NUMERICAL ALGORITHMS BASED ON THE BACK AND FORTH ERROR COMPENSATION AND CORRECTION

A Thesis
Presented to
The Academic Faculty

by

Lili Hu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mathematics

Georgia Institute of Technology
May 2015

NUMERICAL ALGORITHMS BASED ON THE BACK AND FORTH ERROR COMPENSATION AND CORRECTION

Approved by:

Professor Yingjie Liu, Advisor
School of Mathematics
Georgia Institute of Technology

Professor Haomin Zhou
School of Mathematics
Georgia Institute of Technology

Professor Luca Dieci
School of Mathematics
Georgia Institute of Technology

Professor Jarek Rossignac
School of Interactive Computing
Georgia Institute of Technology

Professor Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Date Approved: 27 August 2014

To my husband,

Yao Li

ACKNOWLEDGEMENTS

Thank my advisor Prof. Yingjie Liu for his continuous and consistent support of my PhD study and research, for his patience, motivation, enthusiasm, immense knowledge and availability. Without him, I couldn't imagine myself finishing my PhD thesis and now defending it. I could have given up at some point while I was on leave without his remote guidance and encouragement.

Thank the rest of my thesis committee: Prof. Luca Dieci, Prof. Sung Ha Kang, Prof. Haomin Zhou and Prof. Jarek Rossignac for taking time reading my thesis, giving me their insightful comments and asking me hard questions.

Thank Prof. Chris Heil for being a wonderful mentor and friend and for introducing me to the world of PhD students studying at School of Math at Georgia Tech.

Thank Prof. Luca Dieci, the former graduate coordinator, for admitting me into the PhD program at School of Math and offering me assistantship.

Thank Prof. Haomin Zhou, Prof. Shui-Nee Chow and Prof. Christine Heitsch for once giving me an opportunity to work on a side project.

Thank Ms. Klara Grodzinsky and Ms. Cathy Jacobson for training me into an eligible teaching assistant and assisting me with my English.

Thank Prof. Igor Belegradek, Prof. Stavros Garoufalidis, Prof. Sung Ha kang, Prof. Michael lacey, Prof. Michael Loss, Prof. Brett Wick and etc, for teaching me graduate level math courses towards my PhD study.

Thank my peer graduate students who may have already graduated for helping me with miscellaneous things in work and in life. To name a few, Jun Lu, Tianjun Ye, Huijun Feng, Yan Shu, Jinyong Ma, Chenchen Mou, Suo Yang, Shannon Bishop, etc.

Thank all the staff and IT support members at School of Math including Ms. Joanne Cook, Ms. Christy Dalton, Mr. Justin Filoseta, Ms Celia Grams, Mr. Matt Hanes, Ms. Karen Hinds, Ms. Sharon McDowell, Ms. Annette Rohrs, Ms. Inetta Worthy, and etc.

Last but not the least, thank my husband Yao Li, the best thing that has ever happened in my life. Thank you for always being there for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this thesis we carry out a further study of the back and forth error compensation and correction (BFECC) method. The first part discusses the time reversibility of numerical schemes. Motivated by the BFECC method, a variety of new numerical schemes that aim at improving the time reversibility are developed and studied. We also introduce an interpolation algorithm based on BFECC in this part. In the second part we introduce a new limiting strategy which requires another backward advection in time so that overshoots/undershoots at the new time level get exposed when they are transformed back to compare with the solution at the old time level. This new technique is very simple to implement even for unstructured meshes and is able to eliminate artifacts induced by jump discontinuities in the solution itself or in its derivatives.

CHAPTER I

INTRODUCTION

The BFECC method was proposed in [10] as a convenient method to achieve better computational accuracy for the level set advection [25]. The idea is that when a solution is advected forward and then backward for a time step, the difference between two copies of the solution at the initial time level provides information about the numerical error of the underlying scheme. This information can be used to compensate the solution before a third advection forward in time, resulting in more accurate numerical solution at the next time level. For the linear advection equation on rectangular meshes, this procedure has been shown in [12] to improve the order of accuracy of an odd order scheme by one (in both space and time) and also stabilize the scheme if its amplification factor is less than 2 with some reasonable conditions.

The BFECC method coupled with an underlying semi-Lagrangian scheme has been studied for various fluid and level set interface advections, e.g. in [11, 16, 17, 18, 19], and for the Lattice Boltzmann method on quad-tree grids [5]. Without the CFL restriction, this combination is easy to implement on various meshes. A generalized MacCormack scheme without CFL restriction is developed and applied to fluid simulations in [27], in which the error information from the forward and backward advections is applied directly to the previously obtained solution at the next time level. These advection equations belong to the class of Hamilton-Jacobi equations. The numerical monotone Hamiltonian along with a high order ENO approach have been developed for Hamilton-Jacobi equations in [26]. See [1] for a Lax-Friedrichs-type numerical monotone Hamiltonian on 2D triangular meshes and [31] for incorporating

a high order WENO reconstruction. High order central schemes for solving Hamilton-Jacobi equations have been proposed in e.g. [9, 22, 4], and some recent development can be found in [20, 3, 21]. A conservative scheme developed for solving a conservation law typically use an r-th degree polynomial interpolation to achieve $(r+1)$-th formal order of accuracy. When the scheme is modified for solving a Hamilton-Jacobi equation, however, $(r+1)$-th degree polynomial interpolation is usually needed to achieve $(r+1)$-th formal order of accuracy. For example, a second order non-oscillatory scheme for solving a Hamilton-Jacobi equation may need a quadratic interpolation. On the other hand, when BFECC is applied to a first order scheme using only local linear interpolation, it can improve both its temporal and spatial order of accuracy to second order. This is very convenient for unstructured meshes since a linear interpolation only uses information from adjacent grid points.

In this thesis, we study a variety of numerical algorithms that adopt the fundamental idea of BFECC. We first focus on the time-reversibility feature of BFECC. In the implementation of BFECC, if there is no back-and-forth error, the underlying numerical scheme is called *time reversible*. We proved that time reversible schemes implies one more order of accuracy. Motivated by the concept of time reversibility, we introduce several new algorithms that attempt to construct time reversible schemes from known numerical schemes. These methods include linear combination of center scheme and Lax-Friedrichs scheme, high order correction using multiple times of BFECC and compositions of BFECC. Moreover, we adopt the idea of BFECC to develop an effective interpolation algorithm that is easy to implement on unstructured meshes. The main idea is to consider interpolation problems as hyperbolic equations with zero advection terms.

To reduce the artificial effects caused by numerical schemes themselves, a number of limiting strategies are proposed. In [12], a limiting technique is introduced which is based on using a locally constant advection velocity to compute the back-and-forth

error wherever a singularity in the velocity field is detected. This technique works only if the solution is at least Lipschitz continuous, such as a level set function. A simple limiting technique is used in [27] for the BFECC and the modified unconditional stable MacCormack scheme by essentially regulating the solution at a grid point within extrema on neighboring grid points. Another limiting technique for BFECC can be found in [13].

In the second part of this thesis, we will introduce a new limiting strategy for BFECC based on the following consideration. Assuming the solution U^n at the time t_n is accurate and we have computed the solution U^{n+1} at the time t_{n+1}, it's very difficult to detect whether and where the new solution U^{n+1} has overshoots/undershoots since we don't know the exact solution. However, if we can approximately advect U^{n+1} backward in time to the time level t_n then we have an accurate solution U^n to compare with. This idea works like an extension to the strategy used in BFECC. The effect of this limiting strategy is demonstrated in a number of linear and nonlinear problems. We will show that artificial effects such as overshoot and undershoot at the discontinuities of solutions can be successfully eliminated by BFECC limiting method.

CHAPTER II

BACK AND FORTH ERROR COMPENSATION AND CORRECTION (BFECC) AND TIME REVERSIBLE SCHEME

2.1 Preliminary: BFECC

Consider the convection equation on $\mathbb{R}^N$

$$\frac{\partial u}{dt} + \mathbf{a} \cdot \bigtriangledown u = 0. \tag{1}$$

Let $\mathcal{L}$ be a numerical scheme that updates the numerical solution from the time t_n to t_{n+1}, $t_n < t_{n+1}$. Let $\mathcal{L}^*$ be the numerical scheme that updates the numerical solution from the time t_{n+1} to t_n by applying $\mathcal{L}$ to the time-reversed equation of (1)

$$\frac{\partial u}{dt} - \mathbf{a} \cdot \bigtriangledown u = 0. \tag{2}$$

Let U^n be the numerical solution given at the time t_n, then the BFECC algorithm can be described as follows [10].

1. **Forward advection.**
   $\tilde{U}^{n+1} = \mathcal{L}U^n$.

2. **Backward advection.**
   $\tilde{U}^n = \mathcal{L}^*\tilde{U}^{n+1}$.

3. **Forward advection again using modified solution at the time t_n.**
   $U^{n+1} = \mathcal{L}(U^n + e^{(1)})$, where $e^{(1)} = \frac{1}{2}(U^n - \tilde{U}^n)$.

Here we call $e^{(1)}$ the back-and-forth error. Since the forward advection and backward advection use the same scheme (note that $\mathcal{L}^*$ is $\mathcal{L}$ applied to the time-reversed equation), we assume that they introduce similar amount of error. Therefore $e^{(1)} = \frac{1}{2}(U^n - \tilde{U}^n)$ provides an estimate of the error which is going to be added from the solution during the forward advection. Consider a rectangular mesh on $\mathbb{R}^N$ with the mesh size h and grid point $x_j = jh$ for any multi-index j. Let U_j^n be the numerical solution at x_j and time t_n, and let $k = t_{n+1} - t_n$ be the time step size. Assume $k = \theta h$ for some fixed constant θ during mesh refinement. Let $\mathbf{a}$ be a constant vector in $\mathbb{R}^N$.

We assume a linear scheme $\mathcal{L}$ can be written in the form of

$$U_i^{n+1} = \sum_{|j|\leq l} C_j U_{i+j}^n, \tag{3}$$

where l is a positive integer and C_j is a real constant depending only on $\mathbf{a}$, the multi-index j and the constant $\theta = k/h$. We will view scheme (3) as

$$U^{n+1}(x) = \sum_{|j|\leq l} C_j U^n(x + jh), \quad \text{for any } x \in \mathbb{R}^N, \tag{4}$$

for convenience in the following Fourier analysis, where $U^n(x)$ is a continuous function with compact support in $\mathbb{R}^N$.

Let $\rho_{\mathcal{L}}$ denote the Fourier symbol of the numerical scheme $\mathcal{L}$, and $\rho_{\mathcal{L}^*}$ denote the Fourier symbol of $\mathcal{L}^*$. For example, the operator $\mathcal{L}$ could be the upwind scheme, Lax-Friedrichs scheme, center difference scheme (unstable), CIR scheme [6] or some other schemes. In addition, we assume $\rho_{\mathcal{L}^*} = \overline{\rho_{\mathcal{L}}}$, where $\overline{\rho_{\mathcal{L}}}$ is the complex conjugate of $\rho_{\mathcal{L}}$. Note that this assumption is true for essentially all commonly used first order linear schemes [12].

It has been proved in [12] that the BFECC algorithm creates a stable scheme if the amplification factor $|\rho_{\mathcal{L}}|$ of $\mathcal{L}$ satisfies $|\rho_{\mathcal{L}}| \leq 2$. Furthermore, if the order of accuracy of the scheme $\mathcal{L}$ is r for some positive odd integer r, then the order of accuracy after applying BFECC is $r + 1$. For the sake of completeness, let me include this theorem from [12]. Also without loss of generality, let us consider only the one dimentional case of equation (1).

Let $\mathcal{L}$ and $\mathcal{L}^*$ be the linear schemes as above. Let $\mathcal{F}$ be the corresponding BFECC scheme. Let $\rho_{\mathcal{L}}$, $\rho_{\mathcal{L}^*}$ and $\rho_{\mathcal{F}}$ be their corresponding Fourier symbols respectively. Assume u in equation (1) is defined for all $x \in \mathcal{R}$ and $u(x, 0)$ is a periodic function with period 1. Expanding u into Fourier series

$$u(x, t) = \sum_{k\in Z} c_k(t) e^{2\pi i k x} \tag{5}$$

and plugging in equation (1), we obtain

$$\frac{dc_k}{dt} = p(ik)c_k \tag{6}$$

where $p(x) = -2a\pi x$ is a linear polynomial with real coefficient. Therefore we can write $c_k(t_n + \Delta t) = e^{\Delta t p(ik)} c_k(t_n)$

We first state the theorem of Lax [23].

Theorem 2.1.1. *Scheme $\mathcal{L}$ is accurate of order r if and only if $\rho_{\mathcal{L}}(k) = e^{\Delta t p(ik)} + \mathcal{O}(|kh|^{r+1})$, as $h \to 0$.*

Using theorem 2.1.1, the following theorem can be proved.

Theorem 2.1.2. *Suppose $\rho_{\mathcal{L}^*}(k) = \overline{\rho_{\mathcal{L}}(k)}$ for any $k \in \mathcal{Z}$ and that scheme $\mathcal{L}$ is accurate of order r for equation (1) with constant coefficients, where r is an odd positive integer. Then scheme $\mathcal{F}$ is accurate of order $r+1$.*

Proof. The accuracy of scheme $\mathcal{L}$ implies that

$$\rho_{\mathcal{L}}(k) = e^{\Delta t p(ik)} + q_{r+1}(ikh) + \mathcal{O}(|kh|^{r+2}), \tag{7}$$

where q_{r+1} is a polynomial of order $r+1$ with real coefficients . Since $r+1$ is even, we have

$$\rho_{\mathcal{L}^*}(k) = \overline{\rho_{\mathcal{L}}(k)} = e^{-\Delta t p(ik)} + q_{r+1}(ikh) + \mathcal{O}(|kh|^{r+2}) \tag{8}$$

Therefore

$$\begin{aligned}
\rho_{\mathcal{F}}(k) &= \rho_{\mathcal{L}}(k)(1 + \tfrac{1}{2}(1 - \rho_{\mathcal{L}^*}(k)\rho_{\mathcal{L}}(k))) \\
&= \rho_{\mathcal{L}}(k)(1 - \tfrac{1}{2}(e^{-\Delta t p(ik)} + e^{\Delta t p(ik)})q_{r+1}(ikh) + \mathcal{O}(|kh|^{r+2})) \\
&= (e^{\Delta t p(ik)} + q_{r+1}(ikh) + \mathcal{O}(|kh|^{r+2}))(1 - q_{r+1}(ikh) + \mathcal{O}(|kh|^{r+2})) \\
&= e^{\Delta t p(ik)} + \mathcal{O}(|kh|^{r+2})
\end{aligned} \tag{9}$$

The proof is complete. □

2.2 Time Reversible Scheme

In BFECC, the back-and-forth error is used to estimate the error introduced by the underlying scheme and is also used to correct the solution at the old time level t_n before the forward advection to the solution at the new time level t_{n+1}. This motivate us to consider the case where the back-and-forth error is zero. Take a look at $\tilde{U}^n = \mathcal{L}^*\mathcal{L}U^n$ in the BFECC algorithm. If $\tilde{U}^n = U^n$, we call $\mathcal{L}$ *time-reversible*. We have the following results for a time-reversible linear scheme.

Lemma 2.2.1. *A linear scheme $\mathcal{L}$ is time-reversible if and only if $|\rho_{\mathcal{L}}| = 1$.*

Proof. Let $U^{n+1} = \mathcal{L}(U^n)$ and $\tilde{U}^n = \mathcal{L}^*(U^{n+1})$. Applying the Fourier transform on the two equations and using the assumption $\rho_{\mathcal{L}^*} = \overline{\rho_{\mathcal{L}}}$, we have $\hat{\tilde{U}}^n = |\rho_{\mathcal{L}}|^2\hat{U}^n$. Since $\tilde{U}^n = U^n$ by the definition of time-reversible scheme, $|\rho_{\mathcal{L}}|^2 = 1$. The proof is complete. □

Theorem 2.2.2. *If a linear scheme $\mathcal{L}$ is time-reversible and is at least first order accurate, then $\mathcal{L}$ is at least second order accurate.*

The proof follows that of [23].

Proof. Let $U^{n+1} = \mathcal{L}U^n$. Applying the Fourier transform to the differential equation $u_t = -\mathbf{a} \cdot \bigtriangledown u$ we have

$$\hat{u}_t = -\mathbf{a} \cdot \xi i \hat{u},$$

where ξ is the Fourier dual variable. Therefore

$$\hat{u}(t_{n+1}) = e^{-\mathbf{a}\cdot\xi k i}\hat{u}(t_n) = \{1 - \mathbf{a} \cdot \xi k i - \frac{1}{2}(\mathbf{a} \cdot \xi k)^2 + \mathcal{O}(|\xi k|^3)\}\hat{u}(t_n).$$

Since $\mathcal{L}$ is first order accurate, we can write (see [23])

$$\rho_{\mathcal{L}} = 1 - \mathbf{a} \cdot \xi k i + bk^2 + \mathcal{O}(|\xi k|^3),$$

where b is real since the coefficients in the linear scheme $\mathcal{L}$ (equation (3), equation (4)) are real and independent of k or h. Since $|\rho_{\mathcal{L}}| = 1$ according to Lemma 2.2.1, we have

$$1 = |\rho_{\mathcal{L}}|^2 = 1 + (\mathbf{a} \cdot \xi k)^2 + 2bk^2 + \mathcal{O}(|\xi k|^3).$$

Therefore

$$b = -\frac{1}{2}(\mathbf{a} \cdot \xi)^2.$$

The proof is complete. □

Remark 2.2.3. Consider an ordinary differential equation (ODE) of the form

$$\dot{y} = f(y) \in \mathbb{R}^n, \quad y(0) = y_0\,. \tag{10}$$

A numerical scheme for solving equation (10) has the form $\Phi_{h,f} : \mathbb{R}^n \to \mathbb{R}^n$, where $h > 0$ is the step size. A numerical scheme is said to be *symmetric* if $\Phi_{h,f} = \Phi^{-1}_{-h,f}$ where $\Phi^{-1}_{-h,f}$ is the inverse of $\Phi_{h,f}$ with reversed time step $-h$. Therefore it is easy to check that a numerical scheme is symmetric if and only if it is time reversible.

A large class of semi-implicit schemes for solving ODEs are symmetric. For example

$$y_{(n+1)h} = y_{nh} + h\frac{1}{2}\left(f(y_{nh}) + f(y_{(n+1)h})\right)$$

is a second order symmetric (reversible) scheme. $y_{(n+1)h}$ can be obtained by solving a nonlinear equation. It is known that symmetric (reversible) schemes for solving Hamiltonian systems have higher order of accuracy and can preserve invariant tori over long time intervals [15].

2.3 Attempts to Construct Time Reversible Schemes and Examples

It is known from the last section that time reversible scheme has zero forward-backward error and the complex modulus of its Fourier symbol is a constant one. And it can potentially increase the order of accuracy by one. Motivated by this,

in this section we develop a few new numerical schemes by minimizing its forward-backward error or making its Fourier symbol modulus as close to one as possible while preserving its numerical stability.

2.3.1 Linear Combination of Numerical Schemes for One Way Wave Equation

Our first motivation is inspired by a comparision of some first order numerical schemes for one way wave equations. Consider the following Cauchy problem of the one way wave equation:

$$u_t + au_x = 0 \quad ; \quad u(x,0) = f(x) \tag{11}$$

It is well known that $u(x,t)$ has the following unique solution

$$u(x,t) = f(x - at) \tag{12}$$

Consider the rectangular mesh $\{x_i, t_n\}$ with space grid size Δx and time grid size Δt respectively. The CFL number is $\lambda = \frac{a\Delta t}{\Delta x}$. Let $\mathbf{U}^n = \{U_i^n\}$ be the numerical solution at those grid points when $t = t_n$ where U_i^n approximates the exact solution $u(x_i, t_n)$.

We compare the following numerical schemes:

1. Center Scheme: $U_i^{n+1} = U_i^n - \lambda(U_{i+1}^n - U_{i-1}^n)/2$

2. Lax-Friedrichs Scheme: $U_i^{n+1} = (U_{i+1}^n + U_{i-1}^n)/2 - \lambda(U_{i+1}^n - U_{i-1}^n)/2$

3. Upwind Scheme: $U_i^{n+1} = U_i^n - \lambda(U_i^n - U_{i-1}^n)$ if $\lambda > 0$

Without loss of generality assume that $u(x, t_n)$ is concave in the neighborhood of x_i. By comparing the analytical solution and numerical solutions, it is easy to see that the numerical solution U_i^{n+1} obtained by the center scheme is above the exact solution (overshoot); the numerical solution U_i^{n+1} computed by the Lax-Friedrichs scheme is under the exact solution (undershoot); the numerical solution U_i^{n+1} derived

by the upwind scheme is also under the exact solution but the numerical error is smaller than that of the Lax-Friedrichs scheme. See Figure 1

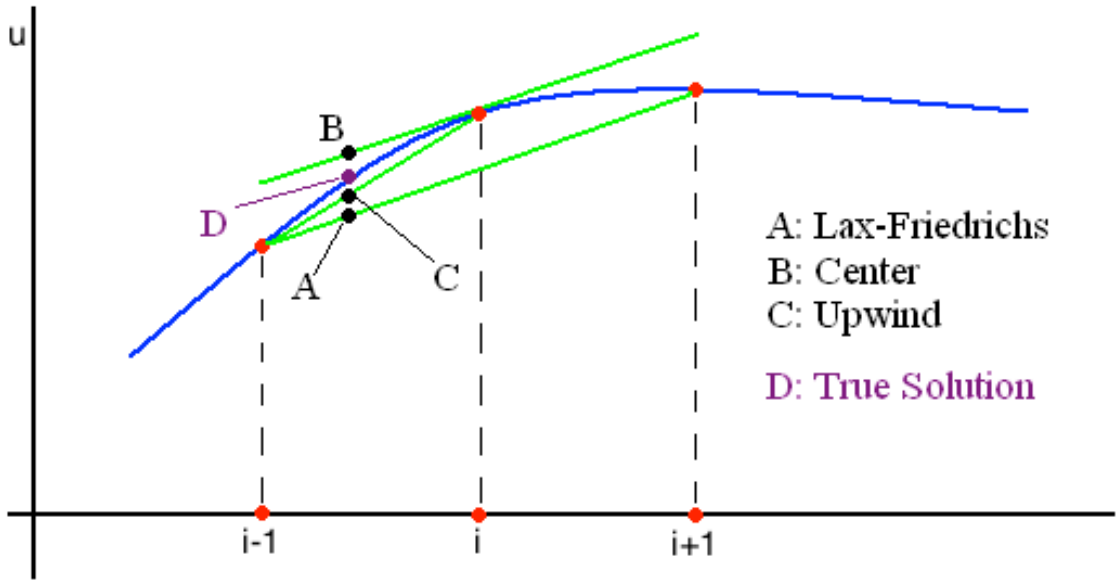

Figure 1: comparison between center scheme, upwind scheme and Lax-Friedrichs scheme for one way wave equation

Therefore, the center scheme tends to amplify the oscillation of numerical solutions and is unconditionally unstable. The upwind scheme and Lax-Friedrichs scheme tend to reduce the oscillation of numerical solutions. Although upwind scheme and Lax-Friedrichs scheme are numerically stable, numerical diffusions are inevitable in these types of schemes.

Motivated by this, we propose to construct a class of new numerical schemes by taking a linear combination of center scheme and Lax-Friedrichs scheme. The goal is to minimize the overshoot/undershoot while maintaining the numerical stability.

Consider a new scheme whose operator has the from $\theta LF+(1-\theta)C$, where LF and C are the operator of the Lax-Friedrichs Scheme and the Center Scheme respectively and θ is a constant in $(0,1)$ that is to be determined.

Write this scheme explicitly

$$\begin{aligned} U_i^{n+1} &= \theta[(U_{i+1}^n + U_{i-1}^n)/2 - \lambda(U_{i+1}^n - U_{i-1}^n)/2] + (1-\theta)[U_i^n - \lambda(U_{i+1}^n - U_{i-1}^n)/2] \\ &= (1-\theta)U_i^n + \theta(U_{i+1}^n + U_{i-1}^n)/2 - \lambda(U_{i+1}^n - U_{i-1}^n)/2 \end{aligned} \tag{13}$$

Theorem 2.3.1. *Let $L = \theta LF + (1-\theta)C$ be the new scheme constructed from the linear combination of Lax-Friedrichs scheme and Center scheme for equation (11). Let λ be the CFL number. Then the optimal θ is λ^2 in the sense that L is stable, inherits least diffusion from $L - F$ scheme, and its modulus of Fourier symbol is the closest to* 1.

Proof. From equation (13), it is easy to see that the Fourier symbol of L is

$$\begin{aligned} \rho &= (1-\theta) + (\theta-\lambda)e^{i\xi}/2 + (\theta+\lambda)e^{-i\xi}/2 \\ &= (1-\theta) + \theta\cos\xi - i\lambda\sin\xi \end{aligned} \tag{14}$$

Therefore,

$$\begin{aligned} |\rho|^2 &= (1-\theta)^2 + 2\theta(1-\theta)\cos\xi + \lambda^2 + (\theta^2-\lambda^2)\cos^2\xi \\ &= 2\theta(1-\theta)\cos\xi + (\theta^2-\lambda^2)\cos^2\xi + (1-\theta)^2 + \lambda^2 \end{aligned} \tag{15}$$

Let $x = \cos\xi$, then $|\rho|^2 = f(x) = 2\theta(1-\theta)x + (\theta^2-\lambda^2)x^2 + (1-\theta)^2 + \lambda^2$. Here $|x| \le 1$ and $0 \le \theta \le 1$

To make L stable and inherit the least diffusion from L-F scheme, θ should be as small as possible while $f(x) \le 1$ for any $-1 \le x \le 1$.

It's easy to verify that $f(1) = 1$ and $f(-1) = (2\theta-1)^2 \le 1$. Obviously $f(x) \ge 0$. To minimize θ and let $|\rho|^2$ be as close to 1 as possible, we let the parabola open downwards and with the axis of symmetry exact at $x = 1$. That means $\theta^2 - \lambda^2 < 0$ and $\frac{\theta^2-\theta}{\theta^2-\lambda^2} = 1$

Therefore, the optimal of θ is λ^2. The proof is complete. □

Example: Consider a specific example of one way wave equation

$$u_t + 0.5u_x = 0 \quad ; \quad u(x,0) = sin(2\pi x)$$

and let $\lambda = 0.5$. We compare the L^1 error for different linear combination of L-F and Center scheme.

Table 1: Numerical accuracy of different linear combination of L-F and Center. CFL number = 0.5;

L^1 error	$\theta = 1$	$\theta = \|\lambda\|$	$\theta = \lambda^2$
N = 100	0.0868	0.0303	$9.9244e-04$
N = 200	0.0452	0.0154	$2.4744e-04$
N = 400	0.0231	0.0078	$6.1772e-05$

$\theta = \lambda^2$ results in the best performance. It actually has second order and is essentially the quadratic ENO scheme.

2.3.2 High order correction using multiple times of BFECC

A numerical scheme with back and forth error compensation and correction causes the forward-backward error smaller in general compared to the original scheme alone. What about we perform BFECC iteratively?

Let S be the numerical operator for the underlying numerical scheme. The high order correction iterative algorithm can be described as follows where $\tilde{\mathbf{U}}$ and $\tilde{\tilde{\mathbf{U}}}$ are temporary variables.

1. **Initialization**

   Let $\tilde{\mathbf{BF}}_0 = \mathbf{U^n}$, $\mathbf{BF}_0 = S(\mathbf{U^n})$

2. **Backward advection**

   $\tilde{\mathbf{U}} = S^*(\mathbf{BF}_m)$

3. **Adjustment due to previous iterations**

   $\tilde{\tilde{\mathbf{U}}} = \tilde{\mathbf{U}} + (\tilde{\mathbf{BF}}_\mathbf{m} - \mathbf{U^n})$

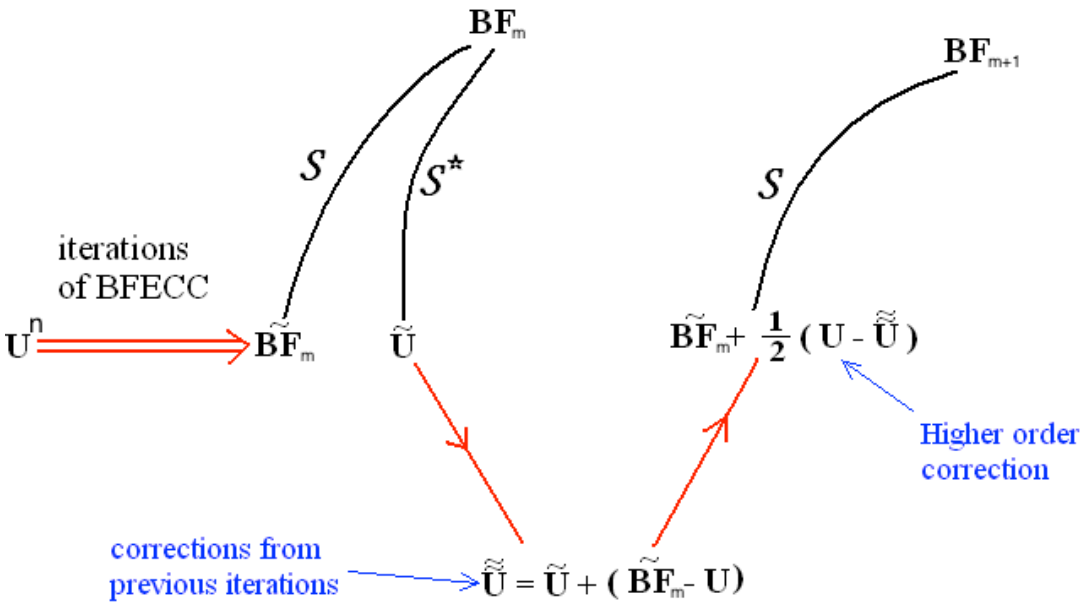

Figure 2: Iterative BFECC

4. **High order correction**

   $\tilde{\mathbf{BF}}_{m+1} = \tilde{\mathbf{BF}}_m + \frac{1}{2}(\mathbf{U}^n - \tilde{\tilde{\mathbf{U}}})$

5. **Forward advection**

   $\mathbf{BF}_{m+1} = S(\tilde{\mathbf{BF}}_{m+1})$

Here m is the number of iterations. When $m = 0$, the numerical scheme is $\mathbf{BF}_0$ which is exactly the underlying scheme S. When $m = 1$, it is $\mathbf{BF}_1$, i.e S plus BFECC. And when $m = 2$, it is $\mathbf{BF}_2$ which includes two forward-backward error corrections and one subsequent forward advection. That is three forward advections and two backward advections in total. As you see, the computational complexity increases by one additional forward advection and one additional backward advection as the number of iterations increases by one.

Theorem 2.3.2. *Let* $\mathbf{BF}_m$ *be the above numerical scheme where* m *is the number of*

iterations. Let ρ_m be the Fourier symbol of $\mathbf{BF}_m$. If $|\rho| = |\rho_0| < 1$, then

$$\lim_{m\to\infty} |\rho_m| := \rho_* = \frac{2|\rho|}{1+|\rho|^2}\,. \tag{16}$$

Further

$$\rho_* - |\rho_{m+1}| = \frac{1}{2}(1-|\rho|^2)(\rho_* - |\rho_m|)\,. \tag{17}$$

Proof. Let $\tilde{\rho}_m$ be the Fourier symbol for $\tilde{\mathbf{BF}}_m$. Given that $\mathbf{BF}_m$ is just the forward advection of $\tilde{\mathbf{BF}}_m$, we have $\rho_m = \rho\tilde{\rho}_m$.

From the iterative algorithm, we have

$$\begin{aligned} \tilde{\mathbf{BF}}_{m+1} &= \tilde{\mathbf{BF}}_m + \tfrac{1}{2}(\mathbf{U} - \tilde{\tilde{U}}) \\ &= \tilde{\mathbf{BF}}_m + \tfrac{1}{2}(\mathbf{U} - \tilde{\mathbf{U}} - \tilde{\mathbf{BF}}_m + \mathbf{U}) \\ &= \tfrac{1}{2}\tilde{\mathbf{BF}}_m + \mathbf{U} - \tfrac{1}{2}\tilde{\mathbf{U}} \end{aligned} \tag{18}$$

Therefore $\tilde{\rho}_{m+1} = \frac{1}{2}\tilde{\rho}_m + 1 - \frac{1}{2}\bar{\rho}\rho_m$. Multiply ρ on both sides, we have

$$\begin{aligned} \rho_{m+1} &= \tfrac{1}{2}\rho_m + \rho - \tfrac{1}{2}\rho\bar{\rho}\rho_m \\ &= \rho + \tfrac{1}{2}(1-\rho\bar{\rho})\rho_m. \end{aligned} \tag{19}$$

Take the modulus on both sides and use the fact that $\rho\bar{\rho}$ is a real number, we got $|\rho_{m+1}| = |\rho| + \frac{1}{2}(1-\rho\bar{\rho})|\rho_m|$. As m goes to infinity, we have $\rho_* = |\rho| + \frac{1}{2}(1-|\rho|^2)\rho_*$. Solve for ρ_* and we get $\rho_* = \frac{2|\rho|}{1+|\rho|^2}$. Also equation (17) can be easily verified. The proof is complete. □

Since $|\rho| \le 1$, we have $\rho_* \ge |\rho|$ and $\rho_* \le 1$. In other words, by doing the high order correction, we can improve the Fourier symbol and make its modulus closer to 1. Also the closer $|\rho|$ is to 1, the faster it converges. However, it is not converging to a reversible scheme.

2.3.3 Composed BFECC

In the last section, we demonstrated the effect of multiple iterations of forward and backward error correction. Note that for the high order correction, the underlying

scheme for the forward or backward advection is always the original scheme S. One more iteration will only cause one more forward advection and one more backward advection. Now we consider a different type of iteration based on BFECC, where the underlying scheme is the numerical scheme of previous iteration.

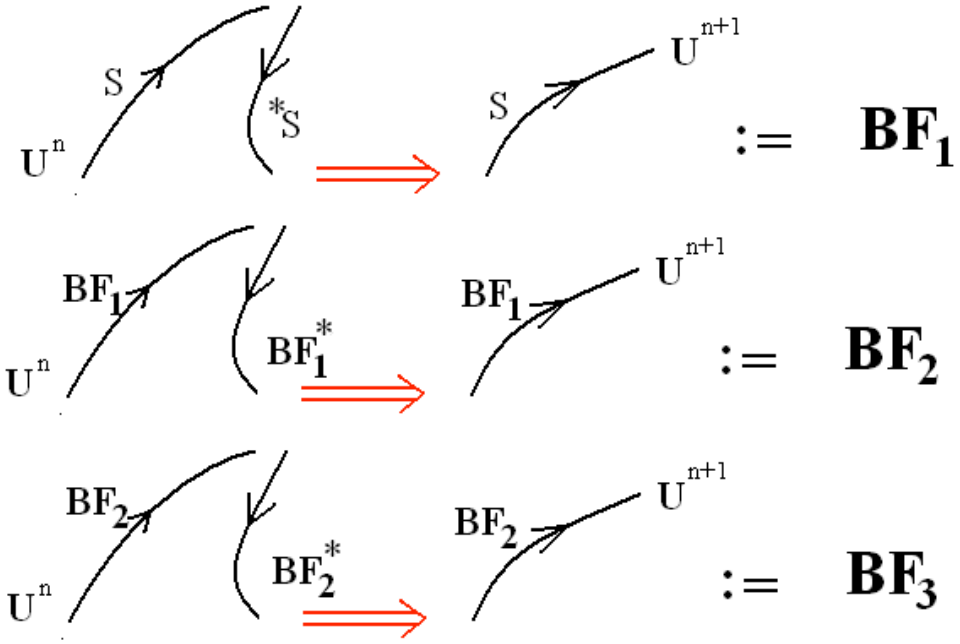

Figure 3: composed BFECC

In the composed BFECC, $\mathbf{BF}_1$ has two forward advections of underlying scheme S and one backward advection for S^* in total while $\mathbf{BF}_2$ results in six forward advection of S and three backward advection for S^*. In other words, one more iteration would cause three times of computational complexity, which is actually unpractical in real computaion.

Theorem 2.3.3. *Let ρ_n be the Fourier symbol of BF_n. If $|\rho_0| < 1$, then*

$$\lim_{n\to\infty} |\rho_n| := 1.$$

Proof. From the Fourier symbol relation between underlying scheme and the underlying scheme plus BFECC, it is straightforward to get $\rho_{n+1} = \rho_n(1 + \frac{1}{2}(1 - \bar{\rho_n}\rho_n))$

Let $x_n = |\rho_n|$

$x_{n+1} = x_n(1 + \frac{1}{2}(1 - x_n^2))$

$1 - x_{n+1} = \frac{2-x_n-x_n^2}{2}(1 - x_n)$

Since$0 \leq x_n \leq 1$, $|x_n| \to 1$ The proof is complete. □

Different from the high order correction type of iteration, this kind of iteration can actually result in a time reversible scheme, though the computational complexity triples for every one more iteration.

2.4 An interpolation algorithm motivated by BFECC

As an independent extension of BFECC method, in this section we introduce an interpolation algorithm called BFECC interpolation using the idea of back and forth error correction.

Consider a C^3 smooth function $f(x)$ whose value is only known on a mesh $\{x_i\}_{i=1}^N$. To estimate the value of $f(x_I)$ at some intermediate point x_I, one can either only use the information of its nearest neighbors to obtain a second order approximation; or use the information of more points to obtain a more accurate estimation. For the latter method, looking beyond closest neighbors might not be easy for unstructured mesh especially in higher dimensions.

One promising idea is to consider the interpolation problem as a one-way wave equation $u_t = 0u_x$ with initial condition $u(x, 0) = f(x)$. Then we choose a finite number of auxiliary points $x_i^I, \cdots, x_k^I$ in the neighborhood of x_I to construct a local new mesh together with x_I. Then we implement BFECC on this equation:

1. **Forward interpolation:**

   Use $f(x_i)$, $i = 1 \sim N$ and linear interpolation to estimate $\hat{f}(x_i^I)$, $i = 1 \sim k$;

2. **Backward interpolation:**

   Use $\hat{f}(x_i^I)$ and linear interpolation to estimate $\tilde{f}(x_{k_1}), \cdots, \tilde{f}(x_{k_m})$, where $x_{k_1}, \cdots, x_{k_m}$

are located in the neighborhood of x_I;

3. **Forward interpolation using modified data:**
   Let $f^*(x_{k_i}) = f(x_{k_i}) + \frac{1}{2}(f(x_{k_i}) - \tilde{f}(x_{k_i}))$ for all $i = 1 \sim m$. Use linear interpolation and $f^*(x_{k_i})$, $i = 1 \sim m$ to estimate $f(x_I)$.

The main advantage of this BFECC interpolation algorithm is that one can apply local linear interpolation a few times to obtain third order accuracy, which could be more convenient for unstructured meshes since one only needs to interpolate with closest neighbors. If using direct local quadratic interpolation, one needs to look beyond closest neighbors. Assume we can construct suitable x_i^I as the new local mesh. Similar as the BFECC method, the BFECC interpolation algorithm has one more order of accuracy.

Lemma 2.4.1. *Suppose $\{x_i\}_{i=1}^N$ satisfy $x_{i+1} - x_i = h$ for some constant h. Let $x_k < x_I < x_{k+1}$ and consider the local auxiliary points $x_1^I = x_{k-1} + (x_I - x_k)$ and $x_2^I = x_{k+1} + (x_I - x_k)$. If $f(x)$ is C^3 smooth, then BFECC interpolation algorithm has third order of accuracy.*

Proof. Let $\alpha = (x_I - x_k)/h$. Then we have

$$\hat{f}(x_I) = (1-\alpha)f(x_k) + \alpha f(x_{k+1})$$

$$\hat{f}(x_1^I) = (1-\alpha)f(x_{k-1}) + \alpha f(x_k)$$

and

$$\hat{f}(x_2^I) = (1-\alpha)f(x_{k+1}) + \alpha f(x_{k+2})\,.$$

Therefore

$$\tilde{f}(x_k) = f(x_k) + \alpha(1-\alpha)[f(x_{k-1}) + f(x_{k+1}) - 2f(x_k)]$$

and

$$\tilde{f}(x_{k+1}) = f(x_{k+1}) + \alpha(1-\alpha)[f(x_k) + f(x_{k+2}) - 2f(x_{k+1})]\,.$$

Since $f(x)$ is C^3 smooth, after some calculation one has

$$f^*(x_k) = f(x_k) - \frac{1}{2}\alpha(1-\alpha)f''(x_k)h^2 + o(h^3)$$

and

$$f^*(x_{k+1}) = f(x_{k+1}) - \frac{1}{2}\alpha(1-\alpha)f''(x_k)h^2 + o(h^3)\,.$$

Therefore

$$\begin{aligned} f^*(x_I) &= (1-\alpha)f^*(x_k) + \alpha f^*(x_{k+1}) \\ &= f(x_k) + \alpha(f(x_{k+1}) - f(x_k)) - \frac{1}{2}\alpha(1-\alpha)f''(x_k)h^2 + o(h^3) \\ &= f(x_k) + f'(x_k)(\alpha h) + \frac{1}{2}f''(x_k)(\alpha h)^2 + o(h^3) \\ &= f(x_k + \alpha h) + o(h^3) = f(x_I) + o(h^3)\,. \end{aligned}$$

This completes the proof.

□

In other words, the new local mesh can be obtained by shifting the old mesh such that x_I becomes a mesh point in the new mesh. This idea works for higher dimensional problems too.

We demonstrate three numerical examples including linear interpolation on 1d mesh, linear interpolation on 2d Equilateral-Triangle-Mesh (ETM) and bilinear interpolation on 2d square mesh. The average interpolation error of 100 randomly chosen x_I are demonstrated in each of three examples. Note that for the bilinear interpolation on square mesh we mean the interpolation polynomial has the form of $b_1 + b_2x + b_3y + b_4xy$ piecewisely.

The most practical application of BFECC interpolation is the high order interpolation on high dimensional unstructured mesh. The computational cost of obtaining information of neighbor data points on unstructured mesh is high. BFECC interpolation only uses information of adjacent data points, which is a significant advantage

Table 2: Linear Interpolation with and without BFECC on 1D mesh. N data points are uniformly distributed on the unit interval [0, 1]. Interpolation function $f(x) = \sin x$. Interpolation errors are the average error at 100 randomly chosen intermediate points.

Number of points	Avarage error without BFECC	order	Average error with BFECC	order
11	1.1442×10^{-4}	N/A	1.7467×10^{-7}	N/A
21	2.8285×10^{-5}	2.016	1.9723×10^{-8}	3.147
41	7.7154×10^{-6}	1.874	2.5145×10^{-9}	2.972
81	1.8399×10^{-6}	2.068	3.0301×10^{-10}	3.052

Table 3: Linear Interpolation with and without BFECC on triangular 2D mesh. Data points form equilateral triangles on the unit square $[0, 1] \times [0, 1]$. The grid size is the length of sides of triangles. Interpolation function $f(x, y) = \sin(x + y)$. Interpolation errors are the average error at 100 randomly chosen intermediate points.

Number of points	Avarage error without BFECC	order	Average error with BFECC	order
11×11	1.1338×10^{-3}	N/A	6.8822×10^{-6}	N/A
21×21	3.0108×10^{-4}	1.913	7.6107×10^{-7}	3.177
41×41	7.6802×10^{-5}	1.971	9.1614×10^{-8}	3.054
81×81	1.7828×10^{-5}	2.107	1.0648×10^{-8}	3.105

in this situation. However, the way to choose auxiliary local mesh is a nontrivial problem. If the local new mesh is obtained by shifting the old mesh, the accuracy can be significantly improved. However according to our numerical experiments, the improvement is less than one more order of accuracy. We propose to develop an effective way to construct the auxiliary mesh in our future work.

Table 4: Bilinear Interpolation with and without BFECC on triangular 2D mesh. Data points form squares on the unit square $[0, 1] \times [0, 1]$. The grid size is the length of sides of squares. Interpolation function $f(x, y) = \sin(x + y)$. Interpolation errors are the average error at 100 randomly chosen intermediate points.

Number of points	Avarage error without BFECC	order	Average error with BFECC	order
11×11	1.6028×10^{-3}	N/A	7.7288×10^{-6}	N/A
21×21	3.8220×10^{-4}	2.068	7.6521×10^{-7}	3.336
41×41	9.2637×10^{-5}	2.045	7.3007×10^{-8}	3.390
81×81	2.5025×10^{-5}	1.888	9.3541×10^{-9}	2.964

CHAPTER III

A LIMITING STRATEGY FOR THE BFECC METHOD

3.1 Eliminating Spurious Oscillations

When the solution is not smooth, a nonlinear limiting technique is usually required for second order (or higher order) schemes to remove spurious oscillations from the numerical solution.

Let $\mathcal{L}$ be a linear scheme and let $e^{(1)} = \frac{1}{2}(U^n - \mathcal{L}^*\mathcal{L}U^n)$ be the back-and-forth error where U^n is the numerical solution at the time t_n. If we replace $\mathcal{L}$ with $\mathcal{L}_k = \mathbf{BF_k}$, the k-th iteration of the BFECC procedure as defined in Sec. 2.3.3, we expect the back-and-forth error $e^{(1)}$ to decrease with increasing k wherever the solution is locally sufficiently smooth because of Theorem 2.3.3. However, this is not practical due to the complexity of even computing $\mathcal{L}_1^*\mathcal{L}_1U^n$ (with $k = 1$ in Figure (3)). $\mathcal{L}_1U^n$ is supposed to be more accurate than $\mathcal{L}U^n$ in approximating the solution at the time level t_{n+1} provided the solution is smooth. If we transform $\mathcal{L}_1U^n$ back to the time level t_n with $\mathcal{L}^*$ instead of $\mathcal{L}_1^*$, a larger error of approximately $e^{(1)}$ is also introduced. Therefore with the error correction, $\mathcal{L}^*\mathcal{L}_1U^n + e^{(1)}$ should be an accurate approximation of $\mathcal{L}_1^*\mathcal{L}_1U^n$ with smaller cost. We define another error at the time level t_n as $e^{(2)} = U^n - (\mathcal{L}^*\mathcal{L}_1U^n + e^{(1)})$ and have the following theorem.

Theorem 3.1.1. *Let $\mathcal{L}$ be a linear scheme, $e^{(1)} = \frac{1}{2}(U^n - \mathcal{L}^*\mathcal{L}U^n)$ and $e^{(2)} = U^n - (\mathcal{L}^*\mathcal{L}_1U^n + e^{(1)})$ where $\mathcal{L}_1$ is the scheme obtained by applying BFECC to $\mathcal{L}$, then $e^{(2)} = e^{(1)} - \mathcal{L}^*\mathcal{L}e^{(1)}$.*

Proof.

$$\begin{aligned} e^{(2)} &= U^n - \mathcal{L}^*\mathcal{L}_1 U^n - e^{(1)} \\ &= U^n - \mathcal{L}^*[\mathcal{L}(\tfrac{3}{2}I - \tfrac{1}{2}\mathcal{L}^*\mathcal{L})]U^n - e^{(1)} \\ &= U^n - \mathcal{L}^*\mathcal{L}U^n - \mathcal{L}^*\mathcal{L}[\tfrac{1}{2}(U^n - \mathcal{L}^*\mathcal{L}U^n)] - e^{(1)} \\ &= 2e^{(1)} - \mathcal{L}^*\mathcal{L}e^{(1)} - e^{(1)} \\ &= e^{(1)} - \mathcal{L}^*\mathcal{L}e^{(1)}. \end{aligned} \tag{20}$$

The proof is complete. □

Now we are able to estimate the size of $e^{(2)}$ relative to $e^{(1)}$.

Corollary 3.1.2. *Let $\mathcal{L}$ be a linear scheme with its amplification factor $|\rho_{\mathcal{L}}| \leq \sqrt{2}$, then $||e^{(2)}|| \leq ||e^{(1)}||$ where $||\cdot||$ denotes the l^2-norm.*

Proof. From Theorem 3.1.1 we obtain via Fourier transform

$$\hat{e}^{(2)} = (1 - |\rho_{\mathcal{L}}|^2)\hat{e}^{(1)}.$$

With the Parseval's identity, the proof is complete. □

Therefore on average $|e^{(2)}|$ is no more than $|e^{(1)}|$ as we expect. However in the non-smooth area of the solution, $|e^{(2)}|$ could be larger than $|e^{(1)}|$. In fact at a grid point where $|e^{(2)}|$ is greater than $|e^{(1)}|$, there could be overshoots of the numerical solution caused by large values of $e^{(1)}$ at adjacent grid points. To see this, let's suppose

$$(\mathcal{L}^*\mathcal{L}e^{(1)})(x_i) = \sum_{j\in I} c_j e^{(1)}_{i+j} \tag{21}$$

where the set $i + I$ contains all grid point indices involved in the computation of $\mathcal{L}^*\mathcal{L}e^{(1)}$ at x_i, in particular, $i \in i + I$. Suppose $\mathcal{L}$ is consistent, monotone and at least first order accurate. Then it's easy to see by using the Taylor expansion around x_i that

$$\begin{aligned} &(a)\ \ 0 \leq c_j \leq 1, \ \text{ for all } \ j \in I; \\ &(b)\ \ \textstyle\sum_{j\in I} c_j = 1; \ \text{ and} \\ &(c)\ \ \textstyle\sum_{j\in I, j\neq 0} j c_j = 0. \end{aligned} \tag{22}$$

Theorem 3.1.3. *Suppose the linear scheme $\mathcal{L}$ is consistent, monotone and at least first order accurate. If $|e_i^{(1)}|$ is a maximum among $\{|e_{i+j}^{(1)}| : j \in I\}$, and $(\mathcal{L}^*\mathcal{L}e^{(1)})(x_i)$ is of the same sign as $e_i^{(1)}$, then*

$$|e_i^{(2)}| \leq |e_i^{(1)}|.$$

Proof. Without loss of generality, suppose $e_i^{(1)} > 0$. Since $\sum_{j\in I} c_j e_{i+j}^{(1)}$ is a convex average of $e_{i+j}^{(1)}$ and $e_i^{(1)}$ is the local maximum, we have

$$\sum_{j\in I} c_j e_{i+j}^{(1)} \leq e_i^{(1)}.$$

Also since $(\mathcal{L}^*\mathcal{L}e^{(1)})(x_i) = \sum_{j\in I} c_j e_{i+j}^{(1)} > 0$ from (21) and the assumption of the theorem, we conclude that

$$|e_i^{(1)} - (\mathcal{L}^*\mathcal{L}e^{(1)})(x_i)| \leq |e_i^{(1)}|.$$

Recalling Theorem 3.1.1, the proof is complete. □

Therefore a local maximum in $|e^{(1)}|$ that is large enough will result in a smaller $|e^{(2)}|$ at the same location. However, since

$$e_i^{(2)} = e_i^{(1)} - (\mathcal{L}^*\mathcal{L}e^{(1)})(x_i) = e_i^{(1)} - \sum_{j\in I} c_j e_{i+j}^{(1)},$$

a $e_{i+j}^{(1)}$ with much larger absolute value than that of $e_i^{(1)}$ tends to cause $|e_i^{(2)}| > |e_i^{(1)}|$. Therefore wherever $|e_i^{(2)}| > |e_i^{(1)}|$ is detected, we limit its adjacent back-and-forth error $e_{i+j}^{(1)}$ to no larger than $e_i^{(1)}$ in their absolute values.

Remark. Let r be the order of accuracy of the scheme $\mathcal{L}$. Even if this limiting procedure is applied accidentally in the smooth area of the solution, it's only going to create a $\mathcal{O}(h^{r+2})$ local error which won't change the order of accuracy after applying the BFECC algorithm, as long as the modified value $\tilde{e}_{i+j}^{(1)}$ (after applying the limiting

procedure) is a convex average of $e_{i+j}^{(1)}$ and $e_i^{(1)}$. In fact,

$$\begin{aligned} \tilde{e}_{i+j}^{(1)} - e_{i+j}^{(1)} &= (1-\theta)e_{i+j}^{(1)} + \theta e_i^{(1)} - e_{i+j}^{(1)}, \text{ for some } \theta \in [0,1] \\ &= \theta(e_i^{(1)} - e_{i+j}^{(1)}) \\ &= \mathcal{O}(h^{r+2}), \end{aligned} \tag{23}$$

since $e^{(1)} = \mathcal{O}(h^{r+1})$ (it is proportional to the local error of scheme $\mathcal{L}$).

3.2 Limiting Algorithm

Let $\mathcal{L}$ be a linear scheme and U^n be the numerical solution given at the time t_n, then the BFECC algorithm coupled with the limiting technique can be implemented as follows.

1. **Forward advection.**

   $\tilde{U}^{n+1} = \mathcal{L}U^n$.

2. **Backward advection.**

   $\tilde{U}^n = \mathcal{L}^*\tilde{U}^{n+1}$.

3. **Forward advection again using modified solution at the time t_n.**

   $\tilde{V}^{n+1} = \mathcal{L}(U^n + e^{(1)})$, where $e^{(1)} = \frac{1}{2}(U^n - \tilde{U}^n)$.

4. **Backward advection to define a comparative error $e^{(2)}$.**

   $e^{(2)} = U^n - (\mathcal{L}^*(\tilde{V}^{n+1}) + e^{(1)})$.

5. **Limiting.**

   Define a copy of $e^{(1)}$, $\tilde{e}^{(1)} = e^{(1)}$. At every grid point x_i such that $|e_i^{(2)}| > |e_i^{(1)}|$, perform the limiting at adjacent grid points so that $\tilde{e}_j^{(1)} = \text{minmod}(e_i^{(1)}, \tilde{e}_j^{(1)})$, for every grid point j adjacent to grid point i, $j \neq i$.

6. **Forward advection with modified solution at the time t_n.**

   $U^{n+1} = \mathcal{L}(U^n + \tilde{e}^{(1)})$.

Here

$$\text{minmod}(x,y) = \begin{cases} \min(x,y), & \text{if } x,y > 0, \\ \max(x,y), & \text{if } x,y < 0, \\ 0, & \text{otherwise.} \end{cases}$$

It is a commonly used limiter function that returns a convex average of x and y. This procedure is very easy to implement because basically it calls a subroutine (scheme $\mathcal{L}$) 5 times. Note that the limiting procedure modifies $e^{(1)}$ only in the vicinities of singularities of the solution, and the backward advection step in Step 4. can be applied selectively. This could reduce the complexity to about 4 times that of scheme $\mathcal{L}$ by first using a low cost detector(e.g. in [12]) to find the non smooth area of the solution. One could further reduce the cost by applying the last advection step in the non smooth area only.

3.3 Numerical Results

The performance of BFECC with the limiting algorithm is demonstrated by the following numerical examples.

3.3.1 1-D linear equation

Consider the following 1-D linear equation

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial u(x,t)}{\partial x} = 0, \qquad (x,t) \in [0,2] \times [0,20] \tag{24}$$

$$u(x,0) = 1, \qquad x \in [\frac{2}{3}, \frac{4}{3}] \tag{25}$$

$$u(x,0) = 0, \qquad \text{otherwise}$$

with periodic boundary conditions. The solution of equation (24) at the final time $T = 20$ is identical to the initial solution. We compare the performance of CIR, CIR+BFECC and CIR+BFECC+limiting algorithms for the present 1-D linear equation with varying CFL numbers.

Figure 4 presents the numerical solutions with $CFL = 0.8$. Note that the CIR scheme with CFL number less than 1 is the same as the upwind scheme. Our limiting algorithm eliminates the spurious oscillations that appear in the CIR+BFECC algorithm. Also the shape of the square wave is well-preserved by the CIR+BFECC+limiting algorithm at $T = 20$. By contrast, very strong numerical smearing exists in the CIR method.

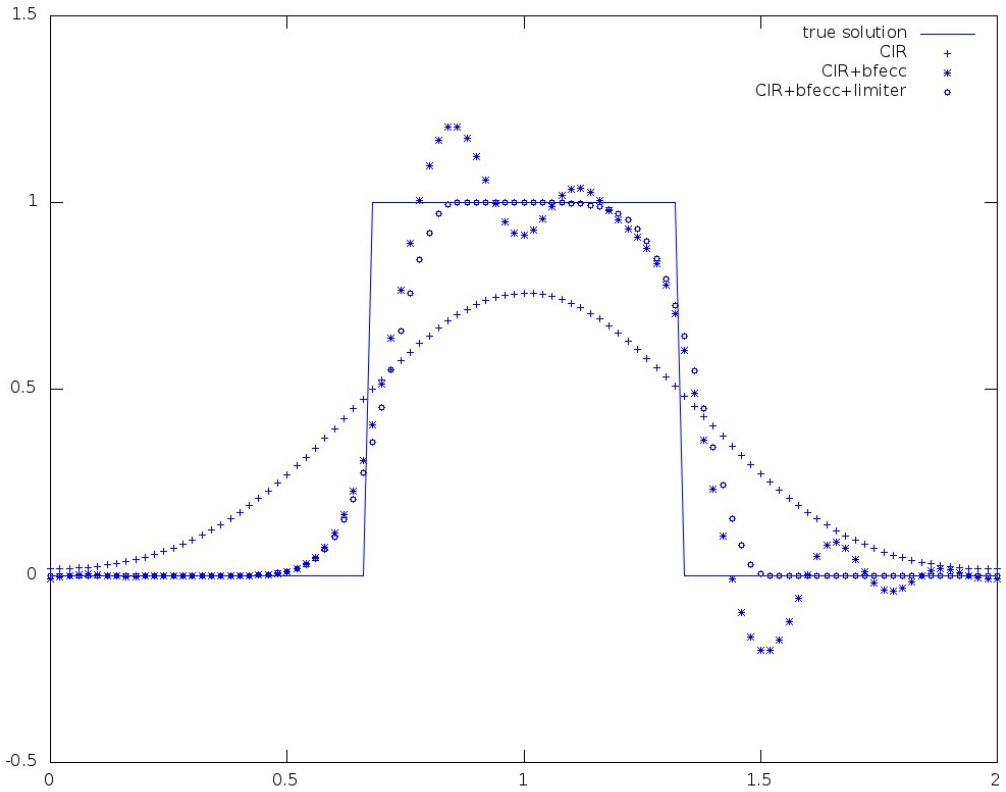

Figure 4: 1-D linear equation with CFL = 0.8, square wave initial condition. $h = 0.02$, $T = 20$.

The numerical results with CFL number 5.8 are shown in Figure 5. The shape of the square wave is better preserved by all three methods because there are less computational steps. The spurious oscillations in the CIR+BFECC algorithm are again eliminated by the limiting procedure, while strong numerical diffusion is still appearing in the CIR scheme. Numerical solutions of equation (24) with different

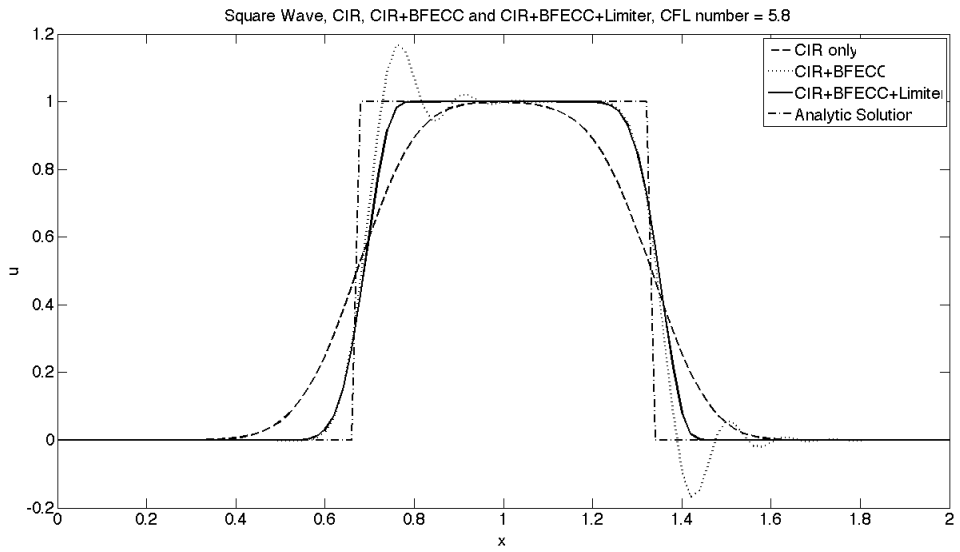

Figure 5: 1-D linear equation with CFL $= 5.8$, square wave initial condition, $h = 0.02$, $T = 20$.

initial conditions are presented in the following. For the *pyramid* initial condition, we mean the following function defined on the interval $[0, 2]$

$$u(x, 0) = 2(1 - |x - 1|).$$

And the *curved square wave* initial condition is the cubic function defined on the interval $[0, 2]$ as follows

$$u(x, 0) = \begin{cases} \frac{53}{26} - \frac{4}{13}(2 - x)^3, & x \in [\frac{2}{3}\frac{4}{3}) \\ 0, & \text{otherwise}\,. \end{cases}$$

In Figure 6 we compare the results of three different methods (CIR, CIR+BFECC and CIR+BFECC+Limiting) for equation (24) with the pyramid initial condition. The results with the curved square wave initial condition are shown in Figure 7. Clearly the limiting procedure removes all the artifacts generated by CIR+BFECC while retaining its higher resolution.

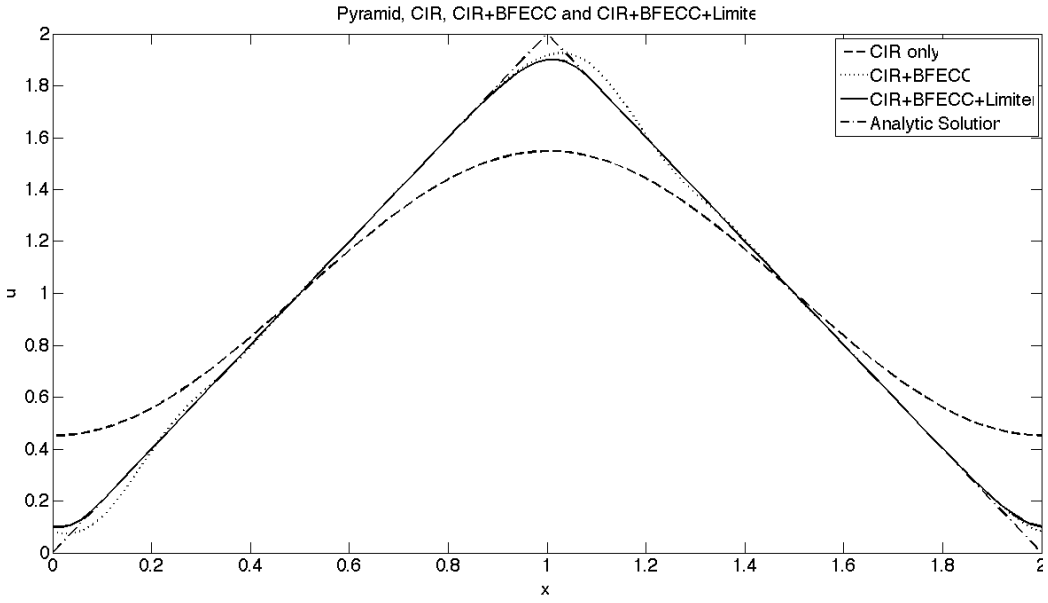

Figure 6: 1-D linear equation with CFL = 0.8 , pyramid initial condition, $h = 0.02$, $T = 20$.

3.3.2 1D linear problem with nonzero forcing

We also consider the following problem with a nonzero forcing term

$$\begin{aligned} &\frac{\partial u(x,t)}{\partial t} + \frac{\partial u(x,t)}{\partial x} = \frac{1}{2}, \qquad (x,t) \in [0,2] \times [0,20] \\ &u(x,0) = 2(1 - |x-1|), \end{aligned} \tag{26}$$

with periodic boundary conditions. The initial solution is still the "pyramid" function. In Figure 8 and 9, numerical solutions of equation (26) are demonstrated. The CFL numbers used are 0.8 and 5.8 respectively. It's clear that the limiting algorithm performs well for this problem.

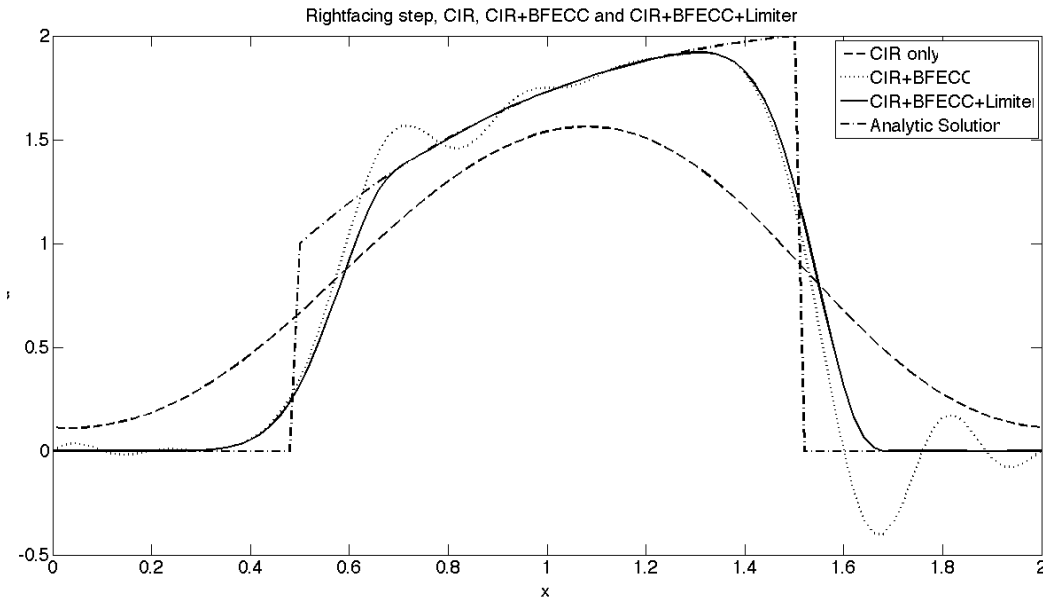

Figure 7: 1-D linear equation with CFL $= 0.8$, curved square wave initial condition, $h = 0.02$, $T = 20$.

3.3.3 2D linear problem

We study a 2-D rotation of a "cubic stair" on the domain $[0\,,100] \times [0\,,100]$

$$u_0(x,y) = \begin{cases} 64000(\frac{53}{26} - \frac{4}{13}(\frac{x}{40} + \frac{3}{8})^3), & (x,y) \in [5\,,45] \times [5\,,45] \\ 0, & \text{otherwise}\,. \end{cases}$$

Consider the equation

$$\frac{\partial \phi}{\partial t} + \vec{v} \cdot \nabla \phi = 0 \tag{27}$$

with the initial condition $\phi(x,y,0) = u_0(x,y)$, where

$$\vec{v}(x,y) = (\frac{\pi}{314}(50-y)\,, \frac{\pi}{314}(50-x))\,.$$

Equation (27) describes the linear rotation around the center $(50\,,50)$ with the shape of $u_0(x,y)$ preserved.

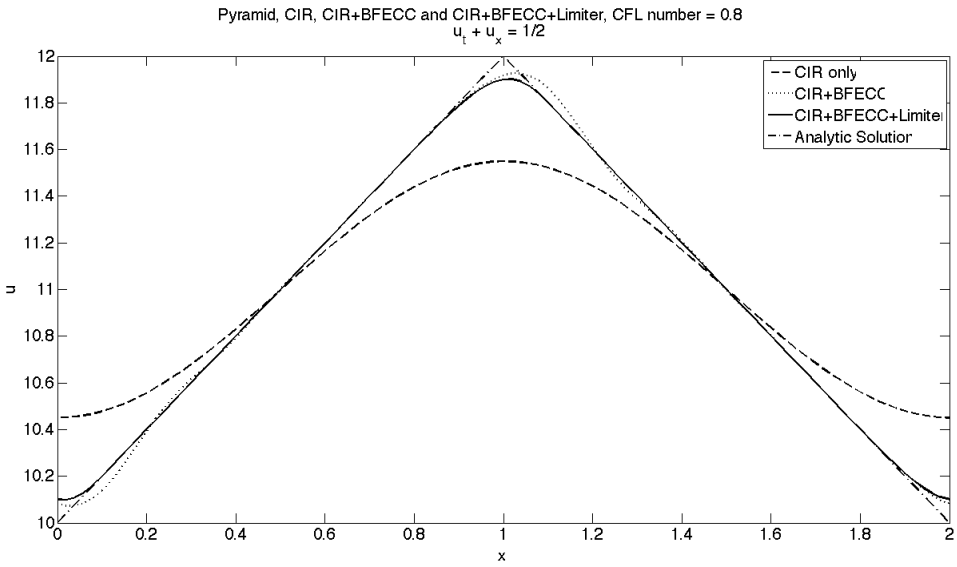

Figure 8: 1-D linear equation with nonzero right-hand side, CFL = 0.8 , pyramid initial condition, $h = 0.02$, $T = 20$.

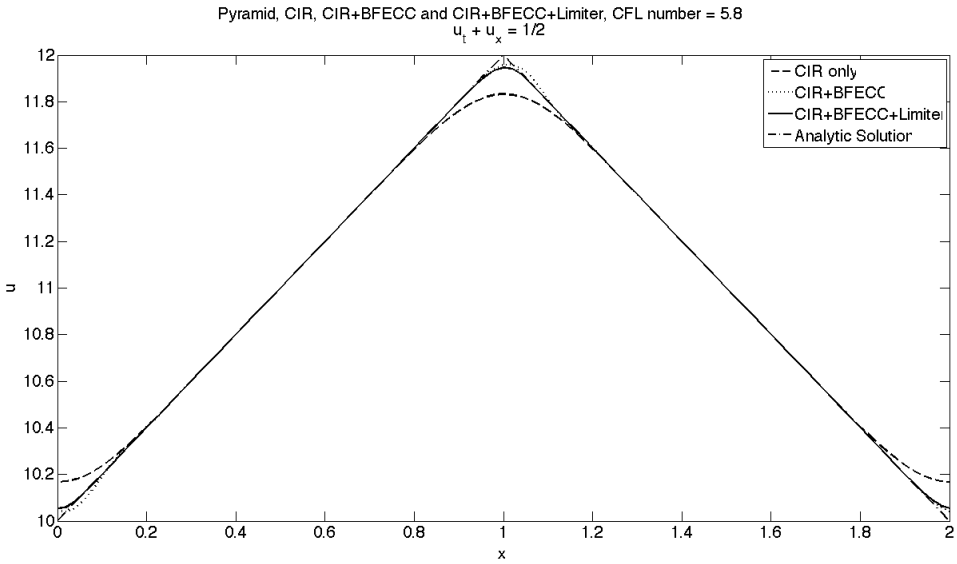

Figure 9: 1-D linear equation with nonzero right-hand side, CFL = 5.8 , pyramid initial condition, $h = 0.02$, $T = 20$.

We solve equation (27) numerically by using CIR scheme only, CIR+BFECC and CIR+BFECC+Limiting on a triangular mesh. The numerical results are demonstrated in Figure 10. It is easy to see that the CIR scheme alone has significant numerical diffusion while CIR+BFECC causes some overshoots near the edge of the

"cubic stair" due to the discontinuity of the solution. With the limiting algorithm, the spurious oscillations are eliminated without introducing extra numerical diffusion.

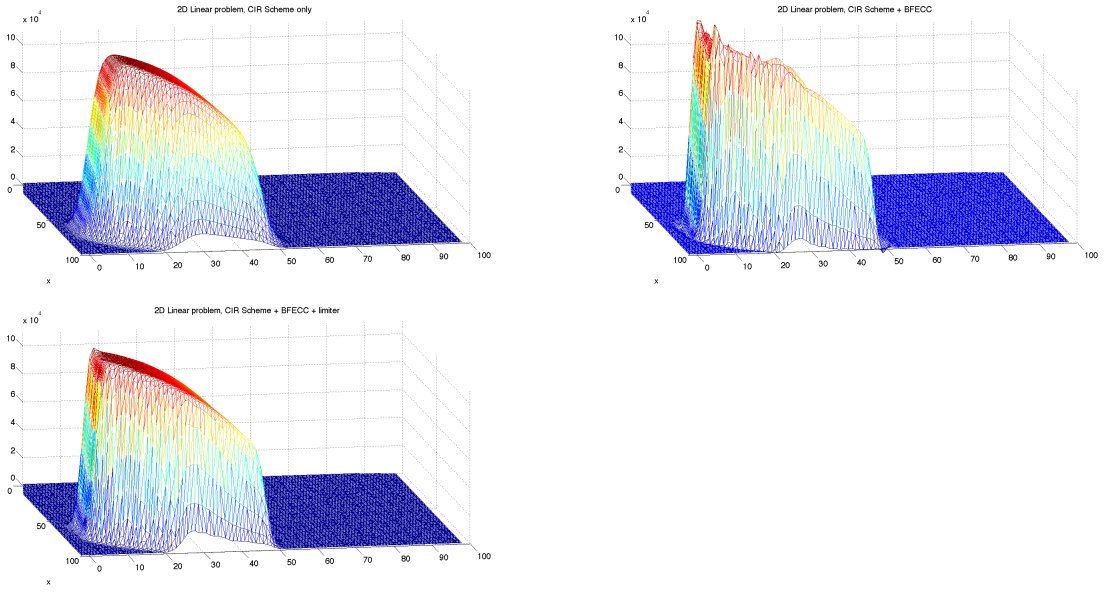

Figure 10: 2-D linear equation at $t = 157$. Uniform triangular mesh with $h = 1.0$ and $CFL = 3.0$. Top left: CIR only; Top right: CIR + BFECC; Bottom: CIR + BFECC + Limiting.

3.3.4 2D nonlinear Hamilton-Jacobi equations

The following 2D nonlinear Hamilton-Jacobi equation (see e.g. [26]) is commonly used in numerical tests. It is known that the exact solution has continuous derivatives everywhere when $T = 0.05$ while at $T = 0.15$ the solution is conituous but with discontinuous derivatives.

$$\begin{aligned} \phi_t + \frac{(\phi_x + \phi_y + 1)^2}{2} &= 0, \qquad (x, y) \in [-2, 2] \times [-2, 2] \\ \phi(x, y, 0) &= -\cos(\frac{\pi(x + y)}{2}) \,. \end{aligned} \tag{28}$$

This equation is computed on a triangular mesh by applying BFECC and limiting to a first order scheme with a Lax-Friedrichs-type monotone Hamiltonian. For more

details of the first order scheme with monotone Hamiltonian (L-F for short) developed by Abgrall, see [1]. The results are shown in Figure 11. We can see that BFECC without limiting is adequate for this problem and we find almost no difference when the limiting procedure is turned on.

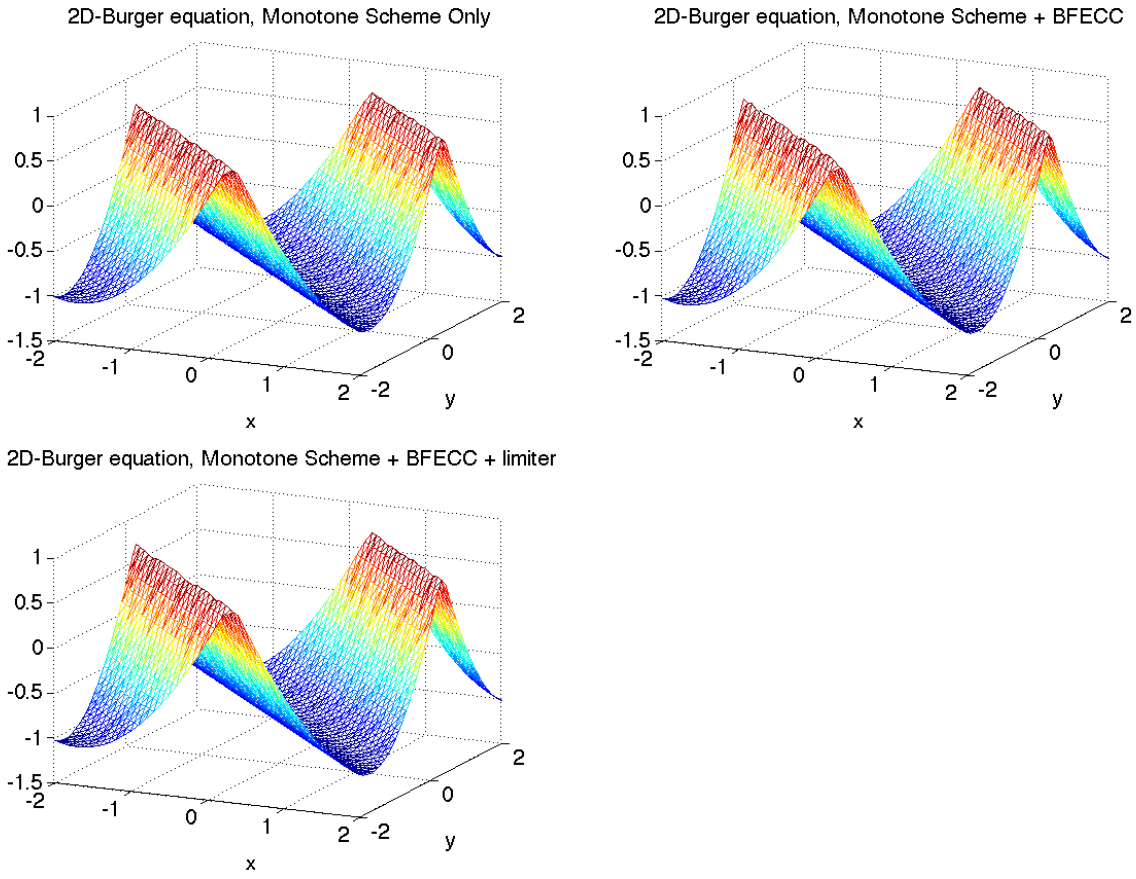

Figure 11: 2-D nonlinear Hamilton-Jacobi equation at $t = 0.15$, Uniform triangular mesh with $h = 0.1$. Top left: L-F only; Top right: L-F + BFECC; Bottom: L-F + BFECC + Limiter.

In addition, we tested the order of accuracy of L-F + BFECC + Limiting. The L^2-accuracy and L^∞-accuracy of this 2D nonlinear Hamilton-Jacobi equation are demonstrated in Table 5, Table 6, Table 7 and Table 8. We can see from the data that the limiting technique improves the accuracy at $T = 0.15$ when singularities of the solution have formed. At $T = 0.015$ when the solution is still smooth, the limiting procedure slightly reduces the L^∞-accuracy of the numerical solution compared to the

one without limiting.

Table 5: Numerical accuracy of L-F + BFECC + limiting. CFL number = 0.1; $T = 0.15$

number of points	L^2 error	order	L^∞ error	order
41×41	0.262	N/A	0.0451	N/A
81×81	0.0550	2.25	0.0135	1.73
161×161	0.0200	1.46	0.00536	1.33

Table 6: Numerical accuracy of L-F + BFECC. CFL number = 0.1; $T = 0.15$

number of points	L^2 error	order	L^∞ error	order
41×41	0.366	N/A	0.0649	N/A
81×81	0.101	1.86	0.0210	1.62
161×161	0.0370	1.45	0.00797	1.39

Table 7: Numerical accuracy of L-F + BFECC+limiting. CFL number = 0.1; $T = 0.015$

number of points	L^2 error	order	L^∞ error	order
41×41	0.289	N/A	0.0157	N/A
81×81	0.0976	1.57	0.00531	1.56
161×161	0.00631	3.95	0.00126	2.08

Table 8: Numerical accuracy of L-F + BFECC. CFL number = 0.1; $T = 0.015$

number of points	L^2 error	order	L^∞ error	order
41×41	0.288	N/A	0.0153	N/A
81×81	0.0968	1.57	0.00512	1.57
161×161	0.00638	3.92	0.000851	2.59

3.3.5 2D-Riemann Problem

We also test our algorithm in the following 2D Riemann Problem (see [26]).

$$\begin{aligned} \phi_t + sin(\phi_x + \phi_y) &= 0 \\ \phi(x, y, 0) &= \pi(|y| - |x|)\,. \end{aligned} \tag{29}$$

We compute the equation on a triangular mesh from $t = 0$ to $t = 1$ by applying BFECC and limiting to the first order scheme with monotone Hamiltonian [1]. The numerical results at time $t = 1$ are shown in Figure 12. Again we observe that BFECC without limiting is adequate for this problem and we find almost no difference when the limiting procedure is turned on.

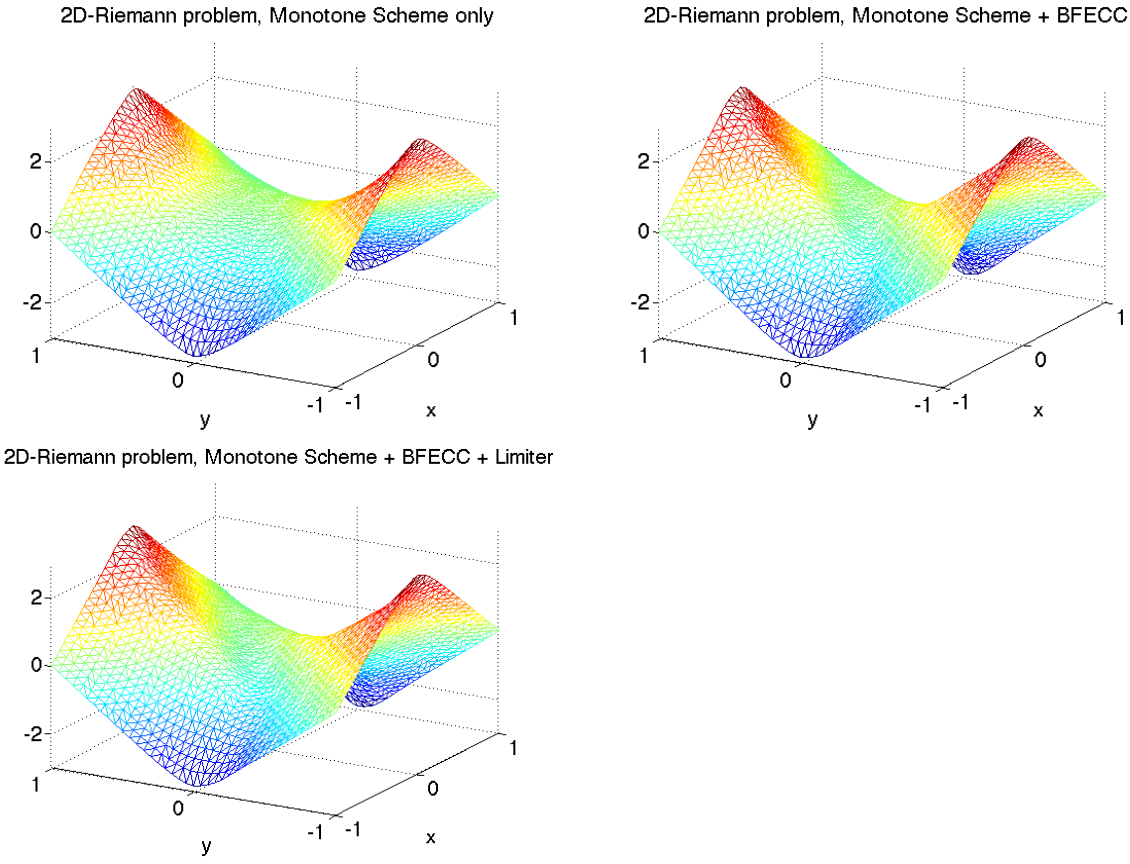

Figure 12: 2-D Riemann problem, Uniform triangular mesh with $h = 0.05$. Top left: L-F only; Top right: L-F + BFECC; Bottom: L-F + BFECC + Limiter.

3.3.6 Bubble Merging Problem

We study the merging of 4 bubbles (circles) centered at $(40, 60), (60, 60), (40, 40), (60, 40)$ with radius $9, 7, 10, 8$ respectively and expanding with constant normal velocity 0.2. The time evolution of these merging bubbles can be described by the level set method [25] with the level set function ϕ ($\phi < 0$ inside each circle) satisfying the following equation

$$\phi_t + 0.2\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla\phi = 0 \tag{30}$$

We compute the equation on a triangular mesh by applying BFECC and limiting to the first order scheme with monotone Hamiltonian [1]. The numerical results are compared in Figure 13. The top two bubbles (centered at $(40, 60)$ and $(60, 60)$) should have merged at the time $T = 11$. This is correctly captured with the limiting procedure. With BFECC and no limiting, the merging of the two bubbles has been delayed at $T = 11$. At the time $T = 26$, we can see that the smallest drop in the graph (bottom right) is kept when BFECC is used with the limiting, almost as well as without the limiting procedure (bottom middle graph).

3.3.7 Shrinking Square Problem

We consider the following problem: A square centered at $(0, 0)$ with side length 10 shrinks with the normal speed 0.2. Therefore at time $T = 10$ one should expect a 6×6 square. This problem can be described by equation (31), which is the same as the bubble merging problem from the previous subsection. The initial condition is an indicator function of $[-5, 5] \times [-5, 5] \subseteq \mathbb{R}^2$.

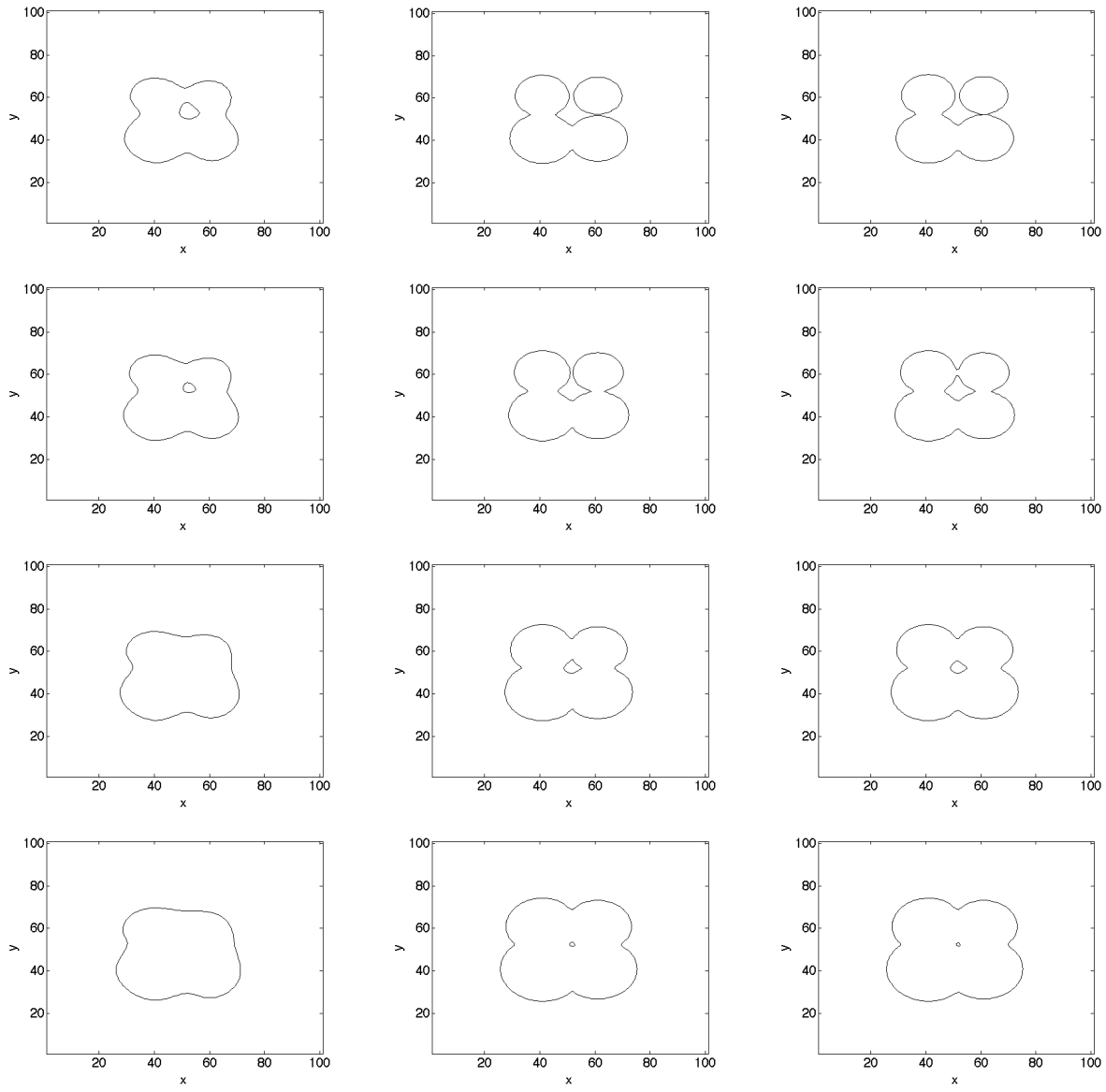

Figure 13: Expanding circles of radii $9, 7, 10$ and 8 with normal velocity 0.2. Uniform triangular mesh with $h = 1$, $\Delta t = 0.4$. Left: Lax-Friedrichs-type monotone Hamiltonian scheme (L-F); Middle: L-F + BFECC; Right: L-F + BFECC + Limiter. T = 9, 11, 18, 26 from top to bottom.

$$\phi_t + 0.2\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\phi = 0 \tag{31}$$

$$\phi(x, y, 0) = 1, \quad (x, y) \in [-5.5] \times [-5, 5] \tag{32}$$

$$\phi(x, y, 0) = 0, \quad (x, y) \notin [-5, 5] \times [-5, 5]$$

Our computations are based on the same first order Lax-Friderichs-type scheme with monotone Hamiltonian (L-F for short) as in Section 4.4. Equation (31) is computed on the triangular mesh. Numerical results obtained from three schemes, L-F scheme only, L-F scheme + BFECC and L-F scheme + BFECC + limiting, are compared in Figure 14. It can be observed that the L-F scheme leads to significant numerical diffusion. Such numerical diffusion can be reduced by BFECC. However, strong undershoots are generated by BFECC for this problem. With the help of the limiting algorithm, higher accuracy of BFECC can be preserved, while numerical artifacts are essentially eliminated.

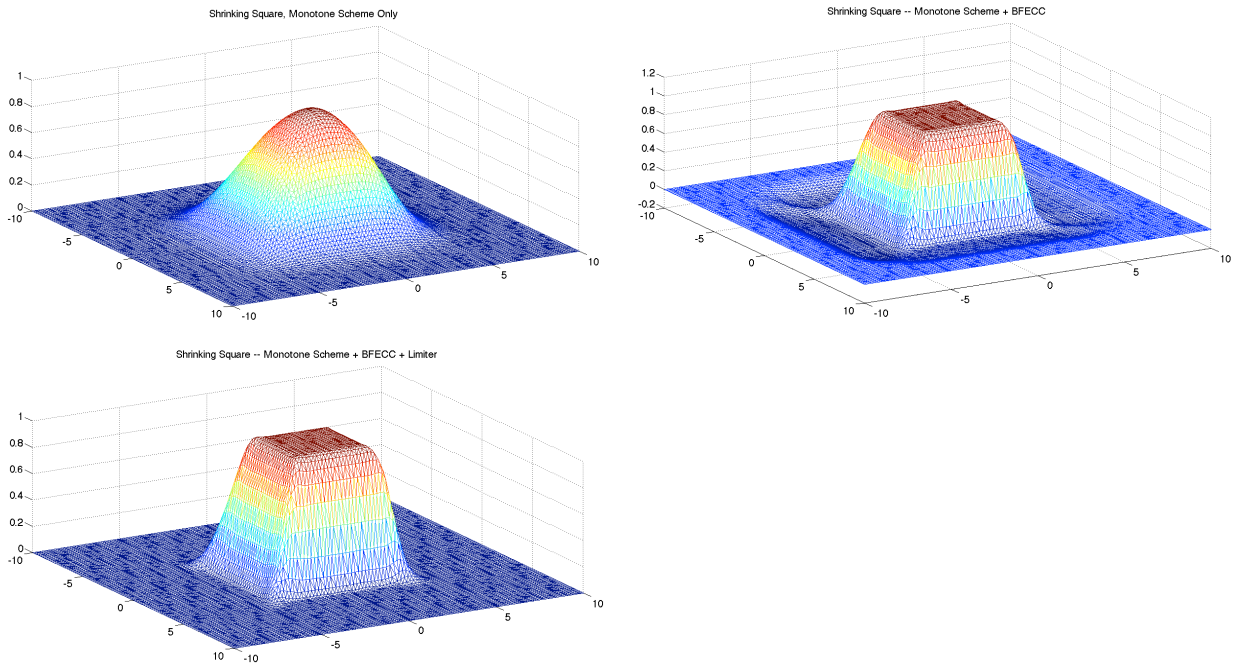

Figure 14: Shrinking Square Problem, Uniform triangular mesh with 101×101 mesh points. Top left: Monotone scheme (L-F) only; Top right: L-F + BFECC; Bottom: L-F + BFECC + Limiting.

CHAPTER IV

CONCLUSION AND FUTURE WORK

Three important extensions of the BFECC method are studied in this thesis: the time-reversible numerical schemes, a new algorithm for interpolation and the limiting strategy.

We first studied the time-reversibility of numerical schemes. A numerical scheme is said to be *time reversible* if it has zero back-and-forth error. This definition is inspired by the BFECC method. It is proved that time reversible schemes have higher order of accuracy. Moreover, by reducing back-and-forth error, the BFECC method can improve the time reversibility and produce time reversible numerical schemes. A variety of other time reversible numerical schemes are also proposed and investigated. In addition, we introduce a new interpolation algorithm based on BFECC method. It is based on linear interpolations and only uses information from adjacent grid points. Therefore, it is effective on unstructured meshes and could result in one more order of accuracy.

The last part of this thesis studies the limiting strategy. Inspired by the BFECC method, we perform one more backward advection in time. Therefore overshoots or undershoots at the new time level get exposed when they are transformed back to compare with the solution at the old time level. This new technique is very simple to implement even for unstructured meshes and is able to eliminate artifacts induced by jump discontinuities in the solution itself or in its derivatives. A series of linear and nonlinear examples in 1D and 2D are also demonstrated.

This thesis is the start point of our investigation. As our future work, we will continue our study in the aspects of both theory and application. On one hand,

there are a number of theoretical problems that are yet to be answered. For example, can we find some sufficient conditions on which the limiting strategy is effective (i.e. when does BFECC have undesired artificial effects)? What further improvement can be done for such limiting strategy? On the other hand, our numerical algorithms including BFECC, interpolation and limiting strategy can be applied in many scientific fields such as computer graphics, computational fluid dynamics and computational electromagnetics. In fact, these applications will be a very important component of our future projects.

We would like to apply our methods to the following three problems. First, many critical phenomena such as shock waves and turbulence can appear in computational fluid dynamics [38]. It is crucial to reduce the artifacts of numerical algorithms at discontinuities of solution itself or its derivatives. To attack this problem, a variety of new developed algorithms, including the hierarchical reconstruction introduced in [32, 33, 34, 35, 36], and the limiting strategy introduced in this thesis can be applied. We would like to apply our algorithm in the numerical simulations of water wave, air flow or smokes.

Second, we will apply the concept of reversible numerical schemes in computational electromagnetics. It is well known that finite-difference time-domain method like Yee's scheme [37] and its extensions can be applied to solve Maxwell's equation. In particular, Yee's scheme is time reversible on rectangular mesh. However, when the mesh is locally refined, Yee's scheme is challenged. In other words, Yee's scheme loses its reversibility on unstructured meshes or at the interface of two different-sized rectangular meshes. Therefore, it is tempting to apply some reversible or nearly reversible numerical schemes such as composed BFECC only at the interface of distinct sized meshes to preserve the high accuracy, while to run Yee's scheme within the rectangular meshes. Additionally, by using our limiting stragegy, we expect a fast, reversible high order numerical solver of Maxwell's equation on rectangular mesh with

locally refined grids.

Last but not least, we want to apply the BFECC interpolation algorithm in computer graphics. In computer graphics, it is a very common practice to use numerical interpolation to fill frames between the key frames. Therefore, instead of the auxiliary new local mesh, we have a natural selected new mesh grid. In this setting, using BFECC interpolation to estimate values of functions on new grid points all together would have significant lower cost than using higher order interpolation to approximate every grid points one by one.

REFERENCES

[1] R. Abgrall, Numerical discretization of the rst-order Hamilton-Jacobi equation on triangular meshes, *Comm. Pure Appl. Math.*, 49 (1996), 13391373.

[2] J. P. Boris and D. L. Book, Flux-Corrected Transport I. SHASTA, A Fluid Transport Algorithm That Works, *J. Comput. Phys.*, 11 (1973), 38–69.

[3] S. Bryson, A. Kurganov, D. Levy and G. Petrova, Semi-Discrete Central-Upwind Schemes with Reduced Dissipation for Hamilton-Jacobi Equations, *IMA J. Numer. Anal.*, 25 (2005), 113–138.

[4] S. Bryson and D. Levy, High-Order Semi-Discrete Central-Upwind Schemes for Multi-Dimensional Hamilton-Jacobi Equations, *J. Comp. Phys.*, 189 (2003), 63–87.

[5] Y. Chen, Q. Kang, Q. Cai and D. Zhang, Lattice Boltzmann method on quadtree grids, *Physical Review E*, 83 (2011).

[6] R. Courant and E. Isaacson and M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, *Comm. Pure Appl. Math.*, 5 (1952), 243–255.

[7] D. Enright, F. Losasso and R. Fedkiw, A Fast and Accurate Semi-Lagrangian Particle Level Set Method, *Computers and Structures*, 83 (2005), 479–490.

[8] M. Lentine, J. Gretarsson and R. Fedkiw, An Unconditionally Stable Fully Conservative Semi-Lagrangian Method, *J. Comp. Phys.* 230 (2011), 2857–2879.

[9] C.-T. Lin and E. Tadmor, High-resolution non-oscillatory central schemes for Hamilton- Jacobi Equations, *SIAM J. Sci. Comput.*, 21 (2000), 2163–2186.

[10] T. F. Dupont and Y.-J. Liu, Back and Forth Error Compensation and Correction Methods for Removing Errors Induced by Uneven Gradients of The Level Set Function, *J. Comput. Phys.*, 190 (2003), 311–324.

[11] T. F. Dupont and Y.-J. Liu, Back and Forth Error Compensation and Correction Methods for Semi-Lagrangian Schemes with Application to Interface Computation Using Level Set Method, *Tech. Rep. CDSNS2003-391*, School of Math., Georgia Inst. of Tech., 2004.

[12] T. F. Dupont and Y.-J. Liu, Back and Forth Error Compensation and Correction Methods for Semi-Lagrangian Schemes with Application to Level Set Interface Computations, *Math. Comp.*, 76 (2007), 647–668.

[13] I.V. Gugushvili and N. M. Evstigneev, Semi-Lagrangian Method for Advection Equation on GPU in Unstructured R^3 Mesh for Fluid Dynamics Application, *World Academy of Science, Engineering and Technology*, 60 (2009).

[14] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III, *J. Comput. Phys.*, 71 (1987), 231–303.

[15] F. Iavernaro, D. Trigiante, On some conservation properties of the trapezoidal method applied to Hamiltonian systems. ICNAAM 2005 proceedings, T.E.Simos, G.Psihoyios, Ch.Tsitouras (Eds.) . Wiley-VCH, Weinheim, 2005, pp. 254–257

[16] B.-M. Kim, Y.-J. Liu, I. Llamas and J. Rossignac, FlowFixer: Using BFECC for Fluid Simulation, *Eurographics Workshop on Natural Phenomena*, 2005.

[17] B.-M. Kim, Y.-J. Liu, I. Llamas and J. Rossignac, Advections with Significantly Reduced Dissipation and Diffusion, *IEEE Trans. Visual. and Comput. Graph.*, 13 (2007), 135–144.

[18] B.-M. Kim, Y.-J. Liu, I. Llamas, X.-M. Jiao and J. Rossignac, Simulation of Bubbles in Foam by Volume Control, *ACM SIGGRAPH* 2007.

[19] T. Kim and M. Carlson, A simple boiling module, *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 27–34.

[20] A. Kurganov, S. Noelle and G. Petrova, Semi-Discrete Central-Upwind Schemes for Hyperbolic Conservation Laws and Hamilton-Jacobi Equations, *SIAM J. Sci. Comput.*, 23 (2001), 707–740.

[21] A. Kurganov and G. Petrova, Adaptive Central-Upwind Schemes for Hamilton-Jacobi Equations with Nonconvex Hamiltonians, *J. Sci. Comput.*, 27 (2006), 323–333.

[22] A. Kurganov and E. Tadmor New High-Resolution Semi-Discrete Central Schemes for Hamilton-Jacobi Equations *J. Comput. Phys.*, 160 (2000), 720–742.

[23] P. D. Lax, On the Stability of Difference Approximations to Solutions of Hyperbolic Equations with Variable Coefficients, *Comm. Pure Appl. Math.*, 14 (1961), 497–520.

[24] B. van Leer, Toward the ultimate conservative dierence scheme: II. Monotonicity and conservation combined in a second order scheme, *J. Comput. Phys.*, 14 (1974), 361–370.

[25] S. Osher and J. Sethian, Fronts Propagating with Curvature-Dependent Speed: Algorithms based on Hamilton-Jacobi Equations, *J. Comput. Phys.*, 79 (1988), 12–49.

[26] S. Osher and C.-W. Shu, High-order essentially nonoscillatory schemes for HamiltonJacobi equations, *SIAM J. Numer. Anal.*, 28 (1991), 907922.

[27] A. Selle, R. Fedkiw, B.-M. Kim, Y.-J. Liu, and J. Rossignac, An Unconditionally Stable MacCormack Method, *J. Sci. Comput.*, 35 (2008), 350–371.

[28] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.*, 77 (1988), 439–471.

[29] J. Strain, Semi-Lagrangian methods for level set equations, *J. Comput. Phys.*, 151 (1999), 498–533.

[30] S. T. Zalesak, Fully Multidimensional Flux-Corrected Transport, *J. Comput. Phys.*, 31 (1979), 335–362.

[31] Y.-T. Zhang and C.-W. Shu, High order WENO schemes for Hamilton-Jacobi equations on triangular meshes, *SIAM J. Sci. Comput.*, 24 (2003), 1005–1030.

[32] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang, Central Discontinuous Galerkin Methods on Overlapping Cells with Non-Oscillatory Hierarchical Reconstruction *SIAM J. Numer. Anal.*, 45 (2007), 2442-2467

[33] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang, Non-Oscillatory Hierarchical Reconstruction for Central and Finite Volume Schemes *Comm. Comput. Phys.*, 2 (2007), 933-963

[34] Z.-L. Xu, Y.-J. Liu, C.-W. Shu, Hierarchical Reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO type linear reconstruction and partial neighboring cells *J. Comput. Phys.*, 228 (2009), 2194-2212

[35] Z.-L. Xu, Y.-J. Liu, C.-W. Shu, Hierarchical Reconstruction for Spectral Volume Method on Unstructured Grids *J. Comput. Phys.*, 228 (2009), 5787-5802

[36] Y.-J. Liu, C.-W. Shu, Z.-L. Xu, Hierarchical Reconstruction with up to Second Degree Remainder for Solving Nonlinear Conservation Laws, *Nonlinearity*, 22 (2009), 2799-2812

[37] K. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Transactions on Antennas and Propagation*, 14 (3) (1966), 302-307

[38] R. J. LeVeque, Numerical Methods for Conservation Laws, *Birkhäuser*