  
24:      end if  
25:    end for  
26:  end for  
27: end for
```

4. Convergence

In this section we will prove the convergence of the 2D Tree-Grid method. First we will quickly summarize the classical convergence theory of Barles and Souganidis [15] and in the second part of this section we will use this theory to prove the convergence of our scheme. Let us note that the algorithm was derived by discretizing the dynamics in the original SCP (1)–(4), but we will prove that the scheme is consistent with the HJB equation (6)–(8).

4.1. The convergence theory

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

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

We suppose there exists a *viscosity solution* (see [16]) of the equation $FV(s) = 0$, and denote this solution simply by $V(s)$. To find some approximation of the viscosity solution we define a *discrete approximation scheme*

$$Gv(s) = G(v(s), v(s + b_1h), v(s + b_2h), \dots, v(s + b_nh)), \quad (59)$$

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

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

$$S_h = \{s_i \in \mathbb{R}^K \mid i = 1, 2, \dots, N_h\},$$

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

Definition 1 (Numerical scheme). *The system of equations $Gv(s) = 0$ with $s \in S_h$ depending on a parameter $h \in \mathbb{R}^+$ is called a numerical scheme.*

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

180 **Definition 2** (Monotonicity). *A discrete approximation scheme $Gv(s) = G(v(s), v(s + b_1h), v(s + b_2h), \dots, v(s + b_nh))$ is monotone, if the function G is non-increasing in $v(s + b_ih)$ for $b_i \neq 0, i = 1, \dots, n$.*

Definition 3 (Consistency). *The scheme $Gv(s) = G(v(s), v(s + b_1h), v(s + b_2h), \dots, v(s + b_nh))$ is a consistent approximation of $FV(s)$, if $\lim_{h \rightarrow 0} |F\phi(s) - G\phi(s)| = 0$, for any C^∞ -smooth test function $\phi(s)$.*

185

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 [17], this leads to an equivalent definition.

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

The following Theorem of Barles and Souganidis [15] is the key for proving convergence of a numerical scheme approximating a nonlinear PDE:

195 **Theorem 1** (Barles-Souganidis). *If the equation $FV(s) = 0$ satisfies the strong uniqueness property (see [15]) and if the numerical scheme $Gv_h(s) = 0, s \in S_h$ approximating the equation $FV(s) = 0$ is monotone, consistent and stable, its solution $v_h(s)$ converges locally uniformly to the solution $V(s)$ of $FV(s) = 0$ with $h \rightarrow 0$.*

200 The above mentioned strong uniqueness property [15] is a property of the problem and not of the numerical scheme. Therefore, we will simply assume that our problem possess this property without actually proving it.

4.2. Convergence of the 2D Tree-Grid method

In this section, we will prove the convergence of the 2D Tree-Grid method. For the purpose of this convergence analysis we will rewrite equations (55), (56)

as

$$\begin{aligned}
Gv(x_i, y_j, t_k) &= G(v_o^{k+1}, v_{x+}^{k+1}, v_{x-}^{k+1}, v_{y+}^{k+1}, v_{y-}^{k+1}, v_{xy+}^{k+1}, v_{xy-}^{k+1}) \\
&= \frac{1}{\Delta t} \left(v_{i,j}^k - \max_{\theta \in \Theta} \left(f_{i,j}^k(\theta) \Delta t + (1 + r_{i,j}^k(\theta) \Delta t) \right. \right. \\
&\quad \cdot \left(p_{x+} v_{x+}^{k+1} + p_{x-} v_{x-}^{k+1} + p_{y+} v_{y+}^{k+1} + p_{y-} v_{y-}^{k+1} \right. \\
&\quad \left. \left. \left. + p_o v_o^{k+1} + p_{xy} (v_{xy+}^{k+1} + v_{xy-}^{k+1}) \right) \right) \right) = 0. \tag{60}
\end{aligned}$$

Using the theory of the previous Section 4.1, our goal is to show that equation (60) is a monotone, consistent, and stable approximation of the nonlinear differential operator FV defined by the PDE (6):

$$FV(x, y, t) = -\frac{\partial V}{\partial t} - \max_{\theta \in \Theta} \left(LV + r(\cdot)V + f(\cdot) \right). \tag{61}$$

Let us define the difference operators

$$\Delta_{o+} z = \Delta_+ z, \quad \Delta_{o-} z = -\Delta_- z, \tag{62}$$

for $z \in \{x, y\}$. At first, we will show the consistency of the scheme in an arbitrary point (x_i, y_j, t_k) :

Lemma 1 (Consistency). *The discrete scheme (60) is consistent with the PDE operator (61).*

Proof. Let $\phi : \mathbb{R}^2 \times [0, T] \rightarrow \mathbb{R}$ be a C^∞ -smooth function. Let us define $\phi^k(x, y) = \phi(x, y, t_k)$ and use the short notation defined by (50)–(54) for ϕ^{k+1} instead of v^{k+1} . At first, let us sketch the main idea of the proof of consistency in an arbitrary point (x_i, y_j, t_k) : We will write all values ϕ_α^{k+1} for $\alpha \in \{x+, x-, y+, y-, xy+, xy-\}$ as Taylor expansions around $\phi_{i,j}^k$. We will substitute these Taylor expansions into the discrete scheme (60), group terms with the same derivatives together and estimate the sum of the coefficients in front of them. We will end up with the PDE operator (61) and some terms of order $\mathcal{O}(h^\lambda)$, where $\lambda = 1/2$ if $|\rho| = 1$ or $\sigma_x = 0$ or $\sigma_y = 0$ and $\lambda = 1$ else. Let us suppose that $W_{xy} \geq 0$ in (x_i, y_j, t_k) . The case of negative W_{xy} can be

treated analogously. For $* \in \{+, -\}$, let us define the operators:

$$\begin{aligned} A_{xy*}\phi &= \frac{\partial\phi}{\partial x}\Delta_{o*x} + \frac{\partial\phi}{\partial y}\Delta_{o*y} + \frac{1}{2}\frac{\partial^2\phi}{\partial x^2}(\Delta_{o*x})^2 + \frac{\partial^2\phi}{\partial x\partial y}\Delta_{o*x}\Delta_{o*y} \\ &\quad + \frac{1}{2}\frac{\partial^2\phi}{\partial y^2}(\Delta_{o*y})^2, \end{aligned} \quad (63)$$

$$\begin{aligned} B_{xy*}\phi &= \frac{1}{6}\frac{\partial^3\phi}{\partial x^3}(\Delta_{o*x})^3 + \frac{1}{2}\frac{\partial^3\phi}{\partial x^2\partial y}(\Delta_{o*x})^2\Delta_{o*y} + \frac{1}{2}\frac{\partial^3\phi}{\partial x\partial y^2}\Delta_{o*x}(\Delta_{o*y})^2 \\ &\quad + \frac{1}{6}\frac{\partial^3\phi}{\partial y^3}(\Delta_{o*y})^3, \end{aligned} \quad (64)$$

$$\begin{aligned} C_{xy*}\phi &= \frac{1}{24}\frac{\partial^4\phi}{\partial x^4}(\Delta_{o*x})^4 + \frac{1}{6}\frac{\partial^4\phi}{\partial x^3\partial y}(\Delta_{o*x})^3\Delta_{o*y} + \frac{1}{4}\frac{\partial^4\phi}{\partial x^2\partial y^2}(\Delta_{o*x})^2(\Delta_{o*y})^2 \\ &\quad + \frac{1}{6}\frac{\partial^4\phi}{\partial x\partial y^3}\Delta_{o*x}(\Delta_{o*y})^3 + \frac{1}{24}\frac{\partial^4\phi}{\partial y^4}(\Delta_{o*y})^4. \end{aligned} \quad (65)$$

Now we can expand ϕ_{xy*}^{k+1} around $\phi_{i,j}^k$ as follows:

$$\begin{aligned} \phi_{xy*}^{k+1} &= \phi_{i,j}^k + \frac{\partial}{\partial t}\phi_{i,j}^k\Delta t + A_{xy*}\phi_{i,j}^k + \frac{\partial}{\partial t}(A_{xy*}\phi_{i,j}^k)\Delta t + B_{xy*}\phi_{i,j}^k \\ &\quad + \frac{\partial}{\partial t}(B_{xy*}R_{xy*})\Delta t + C_{xy*}R_{xy*} + \mathcal{O}((\Delta t)^2), \end{aligned} \quad (66)$$

where

$$R_{xy*} = \phi(x_i + \epsilon_{xy*}\Delta_{o*x}, y_j + \delta_{xy*}\Delta_{o*y}, t_k + \gamma_{xy*}\Delta t), \quad (67)$$

for some $\epsilon_{xy*}, \delta_{xy*}, \gamma_{xy*} \in [0, 1]$. We expand $\phi_{x*}^{k+1}, \phi_{y*}^{k+1}$, $* \in \{+, -\}$ in the same manner: for ϕ_{x*}^{k+1} we only need to substitute Δ_{o*y} with 0 and for ϕ_{y*}^{k+1} we only need to substitute Δ_{o*x} with 0 in the Taylor expansion (66) and change the index $xy*$ to $x*$ resp. $y*$ in all expressions (63)–(67). Now, by substituting the Taylor expansions into the scheme $G\phi_{i,j}^k$ defined by (60) we get:

$$\begin{aligned} G\phi_{i,j}^k &= \frac{1}{\Delta t} \left(\phi_{i,j}^k - \max_{\theta \in \Theta} \left(f_{i,j}^k(\theta)\Delta t + (1 + r_{i,j}^k(\theta)\Delta t) \right. \right. \\ &\quad \left. \left. \cdot \left(\phi_{i,j}^k + \frac{\partial\phi_{i,j}^k}{\partial t}\Delta t + (L\phi_{i,j}^k + \mathcal{O}(h^\lambda))\Delta t \right) \right) \right) \\ &= F\phi_{i,j}^k + \mathcal{O}(h^\lambda), \end{aligned} \quad (68)$$

where $\lambda = 1/2$ if $|\rho| = 1$ or $\sigma_x = 0$ or $\sigma_y = 0$ and $\lambda = 1$ else. In the first equation of (68) we used the estimates of the linear combinations of higher order terms from the Taylor expansions. The coefficients of these linear combinations are

210

the probabilities $p_o, p_{x+}, p_{x-}, p_{y+}, p_{y-}, p_{xy+}, p_{xy-}$ (according to the definition of G (60)). We describe here these estimates:

1. For the linear combinations of the terms included in $A_\alpha \phi_{i,j}^k$, $\alpha \in \{x+, x-, y+, y-, xy+, xy-\}$ we used the properties of the scheme (17)–(22). After summing all expressions obtained by applying (17)–(22) we get $(L\phi_{i,j}^k + \mathcal{O}(h^{1/2})) \Delta t$. Moreover, following Section 3, if $|\rho| \neq 1, \sigma_x \neq 0, \sigma_y \neq 0$, for h small enough we end up just with $L\phi_{i,j}^k \Delta t$.
2. In the same way, for the linear combinations of the terms included in $\frac{\partial}{\partial t} (A_\alpha \phi_{i,j}^k) \Delta t$ we get $(L(\partial \phi_{i,j}^k / \partial t) + \mathcal{O}(h^\lambda)) (\Delta t)^2 = \mathcal{O}(h) \Delta t$.
3. For the linear combinations of the terms included in $B_\alpha \phi_{i,j}^k$ we used the following estimates:

$$\begin{aligned} & (p_{x+} + p_{xy+})(\Delta_{o+x})^3 + (p_{x-} + p_{xy-})(\Delta_{o-x})^3 \\ & = W_x(\Delta_{o+x} + \Delta_{o-x}) - E_x \Delta_{o+x} \Delta_{o-x} = \mathcal{O}(h) \Delta t, \end{aligned} \quad (69)$$

$$\begin{aligned} & p_{xy+}(\Delta_{o+x})^2 \Delta_{o+y} + p_{xy-}(\Delta_{o-x})^2 \Delta_{o-y} = \\ & \frac{W_{xy}((\Delta_{o+x})^2 \Delta_{o+y} + (\Delta_{o-x})^2 \Delta_{o-y})}{\Delta_{o+x} \Delta_{o+y} + \Delta_{o-x} \Delta_{o-y}} = \mathcal{O}(h) \Delta t. \end{aligned} \quad (70)$$

Here we used that $\Delta_{o+x} + \Delta_{o-x} = \mathcal{O}(h)$, $(\Delta_{o+x})^2 \Delta_{o+y} + (\Delta_{o-x})^2 \Delta_{o-y} = \mathcal{O}(h^2)$. We used analogous estimates also for the terms where x and y are switched symmetrically.

4. As $(\Delta_{o*x})^2 = \mathcal{O}(h)$, $(\Delta_{o*y})^2 = \mathcal{O}(h)$ it is clear that all terms in $\frac{\partial}{\partial t} (B_\alpha R_\alpha) \Delta t$ are of order $\mathcal{O}(h) \Delta t$.
5. For the linear combinations of the terms included in $C_\alpha R_\alpha$ we constructed the estimate in the following way: Let us define

$$\psi = (p_{x+} a_{x+} + p_{xy+} a_{xy+})(\Delta_{o+x})^2 + (p_{x-} a_{x-} + p_{xy-} a_{xy-})(\Delta_{o-x})^2.$$

Then, it holds:

$$\begin{aligned} m(\sigma_x^2 + \mathcal{O}(h)) \Delta t & \leq \psi \leq M(\sigma_x^2 + \mathcal{O}(h)) \Delta t, \\ m & = \min(a_{x+}, a_{x-}, a_{xy+}, a_{xy-}), \quad M = \max(a_{x+}, a_{x-}, a_{xy+}, a_{xy-}). \end{aligned}$$

225 If $m = \mathcal{O}(h)$, $M = \mathcal{O}(h)$ then $\psi = \mathcal{O}(h)\Delta t$. Now for $(a_{x+}, a_{x-}, a_{xy+}, a_{xy-})$ equal to

- $\frac{1}{24} \frac{\partial^4}{\partial x^4} ((\Delta_{o+x})^2 R_{x+}, (\Delta_{o-x})^2 R_{x-}, (\Delta_{o+x})^2 R_{xy+}, (\Delta_{o-x})^2 R_{xy-}),$
- $\frac{1}{6} \frac{\partial^4}{\partial x^3 \partial y} (0, 0, \Delta_{o+x} \Delta_{o+y} R_{xy+}, \Delta_{o-x} \Delta_{o-y} R_{xy-}),$
- $\frac{1}{4} \frac{\partial^4}{\partial x^3 \partial y} (0, 0, (\Delta_{o+y})^2 R_{xy+}, (\Delta_{o-y})^2 R_{xy-}),$

230 which are all $\mathcal{O}(h)$, we get that the linear combinations of the first three terms in $C_\alpha R_\alpha$ are of order $\mathcal{O}(h)\Delta t$. Using analogous estimates, also the linear combinations of the fourth and fifth term are of order $\mathcal{O}(h)\Delta t$.

Using the estimates above, we proved (68) which is the first order consistency of our scheme if $|\rho| < 1$, $\sigma_x > 0$, $\sigma_y > 0$ and consistency of order 1/2 else. \square

235 Now the lemma establishing monotonicity of the scheme follows:

Lemma 2 (Monotonicity). *If $1 + r_{i,j}^k(\theta)\Delta t \geq 0$ for all possible i, j, k, θ , then the discrete scheme (60) is monotone.*

Proof. In this case, monotonicity is a direct implication of the non-negativity of $p_o, p_{x+}, p_{x-}, p_{y+}, p_{y-}, p_{xy+}, p_{xy-}$. \square

240 **Remark 1.** *As already mentioned by the authors in [4], even if $1 + r_{i,j}^k(\theta)\Delta t < 0$ for some i, j, k, θ , we can obtain a monotone scheme if we substitute $1 + r_{i,j}^k(\theta)\Delta t$ by $1/(1 - r_{i,j}^k(\theta)\Delta t)$ in (60) for these parameters i, j, k, θ . Note that this change does not harm the consistency, nor the stability of the scheme.*

245 The stability analysis of the 2D Tree-Grid method is basically identical to the stability analysis of the one-dimensional Tree-Grid method done in [4]. Therefore we state here just the stability condition and the Lemma about stability of the scheme.

Property 1 (Stability condition of the problem). *We suppose that:*

- 250 1. *There exist constants C_f, C_r such that for all $x, y, t, \theta \in [x_1, x_{N_x}] \times [y_1, y_{N_y}] \times [t_1, t_M] \times \Theta$ holds: $|f(x, y, t, \theta)| < C_f$, $|r(x, y, t, \theta)| < C_r$.*

2. There exist a constant C_{BC} such that $|BC(x, y, t)| < C_{BC}$ holds for all $t \in [t_1, t_M]$ and for all possible outer-domain values (x, y) of the variables $(x_+, y_j), (x_-, y_j), (x_i, y_+), (x_i, y_-), (x_+, y_+), (x_+, y_-), (x_-, y_+), (x_-, y_-)$ for any grid.

255

The lemma about stability of the scheme follows:

Lemma 3 (Stability). *If the problem satisfies the stability conditions defined in Property 1, then the scheme (60) is stable.*

Proof. The proof of this lemma can be found in [4]. □

260

Finally, the convergence theorem follows:

Theorem 2 (Convergence of the 2D Tree-Grid Method). *The approximation computed with the 2D Tree-Grid Method (defined by Algorithm 1) for solving the SCP (1)–(4) and the corresponding HJB equation (6)–(8) satisfying the strong uniqueness property (see [15]) and the stability conditions defined in Property 1 converges to the viscosity solution of this SCP (and the HJB equation).*

265

Proof. The proof follows from the Theorem 1 and Lemmas 1, 2, 3. □

5. Application to option pricing models

5.1. Two-factor uncertain volatility model

Here we will use the 2D Tree-Grid method for pricing options on two different risky assets under uncertain volatility. In this setting, the volatilities and the correlation of the assets are only known to lie in certain bounds. In this case the maximal option price can be computed as the solution of the HJB equation

$$\frac{\partial V}{\partial t} + \max_{\theta \in \Theta} (LV - rV) = 0, \quad (71)$$

$$LV = \frac{\sigma_x^2}{2} x^2 \frac{\partial^2 V}{\partial x^2} + \rho \sigma_x \sigma_y xy \frac{\partial^2 V}{\partial x \partial y} + \frac{\sigma_y^2}{2} y^2 \frac{\partial^2 V}{\partial y^2} + rx \frac{\partial V}{\partial x} + ry \frac{\partial V}{\partial y}, \quad (72)$$

$$V(x, y, T) = V_T(x, y), \quad (73)$$

$$0 < t < T, \quad x, y \in \mathbb{R}.$$

Here, x, y, t denote the first and the second asset price and the time, r denotes the risk-free interest rate, T is the maturity time, the terminal condition $V_T(x, y)$ is the payoff function and for the control variable holds $\theta = (\sigma_x, \sigma_y, \rho)$, where σ_x, σ_y, ρ are uncertain volatilities of the first and the second asset, and their correlation. For the set of possible controls holds:

$$\Theta = [\sigma_{x,min}, \sigma_{x,max}] \times [\sigma_{y,min}, \sigma_{y,max}] \times [\rho_{min}, \rho_{max}]. \quad (74)$$

To obtain the minimal option price, we have to replace 'max' by 'min' in the HJB equation (71). This model was discussed for example in [18] and later solved with a hybrid (combining fixed and wide stencils) implicit method in [7]. As explained in [7], the optimal control lies in a subset of Θ ,

$$\begin{aligned} \tilde{\Theta} = & \left((\{\sigma_{x,min}, \sigma_{x,max}\} \times [\sigma_{y,min}, \sigma_{y,max}]) \right. \\ & \left. \cup ([\sigma_{x,min}, \sigma_{x,max}] \times \{\sigma_{y,min}, \sigma_{y,max}\}) \right) \times \{\rho_{min}, \rho_{max}\}. \end{aligned} \quad (75)$$

Therefore, we will search for the control in $\bar{\Theta}$, a discretized version of $\tilde{\Theta}$. To verify the implementation, we will also solve a problem with an one-element set $\bar{\Theta} = \{(\sigma_x, \sigma_y, \rho)\}$. In this case, the equation (71) is simply the 2-dimensional Black-Scholes equation [19] for which the analytical solution is known.

Terminal condition: As a terminal condition, we use the payoff function in the form of a butterfly spread on the maximum of two assets:

$$\begin{aligned} V_T(x, y) &= (M - K_1)^+ - 2(M - (K_1 + K_2)/2)^+ + (M - K_2)^+ \\ M &= \max(x, y), \quad K_1 = 34, \quad K_2 = 46. \end{aligned}$$

Boundary conditions: We will use Dirichlet boundary conditions. On the upper and the right boundary, we will set the value to 0:

$$BC(x > x_{max}, y, t) = 0, \quad BC(x, y > y_{max}, t) = 0, \quad x_{max} = y_{max} = 144.$$

To verify that our computational domain is large enough, we solved the HJB equation and the Black-Scholes equation also for $x_{max} = y_{max} = 400$ and obtained in node $(x, y, t) = (40, 40, 0)$ the same values as for $x_{max} = y_{max} = 144$.

On the lower ($y = 0$) and left ($x = 0$) boundary, the equation (71) degenerates to a HJB equation for the one-dimensional uncertain volatility model from [2]. We solve it with the one-dimensional Tree-Grid method, as in paper [4]. For the case that the values from outside of the computational domain $[0, x_{max}] \times [0, y_{max}]$ are needed, we artificially define the solution for $x < 0, y < 0$ to have the same values as the solution on the boundary:

$$BC(x < 0, y) = BC(0, y), \quad BC(x, y < 0) = BC(x, 0), \quad BC(x < 0, y < 0) = 0.$$

5.2. Numerical results

Here we present the experimental convergence results of the 2D Tree-Grid method applied to the Black-Scholes model and the uncertain volatility model. We implemented our method in Python ¹ and tested on an Intel Core i7-4770 CPU 3.40GHz computer with 16 GB RAM. We performed the simulations on four different sets of parameters:

- Black-Scholes model with moderate volatility and correlation: $\sigma_x = 0.3$, $\sigma_y = 0.5$, $\rho = 0.4$. Results of the simulation are in Table 1.
- Black-Scholes model with extreme parameters: $\sigma_x = 0.05$, $\sigma_y = 0.05$, $\rho = -0.95$. Results of the simulation are in Table 2.
- Uncertain volatility model, maximal option price (worst case scenario) with parameters $\sigma_{x,min} = \sigma_{y,min} = \rho_{min} = 0.3$, $\sigma_{x,max} = \sigma_{y,max} = \rho_{max} = 0.5$. Results of the simulation are in Table 3.
- Uncertain volatility model, minimal option price (best case scenario). The max operator is replaced by min in equation (71), all other parameters are the same as in the worst case scenario. Results of the simulation are in Table 4.

In all models we used the parameters $T = 0.25$, $r = 0.05$. For each model we computed the approximations of the solution on different refinement levels. Let

¹The code can be downloaded from <https://github.com/igor-vyr/Tree-Grid-method>

us denote the final time-layer ($t = 0$) of the approximation of the solution on the k -th refinement level as A_k , and final time-layer of the reference solution as A^{ref} . We measured the *approximation error* on each refinement level in two different ways:

1. the L_1 error - the error was computed using the formula:

$$Error A^k = \|A^k - A^{ref}\|_1, \quad (76)$$

2. the error in $(x = 40, y = 40)$, was computed using the formula

$$Error A^k = |A^k(40, 40) - A^{ref}(40, 40)|. \quad (77)$$

We use this error and the value in $(x = 40, y = 40)$ to compare our results with results from the paper [7], where the same parameters are used for the uncertain volatility model.

To compute the *experimental order of convergence (EOC)* we used the following formula:

$$EOC A^k = \frac{\log(Err A^{k-1}) - \log(Err A^k)}{\log(h_{k-1}) - \log(h_k)}. \quad (78)$$

In all refinement levels we used a (rectangular) grid with equidistant time-stepping and non-equidistant space-stepping with more nodes near to the non-smooth regions of the terminal conditions. The refinements were done uniformly.

From Table 2 we can deduce, that the numerical solution converges also in the case of very small volatility and large correlation, although the convergence is not as smooth as in the case of moderate parameters (Table 1). As we can see in Tables 1, 3, the point-wise convergence may be quite non-smooth, even if the approximation is converging relatively smoothly in L_1 . In the uncertain volatility model, the values in $(x = 40, y = 40)$ are similar to the values of Ma and Forsyth [7], what verifies our method. The experimental order of convergence in the Tables 1, 3, 4 seems to be slightly better than the theoretical order 1, the experimental order of convergence in Table 2 is quite non-smooth.

The increase of the experimental order of convergence in the last refinement level in the uncertain volatility model results from using a solution computed on

Table 1: Black-Scholes model, $\sigma_x = 0.3, \sigma_y = 0.5, \rho = 0.4$. M -number of time steps, N -number of space nodes. Value, error and experimental order of convergence (EOC) in the final time layer in the point $(x, y) = (40, 40)$ and error, EOC in L_1 norm in the final time layer. As reference solution the exact solution (computed with the R-library fExoticOptions) was used.

M	N	Convergence in $(x, y) = (40, 40)$			Convergence in L_1	
		Value	Error	EOC	Error	EOC
25	35^2	1.9910	1.77E-01	-	1.21E-02	-
50	69^2	1.8211	7.02E-03	4.66	1.84E-03	2.72
100	137^2	1.8229	8.88E-03	-0.34	7.16E-04	1.36
200	273^2	1.8177	3.67E-03	1.27	3.02E-04	1.25
400	545^2	1.8141	5.31E-05	6.11	1.24E-04	1.28
800	1089^2	1.8138	1.96E-04	-1.89	5.11E-05	1.28
1600	2177^2	1.8139	1.38E-04	0.51	2.54E-05	1.01

Table 2: Black-Scholes model, $\sigma_x = 0.05, \sigma_y = 0.05, \rho = -0.95$. M -number of time steps, N -number of space nodes. Value, error and experimental order of convergence (EOC) in the final time layer in the point $(x, y) = (40, 40)$ and error, EOC in L_1 norm in the final time layer. As reference solution the exact solution (computed with the R-library fExoticOptions) was used.

M	N	Convergence in $(x, y) = (40, 40)$			Convergence in L_1	
		Value	Error	EOC	Error	EOC
25	35^2	3.3619	1.28E+00	-	3.60E-02	-
50	69^2	3.8702	7.71E-01	0.73	1.96E-02	0.88
100	137^2	4.3095	3.31E-01	1.22	7.62E-03	1.37
200	273^2	4.6339	6.91E-03	5.58	5.53E-04	3.78
400	545^2	4.6465	5.73E-03	0.27	3.86E-05	3.84
800	1089^2	4.6468	5.97E-03	-0.06	4.61E-05	-0.26
1600	2177^2	4.6416	8.35E-04	2.84	9.85E-06	2.22

a fine grid as reference solution (that is disproportionally closer to the solution on the last refinement level in contrast to the solutions on previous refinement

Table 3: Uncertain volatility model, worst case scenario (maximization). M -number of time steps, N -number of space nodes, Q -number of controls. Value, error and experimental order of convergence (EOC) in the final time layer in the point $(x, y) = (40, 40)$ and error, EOC in L_1 norm in the final time layer. As reference solution, a solution computed on a fine grid with 400 time steps, 545^2 space nodes and 256 controls was used.

M	N	Q	Convergence in $(x, y) = (40, 40)$			Convergence in L_1	
			Value	Error	EOC	Error	EOC
25	35^2	16	2.8364	1.59E-01	-	1.17E-02	-
50	69^2	32	2.6619	1.53E-02	3.38	3.64E-03	1.68
100	137^2	64	2.6784	1.19E-03	3.68	1.17E-03	1.64
200	273^2	128	2.6784	1.20E-03	-0.01	3.07E-04	1.93

Table 4: Uncertain volatility model, best case scenario (minimization). M -number of time steps, N -number of space nodes, Q -number of controls. Value, error and experimental order of convergence (EOC) in the final time layer in the point $(x, y) = (40, 40)$ and error, EOC in L_1 norm in the final time layer. As reference solution, a solution computed on a fine grid with 400 time steps, 545^2 space nodes and 256 controls was used.

M	N	Q	Convergence in $(x, y) = (40, 40)$			Convergence in L_1	
			Value	Error	EOC	Error	EOC
25	35^2	16	0.9847	6.95E-02	-	4.29E-03	-
50	69^2	32	0.9475	3.23E-02	1.11	2.01E-03	1.09
100	137^2	64	0.9270	1.18E-02	1.46	7.59E-04	1.41
200	273^2	128	0.9173	2.07E-03	2.50	1.31E-04	2.53

315 levels).

In Figure 2, we see the final time layer ($t = 0$) of the approximation of option prices under the uncertain volatility model for both best and worst case scenario.

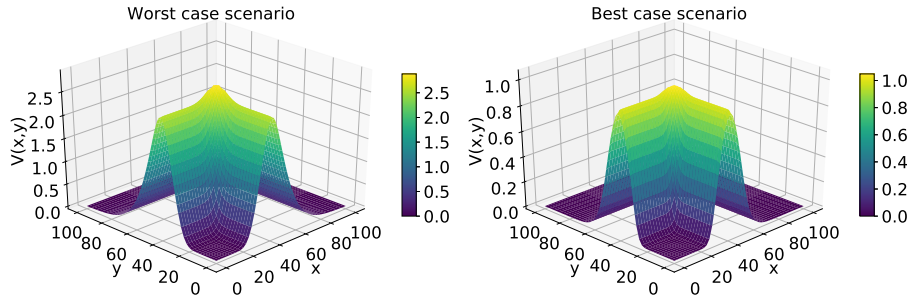


Figure 2: Final time layers ($t=0$) of the approximations of the worst case option price (maximization) and of the best case option price (minimization) from the uncertain volatility model computed with the 2D Tree-Grid method with 50 time steps, 69^2 space nodes and 32 controls.

6. Conclusion

320 In this paper we generalized the one-dimensional Tree-Grid method to 2 space dimensions. Although the ideas behind this two-dimensional method are similar to the one-dimensional case, the algorithm construction is more challenging, mainly because of the correlation between the two spatial variables. Our 2D Tree-Grid method is explicit and hence suitable for parallelization, but
 325 still unconditionally stable and monotone in contrast to other explicit methods [1, 9, 10]. Moreover, unlike other wide-stencil methods [7, 9, 10], it doesn't use any interpolation. These properties make the method very flexible and, by simply following of the Algorithm 1, also easy to implement.

We proved the convergence of the 2D Tree-Grid method using the theory of
 330 Barles and Souganidis [15]. We tested the method on the two-factor uncertain volatility model and on the Black-Scholes model for option pricing and verified the convergence also experimentally.

Acknowledgements

This research was supported within the bilateral German-Slovakian Project
 335 *ENANEFA - Efficient Numerical Approximation of Nonlinear Equations in Financial Applications* financed by DAAD and the Slovakian Ministry of Education (01/2018–12/2019).

References

- [1] H. Kushner, P. G. Dupuis, Numerical methods for stochastic control problems in continuous time, Vol. 24 of Applications of Mathematics, Springer Science & Business Media, 2013.
- [2] P. A. Forsyth, K. R. Vetzal, Numerical methods for nonlinear PDEs in finance, in: Handbook of Computational Finance, Springer, 2012, pp. 503–528.
- [3] S. Kilianová, D. Ševčovič, A transformation method for solving the Hamilton-Jacobi-Bellman equation for a constrained dynamic stochastic optimal allocation problem, The ANZIAM Journal 55 (01) (2013) 14–38.
- [4] I. Kossaczký, M. Ehrhardt, M. Günther, A new convergent explicit Tree-Grid method for HJB equations in one space dimension, Applied Mathematics Letters 11 (2018) 1–29.
- [5] I. Kossaczký, M. Ehrhardt, M. Günther, The Tree-Grid Method with control-independent stencil, in: Proceedings of Equadiff 2017 Conference, Bratislava, 2017, pp. 79–88.
- [6] 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.
- [7] K. Ma, P. Forsyth, An unconditionally monotone numerical scheme for the two-factor uncertain volatility model, IMA Journal of Numerical Analysis 37 (2) (2016) 905–944.
- [8] Y. Chen, J. W. Wan, Monotone mixed narrow/wide stencil finite difference scheme for Monge-Ampère equation, arXiv preprint arXiv:1608.00644.
- [9] J. F. Bonnans, H. Zidani, Consistency of generalized finite difference schemes for the stochastic HJB equation, SIAM Journal on Numerical Analysis 41 (3) (2003) 1008–1021.

- 365 [10] K. Debrabant, E. Jakobsen, Semi-Lagrangian schemes for linear and fully
non-linear diffusion equations, *Mathematics of Computation* 82 (283)
(2013) 1433–1462.
- [11] K. Debrabant, E. R. Jakobsen, Semi-Lagrangian schemes for linear
and fully non-linear Hamilton-Jacobi-Bellman equations, arXiv preprint
370 arXiv:1403.1217.
- [12] J. Yong, X. Y. Zhou, *Stochastic controls: Hamiltonian systems and
HJB equations*, Vol. 43 of *Stochastic Modelling and Applied Probability*,
Springer Science & Business Media, 1999.
- [13] S. Mataramvura, B. Øksendal, Risk minimizing portfolios and HJBI equa-
375 tions for stochastic differential games, *Stochastics An International Journal
of Probability and Stochastic Processes* 80 (4) (2008) 317–337.
- [14] Q. Song, Convergence of Markov chain approximation on generalized HJB
equation and its applications, *Automatica* 44 (3) (2008) 761–766.
- [15] G. Barles, P. E. Souganidis, Convergence of approximation schemes for fully
380 nonlinear second order equations, in: *Asymptotic Analysis*, no. 4, 1991, pp.
2347–2349.
- [16] 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.
- 385 [17] P.-L. Lions, Optimal control of diffusion processes and Hamilton–Jacobi–
Bellman equations part 2: viscosity solutions and uniqueness, *Communi-
cations in Partial Differential Equations* 8 (11) (1983) 1229–1276.
- [18] D. M. Pooley, P. A. Forsyth, K. R. Vetzal, Numerical convergence prop-
erties of option pricing PDEs with uncertain volatility, *IMA Journal of*
390 *Numerical Analysis* 23 (2) (2003) 241–267.
- [19] F. Black, M. Scholes, The pricing of options and corporate liabilities, *Jour-
nal of Political Economy* 81 (3) (1973) 637–654.