Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

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

Preprint BUW-IMACM 21/26

Tatiana Kossacká, Matthias Ehrhardt and Michael Günther

A Deep Smoothness WENO Method with Applications in Option Pricing

August 10, 2021

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

A Deep Smoothness WENO Method with Applications in Option Pricing

Tatiana Kossaczká, Matthias Ehrhardt, and Michael Günther

Abstract We present the novel deep smoothness weighted essentially non-oscillatory (WENO-DS) method and its application in finance. To improve the existing WENO method, we apply a deep learning algorithm to modify the smoothness indicators of the method. This is done in a way that preserves the consistency and accuracy of the method. We present our results using a European digital option as an illustrating example. Here we avoid the undesirable oscillations, especially in the first time steps of the numerical solution.

1 Introduction

In this work, we use the newly developed weighted essentially non-oscillatory (WENO-DS) method for solving the (backward-in-time) Black-Scholes equation

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV = 0, \quad t \in [0,T], \tag{1}$$

where S is the price of an underlying asset at time t, $r > 0$ is the riskless interest rate and σ^2 is the volatility.

The WENO method [9] is a high-order method, originally developed for solving hyperbolic conservation laws, where strong discontinuities appear in the solution. Later, it was also generalized also for solving of nonlinear degenerate parabolic equations [10]. Many modifications of the original WENO schemes have been done later and we focus in this paper on the WENO-Z method introduced in [1] and MWENO method developed in [2].

In computational finance problems, we often face the problems with discontinuous initial or terminal data. Therefore, the WENO scheme has been used, e.g. in

Tatiana Kossaczká · Matthias Ehrhardt · Michael Günther
Angewandte Mathematik und Numerische Analysis, Bergische Universität Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany, e-mail: {kossaczka, ehrhardt, guenther}@uni-wuppertal.de

[3, 6] for solving of these problems. In this paper, we solve the European digital option pricing problem with the following terminal and boundary conditions:

$$V(S,T) = \begin{cases} 1, & \text{if} \quad S \geq K, \\ 0, & \text{if} \quad S < K, \end{cases} \tag{2}$$
$$V(S,t) \to 0, \quad \text{for} \quad S \to 0, \quad V(S,t) \to e^{-r(T-t)}, \quad \text{for} \quad S \to \infty,$$

with K being a strike price.

Although the WENO scheme should avoid the spurious oscillations in the solution, they are still present in some cases, especially in the first time steps of the numerical solution. This motivates us to use the enhanced WENO-DS scheme [7, 8] for solving the European digital option pricing problem.

2 The WENO-DS scheme

Here we briefly summarize the basic idea of the WENO-DS method. We consider the following diffusion-convection-reaction partial differential equation (PDE):

$$\frac{\partial u(x,t)}{\partial t} = a_0 \frac{\partial^2 u(x,t)}{\partial x^2} + a_1 \frac{\partial u(x,t)}{\partial x} + a_2 u(x,t), \quad (x,t) \in \Omega \times (0,\infty), \tag{3}$$

where a_0, a_1 and a_2 are constant coefficients. We introduce the uniform spatial grid $x_i = x_0 + i\Delta x$, $i = 0,\dots,N$. The semi-discrete formulation of (3) can be written as

$$\frac{du_i(t)}{dt} = a_0 \frac{\hat{u}_{i+\frac{1}{2}} - \hat{u}_{i-\frac{1}{2}}}{\Delta x^2} + a_1 \frac{\tilde{u}_{i+\frac{1}{2}} - \tilde{u}_{i-\frac{1}{2}}}{\Delta x} + a_2 u_i(t), \quad t > 0, \tag{4}$$

where $u_i(t)$ approximates pointwise $u(x_i,t)$ and $\hat{u}_{i+1/2} = \hat{u}(u_{i-2},\dots,u_{i+3})$, $\tilde{u}_{i+1/2} = \tilde{u}(u_{i-2},\dots,u_{i+2})$ are the numerical flux functions. In order to obtain these values, the WENO discretization is used.

The basic idea of the WENO scheme is to combine the numerical approximations of the flux functions on three substencils to a final numerical approximation on the main stencil. For this purpose, the nonlinear weights ω_m, $m = 0,1,2$, have to be calculated. For example, for the approximation of the positive part of the numerical flux of the parabolic term, one obtains

$$\hat{u}_{i+\frac{1}{2}} = \sum_{m=0}^{2} \omega_m \hat{u}^m_{i+\frac{1}{2}}, \tag{5}$$

where the explicit formulas for $\hat{u}^m_{i+1/2}$ as well as expressions of ω_m can be found in [2]. For the formulas of the numerical fluxes and the nonlinear weights for the hyperbolic term we refer to [1].

To measure the smoothness of the solution on each of three candidate substencils, the smoothness indicators β_m, $m = 0,1,2$ [4] is used. In [7] a new idea of improving these smoothness indicators was introduced. Namely they are computed as the multiplication of the original smoothness indicators β_m and the perturbations δ_m, where δ_m is an output of a particular neural network algorithm. The new smoothness indicators take the form

$$\beta_m^{DS} = \beta_m(\delta_m + C), \qquad m = 0,1,2, \tag{6}$$

where C is a constant that ensures the consistency and high-order accuracy of the new method, which was analytically proven in [7] and [8]. Here, also a detailed explanation of this method can be found.

3 Numerical Results

We first use the following variable transformation:

$$S = Ke^x, \quad \tau = T - t, \quad V(S,t) = Ku(x,\tau) \tag{7}$$

and substitute this into (1) and (2). Then we obtain the (forward-in-time) PDE:

$$u_\tau = \frac{\sigma^2}{2}u_{xx} + \left(r - \frac{\sigma^2}{2}\right)u_x - ru, \quad x \in \mathbb{R},\ 0 \le \tau \le T. \tag{8}$$

This equation is of the form (3) and can be easily discretized using the WENO-DS scheme for both the hyperbolic and parabolic terms. It should be noted that for the temporal discretization, we use a third-order total variation diminishing (TVD) Runge-Kutta method [12], imposing intermediate boundary conditions as in [3]. Python with the Pytorch library is used for the implementation [11].

To obtain the enhanced WENO-DS scheme for solving the European digital option pricing problem, we train a convolutional neural network (CNN) on a large set of data. For the training, we set $K = 50$, $T = 1$, and randomly generate the parameters

$$\begin{aligned} \sigma &= 0.31 + \max(0.07a, -0.3), \\ r &= 0.11 + \max(0.07b, -0.1), \end{aligned} \tag{9}$$

where a and b are normally distributed. Here, the problems with different combinations of σ and r are covered. We use the computational domain $[x_L, x_R] = [-6, 1.5]$ partitioned into 100 space steps and use the temporal step size $\Delta\tau = 0.8\Delta x^2/\sigma^2$. As we mentioned earlier, the spurious oscillations mainly occur in the first time steps of a numerical solution. Therefore, we proceed with a training as follows.

First, the parameters (9) are randomly generated. We initialize the weights of the CNN randomly and perform a single time step of a solution. The structure of the CNN can be seen in Figure 1. We emphasize that we use a rather small CNN to be computationally efficient. We use the same CNN structure for training both WENO-DS for the hyperbolic term and WENO-DS for the parabolic term. We compute the values u_{diff1}, u_{diff2}, which represent an effective preprocessing of the solution from the current time step, since they give us information about the smoothness of the solution. They are given by

$$u_{\text{diff1},i} = \bar{u}(\bar{x}_{i+1}) - \bar{u}(\bar{x}_{i-1}), \quad u_{\text{diff2},i} = \bar{u}(\bar{x}_{i+1}) - 2\bar{u}(\bar{x}_i) + \bar{u}(\bar{x}_{i-1}), \tag{10}$$

with

$$\begin{aligned} \bar{x}_i &= (x_{i-k}, x_{i-k+1}, \dots, x_{i+k}), \\ \bar{u}(\bar{x}_i) &= \big(u(x_{i-k}), u(x_{i-k+1}), \dots, u(x_{i+k})\big), \end{aligned} \tag{11}$$

where $2k+1$ is the size of the receptive field of the whole CNN. They are then used as input values for the first hidden layer.

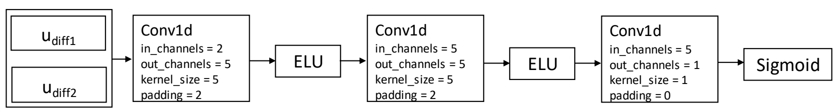


Fig. 1: The structure of the convolutional neural network.

Then we calculate a loss with

$$\text{LOSS}(u) = \sum_{i=0}^{N-1} \big[\max(u_i - u_{i+1}, 0)\big], \tag{12}$$

where u_i is a numerical approximation of $u(x_i)$. This loss is positive, if the approximation of the solution is decreasing in x (in true solution it should be only increasing), so we test the monotonicity of the solution. After that, the gradient with respect to the weights of the CNN is calculated using the backpropagation algorithm. Then, the Adam optimizer [5] with a learning rate of 0.001 is used to update the weights. Next, we test the model on a validation set and repeat the above steps with newly generated parameters (9). After the training, we select the weights from the training step, at which the model performed best on the validation problems.

In the Figure 2, we show the evolution of the loss value for the problems from the validation set. We see that the loss is decreasing and select the model obtained after the last training step as our final WENO-DS scheme.

We compare the solution at the first time step on Figure 3a and see that the WENO-DS reliably eliminates the oscillations that occur when using the original WENO scheme (WENO-Z scheme [1] for the approximation of the hyperbolic term and MWENO scheme [2] for the approximation of the parabolic term).

In most cases, the original WENO scheme is able to handle these oscillations with increasing number of time steps. However, in some cases the oscillations are

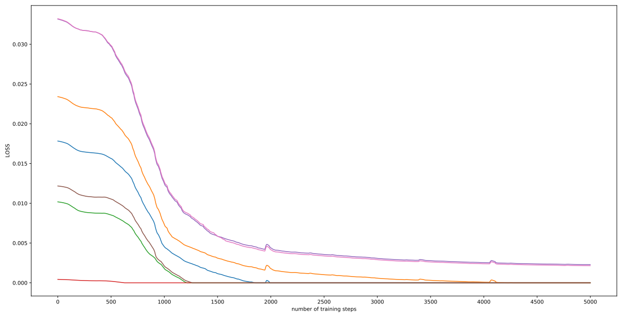
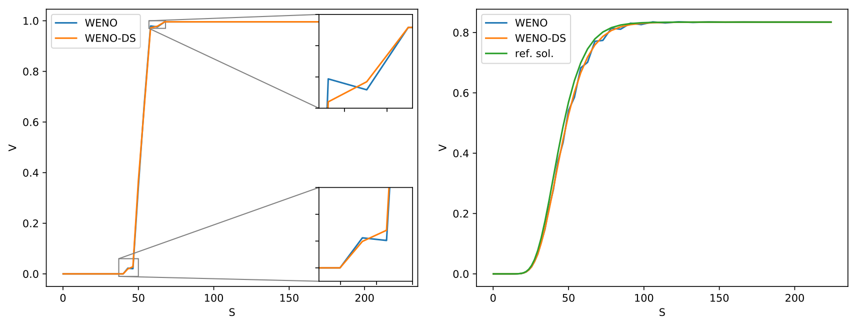


Fig. 2: Loss values for different validation problems.

still present. Figure 3b shows the solution at time $T = 1$ and we see that our method produces a smooth solution unlike the original WENO method.



(a) Solution at the first time step, $\sigma = 0.4$ and $r = 0.15$

(b) Solution at the last time step, $T = 1$, $\sigma = 0.3$ and $r = 0.2$

Fig. 3: Comparison of the original WENO and WENO-DS methods, $N = 100$.

We compare the L^∞ and L^2 errors in Table 1 and show that the WENO-DS method has a smaller error in all cases. Thus, we are not only able to eliminate the spurious oscillations, but also improve the quality of the numerical solution.

4 Conclusion

In this paper, we applied the newly developed WENO-DS method to the European digital option pricing problem that has discontinuous terminal data. In this problem, the spurious oscillations are present in the solution when the standard WENO scheme is used. We have shown that they can be successfully eliminated using the

		L^∞		L^2	
σ	r	WENO	WENO-DS	WENO	WENO-DS
0.28	0.13	0.000933	0.000908	0.000660	0.000644
0.1	0.05	0.002751	0.002655	0.001196	0.001158
0.3	0.2	0.001120	0.000858	0.000650	0.000621
0.2	0.1	0.001833	0.001687	0.000890	0.000865
0.15	0.05	0.002446	0.002352	0.001055	0.001034
0.4	0.1	0.000676	0.000661	0.000570	0.000557

Table 1: Comparison of the L^∞ and L^2-error of original WENO and WENO-DS methods for the solution of the transformed Black-Scholes equation (8) with various parameters σ and r.

WENO-DS method. To this end, we trained a CNN to modify the smoothness indicators of the original method. Since we can obtain smaller errors with the proposed algorithm, the quality of the numerical solution was also improved.

References

1. Borges, R., Carmona, M., Costa, B., and Don, W. S.: An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. J. Comput. Phys. 227.6 (2008): 3191-3211.
2. Hajipour, M., and Malek, A.: High accurate NRK and MWENO scheme for nonlinear degenerate parabolic PDEs. Appl. Math. Model. 36.9 (2012): 4439-4451.
3. Hajipour, M., and Malek, A.: High accurate modified WENO method for the solution of Black–Scholes equation. Comput. Appl. Math. 34.1 (2015): 125-140.
4. Jiang, G.-S., and Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. 126.1 (1996): 202-228.
5. Kingma, D. P., and Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014), published as a conference paper at ICLR 2015.
6. Kossaczká, T.: The Weighted Essentially Non-Oscillatory Method for Problems in Finance. Master Thesis, University of Wuppertal (2019).
7. Kossaczká, T, Ehrhardt, M., and Günther, M.: Enhanced fifth order WENO shock-capturing schemes with deep learning. arXiv preprint arXiv:2103.04988 (2021).
8. Kossaczká, T., Ehrhardt, M., and Günther, M.: A neural network enhanced WENO method for nonlinear degenerate parabolic equations. tech. report, IMACM Preprint 21/16 (2021).
9. Liu, X.-D., Osher, S., and Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. 115.1 (1994): 200-212.
10. Liu, Y., Shu, C.-W., and Zhang, M.: High order finite difference WENO schemes for nonlinear degenerate parabolic equations. SIAM J. Sci. Comput. 33.2 (2011): 939-965.
11. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inform. Process. Syst. 32 (2019): 8026-8037.
12. Shu, Ch.-W., and Osher S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. 77.2 (1988): 439-471.