On the accuracy of a numerical method for two-dimensional scalar conservation laws based on dimensional splitting and front tracking

by

K. H. Karlsen
ON THE ACCURACY OF A NUMERICAL METHOD FOR TWO-DIMENSIONAL SCALAR CONSERVATION LAWS BASED ON DIMENSIONAL SPLITTING AND FRONT TRACKING

KENNETH HVISTENDAHL KARLSEN

ABSTRACT. A rigorous proof of an error estimate for a numerical method for two-dimensional scalar conservation laws is presented. The numerical method under consideration is based on the use of dimensional splitting and front tracking to solve the one-dimensional equations. It is shown that the error is bounded by $C((\Delta t)^{1/3} + (\Delta x)^{1/2} + \delta)$, where $\Delta x$ is the space step, $\Delta t$ is the time step, $\delta$ is the parameter measuring the polygonal approximation of the flux functions, and $C$ is a finite constant independent of the discretization parameters. Furthermore, we implement the method on a computer to supplement the theoretical results with numerical examples.

1. INTRODUCTION

Recently Holden and Risebro [6] proposed a new numerical method, based on dimensional splitting and front tracking, for the two-dimensional Cauchy problem

\[ u_t + f(u)_x + g(u)_y = 0, \quad x, y \in \mathbb{R} \]
\[ u(x, y, 0) = u^0(x, y), \quad x, y \in \mathbb{R}. \]

The purpose of this paper is to present an error analysis for this method. In [6] it is shown that the method produces a subsequence which converges to the unique physical solution solution of (1.1) as both the time and space discretization go to zero. However, no estimate of the error is provided. We obtain an explicit estimate showing at what rate the approximate solutions converge to the exact solution. More precisely we will prove that the convergence rate\(^1\) is no less than $1/2$. This result will be supplemented with numerical experiments. These experiments will indicate that the suggested estimate of $1/2$ is a reasonable result for a general problem. We will also make some observations concerning the CPU-time, and relate this to the actual error done by the method.

The error estimate is proved using some rather general approximation theory developed by Kuznetsov [9]. This theory has already been used to estimate the accuracy

\(^1\)Convention: If the error in some numerical method is of order $(\Delta)^\alpha$, where $\Delta$ is the discretization parameter, then $\alpha$ will be referred to as the "rate of convergence".
of a certain class of approximate methods including, in particular, the method of vanishing viscosity, the method of smoothing, and several widely used finite difference methods, cf. [9]. For instance, error estimates for the methods of Glimm, Godunov and LeVeque was obtained by Lucier [10] using this technique.

It is well known that solutions to (1.1) typically develop discontinuities as $t$ increases to some $t_0$. Thus the differential equation must be understood in a weak sense. However, weak solutions are not unique, and additional principles, so-called entropy conditions, is needed to select the unique “physical” weak solution. We will use Kružkov’s formulation of the entropy condition. With a solution to (1.1) we understand a bounded, measurable function which satisfies Kružkov definition of the entropy weak solution.

**Definition 1.1 (Kružkov [8]).** $u$ is the entropy weak solution of (1.1) if for all constants $a$, all $\phi \in C^1_0(\mathbb{R}^2 \times \mathbb{R}^+)$, $\phi \geq 0$, the inequality

$$
\int_{\mathbb{R}^2} \int_0^T \left\{ |u - a| \phi_t + \text{sign}(u - a)(f(u) - f(a))\phi_x + (g(u) - g(a))\phi_y \right\} dt \, dx \, dy + \int_{\mathbb{R}^2} |u^0 - a| \phi(x, y, 0) \, dx \, dy \geq 0.
$$

(1.2)

holds.

Dimensional splitting is a popular technique used to extend one-dimensional numerical methods to multidimensional problems. This technique was first introduced by Godunov in connection with gas dynamics, and has since been extended by various authors. Loosely speaking, we may say that the idea is simply to alternate between solving one-dimensional (1D) problems in $x$ and $y$-direction. To describe the method of dimensional splitting in more detail, we have to introduce some notation. From now on $S(t)$ will denote the solution operator which takes an initial function $u^0$ on $\mathbb{R}^2$ to the entropy weak solution of (1.1) at time $t$, i.e., $u(x, y, t) = S(t)u^0(x, y)$. Furthermore, let $S^{f,x}(t)$ and $S^{g,y}(t)$ denote the solution operators which take initial functions $v^0$ and $w^0$ on $\mathbb{R}$ to the entropy weak solutions of

$$
v_t + f(v)_x = 0
$$

$$
v(x, 0) = v^0(x)
$$

and

$$
w_t + g(w)_y = 0
$$

$$
w(y, 0) = w^0(y)
$$

at time $t$, respectively, i.e., $v(x, t) = S^{f,x}(t)v^0(x)$ and $w(y, t) = S^{g,y}(t)w^0(y)$. The idea behind dimensional splitting is to alternately apply the one-dimensional solution operators $S^{f,x}$ and $S^{g,y}$ for small time steps $\Delta t$ to approximate $S(t)u^0$. The dimensional splitting solution $\{u_{\Delta t}^n\}_{n=1}^N$, where $n\Delta t = t_n$, $N\Delta t = T < T_0$, for some fixed $T_0$, is defined by

$$
u_{\Delta t}^n(x, y) = [S^{g,y}(\Delta t)S^{f,x}(\Delta t)]^n u^0(x, y).
$$

(1.3)
ON THE ACCURACY OF A NUMERICAL METHOD FOR TWO-DIMENSIONAL ... 3

Observe that the solution is only defined for discrete \( t \)-values, \( t = t_n, \ n = 1, \ldots, N \).

In order to use the splitting (1.3) in a numerical method we simply replace the exact one-dimensional solution operators \( S_{x}^{1} \) and \( S_{y}^{2} \) by numerical methods (approximate 1D solution operators). When solving one-dimensional problems we may choose form a variety of methods. Crandall and Majda [3] analyze the method of dimensional splitting for monotone schemes, the Glimm method and the Lax-Wendroff scheme. In this paper we will analyze the method of dimensional splitting for the front tracking scheme. This method, first proposed by Holden and Risebro [6], has the advantage of yielding an unconditionally stable approximation in the sense that the time step is not limited by the space step, i.e., one does not need a CFL condition.

The method of front tracking is based on the idea of replacing the flux function with a piecewise linear approximation, the initial data with a piecewise constant approximation, and to solve the resulting perturbed problem exactly. This idea was first suggested by Dafermos [4], and later developed into a numerical method for one-dimensional scalar conservation laws by Holden, Holden, and Høegh-Krohn [5].

In a recent article by Teng [14], it is shown that the splitting (1.3) has a convergence rate of 1/2. We show that the numerical method obtained by using the splitting (1.3) with the exact solution operators \( S_{x}^{1} \) and \( S_{y}^{2} \) replaced by the front tracking scheme has the same convergence rate. The argument needed to conclude with a convergence rate of 1/2 for our method will, in the case of exact 1D solution operators instead of the front tracking scheme, reduce to give a convergence rate of 1/2 for the splitting (1.3).

The rest of this paper is divided into four main parts. In §2 we explain in detail how the family of approximate solutions is constructed. Here, we also describe the front tracking method in more detail. Next, in §3 we introduce Kuznetsov's approximation theory and we use this theory to state and prove an error estimate for our numerical method. In §4 we present the numerical experiments, and, finally, in §5 we end the paper by making some concluding remarks.

2. CONSTRUCTION OF APPROXIMATE SOLUTIONS

From now on we will refer to the method of dimensional splitting together with front tracking to solve the one-dimensional equations simply as the method of dimensional splitting. The splitting (1.3) will be referred to as the semi-discrete method.

Most of the material contained in this section may be found in [6], and is included for the sake of completeness. Before we continue let us introduce some basic mathematical notation which will be used extensively throughout this paper. If \( K \) is a domain in \( \mathbb{R}^2 \), then \( L^p(K) \), \( 1 \leq p \leq \infty \) will denote the classical \( L^p \) spaces of real valued functions on \( K \), and the norms on \( L^p(K) \) are denoted by \( \| \cdot \|_p \). The localized versions of \( L^p \), consisting of functions on \( \mathbb{R}^2 \) which are in \( L^p(K) \) for any compact subset \( K \) of \( \mathbb{R}^2 \), are denoted by \( L^p_{\text{loc}}(\mathbb{R}^2) \). Furthermore, \( BV(\mathbb{R}^2) \) denotes the subspace of \( L^p_{\text{loc}}(\mathbb{R}^2) \) consisting of functions with bounded variation, i.e.,

\[
BV(\mathbb{R}^2) = \{ v \in L^p_{\text{loc}}(\mathbb{R}^2) : T.V.(x,y)(v) < \infty \},
\]
where
\[(2.1) \quad T.V.(x,y)(v) = \int_{\mathbb{R}} T.V.x(v,\cdot)\,dy + \int_{\mathbb{R}} T.V.y(v(x,\cdot))\,dx.\]
If \(v = v(x)\) is a function on \(\mathbb{R}\), then \(T.V.x(v)\) is defined as
\[T.V.x(v) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \int_{\mathbb{R}} |h(x) - h(x - \varepsilon)|\,dx.\]
The class of Lipschitz continuous functions on a domain \(K \subset \mathbb{R}\) is denoted by \(\text{Lip}(K)\). More precisely
\[\text{Lip}(K) = \{v \in L^\infty(K) : \|v\|_{\text{Lip}} < \infty\},\]
where
\[\|v\|_{\text{Lip}} = \sup_{x \neq y} \frac{|v(x) - v(y)|}{|x - y|}.\]
We denote by \(C^k(K)\) the space of all real valued functions on \(K\) possessing continuous (partial) derivatives of order \(\leq k\), and we set \(C^\infty(K) = \bigcap_1^{\infty} C^k(K)\). The usual \(O\)-notation will also be used frequently throughout this paper. We recall that if a function \(v = v(x)\) on \(\mathbb{R}^+\) satisfies
\[|v(x)| \leq Cx^\alpha, \quad x \leq x_0\]
for some positive constants \(C, \alpha\) and \(x_0\), then we say that \(v\) is of order \(x^\alpha\) as \(x \to 0^+\) and write \(v(x) = O(x^\alpha)\). Left and right-hand limits are denoted by \(v(x-)\) and \(v(x+)\), respectively.

Let \(\Delta x\) and \(\Delta y\) be given small positive numbers and define a grid \(\{(i \Delta x, j \Delta y)\}_{i,j \in \mathbb{Z}}\) in the \(x-y\) plane. Furthermore, let \(\delta > 0\) denote the parameter measuring the polygonal approximation of the flux functions. Let \(\pi\) be the projection operator, from \(BV(\mathbb{R}^2)\) to functions that are constant on each square
\[z_{i,j} = \{(x, y) : i \Delta x < x < (i + 1) \Delta x, j \Delta y < y < (j + 1) \Delta y\}\]
for \(i, j \in \mathbb{Z}\), defined by
\[\pi u(x, y) = \frac{1}{\Delta x \Delta y} \int_{z_{i,j}} u(\tilde{x}, \tilde{y})\,d\tilde{x}\,d\tilde{y}, \quad \forall (x, y) \in z_{i,j}.\]
\(\pi u(x, y)\) is simply the grid average of \(u\) over the grid block \(z_{i,j}\), and it is easy to see that the error caused by the projection operator is of order \(\max(\Delta x, \Delta y)\), i.e., \(\|u - \pi u\|_1 = O(\max(\Delta x, \Delta y))\). Since we will use front tracking to solve the one-dimensional equations, we define \(f_\delta\) and \(g_\delta\) to be piecewise linear and continuous approximations to \(f\) and \(g\), respectively. Let \(u_0, \delta\) be some given real numbers with \(\delta > 0\), and let \(u_i = u_0 + i\delta, f_i = f(u_i)\) for \(i = 1, \ldots, N\). We define \(f_\delta\) by
\[u \in [u_i, u_{i+1}] \Rightarrow f_\delta(u) = \frac{f_{i+1} - f_i}{u_{i+1} - u_i}(u - u_i) + f_i, \quad i = 0, \ldots, N - 1,\]
and
\[u \leq u_0 \Rightarrow f_\delta(u) = f_0, \quad u \geq u_N \Rightarrow f_\delta(u) = f_N.\]
ON THE ACCURACY OF A NUMERICAL METHOD FOR TWO-DIMENSIONAL ...

We define $g_\delta$ in a similar manner. With this notation in hand we can now give a brief description of the front tracking method as we are going to use it to solve the one-dimensional equations. Consider the Riemann problem with $u_L = u_0$ and $u_R = u_N$, let $f_{\delta,c}$ denote the lower convex envelope of $f_\delta$. Since $f_\delta$ is piecewise linear so is $f_{\delta,c}$. Let $\bar{u}_0 < \bar{u}_1 < \cdots < \bar{u}_N$ be such that

$$\bar{u}_0 = u_0, \quad \bar{u}_M = u_N \quad \{\bar{u}_0, \ldots, \bar{u}_M\} \subseteq \{u_0, \ldots, u_N\},$$

and such that $f_{\delta,c}$ is linear in each interval $[\bar{u}_i, \bar{u}_{i+1}]$. The solution of the Riemann problem with left state $u_L = u_0$ and right state $u_R = u_N$ is then given by

$$(2.2) \quad u(x, t) = \begin{cases} 
  u_L & x \leq \bar{s}_0t \\
  \bar{u}_i & \bar{s}_i t < x \leq \bar{s}_{i+1} t, \quad i = 1, \ldots, N - 1 \\
  u_R & x > \bar{s}_{N-1} t.
\end{cases}$$

where

$$\bar{s}_i = \frac{\bar{f}_{i+1} - \bar{f}_i}{\bar{u}_{i+1} - \bar{u}_i} = f'_{\delta,c}(u_i+), \quad i = 0, \ldots, N - 1$$

and $\bar{f}_i = f(\bar{u}_i)$. When $u_L > u_R$ there is an similar formula involving the upper convex envelope. The front tracking method consists of replacing the flux function $f$ with $f_\delta$ and the initial data with some piecewise constant function, and to solve the resulting perturbed problem exactly. This solution may be constructed as follows. Observe that each jump in the initial data defines a Riemann problem. The solution of these problems leads to a series of discontinuities propagating in the $x - t$ plane. By “Gluing” together the solutions of the Riemann problems we have the solution until, at some point, two or more of these discontinuities interact, and we have what is called a shock collision. When two or more neighboring discontinuities collide they define a new Riemann problem with left and right states the given by the values immediately to the left and the right of the collision. This Riemann problem is solved, and we have the solution until the next shock collision occurs. This collision is of course handled in the same way, and so are also the succeeding ones. The method for constructing the global solution to the perturbed problem may briefly be summarized as follows:

1. Solve the Riemann problems defined by the piecewise constant initial data.
2. Keep track of the shock collisions, and solve the Riemann problems arising at the collision points.

From the construction it is easily seen that the solution $v$ will be piecewise constant. For a more detailed treatment of the front tracking method we refer the reader to Holden, Holden, and Høegh-Krohn [5].

Let $S_t(t)$ denote the solution operator which takes an initial function $u^0$ on $\mathbb{R}^2$ to the entropy weak solution of

$$(2.3) \quad u_t + f_s(u)_x + g_s(u)_y = 0, \quad x, y \in \mathbb{R}$$

and

$$u(x, y, 0) = u^0(x, y), \quad x, y \in \mathbb{R}.$$
Furthermore, let $S_{f}^{\pi_x}(t)$ and $S_{g}^{\pi_y}(t)$ denote the solution operators of

$$v_t + f_{\delta}(v)_x = 0$$

and

$$w_t + g_{\delta}(w)_y = 0,$$

respectively, i.e., $v(\cdot, t) = S_{f}^{\pi_x}(t)v_0$ and $w(\cdot, t) = S_{g}^{\pi_y}(t)w_0$ are the entropy weak solutions of (2.4) and (2.5) with initial data $v_0$ and $w_0$, respectively. If $v_0$ is piecewise constant taking a finite number of values, then $S_{f}^{\pi_x}(t)v_0$ is identical with the solution generated by the front tracking scheme since $f_{\delta}$ is piecewise linear. The dimensional splitting solution $\{u_{n}^{0}\}_{n=1}^{N}$, where $n\Delta t = t_n$, $N\Delta t = T < T_0$, for some fixed $T_0$ and $\eta = (\Delta x, \Delta y, \Delta t)$ is defined by

$$u_{n}^{0}(x, y) = [\pi S_{g}^{\pi_y}(\Delta t)\pi S_{f}^{\pi_x}(\Delta t)]^{n}u_{0}(x, y),$$

where $u_{0}(x, y) = \pi u_0(x, y)$. We may read (2.6) as follows. The method of dimensional splitting is to use the front tracking scheme in the $x$-direction for a small time step $\Delta t$. The front tracking solution will not necessarily be piecewise constant on the original grid and the solution is therefore projected back onto this grid, before we apply the front tracking scheme in the $y$-direction for a time step $\Delta t$, using the (projected) solution computed in the $x$-direction as initial data. This process is then repeated a finite number of times to yield the approximate solution. Each time after we apply the front tracking scheme, we project the solution back onto the original grid, and thereby obtaining a sequence of functions indexed by the number of iterations $n$, (the time step $\Delta t$) and the mesh size $(\Delta x, \Delta y)$.

For an algorithmic presentation of the method of dimensional splitting we refer the reader to Holden and Risebro [6] or Hvidstenaholm Karlsen [7].

3. AN ERROR ANALYSIS AND EXPLICIT ERROR ESTIMATES

In this section we will state and rigorously prove an error estimate for the method of dimensional splitting. It will be shown that the method produces approximate solutions which converges to the entropy weak solution with a rate of convergence which is no less than 1/2. The argument leading to this estimate is based on the approximation theory developed by Kuznetsov [9]. We will therefore start by briefly introducing this theory. Observe that in the analysis of the speed of convergence we have to consider the approximate solutions as (continuous) functions of $t$, and not merely defined at the discrete $t$-values $t = n\Delta t$. With $\eta = (\Delta x, \Delta y, \Delta t)$, we let $\{u_{\eta}\}$, given by

$$u_{\eta}(x, y, t) = \begin{cases} S_{f}^{\pi_x}(2(t - n\Delta t))u_{n}^{0}(x, y) & \text{for all } t \in [n\Delta t, (n + \frac{1}{2})\Delta t) \\ S_{g}^{\pi_y}(2(t - (n + \frac{1}{2})\eta))u_{n+\frac{1}{2}}^{0}(x, y) & \text{for all } t \in [(n + \frac{1}{2})\Delta t, (n + 1)\Delta t), \end{cases}$$
where we recall that
\[
\begin{align*}
    u^n_n(x, y) & = [S^y_\delta^z(y) S^x_\delta^z(x)]^n u^0(x, y) \\
u^{n+\frac{1}{2}}_n(x, y) & = S^x_\delta^z(x) [S^y_\delta^z(y) S^x_\delta^z(x)]^n u^0(x, y),
\end{align*}
\]
be our family of approximate solutions. It can easily be seen that \( u_n \) coincide with \( u^n \) when \( t = t_n, n = 0, 1, \ldots, N \).

We will consider bounded weak solutions of (1.1) in the half-space \( \mathbb{R}^2 \times \mathbb{R}^+ \). We select from the set of equivalent solutions those that are continuous in \( t \) in the sense of \( L^1(\mathbb{R}^2) \) at each point of the half line \( t \geq 0 \). We denote by \( \mathcal{U} \) the space of elements \( u : \mathbb{R}^+ \to L^1(\mathbb{R}^2) \) for which the following five conditions hold:

1. the limit values \( u(t \pm) \) exist
2. \( u \) is right continuous
3. \( \|u(\cdot, t)\|_\infty \leq \|u^0\|_\infty \)
4. \( T.V. \cdot (x, y)(u(\cdot, \cdot, t)) \leq T.V. \cdot (x, y)(u^0) \)
5. \( \|u(\cdot, t_2) - u(\cdot, t_1)\|_1 \leq C(t_2 - t_1) \), for some finite constant \( C \).

Observe that both \( u_n \) and the exact solution of (1.1) is an element of \( \mathcal{U} \). For a proof of the fact that \( u_n \) satisfies the estimates (3-5), we refer the reader to [6]. We introduce a non-negative function \( \omega \in C^0_0 \), satisfying \( \omega(\sigma) = \omega(-\sigma), \omega(\sigma) \equiv 0 \) for \( |\sigma| \geq 1 \) and \( \int_{\mathbb{R}} \omega(\sigma) d\sigma = 1 \). For \( \varepsilon > 0 \) let
\[
\omega_\varepsilon(\sigma) = \frac{1}{\varepsilon} \omega\left(\frac{1}{\varepsilon}\right)
\]
then

- (a) \( \omega_\varepsilon \in C_0^\infty, \omega(\sigma) \geq 0 \ \forall \sigma \in \mathbb{R} \)
- (b) \( \int_{\mathbb{R}} \omega_\varepsilon(\sigma) d\sigma = 1 \)
- (c) \( \omega_\varepsilon(\sigma) \equiv 0 \) for \( |\sigma| \geq \varepsilon \)
- (d) \( \omega_\varepsilon(\sigma) \leq M_\omega/\varepsilon, \ |\omega_\varepsilon'(\sigma)| \leq M_\omega/\varepsilon^2 \),

where \( M_\omega \) is a finite constant independent \( \varepsilon \). For \( \varepsilon > 0, \varepsilon_0 > 0 \) we define the non-negative test function
\[
\Omega_{\varepsilon, \varepsilon_0} = \Omega_{\varepsilon, \varepsilon_0}(x - x', y - y', t - t') = \omega_\varepsilon(x - x')\omega_\varepsilon(y - y')\omega_{\varepsilon_0}(t - t').
\]

Furthermore, we introduce with \( u, v \in \mathcal{U}, \phi \in C^0_0(\mathbb{R}^2 \times \mathbb{R}^+), \phi \geq 0, \) and \( a \in \mathbb{R} \) the notation:

\[
\begin{align*}
    \mathcal{D}_{T,(f, \phi)}(\phi, u, a) & = \int_{\mathbb{R}^2} \int_{0}^{T} \left\{ \phi_t u - a + \phi_x F_1(u, a) + \phi_y F_2(u, a) \right\} dt \ dx \ dy \\
    & + \int_{\mathbb{R}^2} \phi(x, y, 0)|u^0(x, y)| - a \ dx \ dy \\
    & - \int_{\mathbb{R}^2} \phi(x, y, T)|u(x, y, T - 0)| - a \ dx \ dy, \ \text{and}
\end{align*}
\]

\[
\begin{align*}
    \mathcal{D}_{T,(f, \phi)}(\Omega_{\varepsilon, \varepsilon_0}, v, u) & = \int_{\mathbb{R}^2} \int_{0}^{T} \mathcal{D}_{T,(f, \phi)}(\Omega_{\varepsilon, \varepsilon_0}(x - x', y - y', t - t'), v, u(x', y', t') \ dt' \ dx' \ dy',
\end{align*}
\]
where we have for notational convenience introduced the auxiliary "flux–functions"

\[ F_1(u, a) = \text{sign}(u - a)(f(u) - f(a)) \]
\[ F_2(u, a) = \text{sign}(u - a)(g(u) - g(a)). \]

Since there will be no ambiguity, the \((f, g)\)–subscript on \(D_{T, f,g}^{\epsilon, \eta_0}\) and \(D_{T, f,g}^{\epsilon, \eta_0}\) will be dropped. Both the flux functions are assumed to be Lipschitz continuous and we let \(A = \max(||f||_{L^p}, ||g||_{L^p}).\)

**Theorem 3.1 (Kuznetsov).** If \(u\) is the exact solution of (1.1) and \(v \in U\), then for \(0 < \epsilon_0 < T, \epsilon > 0\), we have

\[ ||u(\cdot, \cdot, T) - v(\cdot, \cdot, T^-)||_1 \leq ||u^0 - v^0||_1 + 2(\epsilon + A\epsilon_0)T.V(\tau, x, y)(u^0) - D_{T, f,g}^{\epsilon, \eta_0}(\Omega, \eta_0, v, u). \]

For a proof of this result see [9]. If \(\{u_{\tau}\} \subset U\) is a sequence of functions for which \(D_{T, f,g}^{\epsilon, \eta_0}(u_{\tau}) \rightarrow 0\) as \(\tau \rightarrow 0\) for fixed \(\epsilon\) and \(\epsilon_0\), then, by making use of the arbitrariness in the choice of \(\epsilon\) and \(\epsilon_0\), it is possible to estimate in general the rate of convergence to zero of \(||u - u_{\tau}||_1\). A major step towards showing that the method of dimensional splitting has a convergence rate of 1/2 when applied to problem (1.1), is first to show this when we apply the method to problem (2.3). In particular, we shall prove the following proposition:

**Proposition 3.2.** Assume that \(u^0 \in L^1(R^2) \cap L^\infty(R^2) \cap BV(R^2)\). If \(N\Delta t = T < T_0\) for some fixed \(T_0\) and \(\Delta x = \Delta y = O(\Delta t)\), then as \(\eta = (\Delta x, \Delta y, \Delta t) \rightarrow 0+\) and \(N \rightarrow \infty,\)

\[ ||S_{\eta}(T)u^0(\cdot, \cdot) - u^N_{\eta}(\cdot, \cdot)||_1 \leq K((\Delta t)^{1/2} + (\Delta x)^{1/2}), \]

where \(K\) is a finite constant independent of \(\eta\), but dependent on \(T_0, T.V(\tau, x, y)(u^0), ||f'||_\infty\) and \(||g'||_\infty.\)

**Proof.** The proposition follows if we manage to find a proper bound on \(D_{T, f,g}^{\epsilon, \eta_0}(\Omega, \eta_0, u_{\tau}, u)\) in terms of \(\Delta x, \Delta t, \epsilon, \) and \(\epsilon_0\). We start by establishing a bound on \(D_{T}(f, \phi, \eta_0, u)\) for arbitrary test functions \(\phi\) and constants \(a\): Assume \(v^0 \in L^1(R^2) \cap L^\infty(R^2) \cap BV(R^2)\) and \(w^0 \in L^1(R^2) \cap L^\infty(R^2) \cap BV(R^2)\), and let \(v(x, y, t) \equiv S_{t, x}(t)v^0(\cdot, y)\) and \(w(x, y, t) \equiv S_{t, y}(t)w^0(x, \cdot)\) denote the corresponding entropy weak solutions of the one–dimensional conservation laws. Under the above assumptions we have the following (cf. [3]): If \(\phi \in C_0^1(R^2 \times R^+), \phi \geq 0\) and \(t_2 \geq t_1 \geq 0,\) then

\[ \int_{R^2} \int_{t_1}^{t_2} |v - a| \phi_t + F_1(v, a) \phi_x dt dx dy \geq \int_{R^2} |v(x, y, t_2) - a| \phi(x, y, t_2) dx dy \]
(3.1)

\[ \int_{R^2} \int_{t_1}^{t_2} |w - a| \phi_t + F_2(w, a) \phi_y dt dx dy \geq \int_{R^2} |w(x, y, t_2) - a| \phi(x, y, t_2) dx dy \]
(3.2)
ON THE ACCURACY OF A NUMERICAL METHOD FOR TWO-DIMENSIONAL

Let us now consider a new test function $\hat{\phi}$ defined by $\hat{\phi}(x, y, t) = \phi(x, y, t/2)$. Furthermore, define $v_n(t) \equiv S^{f, x}(t)[S^{g, x}(\Delta t)S^{f, x}(\Delta t)]^n u^0$ and $w_n(t) \equiv S^{g, y}(t)S^{f, x}(\Delta t)[S^{g, y}(\Delta t)S^{f, x}(\Delta t)]^n u^0$, and apply (3.1) and (3.2) to $v_n(t)$ and $w_n(t)$, respectively, to obtain

$$\int_{\mathbb{R}^2} \int^{(n+\frac{1}{2})\Delta t}_{n\Delta t} \frac{1}{2} \phi_t |u_n - a| + \phi_x F_1(U_n, a) \, dt \, dx \, dy \equiv$$

$$\frac{1}{2} \int_{\mathbb{R}^2} \int^{\Delta t}_0 \phi(x, y, \tau + 2n\Delta t)|v_n(\tau) - a| + \phi_x(x, y, \tau + 2n\Delta t)F_1(v_n(\tau), a) \, d\tau \, dx \, dy$$

$$\geq \frac{1}{2} \int_{\mathbb{R}^2} \phi(x, y, (n + \frac{1}{2})\Delta t)|u_n(x, y, (n + \frac{1}{2})\Delta t) - a| \, dx \, dy$$

(3.3)

$$-\frac{1}{2} \int_{\mathbb{R}^2} \phi(x, y, n\Delta t)|u_n(x, y, n\Delta t) - a| \, dx \, dy$$

and similarly

$$\int_{\mathbb{R}^2} \int^{(n+\frac{1}{2})\Delta t}_{(n+1)\Delta t} \frac{1}{2} \phi_x |u_n - a| + \phi_y F_2(u_n, a) \, dt \, dx \, dy$$

$$\geq \frac{1}{2} \int_{\mathbb{R}^2} \phi(x, y, (n + 1)\Delta t)|u_n(x, y, (n + 1)\Delta t) - a| \, dx \, dy$$

(3.4)

$$-\frac{1}{2} \int_{\mathbb{R}^2} \phi(x, y, (n + 1)\Delta t)|u_n(x, y, (n + 1)\Delta t) - a| \, dx \, dy.$$

We add the inequalities (3.3) and (3.4), and sum over $n$ (having in mind that $u_n((n + \frac{1}{2})\Delta t -)$ and $u_n((n + \frac{1}{2})\Delta t +)$ are not equal to $u_n(n\Delta t +)$ and $u_n((n + \frac{1}{2})\Delta t +)$, respectively). If we let $\chi_n$ denote the characteristic function of $\{(x, y, t) : n\Delta t \leq t \leq (n + \frac{1}{2})\Delta t\}$, then the resulting inequality\(^2\) may be rewritten in the following form:

$$-D_T(\phi, u_n, a) \leq I_1 + I_2 + I_3 + I_4$$

where $I_i = I_i(\phi, u_n, a)$, $i = 1, 2, 3, 4$ is defined by

$$I_1 = \sum_{n=0}^{N-1} \int_{\mathbb{R}^2} \int^{(n+\frac{1}{2})\Delta t}_{n\Delta t} 2\chi_n F_1(u_n, a) \phi_x \, dt \, dx \, dy - \int_{\mathbb{R}^2} \int^{T}_{0} F_1(u_n, a) \phi_x \, dt \, dx \, dy$$

$$I_2 = \sum_{n=0}^{N-1} \int_{\mathbb{R}^2} \int^{(n+\frac{1}{2})\Delta t}_{n\Delta t} 2(1 - \chi_n) F_2(u_n, a) \phi_y \, dt \, dx \, dy - \int_{\mathbb{R}^2} \int^{T}_{0} F_2(u_n, a) \phi_y \, dt \, dx \, dy$$

$$I_3 = -\sum_{n=1}^{N-1} \int_{\mathbb{R}^2} \phi(x, y, n\Delta t)(|u_n(n\Delta t -) - a| - |u_n(n\Delta t +) - a|) \, dx \, dy$$

$$I_4 = -\sum_{n=0}^{N-1} \int_{\mathbb{R}^2} \phi(x, y, (n + \frac{1}{2})\Delta t)(|u_n((n + \frac{1}{2})\Delta t -) - a| - |u_n((n + \frac{1}{2})\Delta t +) - a|) \, dx \, dy.$$

For $i = 1, \ldots, 4$, define $I_i^{e, \epsilon_0} = I_i^{e, \epsilon_0}(\Omega_{e, \epsilon_0}, u_n, u)$ by

$$I_i^{e, \epsilon_0} = \int_{\mathbb{R}^2} \int^{T}_{0} I_i(\Omega_{e, \epsilon_0}(x - x', y - y', t - t'), u_n(x, y, t), u(x', y', t')) \, dt' \, dx' \, dy'.$$

\(^2\)This resulting inequality should be compared with inequality (3.11) in [3]
Observe that
\[-D_T^{t,e_0}(\Omega^{t,e_0}, u_{\Delta t}, u) \leq I_1^{t,e_0} + I_2^{t,e_0} + I_3^{t,e_0} + I_4^{t,e_0}.\]

We start by estimating $I_1^{t,e_0}$ and $I_2^{t,e_0}$. Substitution of
\[F_1(u_{\eta}(t), u(t')) = F_1(u_{\eta}(n\Delta t), u(t')) + [F_1(u_{\eta}(t), u(t')) - F_1(u_{\eta}(n\Delta t), u(t'))]\]
into $I_1^{t,e_0}$ yields the equality $|I_1^{t,e_0}| \equiv |I_{1,1}^{t,e_0} + I_{1,2}^{t,e_0}|$, where
\[
I_1^{t,e_0} = \sum_{n=0}^{N-1} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \omega_\varepsilon(x - x')_x \omega_\varepsilon(y - y') \times
\left\{ \int_0^T \int_{n\Delta t}^{(n+\frac{1}{2})\Delta t} 2F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}(t - t') \, dt \, dt' - \int_0^T \int_{n\Delta t}^{(n+1)\Delta t} F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}(t - t'') \, dt \, dt' \right\} \, dx \, dy \, dx' \, dy'.
\]
and
\[
I_2^{t,e_0} = \sum_{n=0}^{N-1} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \omega_\varepsilon(x - x')_y \omega_\varepsilon(y - y') \times
\left\{ \int_0^T \int_{n\Delta t}^{(n+\frac{1}{2})\Delta t} 2(F_1(u_{\eta}(t), u(t')) - F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}(t - t') \, dt \, dt' - \int_0^T \int_{n\Delta t}^{(n+1)\Delta t} (F_1(u_{\eta}(t), u(t')) - F_1(u_{\eta}(n\Delta t), u(t'))) \omega_{e_0}(t - t'') \, dt \, dt' \right\} \, dx \, dy \, dx' \, dy'.
\]

For each $t \in [n\Delta t, (n + 1)\Delta t]$, $t' \in [0, T]$ there exists a number $z_n$ between $n\Delta t$ and $t$ such that $\omega_{e_0}(t - t') = \omega_{e_0}(n\Delta t - t') + \omega_{e_0}'(z_n - t')(t - n\Delta t)$. This implies that
\[
\int_0^T \int_{n\Delta t}^{(n+\frac{1}{2})\Delta t} 2F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}(t - t') \, dt \, dt'
- \int_0^T \int_{n\Delta t}^{(n+1)\Delta t} F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}(t - t') \, dt \, dt'
= \int_0^T \int_{n\Delta t}^{(n+\frac{1}{2})\Delta t} 2F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}'(z_n - t')(t - n\Delta t) \, dt \, dt'
- \int_0^T \int_{n\Delta t}^{(n+1)\Delta t} F_1(u_{\eta}(n\Delta t), u(t')) \omega_{e_0}'(z_n - t')(t - n\Delta t) \, dt \, dt'.
\]

(3.5)

Using (3.5), we have $|I_1^{t,e_0}| \equiv |I_{1,1}^{t,e_0} - I_{1,1,2}^{t,e_0}|$, where
\[
I_{1,1}^{t,e_0} = \sum_{n=0}^{N-1} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \int_0^T \omega_\varepsilon(x - x')_x \omega_\varepsilon(y - y') \omega_{e_0}'(z_n - t')(t - n\Delta t) \, dt \, dx \, dy \, dx' \, dy'.
\]
and
\[ \mathcal{I}^{\varepsilon_0}_{1,1,2} = \sum_{n=0}^{N-1} \int_{R^2} \int_{R^2} \int_0^{\Delta t} \int_0^{(n+1)\Delta t} \omega_{\varepsilon}(x-x') \omega_{\varepsilon}(y-y') F_1(u_{\eta}(x,y,n\Delta t), u(x',y',t')) \times \omega_{\varepsilon_0}(z_n - t') dt' dt dx dy dx' dy'. \]

Furthermore, since \( \int_{R^2} \omega_{\varepsilon}(x-x') dx = 0 \) and since \( F_1 \) satisfies a Lipschitz condition with constant \( A \), it follows that
\[ |\mathcal{I}^{\varepsilon_0}_{1,1,1}| \leq 2A\Delta t \sum_{n=0}^{N-1} \int_{R^2} \int_{R^2} \int_0^{\Delta t} \int_0^{(n+\frac{1}{2})\Delta t} |\omega_{\varepsilon}(x-x')| |\omega_{\varepsilon}(y-y')| \times |u_{\eta}(x,y,n\Delta t) - u_{\eta}(x',y,n\Delta t)||\omega_{\varepsilon_0}(z_n - t')| dt' dt dx dy dx' dy'. \]

Observe that
\[ \int_{R^2} \int_{R^2} |\omega_{\varepsilon}(x-x')| |u_{\eta}(x,y,n\Delta t) - u_{\eta}(x',y,n\Delta t)||\omega_{\varepsilon}(y-y')| dx dy dx' dy' \leq T.V_{(x,y)}(u^0). \]

This follows by using Fubini's theorem, \( \int_{R^2} |\omega_{\varepsilon}'(z)||z| dz = 1 \), and the fact that that \( u_{\eta}(\cdot,\cdot,t) \) is in \( BV(R^2) \) with \( BV \)-norm bounded independently of \( t \). With this inequality we obtain that
\[ |\mathcal{I}^{\varepsilon_0}_{1,1,1}| \leq 2T.V_{(x,y)}(u^0)A\Delta t \sum_{n=0}^{N-1} \int_0^{(n+\frac{1}{2})\Delta t} \int_0^{T} |\omega_{\varepsilon_0}(z_n - t')| dt' dt. \]

Note that \( \omega_{\varepsilon_0}(z_n - t') \equiv 0 \) when \( t' \) is not in \([n\Delta t - \varepsilon_0, (n+1)\Delta t + \varepsilon_0]\), which finally yields the following bound on \( |\mathcal{I}^{\varepsilon_0}_{1,1,1}| \):
\[ |\mathcal{I}^{\varepsilon_0}_{1,1,1}| \leq T.V_{(x,y)}(u^0)AT\Delta t \frac{M_\omega(\Delta t + 2\varepsilon_0)}{\varepsilon_0^3}. \]

\( |\mathcal{I}^{\varepsilon_0}_{1,1,2}| \) is bounded by the same expression which implies that
\[ |\mathcal{I}^{\varepsilon_0}_{1,1} | \leq |\mathcal{I}^{\varepsilon_0}_{1,1,1}| + |\mathcal{I}^{\varepsilon_0}_{1,1,2}| \leq 2AT.V_{(x,y)}(u^0)T\Delta t \frac{M_\omega(\Delta t + 2\varepsilon_0)}{\varepsilon_0^3}. \]

To finally bound \( |\mathcal{I}^{\varepsilon_0}_{1,2}| \), we must first derive a proper upper bound on \( |\mathcal{I}^{\varepsilon_0}_{1,2}| \). Recall that \( |\mathcal{I}^{\varepsilon_0}_{1,2}| \) takes the form \( |\mathcal{I}^{\varepsilon_0}_{1,2,1} = |\mathcal{I}^{\varepsilon_0}_{1,2,1} - |\mathcal{I}^{\varepsilon_0}_{1,2,2}| \), where
\[ \mathcal{I}^{\varepsilon_0}_{1,2,1} = 2 \sum_{n=0}^{N-1} \int_{R^2} \int_{R^2} \int_0^{T} \int_0^{(n+\frac{1}{2})\Delta t} (F_1(u_{\eta}(t), u(t')) - F_1(u_{\eta}(n\Delta t), u(t')) \times \omega_{\varepsilon}(x-x') \omega_{\varepsilon}(y-y') \omega_{\varepsilon_0}(t-t') dt' dt dx dy dx' dy', \]

and
\[ \mathcal{I}^{\varepsilon_0}_{1,2,2} = \sum_{n=0}^{N-1} \int_{R^2} \int_{R^2} \int_0^{T} \int_0^{(n+1)\Delta t} (F_1(u_{\eta}(t), u(t')) - F_1(u_{\eta}(n\Delta t), u(t')) \times \omega_{\varepsilon}(x-x') \omega_{\varepsilon}(y-y') \omega_{\varepsilon_0}(t-t') dt' dt dx dy dx' dy'. \]
If we, once more, exploit the fact that $F_1$ satisfies a Lipschitz condition with constant $A$ and that $u_\eta$ has $L^1$-norm which is Lipschitz continuous in the time variable$^3$, then we obtain the following bound

$$
|T_{1,2,1}^{\varepsilon_0}| \leq 2A \sum_{n=0}^{N-1} \int_{R^3} \int_0^T \int_{n\Delta t}^{(n+\frac{1}{2})\Delta t} |u_\eta(x, y, t) - u_\eta(x, y, n\Delta t)| \times
$$

$$
|\omega_\varepsilon(x - x') \cdot | \omega_\varepsilon(y - y') \cdot \omega_\varepsilon_0(t - t') dt \ dy \ dy' \ dx \ dx' \ dv' \ dv \ \leq T.V.(x, y)(u^0)A^2T\Delta t \frac{M_\omega}{\varepsilon},
$$

where we have used that $\int_{R^3} \omega_\varepsilon(y - y') dy' = 1$, $\int_0^T \omega_\varepsilon_0(t - t') dt' \leq 1$ and $\int_{R^3} |\omega_\varepsilon(x - x') \cdot dx' \leq 2M_\omega/\varepsilon$. Similarly

$$
|T_{1,2,1}^{\varepsilon_0}| \leq 2T.V.(x, y)(u^0)A^2T\Delta t \frac{M_\omega}{\varepsilon}.
$$

From this we conclude that the following estimate holds:

$$
|T_{1,2}^{\varepsilon_0}| \leq |T_{1,1}^{\varepsilon_0}| + |T_{1,2,2}^{\varepsilon_0}| \leq 3T.V.(x, y)(u^0)A^2T\Delta t \frac{M_\omega}{\varepsilon}.
$$

And therefore

$$
|T_1^{\varepsilon_0}| \leq |T_{1,1}^{\varepsilon_0}| + |T_{1,2}^{\varepsilon_0}|
$$

$$
\leq 2T.V.(x, y)(u^0)A^2T\Delta t \frac{M_\omega(\Delta t + 2\varepsilon_0)}{\varepsilon_0^2} + 3T.V.(x, y)(u^0)A^2T\Delta t \frac{M_\omega}{\varepsilon}
$$

$$
= \Delta t \left\{ \frac{K_1(\Delta t + 2\varepsilon_0)}{\varepsilon_0^2} + \frac{K_2}{\varepsilon} \right\}.
$$

A similar analysis for $T_2^{\varepsilon_0}$ yields the same bound, and it follows that

$$
(3.6) \quad |T_1^{\varepsilon_0}| + |T_2^{\varepsilon_0}| \leq \Delta t \left\{ \frac{K_1(\Delta t + 2\varepsilon_0)}{\varepsilon_0^2} + \frac{K_2}{\varepsilon} \right\},
$$

for some constants $K_1$ and $K_2$. It now remains to estimate $|T_3^{\varepsilon_0}|$ and $|T_4^{\varepsilon_0}|$. The technique used to estimate these two terms is strongly inspired by the work of Lucier [10], and we therefore refer the reader to [10] (or [7]) for a more detailed derivation of the bounds on $|T_3^{\varepsilon_0}|$ and $|T_4^{\varepsilon_0}|$. Recall that $T_3^{\varepsilon_0}$ takes the form

$$
T_3^{\varepsilon_0} = \sum_{n=1}^{N-1} \int_0^T \int_{R^3} \left\{ |u_\eta(x, y, n\Delta t +) - u(x', y', t')| - |u_\eta(x, y, n\Delta t -) - u(x', y', t')| \right\} \times
$$

$$
\Omega_{\varepsilon_0}(x - x', y - y', n\Delta t - t') dx \ dy \ dx' \ dy' \ dt',
$$

and that

$$
(3.7) \quad u_\eta(x, y, n\Delta t +) = \frac{1}{\Delta x \Delta y} \int_{\eta} \int_{\eta} u_\eta(X, Y, n\Delta t -) dY dX
$$

$^3$More precisely, we make use of the following result: $||u_\eta(\cdot, t_2) - u_\eta(\cdot, t_2)||_1 \leq A(t_2 - t_1)T.V.(x, y)(u^0)$, for $t_2 > t_1$. 
for all \((x, y) \in (i \Delta x, (i + 1) \Delta x) \times (j \Delta y, (j + 1) \Delta y) \equiv I^x_i \times I^y_j\). Substitution of (3.7) into \(T^{e,0}_3\), yields

\[
T^{e,0}_3 = \frac{1}{N-1} \sum_{n=1}^{N-1} \int_0^T \int_{R^2} \sum_{i,j} \int_{I^x_i \times I^y_j} \int_{I^x_i \times I^y_j} \frac{1}{\Delta x \Delta y} u_\eta(X, Y, n\Delta t-) \, dY \, dX - u(x', y', t') -
\]

\[
|u_\eta(x, y, n\Delta t-) - u(x', y', t')| \Omega_{e,e_0}(x - x', y - y', n\Delta t - t') \, dy \, dx' \, dy' \, dt' = \frac{1}{2} \sum_{n=1}^{N-1} \int_0^T \int_{R^2} \left( \Omega_{e,e_0}(x - x', y - y', n\Delta t - t') - \Omega_{e,e_0}(X - x', Y - y', n\Delta t - t') \right) \times
\]

\[
\sum_{i,j} \int_{I^x_i \times I^y_j} \int_{I^x_i \times I^y_j} \frac{1}{\Delta x \Delta y} \left( |u_\eta(X, Y, n\Delta t-) - u(x', y', t')| -
\]

\[
|u_\eta(x, y, n\Delta t-) - u(x', y', t')| \right) \, dY \, dX \, dy \, dx' \, dy' \, dt'.
\]

Taking absolute values and using the inverse triangle inequality we achieve

\[
|T^{e,0}_3| \leq \frac{1}{2} \sum_{n=1}^{N-1} \int_0^T \int_{R^2} \sum_{i,j} \int_{I^x_i \times I^y_j} \int_{I^x_i \times I^y_j} \frac{1}{\Delta x \Delta y} |u_\eta(X, Y, n\Delta t-) - u_\eta(x, y, n\Delta t-)| \times
\]

\[
|\Omega_{e,e_0}(x - x', y - y', n\Delta t - t') - \Omega_{e,e_0}(X - x', Y - y', n\Delta t - t')| \, dY \, dX \, dy \, dx' \, dy' \, dt'
\]

If we integrate over \(x', y'\) and \(t'\), and assume \((x, y) \in I^x_i \times I^y_j\), we find that

\[
\int_0^T \int_{R^2} |\Omega_{e,e_0}(x - x', y - y', n\Delta t - t') - \Omega_{e,e_0}(X - x', Y - y', n\Delta t - t')| \, dx' \, dy' \, dt'
\]

\[
\leq (\Delta x + \Delta y) \frac{||\omega'||}{\varepsilon}.
\]

Trivially

\[
|u_\eta(X, Y, n\Delta t-) - u_\eta(x, y, n\Delta t-)| = |u_\eta(x, Y, n\Delta t-) - u_\eta(x, y, n\Delta t-)|
\]

\[
\leq T.V.(y \in I^y_j)(u_\eta(x, , n\Delta t-)).
\]

Therefore it follows after some suitable manipulations that

\[
|T^{e,0}_3| \leq \frac{1}{2} \sum_{n=1}^{N-1} \sum_{i,j} \int_{I^x_i \times I^y_j} \int_{I^x_i \times I^y_j} \frac{1}{\Delta x \Delta y} (\Delta x + \Delta y) \frac{||\omega'||}{\varepsilon} \times
\]

\[
\int_{I^x_i} T.V.(y \in I^y_j)(u_\eta(x, , n\Delta t-)) \, dx \, dX \, dY
\]

\[
\leq \frac{1}{2} (\Delta x + \Delta y) \Delta y \frac{T}{\Delta t} \frac{||\omega'||}{\varepsilon} T.V.(x,y)(u_\eta^0).
\]

An identical analysis shows that \(T^{e,0}_4\) is bounded by almost the same expression,

\[
|T^{e,0}_4| \leq \frac{1}{2} (\Delta x + \Delta y) \Delta x \frac{T}{\Delta t} \frac{||\omega'||}{\varepsilon} T.V.(x,y)(u_\eta^0),
\]
which leads to
\[ |I_3^{\varepsilon} + I_4^{\varepsilon}| \leq \frac{K_3(\Delta x + \Delta y)^2}{\varepsilon \Delta t} \]
for some constant $K_3$, and we have obtained a proper estimate of $|D_T^{\varepsilon}(\Omega_{\varepsilon, \eta_0}, u_0, u)|$. With the aid of Kuznetsov theorem we are now in the position of giving the following expression for the error
\[
\|S_\varepsilon(T)u^0(\cdot, \cdot) - u_\eta(\cdot, \cdot, T^-)\|_1 \leq \|u^0 - u^0_\eta\|_1 + K_0\varepsilon_0 + K_0\varepsilon + \frac{K_1\Delta t(2\varepsilon_0 + \varepsilon_0^2)}{\varepsilon \Delta t} + \frac{K_3\Delta t + 4K_3(\Delta x)^2}{\varepsilon \Delta t}.
\]

If we let $\varepsilon_0 = (\Delta t)^{1/2}$, $\varepsilon = (\Delta x)^{1/2}$ and assume $\Delta x = \mathcal{O}(\Delta t)$, then for some constant $K$,
\[
\|S_\varepsilon(T)u^0(\cdot, \cdot) - u_\eta(\cdot, \cdot, T^-)\|_1 \leq K((\Delta t)^{1/2} + (\Delta x)^{1/2}),
\]
as $\Delta x, \Delta t \to 0+$. This concludes the proof of Proposition 3.2.

The previous proposition says that $\|S_\varepsilon(T)u^0(\cdot, \cdot) - u_\eta(\cdot, \cdot, T^-)\|_1$ is of order $(\Delta t)^{1/2} + (\Delta x)^{1/2})$. To extend this result to the case where the flux functions are not necessarily piecewise linear, we have to bound $\|S(T)u^0(\cdot, \cdot) - S_\varepsilon(T)u^0(\cdot, \cdot)\|_1$. In one space dimension, Lucier [10] proved that
\[
\|u(t) - v(t)\|_1 \leq \|u^0 - v^0\|_1 + \|f - g\|_{Lip} \min \{T.V_z(u^0), T.V_z(v^0)\},
\]
where $u$ and $v$ are the exact solutions of the scalar conservation law with flux functions $f$ and $g$, and initial data $u^0$ and $v^0$, respectively. This result is easily extended to the $n$-dimensional case. We therefore state the proposition without proof (the proof may be found in [7]).

**Proposition 3.3.** Let $f_i, g_i$, $i = 1, \ldots, n$, be Lipschitz continuous functions, $u^0, v^0 \in BV(\mathbb{R}^n)$ and $u$ and $v$ are solutions of
\[
\begin{align*}
&u_t + \sum_{i=1}^n f_i(u)x_i = 0 \quad x \in \mathbb{R}^n \\
&u(x, 0) = u^0(x) \quad x \in \mathbb{R}^n
\end{align*}
\]
(3.8)
and
\[
\begin{align*}
&v_t + \sum_{i=1}^n g_i(u)v_i = 0 \quad x \in \mathbb{R}^n \\
&v(x, 0) = v^0(x) \quad x \in \mathbb{R}^n
\end{align*}
\]
(3.9)
then for any $T > 0$,
\[
\|u(T) - v(T)\|_1 \leq \|u^0 - v^0\|_1 + T \max_{1 \leq i \leq n} \|f_i - g_i\|_{Lip} \min \{T.V_{(x_1, \ldots, x_n)}(u^0), T.V_{(x_1, \ldots, x_n)}(v^0)\}.
\]
Theorem 3.4 (Error estimate). If \( u^0 \in L^1(\mathbb{R}^2) \cap L^\infty(\mathbb{R}^2) \cap BV(\mathbb{R}^2) \), \( f \) and \( g \) are piecewise \( C^2 \) on \( I = [-\|u^0\|_\infty, \|u^0\|_\infty] \), \( N\Delta t = T < T_0 \) for some fixed \( T_0 \), \( \Delta x = \Delta y = O(\Delta t) \), and \( \delta \) is the parameter measuring the polygonal approximation of the flux functions, then as \( \eta = (\Delta x, \Delta y, \Delta t) \to 0^+ \) and \( N \to \infty \),

\[
\|S(T)u^0(\cdot, \cdot) - u^N_\eta(\cdot, \cdot)\|_1 \leq C((\Delta t)^{1/2} + (\Delta x)^{1/2} + \delta),
\]

where \( C \) is a finite constant independent of \( \eta \), but dependent on \( T_0, T.V.(\eta, y)(u^0) \), \( \|f\|_\infty \) and \( \|g\|_\infty \).

Proof. If \( f_\delta \) is piecewise linear interpolant of \( f \) (\( f \) Lipschitz continuous and piecewise \( C^2 \)) with breakpoints at \( \delta \), then

\[
\|f - f_\delta\|_{Lip} \leq \frac{\delta}{2}\|f''\|_\infty.
\]

Therefore, by using Proposition 3.3 with \( n = 2 \), we have

\[
\|S(T)u^0(\cdot, \cdot) - S_\delta(T)u^0(\cdot, \cdot)\|_1 \leq C\delta
\]

for some constant \( C \). The theorem now follows from Proposition 3.2, (3.11) and the triangle inequality.

A few remarks may be in order. First, the error estimate (3.10) is not a surprising result considering the fact that if the discretization parameters obey the strict CFL condition \( (\Delta t / \Delta x) \leq \frac{1}{2} \), then the method of dimensional splitting together with with front tracking, is identical to using dimensional splitting and the Godunov scheme. Our method can therefore not have a convergence rate which is better than \( 1/2 \), since it is known that the Godunov scheme has an optimal convergence rate of \( 1/2 \). For details concerning error estimates for Godunov's method we refer the reader to Lucier [10].

Secondly, it is obvious that there are two sources of error in the dimensional splitting - front tracking method. The intrinsic error involved in using the splitting technique and the spatial discretization error involved in using the front tracking scheme and the projection operator. In the general case these two sources of error interact in a complex fashion. It is therefore interesting to see how they are represented in the inequality (3.10). The \((\Delta t)^{1/2}\)-term represents the temporal error, and the \((\Delta x)^{1/2}\)-term the spatial error. The error due to the fact that we replace the flux functions with some approximate functions is represented by the \( \delta \)-term, and the \((\Delta x)^{1/2}\)-term represents the (global) error caused by the projection operator.

Finally, observe that the analysis performed above also yield a convergence rate for the semi-discrete method (1.3). To conclude with Theorem 3.4 we had to find proper bounds on \( I_i, i = 1, \ldots, 4 \). If we, instead of \( u_\eta \), substitute \( u_{\Delta t} \) into \( I_i, i = 1, \ldots, 4 \), then \( I_3 \) and \( I_4 \) become identical to zero. An immediate consequence is that the following estimate of the error holds for the semi-discrete method (1.3),

\[
\|S(T)u^0(\cdot, \cdot) - u^N_{\Delta t}(\cdot, \cdot)\|_1 \leq C(\Delta t)^{1/2}.
\]

This result was shown for the first time by Teng [14].
4. Numerical experiments

In this section we present some examples where we have used dimensional splitting to calculate approximate solutions to (1.1). The purpose of the numerical experiments is to explore some of the properties of the method, such as convergence rates and computational cost. In the previous section a rigorous argument showed that the rate of convergence is no less than 1/2. This result has to be considered as a "worst case" estimate. Thus the method may in many cases have a convergence rate which is noticeably better, but never worse. One of our main objectives is therefore to try to extract from experiments some information concerning the optimality of this estimate of the convergence rate. Another property we would like to investigate is the CPU-time needed to obtain a certain degree of accuracy. The rate of convergence should always be compared with computational cost.

We will measure the error, at a fixed time $T$, in a relative $L^1$-norm, i.e.,

$$\mathcal{E} = \frac{||u^n - u^*||_1}{||u^*||_1}$$

where $n\Delta t = T$, $u^n$ is the solution generated by our numerical method, and $u^*$ is some reference solution. To estimate the rate of convergence, we assume that the error on a grid with size $\Delta x$ (we always use $\delta = C\Delta x$ for some fixed constant $C$, and $\Delta y = \Delta x$) is of the form $\mathcal{E} = \alpha_1(\Delta x)^{\beta_1}$, where $\alpha_1$ is a constant independent of $\Delta x$ and $\beta_1$ the rate of convergence. In each of our examples we will compute approximate solutions for $\Delta x = 2^{-k}, k = 2, \ldots, 8$. Furthermore, two relations between $\Delta t$ and $\Delta x$ will be considered:

- Case 1) $\frac{\Delta t}{\Delta x} \leq \frac{1}{2}$ (strict CFL condition): In this case no shock collisions are allowed to take place.
- Case 2) $\Delta t = 0.95(\Delta x)^{1/2}$: Here we take advantage of the fact that no CFL condition is needed to make the method stable. The effect of using this relation is that the time step will be larger than the space step, and in this way we allow shock collisions to occur.

The parameters $\alpha_1$ and $\beta_1$ are determined using a standard regression analysis. In the examples where the exact solution is not explicitly known, we will use our method to calculate the reference solution using a very fine grid: $\delta = \Delta x = \Delta y = 2^{-9}, \Delta t = 0.95(\Delta x)^{1/2}$. The reason for using $\Delta t = 0.95(\Delta x)^{1/2}$ to calculate the reference solution, is that numerical experiments indicate that the method is considerably improved, in every respect, using this relation between the space step and the time step. The essential fact here is that the time step should be larger than the space step. The effect of using a very fine grid solution as the reference solution is that the convergence rate will be too large when the coarser grid approaches the grid used to calculate the reference solution. Similarly, if the grid becomes too coarse we may not pick up asymptotic effects. We will therefore use a regression analysis on the observation for $k = 4, 5, 6, 7$, to compute $\alpha_1$ and the rate of convergence $\beta_1$ (in the cases where the exact solution is known we use a regression analysis on the observations for $k = 4, 5, 6, 7, 8$). Similarly to the error $\mathcal{E}$, we assume that the CPU-
time (measured in seconds) used by the method on a particular problem is of the form \( T = \alpha_2 \varepsilon^{\beta_2} \) for some constants \( \alpha_2 \) and \( \beta_2 \).

The method of dimensional splitting using the front tracking scheme as approximate solution operators was implemented in the programming language C on a SILICON GRAPHICS (INDY) work-station, MIPS R4000/100 MHZ/32 MB RAM. We will present three examples where our method has been used to calculate approximate solutions. The first one is a linear problem, though, with discontinuous initial data. It is known that Godunov's method behave badly (slow convergence rate, etc.) on this problem in one dimension, cf. Lucier [10]. This example is therefore used to convince ourselves and the reader of the existence of a problem where the convergence rate actually is close to \( 1/2 \). The remaining two examples are true non-linear problems where shock collisions take place and perhaps the most interesting examples with respect to practical use of the method.

4.1. Example 1. We will first consider the scalar advection equation

\[
u_t + u_x + u_y = 0,
\]

with initial data

\[
u(x, y, 0) = u^0(x, y) = \begin{cases} 1 & x \leq 0 \\ 0 & x > 0. \end{cases}
\]

The exact solution is given by

\[
u(S(t))u^0(x, y) = \begin{cases} 1 & x \leq t \\ 0 & x > t. \end{cases}
\]

This is essentially a one-dimensional Riemann problem. It is known that the (1D) Godunov scheme has a convergence rate of \( 1/2 \) on this problem. We compute approximate solutions to \( S(T)u^0 \) with \( T = 0.5 \). The computational domain is taken to be \([-0.5, 2] \times [-0.5, 2] \).

As mentioned before, we will study the behaviour of the method using two different relations between the space step and the time step. First, we let \( \Delta x \) and \( \Delta t \) obey a strict CFL condition; \( \Delta t = 0.5\Delta x \) (Case 1). The results of the simulations are presented in Table 1. There we list the grid size \( \Delta x \), the error \( \varepsilon \), the CPU-time \( T \), the total number of solved Riemann problems, and the total number shock collisions that occurred during the simulation. A picture of an approximate solution is shown in Figure 1. This solution was generated with \( \Delta x = 2^{-6}, \Delta t = 0.5\Delta x \), and \( \delta = 4\Delta x \).

The different values are shown on a grey scale such that black corresponds 0 and white corresponds to 1. The labels on the axes denote grid blocks, the lower left-hand corner has coordinates \((-0.5, -0.5)\) and the upper right-hand corner \((2.0, 2.0)\). The experiments suggest a convergence rate \( \beta_1 = 0.50 \) and a CPU-time which increases like \( O(\varepsilon^{-5.74}) \). Next, we let \( \Delta t = 0.95(\Delta x)^{1/2} \) (Case 2), i.e., we use a time step that is larger than the space step. The results are presented in Table 2. We observe that the convergence rate is slightly improved; \( \beta_1 = 0.83 \), but more importantly the CPU-time has decreased dramatically. The CPU-time now increases, as the error
<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>$\mathcal{E}$ (Error)</th>
<th>$T$ (CPU-time)</th>
<th># Riemann problems</th>
<th># shock collisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{-2}$</td>
<td>0.199473</td>
<td>0.01</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-3}$</td>
<td>0.133270</td>
<td>0.04</td>
<td>576</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-4}$</td>
<td>0.100053</td>
<td>0.23</td>
<td>4352</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-5}$</td>
<td>0.069318</td>
<td>1.73</td>
<td>30976</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-6}$</td>
<td>0.049099</td>
<td>12.46</td>
<td>196467</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-7}$</td>
<td>0.034930</td>
<td>93.50</td>
<td>1127427</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-8}$</td>
<td>0.024567</td>
<td>709.25</td>
<td>6263138</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 1.** Example 1, $\Delta t = 0.5\Delta x$, $\delta = 4\Delta x$. Based on a standard regression analysis we compute: $\mathcal{E} = 0.4(\Delta x)^{0.50}$ with an estimated error of 0.0002 and $T = 0.0000004\mathcal{E}^{-5.74}$ with an estimated error of 0.007.

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>$\mathcal{E}$ (Error)</th>
<th>$T$ (CPU-time)</th>
<th># Riemann problems</th>
<th># shock collisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{-2}$</td>
<td>0.050283</td>
<td>0.01</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-3}$</td>
<td>0.060023</td>
<td>0.01</td>
<td>63</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-4}$</td>
<td>0.037273</td>
<td>0.04</td>
<td>252</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-5}$</td>
<td>0.024850</td>
<td>0.15</td>
<td>756</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-6}$</td>
<td>0.015351</td>
<td>0.81</td>
<td>3360</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-7}$</td>
<td>0.007311</td>
<td>3.72</td>
<td>9099</td>
<td>0</td>
</tr>
<tr>
<td>$2^{-8}$</td>
<td>0.003941</td>
<td>21.10</td>
<td>30330</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 2.** Example 1, $(\Delta t = 0.95(\Delta x)^{1/2}$, $\delta = 4\Delta x$). Based on a standard regression analysis we compute: $\mathcal{E} = 0.41(\Delta x)^{0.83}$ with an estimated error of 0.04 and $T = 0.0000006\mathcal{E}^{-2.73}$ with an estimated error of 0.2.

tends to zero, like $O(\mathcal{E}^{-2.73})$. To obtain an error of e.g. 0.02 we need a CPU-time of more than 709 seconds in Case 1, while in Case 2 we need less than 1 second.

The CPU-time used by the method may also be expressed in terms of the grid size $\Delta x$. In Case 1 we get $T = O((\Delta x)^{-2.87})$ and in Case 2, $T = O((\Delta x)^{-2.23})$. In the concluding remarks we will argue that the CPU-time should be of order $(\Delta x)^{-\gamma}$, where $\gamma$ in Case 1 is close to 3 and in Case 2 close to 2.5.

A nice feature with the front tracking method is that it does not smear out discontinuous parts of the solution (shocks), which is a major problem for many finite difference methods. Since the projection operator $\pi$ is simply the grid block average, our method will have this problem. This is clearly illustrated in Figure 1. Of course this "smearing effect" can be, more or less, avoided by using a more sophisticated projection operator which "tries to discover" where the solution is smooth and where it is discontinuous, cf. Bratvedt, et. al. [1]
4.2. Example 2. We now present an example where we generate approximate solutions to the equation
\[ u_t + f(u)_x + g(u)_y = 0 \]
with initial data
\[ u(x, y, 0) = u^0 = \begin{cases} 
0 & x^2 + y^2 < 0.23 \\
1 & \text{elsewhere,} 
\end{cases} \]
and
\[ g(u) = \frac{u^2}{u^2 + (1-u)^2}, \quad f(u) = g(u)(1 - (1-u)^2). \]
The flux functions \( f \) and \( g \) both have an S-shaped form with \( f(0) = g(0) = 0 \) and \( f(1) = g(1) = 1 \). This problem is motivated from two-phase flow in porous medium with gravitation in the y-direction. This time the exact solution is not known, so we use our numerical method to compute the reference solution using a very fine grid; \( \Delta y = \Delta x = 2^{-9} \), \( \Delta t = 0.95(\Delta x)^{1/2} \), and \( \delta = 4\Delta x \). We use again \( T = 0.5 \) and a computational domain equal to the square \([-1.5, 1.5] \times [-1.5, 1.5]\). Since most practical problems are indeed non-linear, it is interesting to see how our method performs on this example. It is worth noting that the solutions of the one-dimensional equations consist of both shock and rarefaction waves.

The results are presented in Table 3 and 4. Pictures of approximate solutions are shown in Figure 2, 3, 4, and 5. These solutions were generated with \( \Delta x = 2^{-5} \), \( \Delta t = 0.95(\Delta x)^{1/2} \), and \( \delta = 4\Delta x \). The different values are shown on a grey scale such that black corresponds 0 and white corresponds to 1. The labels on the axes denote grid blocks, the lower left-hand corner has coordinates \((-1.5, -1.5)\) and the upper right-hand corner \((1.5, 1.5)\). Since the problem is non-linear, shock collisions take place in Case 2. This is illustrated in Table 4, where we see that the number
of shock collisions increases monotonically with decreasing \( \Delta x \). In Case 1 we obtain \( \beta_1 = 0.81 \) and in Case 2, \( \beta_1 = 0.93 \). So, as before, we see an improvement in the convergence rate in Case 2 compared with Case 1. Furthermore, if we use a time step that is larger than the space step as opposed to a CFL condition, then the decrease in consumed CPU-time is impressive. In Case 1 we have \( \beta_2 = -3.73 \) and in Case 2, \( \beta_2 = -2.67 \). To achieve an accuracy of 0.02 we need at least a CPU-time of 432 seconds in Case 1, while in Case 2 we need 21 seconds.

If we again express the CPU-time as a function of \( \Delta x \) we obtain in Case 1, \( T = \mathcal{O}((\Delta x)^{-3.0}) \) and in Case 2, \( T = \mathcal{O}((\Delta x)^{-2.48}) \). This is in good agreement with the predictions made in Example 1.

4.3. Example 3. In our last example we will consider the non-linear equation

\[ u_t + \cos(2\pi u)_x + (u^2)_y = 0, \]

with initial data

\[ u(x, y, 0) = u^0(x, y) = \begin{cases} 
-1 & |(x, y) - (0.4, 0.4)| < 0.5, \\
1 & |(x, y) - (-0.4, -0.4)| < 0.5, \\
0 & \text{otherwise} 
\]
**Figure 2.** Example 2: $T = 0.0$.

**Figure 3.** Example 2: $T = 0.17$
**Figure 4.** Example 2: $T = 0.34$.

**Figure 5.** Example 2: $T = 0.5$. 
ON THE ACCURACY OF A NUMERICAL METHOD FOR TWO-DIMENSIONAL … 23

<table>
<thead>
<tr>
<th>( \Delta x )</th>
<th>( \mathcal{E} ) (Error)</th>
<th>( T ) (CPU-time)</th>
<th># Riemann problems</th>
<th># shock collisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2(^{-2})</td>
<td>0.402639</td>
<td>0.02</td>
<td>188</td>
<td>6</td>
</tr>
<tr>
<td>2(^{-3})</td>
<td>0.156726</td>
<td>0.06</td>
<td>392</td>
<td>72</td>
</tr>
<tr>
<td>2(^{-4})</td>
<td>0.162731</td>
<td>0.30</td>
<td>1954</td>
<td>266</td>
</tr>
<tr>
<td>2(^{-5})</td>
<td>0.078906</td>
<td>1.56</td>
<td>12268</td>
<td>1525</td>
</tr>
<tr>
<td>2(^{-6})</td>
<td>0.032520</td>
<td>9.06</td>
<td>73481</td>
<td>10257</td>
</tr>
<tr>
<td>2(^{-7})</td>
<td>0.016000</td>
<td>69.27</td>
<td>478550</td>
<td>52532</td>
</tr>
</tbody>
</table>

**Table 5.** Example 3, \( \Delta t = 0.95(\Delta x)^{1/2}, \delta = 4\Delta x \). Based on a standard regression analysis we compute: \( \mathcal{E} = 3.8(\Delta x)^{1.13} \) with an estimated error of 0.006 and \( T = 0.004\mathcal{E}^{-2.30} \) with an estimated error of 0.1.

The exact solution is again not known, so we use our numerical method to compute the reference solution using a very fine grid; \( \Delta y = \Delta x = 2^{-9}, \Delta t = 0.95(\Delta x)^{1/2} \), and \( \delta = 5\Delta x \). We use \( T = 0.5 \) and a computational domain equal to the square \([-1.5, 1.5] \times [-1.5, 1.5] \). The results are presented in Table 5. Pictures of approximate solutions are shown in Figure 6, 7, 8, and 9. These solutions were generated with the same discretization parameters as in the previous example. The different values are shown on a grey scale, this time, such that black corresponds \(-1 \) and white corresponds to \( 1 \).

Due to hardware limitations we could not perform the calculations using a strict CFL condition (Case 1). Instead we used \( \Delta t = 0.25\Delta x \). Of course this relation between the discretization parameters imply that shock collisions occur. Nevertheless, in this case we obtained a convergence rate \( \beta_1 = 0.75 \) and a CPU-time rate \( \beta_2 = -4.31 \). In Case 2 we got \( \beta_1 = 1.13 \) and \( \beta_2 = -2.30 \). The clear distinction between (Case 1) and Case 2 is highly consistent with the previous observations.

Expressing the CPU-time as a function of the grid size \( \Delta x \) we obtain in (Case 1), \( T = \mathcal{O}((\Delta x)^{-3.38}) \) and in Case 2, \( T = \mathcal{O}((\Delta x)^{-2.8}) \).

5. CONCLUDING REMARKS

The numerical examples indicate that the convergence rate always lies in the interval \([1/2, 1] \). Our estimate of the convergence rate (Theorem 3.4) can therefore be considered as a reasonable result for a general problem. In particular we found an example where the convergence rate turned out to be \( 1/2 \). The suggested estimate of \( 1/2 \) is therefore the best we can hope for keeping in mind that this estimate should be independent of the flux functions, the initial data and the relationship between the discretization parameters. Furthermore, it seems that the numerical convergence rate is improved when we allow the time step to be larger than the space step, but not so much that we achieve, in the general case, a convergence rate of \( 1 \). In Case 2 the convergence rate is, in each of the three examples, noticeably better than \( 1/2 \).

The examples also indicate that the CPU-time as a function of the grid size \( \Delta x \) is of order \( \mathcal{O}((\Delta x)^{-\gamma}) \) where \( \gamma \) is between 2.5 and 3 in Case 1, and between 2 and 2.5 in Case 2. That this is a fairly reasonable result can be argued for by inspecting the
Figure 6. Example 3: $T = 0.0$.

Figure 7. Example 3: $T = 0.17$. 
FIGURE 8. Example 3: $T = 0.34$.

FIGURE 9. Example 3: $T = 0.5$. 
algorithm for dimensional splitting (see [6] or [7]). If we assume that the CPU-time used by the front tracking method on a particular 1D problem is roughly $O((\Delta x)^{-1})$, then the CPU-time used to advance the solution one time step $\Delta t$ is $O((\Delta x)^{-2})$. Therefore the total amount of CPU-time used to calculate an approximate solution to the 2D problem is $O((\Delta x)^{-2}(\Delta t)^{-1})$. In Case 1 this means that the CPU-time should be of order $(\Delta x)^{-3.0}$ and in Case 2 of order $(\Delta x)^{-2.5}$.

Using a time step that is larger than the space step makes our method faster. It uses little CPU-time to obtain a certain degree of accuracy. This, together with the fact that the convergence rate in Case 2 seems to be noticeable better than 1/2, indicates that we here have a method which actually works quite well on most problems, despite the similarities to the method of Godunov. To further improve the method one could use a more sophisticated projection operator to avoid excessive smearing of discontinuities. This has been done in a black oil reservoir simulator where dimensional splitting together with front tracking has been used to solve the saturation equation, cf. [1] for details. We also mention that it is trivial to extend our method to higher dimensions for which the convergence results derived in this paper still hold.

ACKNOWLEDGEMENT

The author would like to thank Dr. Nils Henrik Risebro for introducing me to the fascinating field of hyperbolic conservation laws and for being a constant source of inspiration during my work on this paper. Furthermore, the author would like to thank Jorun Reikvam and Nils Henrik Risebro for their careful reading of the manuscript. Their corrections helped in improving the quality of the language and the clarity of the text. Finally, the author would like to thank Knut-Andreas Lie (NTH) for permission to use his computer code. To increase the reliability of the numerical results both the codes of Lie and the author were used to calculate the approximate solutions. We observed that the rates (convergence rates and CPU-time rates) were roughly independent of the code being used.
REFERENCES


(Kenneth Hvistendahl Karlsen) Department of Mathematics, Mathematics Division, University of Oslo, P.O. Box. 1053, Blindern, N-0316 Oslo, Norway
E-mail address: kennethk@math.uio.no