Chapter 4

WEIGHTED ESSENTIALLY NON-OSCILLATORY FINITE DIFFERENCE AND FINITE VOLUME SCHEMES

High order WENO schemes are one class of high resolution methods suitable for solving hyperbolic conservation laws which shocks and possible discontinuous or sharp gradient regions exist in the solutions. In this study, WENO finite difference and finite volume methods were utilized to solve macroscopic continuum traffic models.

WENO schemes are developed based on the successful ENO schemes. The first WENO scheme was constructed by Liu and colleagues (1994) for a third order finite volume version in one space dimension. Later, Jiang and Shu (1996) developed third and fifth order finite difference WENO schemes in multi-space dimensions with a general framework for the design of the smoothness indicators and nonlinear weights. Very high order finite difference WENO schemes (for order between 7 and 11) have been presented by Balsara and Shu (2000).

In this section, WENO schemes are described for 1D conservation laws

$$\frac{\partial k(x,t)}{\partial t}+\frac{\partial f(k(x,t))}{\partial x}=0, \qquad (4.1)$$

equipped with suitable initial and boundary conditions. Here k is the conserved quantity while f is the flux.

4.1 TVD Runge-Kutta Time Discretization

Before introducing the spatial discretizations, the time discretization shall be first discussed. For the high resolution spatial discretizations applied in this research, the study used a class of high order nonlinearly stable Runge-Kutta time discretizations. A distinctive feature of this class of time discretizations is that they are convex combinations of first order forward Euler steps, hence they maintain strong stability properties in any semi-norm, e.g. total variation norm, maximum norm, entropy condition, etc., of the forward Euler step. These methods were first developed by Shu (1988), and later generalized by Gottlieb and Shu (1998). The most popular scheme in this class is the following third TVD Runge-Kutta method for solving

$$\frac{\partial k}{\partial t} = L(k,t), \tag{4.2}$$

where $L(k,t)$ is a spatial discretization operator:

$$\begin{aligned} k^{(1)} &= k^n + \Delta t L(k^n, t^n), \\ k^{(2)} &= \frac{3}{4}k^n + \frac{1}{4}k^{(1)} + \frac{1}{4}\Delta t L(k^{(1)}, t^n + \Delta t), \\ k^{n+1} &= \frac{1}{3}k^n + \frac{2}{3}k^{(2)} + \frac{2}{3}\Delta t L(k^{(2)}, t^n + \frac{1}{2}\Delta t). \end{aligned} \tag{4.3}$$

4.2 Weighted Essentially Non-Oscillatory Schemes

WENO spatial discretizations were introduced in the following sections. This study considered WENO reconstruction and approximation first, and then the finite difference and finite volume formulations were described.

4.2.1 WENO Reconstruction and Approximation

This section concentrates on WENO interpolation and approximation in one space dimension.

Assume a grid

$$a = x_{0.5} < x_{1.5} < \cdots < x_{N-0.5} < x_{N+0.5} = b\,. \tag{4.4}$$

Cells, cell centers, and cell sizes are respectively defined as follows

$$I_i \equiv [x_{i-0.5}, x_{i+0.5}],\ \ x_i \equiv \frac{1}{2}(x_{i-0.5} + x_{i+0.5}),\ \ \Delta x_i \equiv x_{i+0.5} - x_{i-0.5}, \qquad i = 1,2,\ldots,N\,. \tag{4.5}$$

With the cell averages of a function $v(x)$:

$$\bar{v}_i \equiv \frac{1}{\Delta x_i}\int_{x_{i-0.5}}^{x_{i+0.5}} v(x)dx, \qquad i = 1,2,\ldots,N\,, \tag{4.6}$$

a polynomial $p_i(x)$ of degree at most $k-1$, for each cell I_i can be found, such that it is a k-th order accurate approximation to the function $v(x)$ inside I_i:

$$p_i(x) = v(x) + O(\Delta x^k) \qquad x \in I_i,\ \ i = 1,2,\ldots,N\,. \tag{4.7}$$

Then this gives the k-th order accurate approximations to the functions $v(x)$ at the cell boundaries

$$v^-_{i+0.5} = p_i(x_{i+0.5}), \qquad v^+_{i+0.5} = p_i(x_{i-0.5}), \qquad i = 1,2,\ldots,N\,. \tag{4.8}$$

WENO was developed to improve upon ENO in the idea that using a convex combination of all of the candidate stencils to form the reconstruction instead of using only one of them. Suppose the k candidate stencils

$$S_r(i) = \{x_{i-r}, \cdots, x_{i-r+k-1}\}, \qquad r = 0,1,\ldots,k-1 \tag{4.9}$$

produce k different reconstructions to the value $v_{i+0.5}$,

$$v^{(r)}_{i+0.5} = \sum_{j=0}^{k-1} c_{rj}\bar{v}_{i-r+j}\,, \qquad r = 0,1,\ldots,k-1 \tag{4.10}$$

where

$$c_{rj} = \sum_{m=j+1}^{k} \frac{\sum\limits_{\substack{l=0\\ l\neq m}}^{k} \prod\limits_{\substack{q=0\\ q\neq m,p}}^{k} (r-q+1)}{\prod\limits_{\substack{l=0\\ l\neq m}}^{k} (m-l)} . \qquad (4.11)$$

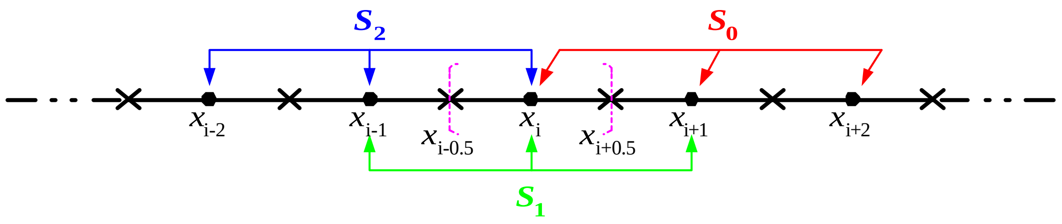

Figure 4.1. The candidate stencils for k = 3.

WENO reconstruction would take a convex combination of all $v_{i+0.5}^{(r)}$ defined in Eqn. (4.10) as a new approximation to the cell boundary value $v(x_{i+0.5})$:

$$v_{i+0.5} = \sum_{r=0}^{k-1} w_r v_{i+0.5}^{(r)} . \qquad (4.12)$$

Obviously, the choice of the weights w_r would be crucial to the success of WENO reconstruction.

$$w_r \geq 0, \qquad \sum_{r=0}^{k-1} w_r = 1 \qquad (4.13)$$

is required for stability and consistency.

If the function $v(x)$ is smooth in all of the candidate stencils (4.9), there are constants d_r such that

$$v_{i+0.5} = \sum_{r=0}^{k-1} d_r v_{i+0.5}^{(r)} = v(x_{i+0.5}) + O(\Delta x^{2k-1}) . \qquad (4.14)$$

For example, d_r for $1 \leq k \leq 3$ are given by

$$d_0 = 1, \qquad k = 1;$$
$$d_0 = \frac{2}{3},\ d_1 = \frac{1}{3}, \qquad k = 2;$$
$$d_0 = \frac{3}{10},\ d_1 = \frac{3}{5},\ d_2 = \frac{1}{10}, \qquad k = 3.$$

In this smooth case, the weights would like to be hold

$$w_r = d_r + O(\Delta x^{k-1}), \qquad r = 0, \ldots k-1 \tag{4.15}$$

which would imply $(2k-1)$-th order accuracy:

$$v_{i+0.5} = \sum_{r=0}^{k-1} w_r v_{i+0.5}^{(r)} = v(x_{i+0.5}) + O(\Delta x^{2k-1})\ . \tag{4.16}$$

When the function $v(x)$ has a discontinuity in one or more of the stencils (4.9), it would be hoped that the corresponding weight(s) w_r to be essentially 0, to emulate the successful ENO idea. This consideration lead to the following form of weights:

$$w_r = \frac{\boldsymbol{a}_r}{\sum_{s=0}^{k-1} \boldsymbol{a}_s}, \qquad r = 0, \ldots k-1 \tag{4.17}$$

with

$$\boldsymbol{a}_r = \frac{d_r}{(\boldsymbol{e} + \boldsymbol{b}_r)^2}\ . \tag{4.18}$$

Here $\boldsymbol{e} > 0$ is a parameter to avoid the denominator to become 0 and is usually taken as $\boldsymbol{e} = 10^{-6}$ in the computation. $\boldsymbol{b}_r$ are the so-called “smooth indicators” of the stencil $S_r(i)$.

When $k = 2$,

$$\begin{aligned} \boldsymbol{b}_0 &= (\overline{v}_{i+1} - \overline{v}_i)^2, \\ \boldsymbol{b}_1 &= (\overline{v}_i - \overline{v}_{i-1})^2. \end{aligned} \tag{4.19}$$

For $k = 3$,

$$\boldsymbol{b}_0 = \frac{13}{12}\left(\overline{v}_i - 2\overline{v}_{i+1} + \overline{v}_{i+2}\right)^2 + \frac{1}{4}\left(3\overline{v}_i - 4\overline{v}_{i+1} + \overline{v}_{i+2}\right)^2,$$
$$\boldsymbol{b}_1 = \frac{13}{12}\left(\overline{v}_{i-1} - 2\overline{v}_i + \overline{v}_{i+1}\right)^2 + \frac{1}{4}\left(\overline{v}_{i-1} - \overline{v}_{i+1}\right)^2, \tag{4.20}$$
$$\boldsymbol{b}_2 = \frac{13}{12}\left(\overline{v}_{i-2} - 2\overline{v}_{i-1} + \overline{v}_i\right)^2 + \frac{1}{4}\left(\overline{v}_{i-2} - 4\overline{v}_{i-1} + 3\overline{v}_i\right)^2.$$

Finally, for each cell I_i, with the given cell average $\{\overline{v}_i\}$ of a function $v(x)$, upwind biased $(2k-1)$-th order approximations to the function $v(x)$ at the cell boundaries, denoted by $v^+_{i-0.5}$ and $v^-_{i+0.5}$, are obtained.

4.2.2 Finite Difference Formulation

First, It is assumed that the grid is uniform, and Eqn. (4.1) is solved directly by using a conservative approximation to the spatial derivative

$$\frac{dk_i(t)}{dt} = -\frac{1}{\Delta x}\left(\hat{f}_{i+0.5} - \hat{f}_{i-0.5}\right), \tag{4.21}$$

where $k_i(t)$ is the numerical approximation to the point value $k(x_i,t)$, and the numerical flux

$$\hat{f}_{i+0.5} = \hat{f}(k_{i-r},\ldots,k_{i+s})$$

satisfies the following conditions:

1. $\hat{f}$ is a Lipschitz continuous function in all the arguments.

2. $\hat{f}$ is consistent with the physical flux, i.e. $\hat{f}(k_{i-r},\ldots,k_{i+s}) = f(k)$.

Then the Roe speed

$$\overline{a}_{i+0.5} \equiv \frac{f(k_{i+1}) - f(k_i)}{k_{i+1} - k_{i+1}} \tag{4.22}$$

shall be computed. If $\overline{a}_{i+0.5} \geq 0$, the numerical flux would be taken as $\hat{f}_{i+0.5} = v^-_{i+0.5}$.

Otherwise, it shall be taken as $\hat{f}_{i+0.5} = v^{+}_{i+0.5}$. Finally, the scheme (4.21) can be formed.

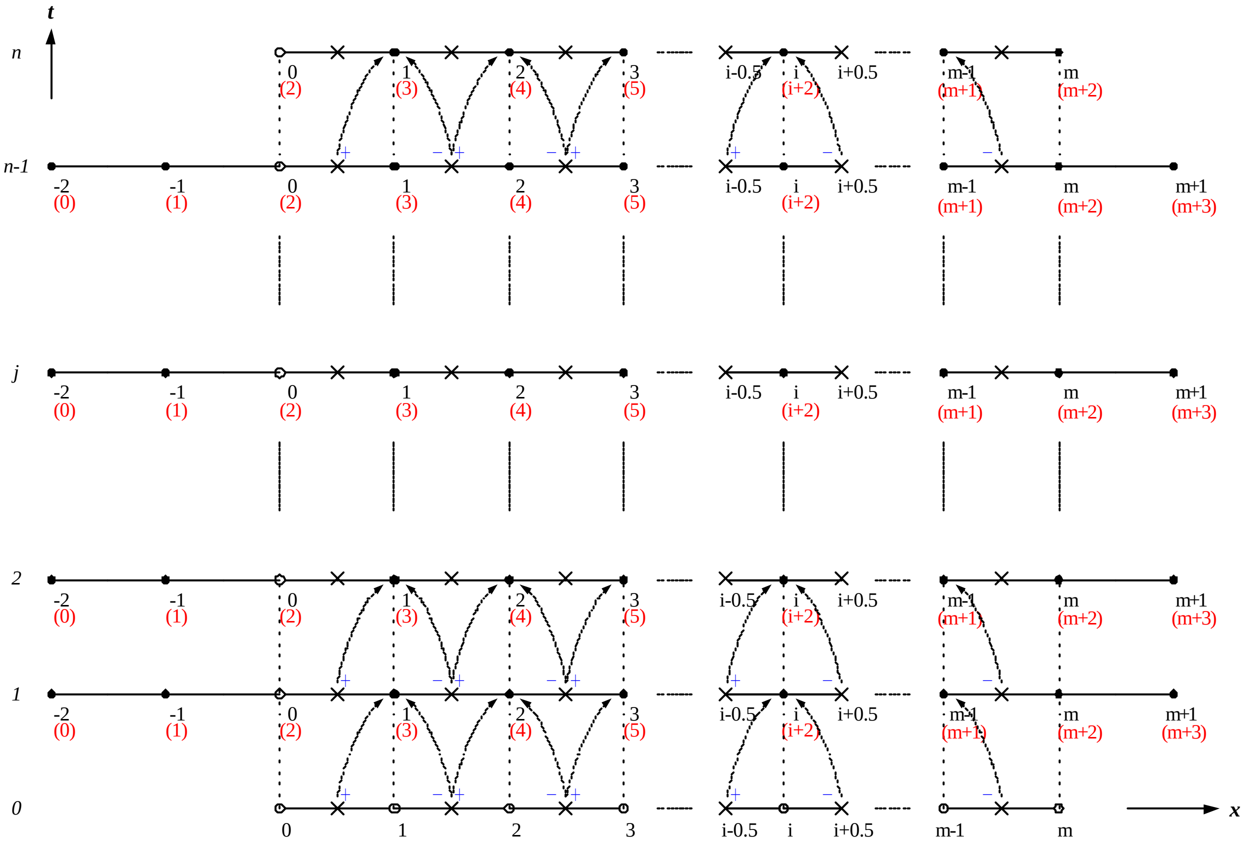

Figure 4.2. WENO Finite Difference Grid Structure.

4.2.3 Finite Volume Formulation

For finite volume schemes, Eqn. (4.1) wouldn' t be solved directly, but by its integrated version. Eqn. (4.1) is integrated over the interval I_i to obtain

$$\frac{d\bar{k}(x_i,t)}{dt} = -\frac{1}{\Delta x_i}\left(f(k(x_{i+0.5},t)) - f(k(x_{i-0.5},t))\right), \tag{4.23}$$

where

$$\bar{k}(x_i,t) \equiv \frac{1}{\Delta x_i}\int_{x_{i-0.5}}^{x_{i+0.5}} k(x,t)dx \tag{4.24}$$

is the cell average. Finite volume schemes approximate Eqn. (4.23) by the following conservative formulation

$$\frac{d\bar{k}_i(t)}{dt} = -\frac{1}{\Delta x_i}\left(\hat{f}_{i+0.5} - \hat{f}_{i-0.5}\right), \tag{4.25}$$

where $\bar{k}_i(t)$ is the numerical approximation to the cell average $\bar{k}(x_i,t)$, and the numerical flux $\hat{f}_{i+0.5}$ is defined by

$$\hat{f}_{i+0.5} = h\left(k^{-}_{i+0.5}, k^{+}_{i+0.5}\right) \tag{4.26}$$

with the value obtained by the WENO reconstruction.

The two-argument function h in Eqn. (4.26) is a monotone flux. It satisfies:

1. $h(a,b)$ is a Lipschitz continuous function in both arguments.

2. $h(a,b)$ is a non-decreasing function in a and a nonincreasing function in b.

3. $h(a,b)$ is consistent with the physical flux f, i.e. $h(a,a) = f(a)$.

In the calculation, this study uses the simple and inexpensive Lax-Friedrichs flux:

$$h(a,b) = \frac{1}{2}\left[f(a) + f(b) - \alpha(b-a)\right] \tag{4.27}$$

where $\alpha = \max_u \left|f'(k)\right|$ is a constant. The maximum is taken over the relevant range of k.

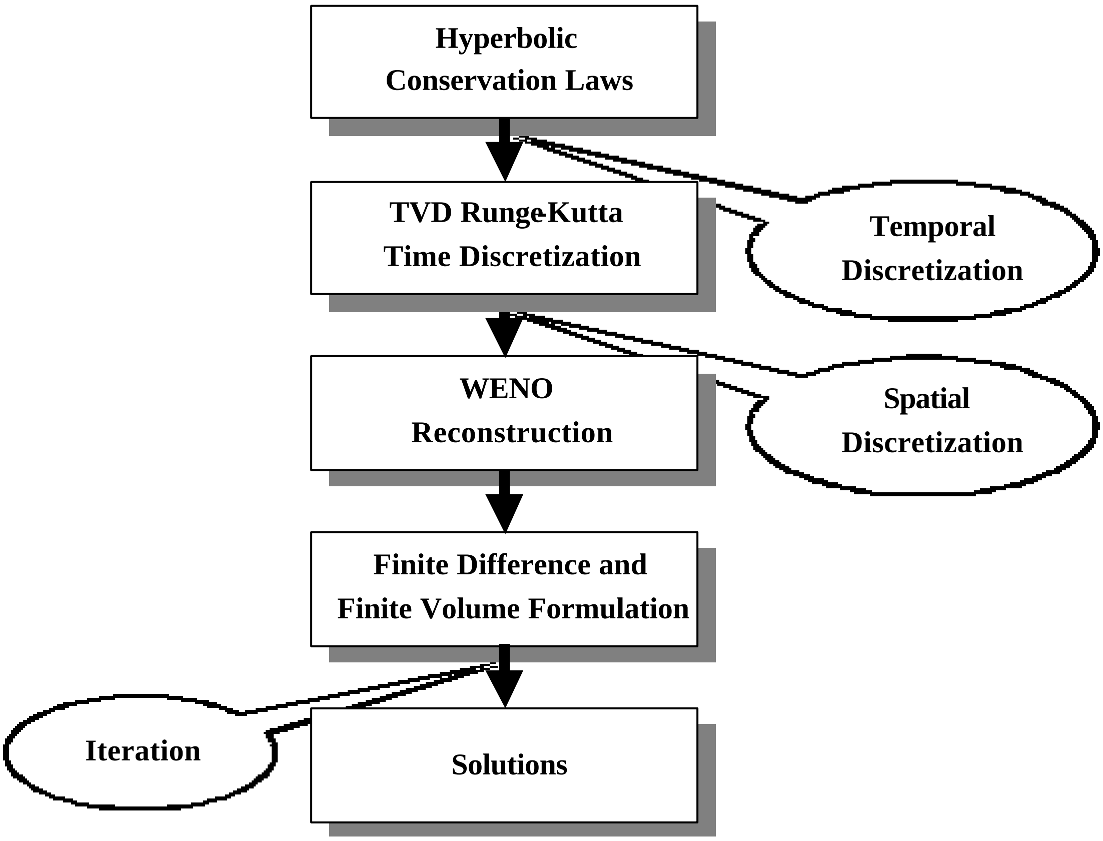
Hyperbolic Conservation Laws
TVD RungeKutta Time Discretization
Temporal Discretization
WENO Reconstruction
Spatial Discretization
Finite Difference and Finite Volume Formulation
Iteration
Solutions

Figure 4.3. Procedure of WENO scheme.

4.3 WENO Schemes for High Order Continuum Traffic Flow Models

High order continuum traffic flow models consist of hyperbolic *m* × *m* systems of PDEs, i.e. the Jacobian $f'(k)$ has m real eigenvalues

$$l_1(k) \leq \ldots \leq l_m(k)\,, \tag{4.28}$$

and a complete set of independent eigenvectors

$$r_1(k), \ldots, r_m(k). \tag{4.29}$$

The matrix whose columns are eigenvectors (4.29) is denoted by

$$R(k) = \left(r_1(k), \ldots, r_m(k)\right). \tag{4.30}$$

Then clearly

$$R^{-1}(k)\,f'(k)R(k)=\Lambda(k)\,, \tag{4.31}$$

where $\Lambda(k)$ is the diagonal matrix with $\lambda_1(k),\ldots,\lambda_m(k)$ on the diagonal. Notice that the rows of $R^{-1}(k)$, denoted by $l_1(k),\ldots,l_m(k)$ (row vectors), are left eigenvectors of $f'(k)$:

$$l(k)f'(k)=\lambda_i(k)l(k),\qquad i=1,\ldots,m\,. \tag{4.32}$$

This study applied the WENO schemes in a component-by-component fashion to solve high order continuum traffic flow models. Component-wise finite difference and finite volume formulation were shown in the following:

4.3.1 Component-wise Finite Difference Formulation

For the finite difference formulation, a smooth flux splitting (4.27) is again needed. The condition (4.27) now becomes that the two Jacobians

$$\frac{\partial f^{+}(k)}{\partial k},\ \frac{\partial f^{-}(k)}{\partial k} \tag{4.33}$$

are still diagonalizable (preferably by the same eigenvectors $R(k)$ as for $f'(k)$), and have only non-negative and non-positive eigenvalues, respectively.

In the scalar case, the exact Riemann solver gives the Godunov flux. Exact Riemann solver can be obtained for many systems including the Euler equations of compressible gas. However, it is usually very costly to get this solution. In practice, approximate Riemann solvers are usually good enough. As in the scalar case, the quality of the solution is usually very sensitive to the choice of approximate Riemann solvers for lower order schemes (first or second order), but this sensitivity decreases with an increasing order of accuracy. The simplest approximate Riemann solver is again the Lax-Friedrichs solver (4.27), except that here the constant α is taken as

$$\alpha=\max_{u}\max_{1\le j\le m}\left|\lambda_j(k)\right|, \tag{4.34}$$

where $l_j(k)$ are the eigenvalues of the Jacobian $f'(k)$. The maximum is again taken over the relevant range of k.

We recommend the Lax-Friedrichs flux splitting (4.27), with a given by Eqn. (4.34), because of its simplicity and smoothness. A somewhat more complicated Lax-Friedrichs type flux splitting is:

$$f^{\pm}(k) = \frac{1}{2}\left(f(k) \pm R(k)\overline{\Lambda}R^{-1}(k)k \right),$$

where $R(k)$ and $R^{-1}(k)$ are defined in Eqn. (4.30), and

$$\overline{\Lambda} = diag(\,\overline{l}_1, \ldots, \overline{l}_m)$$

where $\overline{l}_j = \max_u \left| l_j(k) \right|$, and the maximum is again taken over the relevant range of k. This way the dissipation is added in each field according to the maximum size of eigenvalues in that field, not globally. One could also use other flux splittings, such as the Van Leer splitting for gas dynamics. However, for higher order schemes, the flux splitting must be sufficiently smooth in order to retain the order of accuracy.

With these flux splittings, the scalar recipes can be used again to form the finite difference scheme, i.e just computing the positive and negative fluxes $\hat{f}^{+}_{i+\frac{1}{2}}$ and $\hat{f}^{-}_{i+\frac{1}{2}}$ component by component. The detailed procedure is shown as follows:

1. Find a flux splitting. The simplest example is the Lax-Friedrichs flux-splitting (4.27), with a given by Eqn. (4.34);
2. For each component of the solution k, apply the scalar procedure to reconstruct the corresponding component of the numerical flux $\hat{f}_{i+\frac{1}{2}}$;
3. Form the scheme (4.21).

4.3.2 Component-wise Finite Volume Formulation

For the finite volume formulation, the reconstruction is made by WENO for each of the components of u separately. This produces the left and right values $k^{\pm}_{i+\frac{1}{2}}$ at the cell interface $x_{i+\frac{1}{2}}$. An exact or approximate Riemann solver, $h(k^{-}_{i+\frac{1}{2}}, k^{+}_{i+\frac{1}{2}})$, is then used to build the scheme (4.25)-(4.26). The exact Riemann solver is given by the exact solution of Eqn. (4.1) with the following step function as initial condition

$$k(x,0) = \begin{cases} k^{-}_{i+\frac{1}{2}}, & x \le 0 \\ k^{+}_{i+\frac{1}{2}}, & x > 0 \end{cases}, \qquad (4.35)$$

evaluated at the center x=0. Notice that the solution to Eqn. (4.1) with the initial condition (4.35) is self-similar, i.e., it is a function of the variable $x = \frac{x}{t}$, hence is constant along x=0. If this solution is denoted by $k_{i+\frac{1}{2}}$, then the flux is taken as

$$h(k^{-}_{i+\frac{1}{2}}, k^{+}_{i+\frac{1}{2}}) = f(k_{i+\frac{1}{2}}).$$

The detailed procedure is shown as follows:

1. For each component of the solution $\bar{k}$, apply the scalar WENO procedure to reconstruct the corresponding component of the solution at the cell interfaces, $k^{\pm}_{i+\frac{1}{2}}$ for all i;

2. Apply an exact or approximate Riemann solver to compute the flux $\hat{f}_{i+\frac{1}{2}}$ for all i in Eqn. (4.26);

3. Form the scheme (4.25).