University of West Bohemia
Faculty of Applied Sciences
Department of Mathematics

Master Thesis

ENO Methods with Radial Basis Functions for Conservation Laws

Pilsen, 2013

Bc. Eva Turnerová

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra matematiky

Diplomová práce

ENO metody s využitím radiálních bázových funkcí pro zákony zachování

Plzeň, 2013

Bc. Eva Turnerová

Declaration

I do hereby declare that the entire bachelor thesis is my original work and that I have used only the cited sources.

V Plzni dne .....

.....
Eva Turnerová

Acknowledgement

I would like to express my sincere gratitude to Ing. Aleš Matas Ph.D. and Doc. Ing. Marek Brandner Ph.D., for introducing me to the theory of optimal recovery and last but not least for their devoted approach and guidance.

Abstract

The theory of essentially non-oscillatory schemes (ENO) is applied to finite volume methods for the numerical solution of hyperbolic conservation laws. ENO are based on the reconstruction problem of the cell averages. The reconstruction is built in the stencil which is a set of the cells in which the data are the smoothest. Therefore, the procedure controls the oscillations as much as possible. Classical ENO schemes use polynomials. In the thesis we introduce also another way of the reconstruction using conditionally positive definite radial basis functions (RBF). The thin plate splines, as the combination of the linear polynomial and radial basis functions, can be suitably extended in multi-dimensional finite volume schemes. We provide some numerical examples of ENO methods using both polynomials and RBFs.

Contents

Introduction

Partial differential equations describe a wide variety of models for real problems like sound propagation, heat, elasticity etc. Since the equations are usually very complicated, only special models can be solved analytically. In this thesis, we present numerical method for solving hyperbolic conservation laws.

In last few years, it increases the interest about methods which give us high order accurate numerical solution. The classical finite difference methods of first order are very inaccurate and higher order of such methods products oscillations near non–smooth data. There exists improved methods which try to reduce these oscillations for example by adding artificial viscosity or reducing the order of the accuracy near the discontinuities, but the lower order spreads globally and the numerical solution falls back to the first order anyway.

In this thesis, we focus on finite volume schemes which are based on approximation theory using preferably polynomials. The main aim of the authors was to remove the oscillations, then the total variation dimishing (TVD) schemes were developed. These methods try to strictly satisfy the non-oscillatory behaviour so that if we use higher order accuracy, it degenerates again to the first order method. First non-TVD finite volume schemes apply the fixed stencil interpolation, but it does not respect the discontinuities and thus these procedures result in oscillations again.

Harten, Engquist, Osher and Chakravarthy in 1987 introduced the so–called Essentially Non-Oscillatory (ENO) schemes. It was created as a numerical method which eliminates the problem of the methods used so far. The word essentially means that it is possible that the method can products new extrema of the size of the truncation error. The theory of (essentially) local extremum dimishing schemes is discussed in [17, 7, 1].

ENO methods use originally polynomial reconstruction with an adaptive stencil. It is known in the approximation theory that the more data we have, the higher order of the accuracy we get. We use the same idea with the stencil for ENO methods. The main task of ENO schemes is how to choose the stencil. We select stencil as a set of the cells which include the most smooth data as much as possible. For the cell i just left from the shock in one dimensional space we try to choose stencil on the same side, for instance $\{i-2, i-1, i\}$ depending on required order of the accuracy. We use divided difference in 1D reconstruction to choose the cells to the stencil according to [4, 2].

ENO methods are studied in one dimensional space by [4, 1]. The first extension of the method into 2D problems is again using the classical polynomials, in [4] on structured mesh, in [2, 5, 8] on unstructured mesh. 2D reconstruction causes bigger problem in the selection of the stencil. First we have more possibilities how to choose the cells. Some authors, for example Sonar in [6], focus on efficiency of the algorithm and they consider only limited number of the possible stencils. Second task is how to choose the most appropriate stencil so that we work with the most smooth data. Traditionally, total variation is used but Jameson in [17] shows an example where the total variation have smaller value for less smooth data than for more smooth data.

Because ENO method is based on reconstruction theory, as finite volume method, recently authors started to focus on different kind of the approximation. Specifically, the thin plate splines with the radial basis functions yield an optimal recovery algorithms according to [6, 7, 10, 13]. The radial basis functions (RBF) are often used in the approximation theory in the last years and they were applied to solving differential equations about 15 years ago. We can find the properties and application of radial basis functions in [9, 14, 15, 16] in details.

The text is organized into two main chapters thus we introduce the essentially non-oscillatory schemes for one dimensional linear problem in the first chapter. We focus on polynomial reconstruction and we are interested in quality of the numerical solution depending on the degree of the polynomial.

The second chapter is concentrated on the conservation law in the two dimensional space. We extend the term reconstruction on triangular mesh. The main task is to compare the quality of the numerical solution depending on approximation of the cell averages using the polynomials and thin plate splines. We apply this theory on scalar linear equation, system of linear equations and also nonlinear equation.

Chapter 1

ENO schemes in one dimensional space

Essentially non–oscillatory schemes have been constructed to solve the hyperbolic partial differential equations with high order accuracy. In one spatial dimension, the hyperbolic PDE is in the form

$$u_t(x,t) + f_x(u(x,t)) = 0, \quad x \in [a,b], t \in (0,T) \tag{1.1}$$

with the initial condition

$$u(x,0) = u_0(x), \quad x \in [a,b]. \tag{1.2}$$

ENO methods are a finite volume methods which try to improve the classical methods. The main part of the finite volume methods is the approximation of the solution u using the cell averages. There exist a lot of methods which try to build the approximation such that the numerical solution does not oscillate. The so called total variation dimishing (TVD) schemes do not product the oscillations even if the stencil contain the cells with discontinuous data, i.e. they strictly satisfy the non-oscillatory behaviour. It implies that the TVD methods of higher order degenerate to the first order anyway. But the methods, which are not TVD and are based on approximation of the data in the fixed stencil, product oscillations.

ENO method is motivated by the previous problems and it introduces the adaptive stencil. The idea is to avoid the discontinuity and interpolate data in the smoothest regions. We study the procedure how to choose the appropriate stencil and thus get numerical solution of high order accuracy in the following sections.

Although the ENO methods are not TVD, they are really high order accurate methods. It means that ENO scheme can products little oscillations but they are comparable with the truncation error, [17, 7, 1].

1.1 Spatial Discretization

Let us consider the hyperbolic partial differential equation in one dimensional space (1.1) in this chapter s.t. the solution is dependent on both spatial and time variable. In this section, the discretization only in space is considered and we leave the problem continuous in time. We assume solving the problem (1.1) in the interval $I = [a, b]$. By discretization of this interval I we understand a grid

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} \ldots < x_{N-\frac{1}{2}} < x_{N+\frac{1}{2}} = b. \tag{1.3}$$

Each interval

$$I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right], \quad \forall i = 1, 2, \ldots, N, \tag{1.4}$$

is called the cell. The cell centeres and cell sizes are denoted by

$$x_i = \frac{1}{2}\left(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}\right) \quad \forall i = 1, 2, \ldots, N, \tag{1.5}$$

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \quad \forall i = 1, 2, \ldots, N. \tag{1.6}$$

In this thesis we consider equidistant grid, i.e. all cell sizes are equal.

1.2 Finite Volume Method

As we mentioned, the ENO scheme is the finite volume method. To build this method, we start with integration of (1.1) over the cell I_i

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u\left(x,t\right)\mathrm{d}x = -\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f_x\left(u\left(x,t\right)\right)\mathrm{d}x. \tag{1.7}$$

We determine the antiderivative of the right side of the last equation and we divide whole equation by the cell size to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u\left(x,t\right)\mathrm{d}x\right) = -\frac{1}{\Delta x}\left(f\left(u\left(x_{i+\frac{1}{2}},t\right)\right) - f\left(u\left(x_{i-\frac{1}{2}},t\right)\right)\right). \tag{1.8}$$

Let us define

$$\bar{u}\left(x_i,t\right) = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u\left(x,t\right)\mathrm{d}x, \tag{1.9}$$

which are the cell averages. The integrated form of the equation (1.1) is in the following form

$$\frac{\mathrm{d}\bar{u}\left(x_i,t\right)}{\mathrm{d}t} = -\frac{1}{\Delta x}\left(f\left(u\left(x_{i+\frac{1}{2}},t\right)\right) - f\left(u\left(x_{i-\frac{1}{2}},t\right)\right)\right). \tag{1.10}$$

The approximation of the flux leads to the equality

$$\frac{\mathrm{d}\bar{u}\left(x_i,t\right)}{\mathrm{d}t}=-\frac{1}{\Delta x}\left(\bar{f}_{i+\frac{1}{2}}-\bar{f}_{i-\frac{1}{2}}\right), \tag{1.11}$$

where the numerical flux is defined by

$$\bar{f}_{i+\frac{1}{2}}=h\left(u_{i+\frac{1}{2}}^{-},u_{i+\frac{1}{2}}^{+}\right). \tag{1.12}$$

The value $u_{i+\frac{1}{2}}^{-}$, resp. $u_{i+\frac{1}{2}}^{+}$, is the function value of the unknown solution u at the point $x_{i+\frac{1}{2}}$ according to the cell I_i, resp. I_{i+1}. The basic problem of the finite volume method is that we do not know the values $u_{i+\frac{1}{2}}^{\pm}$ because we have available only the cell averages (1.9). Therefore (1.11) can not be formulated. We thus have to estimate these values.

We obtain them by a procedure of reconstruction which we study in the section 1.3 in details. This procedure does not compute the values $u_{i+\frac{1}{2}}^{\pm}$ exactly. We thus consider the cell averages of the solution in time to be approximate too. We denote its approximation as $\bar{U}_i\left(t\right)$ and $U_{i+\frac{1}{2}}^{\pm}$. This leads to the conservative scheme

$$\frac{\mathrm{d}\bar{U}_i\left(t\right)}{\mathrm{d}t}=-\frac{1}{\Delta x}\left(\bar{f}_{i+\frac{1}{2}}-\bar{f}_{i-\frac{1}{2}}\right), \tag{1.13}$$

Let us assume that $U_{i+\frac{1}{2}}^{\pm}$ are already known for each cell. Then the function h in (1.12), i.e. the approximation to the numerical flux, can be defined for example by one of the following ways

- Godunov flux:

$$h\left(U_{i+\frac{1}{2}}^{-},U_{i+\frac{1}{2}}^{+}\right)=\begin{cases}\min_{U_{i+\frac{1}{2}}^{-}\leq u\leq U_{i+\frac{1}{2}}^{+}}f\left(u\right) & \text{if } U_{i+\frac{1}{2}}^{-}\leq U_{i+\frac{1}{2}}^{+}\\ \max_{U_{i+\frac{1}{2}}^{+}\leq u\leq U_{i+\frac{1}{2}}^{-}}f\left(u\right) & \text{if } U_{i+\frac{1}{2}}^{-}>U_{i+\frac{1}{2}}^{+}\end{cases}, \tag{1.14}$$

- Lax-Friedrichs flux:

$$h\left(U_{i+\frac{1}{2}}^{-},U_{i+\frac{1}{2}}^{+}\right)=\frac{1}{2}\left[f\left(U_{i+\frac{1}{2}}^{-}\right)+f\left(U_{i+\frac{1}{2}}^{+}\right)-\alpha\left(U_{i+\frac{1}{2}}^{+}-U_{i+\frac{1}{2}}^{-}\right)\right], \tag{1.15}$$

where $\alpha=\max_u\left|f'\left(u\right)\right|$ is a constant and the maximum is taken over the relevant interval of u.

Notice that the initial condition can be discontinuous in real problem. In such case, we can not study a solution of the problem in the classical sense.

Therefore we use concept of weak solution which has to satisfy the following integral identity

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u\left(x,t\right)\mathrm{d}x + \frac{1}{\Delta x}\left(f\left(u\left(x_{i+\frac{1}{2}},t\right)\right) - f\left(u\left(x_{i-\frac{1}{2}},t\right)\right)\right) = 0, \quad (1.16)$$

over every cell I_i, $i = 1, \ldots, N$. The weak solution defined in this way is not generally unique which leads to the usage of so called entropy conditions. For this topic, more details could be found in [12, 13, 7].

1.3 Reconstruction

In the previous section, we mentioned that ENO scheme uses only the discrete cell averages (1.9) but the unknown values $U^{\pm}_{i+\frac{1}{2}}$ at the cell boundaries $x_{i+\frac{1}{2}}$ are required to formulate (1.13). These unknowns are determined using the so called reconstruction. By reconstruction we understand an approximation of the function $u\left(x\right)$ in each cell I_i based on the values of the cell averages.

It is possible to use more ways to approximate data. We discuss a polynomial reconstruction in 1D problem and we add reconstruction using radial basis functions to 2D problems. The polynomial reconstruction is determined for every cell I_i. At first, the condition for conservation of cell averages is used. Let us assume that $p_i(x)$ is a polynomial reconstruction of u for the cell I_i in the time t. Therefore

$$\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i\left(x\right)\mathrm{d}x = \bar{U}_i. \quad (1.17)$$

The simplest way of the reconstruction is to use a constant polynomial, see Figure 1.1. To get the approximation using constant polynomial and hold the cell average (1.17) at the same time it provides unique result $p_i(x) = \bar{U}_i$. If we repeat it for each cell I_i then we obtain reconstructed data of u. The reconstruction $p_i(x)$ is defined at each point, i.e. also at the cell boundaries $x_{i\pm\frac{1}{2}}$ where we look for the approximation of $U_{i\pm\frac{1}{2}}$.

For the given reconstruction $p_i(x)$ in the cell I_i, the unknown value $U^{+}_{i-\frac{1}{2}}$ is obtained as function value of the polynomial p_i at the point $x_{i-\frac{1}{2}}$ and $U^{-}_{i+\frac{1}{2}}$ as function value of the polynomial p_i at the point $x_{i+\frac{1}{2}}$. In the special case of constant recovery, these unknows are simply the values of the cell average of the corresponding cell I_i. According to the Figure 1.1, the values from left and right $U^{\pm}_{i-\frac{1}{2}}$ do not have to be equal.

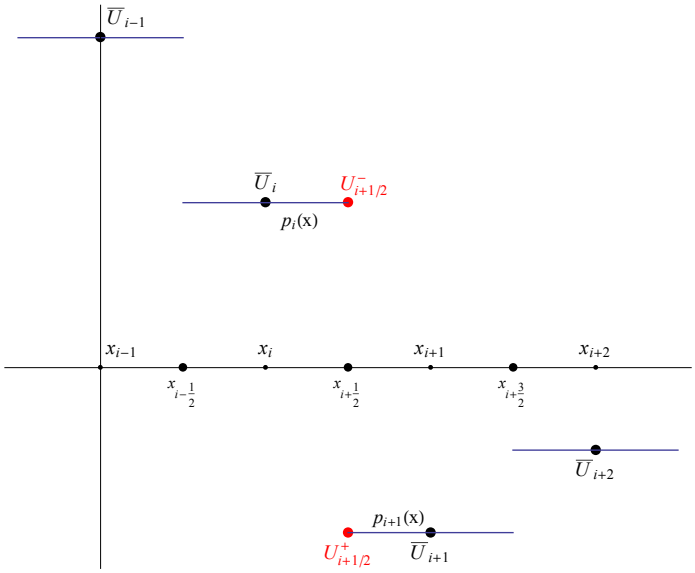

Figure 1.1: Polynomial reconstruction of the cell averages using constant polynomial.

If we consider a polynomial reconstruction, we can generally use polynomial of any degree. It holds that for method of the k-th order, we use a polynomial of degree at most $k-1$

$$p_i(x) = u(x) + O\left(\Delta x^k\right), \quad x \in I_i, i = 1, 2, \ldots, N. \tag{1.18}$$

We can read the proof in [4]. Let us assume that we want to construct a linear reconstruction for the cell I_i, i.e. we use a polynomial of the form $p_i(x) = a_0 + a_1(x - x_i)$. To determine the linear polynomial we need two conditions to compute the unknown coefficients a_0 and a_1. The first condition is the same as for a constant polynomial because we require the conservation of the cell average of the polynomial over cell I_i. One of the neighboring cells I_{i-1} or I_{i+1} gives us another condition. We require also for one of the cells I_{i-1} or I_{i+1} to fulfill the cell average

$$\frac{1}{\Delta x}\int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} p_i(x)\mathrm{d}x = \bar{U}_{i-1} \tag{1.19}$$

or

$$\frac{1}{\Delta x}\int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} p_i(x)\mathrm{d}x = \bar{U}_{i+1}. \tag{1.20}$$

It is not evident which neighbouring cell to choose. Mainly, it is very important for discontinuous data. The reason comes from the classical finite

difference method. Specifically, finite difference method of the first order is not very accurate and it products oscillations using finite difference method of the second order near non–smooth data.

ENO methods try to fix the problem of the finite difference method mentioned above. We say that for the corresponding cell I_i we want to select the so called adaptive stencil S_i. It is a set of neighbouring cells over which we look for the reconstruction.

In our example for linear polynomial, we start with the stencil $S_i = \{I_i\}$. The next step is to add one of the neighbours, as we discussed above, and get either the stencil $S_i = \{I_{i-1}, I_i\}$ or $S_i = \{I_i, I_{i+1}\}$. We show both options in the Figure 1.2 and we discuss how to select more appropriate stencil in the following paragraphs.

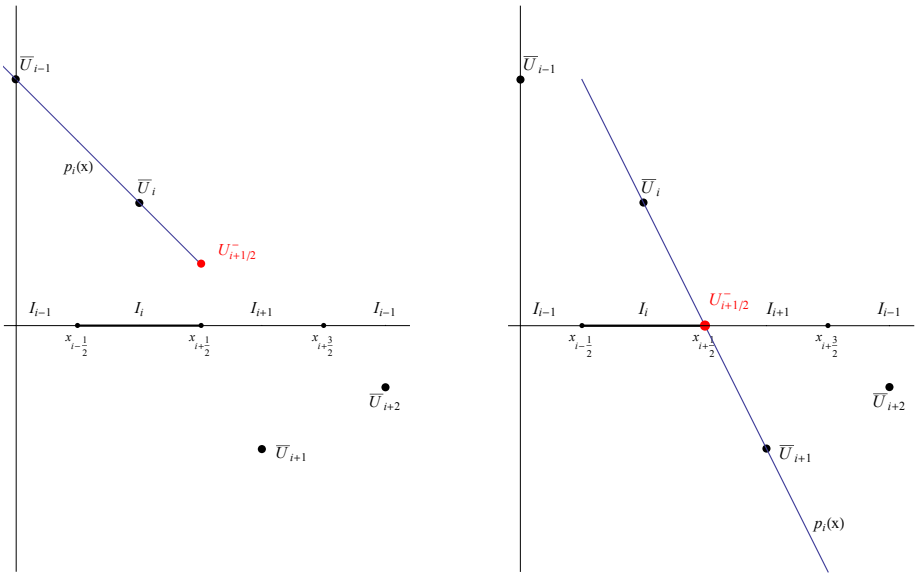

Figure 1.2: Polynomial reconstruction of the cell averages using linear polynomial, where the stencil is $\{I_{i-1}, I_i\}$ (left) or $\{I_i, I_{i+1}\}$ (right).

Generally in case of the polynomial approximation, we look for stencil including k cells for $k-1$ degree of the polynomial. The cells can be on both sides around the corresponding cell I_i for which we form the reconstruction. Let say that there are r cells to the left and s cells to the right, where r, s are nonnegative and it holds

$$r + s + 1 = k. \tag{1.21}$$

Then the stencil is the following set

$$S(i) = \{I_{i-r}, \ldots, I_{i+s}\}, \tag{1.22}$$

assuming each cell between I_{2-k} and I_{N+k-1} is available. Now we need an appropriate criterion how to choose the neighboring cell to prevent the situation that oscillations occur. We use the Newton divided difference according to [4]. For this purpose, we define the primitive function $V(x)$ of the solution u

$$V(x) = \int_{-\infty}^{x} u(\xi)\mathrm{d}\xi, \tag{1.23}$$

where the lower limit $-\infty$ in not important and can be replaced by any fixed number. We apply the divided differences of the function $V(x)$ to build the appropriate stencil. The 0–th degree divided difference for the primitive function V is defined by

$$V\left[x_{i-\frac{1}{2}}\right] = V\left(x_{i-\frac{1}{2}}\right), \tag{1.24}$$

where the value of $V(x)$ at the point $x_{i+\frac{1}{2}}$ is expressed as

$$V\left(x_{i+\frac{1}{2}}\right) = \sum_{j=-\infty}^{i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(\xi)\mathrm{d}\xi. \tag{1.25}$$

The cell average is substituted in the last expression and thus the relation between the primitive function $V(x)$ at the cell boundaries and the cell averages $\bar{U}_i$ is obtained

$$V\left(x_{i+\frac{1}{2}}\right) = \sum_{j=-\infty}^{i} \bar{U}_j \Delta x. \tag{1.26}$$

The last expression implies that if we know the cell averages $\bar{U}_i$ exactly then we can determine the values of the primitive function $V(x)$ at the cell boundaries exactly. Assume e, f to be nonnegative integer numbers then we can define the j–th degree divided differences

$$V\left[x_{i-\frac{1}{2}-e}, \ldots, x_{i-\frac{1}{2}+f}\right] = \frac{V\left[x_{i+\frac{1}{2}-e}, \ldots, x_{i-\frac{1}{2}+f}\right] - V\left[x_{i-\frac{1}{2}-e}, \ldots, x_{i-\frac{3}{2}+f}\right]}{x_{i-\frac{1}{2}+f} - x_{i-\frac{1}{2}-e}}, \tag{1.27}$$

where $e+f = j$. Using (1.26) we get the divided differences of $V(x)$ expressed by the cell averages $\bar{U}_i$. Let us show an example of the first degree divided difference of $V(x)$ for cell $I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$ (i.e. $e = 0$, $f = 1$)

$$V\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] = \frac{V\left[x_{i+\frac{1}{2}}\right] - V\left[x_{i-\frac{1}{2}}\right]}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} = \frac{V\left(x_{i+\frac{1}{2}}\right) - V\left(x_{i-\frac{1}{2}}\right)}{\Delta x} =$$

$$= \frac{\sum_{j=-\infty}^{i} \bar{U}_j \Delta x - \sum_{j=-\infty}^{i-1} \bar{U}_j \Delta x}{\Delta x} = \frac{\bar{U}_i \Delta x}{\Delta x} = \bar{U}_i. \tag{1.28}$$

We now discuss why the divided difference is appropriate for selection of the cells. It holds according to [4]

$$V\left[x_{i-\frac{1}{2}-e}, \ldots, x_{i-\frac{1}{2}+f}\right] = \frac{V^j(\xi)}{j!} \tag{1.29}$$

for some ξ inside $x_{i-\frac{1}{2}-e} < \xi < x_{i-\frac{1}{2}+f}$ and $e + f = j$, provided the primitive function $V(x)$ is smooth in the interval. If $V(x)$ is discontinuous at some point in the above interval then

$$V\left[x_{i-\frac{1}{2}-e}, \ldots, x_{i-\frac{1}{2}+f}\right] = O\left(\frac{1}{\Delta x^j}\right). \tag{1.30}$$

It implies that the divided difference measures the smoothness of the function $V(x)$ in the interval $\left(x_{i-\frac{1}{2}-e}, x_{i-\frac{1}{2}+f}\right)$, where $e + f = j$.

Let us remind that we wanted to build a reconstruction from the cell averages for a stencil S and we constructed Newton divided difference to decide what is the most appropriate stencil. The property of the divided difference mentioned above is very important because we look for a special type of interpolation polynomial. It is the Newton interpolation which is constructed just by the Newton divided differences. It means that if we find a set of such cells (i.e. the stencil) in which the primitive function $V(x)$ is the most smooth then we get optimal recovery.

A stencil $S_i = \{I_{i-r}, \ldots, I_{i+s}\}$ can be represented by a set of the boundary points $\left\{x_{i-\frac{1}{2}-r}, \ldots, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, \ldots, x_{i-\frac{1}{2}+s+1}\right\}$. We can immediately see that $e = r$, $f = s + 1$ and degree of the divided difference is $j = r + s + 1$ in (1.27). It implies that for each stencil $S_i = \{I_{i-r}, \ldots, I_{i+s}\}$ we can find Newton inporpolation polynomial $P_i(x)$ of degree k of the function $V(x)$ at points $\left\{x_{i-\frac{1}{2}-r}, \ldots, x_{i-\frac{1}{2}+s+1}\right\}$ as follows

$$P_i(x) = \sum_{j=0}^{k} V\left[x_{i-\frac{1}{2}-r}, \ldots, x_{i-\frac{1}{2}-r+j}\right] \cdot \prod_{m=0}^{j-1} \left(x - x_{i-\frac{1}{2}-r+m}\right), \tag{1.31}$$

where $j = r + s + 1$. Because $V(x)$ is the primitive function of the unknown solution $u(x)$ thus the derivative of (1.31) can be computed and we can finally express the Newton interpolation polynomial of k-th degree of the function $u(x)$

$$p_i(x) = \sum_{j=1}^{k} V\left[x_{i-\frac{1}{2}-r}, \ldots, x_{i-\frac{1}{2}-r+j}\right] \cdot \sum_{m=0}^{j-1} \prod_{l=0, l\neq m}^{j-1} \left(x - x_{i-\frac{1}{2}-r+l}\right), \tag{1.32}$$

where $j = r + s + 1$. Although this polynomial is determined by the divided difference of the primitive function $V(x)$, using (1.26) we know that we can express the divided difference using the cell averages $\bar{U}_i$ which we know.

Let us show how to find the most appropriate stencil $S(i)$, containing k neighboring cells for the i–th cell I_i using divided difference. We always start with the corresponding cell I_i, i.e. $S(i) = \{I_i\}$. It is enough to satisfy first order of the method. To get higher order, we have to add one of the neighbour cells I_{i-1} or I_{i+1} as mentioned above. But we can finally decide computationally which one. Smaller value of the divided difference implies the smoother function thus we avoid the oscillations. We compare the divided difference for the stencil $S^1(i) = \{I_{i-1}, I_i\}$ and $S^2(i) = \{I_i, I_{i+1}\}$. If it holds

$$\left|V\left[x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]\right| < \left|V\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}\right]\right|, \tag{1.33}$$

we add the left neighbour cell I_{i-1}. Otherwise we add the right neighbour I_{i+1}. If we use the chosen stencil to express the reconstruction (1.32) at this moment, we would get method of the second order. But to obtain method of even higher order we proceed by the same way adding next cells until we reach k cells in the stencil $S(i)$.

Let us assume that we have determined the stencil with needed number of the cells according to the required order of the method using the divided differences. For such a stencil we can now find the Newton interpolation (1.32) and thus we can reconstruct the unknown function u at each point.

The last task is to calculate the polynomial $p_i(x)$ at the cell boundaries for the corresponding cell I_i. We do it by substituting the points $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ into the interpolation $p(x)$

$$U^-_{i+\frac{1}{2}} = p_i\left(x_{i+\frac{1}{2}}\right), \forall i = 1, \ldots, N, \tag{1.34}$$

$$U^+_{i-\frac{1}{2}} = p_i\left(x_{i-\frac{1}{2}}\right), \forall i = 1, \ldots, N. \tag{1.35}$$

Another and for uniform grid better way is described, in [4], how to get the interpolation polynomial. We can proceed similarly to [4] to obtain the following expression for the general approximate polynomial of arbitrary degree

$$p_i(x) = \sum_{m=0}^{k} \sum_{j=0}^{m-1} \bar{u}_{i-r+j} \Delta x_{i-r+j} \left(\frac{\sum_{l=0, l\neq m}^{k} \prod_{q=0, q\neq m, l}^{k} \left(x - x_{i-r+q-\frac{1}{2}}\right)}{\prod_{l=0, l\neq m}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}\right)} \right). \tag{1.36}$$

The values of the polynomial at $x = x_{i+\frac{1}{2}}$ are expressed by

$$U^-_{i+\frac{1}{2}} = p_i\left(x_{i+\frac{1}{2}}\right) =$$

$$= \sum_{j=0}^{k-1} \left(\sum_{m=j+1}^{k} \frac{\sum_{l=0,l\neq m}^{k} \prod_{q=0,q\neq m,l}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{l=0,l\neq m}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \bar{U}_{i-r+j}. \tag{1.37}$$

The last equation is simplified and thus the constants c_{rj} are established as follows

$$c_{rj} = \left(\sum_{m=j+1}^{k} \frac{\sum_{l=0,l\neq m}^{k} \prod_{q=0,q\neq m,l}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{l=0,l\neq m}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j}, \tag{1.38}$$

which we can rewrite for a uniform grid by

$$c_{rj} = \sum_{m=j+1}^{k} \frac{\sum_{l=0,l\neq m}^{k} \prod_{q=0,q\neq m,l}^{k} (r-q+1)}{\prod_{l=0,l\neq m}^{k} (m-l)}. \tag{1.39}$$

Then we obtain the final relation for the approximate boundary values of the function $u(x)$

$$U_{i+\frac{1}{2}}^{-} = p_i \left(x_{i+\frac{1}{2}} \right) = \sum_{j=0}^{k-1} c_{rj} \bar{U}_{i-r+j}. \tag{1.40}$$

We can find a table of these constants c_{rj} for the uniform grid in [4], for order of accuracy between $k = 1$ and $k = 7$.

We have to note that the expressions above are related to the right boundary point $x_{i+\frac{1}{2}}$ in I_i. If we want to determine the value for $x_{i-\frac{1}{2}}$ in the cell I_i, there exist a little modified relation for the approximate value $U_{i-\frac{1}{2}}^{+}$

$$U_{i-\frac{1}{2}}^{+} = p_i \left(x_{i-\frac{1}{2}} \right) = \sum_{j=0}^{k-1} \tilde{c}_{rj} \bar{U}_{i-r+j}, \tag{1.41}$$

where $\tilde{c}_{rj} = c_{r-1,j}$. At this time, we are able to construct an interpolation polynomial for any data, to get the approximation of the flux and the very last step is to formulate (1.13). We concentrate on solving of ordinary differential equation according to the time variable in the next section.

1.4 Time Discretization

We described the basic idea of ENO method in the previous section. We obtained an approximation of the unknown function $u(x,t)$ in each cell I_i and we computed the required values of the solution at the cell boundary points $U_{i-\frac{1}{2}}^{\pm}$ so far. The very last step is to form the equation (1.13), i.e. solve

differential equation according to the time variable t. We will be interested in the Euler and Runge-Kutta methods.

Let us discretize the time interval $(0, T)$ such that

$$0 = t_1 < t_2 < \ldots < t_n < t_{n+1} < \ldots < t_{K-1} < t_K = T. \tag{1.42}$$

Moreover, we define step Δt as follows

$$\Delta t = t_{n+1} - t_n, \quad \forall n = 1, 2, \ldots, K. \tag{1.43}$$

We look for the approximate solution u of the following

$$\frac{\mathrm{d}\bar{U}_i(t)}{\mathrm{d}t} = -\frac{1}{\Delta x}\left(\bar{f}_{i+\frac{1}{2}} - \bar{f}_{i-\frac{1}{2}}\right), \tag{1.44}$$

where the right side is obtained by ENO approximation.

Euler method

First, let us focus on Euler method which is method of the first order accuracy. The following equality is obtained by integration of the relation (1.44) with respect to time over the interval $[t_n, t_{n+1}]$

$$\int_{t_n}^{t_{n+1}} \frac{\mathrm{d}\bar{U}_i(t)}{\mathrm{d}t}\mathrm{d}t = -\frac{1}{\Delta x}\int_{t_n}^{t_{n+1}} \left(\bar{f}_{i+\frac{1}{2}} - \bar{f}_{i-\frac{1}{2}}\right)\mathrm{d}t =$$

$$= -\frac{\Delta t}{\Delta x}\left(\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \bar{f}_{i+\frac{1}{2}}\mathrm{d}t - \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \bar{f}_{i-\frac{1}{2}}\mathrm{d}t\right),$$

since $\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \bar{f}_{i+\frac{1}{2}}\mathrm{d}t$ is an average of the flux, which we denote $\bar{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$, thus

$$\bar{U}_i(t_{n+1}) = \bar{U}_i(t_n) - \frac{\Delta t}{\Delta x}\left(\bar{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \bar{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right).$$

We approximate $\bar{U}_i(t_{n+1})$ and $\bar{U}_i(t_n)$ by $\bar{U}_i^{n+1}$ and $\bar{U}_i^n$ then we get the explicit Euler method written in the following form

$$\bar{U}_i^{n+1} = \bar{U}_i^n - \frac{\Delta t}{\Delta x}\left(\bar{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \bar{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right). \tag{1.45}$$

Runge – Kutta methods

The class TVD Runge–Kutta methods is suitable to solve initial value problem of ordinary differential equations with higher accuracy. In connection to the ENO method, we want to apply Runge – Kutta method to solve the folloving ODE

$$\frac{\mathrm{d}\bar{U}_i(t)}{\mathrm{d}t} = L\left(\bar{U}_i(t)\right), \tag{1.46}$$

where $L\left(\bar{U}_i(t)\right)$ represents the right side of the equation (1.44) which is obtained by ENO method. ENO method, described in the previous section, provides numerical solution of high order accuracy in space. Euler method is method of the first order. The Runge – Kutta method is applied to get higher accuracy also for approximation in time. Special Runge – Kutta method is used for hyperbolic conservation laws, i.e. total variation diminishing (TVD) Runge – Kutta methods which maintains stability of the numerical solution. The theory about the TVD and non–TVD Runge – Kutta methods are studied for instance in [19, 4, 20].

Second order TVD Runge–Kutta method, [4], is given by

$$\begin{aligned} \bar{U}_i^{(1)} &= \bar{U}_i^n + \Delta t L\left(\bar{U}_i^n\right) \\ \bar{U}_i^{n+1} &= \frac{1}{2}\bar{U}_i^n + \frac{1}{2}\bar{U}_i^{(1)} + \frac{1}{2}\Delta t L\left(\bar{U}_i^{(1)}\right). \end{aligned} \tag{1.47}$$

The third order TVD Runge–Kutta method is expressed

$$\begin{aligned} \bar{U}_i^{(1)} &= \bar{U}_i^n + \Delta t L\left(\bar{U}_i^n\right) \\ \bar{U}_i^{(2)} &= \frac{3}{4}\bar{U}_i^n + \frac{1}{4}\bar{U}_i^{(1)} + \frac{1}{4}\Delta t L\left(\bar{U}_i^{(1)}\right) \\ \bar{U}_i^{n+1} &= \frac{1}{3}\bar{U}_i^n + \frac{2}{3}\bar{U}_i^{(2)} + \frac{2}{3}\Delta t L\left(\bar{U}_i^{(2)}\right). \end{aligned} \tag{1.48}$$

1.5 Algorithm

In this section, we summarize in a brief overview the algorithm used to implement ENO method for solving a hyperbolic partial differential equation (1.1) with an initial condition (1.2).

1. **Discretization**
   Let us consider the uniform discretization for all calculations with constant size of cell Δx. We discretize the interval $[a, b]$ to get the cell

boundaries $x_{i+\frac{1}{2}}$ and the cell centers x_i, exactly according to the description in the section 1.1. At the same time, the cell averages of the initial condition (1.2) are determined using the expresstion (1.9) for each cell I_i in the interval $[a, b]$. Next we discretize the time interval $[0, T]$ with constant step Δt as described in 1.4.

The following steps proceed until the time T is reached.

2. **Selection of stencil for method of k–th order**
   This problem is described in details in the section 1.3. We have to determine the stencil S_i for each cell I_i. The stencil includes k cells for the method of the order k. In the process, we always start with stencil including the corresponding cell I_i, i.e. as if we get the first order of the method. Next we consider two possible stencils, one $S_i^1 = \{I_{i-1}, I_i\}$ by adding left cell and second $S_i^2 = \{I_i, I_{i+1}\}$ adding the right cell.

   The relation (1.27) is used to compute the divided difference and we choose such a stencil which has smaller value of the divided difference according to (1.33). The second order of the method is thus reached.

   Next we select between $S_i^1 = \{I_{i-2}, I_{i-1}, I_i\}$ and $S_i^2 = \{I_{i-1}, I_i, I_{i+1}\}$ if we added I_{i-1} in the last step. Or we select between $S_i^1 = \{I_{i-1}, I_i, I_{i+1}\}$ and $S_i^2 = \{I_i, I_{i+1}, I_{i+2}\}$. We again compute the divided difference for them and use the condition (1.33). Third order of the method is obtained now. We proceed in a similar way up to reaching k-th order of the method.

3. **Computing of polynomial approximation in stencil**
   In the previous step, the stencil was determined, i.e. the interval in a space domaine where reconstruction of function u is computed. We use (1.32) to obtain the polynomial $p_i(x)$ of the degree at most $k-1$ for each cell I_i.

4. **Determination of cell boundaries**
   The values of polynomial $p_i(x)$ at the points $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ are the approximations of the function $u(x)$ at the cell boundaries $U^+_{i-\frac{1}{2}}$ and $U^-_{i+\frac{1}{2}}$ according to (1.34) and (1.35). Second possibility is to directly use expressions (1.40) and (1.41). Let us remind that we can skip the step 3 if we use (1.40) and (1.41).

5. **Approximation of flux**
   Let us choose (1.14) or (1.15) as the scheme for approximation of the

numerical flux and we use (1.12) to compute $\bar{f}_{i+\frac{1}{2}}$ for all i.

6. **Computing of the next time step**
   The evolution problem (1.44) is solved in time using Euler method (1.45) or Runge – Kutta method (1.47) or (1.48).

   Approximation of the solution was computed for t_{n+1} and the steps $2-6$ follows.

1.6 Numerical Results

ENO method is applied to scalar linear partial differential equation in this section. We are interested in quality of the numerical solution depending on the order of the method. Let us study the transport equation with transport velocity c

$$u_t + cu_x = 0, \quad x \in [a,b], t \in (0,T) \tag{1.49}$$

with an initial condition

$$u(x,0) = u_0(x), \quad x \in [a,b]. \tag{1.50}$$

Outflow boundary conditions are applied in the experiments which could be represented by Neumann boundary condition. In this thesis, we focus on properties of the ENO methods and quality of the numerical solution, but theory about boundary condition can be found e.g. in [19] in details. The equation (1.49) has analytical solution which is briefly calculated before we present the numerical experiments. We rewrite the equation (1.49) as an inner product of two vectors

$$(1,c) \cdot (u_x, u_y) = 0,$$

where (u_x, u_y) is gradient of the function u. The left side of the equality represents the derivative of the function u with respect to the vector $v = (1,c)$, i.e.

$$\frac{\partial u}{\partial v} = 0.$$

Zero derivative means that the function u is constant along the lines with the direction vector $(1,c)$. Therefore, the solution of the transport equation has the following form

$$u(x,t) = u_0(x - ct). \tag{1.51}$$

The solution is constant in the directions of the vector $(1,c)$ and it moves with speed c in the time t. If the constant c is positive then the solution

travelles to the right and vice versa. We talk about right or left travelling wave, respectively.

Now, we turn our attention to the numerical experiments. Specifically, let us solve the transport equation (1.49) for $[a, b] = [-18, 18]$. We use the equidistant grid between points $x_{i+\frac{1}{2}}$, Godunov flux (1.14) in the ENO method and Euler method for the equation in time. Since we know the analytical solution, we can compute the error of the numerical approximation. Assume that $\hat{u}(x_i, t_n)$ is the average of the exact solution and $\bar{U}_i^n$ is the average of the numerical solution of the transport equation at the cell centers x_i and at the time t_n. The error of the numerical solution is given by

$$\Delta x \sum_i \left| \bar{U}_i^n - \hat{u}(x_i, t_n) \right|. \tag{1.52}$$

Experiment 1

Let us consider an example using the following initial condition which is also showed in the Figure 1.3

$$u(x, 0) = \begin{cases} -1 & \text{if} \quad x < -1 \\ 1 & \text{if} \quad -1 \leq x \leq 1 \\ -1 & \text{if} \quad 1 < x. \end{cases}$$

We choose $\Delta x = 0.05$, the time step $\Delta t = 0.06$ and the speed $c = 1$. We can compare the numerical and analytical solution at the time $t = 0.3$ in the Figure 1.4 for ENO method of the first and second order. We can see that the numerical solution oscillates strongly after only five time steps. The reason is in the stability of the method. Each finite volume method has to satisfy the so called Courant-Friedrichs-Lewy condition (CFL condition) for stability of the numerical solution. The theory of the CFL condition and stability is described in details in [12, 19]. We have to fulfil the following condition for the transport equation, see e.g. [12]

$$|c| \frac{\Delta t}{\Delta x} \leq 1. \tag{1.53}$$

In case of general hyperbolic equation (1.1), CFL condition has the form

$$\max_u |f'(u)| \frac{\Delta t}{\Delta x} \leq 1. \tag{1.54}$$

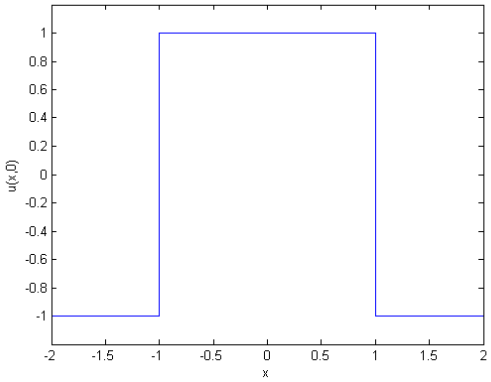

Figure 1.3: Initial condition.

We consider satisfying CFL condition in the rest of the experiments in this section. Figures 1.5 and 1.6 show numerical solution of the transport equation (1.49) for $\Delta x = 0.05$, $\Delta t = 0.02$, $c = 1$, $T = 4$. The graphs confirm that the numerical solution does not oscillate in this case because we fulfil CFL condition. Numerical solution of ENO method is compared with the analytical solution in each graph. ENO method of the first order (i.e. using constant reconstruction) is very inaccurate according to the Figure 1.5 (left). But the quality of the numerical solution rises as the order of the method increases.

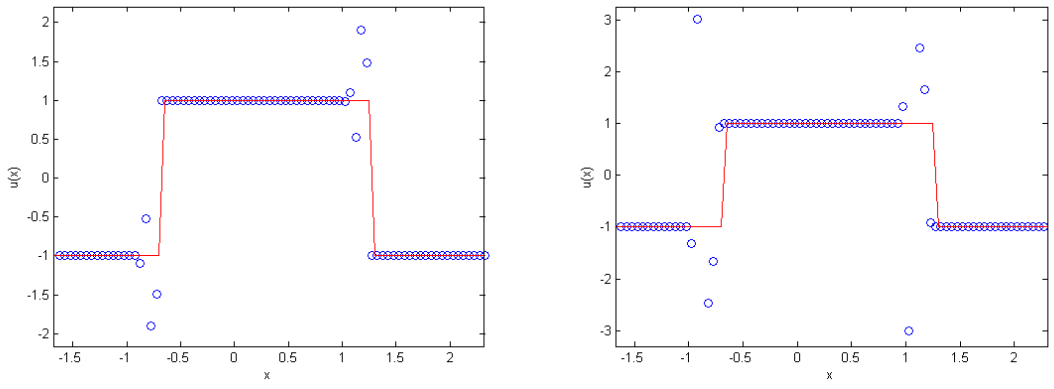

Figure 1.4: Numerical (circles) and analytical (solid line) solution after 5 time steps. ENO method of the first (left) and second (right) order is used for $\Delta x = 0.05$, $\Delta t = 0.06$.

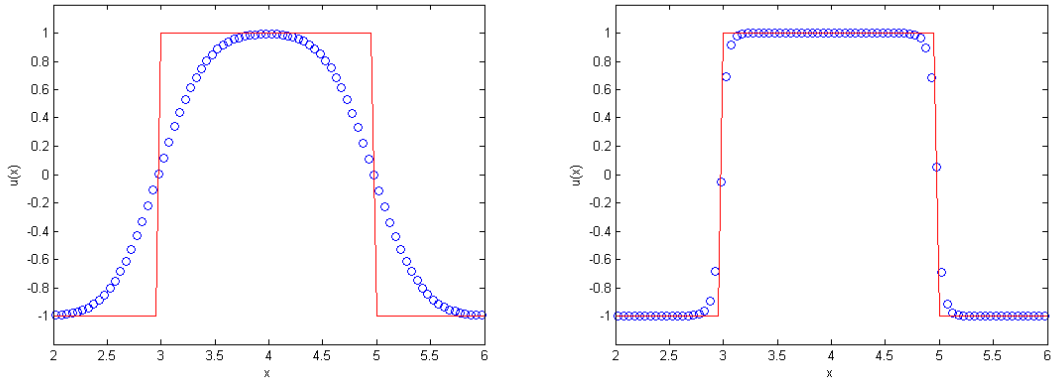

Figure 1.5: Numerical (circles) and analytical (solid line) solution after 200 time steps. ENO method of the first (left) and second (right) order is used for $\Delta x = 0.05$, $\Delta t = 0.02$.

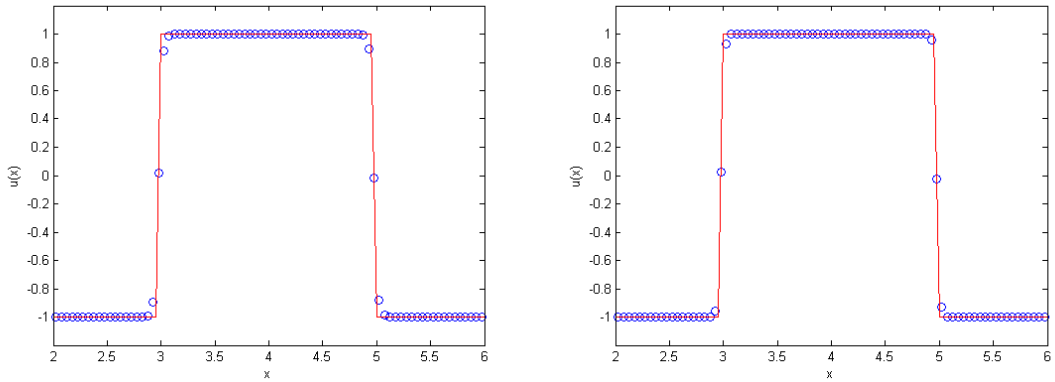

Figure 1.6: Numerical (circles) and analytical (solid line) solution after 200 time steps. ENO method of the third (left) and fourth (right) order is used for $\Delta x = 0.05$, $\Delta t = 0.02$.

The error of the numerical solution (1.52) is entered in the Table 1.1. It confirms the behaviour of the numerical solution in the previous figures. The value of the error decreases with increasing order of the method. The biggest decrease of the error is between ENO method of the first and second order. While the error goes down much slower for higher order of the method.

According to the above example, we would choose ENO method of the fourth order, if we are interested only in numerical solution with the smallest error. But the efficiency of the algorithm can be high using very small Δx and Δt and it rises for higher order of the accuracy of the method. The efficiency

of the algorithm is displayed in the Table 1.2 for the previous example after 200 steps. If we thus want to use ENO method of the fourth order, we should not forget to higher computational time. Perhaps it is better to use method of the third order because the error for ENO scheme of the third and fourth order is about the same value and method of the third order is bit faster.

k	error
1	1.1099
2	0.1892
3	0.1243
4	0.1119

Table 1.1: Error of the numerical solution after 200 time steps for $\Delta x = 0.05$, $\Delta t = 0.02$, $c = 1$.

k	time [s]
1	3.2
2	4.3
3	6.4
4	7.4

Table 1.2: Efficiency of the algorithm of the numerical solution after 200 time steps for $\Delta x = 0.05$, $\Delta t = 0.02$, $c = 1$.

We now choose more fine grid $\Delta x = 0.025$ and we are interested if the error of the numerical solution decreases. The results are displayed in the Figures 1.7, 1.8 at time $T = 4$ for $\Delta t = 0.01$. According to the graphs, the numerical solution is improved for each order of the method compared to the previous experiment on coarse grid. The values of the error in the Table 1.3 confirm that the magnitude of the error is half than the error on coarse grid $\Delta x = 0.05$. However the algorithm works about four times longer.

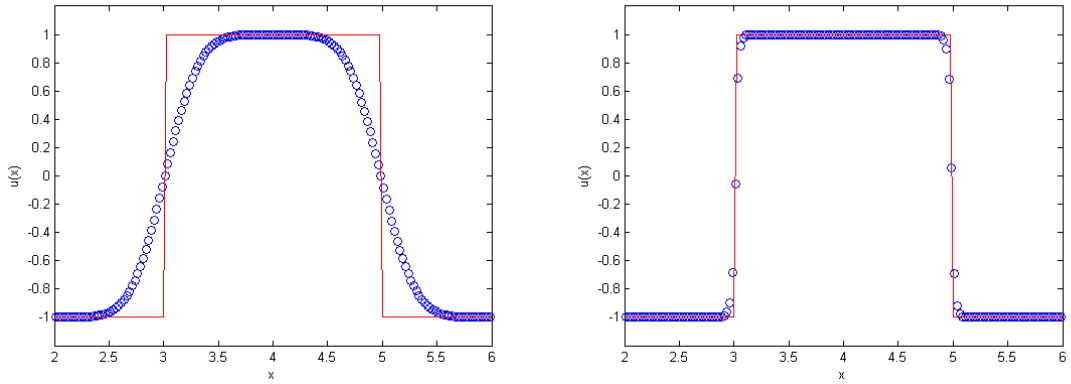

Figure 1.7: Numerical (circles) and analytical (solid line) solution after 400 time steps. ENO method of the first (left) and second (right) order is used for $\Delta x = 0.025$, $\Delta t = 0.01$.

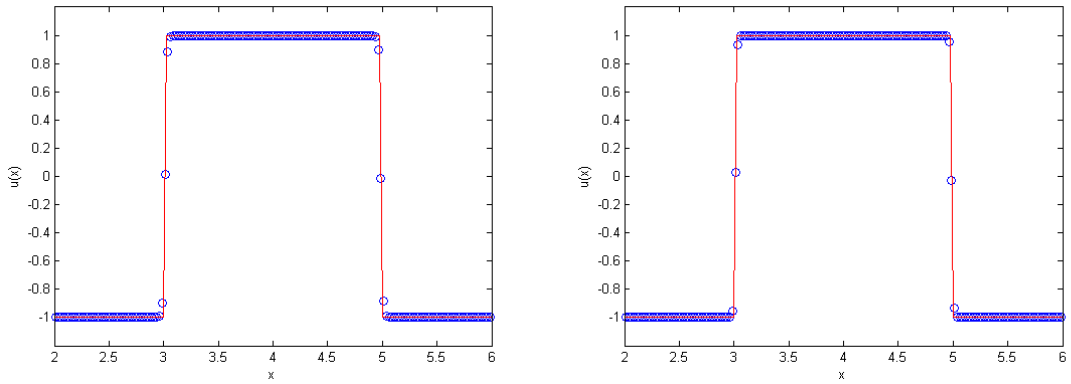

Figure 1.8: Numerical (circles) and analytical (solid line) solution after 400 time steps. ENO method of the third (left) and fourth (right) order is used for $\Delta x = 0.025$, $\Delta t = 0.01$.

k	error	time [s]
1	0.7833	11.6
2	0.0949	17.3
3	0.0622	25.5
4	0.0560	29.6

Table 1.3: Error of the numerical solution and efficiency of the algorithm after 400 time steps for $\Delta x = 0.025$, $\Delta t = 0.01$, $c = 1$.

Experiment 2

It has been found out in the previous experiment that ENO method of third or fourth order provides acceptable results. We are interested whether the accuracy using such method is higher than using the classical finite difference methods. Finite difference methods are described in details in [12]. The Figure 1.9 compares ENO method of the first and second order with finite difference method of the first order for $\Delta x = 0.05$, $\Delta t = 0.02$ and $c = 1$. The first order of the ENO method and finite difference method gives the same quality of the numerical solution.

Finite difference method of the second order is more accurate than ENO method of the first order but it oscillates near discontinuity, see Figure 1.10. Moreover, already numerical solution of the ENO method of the second order is closer to the analytical solution than finite difference method of the second order.

Let us consider now the so called Harten–Zwas method. It is a method which uses second order finite difference method around points where solution is smooth and it switches to the first order otherwise, because first order does not product oscillations. We again compare this method with ENO method of first and second order in the Figure 1.11. The results are similar to the previous situation except that Harten – Zwas method does not product oscillations, i.e. the switching method improves ENO method of the first order. But the quality of the ENO method of the second order is higher than Harten–Zwas method. It thus appears that ENO method of the second and higher order provides numerical solution of the transport equation with smaller error than finite difference methods.

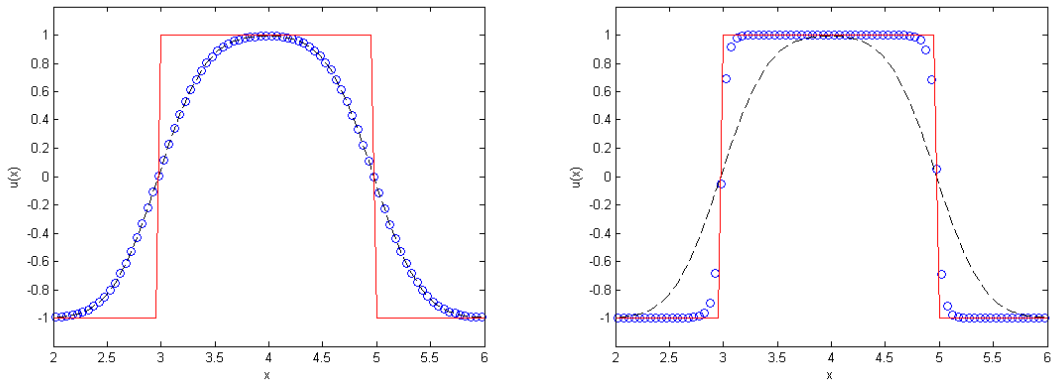

Figure 1.9: Numerical (circles) and analytical (solid line) solution at time $t = 4$. ENO method of the first (left) and second (right) order is compared with finite difference method of the first order. $\Delta x = 0.05$, $\Delta t = 0.02$.

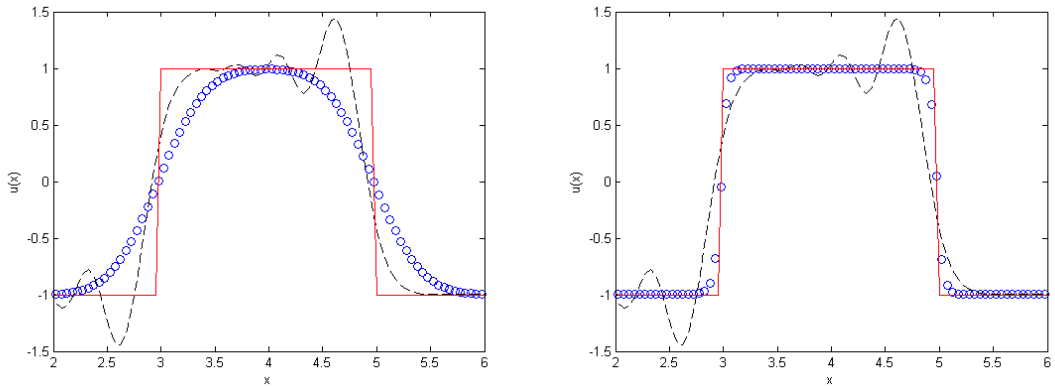

Figure 1.10: Numerical (circles) and analytical (solid line) solution at time $t = 4$. ENO method of the first (left) and second (right) order is compared with finite difference method of the second order. $\Delta x = 0.05$, $\Delta t = 0.02$.

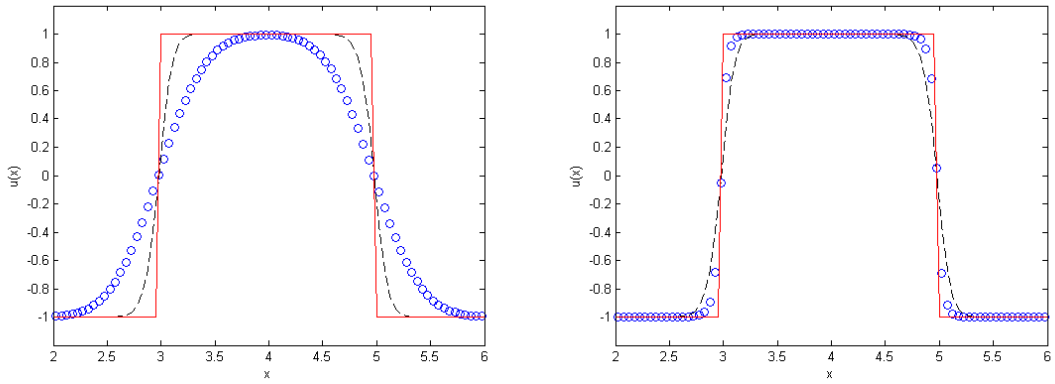

Figure 1.11: Numerical (circles) and analytical (solid line) solution at time $t = 4$. ENO method of the first (left) and second (right) order is compared with the Harten – Zwas method. $\Delta x = 0.05$, $\Delta t = 0.02$.

Experiment 3

We studied the quality of the numerical solution using discontinuous initial condition so far. We are now interested in more complicated initial condition. We use both continuous and discontinuous initial condition, see Figure 1.12. Let us consider $\Delta x = 0.05$, $\Delta t = 0.02$, $c = 1$ and $T = 8$.

ENO method of the first order is applied in the Figure 1.13. The numerical solution is compared with the analytical solution. The numerical solution behaves similarly to the previous examples. The shape of the numerical so-

lution does not quite fit to the analytical solution because of the diffusion effect.

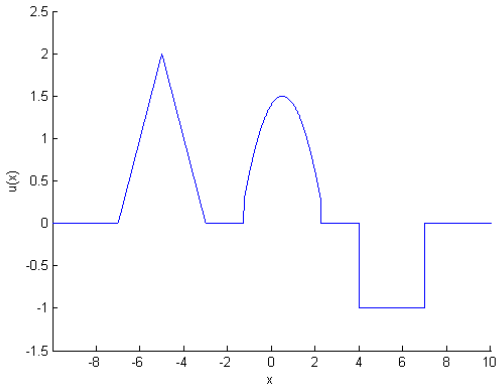

Figure 1.12: Initial condition.

If we use ENO scheme of the second order in the Figure 1.14 (left), we get correction of the numerical solution compared to the first order. The Table 1.4 with the error (1.52) confirms this improvement. The numerical solution converges to the analytical solution in case of the discontinuous data. But another problem appears for continuous data as we can see in the figure. This negative effect for continuous data is fixed using smaller value of the step Δt as it is showed in the right graph of the Figure 1.14. But more time steps cause that the quality of the numerical solution decreases for the discontinuous data.

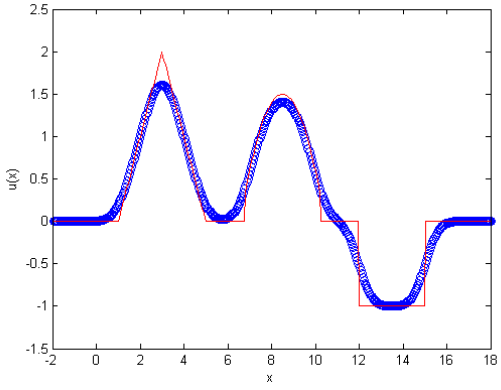

Figure 1.13: Initial condition (left) and numerical solution (circles) compared with analytical solution (solid line) at time $t = 8$ (right). ENO method of the first order is used for $\Delta x = 0.05$, $\Delta t = 0.02$.

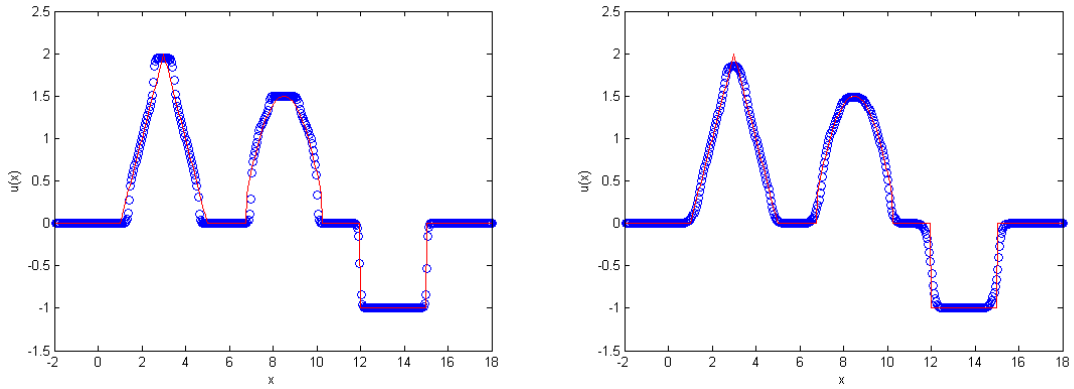

Figure 1.14: Numerical (circles) and analytical (solid line) solution at time $t = 8$. ENO method of the second order is used for $\Delta x = 0.05$, $\Delta t = 0.02$ (left) and $\Delta t = 0.005$ (right).

Figures 1.15 and 1.16 show numerical results using ENO method of the third order for various time step Δt. The structure of the numerical solution behaves similarly to the method of second order, i.e. the quality of the numerical solution increases with decreasing step Δt for continuous data but it is vice versa for the discontinuous data. The figures show that we have to be careful in case of non–smooth condition. The reason is that the numerical solution tends to almost oscillate for bigger time step Δt around the point where the initial condition is non–smooth. This oscillations are not evident from to the Table 1.4 because the error is the smallest for the method of the third order.

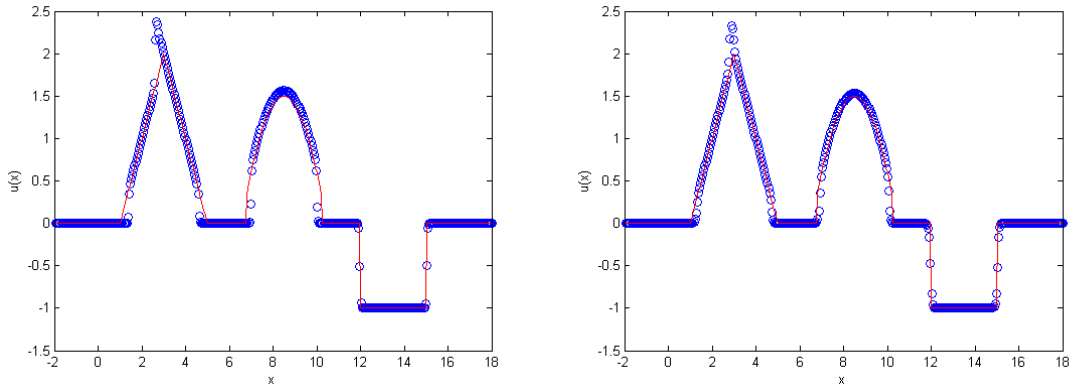

Figure 1.15: Numerical (circles) and analytical (solid line) solution at time $t = 8$. ENO method of the third order is used for $\Delta x = 0.05$, $\Delta t = 0.02$ (left) and $\Delta t = 0.01$ (right).

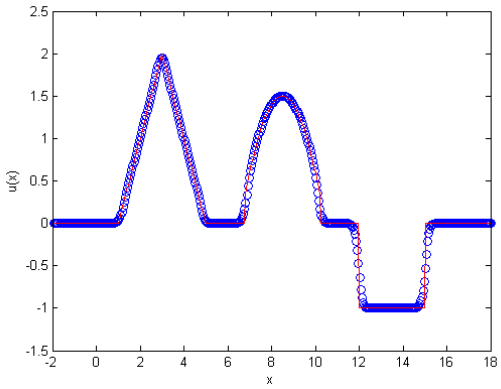

Figure 1.16: Numerical (circles) and analytical (solid line) solution at time $t = 8$. ENO method of the third order is used for $\Delta x = 0.05$, $\Delta t = 0.001$.

k	error			
	$\Delta t = 0.02$	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
1	1.7618	2.1389	2.3128	2.4458
2	0.7612	0.5845	0.5855	0.6357
3	0.7654	0.4600	0.3470	0.3375

Table 1.4: Error of the numerical solution at time $t = 8$ for $\Delta x = 0.05$, $c = 1$.

We tested ENO method applied on transport equation in the previous examples. The results showed that ENO method products more accurate numerical solution than classical finite difference methods. But the quality of the numerical solution for ENO scheme varies for continuous and discontinuous initial condition depending on size of Δx and Δt. From the given experiments, it follows that ENO scheme of the third order can provide optimal approximation of the solution of transport equation for both continuous and discontinuous inital condition.

Chapter 2

ENO schemes in two dimensional space

Since the class of essentially non–oscillatory schemes for hyperbolic partial differential equations and systems has been constructed, there exists few extensions to multidimensional problems. They are mainly developed for very regular meshes. ENO methods provide this extension and they are applied especially on triangular mesh in 2D.

In this chapter, we focus on hyperbolic partial differential equation in the form

$$u_t(x,y,t) + f_x(u(x,y,t)) + g_y(u(x,y,t)) = 0, \tag{2.1}$$
$$u(x,y,0) = u_0(x,y)$$

where $[x,y] \in \Omega \subset \mathbf{R} \times \mathbf{R}, t \in [0,T]$. We are interested in results of both linear and nonlinear equation but we apply ENO method also to system of equations. In what follows, we introduce ENO schemes on triangular mesh, we find integral formulation and we show how to find the reconstruction of the cell averages. The theory about ENO methods in two dimensional space can be found e.g. in [2, 3, 5, 10].

2.1 Discretization and Finite Volume Method

The partial differential equation (2.1) is defined on a domain $\Omega \subset \mathbf{R} \times \mathbf{R}$ which is discretized for the purpose of the numerical method. We are interested in the triangular mesh.

Let us consider a triangulation $\mathcal{T}$ of the domain Ω. Generally we require each two triangles to have empty intersection. A triangle T_i stands for one cell and the cell center is understood to be the barycenter of each such cell.

We integrate (2.1) over the triangle T_i to get the integrated form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{T_i} u\mathrm{d}\Omega = -\int_{T_i}(f_x + g_y)\mathrm{d}\Omega. \tag{2.2}$$

Similarly to 1D problem, the cell averages are denoted by the formula

$$\bar{u}_i = \frac{1}{|T_i|}\int_{T_i} u\mathrm{d}\Omega. \tag{2.3}$$

Now we substitute (2.3) into the expression (2.2) and the Green theorem is used to get

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{u}_i(t) = -\frac{1}{|T_i|}\oint_{\partial T_i}(f(u)\,n_x + g(u)\,n_y)\mathrm{d}\ell. \tag{2.4}$$

The line integral on the right side express the flux through whole boundary ∂T_i of the triangle T_i. We replace it by the sum of the integrals over each side of the corresponding triangle

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{u}_i(t) = -\frac{1}{|T_i|}\sum_{j=1}^{3}\int_{\Gamma_i^j}\left(f(u)\,n_x^j + g(u)\,n_y^j\right)\mathrm{d}\ell, \tag{2.5}$$

where $\mathbf{n}^j = \left(n_x^j, n_y^j\right)$ is a unit vector of the outer normal vector to the j-th side Γ_i^j of the triangle T_i. Last step is to approximate the numerical flux which gives us the equation in the following form

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{U}_i(t) = -\frac{1}{|T_i|}\sum_{j=1}^{3}\left|\Gamma_i^j\right| h_j\left(U^{in}, U^{out}, \mathbf{n}\right), \tag{2.6}$$

where $\left|\Gamma_i^j\right|$ is the length of the corresponding side of the triangle T_i, the values u^{in}, u^{out} are obtained by the reconstruction process described in the next section which does not provide the exact values of the integrals. Therefore $\bar{U}_i(t)$ are approximations of cell averages $\bar{u}_i(t)$ and U^{in}, U^{out} are approximations of u^{in}, u^{out}.

For example, we can use the following definitions for the function h, called Van Leer

$$h^{VL}\left(U^{in}, U^{out}, \mathbf{n}\right) = \frac{1}{2}(P\left(U^{in}, \mathbf{n}\right) + P\left(U^{out}, \mathbf{n}\right) -$$

$$-\left|\frac{\partial P}{\partial u}\left(\frac{U^{in}+U^{out}}{2}, \mathbf{n}\right)\right|\left(U^{out} - U^{in}\right)). \tag{2.7}$$

$P(u, \mathbf{n}) = n_x f(u) + n_y g(u)$ where $\mathbf{n} = (n_x, n_y)$ is a unit vector of the outer normal.

Similarly to one dimensional case, solution of the problem can not be considered in classical but in weak sense. Weak solution has to satisfy the following integral identity

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{T_i} u\mathrm{d}\Omega + \frac{1}{|T_i|}\int_{\partial T_i}\left(f\left(u\right)n_x + g\left(u\right)n_y\right)\mathrm{d}\ell = 0, \tag{2.8}$$

for every triangle T_i. The entropy conditions are used to get unique solution, see [12, 13, 7].

2.1.1 Polynomial Reconstruction

The reconstruction theory was created to get the values $U^{\pm}_{i+\frac{1}{2}}$ in such a way that we avoid to the oscillations as much as possible in 1D problems. The idea is the same in two dimensional space, we look for unknown values U^{in} and U^{out} to form (2.6). We want to choose an appropriate number of the neighbouring triangles of T_i again and an approximation is constructed over the selected triangles using the cell averages.

First we focus on a polynomial reconstruction using cell averages (2.3). We look for a polynomial $p\left(x,y\right)$ in the form

$$p\left(x,y\right) = \sum_{s=0}^{k}\sum_{i+j=s} a_{ij}x^i y^j, \tag{2.9}$$

where k is degree of the polynomial. It leads to method of $(k+1)$-th order using a polynomial of degree k. Let us assume that u is an unknown solution of a given partial differential equation. We then look for the polynomial $p\left(x,y\right)$ from (2.9) such that fulfill the cell average of the corresponding triangle T_i

$$\bar{U}_i = \bar{p}|_{T_i}. \tag{2.10}$$

Our aim is to find the stencil as a set of triangles which we after use for computing of the approximate polynomial. Now we show how we can choose the triangles to the stencil. We start with method of the first order. From the form of the polynomial (2.9), the number of necessary triangles is immediately found. Because first order of the method is required, we look for a polynomial of the degree 0, i.e. the cell averages are approximated by a constant

$$p\left(x,y\right) = a_{00}x^0y^0 = a_{00}.$$

Only one triangle is required to determine the unknown a_{00}. Naturally, we select the corresponding triangle T_i for which the reconstruction is searched.

We want to fulfil the cell average (2.10), i.e.

$$\bar{U}_i = \bar{p}|_{T_i} = \frac{1}{|T_i|}\int_{T_i} p\,(x,y) = \frac{1}{|T_i|}\int_{T_i} a_{00} = a_{00}\frac{1}{|T_i|}\int_{T_i} 1 = a_{00}.$$

Thus the reconstruction made by a constant polynomial for each triangle T_i is s.t. $p\,(x,y) = \bar{U}_i$. Let us now consider method of the second order, i.e. $k = 1$ and the polynomial has the following form

$$p\,(x,y) = a_{00} + a_{10}x + a_{01}y.$$

Certainly, three triangles are necessary to compute the values of the unknown coefficients. Consider that we have chosen three neighbouring triangles T_i, T_m, T_n (i.e. triangles that each of them has common side with at least one other triangle) such that one of them is the current triangle for which the reconstruction is computed. The cell averages has to be fulfilled as follows

$$\begin{array}{c} \frac{1}{|T_i|}\int_{T_i}(a_{00} + a_{10}x + a_{01}y)\mathrm{d}\Omega = \bar{U}_i, \\ \frac{1}{|T_m|}\int_{T_m}(a_{00} + a_{10}x + a_{01}y)\mathrm{d}\Omega = \bar{U}_m, \\ \frac{1}{|T_n|}\int_{T_n}(a_{00} + a_{10}x + a_{01}y)\mathrm{d}\Omega = \bar{U}_n. \end{array}$$

If we denote

$$\mathbf{A} = \begin{bmatrix} \frac{1}{|T_i|}\int_{T_i} 1\,\mathrm{d}\Omega & \frac{1}{|T_i|}\int_{T_i} x\,\mathrm{d}\Omega & \frac{1}{|T_i|}\int_{T_i} y\,\mathrm{d}\Omega \\ \frac{1}{|T_m|}\int_{T_m} 1\,\mathrm{d}\Omega & \frac{1}{|T_m|}\int_{T_m} x\,\mathrm{d}\Omega & \frac{1}{|T_m|}\int_{T_m} y\,\mathrm{d}\Omega \\ \frac{1}{|T_n|}\int_{T_n} 1\,\mathrm{d}\Omega & \frac{1}{|T_n|}\int_{T_n} x\,\mathrm{d}\Omega & \frac{1}{|T_n|}\int_{T_n} y\,\mathrm{d}\Omega \end{bmatrix}, \tag{2.11}$$

$$\mathbf{f} = \begin{bmatrix} \bar{U}_i \\ \bar{U}_m \\ \bar{U}_n \end{bmatrix}, \tag{2.12}$$

$$\xi = \begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \end{bmatrix}, \tag{2.13}$$

then we can formulate the system in the matrix form

$$\mathbf{A}\xi = \mathbf{f}.$$

By solving this system, the vector of the unknown coefficients is obtained and thus the reconstruction (2.9) is computed.

Now, we describe the algorithm how to choose appropriate triangles T_i, T_m and T_n belonging to stencil. As mentioned above, T_i is the corresponding triangle for which the reconstruction is computed and the three triangles

has to have at least one common side with the other triangle. There exists several options how to choose the triangles T_m, T_n. However, we choose a stencil according to [2] such that it minimizes

$$\sum_{i+j=1} |a_{ij}| . \tag{2.14}$$

Analogously, we can continue with method of higher orders. Generally, we can deduce the number M of the triangles in the stencil for method of the $(k+1)$-th order as follows

$$M = \frac{(k+1)(k+2)}{2}. \tag{2.15}$$

Then the condition, for selecting of the most appropriate set of the neighbouring triangles, is to minimize

$$\sum_{i+j=k} |a_{ij}| . \tag{2.16}$$

Certainly, we do not try all possible combinations of M neighbouring triangles round the triangle T_i. We always start with the first order of the method, i.e. stencil contain only the corresponding triangle T_i. We continue with method of the second order. We find the best stencil which minimizes (2.16) for $k = 1$ and then we add the other most appropriate triangles up to next order until we get the required order of the method.

Let us consider now total variation (TV). Total variation is for a differentiable function f defined by the expression (see [17])

$$V(f, \Omega) = \int_{\Omega} \|\nabla f(\mathbf{x})\| \, \mathrm{d}\Omega, \tag{2.17}$$

where $\|.\|$ denotes the L_2 norm. Total variation in 2D, similarly to divided difference in 1D, measures rate of oscillations, i.e. the triangles, for which is the value of the total variation minimum, are selected because the cell averages are expected to vary least. Sonar in [8] proves the connection between measurement of the oscillations using coefficients and total variation for polynomial recovery.

Although Jameson, see in [17], demonstrates an example of two functions in 2D such that one of them is more smooth than the other one. The value of total variation should be lower for the more smooth than for less smooth function because TV should measure the oscillation behaviour. But the opposite holds in the example in [17]. It is thus possible that total variation is not optimal criteria.

Let us suppose that we have already chosen the stencil using least sum of the coefficients or total variation. We thus compute the reconstruction (2.9) for the corresponding triangle. Reconstruction of the cell averages provides an approximation of the solution u of the hyperbolic PDE at every point, i.e. using polynomial reconstruction we can compute the unknown values U^{in} and U^{out}, for each side of the corresponding triangle T_i, and thus formulate (2.6). The values U^{out} can be simply computed as a value of the recovery polynomial in the center of the sides.

$$U^{out} = p\left(\mathbf{S}_i^j\right), \tag{2.18}$$

where $\mathbf{S}_i^j$ is the center of j-th side of the corresponding triangle T_i. Also an average of the values in the center and in the end points of the side of the triangle can be used or an average over the whole side. The values U^{in} are not determined separately, but we take it as an outer value U^{out} of the neighbouring triangle with the common corresponding side. Considering the described procedure, the values U^{in}, U^{out} can but need not to equal. As soon as we get these values we can formulate (2.6).

2.1.2 Reconstruction Using Radial Basis Functions

So far, we were interested in the approximation of the solution of the hyperbolic conservation law in 1D and 2D depending on the recovery using polynomials. Now, we ask if there exists another way of reconstruction which could provide more accurate numerical solution. In what follows, we present reconstruction of the solution from the cell averages using radial basis functions (RBF). In [6, 7], the authors mention that the polynomial recostruction is succefully used in one dimensional problem but a usage of RBFs is more appropriate for ENO approximation in multiple space dimensions.

Let us consider a function of one real variable

$$\varphi : [0, \infty) \longrightarrow \mathbf{R}.$$

We define the radial basis function $\phi : \mathbf{R}^N \to \mathbf{R}$ in the following way, (see e.g. [9, 16])

$$\phi_i(\mathbf{x}) = \varphi\left(\|\mathbf{x} - \mathbf{x}_i\|\right), \tag{2.19}$$

where $\mathbf{x}_i \subset \mathbf{R}^N$ is any point which is so–called the center and $\|.\|$ is a norm in $\mathbf{R}^N$. The following functions are examples of the function φ which generates radial basis functions

$$\varphi(r) = e^{-(\epsilon r)^2}, \tag{2.20}$$

$$\varphi(r) = r^2 \log r, \tag{2.21}$$

where $r = \|\mathbf{x}\|$ and ϵ is a parameter. We can read in [9, 14, 15] how is the value of the parameter ϵ important. The generators φ of the RBFs are often used in the approximation theory. We use these functions to construct the reconstruction as linear combination of the radial basis functions, but according to [7] we add linear polynomial. Then, we consider the approximation of cell averages in the following form

$$s(x,y) = \sum_{j=1}^{m} a_j \phi_j(x,y) + a_{m+1} + a_{m+2}x + a_{m+3}y. \tag{2.22}$$

If logarithmic generator $\varphi(r)$ of the radial basis function is used, we call the function (2.22) as thin plate spline. If the second generator (2.20) is applied, we call this type of approximation Gaussian spline. The idea of the thin plate splines is that the approximate function $s(x,y)$ minimizes the bending energy

$$\int\int_{\mathbf{R}^2} \left(s_{xx}^2 + 2s_{xy}^2 + s_{yy}^2\right) \mathrm{d}x\mathrm{d}y.$$

More details about radial basis functions, TPS and their application can be found in [18, 16, 9, 14].

The reason, why we build such an approximate function, is that the polynomial hold a global information and the role of the RBFs is local, it keeps the details. Hence, we suppose that RBFs improve the linear polynomial recovery. But we actually can not consider the order of the method using radial basis function. Further details about formal order of the accuracy can be found e.g. in [7].

Similarly to polynomial recovery problem, also function (2.22) with radial basis functions is used to reconstruct solution u of the PDE from cell averages. To determine the spline (2.22), $m+3$ conditions have to be required.

Let us assume $S = \{T_i^1, T_i^2, \ldots, T_i^m\}$ to be a stencil, where $T_i^l, l = 1, \ldots, m$ are the neighbouring triangles round the corresponding triangle T_i (including T_i). We require conservation of the cell averages for each triangle in the stencil, i.e.

$$\bar{U}_l = \bar{s}|_{T_i^l}, \quad l = 1, \ldots, m. \tag{2.23}$$

This conditions lead to m equations for $m+3$ unknowns, we thus have to add three other information. The additional conditions are considered according to [10, 18, 24, 25] as follows

$$\begin{aligned} \textstyle\sum_{j=1}^{m} \frac{1}{|T_i^j|} \int_{T_i^j} a_j \mathrm{d}\Omega = 0, \\ \textstyle\sum_{j=1}^{m} \frac{1}{|T_i^j|} \int_{T_i^j} a_j x \mathrm{d}\Omega = 0, \\ \textstyle\sum_{j=1}^{m} \frac{1}{|T_i^j|} \int_{T_i^j} a_j y \mathrm{d}\Omega = 0. \end{aligned} \tag{2.24}$$

This conditions are required for function s in (2.22) to have quare integrable second derivatives. The conditions are called orthogonality or side conditions. The problem to determine the approximate function (2.22) thus leads to solving the system

$$\begin{bmatrix} & & \\ \mathbf{B} & \mathbf{P} & \\ & & \\ \mathbf{P}^T & \mathbf{0} & \\ & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ a_{m+1} \\ a_{m+2} \\ a_{m+3} \end{bmatrix} = \begin{bmatrix} \bar{U}_i^1 \\ \bar{U}_i^2 \\ \vdots \\ \bar{U}_i^m \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.25}$$

with matrices

$$\mathbf{B} = \begin{bmatrix} \frac{1}{|T_i^1|}\int_{T_i^1}\phi_1(x,y)\,\mathrm{d}\Omega & \cdots & \frac{1}{|T_i^1|}\int_{T_i^1}\phi_m(x,y)\,\mathrm{d}\Omega \\ \frac{1}{|T_i^2|}\int_{T_i^2}\phi_1(x,y)\,\mathrm{d}\Omega & \cdots & \frac{1}{|T_i^2|}\int_{T_i^2}\phi_m(x,y)\,\mathrm{d}\Omega \\ \vdots & & \vdots \\ \frac{1}{|T_i^m|}\int_{T_i^m}\phi_1(x,y)\,\mathrm{d}\Omega & \cdots & \frac{1}{|T_i^m|}\int_{T_i^m}\phi_m(x,y)\,\mathrm{d}\Omega \end{bmatrix}, \tag{2.26}$$

$$\mathbf{P} = \begin{bmatrix} \frac{1}{|T_i^1|}\int_{T_i^1}1\mathrm{d}\Omega & \frac{1}{|T_i^1|}\int_{T_i^1}x\mathrm{d}\Omega & \frac{1}{|T_i^1|}\int_{T_i^1}y\mathrm{d}\Omega \\ \frac{1}{|T_i^2|}\int_{T_i^2}1\mathrm{d}\Omega & \frac{1}{|T_i^2|}\int_{T_i^2}x\mathrm{d}\Omega & \frac{1}{|T_i^2|}\int_{T_i^2}y\mathrm{d}\Omega \\ \vdots & & \vdots \\ \frac{1}{|T_i^m|}\int_{T_i^m}1\mathrm{d}\Omega & \frac{1}{|T_i^m|}\int_{T_i^m}x\mathrm{d}\Omega & \frac{1}{|T_i^m|}\int_{T_i^m}y\mathrm{d}\Omega \end{bmatrix}. \tag{2.27}$$

Notice that using $m = 2$ or $m = 1$, the matrix of the system (2.25) is singular, i.e. the matrix $\mathbf{P}$ would be of the type (1×3) or (2×3) and the whole matrix in (2.25) becomes singular. It follows that the minimum number of radial basis functions in (2.22) has to be three (i.e. $m \geq 3$) and it also implies that each stencil has to include at least three triangles if we consider reconstruction using splines.

Moreover Sonar in [7] claims that if we use reconstruction using thin plate spline or Gaussian spline, at least four tringles has to be used for stencil otherwise reconstruction degenerates into linear recovery. We try to confirm that in the numerical experiments in the following section.

If $m \geq 3$, the matrix of the system (2.25) is symmetric and positive semi-definite. Regularity of the matrix is proved e.g. in [13].

Let us suppose that all of the possible options of the stencil are available and the approximate functions were computed over each of the stencil. Next

step is to select the most appropriate stencil. First idea is to use similar rule as in case of polynomial recovery, i.e. we choose such a stencil for which the coefficients minimizes

$$\sum_{j=1}^{m} |a_j| + |a_{m+2}| + |a_{m+3}| \,. \tag{2.28}$$

But in any article, which is known to us, the authors do not apply this way of selection of the triangles. All of the authors use total variation, defined by (2.17), as limiter of the oscillations, i.e. the stencil with the least total variation is selected.

However, let us recall the example, mentioned in the previous subsection about polynomial reconstruction, where total variation had smaller value for less smooth data than for more smooth data. The theory about the selection of stencil can be found e.g. in [9].

Let us assume that the most appropriate stencil has been chosen by one of the criterion. The reconstruction using radial basis functions can be thus computed. Using the approximate function, we can determine the unknown values U^{in}, U^{out} as it is described in the previous subsection 2.1.1.

Special reconstruction using RBFs

We can consider one more way of the reconstruction s.t. only the combination of the radial basis functions is used. We can modify the definition (2.22) and look for the following recovery

$$\tilde{s}\,(x,y) = \sum_{j=1}^{m} a_j \phi_j\,(x,y)\,. \tag{2.29}$$

We do not have to require the additional conditions (2.24) to have regular matrix which is in this case in the form

$$\begin{bmatrix} \frac{1}{|T_i^1|}\int_{T_i^1}\phi_1\,(x,y)\;\mathrm{d}\Omega & \cdots & \frac{1}{|T_i^1|}\int_{T_i^1}\phi_m\,(x,y)\;\mathrm{d}\Omega \\ \frac{1}{|T_i^2|}\int_{T_i^2}\phi_1\,(x,y)\;\mathrm{d}\Omega & \cdots & \frac{1}{|T_i^2|}\int_{T_i^2}\phi_m\,(x,y)\;\mathrm{d}\Omega \\ \vdots & & \vdots \\ \frac{1}{|T_i^m|}\int_{T_i^m}\phi_1\,(x,y)\;\mathrm{d}\Omega & \cdots & \frac{1}{|T_i^m|}\int_{T_i^m}\phi_m\,(x,y)\;\mathrm{d}\Omega \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \bar{U}_i^1 \\ \bar{U}_i^2 \\ \vdots \\ \bar{U}_i^m \end{bmatrix}. \tag{2.30}$$

We consider the same way of the selection of the stencil as we use for spline reconstruction as well as the calculation of the values U^{in}, U^{out}. The main task is how many triangles to choose for the recovery using only radial basis function (2.29). We can expect improvement in case of linear reconstruction

with RBFs. But approximation using only RBFs is not connected to linear polynomial at all. We thus use the analogy to the spline recovery so that the same number of the triangles for stencil is considered as for the spline reconstruction.

Selection methods of triangles

In the theory of the reconstruction - using both polynomials and radial basis functions - we were only interested in number of triangles required for stencil and criterion (i.e. sum of the coefficients or total variation) to decide which triangle is more appropriate for reconstruction. We assumed that we selected some triangles from the neighborhood of the given triangle T_i. Now, we discuss which neighbouring triangles are generally considered to be tested as possible cells for stencil.

First way is to test all of the possible triangles but certainly it is very time–consuming for the algorithm. Authors like Sonar in [7] use special procedure. They consider only limited number of the triangles which can be included in the stencil.

In our work, two ways of the selection of the stencil are compared. The first way is selection of triangles one by one. The process is showed in the Figure 2.1. We start with the given triangle T_i, i.e. the stencil $S = \{T_i\}$. The triangle T_i has generally three neighbours, which are candidates for stencil. We apply procedure described in this section so that we select one of the three neighbours (e.g. T_l) using sum of the coefficients or total variation. Thus we get stencil including two triangles $S = \{T_i, T_l\}$, which is illustrated in the first graph in the Figure 2.1.

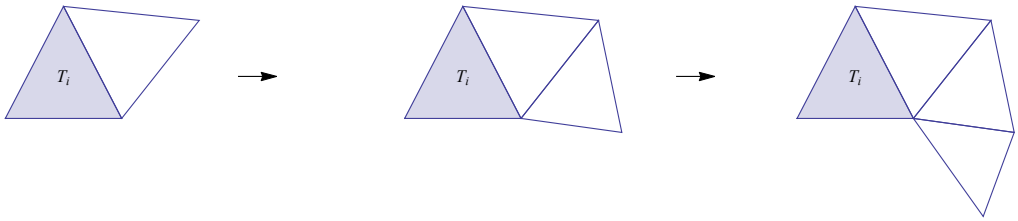

Figure 2.1: Possible expansion of the stencil using first selection method.

To add one more triangle, we proceed similarly to previous step. The difference is that also the neighbours of the added triangle T_l are candidates for the stencil. Overall, we have to test four possibilities and for instance triangle T_n is selected, i.e. $S = \{T_i, T_l, T_n\}$. It is possible that the triangle T_n does not have to be direct neighbour of the corresponding triangle T_i (see

middle graph of the Figure 2.1) using the described process. We proceed in the similar way until reaching m triangles in the stencil.

Another approach is to try to choose such triangles which are nearest to the given triangle T_i as much as possible. The Figure 2.2 shows the possibilities which we consider if we want three triangles included in the stencil. To add other triangles, we continue by the same way for each already chosen triangle in the stencil.

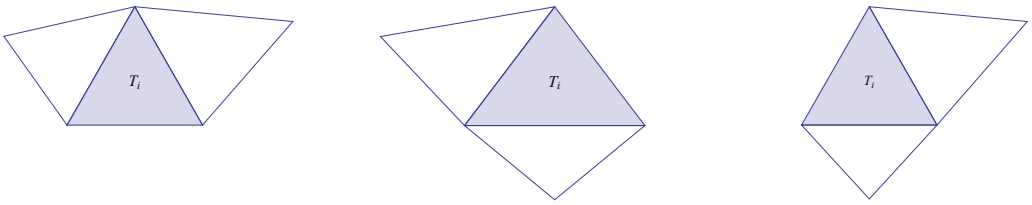

Figure 2.2: Possible configurations of the stencil using second selection method.

In what follows, we test both of the described approaches. The first way (resp. the second way) of the selection method of the triangles, we will mark W_1 (resp. W_2)

2.2 Numerical Results

Now we present some numerical experiments of application of the ENO method for the hyperbolic conservation law in two dimensional space. We use both the polynomials and RBFs for reconstruction. Our aim is to compare which method provides better – lets say more accurate – result.

Above, we introduced two possible selection methods (W_1 and W_2) how to add triangles into the stencil. We study which one of them can give us numerical solution with higher quality.

We apply ENO method in 2D for linear scalar equation, system of linear equations and nonlinear equation. We use continuous and also discontinuous initial condition.

2.2.1 Linear Equation

Let us commence by solution of the transport equation – scalar linear hyperbolic equation – with an initial condition

$$u_t + au_x + bu_y = 0, \quad \mathbf{x} \in [-1,1] \times [-1,1], t \in (0,T), \tag{2.31}$$

$$u(x, y, 0) = u_0(x, y), \quad \mathbf{x} \in [-1, 1] \times [-1, 1].$$

Similarly to 1D problems, outflow boundary conditions are applied. The advantage of the transport equation is that the analytical solution is known similarly to 1D problem. The analytical solution at time t in 1D is just shift of the initial condition with respect to the value of the constant c. In 2D, there we have given a directional vector (a, b). The constant a (resp. b) represents move in the direction of the axis x (resp. y). The analytical solution of the transport equation in 2D is the shift of the initial condition u_0 in the direction of the vector (a, b)

$$u(x, y, t) = u_0(x - at, y - bt). \tag{2.32}$$

Because the analytical solution is available, we are thus interested in the error of the numerical solution. Let us consider that $\hat{u}(\mathbf{x}_i, t_n)$ is the average of the analytical solution and $\bar{U}_i^n$ are the averages of the numerical solution of the transport equation at the centers $\mathbf{x}_i$ of the triangle T_i and in the time t_n. We can define then the error of the numerical solution in the following way

$$\sum_{i=1}^{N} |T_i| \left| \hat{u}(\mathbf{x}_i, t_n) - \bar{U}_i^n \right|, \tag{2.33}$$

where $|T_i|$ is the area of the triangle T_i and N is the number of the triangles.

We introduced ENO scheme in 2D on triangulation mesh in the beginning of the section 2.1. Transport equation (2.31) is defined in the square $[-1, 1] \times [-1, 1]$. The Figures 2.3 and 2.4 show various triangulations of this domain which we use in the following numerical experiments.

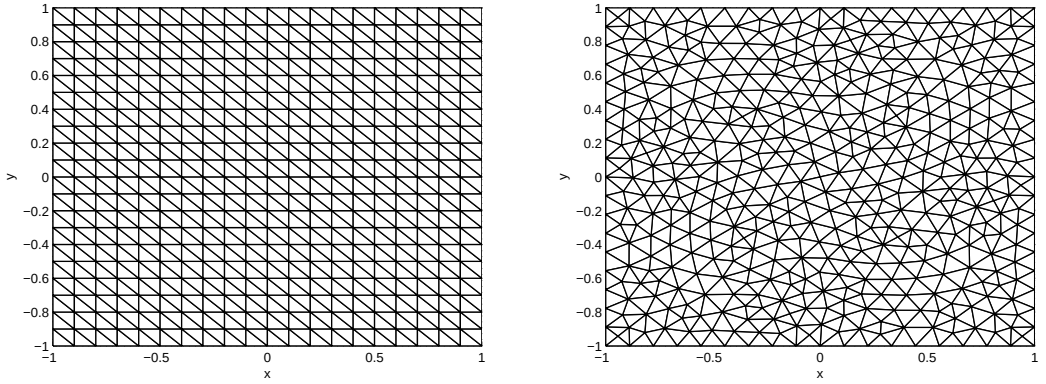

Figure 2.3: Right–angled (left) triangulation $\mathcal{T}_1$ with 800 triangles and general (right) triangulation $\mathcal{T}_2$ with 856 triangles.

We used software Matlab to generate the first right–angled triangulation with 800 triangles showed in the left graph of the Figure 2.3 s.t. all of the triangles have the same area. We mark this triangulation by $\mathcal{T}_1$.

The remaining triangulations in the graphs were generated in the software GMSH which makes triangular meshes of arbitrary domain in plane and in space too. This software generates general triangulations close to the equilateral triangles. Triangulation in the right graph of the Figure 2.3 is created by 856 triangles. Let us mark this triangulation by $\mathcal{T}_2$. Triangulations in the Figure 2.4 contain 2054, resp. 3770, triangles which we mark by $\mathcal{T}_3$, resp. $\mathcal{T}_4$.

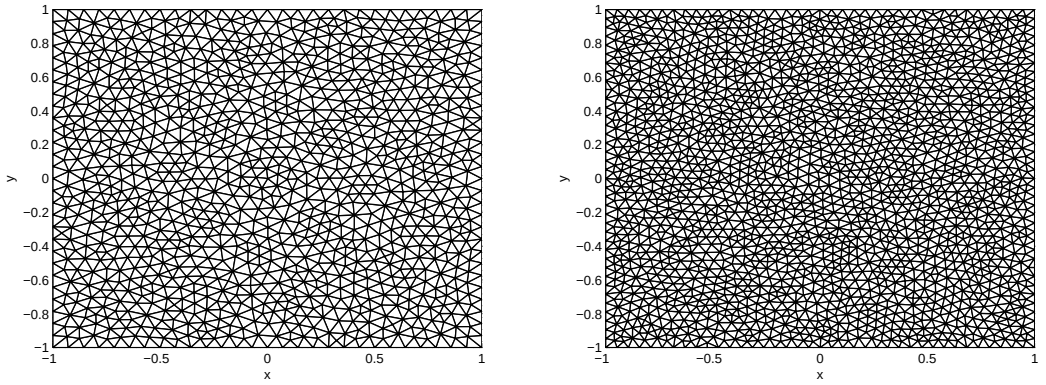

Figure 2.4: Triangulation $\mathcal{T}_3$ with 2054 (left) and triangulation $\mathcal{T}_4$ with 3770 (right) triangles.

In the following experiments, we apply ENO method to transport equation for various initial conditions. ENO method is tested using both polynomials and radial basis functions for reconstruction. Also, we apply mentioned selection methods of stencil W_1, W_2 and all of the triangulations $\mathcal{T}_1$ - $\mathcal{T}_4$ are used.

Experiment 1

Let us consider transport equation (2.31) for which we consider continuous initial condition as in the Figure 2.5.

First, ENO method is tested using polynomial reconstruction. We compare quality of the numerical solution depending on the degree of the polynomial for reconstruction. We consider time discretization $\Delta t = 0.0025$ and direction of the shift of the solution $(a, b) = (1, 1)$. Analytical solution after 300 time steps is showed in the Figure 2.6.

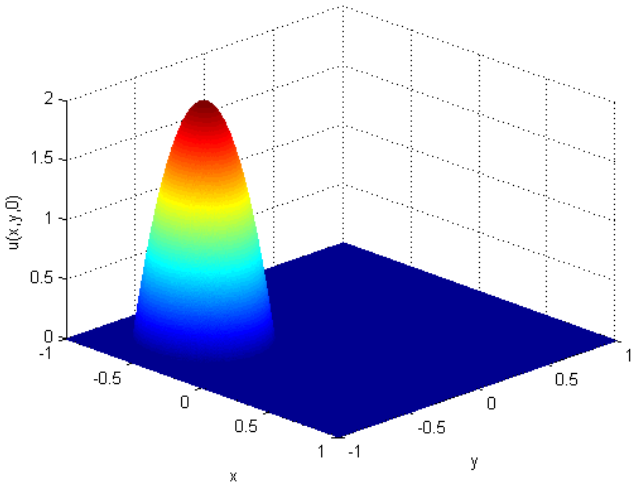

Figure 2.5: Continuous initial condition.

Figures 2.7 – 2.8 show numerical solution using constant reconstruction for every triangulations $\mathcal{T}_1$–$\mathcal{T}_4$ of the domain. Euler method is used for time discretization. Notice that using constant polynomial, the selection methods of triangles W_1 and W_2 are the same because only one triangle is needed for stencil.

We can compare the structure of the numerical solution in the graphs with analytical solution in the Figure 2.6. Evidently, numerical solution using constant polynomial is very inaccurate because the contours spread a lot to the neighbourhood. We can notice that quality of the numerical solution rises with increasing number of the triangles in the mesh.

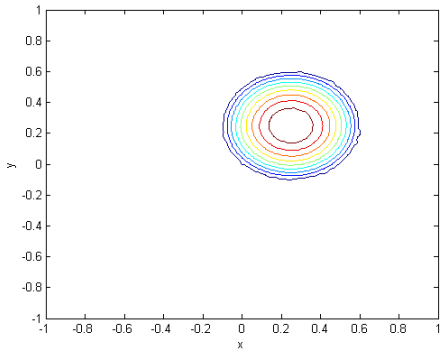

Figure 2.6: Analytical solution.

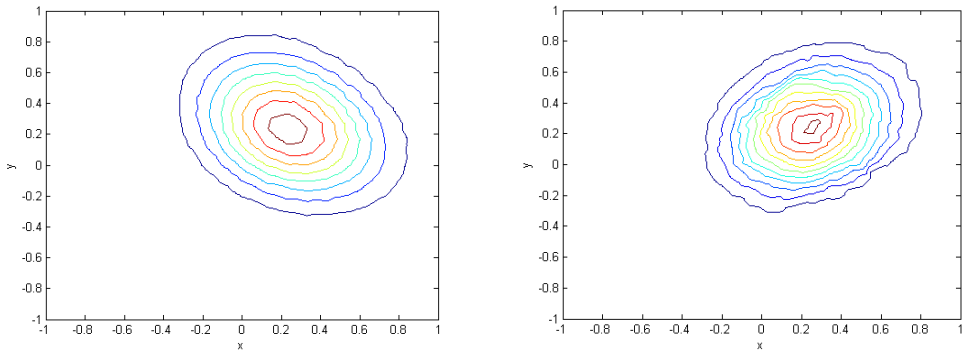

Figure 2.7: Numerical solution using constant recovery after 300 time steps on triangulation $\mathcal{T}_1$ (left) and $\mathcal{T}_2$ (right).

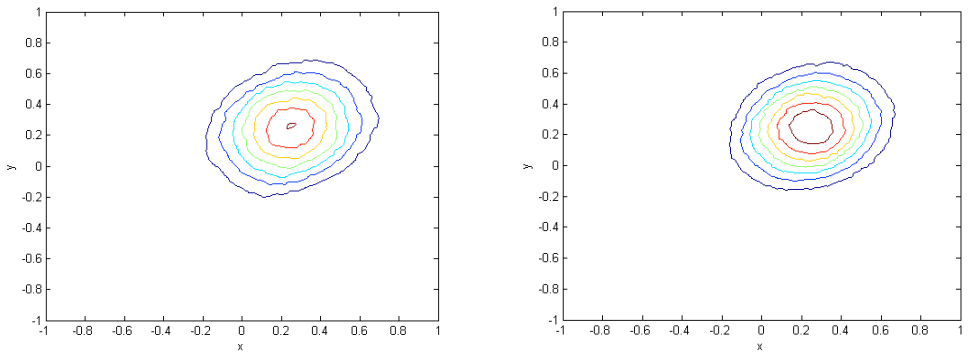

Figure 2.8: Numerical solution using constant recovery after 300 time steps on triangulation $\mathcal{T}_3$ (left) and $\mathcal{T}_4$ (right).

Numerical solution using ENO method with linear reconstruction after 300 time steps is showed in the Figures 2.9 and 2.10. Stencil has to include three triangles to determine linear polynomial. The selection methods of triangles W_1 and W_2 can thus be compared. The first method W_1, resp. second method W_2, is showed in the Figure 2.9, resp. Figure 2.10.

Overall, the quality of the numerical solution is much higher in case of the fine grid than in case of coarse grid. It does not spread so much and the shape of the contours is more rounded. Also, numerical solution using linear recovery improves previous case with constant approximation according to the figures.

But there is no big difference between the numerical solution using W_1 and W_2 in the figures. It only seems that the contours are closer to each other for W_2 than for W_1.

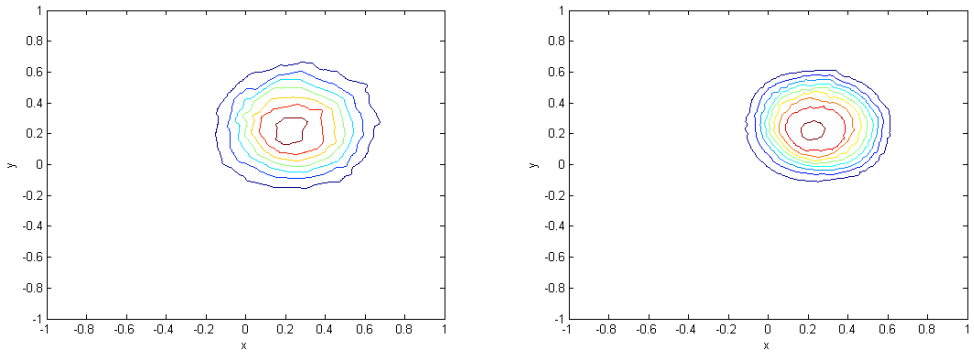

Figure 2.9: Numerical solution using linear recovery after 300 time steps on triangulation $\mathcal{T}_2$ (left) and $\mathcal{T}_4$ (right) for selection method W_1.

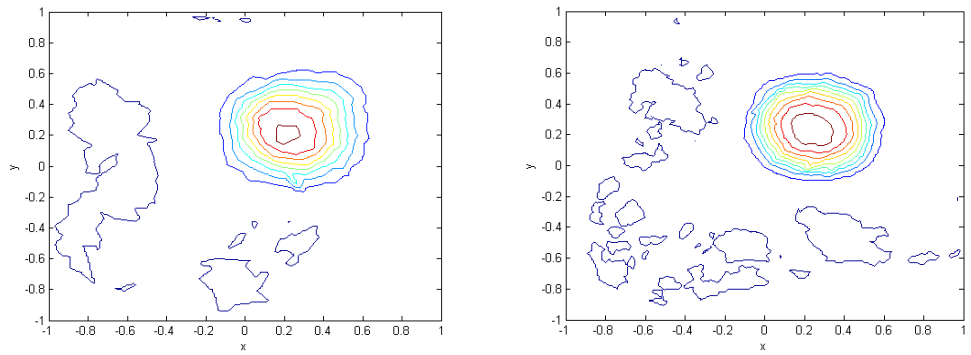

Figure 2.10: Numerical solution using linear recovery after 300 time steps on triangulation $\mathcal{T}_2$ (left) and $\mathcal{T}_4$ (right) for selection method W_2.

We show the error of the numerical solution after 300 time steps in the Table 2.1 according to the degree of the polynomial k and the number of the triangles using both selection methods W_1 and W_2. The table confirms described behaviour of the numerical solution showed in the previous graphs. The value of the error decreases as the number of the triangles and order of the method increase. It holds except method of the third order in case of selection method W_2. The value of the error is unlimited. This example show how is the way of the selection of the triangles important and that we can not choose arbitrary stencil.

We mentioned that the selection method W_2 is similar to selection which use Sonar in [6, 7, 13]. We can ask why such selection would be used if it products low quality of the numerical solution. In fact, Sonar developed this way to use it specially for linear recovery and reconstruction using radial

basis functions with linear polynomial. For higher degree of the polynomial, we recommmend to use different selection. For example Abgrall in [2] studies special process for high degree polynomial.

For the same reason mentioned above, the error for ENO method using constant polynomial is equal for both selection methods W_1 and W_2.

k	error			
	triangulation $\mathcal{T}_1$	triangulation $\mathcal{T}_2$	triangulation $\mathcal{T}_3$	triangulation $\mathcal{T}_4$
	selection method W_1			
0	0.3750	0.2988	0.2171	0.1726
1	0.2114	0.1664	0.0898	0.0616
2	0.1339	0.0975	0.0548	0.0374
	selection method W_2			
0	0.3750	0.2988	0.2171	0.1726
1	0.1644	0.1223	0.0633	0.0458
2	0.1068	—	—	—

Table 2.1: Error of the numerical solution after 300 time steps.

Similarly to 1D problem, we are interrested in maximum height of the numerical solution because we know that the numerical method smoothed down the data. The same behaviour we have in two dimensional space according to the Table 2.2. The maximum height changes as it can be expected according to the values of the error in the Table 2.1, i.e. the maximum height rises with increasing number of triangles and order of the method.

k	maximum height			
	triangulation $\mathcal{T}_1$	triangulation $\mathcal{T}_2$	triangulation $\mathcal{T}_3$	triangulation $\mathcal{T}_4$
	selection method W_1			
0	0.8548	1.1343	1.4133	1.5393
1	1.3031	1.5230	1.7751	1.8278
2	1.8205	1.8929	1.9584	1.9681
	selection method W_2			
0	0.8548	1.1343	1.4133	1.5393
1	1.4256	1.6602	1.8316	1.8825
2	1.8163	—	—	—

Table 2.2: Maximum height of the numerical solution after 300 time steps.

We can see that the quality of the numerical solution is very low on triangulation $\mathcal{T}_1$. The maximum height is under the value 1 in this case, while the maximum height of the analytical solution is 2. By contrast, maximum height of the numerical solution using ENO method of the third order on triangulations $\mathcal{T}_3$ and $\mathcal{T}_4$ is very satisfying in case of selection method W_1.

From the existing results, it follows that the possible best choice to get the highest quality of the numerical solution is to use higher order of the method and the most fine grid. But according to the values of efficiency of the algorithm in the Table 2.3, more triangles and higher order of the method imply higher time–consuming. In the table, it is entered time required for one time step of the ENO method using W_1. The efficiency for selection method W_2 is almost the same, we thus do not display extra table for it. We leave choice to the reader if it is more satisfactory to get approximate solution in short time or numerical solution with smaller error against higher efficiency.

k	Efficiency [s]			
	triangulation $\mathcal{T}_1$	triangulation $\mathcal{T}_2$	triangulation $\mathcal{T}_3$	triangulation $\mathcal{T}_4$
0	0.1	0.2	0.5	1.0
1	0.5	0.5	1.3	2.5
2	1.1	1.2	3.0	5.7

Table 2.3: Efficiency of the algorithm for one time step.

In what follows, ENO scheme using linear recovery in combination with radial basis functions are tested. We are interested if reconstruction using splines improves classical linear recovery. Method is applied to the transport equation (2.31) using continuous initial condition in the Figure 2.5. Direction of the shift is still $(a, b) = (1, 1)$ and time size $\Delta t = 0.0025$.

Both spline (2.22) and approximation (2.29) are defined using m radial basis functions. Gaussian approximation with parameter ϵ is used in our experiments and thin plate spline is applied in the end of this section too.

Figures 2.11 and 2.12 show dependence of the error of the numerical solution on choice of the parameter ϵ. Gaussian spline was used in the first figure and RBF approximation without linear polynomial was used in the second figure. Three radial basis functions, i.e. $m = 3$, are used in the left graphs of the figures and six RBFs, i.e. $m = 6$, are used in the right graphs.

According to the figures, quality of the numerical solution using Gaussian reconstruction is almost the same for all values of the parameter ϵ. Only the least value of the parameter $\epsilon = 0.5$ causes unlimited error. By contrast, error of the numerical solution using only RBF approximation is different

for various values of the parameter. An advantage thus implies for Gaussian recovery compared to reconstruction using only RBFs because the quality of the numerical solution is not so dependent on the parameter ϵ.

We choose parameter $\epsilon \approx 3$ for approximation using three and four RBFs ($m = 3$ and $m = 4$) and $\epsilon \approx 5$ using six RBFs ($m = 6$) in the experiments. We use the same values for Gaussian recovery.

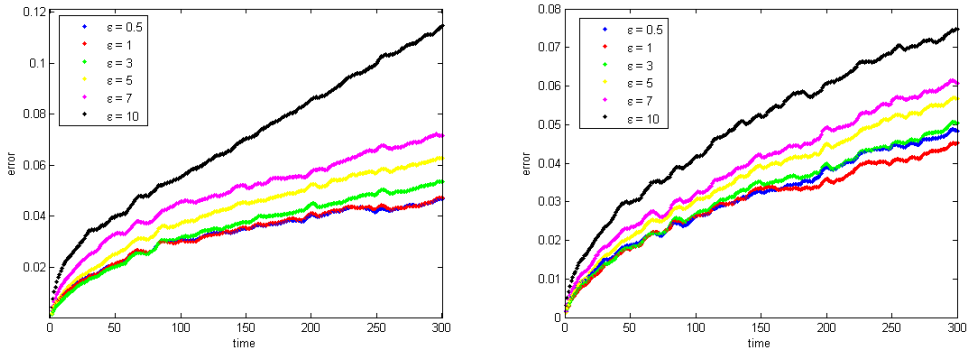

Figure 2.11: Error of the numerical solution in time compared for various values of the parameter ϵ. Reconstruction uses only RBFs, $m = 3$ (left), $m = 6$ (right).

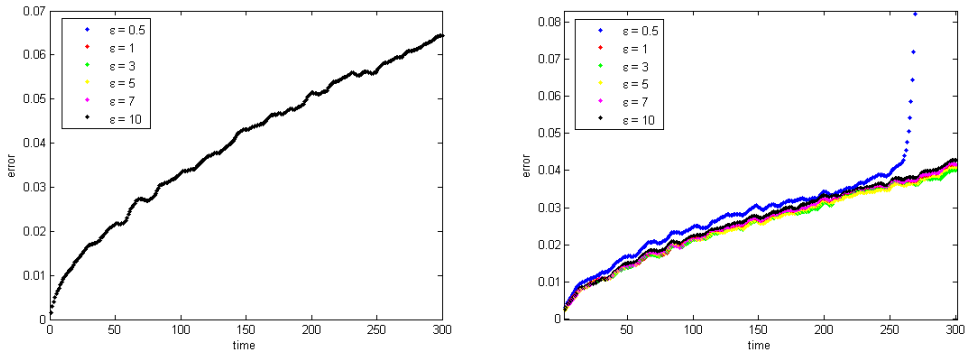

Figure 2.12: Error of the numerical solution in time compared for various values of the parameter ϵ. Reconstruction uses Gaussian splines, $m = 3$ (left) and $m = 6$ (right).

The Figures 2.13 – 2.16 show numerical solution after 300 time steps for triangulation $\mathcal{T}_2$. We display one graph using $m = 3$ and one for $m = 6$ in each figure. All of the cases of the reconstruction are showed, i.e. reconstruction using Gaussian spline or using only RBFs. We decide about the stencil using both coefficients and total variation and W_2 - as selection method of triangles - is used.

According to the figures below, the highest quality is reached in case of the reconstructions using only RBFs via coefficients (i.e. we choose stencil by the sum of the coefficients) for $m = 3$, $m = 6$ and next using both of the Gaussian recovery for $m = 6$. The contours are close to each other and the shapes are the most symmetric and most similar to the analytical solution in Figure 2.6. Also, let us notice that the quality of the mentioned cases implies superiority of RBF approximation to linear polynomial reconstruction on triangulation $\mathcal{T}_2$ which is in the Figure 2.9. But for example the structure of the numerical solution showed in the left graphs of the Figures 2.15 – 2.16 is similar to linear recovery in the Figure 2.9.

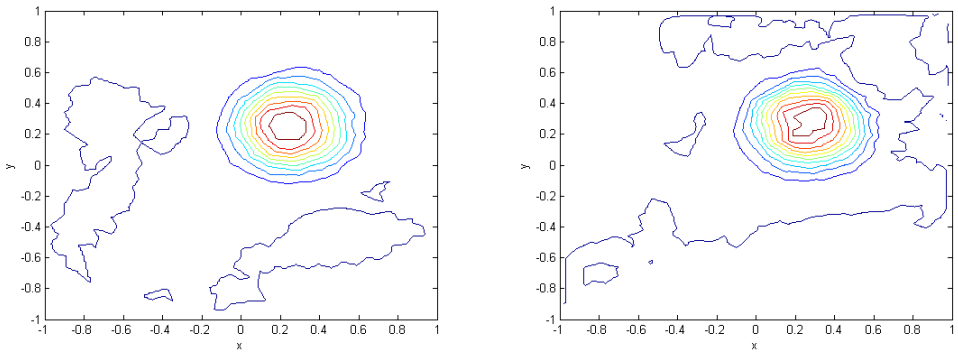

Figure 2.13: Numerical solution after 300 time steps using only RBFs for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_2$. The stencil is chosen via coefficients.

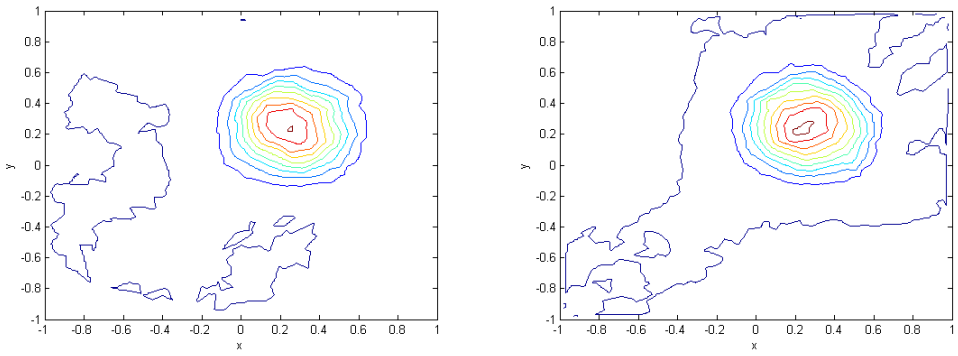

Figure 2.14: Numerical solution after 300 time steps using only RBFs for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_2$. The stencil is chosen via Total variation.

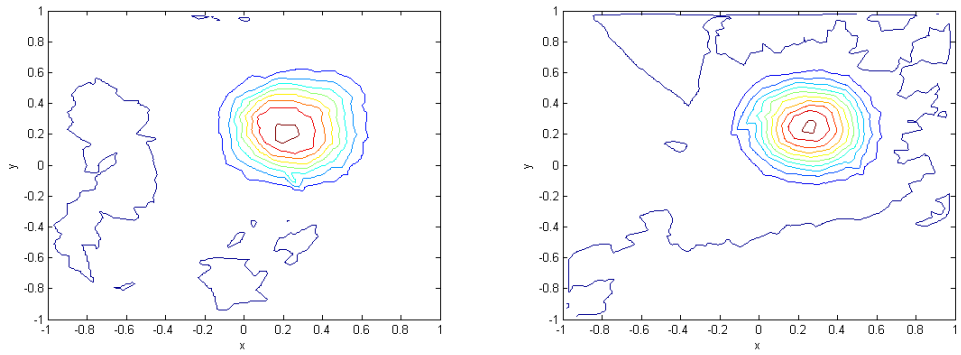

Figure 2.15: Numerical solution after 300 time steps using Gaussian splie for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_2$. The stencil is chosen via coefficients.

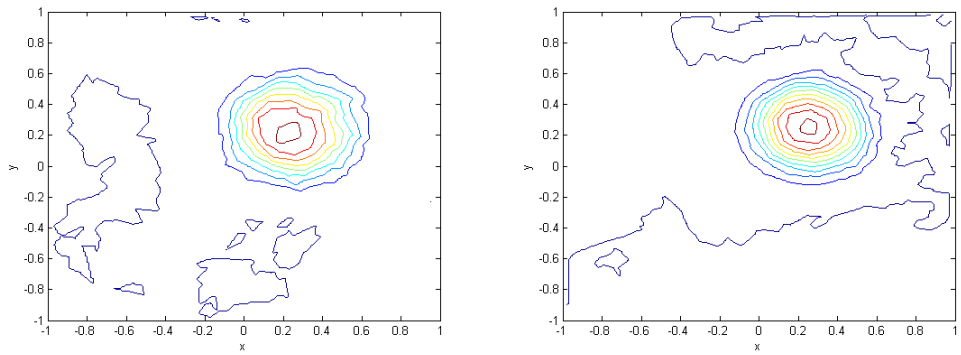

Figure 2.16: Numerical solution after 300 time steps using Gaussian spline for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_2$. The stencil is chosen via TV.

We try to confirm our observations in the Tables 2.4, 2.6 and 2.8 (resp. 2.5, 2.7 and 2.9) which show the error (resp. maximum height) of the numerical solution depending on the type of the reconstruction for $m = 3$, $m = 4$ and $m = 6$. In the tables, we mark RBF, resp. RBF+TV, as reconstruction of the cell averages using only radial basis functions via coefficients, resp. total variation. Next GS, resp. GS+TV, represents reconstruction using Gaussian spline via coefficients, resp. total variation.

First of all, the error decreases with increasing number of the triangles. Let us consider now only approximation using Gaussian spline. We get values of the error for $m = 3$ similar to linear polynomial which is entered in the Table 2.1. The error decreases as we add the radial basis functions to Gaussian spline and the maximum height converges to exact maximum height which is equal to 2. It thus seems that the statement of Sonar in [7] holds, i.e. at least

four radial basis functions have to be used to improve linear polynomial.

	error							
	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
$\mathcal{T}_1$	0.1994	0.2132	0.2066	0.2082	0.1162	0.1589	0.8273	0.8301
$\mathcal{T}_2$	0.1588	0.1695	0.1650	0.1643	0.0895	0.1179	0.1223	0.1187
$\mathcal{T}_3$	0.0801	0.0900	0.0887	0.0881	0.0536	0.0643	0.0633	0.0643
$\mathcal{T}_4$	0.0559	0.0601	0.0608	0.0603	0.0449	0.0427	0.0458	0.0431

Table 2.4: Error of the numerical solution after 300 time steps for $m = 3$.

	maximum height							
	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
$\mathcal{T}_1$	1.6281	1.4823	1.3266	1.3269	1.7736	1.7015	1.9218	1.9403
$\mathcal{T}_2$	1.7784	1.7001	1.5292	1.5287	1.9650	1.8141	1.6602	1.6918
$\mathcal{T}_3$	2.0172	1.9205	1.7740	1.7752	1.9941	1.9721	1.8316	1.8322
$\mathcal{T}_4$	1.9991	1.9286	1.8262	1.8276	2.0410	1.9897	1.8825	1.8831

Table 2.5: Maximum height of the numerical solution after 300 time steps for $m = 3$.

	error							
	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
$\mathcal{T}_1$	0.1732	0.1815	0.1797	0.1647	0.0737	0.1717	0.0944	0.1300
$\mathcal{T}_2$	0.1148	0.1188	0.1108	0.1114	0.0660	0.1202	0.0703	0.0935
$\mathcal{T}_3$	0.0636	0.0619	0.0584	0.0585	0.0473	0.0579	0.0486	0.0545
$\mathcal{T}_4$	0.0464	0.0456	0.0432	0.0439	0.0466	0.0399	0.0495	0.0445

Table 2.6: Error of the numerical solution after 300 time steps for $m = 4$.

Error of the numerical solution using Gaussian recovery with total variation, selection method W_1 and $m = 6$ is half of the error than for linear polynomial on triangulations $\mathcal{T}_2$, $\mathcal{T}_3$, $\mathcal{T}_4$. This error is even smaller than error for ENO method of the third order using polynomial. On contrary, the previous observation holds for selection method W_2 only on triangulation $\mathcal{T}_2$. In other cases, the improvement is not so important for selection method W_2

as for W_1. According to the tables, quality of the numerical solution using Gaussian spline via both coefficients and total variation is similar.

Now, we focus on approximation using only the radial basis function without linear polynomial. According to the tables, the reconstruction using only RBFs does not product satisfactory results because the changes of error and maximum height does not hold any rule as we increase number of RBFs m. Error decreases in some cases if we change m from 3 to 4, but error then rises changing m from 4 to 6. Similar behaviour holds for maximum height too.

	maximum height							
	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
$\mathcal{T}_1$	1.7704	1.6626	1.5201	1.5613	1.9901	1.4608	1.9922	1.7829
$\mathcal{T}_2$	1.9946	1.9554	1.6984	1.7004	2.0452	1.8212	2.1654	2.0553
$\mathcal{T}_3$	2.0599	2.0324	1.8715	1.8697	2.0421	1.9388	2.0730	2.0447
$\mathcal{T}_4$	2.0443	2.0335	1.8997	1.8998	2.0398	1.9773	2.0379	2.0396

Table 2.7: Maximum height of the numerical solution after 300 time steps for $m = 4$.

	error							
	seslection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
$\mathcal{T}_1$	0.1417	0.2020	0.1496	0.1392	0.1049	0.1321	0.0990	0.0993
$\mathcal{T}_2$	0.0993	0.1510	0.0890	0.0735	0.0788	0.1095	0.0801	0.0684
$\mathcal{T}_3$	0.0567	0.0936	0.0567	0.0471	0.5503	0.0680	0.0446	0.0408
$\mathcal{T}_4$	0.0426	0.0730	0.0371	0.0355	0.0449	0.0524	0.0355	0.0364

Table 2.8: Error of the numerical solution after 300 time steps for $m = 6$.

	maximum height							
	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
$\mathcal{T}_1$	2.1260	1.6137	1.7731	1.7374	2.1548	1.9070	1.9112	1.9734
$\mathcal{T}_2$	2.1248	1.8883	1.8835	1.9555	2.1399	1.8828	2.0484	2.0572
$\mathcal{T}_3$	1.9833	1.8783	1.9833	1.9721	101.7653	1.8811	2.0316	2.0471
$\mathcal{T}_4$	1.9853	1.8628	1.9968	1.9702	2.0319	1.9197	2.0271	2.0395

Table 2.9: Maximum height of the numerical solution after 300 time steps for $m = 6$.

Experiment 2

Most of the real problems contain discontinuities. The results in previous chapter for one dimensional problem showed that the quality of the numerical solution depends on initial condition. We are thus interested in quality of the numerical solution of transport equation (2.31) using discontinuous initial condition in two dimensional space.

The initial condition showed in the Figure 2.17 is used. We choose again $(a, b) = (1, 1)$, $\Delta t = 0.0025$ and parameter $\epsilon = 3$ for $m = 3$, $m = 4$ and $\epsilon = 5$ for $m = 6$ in case of the reconstruction using radial basis functions. We study quality of the numerical solution after 300 time steps.

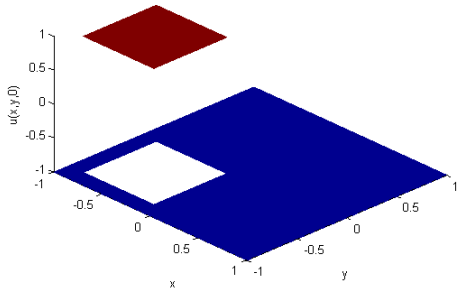

Figure 2.17: Discontinuous initial condition.

Let us begin with the experiments using polynomial reconstruction. We again show the improvement of the numerical solution if we increase the order of the method in the Figures 2.18 and 2.19 using selection method of triangles W_1. In the first figure, we compare first order method for different number of the triangles. Certainly, the method on more fine grid provides higher quality.

Table 2.10 confirms our observation from the figures where polynomial reconstruction is used. Error and maximum height of the numerical solution of ENO method using RBFs are entered in the Tables 2.11 (for $m = 3$), 2.12 (for $m = 4$) and 2.13 (for $m = 6$).

First, let us focus on error using selection method W_2 in the tables. Error is unlimited using quadratic polynomial again similarly to previous experiment for continuous initial condition. Also in case of reconstruction using radial basis functions, the maximum height increases too much over the exact value 1 using $m = 4$ and $m = 6$ for almost all of the types of the reconstruction. It happens using RBFs and Gaussian spline via coefficients for $m = 4$, but it is in all cases of the reconstruction for $m = 6$. From the given results, it

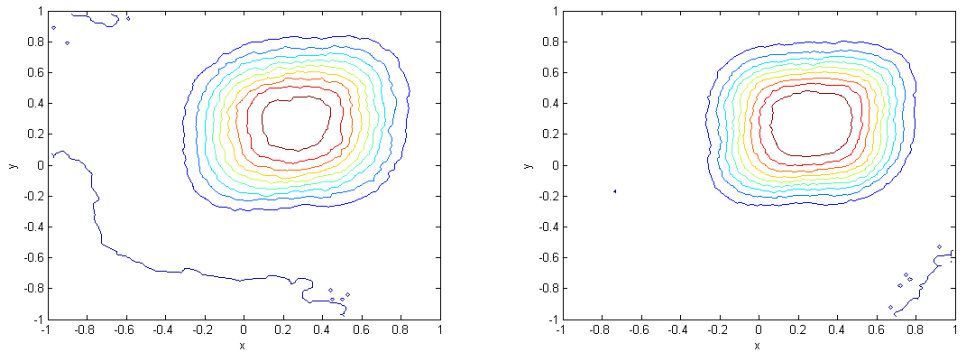

Figure 2.18: Numerical solution of first order of the method after 300 time steps on triangulation $\mathcal{T}_3$ (left) and $\mathcal{T}_4$ (right).

follows that numerical solution using W_2 does not provide satisfactory results for any type of the reconstruction.

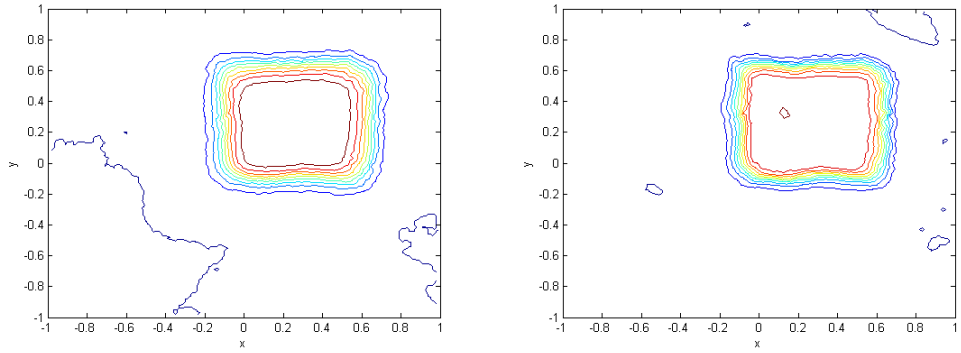

Figure 2.19: Numerical solution of second (left) and third (right) order ENO method after 300 time steps on triangulation $\mathcal{T}_4$.

The error using selection method W_1 changes similarly to continuous initial condition. Approximation using only RBFs via total variation provides the worst results. But quality of the numerical solution using approximation with only RBFs via coefficients and Gaussian spline is comparable for $m = 3$ and $m = 4$. Also, both Gaussian spline via coefficietns and total variation produces similar results. For $m = 3$, quality of the numerical solution degenerates again to linear recovery. As the number of radial basis functions rises, the error degreases. Moreover, the error for $m = 6$ is smaller than error for ENO method using quadratic polynomial and the maximum height keeps exact value 1.

	selection method W_1		selection method W_2	
	triangulation $\mathcal{T}_3$	triangulation $\mathcal{T}_4$	triangulation $\mathcal{T}_3$	triangulation $\mathcal{T}_4$
k	error			
0	0.5216	0.4490	0.5216	0.4490
1	0.2819	0.2262	0.2075	0.1625
2	0.1912	0.1512	—	—
k	maximum height			
0	0.9590	0.9858	0.9590	0.9858
1	0.9982	0.9998	1.0031	1.0029
2	1.0000	1.0003	—	—

Table 2.10: Error and maximum height of the numerical solution after 300 time steps for polynomial reconstruction.

	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
	error							
$\mathcal{T}_3$	0.2787	0.3171	0.2814	0.2813	0.2869	0.2432	0.2075	0.2131
$\mathcal{T}_4$	0.2267	0.2541	0.2256	0.2259	0.2415	0.2204	0.1625	0.2204
	maximum height							
$\mathcal{T}_3$	1.0114	1.0099	0.9983	0.9983	1.0426	1.0255	1.0031	1.0018
$\mathcal{T}_4$	1.0071	1.0041	0.9998	0.9998	1.0227	1.0198	1.0029	1.0198

Table 2.11: Error and maximum height of the numerical solution after 300 time steps, using RBFs and $m = 3$.

	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
	error							
$\mathcal{T}_3$	0.2148	0.2167	0.2111	0.2100	0.2628	0.2353	0.1938	0.1936
$\mathcal{T}_4$	0.1667	0.1781	0.1625	0.1621	0.2334	0.1764	0.2030	0.1510
	maximum height							
$\mathcal{T}_3$	1.0182	1.0196	0.9999	0.9999	1.4336	1.0021	1.4275	1.0251
$\mathcal{T}_4$	1.0101	1.0128	1.0000	1.0000	1.7665	1.0011	1.8270	1

Table 2.12: Error and maximum height of the numerical solution after 300 time steps, using RBFs and $m = 4$.

	selection method W_1				selection method W_2			
	RBF	RBF+TV	GS	GS+TV	RBF	RBF+TV	GS	GS+TV
	error							
$\mathcal{T}_3$	0.2366	0.2839	0.1802	0.1705	—	0.3483	0.1499	0.1529
$\mathcal{T}_4$	0.1761	0.2549	0.1354	0.1265	—	0.1921	0.1121	0.1150
	maximum height							
$\mathcal{T}_3$	1.0346	1.0083	1.0000	1.0000	—	2.7819	1.0927	1.1180
$\mathcal{T}_4$	1.0273	1.0049	1	1.0000	—	1.0093	1.2462	1.2501

Table 2.13: Error and maximum height of the numerical solution after 300 time steps, using RBFs and $m = 6$.

We want to develope method which is stable for various settings and initial conditions. It follows, from the results above which tested several types of the reconstruction, that ENO method using Gaussian spline via coefficients or total variation provides the most appropriate numerical solution so that at least four radial basis functions is used and selection method of triangles W_1 is applied.

The similar quality of numerical solution using Gaussian spline via coefficients and total variation can implies that there exists connection - as for polynomials - between coefficients and total variation which we use as criteria of oscillatory behaviour. Because total variation is more used in various problems, we will prefer approximation using Gaussian splines via total variation.

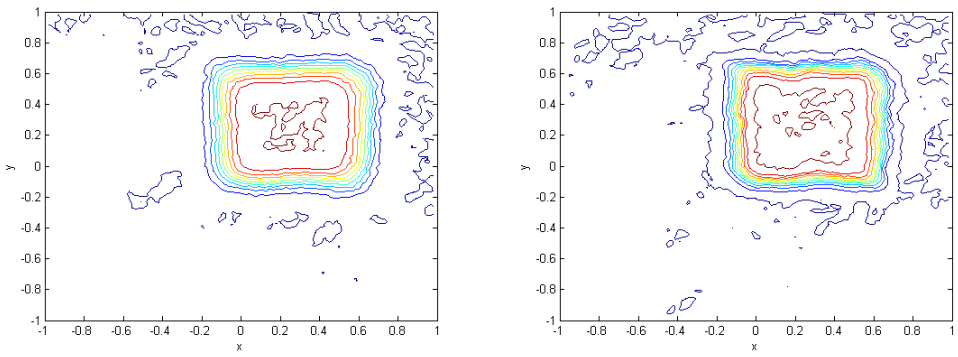

Figure 2.20: Numerical solution after 300 time steps. Reconstruction uses only RBFs for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_4$. Stencil is chosen via coefficients.

Numerical solution using selection method W_1 on the triangulation $\mathcal{T}_4$ is showed in the following figures. Reconstruction using only RBFs is applied in the Figure 2.20, resp. in the Figure 2.21, s.t. the stencil is chosen via

coefficients, resp. via total variation. Gaussian spline is applied in the Figures 2.22 and 2.23. In each figure, $m = 3$ and $m = 6$ is used.

According to the graphs, the structure of the results for $m = 3$ is similar to results using linear recovery (see Figure 2.19). On contrary, all graphs for $m = 6$ show superiority of reconstruction using RBFs to linear recovery (except case in the Figure 2.21) because the contours are closer and keep more the square shape. The quality of the numerical solution of the just mentioned cases is similar to quality of numerical solution using ENO method of the third order (see Figure 2.19) according to the figures.

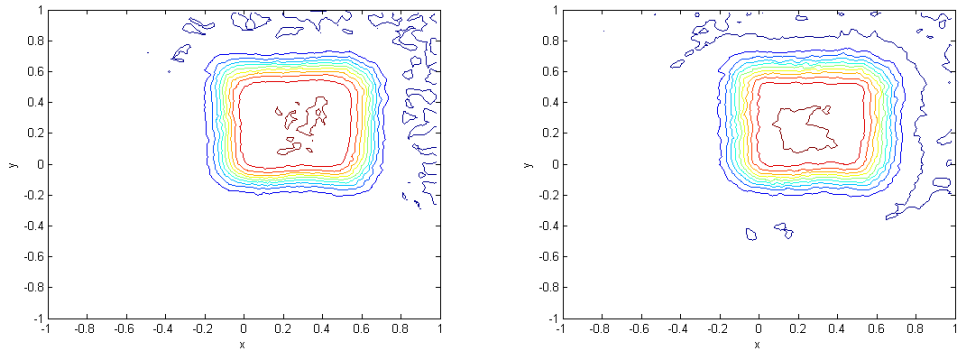

Figure 2.21: Numerical solution after 300 time steps. Reconstruction uses only RBFs for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_4$. Stencil is chosen via total variation.

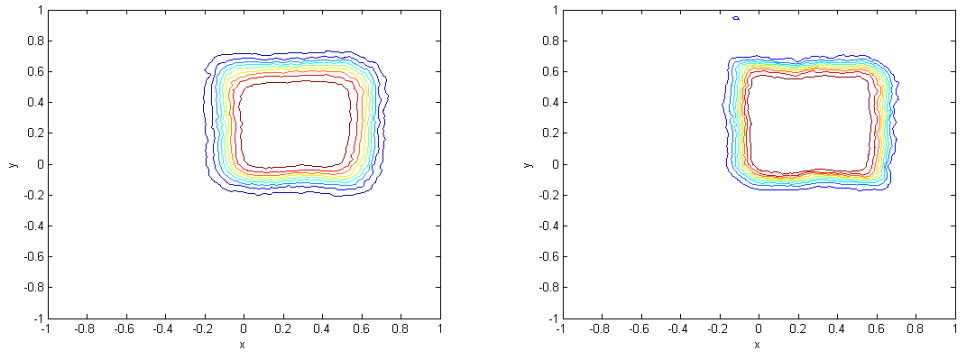

Figure 2.22: Numerical solution after 300 time steps. Reconstruction uses Gaussian spline for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_4$. Stencil is chosen via coefficients.

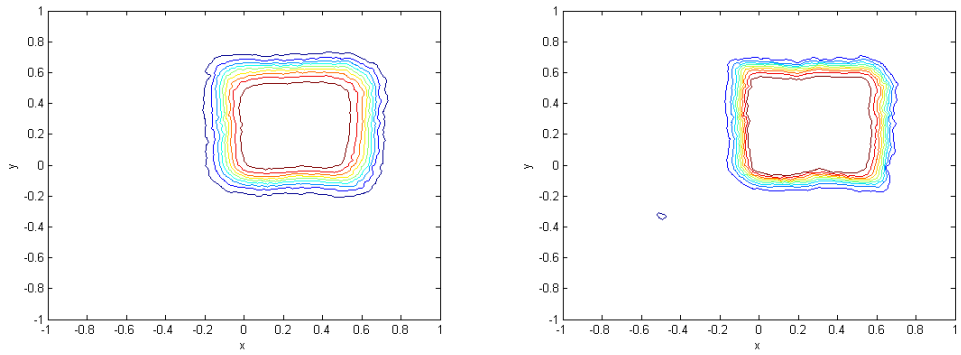

Figure 2.23: Numerical solution after 300 time steps. Reconstruction uses Gaussian spline for $m = 3$ (left) and $m = 6$ (right) on $\mathcal{T}_4$. Stencil is chosen via total variation.

The previous experiments indicate that ENO method which applies reconstruction of cell averages using at least four radial basis functions improves linear recovery. But to reach this correction, higher time efficiency of the algorithm is needed. Tables 2.14 and 2.15 for $m = 3$ and $m = 6$ show efficiency of one time step. Time-consuming is comparable to linear recovery problem using three RBFs in reconstruction. By contrary, algorithm for reconstruction using six RBFs takes about two times more.

	Efficiency [s]			
	RBF	RBF+TV	GS	GS+TV
T_1	0.38	0.37	0.48	0.49
T_2	0.45	0.46	0.57	0.56
T_3	1.11	1.15	1.38	1.38
T_4	2.11	2.38	2.66	2.63

Table 2.14: Efficiency of the algorithm for one time step using $m = 3$.

	Efficiency [s]			
	RBF	RBF+TV	GS	GS+TV
T_1	0.68	0.72	1.08	1.09
T_2	0.85	0.90	1.27	1.29
T_3	2.08	2.19	3.18	3.23
T_4	3.91	4.14	5.96	6.1

Table 2.15: Efficiency of the algorithm for one time step using $m = 6$.

Experiment 3

ENO method has been tested in 2D using both continuous and discontinuous initial conditions. Although numerical experiments showed superiority of reconstruction using radial basis functions to linear recovery, diffusion effect is still problem of ENO method. We thus use special initial condition according to the Figure 2.24 where two discontinuities are located close to each other. We are interested if ENO method causes connection of the two shocks in time.

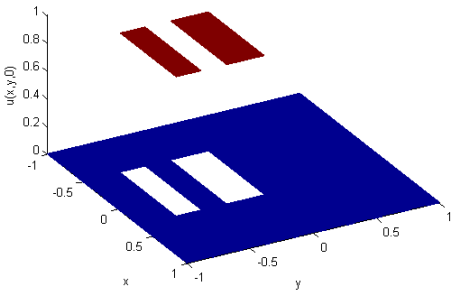

Figure 2.24: Initial condition.

The results of numerical experiments with the initial condition are presented in the Figures 2.25 – 2.29 on triangulation $\mathcal{T}_4$. We apply $(a, b) = (1, 1)$ and $\Delta t = 0.0025$. Numerical solution after 200 time steps, s.t. constant reconstruction is applied, is showed in the Figure 2.25.

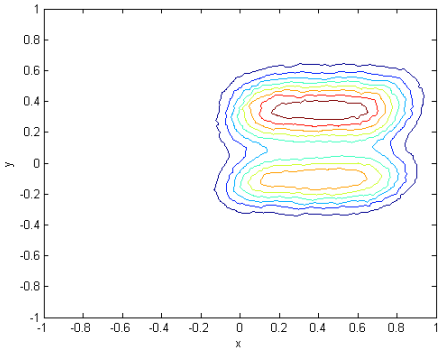

Figure 2.25: Numerical solution after 200 time steps. Reconstruction applies constant polynomial on triangulation $\mathcal{T}_4$.

Certainly, the quality of the numerical solution is in this case very low but

the smoothing causes that the two discontinuities bumps to blur together. We thus expect that we avoid this by using method of higher order which we demonstrate in the rest Figures 2.26 – 2.29.

Numerical solutions using linear recovery and Gaussian spline - via total variation and for $m = 6$ - are showed on triangulation T_4 (for which we expect the highest quality). Moreover, the numerical solution is demostrated in more phases, specifically after 50, 150, 200 and 300 time steps. We thus can compare the progress of the numerical solution and study the progress of numerical diffusion.

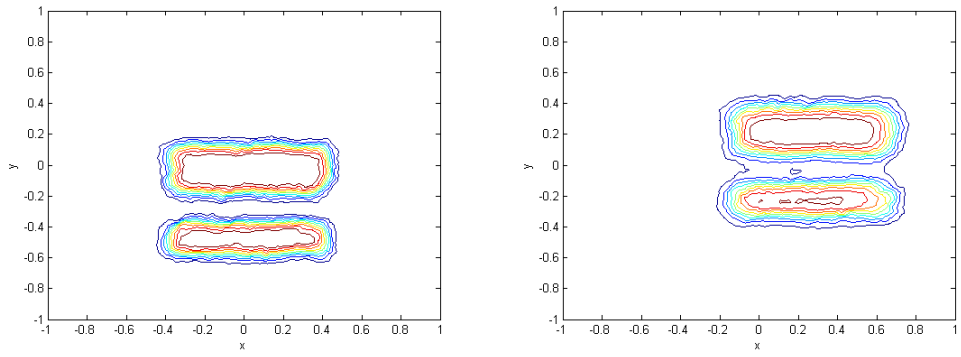

Figure 2.26: Numerical solution after 50 (left) and 150 (right) time steps. Reconstruction applies linear polynomial on triangulation $\mathcal{T}_4$.

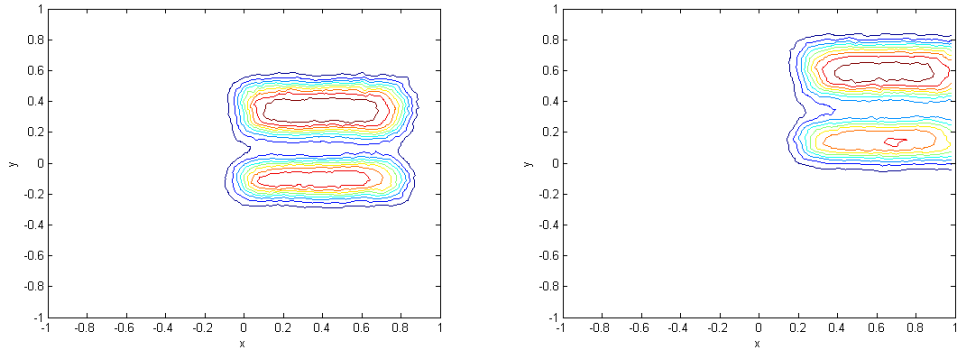

Figure 2.27: Numerical solution after 200 (left) and 300 (right) time steps. Reconstruction applies linear polynomial on triangulation $\mathcal{T}_4$.

We observe that the numerical solution starts to join already around 150 steps using polynomial recovery and it is getting worse and worse over the time. By contrast, the reconstruction using Gaussian spline provides higher

quality, because the contours do not seem to begin to join even after 300 time steps according to the figures. This experiment really shows the superiority of RBFs recovery to linear polynomial. Although we remind higher efficiency of the algorithm.

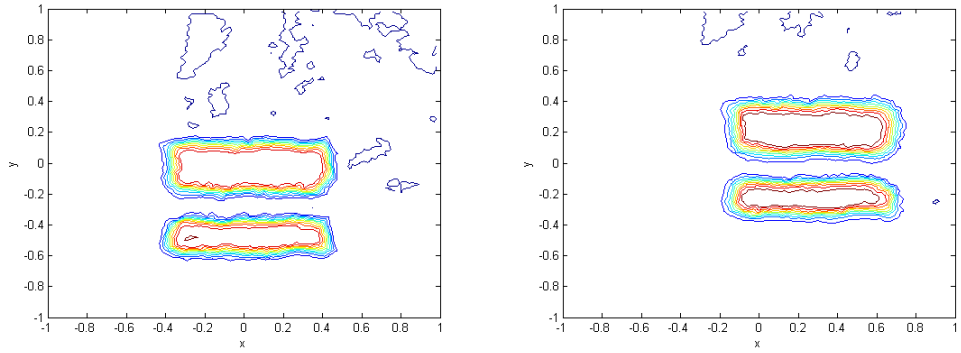

Figure 2.28: Numerical solution after 50 (left) and 150 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

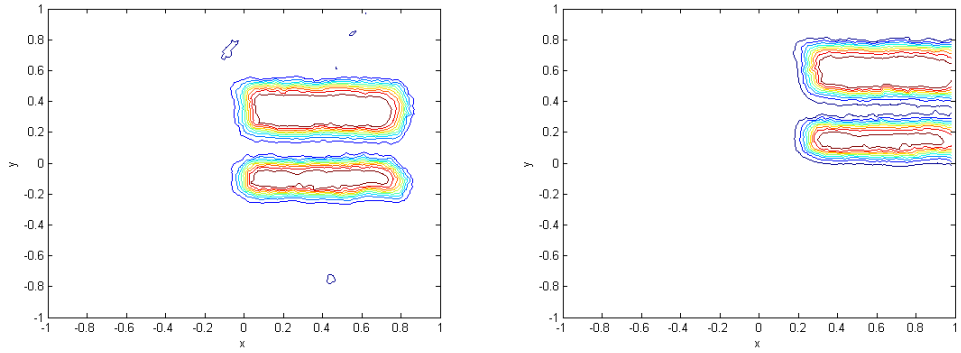

Figure 2.29: Numerical solution after 200 (left) and 300 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

Finally, the error of the numerical solution after 200 time steps is showed in the Tables 2.16 – 2.18 on the triangulation T_4. In the previous examples, there were applied the direction of the shift of the solution $(a, b) = (1, 1)$. Now we add another choice $(a, b) = (1, 0)$, i.e. the initial condition moves to the right as we can see in the Figure 2.30.

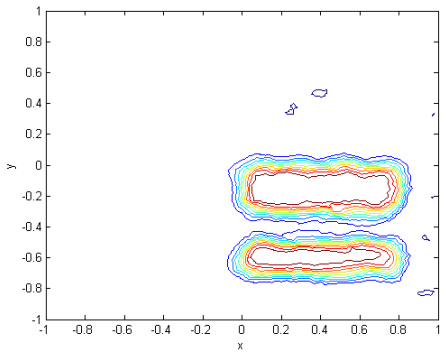

Figure 2.30: Numerical solution after 200 time steps using $(a, b) = (1, 0)$. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

The tables contain results for cases $m = 4$ and $m = 6$ using RBFs, because the result in the case $m = 3$ is similar to linear polynomial. According to the tables, we can see that the error changes by the same way for more complicated initial condition as in the previous experiments. Reconstruction via total variation using only radial basis function without linear polynomial provides the worst quality of the numerical solution again. The error even rises with increasing m in this case. It seems that total variation is not appropriate criteria to measure smoothness of approximation using only radial basis functions.

k	error	
	$(a, b) = (1, 1)$	$(a, b) = (1, 0)$
0	0.2472	0.1844
1	0.1396	0.1258

Table 2.16: Error of the numerical solution using polynomial recovery after 200 time steps for T_4.

	error			
	RBF	RBF+TV	GS	GS+TV
$(a, b) = (1, 1)$	0.0978	0.0998	0.1021	0.1020
$(a, b) = (1, 0)$	0.0989	0.1016	0.1022	0.1021

Table 2.17: Error of the numerical solution using Gaussian spline after 200 time steps for T_4, $m = 4$.

	error			
	RBF	RBF+TV	GS	GS+TV
$(a, b) = (1, 1)$	0.0793	0.1404	0.0832	0.0782
$(a, b) = (1, 0)$	0.0925	0.1422	0.0924	0.0925

Table 2.18: Error of the numerical solution using Gaussian spline after 200 time steps for T_4, $m = 6$.

Gaussian spline has been applied for ENO method so far. Let us study now how the quality of the numerical solution changes if thin plate splines are used, i.e. instead of Gaussian as generator of radial basis function, we use logarithm function defined by (2.21).

The error of the numerical solution of the transport equation in 2D for two discontinuities in the initial condition is showed in the Table 2.19 for $m = 4$ and in Table 2.20 for $m = 6$. The values of the error are similar to Gaussian spline, except that the error decreases in all cases of reconstruction with increasing number of RBFs m.

	error			
	RBF	RBF+TV	TPS	TPS+TV
$(a, b) = (1, 1)$	0.2227	0.1196	0.1025	0.1030
$(a, b) = (1, 0)$	0.1535	0.1624	0.1055	0.1057

Table 2.19: Error of the numerical solution using thin plate spline after 200 time steps for T_4, $m = 4$.

	error			
	RBF	RBF+TV	TPS	TPS+TV
$(a, b) = (1, 1)$	0.1064	0.1196	0.0931	0.0988
$(a, b) = (1, 0)$	0.0939	0.0907	0.0943	0.1013

Table 2.20: Error of the numerical solution using thin plate spline after 200 time steps for T_4, $m = 6$.

The aim of this chapter was to develope method which improves ENO method using linear recovery. All of the experiments above indicate that ENO method using linear reconstruction in combination with radial basis functions provides the correction.

We can ask if it is possible to improve also another degree of the polynomial. Although reconstruction using constant polynomial does not implies high quality of the numerical solution, but we tried to apply constant reconstruction with Gaussian spline. The experiments results in highly oscillatory numerical solution.

Parabolic polynomial is next candidate to correction. If the analogy to linear improvement holds, seven triangles would be minimum number of triangles for stencil to get correction of the parabolic reconstruction. We rather refer to efficiency of the algorithm at this time because already using at least four radial basis functions to improve linear recovery inplies too much time-consuming.

2.2.2 Nonlinear Equation

Previous section concentrated on application of ENO method on transport equation. In case of linear equation, there can be determined analytical solution and thus exactly computed error of numerical solution. In what follows, we will be interested in the application of ENO methods for nonlinear hyperbolic equations for which we do not generally know the analytical solution.

Numerical experiments for the linear equation lead to the conclusion that a numerical solution of ENO method using only radial basis functional reconstruction is not satisfactory which was caused mainly for discontinuous initial condition. Nonlinear equations describe various problems so that the initial condition can be continuous but evolution of the solution products discontinuities in the time. We thus do not recommend to apply reconstruction using only radial basis function to nonlinear equations. We study quality of the numerical solution using ENO method using Gaussian spline or thin plate splines.

Common example of a nonlinear hyperbolic equations is Burger's equation

$$u_t + \left(\frac{1}{2}u^2\right)_x + \left(\frac{1}{2}u^2\right)_y = 0, \quad \mathbf{x} \in [-1,1] \times [-1,1], t \in (0,T),$$

with the identical continuous initial condition from the Figure 2.5. The Figure 2.31 show numerical solution after 300 time steps using linear recovery. It present comparison of a quality of the results on triangulation $\mathcal{T}_2$ and $\mathcal{T}_4$ is compared.

Numerical solution in the Figure 2.32 was computed using linear recovery on triangulation $\mathcal{T}_4$ and selection method of the triangles is used W_2. This selection method is obviously not appropriate again because numerical solution tends to oscillate.

Numerical solution after 50, 150, 200 and 300 time steps is showed in the Figures 2.33 and 2.34 on fine grid $\mathcal{T}_4$ using Gaussian spline via total variation. W_1 is used as the selection method.

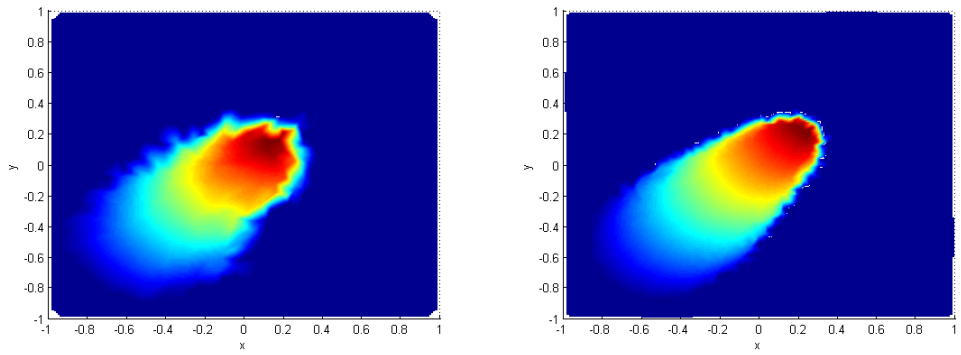

Figure 2.31: Numerical solution of ENO method of the second order after 300 time steps using polynomial recovery on triangulation $\mathcal{T}_2$ (left) and $\mathcal{T}_4$ (right). Selection method of triangles is used W_1.

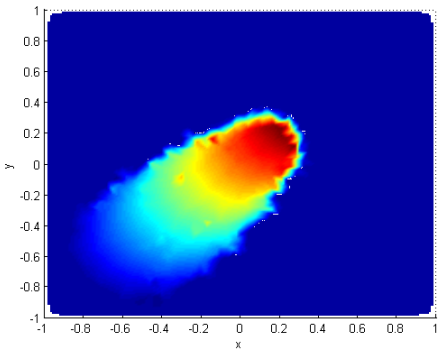

Figure 2.32: Numerical solution of ENO method of the second order after 300 time steps using polynomial recovery on triangulation $\mathcal{T}_2$ (left) and $\mathcal{T}_4$ (right). Selection method of triangles is used W_2.

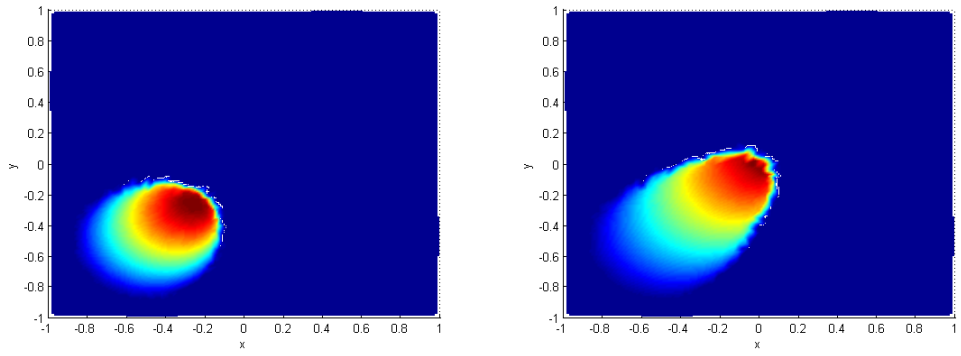

Figure 2.33: Numerical solution after 50 (left) and 150 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

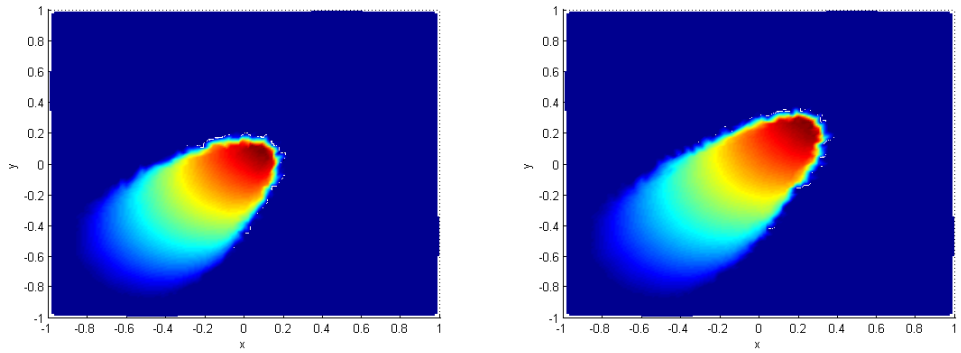

Figure 2.34: Numerical solution after 200 (left) and 300 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

2.2.3 System of Linear Equations

So far, we were interested in a quality of a numerical solution in the case of one linear or nonlinear equation in $\mathbf{R}^2$. Now we focus on a system of linear equations. System of hyperbolic partial differential equations describes a wide variety of real problems. E.g. propagation of acoustic wave is given by

$$\mathbf{q}_t + \mathbf{A}\mathbf{q}_x + \mathbf{B}\mathbf{q}_y = 0, \quad \mathbf{x} \in [-1,1] \times [-1,1], t \in (0,T),$$

where

$$\mathbf{u} = \begin{bmatrix} P \\ u \\ v \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & K_0 & 0 \\ \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & K_0 \\ 0 & 0 & 0 \\ \frac{1}{\rho} & 0 & 0 \end{bmatrix} \tag{2.34}$$

where ρ is the background of gas density and K_0 is the modulus of compressibility of gas. We use the initial conditions as follows

$$P(x, y, 0) = e^{-20\left(x^2+y^2\right)}, \quad \mathbf{x} \in [-1, 1] \times [-1, 1]$$

$$u(x, y, 0) = v(x, y, 0) = 0, \quad \mathbf{x} \in [-1, 1] \times [-1, 1]$$

We demonstrate evolution of the numerical solution of components P and u in the Figures 2.35 - 2.38, i.e. after 1, 100, 200 and 300 time steps. ENO method using Gaussian spline recovery via total variation is applied on triangulation $\mathcal{T}_4$. According to the figures, it leads to the hypothesis that the quality of the solution is very high even near the boundary where the wave travels out of the domain.

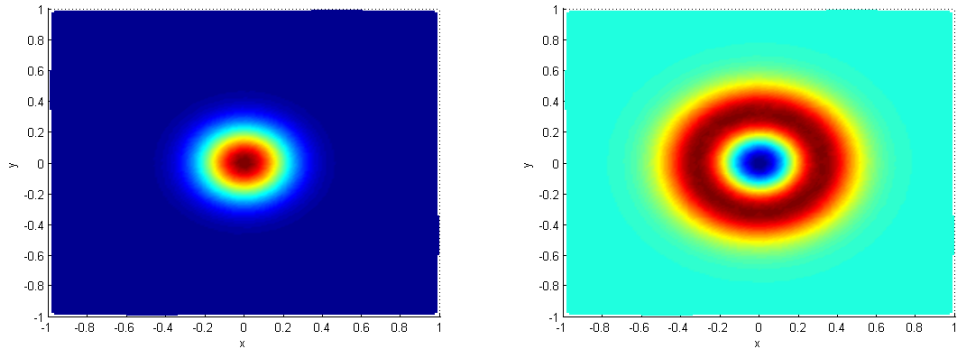

Figure 2.35: Numerical solution of component P after 1 (left) and 100 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

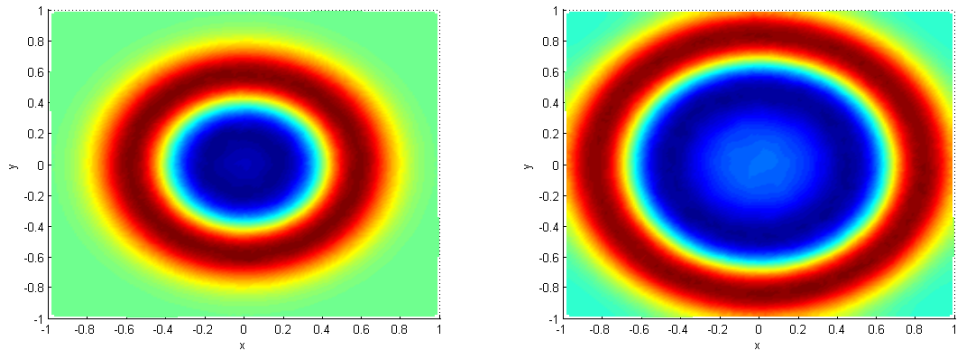

Figure 2.36: Numerical solution of component P after 200 (left) and 300 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

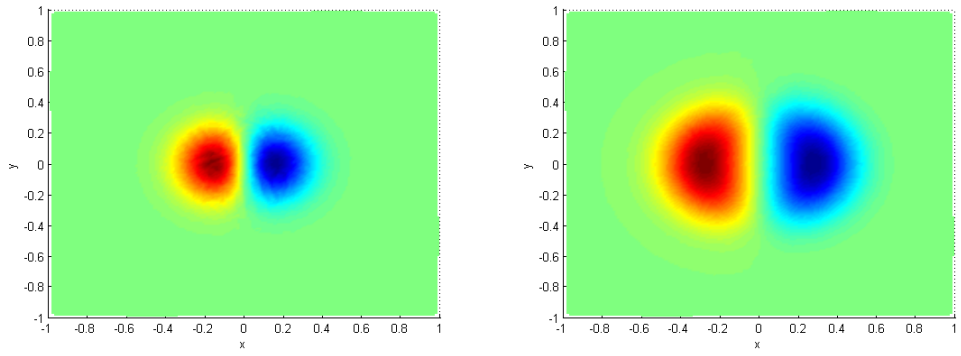

Figure 2.37: Numerical solution of component u after 1 (left) and 100 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

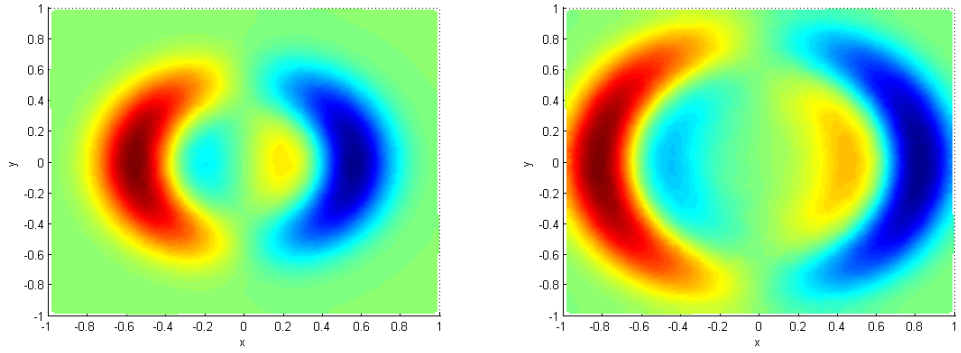

Figure 2.38: Numerical solution of component u after 200 (left) and 300 (right) time steps. Reconstruction applies Gaussian spline for $m = 6$ on triangulation $\mathcal{T}_4$. Stencil is chosen via total variation.

Let us choose the component v. The quality of the numerical solution using ENO method is compared for linear recovery and Gaussian spline via total variation in the Figures 2.39 - 2.42 again after 1, 100, 200 and 300 time steps.

According to the figures, the quality of the numerical solution shows superiority of Gaussian spline to linear polynomial. Mainly, the shape of the contours after one time step produces very low quality using polynomial recovery. According to the figures, it implies that numerical solution is smoother in case of reconstruction using linear polynomial than Gaussian spline. Especially, it is obvious in the last figure because the contours are more narrow for linear recovery. It thus implies also for system of linear equations that linear recovery is improved if radial basis functions are added.

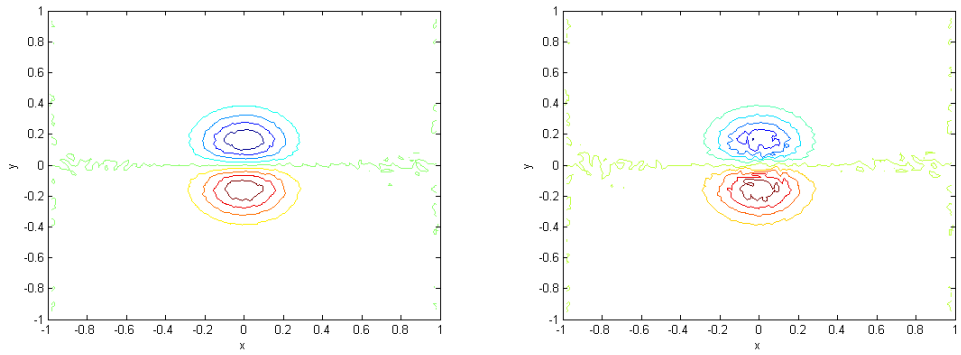

Figure 2.39: Numerical solution of component v after one time step. Reconstruction applies Gaussian spline for $m = 6$ (left) and linear polynomial (right) on triangulation $\mathcal{T}_4$.

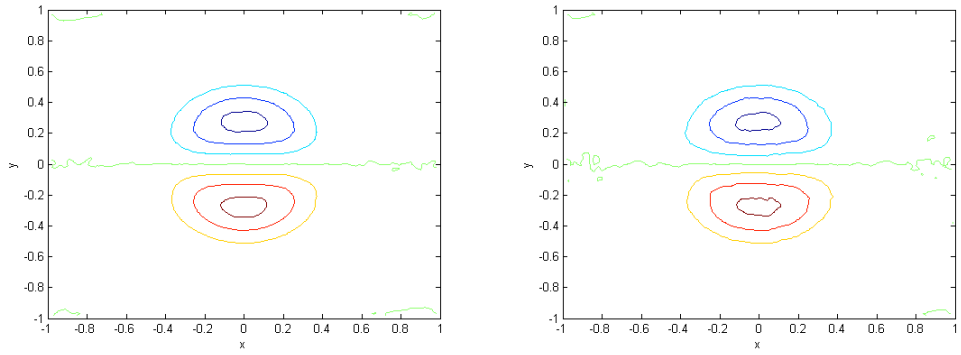

Figure 2.40: Numerical solution of component v after 100 time step. Reconstruction applies Gaussian spline for $m = 6$ (left) and linear polynomial (right) on triangulation $\mathcal{T}_4$.

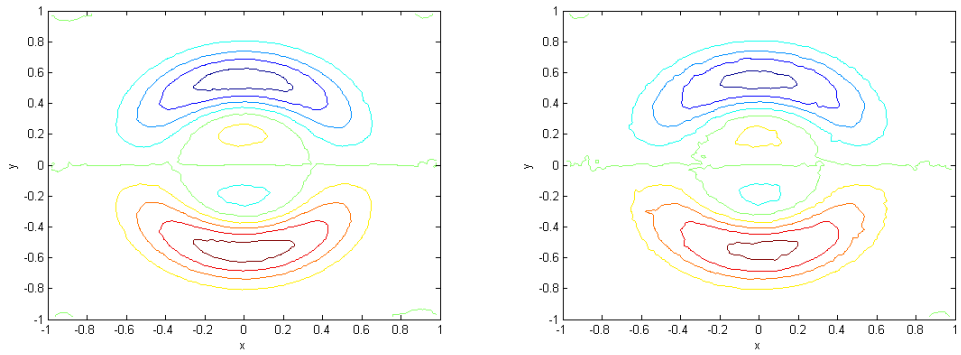

Figure 2.41: Numerical solution of component v after 200 time step. Reconstruction applies Gaussian spline for $m = 6$ (left) and linear polynomial (right) on triangulation $\mathcal{T}_4$.

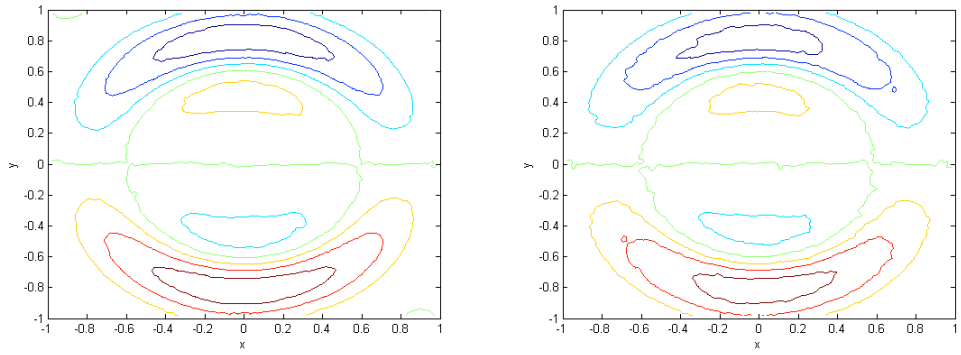

Figure 2.42: Numerical solution of component v after 300 time step. Reconstruction applies Gaussian spline for $m = 6$ (left) and linear polynomial (right) on triangulation $\mathcal{T}_4$.

Conclusion

In the thesis, we study the properties of the essentially non–oscillatory methods so–called ENO methods or ENO schemes. These finite volume methods use for the reconstruction of solution in spatial variable (or variables) a flexible stencil. The introduction to the ENO schemes was presented for the approximation of the solution in the case of 1D hyperbolic conservation law. The second primary question studied in the thesis was the extension of ENO method by the usage of radial basis functions or combination of polynomials and RBFs. Thus, the second part of this thesis is devoted to the analysis of the 2D hyperbolic conservation laws.

Quality of the numerical solution is increased for a higher degree of polynomial used for the reconstruction for both 1D and 2D problems. Therefore, we focused our attention on ENO methods which use polynomial recovery. Moreover, we studied their enhancement by Gaussian spline or thin plate spline for 2D problem. These techniques were compared to the reconstruction based only on RBF too.

ENO method was implemented for both continuous and discontinuous initial conditions. According to the numerical experiments, ENO method provides similar quality of the numerical solution using Gaussian spline or thin plate spline. The results show superiority of the reconstruction using radial basis functions to linear recovery. Mostly, transport equation was tested due to the fact that the analytical solution is known. Thus, we can determine the error of the numerical solution. This implies that at least four radial basis functions are required to improve linear reconstruction. But higher efficiency of the algorithm is required. In case of reconstruction using only radial basis functions without linear polynomial, the quality of the numerical solution changes according to various initial conditions.

A radial basis function, which is generated using Gaussian function, depends on parameter ϵ. Various values of the parameter ϵ almost do no affect reconstruction using Gaussian spline. On contrary, an approximation using only radial basis functions depends significantly on a value of the parameter ϵ.

Two seletion methods of triangles were introduced for 2D problem. First selection method W_1 provides higher quality of the numerical solution than the second selection method W_2 which has tendency to oscillate in case of reconstruction using only RBFs or in the case of nonlinear equation. The procedures W_1 and W_2 determine possible configurations of triangles for stencil.

A total variation or a sum of the coefficients (i.e. the coefficients of linear combination of radial basis functions and linear polynomial) were applied to decide which stencil is the most appropriate so that the numerical solution does not oscillate. In case of polynomial reconstruction, the connection between measurement of the oscillations using coefficients and total variation is proved. This connection is not available in case of reconstruction using radial basis functions, we thus apply both criterion. Experiments show similar results in case of Gaussian spline reconstruction via both coefficients and total variation. Higher quality of the numerical solution was obtained using the sum of coefficients than the total variation in case of an approximation of RBFs without a linear polynomial.

The numerical experiments approved that the ENO schemes based on the Gaussian splines increase the quality and the precision of the approximate solution. The recovery has to use at least 4 radial basis functions accompanied with linear polynomial. On contrary, usage of standalone RBF recovery cannot be recommended due to its high dependence on the value of parameter ϵ.

Bibliography

[1] Harten, A. – Engquist, B. – Osher, S. – Chakravarthy, S. R.: *Uniformly High Order Accurate Essentially Non-oscillatory Schemes III*, Journal of Computational Physics, 131 (1997), pp. 3-47.

[2] Abgrall, R.: *On Essentially Non–oscillatory Schemes on Unstructed Meshes: Analysis and Implementation*, Journal of Computational Physics, 114 (1994), pp. 45-58.

[3] Shu, Chi-Wang: *High Order Numerical Methods for Time Dependent Hamilton–Jacobi Equations*, Lecture Notes, Division of Applied Mathematics, Brown University, 2007.

[4] Shu, Chi-Wang: *Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws*, ICASE Report 97-65, 1997.

[5] Harten, A. – Chakravarthy, S. R.: *Multi–dimensional ENO Schemes for General Geometries*, ICASE Report 91-76, 1991.

[6] Sonar, T.: *On Families of Pointwise Optimal Finite Volume ENO Approximations*, SIAM Journal of Numerical Analysis, 35 (1998), pp. 2350-2369.

[7] Sonar, T.: *Optimal Recovery Using Thin Plate Splines in Finite Volume Methods for the Numerical Solution of Hyperbolic Conservation Laws*, IMA Journal of Numerical Analysis, 16 (1996), pp. 549-581.

[8] Sonar, T.: *On the Construction of Essentially Non-Oscillatory Finite Volume Approximations to Hyperbolic Conservation Laws on General Triangulations: Polynomial Recovery, Accuracy and Stencil Selection*, Computer Methods in Applied Mechanics and Engineering, 140 (1997), pp. 157-181.

[9] Fasshauer, G. E.: *Meshfree Approximation Methods with MATLAB*, World Scientific Publishers Co. Pte. Ltd, 2007.

[10] Aboiyar, T. – Georgoulis, E. H. – Iske, A.: *High Order Weno Finite Volume Schemes Using Polyharmonic Spline Reconstruction*, University of Leicester, Proceedings of the International Conference on Numerical Analysis and Approximation Theory, Cluj-Napoca, pp. 113-126, 2006.

[11] Chambolle, A.: *An Algorithm for Total Variation Minimization and Applications*, Journal of Mathematical Imaging and Vision, 20(2004), pp. 89-97.

[12] Brandner, M. – Egermaier, J. – Kopincová, H.: *Numerické metody pro řešení evolučních parciálních diferenciálních rovnic*, ZČU Plzeň, 2011.

[13] Iske, A. – Sonar, T.: *On the Structure of Function Spaces in Optimal Recovery of Point Functionals for ENO-Schemes by Radial Basis Functions*, Numerische Mathematik, 74 (1996), pp. 177-201.

[14] Buhmann, M. D.: *Radial basis functions: Theory and Implementation*, Cambridge University press, 2003.

[15] Fasshauer, G. E. – Zhang, J. G.: *On Choosing "Optimal" Shape Parameters for RBF Approximation*, Numerical Algorithms, 45 (2007), pp. 345-368.

[16] Turnerová, E.: *Radial Basis Functions in Mathematical Applications*, Bachelor thesis, ZČU Plzeň, 2011.

[17] Jameson, A.: *Positive Schemes and Shock Modelling for Compressible Flows*, International Journal for Numerical Methods in Fluids, 20 (1995), pp. 743-776.

[18] Donato, G. – Belongie, S.: *Approximation Methods for Thin Plate Spline Mappings and Principal Warps*, 7th European Conference on Computer Vision Part III Lecture Notes in Computer Science, Copenhagen, Denmark, 2002, pp. 21-31.

[19] LeVeque, R., J.: *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.

[20] Gottlieb, S. – Shu, Chi-Wang: *Total Variation Diminishing Runge – Kutta Schemes*, Mathematics of Computation, 67 (1998), pp. 73-85.

[21] Hu, Ch.: *Numerical Methods for Hyperbolic Equations on Unstructured Meshes*, thesis, Brown University, 1999.

[22] Hu, Ch. – Shu, Chi-Wang: *Weighted Essentially Non-Oscillatory Schemes on Triangular Meshes*, Journal of Computational Physics, 150 (1999), pp. 97-127.

[23] Ollivier, C. F.: *Quasi–ENO Schemes for Unstructured Meshes Based on Unlimited Data - Dependent Least - Squares Reconstruction*, Journal of Computational Physics, 133 (1997), pp. 6-17.

[24] Carr, J. – Beatson, R. – Cherri, J. – Mitchell, T. – Fright, W. – McCallum, B.: *Reconstruction and representation of 3D objects with radial basis functions*, Proceedings of SIGGRAPH 01, Los Angeles, 2001, pp. 67-76.

[25] Xiaogang, J. – Hanqiu, S. – Qunsheng, P.: *Subdivision interpolating implicit surfaces*, Computers and Graphics, 27 (2003), pp. 763-772.