J Sci Comput (2012) 50:462–492
DOI 10.1007/s10915-011-9493-3

An ENO-Based Method for Second-Order Equations and Application to the Control of Dike Levels

S.P. van der Pijl · C.W. Oosterlee

Received: 5 December 2009 / Revised: 18 August 2010 / Accepted: 24 April 2011 /
Published online: 15 May 2011

Abstract This work aims to model the optimal control of dike heights. The control problem leads to so-called Hamilton-Jacobi-Bellman (HJB) variational inequalities, where the dike-increase and reinforcement times act as input quantities to the control problem. The HJB equations are solved numerically with an Essentially Non-Oscillatory (ENO) method. The ENO methodology is originally intended for hyperbolic conservation laws and is extended to deal with diffusion-type problems in this work. The method is applied to the dike optimisation of an island, for both deterministic and stochastic models for the economic growth.

Keywords Hamilton-Jacobi-Bellman equations · ENO scheme for diffusion · Impulsive control · Dike increase against flooding

1 Introduction

The optimal control of dike heights as a protection against flooding is a trade-off between the investment costs of dike increases and the expected costs due to flooding. This concept of economic optimisation was established by Van Dantzig [19] in the aftermath of the flooding disaster that hit the Netherlands in 1953. Van Dantzig's model was deterministic and discrete in time and was later improved by Eijgenraam [9] to properly account for economic growth.

The present work uses a model in which the stochastic behaviour of economic growth is modelled in continuous time. The resulting optimisation problem leads to a so-called Hamilton-Jacobi-Bellman (HJB) equation. It is a system of second-order partial differential equations that needs to be solved backwards in time. This is achieved by numerical approximation.

S.P. van der Pijl · C.W. Oosterlee (✉)
CWI—Center for Mathematics and Computer Science, Amsterdam, The Netherlands
e-mail: c.w.oosterlee@cwi.nl

C.W. Oosterlee
Delft University of Technology, Delft, The Netherlands

There is a long tradition of numerically solving optimal control problems via the HJB equations, and very nice books and papers have been written on the topic, see [1, 3, 4, 10]. For second-order equations, however, at most second-order accurate discretizations were used, for example, based on the notion of viscosity solution. In the present paper we aim for higher-order discretizations.

A wide variety of numerical methods for partial differential equations exists. The proper choice for a numerical method is motivated by carefully considering the properties of the problem. The state vector can be of high dimension, the time horizon large and the equations of convective type, i.e. the terms containing first-order derivatives may be dominant.

The uniqueness requirement for the solution of nonlinear partial differential equations such as the HJB equations is non-trivial and greatly affects their numerical treatment. This is also encountered, among other areas, in Mathematical Finance, where the relevant solution is also the viscosity solution [10]. Since it is known from theory that a stable, consistent and monotone discretization converges to the viscosity solution [2], researchers such as Chen et al. [6] elaborate on a monotone discretization to guarantee convergence to the viscosity solution. The drawback of a monotone method is that it has limited order of accuracy. Similar issues arise in the closely-related field of hyperbolic conservation laws, where the only relevant solution is the so-called entropy solution, see e.g. [13]. Striving for higher-order accuracy than purely monotone schemes, we will adopt the well-established rationale of the realm of Computational Fluid Dynamics (CFD).

The Total Variation (TV) plays a central role in nonlinear stability theory of CFD methods. It is defined as

$$TV(v) = \limsup_{\epsilon \to 0} \int_{-\infty}^{\infty} |v(x) - v(x-\epsilon)| \, \mathrm{d}x, \tag{1}$$

and a similar definition for its discrete counterpart. An important class of methods in CFD are the so-called Total Variation Diminishing (TVD) methods, i.e. $TV(u(\cdot, t + \Delta t)) \leq TV(u(\cdot, t))$. The TVD property makes the methods TV-stable. This is important, because if a method is in conservation form, consistent and TV-stable, then convergence can be proven [13]. TVD methods are monotonicity preserving in the sense that they prevent Gibbs-like oscillations near discontinuities in the solution. TVD methods are non-linear and their accuracy falls back to first-order near discontinuities.

To reach a higher order of accuracy, we will use Essentially Non-Oscillatory (ENO) methods. ENO methods are not TVD, hence monotonicity preservation and convergence are strictly speaking unproven. However, there is a strong belief that ENO methods are TV-stable, at least for most practical problems [17]. Spurious oscillations, on the level of the truncation error, may occur only in the smooth part of the solution [11]. Hence the name *Essentially* Non-Oscillatory. ENO-type schemes have been used for HJB equations before for first-order equations, see [5] and the references therein. We will encounter second-order HJB equations, so that we have to deal with an ENO discretization for the diffusion terms.

The diffusion term in the equations, associated with the stochastic behaviour of the model, poses new difficulties. We will show that a standard high-order discretization leads to a non-monotone discretization. This is often disregarded and one relies on the smoothing behaviour of the elliptic diffusion operator. However, for non-smooth initial data, undesired results, such as oscillatory and negative values, are encountered at the initial stages. Therefore, in this paper we extend and apply the ENO methodology to the diffusion operator as well.

We will combine the high-order ENO finite differences spatial discretization with a high-order TVD Runge-Kutta time integration method, as prescribed in [17], for the HJB equations, including diffusion. A potential drawback could be that this restricts the time-step

when the diffusion coefficient (volatility) in the model is large, compared to implicit (non-TVD) schemes. However, the high-order accuracy of the spatial discretization reduces the required spatial grid resolution and, as an immediate consequence, the number of time steps as well.

The HJB inequalities that arise in Mathematical Finance applications are governed by non-differentiable or even discontinuous final conditions. We will develop a numerical method which is also applicable to such control problems.

This paper is organised as follows: In Sect. 2, the numerical discretization, by means of ENO schemes is discussed, where Sect. 2.2 introduces ENO schemes for diffusion. In Sect. 3, the mathematical problem of flooding and dike height control is described in detail and numerical results with the ENO scheme are presented. Section 4 concludes.

2 Numerical Approach

Formulating an optimal control problem requires an expression for the total future expected discounted costs and input variables. Optimisation of the costs will yield a control law for the input variables, which governs optimal values. We consider an *impulsive control formulation*, where it is assumed that the input variable increases instantaneously. Both the optimal input variable and the intervention times are typically not known in such *optimal stopping problems* [3] and need to be determined. The numerical treatment of impulse control model equations can be split into two parts: the uncontrolled problem, i.e. between intervention times t_k and $t_{k+1}-$ and the impulse control.

These problems are typically defined backwards in time, starting at final time, making optimal decisions at times t_k until initial time t_0 is reached and a control law is defined.

2.1 Uncontrolled Problem

The uncontrolled part of the problem is typically of convection-diffusion-reaction type and has the following general form:

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f}(u) = \nabla \cdot K(u,t)\nabla u + \mathbf{s}(u,t), \tag{2}$$

complemented with appropriate boundary and initial conditions. For the sake of simplicity, time is reversed to bring it in an initial-value-problem form. Furthermore, the convection part $\nabla \cdot \mathbf{f}$ relates to the deterministic part of the system dynamics, while the diffusion part relates to the stochastic behaviour. The source term, $\mathbf{s}$, in (2) accounts for the running costs and discounting (as we will see in Sect. 3). Note that (2) is brought into conservative form, whereas the original operator in a control problem is not. This is not strictly necessary for the problem under consideration, but is, say, generally beneficial for the numerical treatment of models based on conservation laws.

We mentioned in the introduction that the nonlinearity of the HJB partial differential equations raises the question about uniqueness of the solution and its consequences for the discretization. This motivated us to employ the ENO methodology. It combines high-order accuracy with convergence to the relevant (viscosity, or entropy) solution, albeit strictly speaking unproven, but proven satisfactory in many applications.

The boundaries of the computational domain are often regular in control problems, so a method that relies on Cartesian meshes, possibly combined with coordinate mappings,

would suffice. Furthermore, the state space can be up to three-dimensional and straightforward dimensional splitting would be advantageous, pointing towards a finite difference setting. Finally, to respect the conservative nature of (2), although not strictly necessary for the dike problem discussed in Sect. 3, we use conservative finite differences in the Shu-Osher form [14, 17, 18].

The model equations are of purely convective type when the system dynamics are fully *deterministic*, i.e. $K(u,t)=0$ in (2) and fully diffusive when the *drift* in a stochastic state system vanishes, i.e. $\mathbf{f}(u)=\mathbf{0}$. We do not want to make any assumptions on the magnitude of drift and diffusion and will discuss their discretizations separately. We will now give a brief overview of the ENO method, as originally intended for convection-type problems, with the *purpose of extending it to the discretization of the diffusion operator* in the next section.

2.1.1 Convection

Assume a fully deterministic problem and no source terms. The model equation, (2), reduces to

$$\frac{\partial u}{\partial t}+\nabla\cdot\mathbf{f}(u)=0, \tag{3}$$

which is a hyperbolic conservation law. A discretization that is inherently conservative can be obtained by integrating (3) over a control volume. Assume a Cartesian mesh with coordinate directions x_i, such that $\mathbf{x}=(x_1,\ldots,x_N)^t$, and grid spacings Δx_i and define the sliding average operators $\mathcal{A}_i$ and difference operators Δ_i along the line of Merriman [14] as follows:

$$\mathcal{A}_i\Phi(\mathbf{x})=\frac{1}{\Delta x_i}\int_{-\frac{1}{2}\Delta x_i}^{\frac{1}{2}\Delta x_i}\Phi(\mathbf{x}+\xi\,\mathbf{e}_i)\,\mathrm{d}\xi, \tag{4}$$

$$\Delta_i\Phi(\mathbf{x})=\Phi\left(\mathbf{x}+\frac{1}{2}\Delta x_i\mathbf{e}_i\right)-\Phi\left(\mathbf{x}-\frac{1}{2}\Delta x_i\mathbf{e}_i\right), \tag{5}$$

where $\mathbf{e}_i$ is the unit vector in the i^{th} coordinate direction. The volume average is then $\mathcal{A}_1\mathcal{A}_2\ldots\mathcal{A}_N$, which, applied to (3), yields

$$\begin{aligned}\mathcal{A}_1\mathcal{A}_2\ldots\mathcal{A}_N\frac{\partial u}{\partial t}=&-\frac{\Delta_1\mathcal{A}_2\mathcal{A}_3\ldots\mathcal{A}_N f_1}{\Delta x_1}-\frac{\Delta_2\mathcal{A}_1\mathcal{A}_3\ldots\mathcal{A}_N f_2}{\Delta x_2}+\ldots\\&-\frac{\Delta_N\mathcal{A}_1\ldots\mathcal{A}_{N-2}\mathcal{A}_{N-1}f_N}{\Delta x_N},\end{aligned} \tag{6}$$

where $f_1, f_2, \ldots f_N$ are the components of the flux vector $\mathbf{f}$. Since the mesh widths Δx_i are constant, the difference and average operators commute, so that

$$\frac{\partial u}{\partial t}=-\frac{\Delta_1{\mathcal{A}_1}^{-1}f_1}{\Delta x_1}-\frac{\Delta_2{\mathcal{A}_2}^{-1}f_2}{\Delta x_2}+\ldots-\frac{\Delta_N{\mathcal{A}_N}^{-1}f_N}{\Delta x_N}. \tag{7}$$

Note that this will not work on non-uniform meshes. In that case, we will use a coordinate-transformation to a uniform mesh. When we define h_i as

$$h_i=\mathcal{A}_i^{-1}f_i, \tag{8}$$

equation (3) takes the Shu-Osher conservative-difference form

$$\frac{\partial u}{\partial t} = -\sum_{i=1}^{N} \frac{\Delta_i h_i}{\Delta x_i} = -\sum_{i=1}^{N} \frac{h_i(\mathbf{x} + \frac{1}{2}\Delta x_i \mathbf{e}_i) - h_i(\mathbf{x} - \frac{1}{2}\Delta x_i \mathbf{e}_i)}{\Delta x_i}. \tag{9}$$

A conservative discretization of (3) is obtained by simply evaluating (9) at nodal points. The advantage of the Shu-Osher form is immediately apparent; a "dimension-by-dimension" operator-splitting technique is permitted and, as a consequence, purely one-dimensional reconstructions to find $h_i(\mathbf{x} + \frac{1}{2}\Delta x_i \mathbf{e}_i)$ may be applied to each coordinate direction. This makes the Shu-Osher form very well suited for high-dimensional problems on Cartesian meshes. Note that (9) is still exact when evaluated at nodal points. The remaining question is how to reconstruct h at intermediate locations, $\mathbf{x} + \frac{1}{2}\Delta x_i \mathbf{e}_i$, from the nodal values.

The ENO doctrine reconstructs fluxes recursively with an increasing order of accuracy by adding neighbouring nodes to the stencil. To this end a Newton polynomial in the neighbourhood of $\mathbf{x} + \frac{1}{2}\Delta x \mathbf{e}_i$ is constructed. Starting from a single node, the stencil is recursively extended with neighbouring nodes. These neighbouring nodes are selected based on a criterion on the divided difference table, and is such that it yields the smoothest possible interpolating polynomial.

For the ease of notation, we will exploit operator splitting and consider one spatial dimension only, i.e.

$$\frac{\partial u}{\partial t} = -\frac{h(x + \frac{1}{2}\Delta x) - h(x - \frac{1}{2}\Delta x)}{\Delta x}. \tag{10}$$

First define a Cartesian grid that comprises nodes $x_j = j\Delta x$. We will now use subscripts to refer to nodal values, e.g. $u_j(t) = u(x_j, t)$. The semi-discretization of (10) is simply

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = -\frac{h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}}}{\Delta x}. \tag{11}$$

Shu and Osher [18] introduce the primitive $H(x)$ of $h(x)$:

$$h(x) = \frac{\mathrm{d}H}{\mathrm{d}x}(x). \tag{12}$$

Combining this with the definition of $h(x)$ in (8) yields (omitting the time-dependency of u)

$$\frac{H(x + \frac{1}{2}\Delta x) - H(x - \frac{1}{2}\Delta x)}{\Delta x} = f(u(x)). \tag{13}$$

In other words, the *divided difference table* of H can be computed from the divided difference table of $f(x)$, whose values are known at the nodes, i.e.

$$H[x_{j-\frac{1}{2}}, \ldots, x_{j-\frac{1}{2}+k}] = \frac{1}{k} f[u(x_j), \ldots, u(x_{j+k-1})], \tag{14}$$

where the square brackets indicate the divided difference. Newton polynomials can now be (recursively) constructed to approximate $H(x)$ in a neighbourhood of $x_{j+\frac{1}{2}}$, see e.g. [11]

$$H(x) = H(x_{j+\frac{1}{2}}) + H[x_{\ell^{(1)}-\frac{1}{2}}, x_{\ell^{(1)}+\frac{1}{2}}]\,(x - x_{j+\frac{1}{2}})$$

$$+\sum_{k=2}^{r} H[x_{\ell^{(k)}-\frac{1}{2}},\ldots,x_{\ell^{(k)}-\frac{1}{2}+k}] \prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k-1} (x-x_{m-\frac{1}{2}})$$
$$+e(x)\Delta x^{r+1}+\mathcal{O}(\Delta x^{r+2}), \tag{15}$$

and, using (12) and (14),

$$h(x_{j+\frac{1}{2}}) = f(u(x_{\ell^{(1)}}))$$
$$+\sum_{k=2}^{r} \frac{f[u(x_{\ell^{(k)}}),\ldots,u(x_{\ell^{(k)}+k-1})]}{k} \frac{\mathrm{d}}{\mathrm{d}x} \prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k-1} (x-x_{m-\frac{1}{2}})\Big|_{x=x_{j+\frac{1}{2}}}$$
$$+d(x_{j+\frac{1}{2}})\Delta x^{r}+\mathcal{O}(\Delta x^{r+1}), \tag{16}$$

where $\ell^{(k)}$ is the leftmost node used in the k^{th} stencil. It is chosen such that the smoothest possible interpolating polynomial is obtained, see [11, 17, 18] for details. If we approximate $h(x_{j+\frac{1}{2}})$ by $h_{j+\frac{1}{2}}$,

$$h_{j+\frac{1}{2}} = f(u(x_{\ell^{(1)}}))$$
$$+\sum_{k=2}^{r} \frac{f[u(x_{\ell^{(k)}}),\ldots,u(x_{\ell^{(k)}+k-1})]}{k} \prod_{\substack{m=\ell^{(k-1)}\\ m\neq j+1}}^{\ell^{(k-1)}+k-1} (x_{j+\frac{1}{2}}-x_{m-\frac{1}{2}}), \tag{17}$$

then apparently (11) is an approximation of (10) with truncation error

$$(d(x_{j+\frac{1}{2}})-d(x_{j-\frac{1}{2}}))\Delta x^{r-1}+\mathcal{O}(\Delta x^{r}),$$

which is $\mathcal{O}(\Delta x^r)$ if $d(x)$ is Lipschitz continuous, see [11].

Returning to the selection of the leftmost node at the kth-level recursion, the first node, $x_{\ell^{(1)}}$ is chosen in correspondence with Godunov's scheme, just like MUSCL schemes do, see [17] for more details, hence

$$\ell^{(1)} = \begin{cases} j, & a_{j+\frac{1}{2}} \geq 0, \\ j+1, & \text{otherwise}, \end{cases} \tag{18}$$

where $a_{j+\frac{1}{2}}$ is the advection velocity, e.g. $a_{j+\frac{1}{2}} = \partial f(x_{j+\frac{1}{2}})/\partial u$. The stencil is widened recursively to yield the smoothest possible interpolation polynomial:

$$\ell^{(k+1)} = \begin{cases} \ell^{(k)}-1, & |f[u(x_{\ell^{(k)}-1}),\ldots,u(x_{\ell^{(k)}+k-1})]| \\ & \quad \leq |f[u(x_{\ell^{(k)}}),\ldots,u(x_{\ell^{(k)}+k})]|, \\ \ell^{(k)}, & \text{otherwise}. \end{cases} \tag{19}$$

A nonlinear stability result for the ENO scheme is the following. It is well-known and mentioned before that if the numerical approximation is Total Variation bounded, it converges to the weak solution of (3) for $\Delta x \to 0$. According to Harten et al. [11], the total variation TV decreases in time up to $\mathcal{O}(\Delta x^r)$:

$$TV(u^{n+1}) \leq TV(u^n)+\mathcal{O}(\Delta x^r), \tag{20}$$

under the assumption that the time integration is monotone. Here, superscript n refers to the time level, i.e.

$$u_j^n = u(x_j, t^n), \tag{21}$$

$$u^n = \{u_1^n, u_2^n, \ldots\}. \tag{22}$$

We will apply the high-order Runge-Kutta type TVD time discretizations of Shu and Osher [17] that serve this need.

2.2 Diffusion

We now turn to the discretization of the diffusion operator in (2) by firstly looking at the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \tag{23}$$

with appropriate boundary and initial conditions. The objective is to find a high-order non-oscillatory discretization. Here, we have to keep in mind that the convective part of (2) motivated us to use a Runge-Kutta type TVD time discretization, see Sect. 2.1.1. This is a convex combination of explicit Euler time-steps and according to Shu and Osher [17] it is sufficient for stability to consider a forward-Euler-type numerical method, $u_j^{n+1} = E(u^n; j)$ (hereafter referred as "method E"),

$$E(u^n; j) := u_j^n + \Delta t \; L(u^n; j), \tag{24}$$

where $L(u^n; j)$ is the discretization of $\frac{\partial^2 u}{\partial x^2}(x_j, t^n)$ and again using the notation of (21) and (22).

We will first prove that there is no central and linear scheme, in the sense of (25), (26), possible that is both higher-order (> 2) accurate and yields a monotone numerical method E.

Theorem 2.1 *There is no central difference scheme L of order of accuracy higher than two for which the numerical method E is monotone.*

Proof We will construct the high-order discretization by means of Richardson extrapolation by taking a linear combination of the well-known second-order approximation, with mesh widths $k\Delta x$:

$$L^k(u^n; j) = \frac{1}{(k\Delta x)^2}\left(u_{j-k}^n - 2u_j^n + u_{j+k}^n\right), \tag{25}$$

and

$$L(u^n; j) = \sum_{k=1}^{N} \alpha_k L^k(u^n; j). \tag{26}$$

The constants α_k must be such that

1. L is consistent,
2. L has truncation error $\mathcal{O}(\Delta x^{2(M+1)})$ and $M \geq 1$,
3. E is monotone, i.e. $\frac{\partial E(u;j)}{\partial u_i} \geq 0, \forall i, j$.

ad 1. Consistency
We require

$$\sum_{k=1}^{N} \alpha_k = 1. \tag{27}$$

ad 2. Truncation error
The truncation error τ^k of L^k, with mesh width $k\Delta x$, is

$$\tau^k = K_1(k\Delta x)^2 + \ldots + K_M(k\Delta x)^{2M} + \mathcal{O}(\Delta x^{2(M+1)}), \tag{28}$$

and the truncation error τ of L is then

$$\tau = K_1 \sum_{k=1}^{N} \alpha_k (k\Delta x)^2 + \ldots + K_M \sum_{k=1}^{N} \alpha_k (k\Delta x)^{2M} + \mathcal{O}(\Delta x^{2(M+1)}). \tag{29}$$

For an $\mathcal{O}(\Delta x^{2(M+1)})$ method we require

$$\sum_{k=1}^{N} \alpha_k k^{2m} = 0, \quad m = 1, \ldots, M. \tag{30}$$

ad 3. Monotonicity
Substitution of (25) and (26) in (24) reveals for E:

$$E(u^n; j) = \frac{\Delta t}{\Delta x^2} \left(\sum_{k=1}^{N} \frac{\alpha_k}{k^2} (u^n_{j-k} + u^n_{j+k}) + \left(\frac{\Delta x^2}{\Delta t} - 2 \sum_{k=1}^{N} \frac{\alpha_k}{k^2} \right) u^n_j \right) \tag{31}$$

and to satisfy the monotonicity constraint $\frac{\partial E(u;j)}{\partial u_i} \geq 0, \forall i, j$, we require

$$\alpha_k \geq 0, \quad k = 1, \ldots, N, \tag{32}$$

$$\sum_{k=1}^{N} \frac{\alpha_k}{k^2} \leq \frac{1}{2} \frac{\Delta x^2}{\Delta t}. \tag{33}$$

Substituting (32) into (30) gives $\alpha_k = 0, k = 1, \ldots, N$, which is in contradiction with (27). This proves the theorem. □

The non-monotonicity of the high-order discretization of the diffusion operator is often disregarded and one relies on its smoothing behaviour. However, for non-smooth initial data, undesired results, such as oscillatory and negative values, are encountered at the initial stages.

Theorem 2.1 for the discretization of the heat equation can be viewed as a Godunov order barrier theorem (see for example [13], for discretizations of the first-order convection operator).[1]

Since no higher-order linear scheme exists which has the desired monotonicity properties, we will revert to non-linear schemes and extend the ENO methodology, originally

[1] Godunov's order barrier theorem states that linear numerical schemes for solving first-order PDEs, having the property of not generating new extrema, can be at most first-order accurate.

intended for first-order derivatives, to second-order derivatives. We will present three approaches. The first one is suitable for discretizing simply $\partial^2 u/\partial x^2$ as in (23). The second one is also applicable to $\partial(k(x)\partial u/\partial x)/\partial x$, where $k(x)$ is some scalar coefficient. The third one is a generalisation and suitable for the form $\nabla \cdot K(u,t)\nabla u$ as in (2), where K is a matrix.

2.2.1 Constant Heat Coefficient

We start with the following proposition:

Proposition 2.2 *An essentially non-oscillatory discretization of the heat equation*, (23), *which is* r^{th}*-order accurate in space, assuming we deal with sufficiently smooth solutions, is obtained by a numerical method based on a Runge-Kutta type TVD time discretization, and on* (11), *in which we substitute*

$$h_{j+\frac{1}{2}} = -\frac{u(x_{\ell^{(1)}+1}) - u(x_{\ell^{(1)}})}{\Delta x} - \sum_{k=2}^{r} \frac{u[x_{\ell^{(k)}}, \ldots, x_{\ell^{(k)}+k}]}{k+1} \frac{\mathrm{d}^2}{\mathrm{d}x^2} \prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k} (x - x_{m-\frac{1}{2}}) \Bigg|_{x=x_{j+\frac{1}{2}}}. \tag{34}$$

We then take

1. $\ell^{(1)} = j$, *compare with* (18),
2. *The smoothest possible interpolation scheme for* $r > 1$ (*compare with* (19)), *i.e.*

$$\ell^{(k+1)} = \begin{cases} \ell^{(k)} - 1 & |u[x_{\ell^{(k)}-1}, \ldots, x_{\ell^{(k)}+k}]| \\ & \quad \le |u[x_{\ell^{(k)}}, \ldots, x_{\ell^{(k)}+k+1}]|, \\ \ell^{(k)} & \text{otherwise}. \end{cases} \tag{35}$$

Outline of Proof:
Let's first consider the discretization of (23) and put it in the form of (3) by substituting $-\partial u/\partial x$ for $f(u)$. This gives (10):

$$\frac{\partial u}{\partial t} = -\frac{h(x + \frac{1}{2}\Delta x) - h(x - \frac{1}{2}\Delta x)}{\Delta x},$$

where h is defined by

$$\mathcal{A}h = -\frac{\partial u}{\partial x}. \tag{36}$$

A straightforward extension of (12) is

$$h(x) = \frac{\mathrm{d}^2 H}{\mathrm{d}x^2}(x), \tag{37}$$

which yields (omitting the time-dependency of u)

$$\frac{\frac{\mathrm{d}H}{\mathrm{d}x}(x + \frac{1}{2}\Delta x) - \frac{\mathrm{d}H}{\mathrm{d}x}(x - \frac{1}{2}\Delta x)}{\Delta x} = -\frac{\partial u}{\partial x}(x), \tag{38}$$

and integration gives the analog of (13)

$$\frac{H(x+\frac{1}{2}\Delta x)-H(x-\frac{1}{2}\Delta x)}{\Delta x}=-u(x)+c, \tag{39}$$

where c is some constant. The divided difference table of H can thus be computed from the divided difference table of u, compare (14),

$$H[x_{j-\frac{1}{2}},\ldots,x_{j-\frac{1}{2}+k}]=-\frac{1}{k}u[x_j,\ldots,x_{j+k-1}],\quad k>1. \tag{40}$$

Note that the constant c is not important, since we only need divided differences of H of second-order and higher, i.e. $k>1$. Substitution in (15) and using (37) gives an expression for $h(x_{j+\frac{1}{2}})$:

$$\begin{aligned}h(x_{j+\frac{1}{2}})=&-\sum_{k=2}^{r}\frac{u[x_{\ell^{(k)}},\ldots,x_{\ell^{(k)}+k-1}]}{k}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k-1}(x-x_{m-\frac{1}{2}})\Bigg|_{x=x_{j+\frac{1}{2}}}\\&+d(x_{j+\frac{1}{2}})\Delta x^{r-1}+\quad\mathcal{O}(\Delta x^r),\end{aligned} \tag{41}$$

where the order of the truncation term is now decreased by one compared to (16), since the second-order derivative in $h:=\frac{\mathrm{d}^2H}{\mathrm{d}x^2}$ replaces the first-order derivative in (12).[2]

Changing the indices k to $k+1$ and $\ell^{(k)}$ to $\ell^{(k-1)}$ and separating the first term in the summation gives us:

$$\begin{aligned}h(x_{j+\frac{1}{2}})=&-u[x_{\ell^{(1)}},x_{\ell^{(1)}+1}]\\&-\sum_{k=2}^{r}\frac{u[x_{\ell^{(k)}},\ldots,x_{\ell^{(k)}+k}]}{k+1}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k}(x-x_{m-\frac{1}{2}})\Bigg|_{x=x_{j+\frac{1}{2}}}\\&+\hat{d}(x_{j+\frac{1}{2}})\Delta x^{r}+\quad\mathcal{O}(\Delta x^{r+1}).\end{aligned} \tag{42}$$

If we approximate $h(x_{j+\frac{1}{2}})$ by $h_{j+\frac{1}{2}}$ in (34), then (11) is an approximation of (23) with truncation error:

$$(\hat{d}(x_{j+\frac{1}{2}})-\hat{d}(x_{j-\frac{1}{2}}))\Delta x^{r-1}+\mathcal{O}(\Delta x^r),$$

which is $\mathcal{O}(\Delta x^r)$, if $\hat{d}(x)$ is Lipschitz continuous.

The remaining task is to prescribe the selection of the leftmost node $\ell^{(k)}$ in the k^{th}-level recursion. When discretizing the convection operator, we saw that the first leftmost node $\ell^{(1)}$ was chosen such that the monotone first-order upwind scheme was recovered for $r=1$, see (18). The subsequent nodes $\ell^{(k)}$, $k=2,\ldots,r$ were such that the smoothest possible interpolation scheme was obtained, see (19). Extending this to the discretization of the diffusion operator, we now require that:

[2]If the order of the truncation term is to match the one of the discretization of convection, the upper limit in the summation in (41) has to be increased from r to $r+1$.

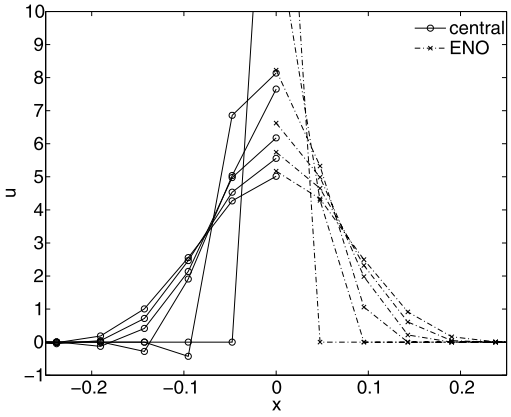


Fig. 1 First six time steps of the numerical solution u of the heat equation; fourth-order central scheme at the *left side*, and fourth-order ENO for diffusion at the *right side* of the picture; third-order RK-TVD time discretization

1. The standard *monotone* three-point central scheme, (25), is recovered for $r = 1$,
2. The scheme is *essentially non-oscillatory* for $r > 1$.

This is guaranteed by the choices in (35). □

Remark 2.3 The first-order scheme, $r = 1$, is actually second-order accurate, provided $\ell^{(1)} = j$, as the contribution of the second-order part of (34), i.e. $k = 2$ in the summation, vanishes, since due to symmetry around $x_{j+\frac{1}{2}}$

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\,(x - x_{j-\frac{1}{2}})(x - x_{j+\frac{1}{2}})(x - x_{j+1\frac{1}{2}})\Big|_{x=x_{j+\frac{1}{2}}} = 0. \tag{43}$$

Remark 2.4 A high-order central and linear scheme in the sense of (25) and (26) can be constructed by taking

$$\ell^{(k+1)} = \begin{cases} \ell^{(k)} - 1 & k \text{ is even}, \\ \ell^{(k)} & k \text{ is odd}. \end{cases} \tag{44}$$

Note that all contributions in (34) now vanish when k is even, due to symmetry as in (43). This is to be expected based on the analysis of the truncation error in (29).

Numerical Example

We will now illustrate the monotonicity of the ENO discretization for the 1D heat equation, (23), with initial condition $u(x, 0) = \delta(x)$, where δ is the Dirac delta function, and homogeneous Neumann boundary conditions. A fourth-order spatial discretization is combined with a third-order Runge-Kutta TVD (RK-TVD-3) time discretization. The spatial grid consists of 21 nodes and comprises the node $x_j = 0$. This is important, since we discretize the initial conditions conservatively as follows

$$u_j^0 = \begin{cases} 0 & x_j \neq 0, \\ \frac{1}{\Delta x} & x_j = 0. \end{cases} \tag{45}$$

The time step is $\Delta t = (\sigma/2)\Delta x^2$, where monotonicity of RK-TVD-3 requires $\sigma \leq 1$, see [17]. We take $\sigma = 1/2$. Results for both the fourth-order central and the ENO scheme of

Proposition 2.2 are depicted in Fig. 1. The central scheme produces oscillatory and negative values for u, whereas the results for the ENO scheme are non-oscillatory and non-negative.

2.2.2 Non-constant Heat Coefficient

Diffusion formulated simply as a second-order derivative appears in many of our applications, as we will see later on. However, there are circumstances where we need to discretize

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} k(x) \frac{\partial u}{\partial x}. \tag{46}$$

One could consider a coordinate transformation and bring this equation into the previously discussed form. Bearing in mind that we need an equidistant mesh, now both in the original (for convection) and transformed coordinates, this approach is not appealing. The discretization proposed here is formulated in the following proposition:

Proposition 2.5 *An essentially non-oscillatory discretization of the heat equation*, (46), *which is* r^{th}*-order accurate in space*, *assuming sufficiently smooth solutions*, *is obtained by a numerical method which is based on a Runge-Kutta type TVD time discretization and a discretization of* (46) *in two steps*, *by firstly computing*

$$f = -k \frac{\partial u}{\partial x}, \tag{47}$$

up to r^{th}*-order accuracy and then*[3]

$$\frac{\partial u}{\partial t} = -\frac{\partial f}{\partial x}. \tag{48}$$

Here,

$$f(x_{j+\frac{1}{2}}) = -k(x_{j+\frac{1}{2}}) \frac{h_{j+1} - h_j}{\Delta x}, \tag{49}$$

where

$$h_j = u(x_{\ell^{(1)}}) + \sum_{k=2}^{r} \frac{u[x_{\ell^{(k)}}, \ldots, x_{\ell^{(k)}+k-1}]}{k} \frac{\mathrm{d}}{\mathrm{d}x} \prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k-1} (x - x_{m-\frac{1}{2}}) \Bigg|_{x=x_j}. \tag{50}$$

For the derivative of u_j *we switch the indices by one-half*:

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = -\frac{g_{j+\frac{1}{2}} - g_{j-\frac{1}{2}}}{\Delta x}, \tag{51}$$

where

$$g_{j+\frac{1}{2}} = f(x_{\ell^{(1)}+\frac{1}{2}}) + \sum_{k=2}^{r} \frac{f[x_{\ell^{(k)}+\frac{1}{2}}, \ldots, x_{\ell^{(k)}+k-\frac{1}{2}}]}{k} \frac{\mathrm{d}}{\mathrm{d}x} \prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k-1} (x - x_m) \Bigg|_{x=x_j+\frac{1}{2}}. \tag{52}$$

Furthermore, *we take*:

[3]This methodology has some similarities to the variational Discontinuous Galerkin technique for second-order equations in [7].

1. $\ell^{(1)} = j$,
2. *The smoothest possible interpolation schemes for* $r > 1$ *for the computation of* h_j *in* (50), *compare with* (19) *and* (35), *i.e.*

$$\ell^{(k+1)} = \begin{cases} \ell^{(k)} - 1 & |u[x_{\ell^{(k)}-1}, \ldots, x_{\ell^{(k)}+k-1}]| \\ & \quad \leq |u[x_{\ell^{(k)}}, \ldots, x_{\ell^{(k)}+k}]|, \\ \ell^{(k)} & \textit{otherwise}, \end{cases} \tag{53}$$

and, *similarly*,

$$\ell^{(k+1)} = \begin{cases} \ell^{(k)} - 1 & |f[x_{\ell^{(k)}-\frac{1}{2}}, \ldots, x_{\ell^{(k)}+k-\frac{1}{2}}]| \\ & \quad \leq |f[x_{\ell^{(k)}+\frac{1}{2}}, \ldots, x_{\ell^{(k)}+k+\frac{1}{2}}]|, \\ \ell^{(k)} & \textit{otherwise}, \end{cases} \tag{54}$$

for the computation of $g_{j+\frac{1}{2}}$ *in* (52).

Outline of Proof:
We maintain the requirements:

1. The standard *monotone* three-point central scheme, similar to (25), is recovered for $r = 1$,
2. The scheme is *essentially non-oscillatory* for $r > 1$.

Instead of imposing these demands on the discretization in whole, we will impose them to (47) and (48) separately.

The second demand requires the ENO reconstruction. We apply the algorithm which is available for the discretization of the convection operator to compute the first-order derivatives. However, it can readily be seen that this easily violates our first demand. To meet the first demand, we have to employ *symmetric differences for* $r = 1$ and use *staggered locations for* f to avoid checker-boarding. Computing f goes in much the same way as before, see (11), leading to (49). Adapting (17) to our needs by evaluating it at x_j and substituting u for f gives an r^{th}-order approximation for $f(x_j)$ by employing (50), under the usual conditions. The central scheme is retained for $r = 1$ by setting $\ell^{(1)} = j$.

An r^{th}-order accurate computation of (48) can easily be derived from (49) and (50) by switching the indices by one-half:

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = -\frac{g_{j+\frac{1}{2}} - g_{j-\frac{1}{2}}}{\Delta x},$$

where $g_{j+\frac{1}{2}}$ is as defined in (52), and using $\ell^{(1)} = j$ to get the central scheme for $r = 1$. □

Remark 2.6 The $k = 2$ term in the summation of (50) cancels in the same manner as described by Remark 2.3:

$$\frac{\mathrm{d}}{\mathrm{d}x}\,(x - x_{j-\frac{1}{2}})(x - x_{j+\frac{1}{2}})\Big|_{x=x_j} = 0 \tag{55}$$

and similarly for (52).

Remark 2.7 If we would have repeated our one-step approximation (instead of the two-step approximation in Proposition 2.5), we would have obtained (10):

$$\frac{\partial u}{\partial t} = -\frac{h(x + \frac{1}{2}\Delta x) - h(x - \frac{1}{2}\Delta x)}{\Delta x},$$

where now, as in (8) and (36), h is defined by

$$\mathcal{A}h = -k(x)\frac{\partial u}{\partial x}. \tag{56}$$

If we would use (37), i.e. $h(x) = \mathrm{d}^2H/\mathrm{d}x^2(x)$, we would obtain (omitting the time-dependency of u)

$$\frac{\frac{\mathrm{d}H}{\mathrm{d}x}(x+\frac{1}{2}\Delta x) - \frac{\mathrm{d}H}{\mathrm{d}x}(x-\frac{1}{2}\Delta x)}{\Delta x} = -k(x)\frac{\partial u}{\partial x}(x) \tag{57}$$

whose right-hand side *cannot* be integrated in general form as in (39) because of the non-constant k. If we would choose, instead of (37),

$$h(x) = \frac{\mathrm{d}}{\mathrm{d}x}k(x)\frac{\mathrm{d}H}{\mathrm{d}x}(x), \tag{58}$$

we would obtain

$$\frac{k(x+\frac{1}{2}\Delta x)\frac{\mathrm{d}H}{\mathrm{d}x}(x+\frac{1}{2}\Delta x) - k(x-\frac{1}{2}\Delta x)\frac{\mathrm{d}H}{\mathrm{d}x}H(x-\frac{1}{2}\Delta x)}{\Delta x} = -k(x)\frac{\partial u}{\partial x}(x), \tag{59}$$

which is again *not* integrable in general form. The divided difference tables of H can not be computed from the divided differences of u as easily as before and therefore this approach isn't appealing.

2.2.3 Cross-Derivatives

The last step to make is to extend the methodology to multiple dimensions, i.e.

$$\frac{\partial u}{\partial t} = \nabla \cdot K(u,t)\nabla u, \tag{60}$$

which is the diffusion term in (2), where K is a matrix, for example related to the correlation between stochastic processes.[4] The diagonal terms can all be placed in the previously discussed form, $\partial(k(x)\partial u/\partial x)/\partial x$. So, without loss of generality, we only need to consider the cross-derivative terms, as expressed by

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}K_{xy}(x,y)\frac{\partial u}{\partial y}. \tag{61}$$

Our discretization of choice is formulated in the following proposition:

Proposition 2.8 *An essentially non-oscillatory discretization of* (61)*, which is* r^{th}*-order accurate in space, based on sufficiently smooth solutions, is obtained by the numerical method of* (51), (52) *and* (54)*, combined with a Runge-Kutta type TVD time discretization.*

Outline of Proof:
Discretizing (61), we can first compute

$$f = -K_{xy}\frac{\partial u}{\partial y}, \tag{62}$$

[4]For example, see (80).

up to r^{th}-order accuracy and then (again) consider (48):

$$\frac{\partial u}{\partial t} = -\frac{\partial f}{\partial x}.$$

The computation of $\partial f/\partial x$, has already been discussed in Proposition 2.5. To compute $\partial f/\partial x|_{(x_j,y_l)}$, inspection of (52) shows that we need $\partial u/\partial y$ at *staggered* locations $(x_{j+\frac{1}{2}}, y_l)$, whereas u-data are known at locations (x_j, y_l). If we want to apply one-dimensional algorithms, we first have to compute $\partial u/\partial y|_{(x_j,y_l)}$ and then interpolate with r^{th}-order accuracy to the staggered locations. For both, the computation of the y-derivatives and the interpolation, the following conditions are again imposed:

1. The standard schemes are recovered for $r = 2$,
2. The schemes are *essentially non-oscillatory* for $r > 2$.

For the computation of the y-derivative we use the one-dimensional method D:

$$\frac{\partial u}{\partial y}(x_j, y_l) = D(u(x_j, \cdot); l), \tag{63}$$

which can be derived without much effort from (11) and (17):

$$D(\phi; l) = \frac{h_{l+\frac{1}{2}} - h_{l-\frac{1}{2}}}{\Delta y}, \tag{64}$$

for some variable $\phi(y)$. Now, $h_{l+\frac{1}{2}}$ is defined as:

$$h_{l+\frac{1}{2}} = \phi(y_{\ell^{(1)}}) + \sum_{k=2}^{r} \frac{\phi[y_{\ell^{(k)}}, \ldots, y_{\ell^{(k)}+k-1}]}{k} \prod_{\substack{m=\ell^{(k-1)} \\ m\neq l+1}}^{\ell^{(k-1)}+k-1} (y_{l+\frac{1}{2}} - y_{m-\frac{1}{2}}). \tag{65}$$

This gives an r^{th}-order accurate approximation under the usual smoothness conditions.

The adjustment we need to make is the selection of $\ell^{(1)}$ and $\ell^{(2)}$, so that the standard central scheme is recovered for $r = 2$, which is the first demand. So, we have to set $\ell^{(1)} = \ell^{(2)} = k$. The other $\ell^{(k)}, k > 2$ are as in (19) and such that the smoothest possible interpolation polynomial is obtained, which gives the choice:

$$\ell^{(l+1)} = \begin{cases} \ell^{(l)} - 1 & |u[x_{\ell^{(l)}-1}, \ldots, x_{\ell^{(l)}+l-1}]| \\ & \quad \le |u[x_{\ell^{(l)}}, \ldots, x_{\ell^{(l)}+l}]|, \\ \ell^{(l)} & \text{otherwise.} \end{cases} \tag{66}$$

For the interpolation of the y-derivatives to the staggered locations, a one-dimensional method is employed:

$$\frac{\partial u}{\partial y}(x_{j+\frac{1}{2}}, y_l) = I\left(\frac{\partial u}{\partial y}(\cdot, y_l); j + \frac{1}{2}\right). \tag{67}$$

The interpolation I, for some $\psi(x)$, is defined as:

$$I\left(\psi; j + \frac{1}{2}\right) := \psi(x_{\ell^{(1)}}) + \sum_{k=2}^{r} \psi[x_{\ell^{(k)}}, \ldots, x_{\ell^{(k)}+k-1}] \prod_{m=\ell^{(k-1)}}^{\ell^{(k-1)}+k-2} (x_{j+\frac{1}{2}} - x_m). \tag{68}$$

It is an r^{th}-order accurate approximation, under the same conditions as before, with $\ell^{(1)} = \ell^{(2)} = j$ and the selection of the leftmost nodes as just prescribed for method D:

$$\ell^{(l+1)} = \begin{cases} \ell^{(l)} - 1, & |\psi[x_{\ell^{(l)}-1}, \ldots, x_{\ell^{(l)}+l-1}]| \\ & \quad \leq |\psi[x_{\ell^{(l)}}, \ldots, x_{\ell^{(l)}+l}]|, \\ \ell^{(l)}, & \text{otherwise.} \end{cases} \tag{69}$$

This concludes the proof of the proposition.

Symbolically, we can write:

$$f(x_{j+\frac{1}{2}}, y_l) = -K_{xy}(x_{j+\frac{1}{2}}, y_k)\ I\left(\{D(u(x_m, \cdot); l), m = \cdot\}; j + \frac{1}{2}\right), \tag{70}$$

with method D from (64) and interpolation I as in (68). □

2.3 Numerical Example

The Laplace equation serves to assess the accuracy of the spatial discretization. Given a prescribed function v, find u that satisfies

$$\Delta u(\mathbf{x}) = \Delta v(\mathbf{x}), \quad \mathbf{x} = (x, y)^t \in \Omega, \tag{71}$$

with $\Omega = \{\mathbf{x} \in \mathbb{R}^2 | x \leq 0 \wedge y \geq 0 \wedge \frac{1}{2} \leq \|\mathbf{x}\| \leq 1\}$, i.e. a quarter of an open disc, and $v(\mathbf{x}) = \sin(4\pi x)$. The boundary conditions are such that u equals v at $\delta\Omega$, i.e. $u(\mathbf{x}) = v(\mathbf{x})$, $\mathbf{x} \in \partial\Omega$. Of course, the error is just $e = v - u$. We employ a coordinate transformation from the physical domain to the computational domain $\{(\xi, \eta) \in [0, 1]^2\}$ and obtain

$$\hat{\nabla} \cdot K(\xi, \eta)\hat{\nabla} u(\mathbf{x}(\xi, \eta)) = \Delta u(\mathbf{x}(\xi, \eta)), \tag{72}$$

where $\hat{\nabla} = (\partial/\partial\xi, \partial/\partial\eta)^t$ and

$$K(\xi, \eta) = \frac{1}{d}\begin{pmatrix} x_\eta^2 + y_\eta^2 & -(x_\xi x_\eta + y_\xi y_\eta) \\ -(x_\xi x_\eta + y_\xi y_\eta) & x_\xi^2 + y_\xi^2 \end{pmatrix}, \tag{73}$$

in which the subscripts indicate partial derivation and $d = x_\xi y_\eta - x_\eta y_\xi$. A uniform Cartesian mesh consisting of $(N + 1) \times (N + 1)$ nodes, including the boundary nodes, is employed. Equation (72) is discretized with the ENO methodology as described in the Sect. 2.2. The discretization schemes are nonlinear and, for simplicity in this model test case, we linearize them by using v for the selection of the leftmost nodes in (53) and (54) and (66) and (69). This allows for a direct solve of the linear system that now has arisen.[5] Alternatively, as exact solutions are not available in practical applications, Picard linearisation can easily be applied to deal with this nonlinear discretization.

Remark 2.9 It is worth noting that we compute the mesh derivatives x_ξ, etc. discretely with the ENO methodology as well. As a matter of fact, data of map $\mathbf{x}(\xi, \eta)$ are only prescribed at nodes (ξ_j, η_l) and the mesh derivatives at the staggered locations are computed in exactly the same manner as the fluxes f in Propositions 2.5 and 2.8.

[5]Note that we do not have time dependency in this test example.

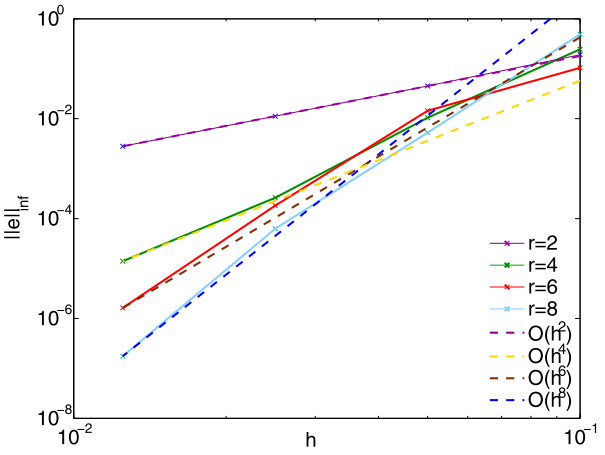


Fig. 2 Grid convergence for the Laplace test-case

The error $e = v - u$, measured in the L_∞-norm is plotted in Fig. 2. The dashed lines in the figure shows the theoretical second-, fourth-, sixth- and eighth-order convergence curve resp. ($O(h^2)$ etc. in the figure), the straight lines show the error convergence achieved by the ENO schemes with different values for discretization order r. The figure shows that the grid convergence with respect to mesh widths $h = \frac{1}{N}$, for different discretization orders r is in accordance with the theoretically expected convergence behaviour. The grid convergence matches the discretization order.

2.4 Butterfly Spread

The first real test-case for our ENO scheme is an application from Mathematical Finance. The problem is formulated as [16]:

$$\frac{\partial u}{\partial t} - rx\frac{\partial u}{\partial x} = \max_{\sigma\in\{\sigma_{min},\sigma_{max}\}}\left(\frac{1}{2}(\sigma x)^2\frac{\partial^2 u}{\partial x^2}\right) - ru, \quad x \in (0, x_R), t \in [0, T], \tag{74}$$

with boundary conditions

$$\frac{\partial u}{\partial t}(0,t) = -ru, \qquad \frac{\partial u}{\partial t}(x_R,t) = 0, \tag{75}$$

and initial condition

$$u(x,0) = \max(x - K_1, 0) - 2\max\left(x - \frac{1}{2}(K_1 + K_2), 0\right) + \max(x - K_2, 0). \tag{76}$$

An uncertain volatility σ is prescribed by

$$\sigma = \begin{cases} \sigma_{min} & \frac{\partial^2 u}{\partial x^2} > 0, \\ \sigma_{max} & \frac{\partial^2 u}{\partial x^2} \le 0. \end{cases} \tag{77}$$

We will not discuss the background of the financial problem but focus on the numerical scheme to solve (74). Pooley et al. [16] show that a non-monotone scheme can lead to incorrect solutions. They utilise a locally first-order upwind-like finite difference discretization

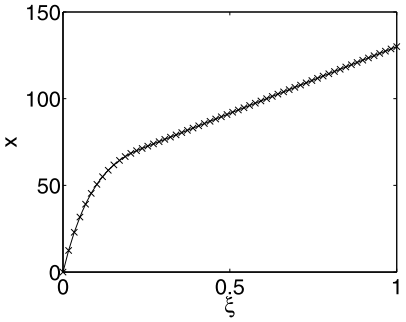


Fig. 3 Coordinate transformation for the Butterfly Spread test-case; $N = 60$

for the convection operator and the common second-order discretization for the diffusion operator. A monotone scheme is obtained, on the condition of an implicit time integration, by either fully implicit or Crank-Nicholson discretizations. This leads to a nonlinear iterative time integration.

To accommodate a non-uniform mesh, we here use a coordinate transformation $x(\xi), \xi \in [0, 1]$.

Approximately 75% of the nodes are uniformly distributed between $x = x_L$ and x_R, $0 < x_L < x_R$, and the remaining nodes are exponentially distributed between $x = 0$ and x_L.

With n_L the number of points in the stretched region, we use the transformation:

$$x_i = \frac{1 - \hat{\alpha}^{i-1}}{1 - \hat{\alpha}^{n_L - 1}} \cdot x_L, \quad 1 \le i \le n_L.$$

Here, the stretching parameter $\hat{\alpha}$ is defined such that

$$\hat{\alpha} \cdot \left(x_{n_L} - x_{n_L - 1}\right) = \frac{x_R - x_L}{n_x - n_L},$$

with n_x the total amount of grid points; $d_L := (x_{n_L} - x_{n_L - 1})$ is the mesh width at the left side of x_L, and $d_R := (x_R - x_L)/(n_x - n_L)$ is the right side (uniform) mesh width. We require $\hat{\alpha} = d_R/d_L$, so that the grid is smooth at x_L. Parameter $\hat{\alpha}$ is determined by Newton's method. An example is given in Fig. 3.

Substitution in (74) yields

$$\frac{\partial u}{\partial t} - r\frac{x}{x'}\frac{\partial u}{\partial \xi} = \max_{\sigma \in \{\sigma_{min}, \sigma_{max}\}} \frac{1}{2}(\sigma x)^2 \frac{1}{x'}\frac{\partial}{\partial \xi}\left(\frac{1}{x'}\frac{\partial u}{\partial \xi}\right) - ru, \quad \xi \in (0, 1), t \in [0, T]. \tag{78}$$

The following data are used: $T = 0.25$, $r = 0.1$, $K_1 = 90$, $K_2 = 110$, $\sigma_{min} = 0.15$, $\sigma_{max} = 0.25$ and we take $x_L = 70$ and $x_R = 130$.

The derivative $\partial u/\partial \xi$ is discretized as explained in Sect. 2.1.1 by setting $f = u$ in (13). Note that the upwind direction in (18) is determined by setting $a_{j+\frac{1}{2}} = -r(x/x')_{j+\frac{1}{2}}$. The discretization of $\partial/\partial\xi(1/x'\partial u/\partial\xi)$ is as described in Proposition 2.5. To ensure stability, we take for Δt:

$$\Delta t = \frac{1}{2}\frac{\Delta x^2}{\frac{1}{2}(\sigma_{max} x_R)^2}. \tag{79}$$

Since Δt is proportional to Δx^2, we set the order n of the RK-TVD time integration equal to the square root of the order r of the spatial discretization. The convergence results of

Table 1 Convergence results of $u(100, T)$

N	$r=6, n=3$	$r=4, n=2$	$r=2, n=1$	Pooley et al. [16]
60	2.3075	2.3073	2.2997	2.3501
120	2.2967	2.2966	2.2945	2.3250
240	2.2974	2.2974	2.2969	2.3116
480	2.2976	2.2976	2.2975	2.3047
960	2.2977	2.2977	2.2976	2.3012
extr.				2.2977

$u(100, T)$ are shown in Table 1. In this table, by 'extr.' the extrapolated values from $N = 240$, 480 and 960, by means of a quadratic (repeated) Richardson extrapolation, is meant. It is reasonable to assume that the numerical solution converges to the same value as obtained by Pooley et al. in [16]. The higher order schemes show fast convergence and reach highly satisfactory approximations for $N = 120$, although the differences in this test are relatively small.

3 The Dike Height Optimisation Problem

In this section we explain in detail the formulation of an optimal dike control problem.

3.1 Model Equations

The future expected costs are comprised of the costs due to flooding, the investment costs of dike level increases and the terminal costs. There were three variables in van Dantzig's original model [19] that set up the state space, the uniform dike height, uniform water level and economic value at risk behind the dike. Extending the model by assuming stochastic behaviour in continuous time, the system dynamics can be put as

$$\mathrm{d}X = a(X(t), t)\,\mathrm{d}t + m(X(t), t)\,\mathrm{d}Z, \tag{80}$$

where $X(t) \equiv x(t) = (x_1(t), x_2(t), x_3(t))^t$ is the state vector. Here x_1 represents the dike height, x_2 the water level and x_3 the economic value at risk, respectively. The deterministic part of the evolution of X is expressed by the drift $a(X, t) = (a_1(X, t), a_2(X, t), a_3(X, t))^t$, while Z expresses Brownian motion and $m(X(t), t) = diag(m_1(X(t), t), m_2(X(t), t), m_3(X(t), t))$ represents the covariance matrix here.

It is important to understand that x_2 represents an *average* water level, see [8] and [20]. De Haan [8] remarks that flooding occurs when high tide is accompanied by a storm. He applies extreme-value theory and connects the occurrence of the extreme event to a Poisson point process. Van Noortwijk et al. [20] use a Poisson process to generate the extreme event, i.e. the flooding, and adopt a peaks-over-threshold distribution for the distribution of the jump magnitude. We will employ these ideas by assuming that the absolute water level, $w(t)$, is a summation of the average water level, $x_2(t)$, and a jump, $\mathcal{J}(t)$:

$$w(t) = x_2(t) + \mathcal{J}(t), \tag{81}$$

see Fig. 4 for an example.

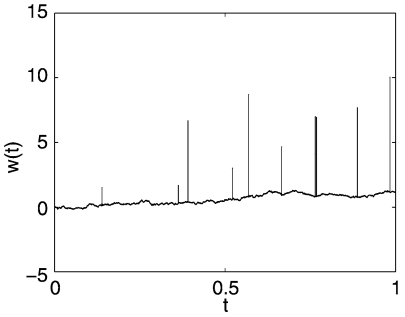


Fig. 4 Example of the composition of the water level by an average level $x_2(t)$ and a jump $\mathcal{J}(t)$. The jump intensity of the Poisson process is $\lambda = 1/10$

Having defined the water level in such a manner, it is possible to express the discounted future losses as:

$$\int_t^T e^{-r(s-t)} x_3(s)\, l_p(x_2(s) + \mathcal{J}(t) - x_1(s))\, \mathrm{d}s,$$

where T is the time horizon, r is a deterministic discount rate and l_p, defined by

$$l_p(y) = \max(1 - e^{-\lambda_p y}, 0), \tag{82}$$

measures the fraction of the economic value $x_3(s)$ that is lost when the absolute water level $w(s)$ exceeds the dike height $x_1(s)$ by an amount $w(s) - x_1(s)$. Several parameters, appearing in the definition of the problem (like λ_p here), are given in Table 2.

The terminal costs at time T, named b_1, are set as

$$b_1(x) = \int_T^\infty e^{-r(s-T)}\, \lambda\, x_3(s)\, \beta(x_1(s) - x_2(s))\, \mathrm{d}s, \tag{83}$$

in which

$$\beta(h) = \int_{-\infty}^{\infty} l_p(y-h)\ f(y)\, \mathrm{d}y. \tag{84}$$

The statistics for the extreme water levels, expressed by the probability density function f, are based on annual data in a discrete-time model, so if we are to use it in our model, we must set the intensity of the Poisson process to once per year, i.e. $\lambda = 1\ \mathrm{yr}^{-1}$, with λ the jump intensity, or the expected frequency of the jumps. $f(y)$ is the probability density function of the jumps in the Poisson process,

$$f(y) = k_1 e^{-k_1(y-k_2)} e^{-e^{-k_1(y-k_2)}} \tag{85}$$

(k_1 and k_2 constants, Table 2).

The construction costs of the dikes, $b_2(x, u)$ in (87), are defined as:

$$b_2(x, u) = k_f + k_u(u^2 \tan(\phi) + u\,(2x_1 \tan(\phi) + \hat{w})), \tag{86}$$

(k_f, k_u, ϕ and $\hat{w}$ constants, Table 2).

Now, the total discounted expected costs J, can be expressed as:

$$J(x, t, u) = \mathbb{E}_x \left\{ \int_t^T e^{-r(s-t)} x_3(s) l_p(x_2(s) + \mathcal{J}(s) - x_1(s))\, \mathrm{d}s \right.$$

$$+\sum_{t\le t_k<T} e^{-r(t_k-t)}b_2(x(t_k-),u_k(x(t_k-)))+e^{-r(T-t)}b_1(x(T))\Bigg\}, \tag{87}$$

where $\mathbb{E}_x$ is the expectation conditioned on x and $u(t)$ denotes the increase in dike height, which acts as an input to the control problem.

The terminal condition $V(x,T)=b_1(x)$ expresses the total discounted expected costs at the time horizon T and is obtained by assuming that the water level and dike height remain constant after the time horizon [12].

The dike increase is imposed at a sequence of intervention times t_k, i.e.

$$x_1(t_k)=x_1(t_k-)+u_k(x(t_k-)). \tag{88}$$

The t_k- in (88) indicates that stochastic process X is càdlàg, i.e. right continuous with left limits.

The optimal cost-to-go function is

$$V(x,t)=\inf_{\hat{u}} J(x,t,\hat{u}). \tag{89}$$

The optimisation in (89) can be carried out over a discrete set $\mathcal{U}$, which is computationally the least demanding, or a continuous set. This will be outlined in the subsections to follow.

By a dynamic programming argument, for details see, for example, [3], and applying Itô's formula, see e.g. [15], the following differential equation is obtained in the case that it is *not* optimal to increase the dike height:

$$\begin{cases} 0=\frac{\partial V}{\partial t}(x,t)+\mathcal{L}V(x,t)-rV(x,t)+\lambda x_3\beta(x_1-x_2), \\ V(x,t-)\ge \inf_{\hat{u}} V(x+(\hat{u},0,0)^t,t)+b_2(x,\hat{u}), \end{cases} \tag{90}$$

where

$$\mathcal{L}V(x,t)=a^t(x,t)\frac{\partial V}{\partial x}(x,t)+\frac{1}{2}\operatorname{trace}\left(m\,m^t\frac{\partial^2 V}{\partial x^2}(x,t)\right), \tag{91}$$

for $t\in(t_{k-1},t_k)$. The running costs are represented by $\lambda\beta(x_1-x_2)x_3$ in (90), where β is defined in (84).

When it is optimal to increase dike heights, we have:

$$V(x,t-)=\inf_{\hat{u}} V(x+(\hat{u},0,0)^t,t)+b_2(x,\hat{u}), \tag{92}$$

for $t\in\{\ldots,t_{k-1},t_k,\ldots\}$.

The combination of (90) and (92) leads to a Hamilton-Jacobi-Bellman (HJB) formulation.

3.2 Impulse Control

Section 2.1 described the numerical approach of the uncontrolled part of the problem, by means of the higher-order explicit finite differences with the ENO scheme. We will now discuss the optimal control, i.e. finding the dike reinforcements u_k at intervention times t_k that minimise the total expected future costs, see (92). As mentioned before, the intervention times t_k are fixed, in practice annually, and the dike increases are instantaneously. This

means that the total expected costs $V(x,t_k)$ from (89), just after the possible dike increase at t_k, are evaluated by integrating the equality of (90) backwards in time from $t_{k+1}-$ to t_k by the methods just described. Arriving at intervention time t_k, one has to decide on the optimal dike increase, u_k, to obtain optimal costs $V(x,t_k-)$. The optimal control is computed from (92), i.e.

$$u_k(x) = \arg\inf_{\hat{u}\in\mathcal{U}} V(x+(\hat{u},0,0)^t,t) + b_2(x,\hat{u}). \tag{93}$$

3.2.1 Discrete Optimisation

Assume that, for computational efficiency, we take a *discrete set* of possible inputs $\mathcal{U}$, i.e. $\mathcal{U} = \{0, \Delta u, 2\Delta u, \ldots\}$.The optimal costs just before the possible dike increase are then

$$V(x,t-) = \inf_{\hat{u}\in\mathcal{U}} V(x+(\hat{u},0,0)^t,t) + b_2(x,\hat{u}). \tag{94}$$

Remark 3.1 For computational efficiency, it is advantageous to have all $x_1+\hat{u}$ coincide with the grid nodes of x_1. This prevents the need to interpolate from the nodal data to $x_1+\hat{u}$ for all discrete x_1 and all $\hat{u}$. This can be achieved by taking $\Delta x_1 = \Delta\hat{u}/m$, where m is an integer.

Remark 3.2 Assume that we have computed a numerical solution of the problem and we want to make a realisation $X(t)$ by integrating the system dynamics, (80) *forward* in time, starting from some initial data. We arrive just before intervention time t_k, with state $X(t_k-)$. The question is how to compute the input u_k. Since $u_k\in\mathcal{U}$ is discrete, we can not simply interpolate u_k from the nodal data to $x_1(t_k-)$ as in (88). Instead, we have the data from the two neighbouring nodes, called u_L and u_R for convenience. Then, we compute

$$u_k(X(t_k-)) = \arg\inf_{\hat{u}\in\{u_L,u_R\}} V(X(t_k-)+(\hat{u},0,0)^t,t_k) + b_2(X(t_k-),\hat{u}), \tag{95}$$

where the $V(\cdot,t_k-)$ is approximated at $X(t_k-)+(\hat{u},0,0)^t$ with an r^{th}-order accurate ENO interpolation.

3.2.2 Continuous Optimisation

There may be a need to optimise over a *continuous set* of inputs. We will use investment costs of the following form:

$$b_2(x,u) = \begin{cases} 0 & u=0, \\ b_2^+(x,u) & u>0, \end{cases} \tag{96}$$

where b_2^+ is a smooth function, such as a polynomial expression and $b_2^+(x,0)\neq 0$. We therefore first compute u_k^+ in the reduced set $\mathcal{U}\setminus\{0\}$:

$$u_k^+(x) = \arg\inf_{\hat{u}\in\mathcal{U}\setminus\{0\}} V(x+(\hat{u},0,0)^t,t) + b_2^+(x,\hat{u}) \tag{97}$$

and then u_k as

$$u_k(x) = \arg\inf_{\hat{u}\in\{0,u_k^+(x)\}} V(x+(\hat{u},0,0)^t,t) + b_2(x,\hat{u}), \tag{98}$$

using an r^{th}-order accurate ENO interpolation for $V(x + (u_k^+(x), 0, 0)^t, t)$. Since $b_2^+(x, u)$ is continuous and continuously differentiable with respect to u, u_k^+ is the solution $\hat{u}$ of

$$\frac{\partial}{\partial \hat{u}} \left\{ V(x + (\hat{u}, 0, 0)^t, t) + b_2^+(x, \hat{u}) \right\} = 0, \tag{99}$$

assuming that V is also continuously differentiable with respect to u. This results in the following condition for u_k^+:

$$\frac{\partial V}{\partial x_1}(x + (u_k^+, 0, 0)^t, t) + \frac{\partial b_2^+}{\partial u}(x, u_k^+) = 0. \tag{100}$$

We solve this equation with a Secant method. The first term, $\partial V/\partial x_1$, has to be computed from nodal values, $V(x_j)$, to arbitrary locations with r^{th}-order accuracy. The Secant method requires continuity of $\partial V/\partial x_1$, so we can not simply employ an ENO reconstruction of V and take its derivative. Instead, we firstly compute $\partial V/\partial x_1$ at a *staggered* location similar to (49) and (50) of Proposition 2.5. Then, we approximate $\partial V/\partial x_1$ at the desired location with an r^{th}-order accurate ENO interpolation. To ensure that we find the global extrema, we take the first iterate in the Secant algorithm from a discrete optimisation, where we set $\Delta u = \Delta h$.

Remark 3.3 Once the control law is computed, consider a realisation $X(t)$. The input $u_k(X(t_k-))$ is computed as follows. Due to discontinuity of b_2 at $u = 0$, u_k itself is discontinuous. Therefore, first $u_k^+(X(t_k-))$ is computed from nodal values of u_k^+ by means of an r^{th}-order accurate ENO interpolation. Thereafter $u_k(X(t_k-))$ is determined by using (98) and setting $x = X(t_k-)$.

3.3 Dike Optimisation; Model I

The test-case here is the dike optimisation problem as described in Sect. 3.1. We consider a Dutch island, and first take data, where applicable, from the discrete-time model of [12]. The system dynamics are deterministic in this test case, translating to $a = (0, \mathrm{d}w/\mathrm{d}t(t), \alpha_3 x_3)^t$ and $m = \mathbf{0}$ in (80), where $w(t)$ is the predicted average water level and α_3 is the economic growth factor. Note that w is now redefined to represent the average water level and should not be confused with its previous definition in (81).

Substitution in (84) yields for the terminal costs

$$b_1(x) = \frac{\lambda\beta(x_1(T) - x_2(T))}{r - \alpha_3} x_3(T). \tag{101}$$

The open parameters in the problem definition are defined in Table 2. The average water level is assumed piecewise linear between the data in Table 3 and the initial water level is $w(0) = 0$ cm. We can reduce the dimension of the problem by setting

$$h = x_1 - x_2, \tag{102}$$

$$\tau = T - t, \tag{103}$$

$$V(x, t) = e^{x_3 W(h,\tau)}. \tag{104}$$

Note that h can be understood as the relative dike height and the time has been reversed by introducing τ, left continuous with right limits, for convenience. Substitution of the system

Table 2 Parameter values for the dike height problem

Parameter	Value	Details
α_3	0.025	econ. growth
$x_3(0)$	34×10^3 MEUR	init. value
k_1	8.16299×10^{-2} cm^{-1}	(85), see [12]
k_2	1.88452×10^2 cm	(85), see [12]
λ_p	1.2×10^{-2} cm^{-1}	(82)
r	0.05	(87)
T	300 yr	
k_f	22.975 MEUR	(86)
k_u	1.921×10^{-4} MEUR/cm^2	(86)
ϕ	1.25	(86)
$\hat{w}$	500 cm	(86)

Table 3 Predicted average water level rise $w(t) - w(0)$

t [yr]	0	50	100	150	200	250	300
$w(t) - w(0)$ [cm]	0	25	60	105	140	165	180

dynamics in the uncontrolled part of the governing equations, (90) yields

$$\begin{cases} \dfrac{\partial W}{\partial \tau} = -\dfrac{\mathrm{d}w}{\mathrm{d}t}(T-\tau)\dfrac{\partial W}{\partial h} - (r-\alpha_3) + \dfrac{\lambda\beta(h)}{e^W}, \\ W(h,\tau+) \geq \ln\left(\inf\limits_{\hat{u}\in\mathcal{U}}\left[e^{W(h+\hat{u},\tau)} + \dfrac{b_2((w(T-\tau)+h, w(T-\tau), x_3(T-\tau))^t, \hat{u})}{x_3(T-\tau)}\right]\right) \end{cases} \tag{105}$$

with $x_2 = w$, subject to initial condition (101)

$$W(h,0) = \max\left(\ln\left(\frac{\lambda\beta(h)}{r-\alpha_3}\right), \epsilon\right), \tag{106}$$

where we choose $\epsilon = 10^{-16}$.

The integral in (84) is computed numerically with an appropriate integration rule, changing the integration interval to $[y_{min}, y_{max}]$ and taking N_y intervals, where $y_{max} = -y_{min} = 500$ cm and $N_y = 1000$.

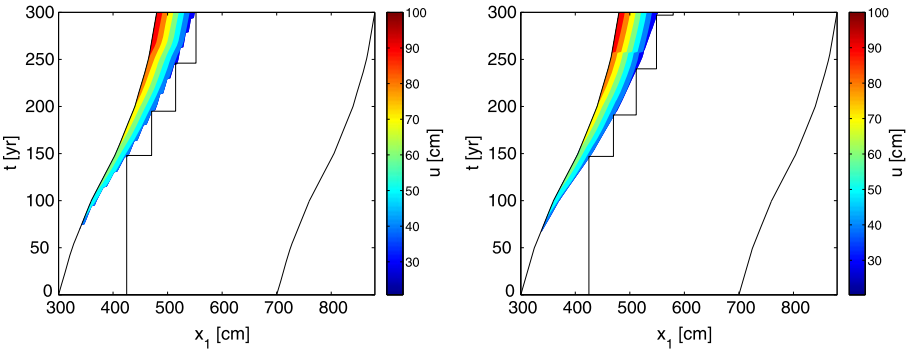
We take $h \in [h_L, h_R]$, with $h_L = 300$ cm, $h_R = 700$ cm. We will take N_h nodes and vary it to show grid convergence. Since $\frac{\mathrm{d}w}{\mathrm{d}t} \geq 0$, it is sufficient to apply the following (somewhat artificial) boundary condition:

$$\frac{\partial W}{\partial \tau}(h_L,\tau) = -(r-\alpha_3) + \frac{\lambda\beta(h_L)}{e^{W(h_L,\tau)}}. \tag{107}$$

The input u is assumed discrete in [12]: $\mathcal{U} = \{0, 5, 10, \ldots\}$ cm. We will adopt these values in case of discrete optimisation. The possible control times τ_k are $\tau_k \in \{0, 1, 2, \ldots, T\}$ yr. The discretization is 4$^{\text{th}}$-order accurate in space and 3$^{\text{rd}}$-order in time. The solution procedure is as explained earlier. Explicit finite differencing based on the TVD-Runge-Kutta scheme is used for (107). Furthermore, ENO discretization is used for

Table 4 Grid resolutions and computing times for the dike-optimisation test-case; both discrete and continuous optimisation

grid	N_h	Δh [cm]	N_t	Δt [yr]	comp. time [s]	
					discr. opt.	cont. opt
1	81	5	300	1	7.20	18.54
2	161	2.5	600	0.5	11.37	24.77
3	321	1.25	1200	0.25	22.27	42.95
4	641	0.625	2400	0.125	54.88	112.06
5	1281	0.3125	4800	0.0625	159.85	374.41



(a) grid 1 (b) grid 5

Fig. 5 (Color online) Control law for the dike-optimisation test-case; continuous optimisation; the control law in this figure depends on the dike-height x_1, along the *horizontal* and time t, along the *vertical axis*. Given some (x_1, t), the *color* indicates the optimal dike increase u, see the adjacent *color-bar* for the scaling; the optimal dike heights for $x_1(0) = 425$ cm are indicated by a *black line*

the PDE in (105), and at the intervention times both the discrete and continuous optimisation procedures, as described in Sect. 3.2, are used in the tests to follow.

Results

Computations are performed on five grids. The grid resolutions and corresponding computing times are presented in Table 4.

The computed control law, i.e. $u(h, t_k)$, is presented in Fig. 5 for the coarsest and finest grids (grids 1 and 5). In this figure, the optimal dike heights for an initial dike-level of $x_1(0) = 425$ cm are indicated by a black line. This line is vertical when the dikes do not have to be increased, whereas the horizontal parts indicate a dike increase. The corresponding optimal dike-reinforcement times and increases are presented in Tables 5 and 6.

Very good grid convergence is observed and the data compare very well with the results of the discrete time model of [12]. The continuous optimisation approximately doubles computing time, while faster grid convergence is observed, most prominently for the third dike reinforcement 'C'.

Table 5 Optimal dike-reinforcements $(t,u)_A$ to $(t,u)_D$ for $x_1(0) = 425$ cm for the different grids; discrete optimisation

	t_A [yr]	u_A [cm]	t_B [yr]	u_B [cm]	t_C [yr]	u_C [cm]	t_D [yr]	u_D [cm]
grid 1	147	45	193	45	245	35	300	30
grid 2	146	45	190	40	238	40	300	30
grid 3	146	45	190	40	237	30	298	30
grid 4	146	45	190	40	236	35	289	30
grid 5	146	45	189	40	236	35	289	30
discr. time	147	50	191	45	239	40	293	35

Table 6 Same as Table 5; continuous optimisation

	t_A [yr]	u_A [cm]	t_B [yr]	u_B [cm]	t_C [yr]	u_C [cm]	t_D [yr]	u_D [cm]
grid 1	147	45.73	194	43.99	245	37.29	300	28.72
grid 2	147	45.87	191	41.55	240	37.04	299	30.80
grid 3	146	44.96	190	41.71	239	37.20	297	30.83
grid 4	146	44.98	190	41.70	239	37.18	296	30.55
grid 5	146	44.99	190	41.70	239	37.18	296	30.61
discr. time	147	50	191	45	239	40	293	35

3.4 Dike Optimisation with Stochastic Economic Growth

We will now extend our previous dike optimisation problem by assuming stochastic economic growth as follows:

$$\mathrm{d}x_3 = \alpha_3 x_3\,\mathrm{d}t + \mu_3 x_3\,\mathrm{d}z, \tag{108}$$

where we take $\mu_3 = 0.15$. The terminal condition is as before. Since x_3 and $\mathcal{J}$ are stochastically independent, (83) is again obtained. We cannot immediately apply the reduction of (104), but set

$$h = x_1 - x_2, \tag{109}$$

$$y = -\alpha_3 t + \ln x_3, \tag{110}$$

$$\tau = T - t, \tag{111}$$

$$V(x,t) = x_3 e^{W(h,y,\tau)}. \tag{112}$$

This particular choice of variables is numerically beneficial, since it keeps W within bounds and transforms it in an, almost, piece-wise linear form.

Note that y is chosen such that $y = \text{constant}$ corresponds to the expected economic growth. The governing equation (105) transforms into

$$\begin{cases} \dfrac{\partial W}{\partial \tau} = -\dfrac{\mathrm{d}w}{\mathrm{d}t}(T-\tau)\dfrac{\partial W}{\partial h} - (r-\alpha_3) \\ \qquad\qquad + \dfrac{1}{2}\mu_3^2\left(1+\dfrac{\partial W}{\partial y}\right)\dfrac{\partial W}{\partial y} + \dfrac{1}{2}\mu_3^2\dfrac{\partial^2 W}{\partial y^2} + \dfrac{\lambda\beta(h)}{e^W}, \\ W(h,y,\tau+) \geq \ln\left(\inf\limits_{\hat{u}\in\mathcal{U}}\left[e^{W(h+\hat{u},y,\tau)} + \dfrac{b_2((w(T-\tau)+h, w(T-\tau), e^{y+\alpha_3(T-\tau)})^t, \hat{u})}{e^{y+\alpha_3(T-\tau)}}\right]\right), \end{cases} \tag{113}$$

for $(h, y, \tau) \in [h_L, h_R] \times [y_L, y_R] \times [0, T]$ and with the same initial conditions as before, i.e.

$$W(h, y, 0) = \max\left(\ln\left(\frac{\lambda\beta(h)}{r-\alpha_3}\right), \epsilon\right) \tag{114}$$

and boundary conditions

$$\begin{aligned} \frac{\partial W}{\partial \tau}(h_L, y, \tau) = &-(r-\alpha_3) \; + \; \frac{1}{2}\mu_3^2\left(1+\frac{\partial W}{\partial y}(h_L, y, \tau)\right)\frac{\partial W}{\partial y}(h_L, y, \tau) \\ &+ \frac{1}{2}\mu_3^2\frac{\partial^2 W}{\partial y^2}(h_L, y, \tau) + \frac{\lambda\beta(h_L)}{e^{W(h_L, y, \tau)}}, \end{aligned} \tag{115}$$

$$\frac{\partial W}{\partial \tau}(h, y_L, \tau) = -\frac{\mathrm{d}w}{\mathrm{d}t}(T-\tau)\frac{\partial W}{\partial h}(h, y_L, \tau) \; - (r-\alpha_3) + \frac{\lambda\beta(h)}{e^{W(h, y_L, \tau)}}, \tag{116}$$

$$\frac{\partial W}{\partial \tau}(h, y_R, \tau) = -\frac{\mathrm{d}w}{\mathrm{d}t}(T-\tau)\frac{\partial W}{\partial h}(h, y_L, \tau) \; - (r-\alpha_3) + \frac{\lambda\beta(h)}{e^{W(h, y_L, \tau)}}. \tag{117}$$

We take $h_L = \ln X_3(0) - \ln\kappa$ and $h_R = \ln X_3(0) + \ln\kappa$, such that

$$\frac{1}{\kappa}\mathbb{E}\{X_3(t)\} \leq x_3 \leq \kappa\mathbb{E}\{X_3(t)\}. \tag{118}$$

We set $\kappa = 4$ and did not observe significantly different results for $\kappa = 5$ or $\kappa = 6$, so we assume that the boundaries $y = y_L$ and $y = y_R$ are sufficiently far away. A heuristic time-step criterion is obtained by considering the stability condition of the time-integration method for convection in h- and y-direction and diffusion in y-direction respectively, i.e.

$$\Delta\tau \leq \sigma \min\left(\frac{\Delta h}{\text{maxabs}(\frac{\mathrm{d}w}{\mathrm{d}t})}, \frac{\Delta y}{\frac{1}{2}\mu_3^2}, \frac{1}{2}\frac{\Delta y^2}{\frac{1}{2}\mu_3^2}\right), \tag{119}$$

where we take $1 \geq \sigma = 0.4$. We choose the maximum possible $\Delta\tau$ such that it satisfies (119) and that the control-time interval $t_{k+1} - t_k$ is a multiple of $\Delta\tau$.

Results

We were satisfied with the results on grid 2 of the previous problem, so we fixed $N_h = 161$. Five different grids are now defined with varying number of nodes in the y-direction, see Table 7. Note that the computing time is now presented in minutes. The optimal dike-reinforcement times and increases, when the economic growth is confined to its expected

Table 7 Grid resolutions and computing times for the dike-optimisation test-case with stochastic economic growth; $N_h = 161$; continuous optimisation

grid	N_y	Δy [cm]	N_t	Δt [yr]	comp. time [min]
1	11	0.5545	600	0.5	1.71
2	21	0.2773	600	0.5	2.96
3	41	0.1386	1200	0.25	8.56
4	81	0.0693	3600	0.0833	39.95
5	161	0.0347	14400	0.0208	297.86

Table 8 Optimal dike-reinforcements $(t,u)_A$ to $(t,u)_D$ for the different grids; the economic growth is confined to its expected value, i.e. the plane $x_3 = \mathbb{E}\{X_3(t)\}$; $x_1(0) = 425$ cm

	t_A [yr]	u_A [cm]	t_B [yr]	u_B [cm]	t_C [yr]	u_C [cm]	t_D [yr]	u_D [cm]
grid 1	148	44.75	191	40.45	241	37.62	297	31.82
grid 2	148	44.74	191	40.46	241	37.62	297	31.82
grid 3	148	44.78	191	40.45	241	37.62	297	31.86
grid 4	148	44.81	191	40.44	241	37.62	297	31.87
grid 5	148	44.82	191	40.44	241	37.62	297	31.88

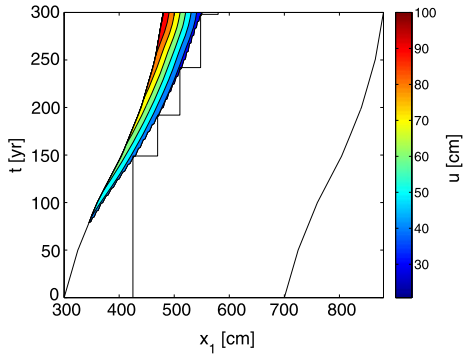


Fig. 6 (Color online) Optimal control law in the plane $x_3 = \mathbb{E}\{X_3(t)\}$ for the dike-optimisation test-case with stochastic economic growth; continuous optimisation. Given some (x_1, t), the *color* indicates the optimal dike increase u in time, see the adjacent *color-bar* for the scaling; the optimal dike heights for $x_1(0) = 425$ cm are indicated by a *black line*

value, are presented in Table 8. Rapid grid convergence is observed. The differences with the previous deterministic economic growth test-case appear to be relatively small. The corresponding optimal control law, i.e. $u(h, \ln(X_3(0)), t_k)$, is presented in Fig. 6 for grid 5. Results on the other grids are the same, as may be expected from the rapid grid convergence in Table 8. Figure 7 shows the optimal control law in (x_1, x_3)-planes on the same grid. Note that the sawtooth behaviour of the smallest isocontourline is an artifact of the plotting software. Recall that the control law is discontinuous, due to the threshold value in the investment costs, see (96). As an illustration, the isocontours of u^+, i.e. optimised over the set $\mathcal{U} \setminus \{0\}$, are plotted in Fig. 8 for $t = 100$ and exhibit a smooth behaviour. For an exposure of the optimal dike-level computation near the discontinuity of u, see Remark 3.3.

4 Conclusions

A model to compute the optimal dike heights and reinforcement times in continuous time has been presented. The problem is formulated as an optimal control problem and based on

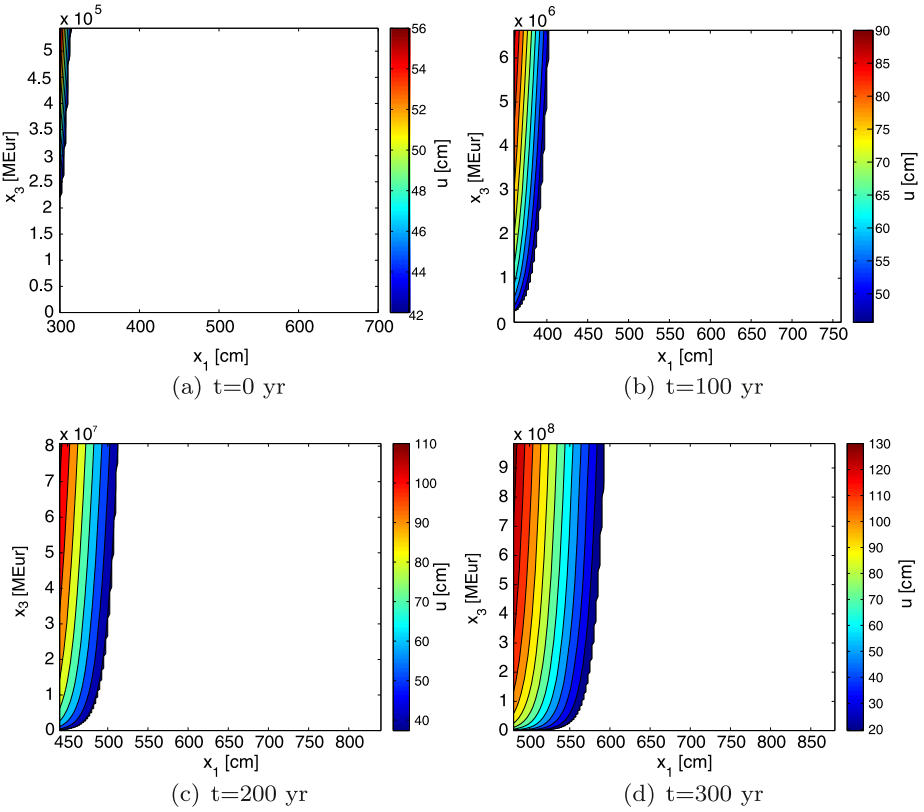


Fig. 7 (Color online) Optimal control law in (x_1, x_3) planes for the dike-optimisation test-case with stochastic economic growth; continuous optimisation; Given some (x_1, x_3), the *color* indicates the optimal dike increase u in time, see the adjacent *color-bar* for the scaling

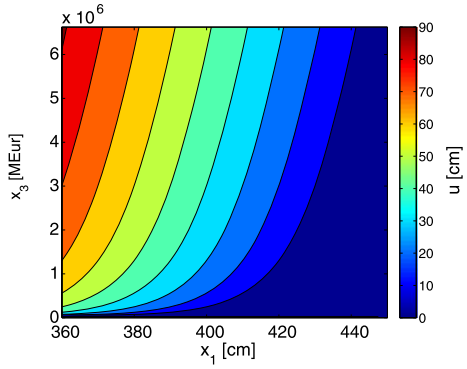


Fig. 8 Control law u^+ over the reduced set $\mathcal{U} \setminus \{0\}$ in an (x_1, x_3) plane for the dike-optimisation test-case with stochastic economic growth; continuous optimisation; grid 5; $t = 100$ yr

the minimisation of future expected losses due to floods, and investment costs. The system dynamics are described by the dike height, *average* water level and economic value at risk. A Poisson point process is adopted to model *extreme* water-levels, enabling an expression for the future expected losses. The control problem leads to the so-called Hamilton-Jacobi-

Bellman (HJB) equation, where the dike-increase and reinforcement times act as input to the control problem.

The HJB equations are a set of partial differential equations that are solved numerically by a conservative finite difference discretization. To ensure the convergence to the proper solution, a high-order Essentially Non-Oscillatory (ENO) method is adopted. The ENO methodology is originally intended for hyperbolic conservation laws and is extended here to deal with diffusion-type problems. The method is validated by considering a test-case from Computational Finance. Faster grid convergence was observed, compared to lower-order results from literature, at the costs of explicit time-integration, limiting the maximum allowable time-step for stability and monotonicity.

The framework presented, based on ENO schemes, may serve as an alternative for techniques based on viscosity solutions. The framework offers discretizations of different orders of accuracy in a natural way. This is particularly attractive for high-dimensional HJB problems, for which one cannot use many grid points per coordinate direction, and for problems governed by steep gradients and discontinuities in the initial conditions, or in the solutions at the intervention times. The present dike optimisation problem does not exhibit these phenomena, which is basically because of a smart choice of the unknowns, in log-scale and scaled by the drift in the economic growth.

References

1. Bardi, M., Capuzzo-Dolcetta, I.: Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Systems & Control: Foundations & Applications Series. Birkhäuser, Boston (1997)
2. Barles, G.: Convergence of numerical schemes for degenerate parabolic equations arising in finance theory. In: Rogers, L.C.G., Talay, D. (eds.) Numerical Methods in Finance, pp. 1–21. Cambridge University Press, Cambridge (1997)
3. Bensoussan, A., Lions, J.L.: Impulsive Control and Quasi-Variational Inequalities. Gauthier-Villars, Paris (1984)
4. Bensoussan, A., Menaldi, J.L.: Hybrid control and dynamic programming. Dyn. Contin. Discrete Impuls. Syst. **3**, 395–442 (1997)
5. Carlini, E., Ferretti, R., Russo, G.: A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **27**(3), 1071–1091 (2005)
6. Chen, Z., Forsyth, P.A.: A numerical scheme for the impulse control formulation for pricing variable annuities with a guaranteed minimum withdrawal benefit (GMWB). Numer. Math. **109**, 535–569 (2008)
7. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal. **35**(6), 2440–2463 (1998)
8. de Haan, L.: Fighting the arch-enemy with mathematics. Stat. Neerl. **44**, 45–68 (1990)
9. Eijgenraam, C.J.J.: Optimal safety standards for dike-ring areas. CPB Discussion Paper 62, Centraal Planbureau, March 2006
10. Fleming, W.H., Soner, H.M.: Controlled Markov Processes and Viscosity Solutions. Stoch. Mod. Appl. Prob. Series, vol. 25. Springer, Berlin (2006)
11. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes, III. J. Comput. Phys. **71**, 231–303 (1987)
12. Kempker, P.: Optimal control for dike levels. Master's thesis, Vrije Universiteit Amsterdam, The Netherlands, August 2008
13. LeVeque, R.J.: Numerical Methods for Conservation Laws. Lectures in Mathematics. Birkhäuser, Basel (1992)
14. Merriman, B.: Understanding the Shu-Osher conservative finite difference form. J. Comput. Phys. **83**, 32–78 (1989)
15. Øksendal, B.: Stochastic Differential Equations: An Introduction with Applications, 5th edn. Universitext. Springer, Berlin (2000)
16. Pooley, D.M., Forsyth, P.A., Vetzal, K.R.: Numerical convergence properties of option pricing PDEs with uncertain volatility. IMA J. Numer. Anal. **23** (2003)

17. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
18. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. J. Comput. Phys. **83**, 32–78 (1989)
19. van Dantzig, D.: Economic decision problems for flood prevention. Econometrica **24**, 276–287 (1956)
20. van Noortwijk, J.M., van der Weide, J.A.M., Kallen, M.J., Pandey, M.D.: Gamma processes and peaks-over-threshold distributions for time-dependent reliability. Reliab. Eng. Syst. Saf. **92**, 1651–1658 (2007)