Meshless Difference Methods with Adaptation and High Resolution

Qun Lin * J.G. Rokne †

Abstract—**Partial differential equations for domains with moving boundaries and moving interfaces are solved numerically using meshless finite difference schemes. The advantage of the approach is that it is easy to add nodes where there is a requirement of higher precision and delete nodes where the precision requirements are less stringent. TVD and ENO schemes are considered in this context and some numerical examples are computed showing applications of the methods.**

Keywords: meshless finite difference method, directional difference quotient, points scattering and connecting

1 Introduction

Moving interfaces and moving boundaries occur in many physical situations. Some examples are liquid-air interfaces, ice melting in water, the deformation of bubbles, oil flowing under ice and the active interface during combustion in a solid fuel rocket engine.

The partial differential equations describing such problems must take the interfaces and boundaries into account since material properties and behaviors are normally discontinuous there. A version of the finite element methods is usually implemented in order to solve the problems numerically with the interfaces and boundaries handled in some fashion (see for example [6]). Other approaches include the smoothed particle hydrodynamics used in theoretical astrophysics computations (see [9, 8]) and radial basis function methods (see [3, 12]).

The finite element methods commonly used for these problems can be classified into methods

- where the grid is fixed and the interface is moving [16],
- where level sets are used [13, 11],
- where the grid deforms keeping pace with the moving discontinuities [15, 2, 14]
- where the grid is refined locally to the discontinuities [4],
- where there is no grid (gridless methods) [17].

The commonality of the first four kinds of methods is a dependence on a grid, either fixed or deforming in some manner. More recently gridless methods have been investigated. Examples of these are methods based on superposition of basis functions of various kinds and smoothed particle hydrodynamics. In this paper a meshless method based on finite differences is considered. In [10] a previous discussion of meshless finite difference methods was presented and some examples were given. This paper can be viewed as a continuation of that paper to problems involving moving interfaces.

There is a clear advantage in using gridless methods for these problems in that a suitably chosen set of nodes can deform and move as the boundaries and interfaces change. This also means that more nodes can be added at critical points where there are large discontinuities thus improving the approximating properties of the solutions.

In this paper a meshless method is established where the only requirement is that a coordinate system is available. The effect of moving the nodes on the computational stencils is considered followed by refining and coarsening of the stensils. A discussion of the construction of meshless difference methods with high resolution properties and some examples concludes the paper.

2 Moving nodes

The problems considered here are two dimensional problems whose solution are time-dependent velocity fields. These velocity fields are assumed to be computed at discrete time-steps. Because of this we call a velocity field at a particular time-step a *time layer*. A set of nodes $\mathcal{P} = \{P\}$ on each time layer define the points where a given problem is approximated (and it should be noted that by *node* is meant both the name of the node as well as the coordinates of the node).

We first introduce some notations:

ν is the velocity field for the present layer,

*Department of Statistics, Xiamen University, Xiamen, China
†Department of Computer Science, The University of Calgary, Calgary, Alberta, Canada

(ν_x, ν_y) is the velocity at a point (x, y),
$\beta = \Delta t \cdot \nu$, where β is an increment and
Δt is a time step,
$\xi = P + \beta$, means that a new point ξ is found
by adding β to node position P.

In the following by ν, β, ξ is meant the velocity component, increment and node position at a given point x, y.

The advantage of meshless methods is that the position of a node can be changed adaptively according to the state of the solution as time progresses.

The progression of a solution from time layer t_k to t_{k+1} is now considered.

Two approaches are normally used:

(i): In the first approach the information from the solution on the present layer is used for the following layer.

For this case a simplified directional difference quotient with respect to time is

$$\begin{aligned}&\frac{u(\xi, t_{k+1}) - u(P, t_k)}{\Delta t}\\ &= \frac{u(x+\beta_x, y+\beta_y, t_k+\Delta t) - u(x, y, t_k)}{\Delta t}\\ &= \frac{\partial u}{\partial t} + \nu_x \frac{\partial u}{\partial x} + \nu_y \frac{\partial u}{\partial y} + O(\Delta t) \qquad (1)\end{aligned}$$

where

$$\nu_x = \frac{\beta_x}{\Delta t}, \quad \nu_y = \frac{\beta_y}{\Delta t}. \qquad (2)$$

In order to approximate $\nu_x \frac{\partial u}{\partial x} + \nu_y \frac{\partial u}{\partial y}$ at a node P two known directional quotients $\frac{\Delta u}{\Delta \vec{l}_1}$ and $\frac{\Delta u}{\Delta \vec{l}_2}$ with respect to space are considered. For this two nodes P_1, P_2 are needed that are at distances $\Delta l_i = |P - P_i|$ from P, $i = 1, 2$ as indicated in Figure 1.

The directional quotients can then be written as

$$\frac{\Delta u}{\Delta \vec{l}_1} = \frac{\Delta x_1}{\Delta l_1} \cdot \frac{\partial u}{\partial x} + \frac{\Delta y_1}{\Delta l_1} \cdot \frac{\partial u}{\partial y} + O(\Delta l_1), \qquad (3)$$

$$\frac{\Delta u}{\Delta \vec{l}_2} = \frac{\Delta x_2}{\Delta l_2} \cdot \frac{\partial u}{\partial x} + \frac{\Delta y_2}{\Delta l_2} \cdot \frac{\partial u}{\partial y} + O(\Delta l_2). \qquad (4)$$

Setting $\frac{\Delta x_i}{\Delta l_i} = S_i$, $\frac{\Delta y_i}{\Delta l_i} = T_i$, $i = 1, 2$ we obtain

$$\begin{aligned}&A\frac{\Delta u}{\Delta \vec{l}_1} + B\frac{\Delta u}{\Delta \vec{l}_2} = (AS_1 + BS_2)\frac{\partial u}{\partial x}\\ &+(AT_1 + BT_2)\frac{\partial u}{\partial y} + O(\Delta l_1 + \Delta l_2) \qquad (5)\end{aligned}$$

where A, B satisfy

$$\begin{cases} AS_1 + BS_2 = \nu_x, \\ AT_1 + BT_2 = \nu_y. \end{cases}$$

This solves to

$$A = \frac{\nu_x T_2 - \nu_y S_2}{S_1 T_2 - S_2 T_1}, \quad B = \frac{\nu_y S_1 - \nu_x T_1}{S_1 T_2 - S_2 T_1} \qquad (6)$$

and we note that it is required that $S_1 T_2 \neq S_2 T_1$ and $P \neq P_i, i = 1, 2$.

Therefore we get from (1)

$$\begin{aligned}&\frac{\partial u}{\partial t} = \frac{u(\xi, t_{k+1}) - u(P, t_k)}{\Delta t}\\ &-A\frac{\Delta u}{\Delta \vec{l}_1} - B\frac{\Delta u}{\Delta \vec{l}_2} + O(\Delta t + \Delta l_1 + \Delta l_2) \qquad (7)\end{aligned}$$

using (3), (4) and (5).

(ii): In this approach, indicated in Figure 2, the solution on the next layer is used as well. The computational cost is larger than in the first approach, but it can adapt more easily to the changes in fluids over time due to the forecasting of the future velocity field. The velocity field is extrapolated as follows:

$$\nu(t + \frac{1}{2}\Delta t, \cdot) = \frac{3}{2}\nu(t, \cdot) - \frac{1}{2}\nu(t - \Delta t, \cdot) \qquad (8)$$

or

$$\nu(t + \frac{1}{2}\Delta t, \cdot) = \frac{11}{6}\nu(t, \cdot) - \frac{7}{6}\nu(t - \Delta t, \cdot) + \frac{1}{3}\nu(t - 2\Delta t, \cdot) \qquad (9)$$

and a β^* is computed iteratively by means of $\frac{dP}{dt} = \nu(P, t) \approx \frac{\beta^*}{\Delta t}$ which leads to the recursion

$$\beta^{m+1} = \Delta t \cdot \nu(t + \frac{1}{2}\Delta t, P + \frac{1}{2}\beta^m), \quad m = 0, 1, \cdots \qquad (10)$$

where $\beta^m \to \beta^*$. Thus

$$\xi = P + \beta^*. \qquad (11)$$

Therefore the simplified directional quotient with respect to time is written as

$$\begin{aligned}&\frac{u(\xi, t_{k+1}) - u(P, t_k)}{\Delta t}\\ &= \frac{u(x+\beta_x^*, y+\beta_y^*, t_k+\Delta t) - u(x, y, t_k)}{\Delta t}\\ &= \frac{\partial u}{\partial t} + \nu_x^* \frac{\partial u}{\partial x} + \nu_y^* \frac{\partial u}{\partial y} + O(\Delta t). \qquad (12)\end{aligned}$$

Similarly to the above, we can use the combination of two known directional quotients with respect to space to eliminate the influence of spatial directions from the movement of a node. That is, we have

$$A^* \frac{\Delta u}{\Delta \vec{l}_1} + B^* \frac{\Delta u}{\Delta \vec{l}_2} = \nu_x^* \frac{\partial u}{\partial x} + \nu_y^* \frac{\partial u}{\partial y} + O(\Delta l_1 + \Delta l_2) \qquad (13)$$

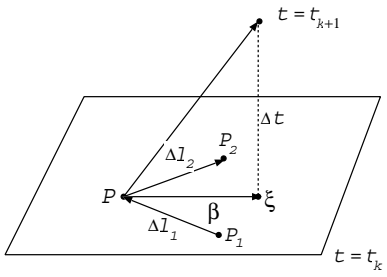


Figure 1: Directional difference quotients based on the present time layer

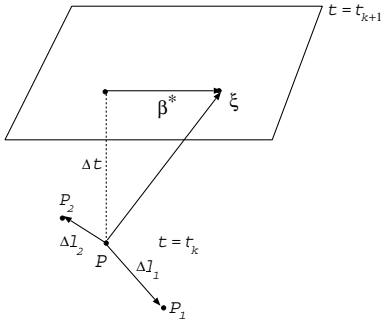


Figure 2: Directional difference quotient using future layer

where A^*, B^* satisfy

$$\begin{cases} A^*S_1 + B^*S_2 = \nu_x^*, \\ A^*T_1 + B^*T_2 = \nu_y^* \end{cases}$$

and we obtain

$$\frac{\partial u}{\partial t} = \frac{u(\xi, t_{k+1}) - u(P, t_k)}{\Delta t} - A^* \frac{\Delta u}{\Delta \vec{l}_1} - B^* \frac{\Delta u}{\Delta \vec{l}_2} + O(\Delta t + \Delta l_1 + \Delta l_2) \quad (14)$$

using (12) and (1).

In order to improve the approximation, we can construct a directional quotient on several time-layers as indicated in Figure 3.

For instance, to improve the approximation in the time direction, we have

$$\begin{aligned} &\frac{u(P+\beta_1, t_{k+1}) - u(P-\beta_{-1}, t_{k-1})}{2\Delta t} \\ &= \frac{1}{2}\frac{u(P+\beta_1, t_{k+1}) - u(P, t_k)}{\Delta t} \\ &+ \frac{1}{2}\frac{u(P, t_k) - u(P-\beta_{-1}, t_{k-1})}{\Delta t} \\ &= \frac{1}{2}\frac{\partial u}{\partial t} + \frac{1}{2}\nu_x^1\frac{\partial u}{\partial x} + \frac{1}{2}\nu_y^1\frac{\partial u}{\partial y} + \frac{1}{4}\Delta t\frac{\partial^2 u}{\partial t^2} \\ &+ \frac{1}{2}\nu_x^1\beta_y^1\frac{\partial^2 u}{\partial x \partial y} + \cdots + \frac{1}{2}\frac{\partial u}{\partial t} + \frac{1}{2}\nu_x^{-1}\frac{\partial u}{\partial x} + \frac{1}{2}\nu_y^{-1}\frac{\partial u}{\partial y} \\ &- \frac{1}{4}\Delta t\frac{\partial^2 u}{\partial t^2} - \frac{1}{2}\nu_x^{-1}\beta_y^{-1}\frac{\partial^2 u}{\partial x \partial y} - \cdots \\ &= \frac{\partial u}{\partial t} + \frac{1}{2}(\nu_x^1 + \nu_x^{-1})\frac{\partial u}{\partial x} + \frac{1}{2}(\nu_y^1 + \nu_y^{-1})\frac{\partial u}{\partial y} + \cdots \quad (15) \end{aligned}$$

After eliminating the effect of moving a node we obtain second order precision in the time direction.

We can also combine time difference quotients to eliminate the effect of spatial shifts.

$$\begin{aligned} &A\frac{u(P+\beta_1, t_{k+1}) - u(P, t_k)}{\Delta t} \\ &+ B\frac{u(P, t_k) - u(P-\beta_{-1}, t_{k-1})}{\Delta t} \\ &+ C\frac{u(P-\beta_{-1}, t_{k-1}) - u(P-\beta_*, t_{k-2})}{\Delta t} \\ &= \frac{1}{\Delta t}[Au(P+\beta_1, t_{k+1}) + (B-A)u(P, t_k) \\ &+ (C-B)u(P-\beta_{-1}, t_{k-1}) + (-C)u(P-\beta_*, t_{k-2})] \\ &= A\cdot\Delta t^{-1}u(P, t_k) + A\frac{\partial u}{\partial t} + A\nu_x^1\frac{\partial u}{\partial x} \\ &+ A\nu_y^1\frac{\partial u}{\partial y} + O(\Delta t) + (B-A)\cdot\Delta t^{-1}u(P, t_k) \\ &+ (C-B)\cdot\Delta t^{-1}u(P, t_k) - (C-B)\frac{\partial u}{\partial t} \\ &- (C-B)\nu_x^{-1}\frac{\partial u}{\partial x} - (C-B)\nu_y^{-1}\frac{\partial u}{\partial y} + O(\Delta t) \\ &- (-C)\cdot\Delta t^{-1}u(P, t_k) - 2(-C)\frac{\partial u}{\partial t} - (-C)\nu_x^*\frac{\partial u}{\partial x} \\ &- (-C)\nu_y^*\frac{\partial u}{\partial y} + O(\Delta t) = \frac{\partial u}{\partial t} + O(\Delta t) \quad (16) \end{aligned}$$

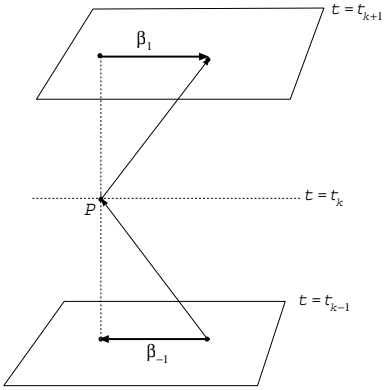


Figure 3: Directional difference multi-layer construction

where A, B, C satisfy

$$\begin{cases} A + B + C = 1, \\ \nu_x^1 A + \nu_x^{-1} B + (\nu_x^* - \nu_x^{-1}) C = 0, \\ \nu_y^1 A + \nu_y^{-1} B + (\nu_y^* - \nu_y^{-1}) C = 0. \end{cases}$$

3 Node refining and coarsening

For each interior node, we have to construct a difference scheme by means of arcs connecting adjacent nodes. Figure 4 shows two types of node stencils for star-like schemes of nine points.

If we need to increase the precision of the approximation then this can be done by combining directional quotients.

If a new node is inserted new arcs can be constructed to existing nodes. Usually, the quotient approximation of (1, 2, 3, 4) order needs a combination of (3, 6, 10, 15) nodes and (2, 5, 9, 14) directional quotients. Finding the combination coefficients requires the solution of a system of linear equations of the (2, 5, 9, 14) order. If any of the systems are singular then the singularity can be avoided by perturbating one or more of the nodes. An example of an order 2 approximation on 5 nodes requiring the solution of a 5×5 system of equations is given in [10], equations (12)-(14).

Except for the case of moving nodes, adaptive computation can add and delete nodes according to the state of the local error. An error indicator for a computational region Ω_L is therefore defined for each t_k as follows:

$$I(P, t_k) = \left| u(P, t_k) - u^k(P) \right|, \quad P \in \text{int } \Omega_L, \tag{17}$$

where $u(P, t_k)$ is the current solution and $u^k(P)$ the approximation such that $I(P, t_k)$ reflects the approximating degree of the solution around the node P on a given time-layer. Here there are two computable constants $C > 0$ and R such that a local posteriori error estimation

$$\left| u(P, t_k) - u^k(P) \right| \le C \cdot R(u^k(P)) \tag{18}$$

holds.

Adding or eliminating nodes is carried through according to the following rule:

Let $I_k^* = \max_{P \in \Omega_L} I(P, t_k)$, and let θ_{ref}, θ_{coa} be two tolerances such that: $0 < \theta_{\text{coa}} < \theta_{\text{ref}} < 1$. Then

$$\text{node } P \text{ is added if } I(P, t_k) > \theta_{\text{ref}} \cdot I_k^* \tag{19}$$

$$\text{node } P \text{ is removed if } I(P, t_k) < \theta_{\text{coa}} \cdot I_k^*. \tag{20}$$

In order to balance the opposing requirements of approximation degree and computational effort we add new nodes in Ω_L if I is large and we remove current nodes from Ω_L if I is small. In addition to this it is necessary to match the schemes for adding or removing nodes. When a new node is added, we must establish a difference scheme based on this node. Similarly when a node is eliminated, we should at the same time delete the difference scheme which is centered on this node. Sometimes, we need to also keep the balance between the number of entering arcs and that of exiting arcs for each node during addition or removal of nodes. As an example consider the scheme in Figure 5.

If we add a new node P_0 with adjacent nodes P_1, P_2, P_3, P_4, assuming that P_1, P_2, P_3, P_4 are arranged in anticlockwise order and that arc directions are $\vec{P_4P_1}, \vec{P_3P_2}$, then these two arcs are removed and the paths $P_4 \to P_0 \to P_1, P_3 \to P_0 \to P_2$ are established. If we remove the node P_0 with adjacent nodes P_1, P_2, P_3, P_4, assuming that original paths are $P_2 \to P_0 \to P_4, P_3 \to P_0 \to P_1$ then these two paths and the node P_0 are removed, and we replace them by the arcs $\vec{P_2P_4}, \vec{P_3P_1}$.

In order to simplify the programming we first mark all nodes which might possibly be used in the computation as well as the difference schemes on these nodes. At the same time we associate two flags to each node to indicate whether the scheme on that node will enter the computational process at that moment. If that node is added at a given moment, then the corresponding scheme is ac-

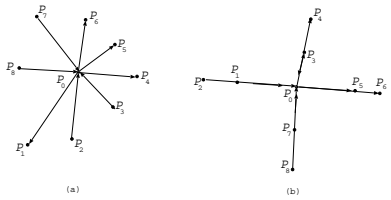
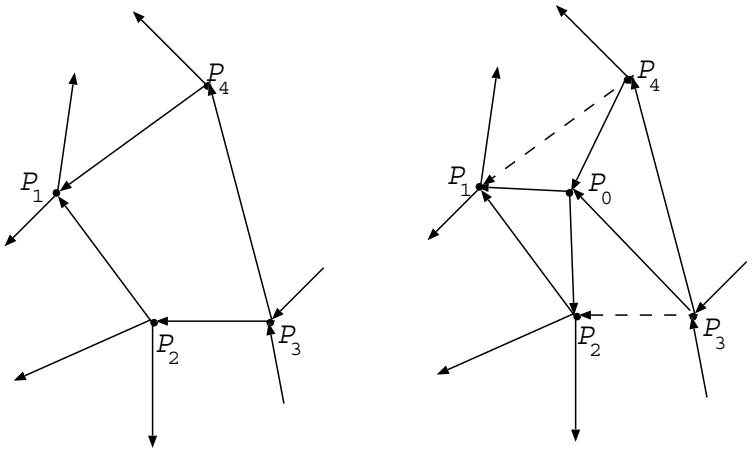
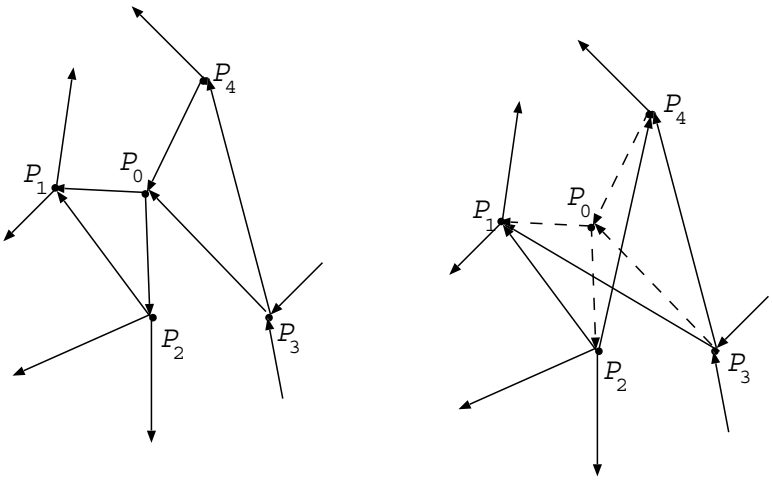


Figure 4: Nine point star-like scheme



(a) adding a node



(b) removing a node

Figure 5: Adding and removing nodes

tivated and the computation is performed. If the node is removed, then the corresponding scheme is said to be dormant.

It must be pointed out that a weakness of Lagrangian methods is that the distance between two nodes may be changed significantly as a node moves. If there is a link between two nodes and they are moved so that they are far apart then the link can be removed if the nodes are in the interior of a region. If, however, the nodes are at the edge of a region, for example near a discontinuous interface, then we cannot modify the linking line between the nodes since we want to avoid destroying the interface which the scheme represents, see Figure 6. In this case, we should add new nodes on the linking line between the two nodes along the approximating interface. If necessary, we should also consider interface indicators such as Level Sets. When we establish the corresponding schemes, it is still noted that the adjacent nodes must be established on one side of a moving interface so as to enhance the resolution for the interface.

In Figure 7 two examples of node progressions are indicated. In Figure 7(A) the nodes follow the flow of the fluids and the movements of the nodes are calculated as part of the process of approximating the equations resulting in the node structure in the lower figure. The local a posteriori estimate (18) is used to estimate the accuracy of the solution on that layer and, as shown in the figure nodes are added when condition (19)) requires it and deleted when condition (20) holds.

The sequence Figure 7(B) indicates that if there is an interface indicated by the dotted line then points on the interface should only be added not deleted even though condition (20) is satisfied.

4 Meshless TVD schemes

We now consider Total Variation Diminishing (TVD) schemes [7, 5].

The 2D scalar conservation law is stated as

$$u_t + f(u)_x + g(u)_y = 0, \tag{21}$$

or

$$u_t + \nabla \cdot F = 0, \;\; F = (f, g)^T, \;\; \nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \tag{22}$$

and

$$\frac{\partial f}{\partial u} = a(u), \;\; \frac{\partial g}{\partial u} = b(u) \tag{23}$$

that is

$$u_t + a(u)u_x + b(u)u_y = 0. \tag{24}$$

It is noted that, given any two independent directions $\vec{l_1}, \vec{l_2}$ we have

$$\frac{\partial f}{\partial \vec{l_1}} = \cos\alpha_1 \frac{\partial f}{\partial x} + \cos\beta_1 \frac{\partial f}{\partial y}, \tag{25}$$

$$\frac{\partial f}{\partial \vec{l_2}} = \cos\alpha_2 \frac{\partial f}{\partial x} + \cos\beta_2 \frac{\partial f}{\partial y} \tag{26}$$

where $\cos\alpha_i, \cos\beta_i$ are the directional cosines of $\vec{l_i}$, $i = 1, 2$. Thus,

$$\frac{\partial f}{\partial x} = A\frac{\partial f}{\partial \vec{l_1}} + B\frac{\partial f}{\partial \vec{l_2}} \tag{27}$$

where

$$\begin{cases} A\cos\alpha_1 + B\cos\alpha_2 = 1, \\ A\cos\beta_1 + B\cos\beta_2 = 0. \end{cases}$$

Similarly

$$\frac{\partial g}{\partial y} = C\frac{\partial g}{\partial \vec{l_1}} + D\frac{\partial g}{\partial \vec{l_2}} \tag{28}$$

where

$$\begin{cases} C\cos\alpha_1 + D\cos\alpha_2 = 0, \\ C\cos\beta_1 + D\cos\beta_2 = 1. \end{cases}$$

Therefore, on a given node, we can construct a "one-dimensional" scheme along a path passing through that node, and then establish a full scheme combining several different paths. Also, it will be shown in Section 7 that we can employ the splitting technique on different computational paths. In this section, we consider the difference scheme

$$u^{k+1}(P^+) = u^k(P) - r^*(\hat{f}^k_{P^+P} - \hat{f}^k_{PP^-}) \tag{29}$$

where $r^* = 2\Delta t/(|P^+P| + |PP^-|)$ and

$$\hat{f}^k_{P^+P} \equiv \hat{f}_+(u^k(P^+), u^k(P); r^*), \tag{30}$$

$$\hat{f}^k_{PP^-} \equiv \hat{f}_-(u^k(P), u^k(P^-); r^*) \tag{31}$$

are the numerical fluxes of this scheme, see Figure 8, which satisfies the following consistency condition

$$\hat{f}_+(u, u; r^*) = \hat{f}_-(u, u; r^*) = f(u). \tag{32}$$

The scheme (29) will approximate the following equation

$$u_t + A\frac{\partial f}{\partial \vec{l_1}} + B\frac{\partial f}{\partial \vec{l_2}} = 0. \tag{33}$$

From now on we do not consider moving the node. Doing this would just add a modification term written as

$$C_1^*\frac{\Delta f}{\Delta \vec{l_1}} + C_2^*\frac{\Delta f}{\Delta \vec{l_2}}$$

to the spatial direction in the scheme.

Let $L = \max\{l\}$ where l is the length of an arc. We suppose that

$$\frac{|P^+P|}{|P^+P| + |PP^-|} \to C^*, \;\; (L \to 0). \tag{34}$$

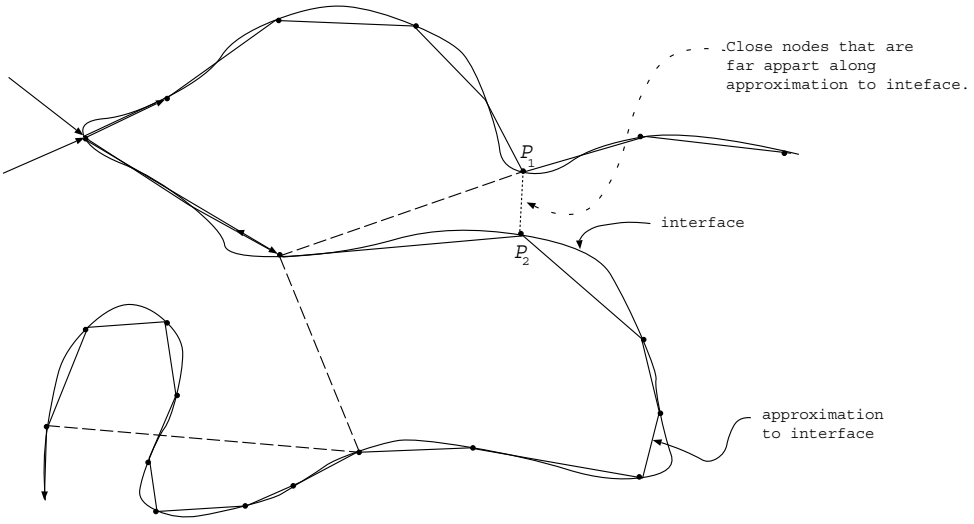


Figure 6: Close edge nodes

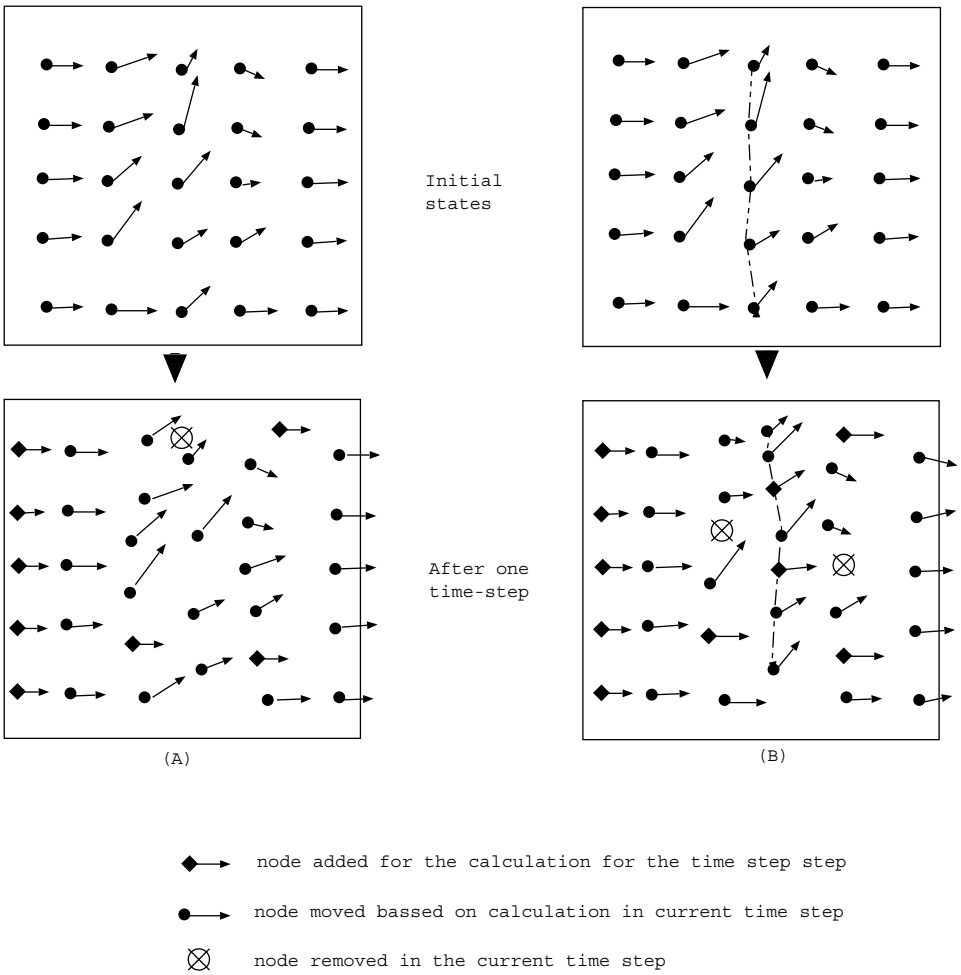


Figure 7: Moving, adding and deleting nodes

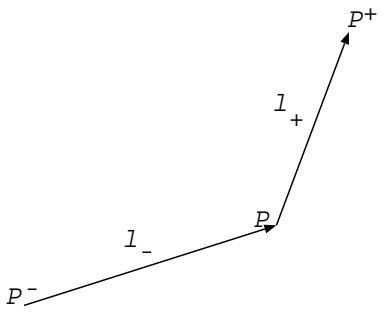


Figure 8: Path through a node

For scheme (29), we have

(i): When conditions (32) and (34) hold, the scheme is consistent. This is because

$$\hat{f}_+(u(P^+),u(P);r^*)=\hat{f}_+(u,u;r^*)$$
$$+(\hat{f}_+)'_1(u,u;r^*)[u(P^+)-u(P)]+O(L^2), \quad (35)$$
$$\hat{f}_-(u(P^+),u(P);r^*)=\hat{f}_-(u,u;r^*)$$
$$+(\hat{f}_-)'_2(u,u;r^*)[u(P^-)-u(P)]+O(L^2). \quad (36)$$

Thus

$$\frac{u(P,t+\Delta t)-u(P,t)}{\Delta t}$$
$$+\frac{\hat{f}_+(u(P^+),u(P);r^*)-\hat{f}_-(u(P),u(P^-);r^*)}{(|P^+P|+|PP^-|)/2}$$
$$=u'_t+(\hat{f}_+)'_1(u,u;r^*)\frac{u(P^+)-u(P)}{|P^+P|}$$
$$\cdot\frac{2|P^+P|}{|P^+P|+|PP^-|}$$
$$-(\hat{f}_-)'_2(u,u;r^*)\frac{u(P^-)-u(P)}{|PP^-|}$$
$$\cdot\frac{2|PP^-|}{|P^+P|+|PP^-|}+O(L)$$
$$=u'_t+2C^*\cdot(\hat{f}_+)'_1(u,u;r^*)\frac{\partial u}{\partial\vec{l}_+}$$
$$+2(1-C^*)\cdot(\hat{f}_-)'_2(u,u;r^*)\frac{\partial u}{\partial\vec{l}_-}+O(L)$$
$$=u'_t+2C^*\frac{\partial\hat{f}_+}{\partial\vec{l}_+}+2(1-C^*)\frac{\partial\hat{f}_-}{\partial\vec{l}_-}+O(L). \quad (37)$$

(ii): If $u^k(P)\to 0$ as node P tends to infinity, then scheme (29) implies that

$$\sum_{\text{all } P}u^{k+1}(P)=\sum_{\text{all } P}u^k(P), \quad (38)$$

that is, scheme (29) is conserving and we note that (29) is the flux form. We also employ another two forms in practice. Let

$$Q^k_{P^+P}=\begin{cases}\begin{cases}r^*\dfrac{f(u^k(P))+f(u^k(P^+))-2\hat{f}^k_{P^+P}}{u^k(P^+)-u^k(P)},\\ \text{when } u^k(P^+)\neq u^k(P),\end{cases}\\ \begin{cases}r^*[(\hat{f}_+)'_1(P)+(\hat{f}_+)'_2(P)],\\ \text{when } u^k(P^+)=u^k(P).\end{cases}\end{cases} \quad (39)$$

Here

$$\hat{f}^k_{P^+P}=\frac{1}{2}[f(u^k(P))+f(u^k(P^+))]$$
$$-\frac{1}{2r^*}Q^k_{P^+P}\cdot[u^k(P)-u^k(P)] \quad (40)$$

and scheme (29) can be written as

$$u^{k+1}(P^+)=u^k(P)-\frac{1}{2}r^*[f(u^k(P^+))-f(u^k(P^-))]$$
$$+\frac{1}{2}[Q^k_{P^+P}\cdot(u^k(P^+)-u^k(P)) \quad (41)$$
$$+-Q^k_{PP^-}\cdot(u^k(P)-u^k(P^-))]. \quad (42)$$

This is called the viscosity form for scheme (29) and $Q^k_{P^+P}$ is called numerical coefficient of viscosity.

Moreover, scheme (29) can still be written as the following increment form

$$u^{k+1}(P^+)=u^k(P)-C^k_{PP^-}\cdot(u^k(P)-u^k(P^-))$$
$$+D^k_{P^+P}\cdot(u^k(P^+)-u^k(P)) \quad (43)$$

where

$$C^k_{PP^-}=\frac{1}{2}(Q^k_{PP^-}+r^*a^k_{PP^-}), \quad (44)$$
$$D^k_{P^+P}=\frac{1}{2}(Q^k_{P^+P}-r^*a^k_{P^+P}) \quad (45)$$

and

$$a^k_{P^+P}=\begin{cases}\begin{cases}\dfrac{f(u^k(P^+))-f(u^k(P))}{u^k(P^+)-u^k(P)}\\ \text{when } u^k(P^+)\neq u^k(P),\end{cases}\\ \begin{cases}\left(\dfrac{\partial f}{\partial u}\right)^k(P),\\ \text{when } u^k(P^+)=u^k(P).\end{cases}\end{cases} \quad (46)$$

In the following several examples are given.

Example 1. Meshless explicit upwind scheme:

$$u^{k+1}(P^+) = u^k(P) - \frac{1}{2}r^*[f(u^k(P^+)) - f(u^k(P^-))] \quad (47)$$

$$+\frac{1}{2}r^*|a^k_{P^+P}| \cdot [u^k(P^+) - u^k(P)]$$

$$-\frac{1}{2}r^*|a^k_{PP^-}| \cdot [u^k(P) - u^k(P^-)]. \quad (48)$$

The numerical flux of the scheme is

$$\hat{f}^k_{P^+P} = \frac{1}{2}[f(u^k(P)) - f(u^k(P^+))]$$

$$-\frac{1}{2}|a^k_{P^+P}| \cdot [u^k(P^+) - u^k(P)]. \quad (49)$$

From this it follows that the numerical coefficient of viscosity is $Q^k_{P^+P} = r^*|a^k_{P^+P}|$.

Here we note that the scheme becomes

$$u^{k+1}(P^+) = u^k(P) - r^*[f(u^k(P)) - f(u^k(P^-))] \quad (50)$$

when $a^k_{P^+P} \geq 0$ and it becomes

$$u^{k+1}(P^+) = u^k(P) - r^*[f(u^k(P^+)) - f(u^k(P))] \quad (51)$$

when $a^k_{P^+P} < 0$.

Example 2. Meshless Lax-Friedrichs scheme:

$$u^{k+1}(P^+) = \frac{1}{2}[u^k(P^+) - u^k(P^-)]$$

$$-\frac{1}{2}r^*[f(u^k(P^+)) - f(u^k(P^-))]. \quad (52)$$

The numerical flux is

$$\hat{f}^k_{P^+P} = \frac{1}{2}\left[f(u^k(P)) + f(u^k(P^+))\right] - \frac{1}{2r^*}[u^k(P^+) - u^k(P)]. \quad (53)$$

From this it follows that the numerical coefficient of viscosity is $Q^k_{P^+P} = 1$.

Example 3. Meshless Lax-Wendroff scheme:

$$u^{k+1}(P^+) = u^k(P) - \frac{1}{2}r^*[f(u^k(P^+)) - f(u^k(P^-))$$

$$+\frac{1}{2}r^{*^2}(a^k_{P^+P})^2 \cdot [u^k(P^+) - u^k(P)]$$

$$-\frac{1}{2}r^{*^2}(a^k_{PP^-})^2 \cdot [u^k(P) - u^k(P^-)]. \quad (54)$$

The numerical flux of this scheme is

$$\hat{f}^k_{P^+P} = \frac{1}{2}[f(u^k(P)) + f(u^k(P^+))]$$

$$-\frac{1}{2}r^*(a^k_{P^+P})^2 \cdot [u^k(P^+) - u^k(P)]. \quad (55)$$

From this it follows that the numerical coefficient of viscosity is $Q^k_{P^+P} = (r^* a^k_{P^+P})^2$.

In the following we will discuss so-called TVD property which is of special significance in the computation of sharply discontinuous interfaces.

The difference solution $\{u^k(P)\}, u^k(P) \equiv 0$ when P is far away from the computational region is given. We call

$$\mathrm{TV}(u^k) = L \cdot \sum_{\text{all } P} |u^k(P^+) - u^k(P)| \quad (56)$$

the total variation of the difference solution. If the inequality

$$\mathrm{TV}(u^{k+1}) \leq \mathrm{TV}(u^k), \ \ \forall k \quad (57)$$

holds, then the corresponding scheme is called a TVD scheme.

For a meshless TVD scheme, we will generalize the following two familiar properties:

PROPERTY 1 *(Harten lemma [7]) For the increment form (43) of scheme (29), if*

$$C^k_{P^+P} \geq 0, \ \ D^k_{P^+P} \geq 0, \ \ C^k_{P^+P} + D^k_{P^+P} \leq 1, \ \ \forall P \quad (58)$$

then this scheme is a TVD scheme.

This is because

$$\text{(a) } u^{k+1}(P^+) = u^k(P^+)$$

$$-C^k_{P^+P} \cdot [u^k(P^+) - u^k(P)]$$

$$+D^k_{\tilde{P}P^+} \cdot [u^k(\tilde{P}) - u^k(P^+)], \quad (59)$$

$$\text{(b) } u^{k+1}(P) = u^k(P) + D^k_{P^+P} \cdot [u^k(P^+) - u^k(P)]$$

$$-C^k_{PP^-}[u^k(P) - u^k(P^-)] \quad (60)$$

(see also Figure 8). Subtracting equation (b) from equation (a) we get

$$[u^{k+1}(P^+) - u^{k+1}(P)]$$

$$= (1 - C^k_{P^+P} - D^k_{P^+P}) \cdot [u^k(P^+) - u^k(P)]$$

$$+C^k_{PP^-} \cdot [u^k(P) - u^k(P^-)]$$

$$+D^k_{\tilde{P}P^+} \cdot [u^k(\tilde{P}) - u^k(P^+)]. \quad (61)$$

Taking the absolute values on both sides and summing for all P, we obtain

$$\sum_{\text{all } P} |u^{k+1}(P^+) - u^{k+1}(P)|$$

$$\leq \sum_{\text{all } P} (1 - C^k_{P^+P} - D^k_{P^+P}) \cdot |u^k(P^+) - u^k(P)|$$

$$+ \sum_{\text{all } P} C^k_{PP^-} \cdot |u^k(P) - u^k(P^-)|$$

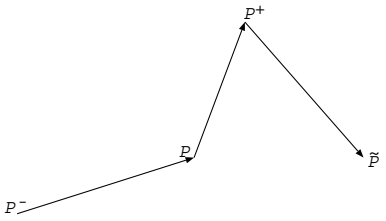


Figure 9: Harten lemma configuration

$$
\begin{aligned}
&+\sum_{\text{all } P} D^k_{\tilde{P}P^+}\cdot|u^k(\tilde{P})-u^k(P^+)|\\
&=\sum_{\text{all } P}\{(1-C^k_{P^+P}-D^k_{P^+P})|u^k(P^+)\\
&-u^k(P)|+C^k_{P^+P}|u^k(P^+)-u^k(P)|\\
&+D^k_{P^+P}\cdot|u^k(P^+)-u^k(P)|\}\\
&=\sum_{\text{all } P}|u^k(P^+)-u^k(P)|. \qquad (62)
\end{aligned}
$$

Hence $\mathrm{TV}(u^{k+1})\le \mathrm{TV}(u^k)$ that is the scheme (42) is a TVD scheme.

PROPERTY 2 *For the viscosity form (42) of scheme (29), if*

$$r^*|a^k_{P^+P}|\le Q^k_{P^+P}\le 1, \qquad (63)$$

then this scheme is a TVD scheme.

This is because the viscosity form (42) can be rewritten as

$$
\begin{aligned}
&u^{k+1}(P)=u^k(P)-\frac{1}{2}(Q^k_{PP^-}+r^*a^k_{PP^-})\\
&\cdot[u^k(P)-u(P^-)]\\
&+\frac{1}{2}(Q^k_{P^+P}-r^*a^k_{P^+P})\cdot[u^k(P^+)-u(P)] \qquad (64)
\end{aligned}
$$

From condition (63) and the Harten lemma it follows that

$$Q^k_{P^+P}\pm r^*a^k_{P^+P}\ge 0, \qquad (65)$$

$$
\begin{aligned}
&\frac{1}{2}(Q^k_{P^+P}+r^*a^k_{P^+P})\\
&+\frac{1}{2}(Q^k_{P^+P}-r^*a^k_{P^+P})\le 1 \qquad (66)
\end{aligned}
$$

for all P, that is, this scheme is a TVD scheme.

Under the condition $r^*|a^k_{P^+P}|\le 1$ and according to (63) it follows that the meshless explicit upwind scheme is a TVD scheme since $Q^k_{P^+P}=r^*|a^k_{P^+P}|\le 1$. The meshless Lax-Friedrichs scheme is also a TVD scheme since $Q^k_{P^+P}=1$. But the meshless Lax-Wendroff scheme is not a TVD scheme since $Q^k_{P^+P}=(r^*a^k_{P^+P})^2$ does not satisfy the condition (63). The above schemes usually are of order one because of the precision limitation of the node stencils.

5 Meshless ENO schemes

In order to improve the precision of TVD schemes, we must extend the node stencils using a meshless essential nonoscillatory (ENO) scheme. Previous work in this area can be found in [1, 3].

5.1 Successive extensions of a node stencil

Let $P^+_{1/2}$ be the middle point of the arc $\overline{P^+P}$ and $P^-_{1/2}$ the middle point of the arc $\overline{PP^-}$, see Figure 10. The equation which we have to approximate is written as

$$\frac{\partial u}{\partial t}=A\frac{\partial f}{\partial \vec{l_1}}+B\frac{\partial f}{\partial \vec{l_2}} \qquad (67)$$

where $\vec{l_1}=\overrightarrow{PP^+}$, $\vec{l_2}=\overrightarrow{P^-P}$.

First the construction of an integral averaging scheme is considered. The above equation will be integrated along the path $\Gamma=\overrightarrow{P^-_{1/2}P^+_{1/2}}$ and noting that $|\Gamma|=\frac{1}{2}(|\overrightarrow{l_1}|+|\overrightarrow{l_2}|)$ we have

$$
\begin{aligned}
&\int_\Gamma\frac{\partial u}{\partial t}ds=-\int_\Gamma\left(A\frac{\partial f}{\partial \vec{l_1}}+B\frac{\partial f}{\partial \vec{l_2}}\right)ds\\
&=-A[f(u(P^+_{1/2}))-f(u(P))]\\
&-B[f(u(P))-f(u(P^-_{1/2}))]. \qquad (68)
\end{aligned}
$$

Defining the path average of $u(x,t)$ as

$$\overline{u_P}=\frac{1}{|\Gamma|}\int_\Gamma u(s,t)ds \qquad (69)$$

we get

$$
\begin{aligned}
&\frac{d}{dt}\overline{u_P}+A\frac{f(u(P^+_{1/2},t))-f(u(P,t))}{|\Gamma|}\\
&+B\frac{f(u(P),t)-f(u(P^-_{1/2},t))}{|\Gamma|}=0. \qquad (70)
\end{aligned}
$$

In the following we will discuss the approximations of higher order to $u(P^+_{1/2})$ and $u(P^-_{1/2})$. In order to avoid discontinuities when extending the stencils we employ directional divided difference quotients to judge the smoothness degree of u within the stencils.

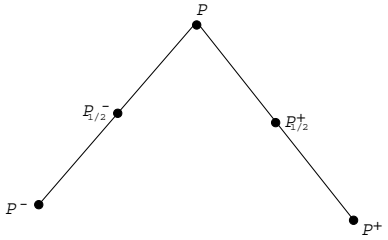


Figure 10: Extending the node stencil

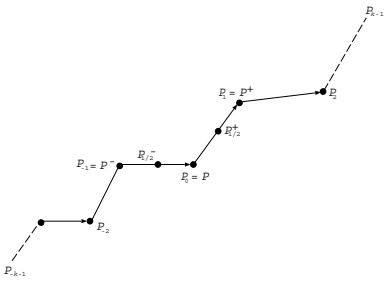


Figure 11: Path through P

Choose a path passing through P. The number of nodes in a local disc with center P is $2k-1$: $\{P_{-k+1},\ldots,P_{-1},P_0,P_1,\ldots,P_{k-1}$ (see Figure 11). The center of $P_{j-1}P_k$ is denoted by $P_{j/2}$. Let

$$\overline{u}[P_0]=\overline{u_{P_0}}. \tag{71}$$

The directional quotient of first order is

$$\overline{u}[P_0,P_1]=\frac{\overline{u}[P_1]-\overline{u}[P_0]}{|P_1P_0|} \tag{72}$$

and the directional quotient of order $k-1$ is

$$\overline{u}[P_0,\ldots,P_{k-1}]=\frac{\overline{u}[P_{k-1},\ldots,P_1]-\overline{u}[P_{k-2},\ldots,P_0]}{|P_{k-1}P_0|} \tag{73}$$

and we note the close connection with divided differences in this development. The arc lengths $|P_jP_0|(j=2,\ldots,k-1)$ can be calculated recursively from the arcs $|P_1P_0|,|P_2P_1|,\ldots,|P_jP_{j-1}|$, that is

$$\begin{aligned}&|P_2P_0|=\\&\sqrt{|P_1P_0|^2+|P_2P_1|^2-2|P_1P_0||P_2P_1|\cdot\cos\theta_1},\\&|P_3P_0|=\\&\sqrt{|P_2P_0|^2+|P_3P_2|^2-2|P_2P_0||P_3P_2|\cdot\cos\theta_2}\\&\ldots\end{aligned}$$

(see Figure 12).

A stencil will be adaptively extended by comparing the absolute values of the directional quotients. At first, the initial stencil $S_1=\{P_0\}$ is determined. To add a node, we have two choices

$$S_2=\begin{cases}S_1\cup\{P_{-1}\},\\ \text{or,}\\ S_1\cup\{P_1\}\end{cases} \tag{74}$$

that is, it will be decided whether to add a node to the left or to the right. We make our choice depending on the absolute values of directional quotients. If

$$|\overline{u}[P_0,P_{-1}]\le|\overline{u}[P_0,P_1]|$$

then we add the node P_{-1}, that is, $S_2=S_1\cup\{P_{-1}\}$. Otherwise we add the node P_1, that is, $S_2=S_1\cup\{P_1\}$. The remainder follows by analogy. We prescribe that the nodes passing through from P to P_j have all been used if a node P_j is picked up on this path except for P. Thus, a stencil can finally be determined on the path $P_{-k+1}\to\cdots\to P_0\to\cdots\to P_{k-1}$. Based on this stencil, we can establish a Newton type interpolation formula with up to degree $k-1$ on the path $P^-_{1/2}\to P\to P^+_{1/2}$:

$$\begin{aligned}&\overline{u}(Z)=\overline{u}(P_0)+\left(\frac{\overline{u}(Z)-\overline{u}(P_0)}{|ZP_0|}\right)\cdot|ZP_0|\\&=\overline{u}(P_0)+\overline{u}[Z,P_0]\cdot|ZP_0|\\&=\overline{u}(P_0)\\&+\left(\overline{u}[P_1,P_0]+\frac{\overline{u}[Z,P_0]-\overline{u}[P_1,P_0]}{|ZP_1|}\cdot|ZP_1|\right)\cdot|ZP_0|\\&=\overline{u}(P_0)+\overline{u}[P_1,P_0]\cdot|ZP_0|\\&+\overline{u}[Z,P_1,P_0]\cdot|ZP_1|\cdot|ZP_0|\\&=\ldots\\&=\sum_{j=0}^{k-1}\overline{u}[P_j,\ldots,P_0]\cdot\Pi_{i=0}^{j-1}|ZP_i|\\&+\overline{u}[Z,P_{k-1},\ldots,P_0]\cdot\Pi_{i=0}^{k-1}|ZP_i|\\&\triangleq\Phi_P(Z)+O(L^k)\end{aligned} \tag{75}$$

where $\triangleq$ means that the quantity on the right is defined by the quantities on the left. Further we get the

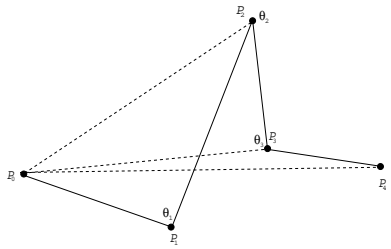


Figure 12: Recursive calculation of difference quotient

approximating values on the both sides of the path as $\overline{u}^-_{P^+_{1/2}} = \Phi_P(P^+_{1/2})$ and $\overline{u}^+_{P^-_{1/2}} = \Phi_P(P^-_{1/2})$ and we have

$$\overline{u}^-_{P^+_{1/2}} = \Phi_P(P^+_{1/2}) = \overline{u}(P^+_{1/2}, t) = O(L^k), \quad (76)$$

$$\overline{u}^+_{P^-_{1/2}} = \Phi_P(P^-_{1/2}) = \overline{u}(P^-_{1/2}, t) = O(L^k). \quad (77)$$

Hence we obtain a semi-discrete scheme

$$\frac{d\overline{u}_P}{dt} + A\frac{\hat{f}(\overline{u}^-_{P^+_{1/2}}, \overline{u}^+_{P^+_{1/2}}) - f(\overline{u}(P,t))}{|\Gamma|} + B\frac{f(\overline{u}(P,t)) - \hat{f}(\overline{u}^-_{P^-_{1/2}}, \overline{u}^+_{P^-_{1/2}})}{|\Gamma|} = 0 \quad (78)$$

where $\hat{f}(u^-, u^+)$ is the numerical flux of ENO methods, which satisfy the following properties:

(i) (consistency) $\hat{f}(u, u) = f(u)$

(ii) (monotonicity) $\hat{f}(\uparrow, \downarrow)$ which is not increasing for the first component and not decreasing for the second component

We can still construct the immediate difference-like scheme on each node.

The value of a node is noted as $u_P = u(P,t)$, and $\hat{f}_{P^+_{1/2}}$ is the numerical flux. In order to construct a k'th order scheme, that is

$$A\frac{\hat{f}_{P^+_{1/2}} - f(u)}{|\Gamma|} + B\frac{f(u) - \hat{f}_{P^-_{1/2}}}{|\Gamma|} = \left(A\frac{\partial f}{\partial \vec{l}_1} + B\frac{\partial f}{\partial \vec{l}_2}\right)_P + O(L^k) \quad (79)$$

we require an interpolating function $\Psi(Z)$ such that

$$\frac{1}{|\Gamma|}\int_{\overline{PP^+}} A\Psi(Z)dZ + \frac{1}{|\Gamma|}\int_{\overline{P^-P}} B\Psi(Z)dZ = f(u(P,t)) + O(L^k) \quad (80)$$

where $\Psi(Z)$ satisfies the following integral averaging interpolation condition

$$\frac{1}{|\Gamma|}\int_{\overline{PP^+}} A\Psi(Z)dZ + \frac{1}{|\Gamma|}\int_{\overline{P^-P}} B\Psi(Z)dZ = f(u_P). \quad (81)$$

Taking $\hat{f}_{P^+_{1/2}} = \Psi(P^+_{1/2})$ we have

$$\left(A\frac{\partial f}{\partial \vec{l}_1} + B\frac{\partial f}{\partial \vec{l}_2}\right)_P = A\frac{\Psi(P^+_{1/2}) - \Psi(P)}{|\Gamma|} + B\frac{\Psi(P) - \Psi(P^-_{1/2})}{|\Gamma|} + O(L^k). \quad (82)$$

Hence we get a semi-discrete scheme

$$\frac{du_P}{dt} + A\frac{\Psi(P^+_{1/2}) - \Psi(P)}{|\Gamma|} + B\frac{\Psi(P) - \Psi(P^-_{1/2})}{|\Gamma|} = 0. \quad (83)$$

5.2 Linear combination ENO methods

The precision of the approximation in a smooth region can be increased by forming linear combinations of stencils. Suppose that there are k stencils

$$S_r(P) = \{P_{-r}, \ldots, P_0, \ldots, P_{k-1-r}\}, \quad r = 0, \ldots k-1$$

for a given difference.

From the above Newton interpolation, we can find k distinct reconstructions $\overline{u}^{(r)}_{P^+_{1/2}}, r = 0, \ldots, k-1$ of $\overline{u_{P^+_{1/2}}}$ on each stencil. The LCENO (Linear Combination of ENO) scheme approximates $\overline{u}(P^+_{1/2}, t)$ using linear combinations of all $\overline{u}^{(r)}_{P^+_{1/2}}$:

$$\overline{u}_{P^+_{1/2}} = \sum_{r=0}^{k-1} w_r \overline{u}^{(r)}_{P^+_{1/2}} \quad (84)$$

where it is required that $\sum_{r=0}^{k-1} w_r = 1$ in order to satisfy the consistency condition.

Moreover, if u is smooth on all the stencils then we can find d_r such that

$$\sum_{r=0}^{k-1} d_r \overline{u}^{(r)}_{P^+_{1/2}} = \overline{u}(P^+_{1/2}, t) + O(L^{2k-1}) \quad (85)$$

and we still have $\sum_{r=0}^{k-1} d_r = 1$ because of consistency.

In the smooth case, we require that $w_r = d_r + O(L^{k-1}), r = 0, \ldots, k-1$ and we get

$$\sum_{r=0}^{k-1} w_r \overline{u}^{(r)}_{P^+_{1/2}} - \sum_{r=0}^{k-1} d_r \overline{u}^{(r)}_{P^+_{1/2}}$$

$$= \sum_{r=0}^{k-1}(w_r - d_r)(\overline{u}^{(r)}_{P^+_{1/2}} - \overline{u}(P^+_{1/2}, t))$$
$$\sum_{r=0}^{k-1} O(L^{k-1}) \cdot O(L^k) = O(L^{2k-1}). \quad (86)$$

The method therefore has order $2k-1$ precision, that is

$$\overline{u}_{P^+_{1/2}} = \sum_{r=0}^{k-1} w_r \overline{u}^{(r)}_{P^+_{1/2}} = \overline{u}(P^+_{1/2}, t) + O(L^{2k-1}). \quad (87)$$

If there is a discontinuity in a LCENO scheme for a given stencil, then w_r should be taken as zero. In practical computations, $\{w_r\}$ may be chosen as follows

$$w_r = \frac{\alpha_r}{\sum_{s=0}^{k-1}\alpha_s}, \quad r = 0, \ldots k-1, \quad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2} \quad (88)$$

where $\beta_r = O(L^{k-1})$ and $\epsilon > 0$ is introduced to avoid a vanishing denominator. Hence we have

$$w_r = \frac{d_r}{\sum_{s=0}^{k-1} d_s \left(\frac{\epsilon+\beta_r}{\epsilon+\beta_s}\right)^2}$$
$$= \frac{d_r}{\sum_{s=0}^{k-1} d_s \left(1 - \frac{\beta_s - \beta_r}{\epsilon+\beta_s}\right)^2} = \frac{d_r}{1 - O(\beta_s - \beta_s)}$$
$$= d_r + O(\beta_s - \beta_r) \Rightarrow \beta_s - \beta_r = O(L^{k-1}). \quad (89)$$

6 Runge-Kutta like time discretization

In the above discussion, we obtained a semi-discrete scheme $u_t = L(u)$, where $L(u)$ is an approximation for

$$-A\frac{\partial f}{\partial \vec{l_1}} - B\frac{\partial f}{\partial \vec{l_2}}.$$

In the following we will further consider the time discretization so as to obtain a full-discrete scheme.

If the total variation norm is defined as $\|\cdot\| = TV(\cdot)$ then the Euler forward difference method

$$u^{k+1} = u^k + \Delta t \cdot L(u^k) \quad (90)$$

is TVD stable if the time step satisfies $\delta t \le \Delta t_0$ and we have

$$\|(I + \Delta t L)(u^k)\| \le \|u^k\|. \quad (91)$$

The general TVD Runge-Kutta time discretization scheme is

$$u^{(i)} = \sum_{s=0}^{i-1}(\alpha_{is} u^{(s)} + \Delta t \cdot \beta_{is} \cdot L(u^{(s)})), i = 1, \ldots, m, \quad (92)$$
$$u^{(0)} = u^k, \quad u^{(m)} = u^{k+1}. \quad (93)$$

Assume that the coefficients α_{is} and β_{is} satisfy $\alpha_{is} \ge 0$, $\beta_{is} \ge 0$, $\sum_{s=0}^{i-1}\alpha_{is} = 1$. If the time step is chosen as

$$\Delta t \le c\Delta t_0 \text{ and } c = \min_{i,s}\frac{\alpha_{is}}{\beta_{is}} \quad (94)$$

then the scheme is a TVD scheme.

There is no TVD Runge-Kutta scheme of fourth order such that $\alpha_{is} \ge 0$, $\beta_{is} \ge 0$. Therefore the conjugate operator $\tilde{L}$ of L is defined in order to introduce the Euler backward difference method:

$$u^{k+1} = u^k - \Delta t \cdot \tilde{L}(u^k). \quad (95)$$

For instance, for the equation

$$u_t + A\frac{\partial f}{\partial \vec{l_1}} + B\frac{\partial f}{\partial \vec{l_2}} = 0 \quad (96)$$

we can define

$$L(u) = -\frac{f(u(P)) - f(u(P^-))}{\frac{1}{2}(|P^+P| + |PP^-|)},$$
$$\tilde{L}(u) = -\frac{f(u(P^+)) - f(u(P))}{\frac{1}{2}(|P^+P| + |PP^-|)}. \quad (97)$$

Thus

1. The computations of quantities needed for L and $\tilde{L}$ are the same.
2. L is stable for the Euler forward difference.
3. $\tilde{L}$ is stable for the Euler backward difference.

If $\alpha_{is} \ge 0$, β_{is} is negative and the operator $\tilde{L}$ is used instead of L, then the scheme (99) is a TVD scheme under the following restriction of the time step

$$\Delta t \le c\Delta t_0 \text{ and } c = \min_{i,s}\frac{\alpha_{is}}{|\beta_{is}|}. \quad (98)$$

6.1 Linear multistep TVD schemes

The general linear m step scheme is

$$u^{k+1} = \sum_{i=0}^{m-1}(\alpha_i u^{k-i} + \Delta t \cdot \beta_i \cdot L(u^{k-i})). \quad (99)$$

Let the coefficients α_{is} and β_{is} satisfy $\alpha_i \ge 0$, $\beta_i \ge 0$ and $\sum_{i=0}^{m-1}\alpha_i = 1$. If the time step is chosen as

$$\Delta t \le c\Delta t_0 \text{ and } c = \min_i \frac{\alpha_i}{\beta_i} \quad (100)$$

then the scheme is a TVD scheme.

There is still no TVD linear multistep scheme of fourth order such that $\alpha_i \ge 0$ and $\beta_i \ge 0$. Therefore the conjugate operator $\tilde{L}$ is used.

If $\alpha_i \geq 0$ and $\beta_i < 0$ and the conjugate operator $\tilde{L}$ is used instead of operator L, then the scheme (98) is a TVD scheme under the following restriction of time step

$$\Delta t \leq c\Delta t_0 \text{ where } c = \min_i \frac{\alpha_i}{|\beta_i|}. \tag{101}$$

7 Splitting, compounding and coupling break

The above only considers the construction of a scheme on a single path, for approximating the equation

$$u_t + A\frac{\partial f}{\partial \vec{l_1}} + B\frac{\partial f}{\partial \vec{l_2}} = 0. \tag{102}$$

In order to obtain an approximation to the original equation

$$u_t + A\frac{\partial f}{\partial x} + B\frac{\partial f}{\partial y} = 0 \tag{103}$$

we must employ a two path construction as shown in Figure 13. This results in

$$\begin{aligned}
&a\left(A_1\frac{\partial f}{\partial \vec{l}_1} + B_1\frac{\partial f}{\partial \vec{l}_2}\right) + b\left(A_2\frac{\partial f}{\partial \vec{l}_3} + B_2\frac{\partial f}{\partial \vec{l}_4}\right)\\
&+c\left(A_3\frac{\partial g}{\partial \vec{l}_1} + B_3\frac{\partial g}{\partial \vec{l}_2}\right) + d\left(A_4\frac{\partial g}{\partial \vec{l}_3} + B_4\frac{\partial g}{\partial \vec{l}_4}\right)\\
&= a\left[A_1\left(\cos\alpha_1\frac{\partial f}{\partial x} + \cos\beta_1\frac{\partial f}{\partial y}\right)\right.\\
&\qquad \left. +B_1\left(\cos\alpha_2\frac{\partial f}{\partial x} + \cos\beta_2\frac{\partial f}{\partial y}\right)\right]\\
&+b\left[A_2\left(\cos\alpha_3\frac{\partial f}{\partial x} + \cos\beta_3\frac{\partial f}{\partial y}\right)\right.\\
&\qquad \left. +B_2\left(\cos\alpha_4\frac{\partial f}{\partial x} + \cos\beta_4\frac{\partial f}{\partial y}\right)\right]\\
&+c\left[A_3\left(\cos\alpha_1\frac{\partial g}{\partial x} + \cos\beta_1\frac{\partial g}{\partial y}\right)\right.\\
&\qquad \left. +B_3\left(\cos\alpha_2\frac{\partial g}{\partial x} + \cos\beta_2\frac{\partial g}{\partial y}\right)\right]\\
&+d\left[A_4\left(\cos\alpha_3\frac{\partial g}{\partial x} + \cos\beta_3\frac{\partial g}{\partial y}\right)\right.\\
&\qquad \left. +B_4\left(\cos\alpha_4\frac{\partial g}{\partial x} + \cos\beta_4\frac{\partial g}{\partial y}\right)\right]\\
&= [(A_1\cos\alpha_1 + B_1\cos\alpha_2)a\\
&+ \qquad (A_2\cos\alpha_3 + B_2\cos\alpha_4)b]\frac{\partial f}{\partial x}\\
&+[(A_1\cos\beta_1 + B_1\cos\beta_2)a\\
&+ \qquad (A_2\cos\beta_3 + B_2\cos\beta_4)b]\frac{\partial f}{\partial y}\\
&+[(A_3\cos\alpha_1 + B_3\cos\alpha_2)c\\
&+ \qquad (A_4\cos\alpha_3 + B_4\cos\alpha_4)d]\frac{\partial g}{\partial x}\\
&+[(A_3\cos\beta_1 + B_3\cos\beta_2)c\\
&+ \qquad (A_4\cos\beta_3 + B_4\cos\beta_4)d]\frac{\partial g}{\partial y}.
\end{aligned} \tag{104}$$

Solving the equations

$$\begin{aligned}
&(A_1\cos\alpha_1 + B_1\cos\alpha_2)a + (A_2\cos\alpha_3 + B_2\cos\alpha_4)b = 1,\\
&(A_1\cos\beta_1 + B_1\cos\beta_2)a + (A_2\cos\beta_3 + B_2\cos\beta_4)b = 0,\\
&(A_3\cos\alpha_1 + B_3\cos\alpha_2)c + (A_4\cos\alpha_3 + B_4\cos\alpha_4)d = 0,\\
&(A_3\cos\beta_1 + B_3\cos\beta_2)c + (A_4\cos\beta_3 + B_4\cos\beta_4)d = 1
\end{aligned} \tag{105}$$

we get

$$\begin{aligned}
&u_t + a\left(A_1\frac{\partial f}{\partial \vec{l}_1} + B_1\frac{\partial f}{\partial \vec{l}_2}\right) + b\left(A_2\frac{\partial f}{\partial \vec{l}_3} + B_2\frac{\partial f}{\partial \vec{l}_4}\right)\\
&+c\left(A_3\frac{\partial g}{\partial \vec{l}_1} + B_3\frac{\partial g}{\partial \vec{l}_2}\right) + a\left(A_4\frac{\partial g}{\partial \vec{l}_3} + B_4\frac{\partial g}{\partial \vec{l}_4}\right)\\
&= u_t + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0.
\end{aligned} \tag{106}$$

Finally we will discuss the case of the system of convection equations, that is, the system of 2D conservation laws

$$U_t + f(U)_x + g(U)_y = 0 \tag{107}$$

where with $u = (u_1, \cdots, u_m)^T$

$$\begin{aligned}
f(U) &= \begin{pmatrix} f_1(u_1, \cdots, u_m)\\ \vdots\\ f_m(u_1, \cdots, u_m)\end{pmatrix},\\
g(U) &= \begin{pmatrix} g_1(u_1, \cdots, u_m)\\ \vdots\\ g_m(u_1, \cdots, u_m)\end{pmatrix}.
\end{aligned} \tag{108}$$

In the following we consider two approaches to such equations.

(1): Splitting-coupling break

According to the above analysis, we have to solve the system of equations

$$\begin{aligned}
&U_t + A_1^*\frac{\partial f(U)}{\partial \vec{l}_1} + A_2^*\frac{\partial f(U)}{\partial \vec{l}_2} + A_3^*\frac{\partial f(U)}{\partial \vec{l}_3}\\
&+A_4^*\frac{\partial f(U)}{\partial \vec{l}_4} + B_1^*\frac{\partial g(U)}{\partial \vec{l}_1} + B_2^*\frac{\partial g(U)}{\partial \vec{l}_2}\\
&+B_3^*\frac{\partial g(U)}{\partial \vec{l}_3} + B_4^*\frac{\partial g(U)}{\partial \vec{l}_4} = 0
\end{aligned} \tag{109}$$

at P.

We may employ a quarter time step Δt to obtain

$$\frac{1}{4}U_t + A_{i+1}^*\frac{\partial f(U)}{\partial \vec{l}_{i+1}} + A_{i+2}^*\frac{\partial f(U)}{\partial \vec{l}_{i+2}} = 0,$$

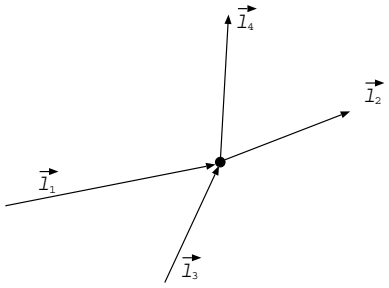


Figure 13: Combination of two paths

$$\frac{1}{4}U_t + B^*_{i+1}\frac{\partial g(U)}{\partial \overrightarrow{l_{i+1}}} + B^*_{i+2}\frac{\partial g(U)}{\partial \overrightarrow{l_{i+2}}} = 0, \tag{110}$$

$$i = 0, 2. \tag{111}$$

This can be solved using the method for scalar equations. Additionally, since

$$\begin{aligned}
&A^*_{i+1}\frac{\partial f(U)}{\partial \overrightarrow{l_{i+1}}} + A^*_{i+2}\frac{\partial f(U)}{\partial \overrightarrow{l_{i+2}}} = \\
&A^*_{i+1}\left(\cos\alpha_{i+1}\frac{\partial f}{\partial x} + \cos\beta_{i+1}\frac{\partial f}{\partial y}\right) \\
&+A^*_{i+2}\left(\cos\alpha_{i+2}\frac{\partial f}{\partial x} + \cos\beta_{i+2}\frac{\partial f}{\partial y}\right) \\
&= (A^*_{i+1}\cos\alpha_{i+1} + A^*_{i+2}\cos\alpha_{i+2})\frac{\partial f}{\partial x} \\
&+(A^*_{i+1}\cos\beta_{i+1} + A^*_{i+2}\cos\beta_{i+2})\frac{\partial f}{\partial y} \\
&= A^*_i\frac{\partial f(U)}{\partial \overrightarrow{l^A_i}}
\end{aligned} \tag{112}$$

where

$$\begin{aligned}
\overline{A^*_i} &= (A^*_{i+1}\cos\alpha_{i+1} + A^*_{i+2}\cos\alpha_{i+2})^2 \\
&+(A^*_{i+1}\cos\beta_{i+1} + A^*_{i+2}\cos\beta_{i+2})^2
\end{aligned}$$

and

$$\begin{aligned}
A^*_i &= \sqrt{\overline{A^*_i}}, \\
\vec{l^A_i} &= ((A^*_{i+1}\cos\alpha_{i+1} + A^*_{i+2}\cos\alpha_{i+2})/A^*_i, \\
&+(A^*_{i+1}\cos\beta_{i+1} + A^*_{i+2}\cos\beta_{i+2})/A^*_i).
\end{aligned} \tag{113}$$

and hence we can write (111) as

$$\begin{aligned}
\frac{1}{4}U_t + A^*_i\frac{\partial f(U)}{\partial \overrightarrow{l^A_i}} &= 0, \\
\frac{1}{4}U_t + B^*_i\frac{\partial g(U)}{\partial \overrightarrow{l^B_i}} &= 0
\end{aligned} \tag{114}$$

or

$$\frac{1}{4}U_t + A^*_i\frac{\partial f(U)}{\partial U}\cdot\frac{\partial U}{\partial \overrightarrow{l^A_i}} = 0,$$

$$\frac{1}{4}U_t + B^*_i\frac{\partial g(U)}{\partial U}\cdot\frac{\partial U}{\partial \overrightarrow{l^B_i}} = 0 \tag{115}$$

where $A^*_i\frac{\partial f(U)}{\partial U}$, $B^*_i\frac{\partial g(U)}{\partial U}$ are computed by a local freezing approach. For instance, the value of f' at P can be chosen as the arithmetic average of two points on the arc $\overline{P^-P^+}$ as $A^*_i f'(\frac{1}{2}(U(P^-_{1/2}) + U(P^+_{1/2})))$, or the Roe average $A^*_i f'(\frac{1}{2}(U^{\mathrm{Roe}}(P^-_{1/2}) + U(P^+_{1/2})))$.

In the following we consider a particular system of equations. Let the matrix $A^*_i\frac{\partial f(U)}{\partial U}$ have m real eigenvalues

$$\lambda_1(U) \le \lambda_2(U) \le \cdots \le \lambda_m(U)$$

and a system of complete eigenvectors

$$r_1(U), r_2(U), \cdots, r_m(U).$$

Then $A^*_i\frac{\partial f(U)}{\partial U}$ can be diagonalized by a similarity transformation, that is,

$$R^{-1}(U)A^*_i\frac{\partial f(U)}{\partial U}R(U) = \Lambda(U) \tag{116}$$

where

$$\begin{aligned}
\Lambda(U) &= \mathrm{diag}(\lambda_1(U), \ldots, \lambda_m(U)), \\
R(U) &= (r_1(U), \ldots, r_m(U)).
\end{aligned} \tag{117}$$

Setting $V = R^{-1}U$, we obtain

$$\begin{aligned}
&\frac{1}{8}R^{-1}U_t + R^{-1}A^*\frac{\partial f(U)}{\partial U}R\cdot R^{-1}\frac{\partial U}{\partial \vec{l_i}} = 0, \\
&\frac{1}{8}V_t + \Lambda\frac{\partial U}{\partial \vec{l^A_i}} = 0.
\end{aligned} \tag{118}$$

Therefore the coupled equations can be broken into m independent equations

$$\left\{\begin{array}{c}
\frac{1}{8}(\nu_1)_t + \lambda_1\left(\frac{\partial \nu_1}{\partial \vec{l_i}}\right) = 0, \\
\vdots \\
\frac{1}{8}(\nu_m)_t + \lambda_m\left(\frac{\partial \nu_m}{\partial \vec{l_i}}\right) = 0
\end{array}\right. \tag{119}$$

which can also be solved using the method for scalar equation. After getting V we find U from $U = RV$.

(2): Compounding-Roe like coupling break

For $U_t + f(U)_x + g(U)_y = 0$, if both sides are integrated on the convex hull $N(P)$ of the node stencil involving in the difference scheme at P we get

$$\frac{d}{dt}\int_{N(P)} U d\Omega + \int_{N(P)} (f(U)_x + g(U)_y) d\Omega = \frac{d}{dt}\int_{N(P)} U d\Omega + \int_{\partial N(P)} (\vec{F}\cdot\vec{n}) ds = 0 \quad (120)$$

where $\partial N(P)$ is the boundary of $N(P)$, $\vec{n}$ is the outer unit normal at a boundary line of $N(P)$ and $\vec{F} = (f, g)$. This is indicated in Figure 14.

The semi-discrete form for this is

$$A_0 \frac{dU}{dt}\Big|_{N(P)} = -\sum_{l\in N(P)} (\vec{F}\cdot\vec{n})_{\vec{l}} \cdot \Delta l \quad (121)$$

where A_0 is the area of $N(P)$, which can be expressed as the linear combination of nodal values on $N(P)$. $\vec{n_l}$ is the outer unit normal of boundary line $\vec{l}$ and Δl is the length of $\vec{l}$.

The numerical flux can be written as

$$(\vec{F}\cdot\vec{n})_{\vec{l}} = \frac{1}{2}\{(\vec{F}\cdot\vec{n})(U^-_{\vec{l}}) + (\vec{F}\cdot\vec{n})(U^+_{\vec{l}}) - \left|\frac{\partial(\vec{F}\cdot\vec{n})}{\partial U}\right|_{\vec{l}} \cdot (U^+_{\vec{l}} - U^-_{\vec{l}}) \quad (122)$$

where $(U^-_{\vec{l}}, U^+_{\vec{l}})$ are the U values of at the sides of $\vec{l}$. As in (32), the simple arithmetic average or Roe average of $\frac{\partial(\vec{F}\cdot\vec{n})}{\partial U}\Big|_{\vec{l}}$ is diagonalized by a similarity transformation:

$$R^{-1}(U)\ \frac{\partial(\vec{F}\cdot\vec{n})}{\partial U}\Big|_{\vec{l}} R(U) = \Lambda(U). \quad (123)$$

Noting that $V^{\pm}_{\vec{l}} = R^{-1}U^{\pm}_{\vec{l}}$ we have

$$(\vec{F}\cdot\vec{n})_{\vec{l}} = \frac{1}{2}\{(\vec{F}\cdot\vec{n})(U^-_{\vec{l}}) + (\vec{F}\cdot\vec{n})(U^+_{\vec{l}}) - R(U)|\Lambda(U)|R^{-1}(U)(U^+_{\vec{l}} - U^-_{\vec{l}}) \quad (124)$$

$$= \frac{1}{2}\{(\vec{F}\cdot\vec{n})(RV^-_{\vec{l}}) + (\vec{F}\cdot\vec{n})(RV^+_{\vec{l}}) - (|\lambda_1|r_1, \cdots, |\lambda_m|r_m)\cdot(V^+_{\vec{l}} - V^-_{\vec{l}}). \quad (125)$$

Computational Example 1. Forward facing step problem.

In Example 2, almost 8000 nodes are assigned for computing, and in Example 3, also 8000 nodes are assigned, and 2D shallow water equation is used.

This is a standard test example for high resolution methods. The problem is as follows: In a windtunnel, 3 long and 1 wide a step is 0.2 high is located 0.6 from the left-hand end of the tunnel (see Figure 15).

We use an Euler system of equations of conservative type to describe the flow of fluids. Almost 12000 nodes are assigned on one time-layer for computing, and a improved Wendroff scheme with a splitting flux and adaptive strategy was implemented as discussed in previous sections.

The initialization is a Mach 3 flow from the right. Reflective bounds are set up along the walls of the tunnel, The in-flow and out-flow bounds are set up at the entrance (right-hand end) and the exit (left-hand end). In the Figures 15-17 we present the result of the improved meshless splitting Wendroff like method of the computation at times $t_a < t_b < t_c$.

Computational Example 2. Double Mach reflection.

The computational domain for this problem is taken to be $[0, 4] \times [0, 1]$. The reflecting wall lies on the bottom of the computational domain starting at 1/6 from the left-hand end (see also Figure 18). Approximately 8000 nodes are provided initially.

The initialization is a Mach 10 shock from the left positioned on the bottom starting at 1/6 from the left-hand end and has a 60^0 angle with the axis. For the bottom bound, the post-shock condition is imposed for the part from left-hand end to 1/6 and a reflective condition is for the remaining distance. For the top bound of the computational domain, the flow values are set up to describe the motion of the Mach 10 shock. In Figures 18 and 19 several results of our meshless difference method are presented.

Computational Example 3. Problem of dam failure with the 2D shallow water equation.

A 200×200 computational domain is considered where there is dam which divides water storage into two equal parts. The upper part is 80 depth, and lower part is 20 deep. A segment on the dam body suddenly breaks. This segment is 75 long and 95 away from one side of the water storage. The flow of water obey 2D shallow water equation. Approximately 8000 nodes are provided initially.

In Figures 20 and 21 several results of our meshless difference method are presented.

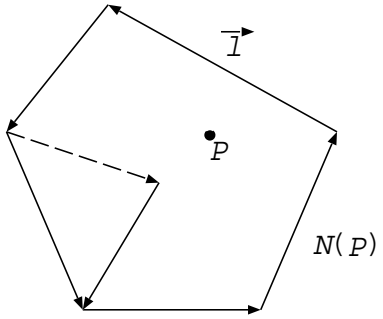


Figure 14: Convex hull of node stencil at P

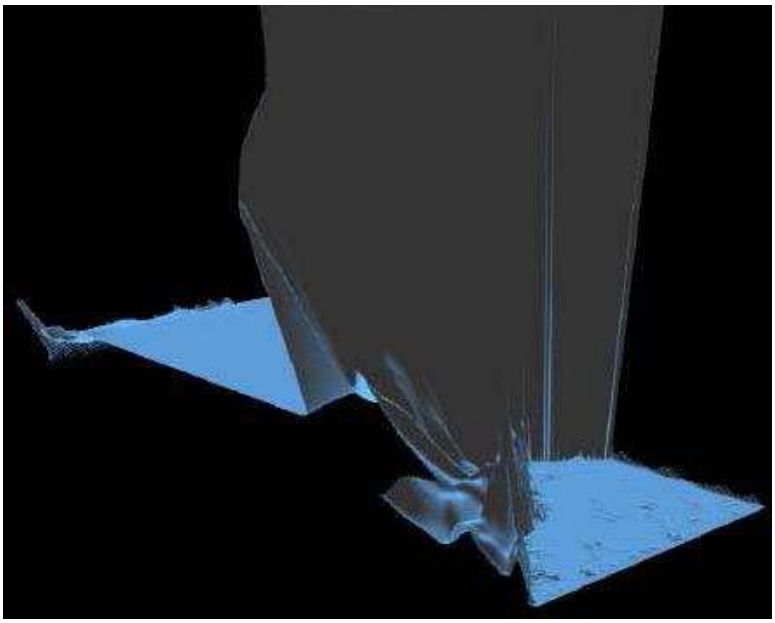


Figure 15: Forward facing step problem, a

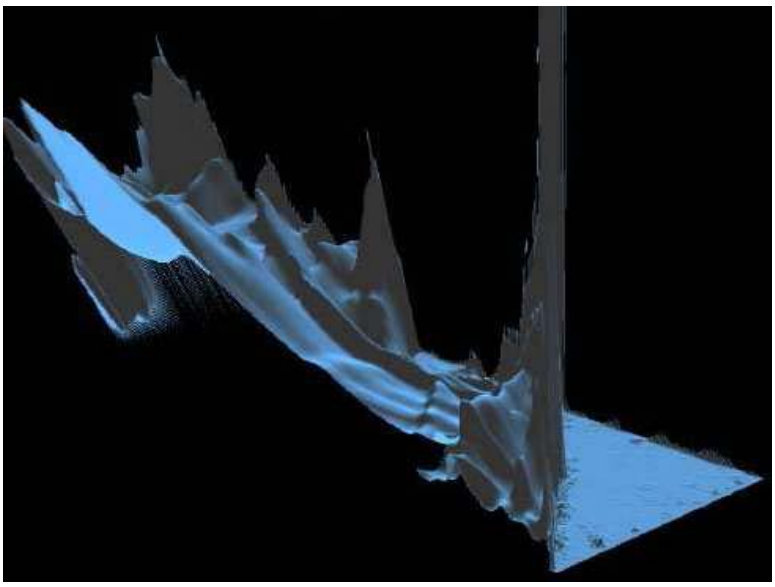


Figure 16: Forward facing step problem, b

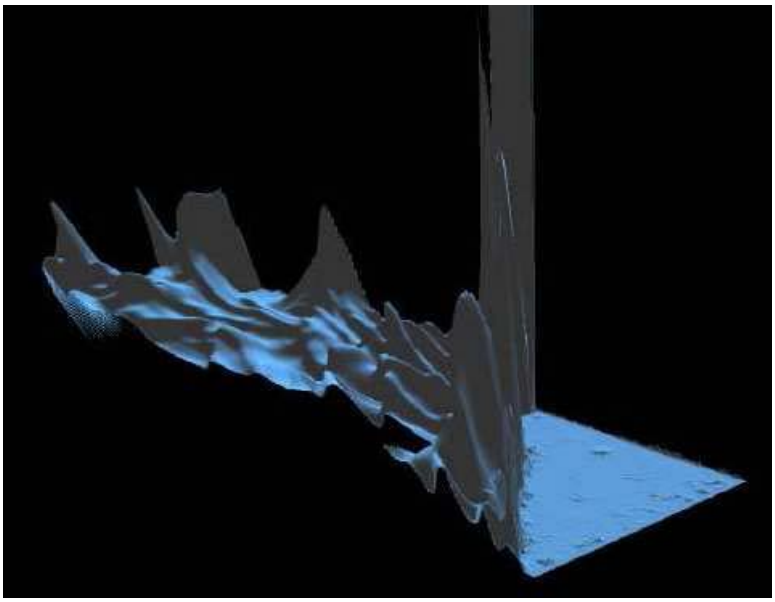


Figure 17: Forward facing step problem, c

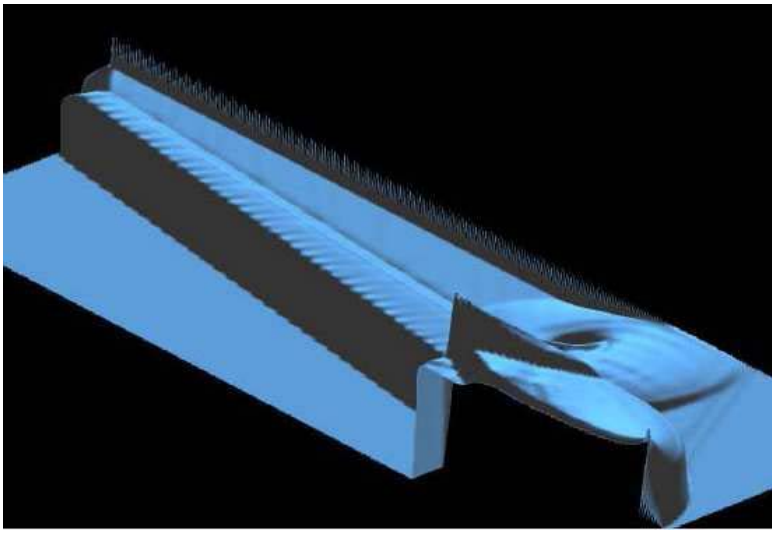


Figure 18: Double Mach reflection, a

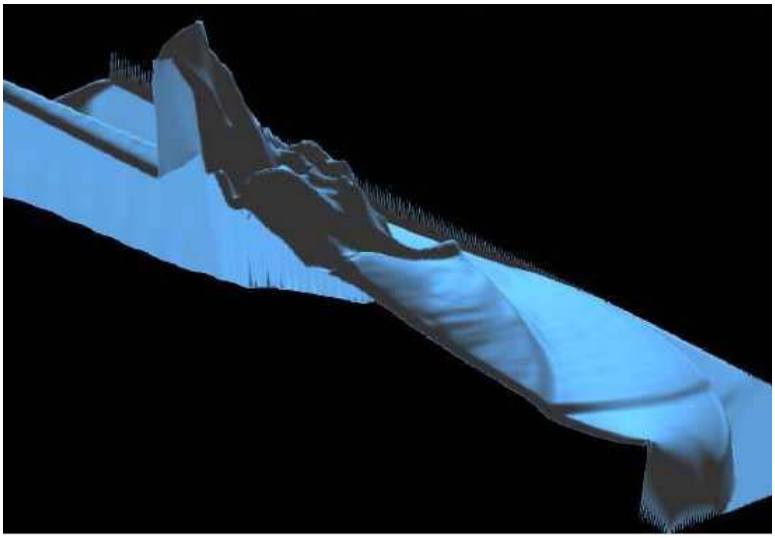


Figure 19: Double Mach reflection, b

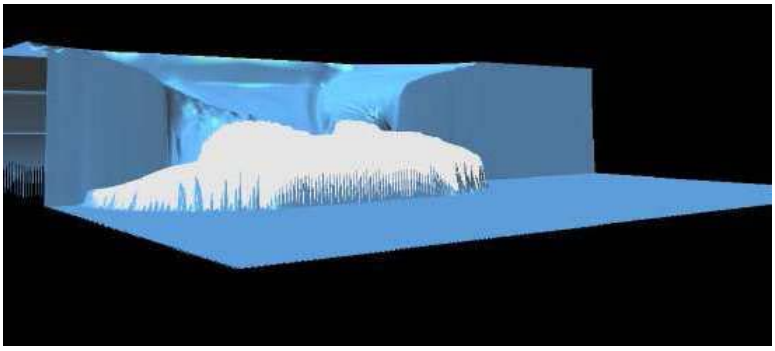


Figure 20: Dam failure, a

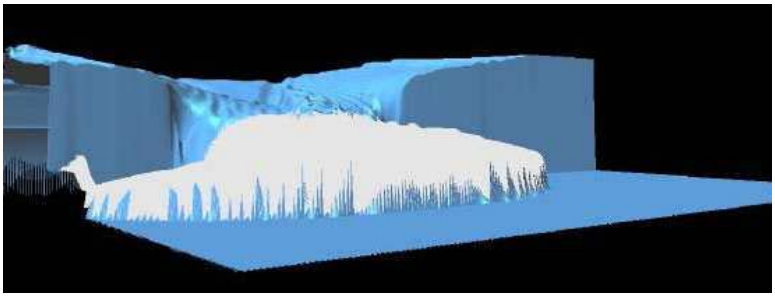


Figure 21: Dam failure, b

References

[1] Abgrall, R., "On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation," *J. Computational Physics*, V114, pp. 45-58, 1994.

[2] Azarenok, B. N., Ivanenko, S. A., Tang, T., "Adaptive mesh redistribution method based on Godunov's scheme," *Comm. Math. Sci.,* V1, pp. 152-179, 2003.

[3] Cecil, T., Quian, J.. Osher, O., "Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions," *J. Computational Physics,* V196, pp. 327-347, 2003.

[4] Devals, C. Heniche, M. Bertrand, F., Hayes, R. E., Tanguy, P. A., "A finite element strategy for the solution of interface tracking problems," *International Journal for Numerical Methods in Fluids,* V49, pp. 1305-1327, 2005.

[5] Gottlieb, S., Shu. C. W., "Total variation diminishing Runge-Kutta schemes," *Mat. Comp.,* V67, pp. 73-85, 1998.

[6] Emmrich, E., *Hyperbolische Erhaltungsgleichungen.* Ph. D. thesis, Magdeburg, 1993.

[7] Harten, A., "High resolution schemes for hyperbolic conservation laws," *Journal of Computational Physics,* V49, pp. 357-393, 1983.

[8] Liu, G. R., Liu, M. B., *Smoothed Particle Hydrodynamics. A Meshfree Particle Method.* World Scientific, Singapore, 2003.

[9] Monaghan, J. J., "Smoothed particle hydrodynamics," *Ann. Rev. Astron, Astrophys.,* V30, pp. 543-574, 1992.

[10] Lin, Q., Rokne, J., "Construction and analysis of meshless finite difference methods," *Computational Mechanics,* V37, pp. 232-248, 2006.

[11] Ransau, S. R., Haegland, B., Holmen, J., "A finite element implementation of the level set equation," *Preprint Numerics,* 4/2004, Norges Teknisk-Naturvitenskaplige Universitet. Downloaded from: http://www.math.ntnu.no/preprint /numerics/2004/N4-2004.pdf, October 12, 2006.

[12] Sarra, S. A., "Adaptive radial basis function methods for time dependent partial differential equations," *Applied Numerical Mathematics,* V54, pp. 79-94, 2005.

[13] Sethian, J. A., *Level Sets and Fast Marching Methods.* Cambridge University Press, 1999.

[14] Tang, T., "Moving mesh methods for computational fluid dynamics," In: *Recent Advances in Adaptive Computation* Eds.: Z. Shi, Z. Chen, T. Tang, and D. Yu, eds. *Contemporary Mathematics,* V383, American Mathematical Society, 2005, *Proceedings of the International Conference on Recent Advances in Adaptive Computation,* May 2004, Hangzhou, China, pp. 141-173, 2005.

[15] Tucker, P. G., "Transport equation based wall distance computations aimed at flows with time-dependent geometry," NASA/TM-2003-212680, NASA Center for AeroSpace Information, Downloaded from: http://library-dspace.larc.nasa.gov/dspace/jsp/ bitstream/2002/13787/1/NASA-2003-tm212680.pdf, October 12, 2006.

[16] Udaykumar, H. S., Mittal, R. Rampunggoon, P., Khanna, Q.: "A Sharp interface cartesian grid method for simulating flows with complex moving boundaries," *J. Computational Physics,* V174, pp. 345-380, 2001.

[17] Wu, N. J., Tsay, T. K., Young, D. L., "Meshless numerical simulation for fully nonlinear water waves," *International Journal for Numerical Methods in Fluids,* V50, pp. 219-234, 2006.