Burgers' Equation

Christof Obertscheider

Abstract

This paper presents a mathematical analysis of the inviscid and the viscid Burgers' equation. Furthermore numerical methods for the inviscid Burgers' equation - a hyperbolic PDE - and partly for the viscid Burgers' equation - a parabolic PDE - are presented.

Contents

1 Introduction

This paper is for the course "Praktikum aus Angewandter Mathematik" in the sommer term 2001 held by Ao.Univ.Prof. Dr. Norbert Mauser at the Institute for Mathematics at the University of Vienna.

In the second section a mathematical analysis of the viscid and the inviscid Burgers' equation is performed.

In the third section numerical methods for the viscid an the inviscid Burgers' equation are presented. The viscid Burgers' equation is a parabolic PDE whereas the inviscid Burgers'equation is a hyperbolic PDE. Therefore we must develope different numerical methods for the viscid and the inviscid Burgers' Equation. The main part of this section deals with the inviscid Burgers' equation: demonstrating the difficulties and developing and analyzing methods which

compute the entropy satisfying soluting. A short subsection presents a numerical method for the viscid Burgers' equation.

In the fourth section results of the numerical methods are presented and interpreted.

Appendix A shows the source code of the Fortran90 program with was used to compute the examples shown in the fourth section.

This work is based on the two books

- Randall J. LeVeque, Numerical Methods of Conservation Laws, Birkhäuser, 1990
- Charles A. Hall & Thomas A. Porsching, Numerical Analysis of Partial Differential Equations, Prentice-Hall, 1990

and the lecture notes

- Norbert Mauser, Angewandte Mathematik, winter term 1999/2000

2 Mathematical Theory

2.1 Viscid Burgers' Equation

Consider the flow of cars on a highway. Let $\rho(x,t)$ denote the density of cars and let $f(x,t)$ denote the traffic flow.

Since cars are conserved the density of cars and the flow must be related by the continuity equation

$$\frac{d\rho^*}{dt^*} + \frac{df}{dx^*} = 0 \quad x \in R, t^* > 0 \tag{1}$$

In order to obtain a scalar conservation law of ρ alone we assume the traffic folw to be

$$f(\rho^*) = \rho^* v(\rho^*) - D\frac{d\rho^*}{dx^*} \tag{2}$$

The second expression effects larger distances between the cars during huge changes of the density of cars.

In this application ρ^* is resticted to a certain range, $0 \leq \rho^* \leq \rho_{max}$, where ρ_{max} is the value at which cars are bumper to bumper.

We assume that v is a given function of ρ^*. This makes sence: On a highway we would optimally like to drive at some speed v_{max} (the speed limit perhabs), but in heavy traffic we slow down, with velocity decreasing as density increases. The simplest model is the linear relation

$$v\left(\rho^*\right) = \frac{v_{max}}{\rho_{max}}\left(\rho_{max} - \rho^*\right) \tag{3}$$

At zero density (empty road) the speed is v_{max}, but decreases to zero as ρ aprroaches ρ_{max}.

We obtain the equation

$$\frac{d\rho^*}{dt^*} + \frac{d}{dx^*}\left[\frac{v_{max}}{\rho_{max}}\left(\rho_{max} - \rho^*\right)\rho^*\right] = D\frac{d\rho^{*2}}{dx^{*2}} \tag{4}$$

Scaling

$$\begin{aligned} \rho &= \rho_{max}\rho^* \quad v_{max} = x_0/t_0 \\ x &= x_0 x^* \\ t &= t_0 t^* \end{aligned}$$

results in

$$\rho_t + \left[(1-\rho)\,\rho\right]_x = \epsilon\rho_{xx} \quad with\ \epsilon = \frac{D}{v_{max}x_0},\ 0 \le \rho \le 1 \tag{5}$$

The transformation $u = 2\rho - 1$ yields in

$$u_t + uu_x = \epsilon u_{xx} \qquad \epsilon \ll 1,\ -1 \le u \le 1,\ x \in R \tag{6}$$

Equation (6) is called the **viscid Burgers' equation**, a parabolic PDE.

The Cauchy problem can be explicitly solved, provided $u(x,0) = u_0(x)\ with\ x \in R,\ u_0 \in C^1(R)$.

We can rewrite equation (6)

$$u_t + \left(\frac{u^2}{2}\right)_x = \epsilon u_{xx}$$

Performing the trick $u = \varphi_x$ yields

$$\varphi_{xt} + \left(\frac{\varphi_x^2}{2}\right)_x = \epsilon\varphi_{xxx}$$

After integrating in x-direction we obtain

$$\varphi_t + \frac{\varphi_x^2}{2} = \epsilon\varphi_{xx}$$

Let $\varphi = -2\epsilon\ln v$. It follows that $v_t = \epsilon v_{xx}$ and for the initial data $v(x,0) = v_0(x) := \exp\left(-\frac{1}{2\epsilon}\int_0^x u_0(\xi)d\xi\right)$. This is the Cauchy problem for the heat equation with the solution

$$\begin{aligned} v(x,t) &= \frac{1}{\sqrt{4\pi\epsilon t}}\int_R v_0(\xi)e^{-\frac{(x-\xi)^2}{4\epsilon t}}d\xi \\ &= \frac{1}{\sqrt{4\pi\epsilon t}}\int_R e^{-\frac{G(\xi;x,t)}{2\epsilon}}d\xi \quad with \qquad (7) \\ G(\xi;x,t) &= \int_0^\xi u_0(\eta)d\eta + \frac{(x-\xi)^2}{2t} \end{aligned}$$

Transforming equation (7) yields

$$u(x,t) = -2\epsilon\frac{v_x}{v} = \frac{\int_R \frac{x-\xi}{t}\,e^{-G/2\epsilon}d\xi}{\int_R e^{-G/2\epsilon}d\xi} \tag{8}$$

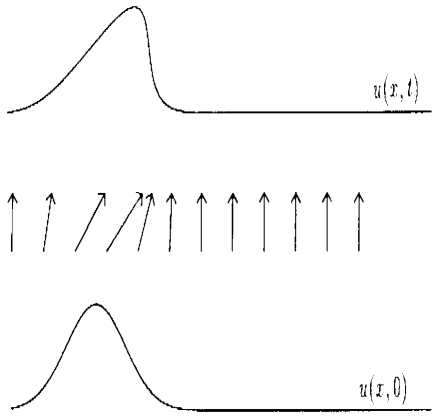

Figure 1: Characteristics and solution for Burgers' equation (small t)

2.2 Inviscid Burgers' Equation

We now consider the **inviscid Burgers' equation** ($\epsilon = 0$)

$$u_t + uu_x = 0 \qquad x \in R,\ u(x,0) = u_0(x) \tag{9}$$

Equation (9) is a nonlinear scalar hyperbolic conservation law

$$u_t + f(u)_x = 0 \quad with\ f(u) = \frac{1}{2}u^2 \tag{10}$$

Consider the inviscid equation (9) with smooth initial data. For small time, a solution can be constructed by following characteristics. The characteristics satisfy

$$x'(t) = u(x(t),t) \tag{11}$$

and along each characteristic u is constant, since

$$\begin{aligned} \frac{d}{dt}u(x(t),t) &= \frac{\partial}{\partial t}u(x(t),t) + \frac{\partial}{\partial x}u(x(t),t)\cdot x'(t) \\ &= u_t + uu_x \\ &= 0 \end{aligned}$$

Moreover since u is constant on each characteristic, the slope $x'(t)$ is constant by equation (**??**) and so the characteristics are straight lines, determined by the initial data.

If the initial data is smooth then this can be used to determine the solution $u(x,t)$ for small enough t that characteristics do not cross: For each (x,t) we can solve the equation

$$x = \xi + u(\xi,0)\cdot t \tag{12}$$

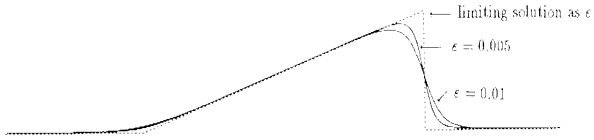

Figure 2: Solution to the viscous Burgers' equation for two different values of ϵ

for ξ and then

$$u(x,t) = u(\xi,0) \tag{13}$$

For larger t the equation (12) may not have a unique solution. This happens when the characteristics first cross. At the time T_b where the characteristics first cross, the function $u(x,t)$ has an infinite slope - the wave "breaks" an a **shock** forms. Beyond this point there is no classical solution of the PDE and the weak solution we wish to determine becomes discontinuous.

For times $t > T_b$ some of the characteristics have crossed and so there are points x where there are three characteristics leading back to $t = 0$. One can view the "solution" u at such a time as a triple-valued function. However in most physical situations this does not make sence. For example, the density of gas cannot possibly be tripple-valued. What happens instead at time T_b?

We can determine the correct physical behaviour by adopting the vanishing viscosity approach. Equation (9) is a model of equation (6) valid only for small ϵ and smooth u. When the solution breaks down, we must return to equation (6). If the initial data is smooth and ϵ very small, then before the wave begins to break the ϵu_{xx} term is neglectible compared to the other terms and the solution to both PDEs look nearly identical.

For very small values of ϵ, a discontinuous solution at $t > T_b$ would be replaced by a smooth continuous function as in 2. As $\epsilon \to 0$ this becomes sharper and approaches the discontinuous solution.

For times $t > T_b$ the viscous solution for $\epsilon > 0$ would continue to be smooth and single valued. The behaviour as $\epsilon \to 0$ is indicated in 2. It is this vanishing viscosity solution that we hope to compute by solving the inviscid equation.

Let C_0^1 be the space of functions that are continuously differentiable with compact support. To define a generalized solution of the inviscid equation we will use test functions $\phi \in C_0^1(R \times R)$. If we multiply $u_t + f_x = 0$ by $\phi(x,t)$ and then integrate over space and time, we obtain

$$\int_0^\infty \int_{-\infty}^\infty [\phi u_t + \phi f(u)_x]\, dxdt = 0 \tag{14}$$

Now integrating by parts, yielding

$$\int_0^\infty \int_{-\infty}^\infty [\phi_t u + \phi_x f(u)]\, dxdt = -\int_{-\infty}^\infty \phi(x,0)u(x,0)\, dx \tag{15}$$

Note that nearly all the boundary terms which normally arise through integration by parts drop out due to the requirement that ϕ have compact support, and

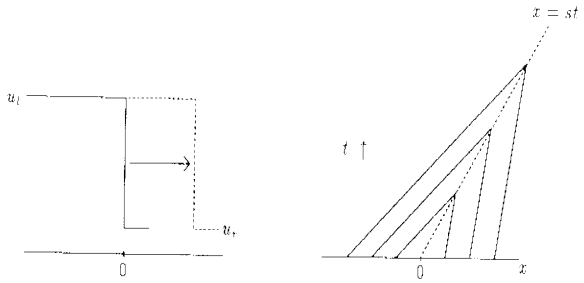

Figure 3: Shock wave

hence vanisch at infinity. The remainding boundary term brings in the initial condition of the PDE which must still play a role in our weak formulation.

DEFINITION 1. The function $u(x,t)$ is called a weak solution of the conservation law if (15) holds for all functions $\phi \in C_0^1(R \times R)$

The vanisching viscosity generalized solution we defined above is a weak solution in the sence of (15), and so this definition includes the solution we are looking for. Unfortunately, weak solutions often are not unique, and so an additional problem is often to identify which weak solution is the physically correct vanishing viscosity solution.

One would like to avoid working with the viscous equation directly, but it turns out that there are ohter conditions one can impose on weak solutions that are easier to check and will also pick out the correct solution. These are usually called entropy conditions by analogy with the gas dynamics case. The vanishing viscosity solution is also called the entropy solution because of this.

Riemann Problem. The conservation law together with piecewise constant data having a single discontinuity is known as the Riemann problem.

Consider Burgers' equation $u_t + uu_x = 0$ together with piecewise constant initial data

$$u(x,t) = \left\{ \begin{array}{ll} u_l & x < 0 \\ u_r & x > 0 \end{array} \right\} \tag{16}$$

The form of the solution depends on the relation between u_l and u_r.

Case I. $u_l > u_r$

In this case there is a unique weak solution

$$u(x,t) = \left\{ \begin{array}{ll} u_l & x < st \\ u_r & x > st \end{array} \right\} \tag{17}$$

where

$$s = \frac{1}{2}(u_l + u_r) \tag{18}$$

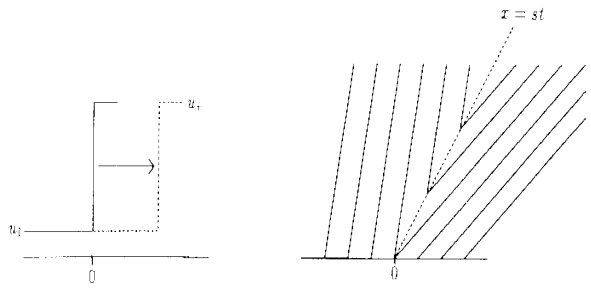

Figure 4: Entropy violating shock

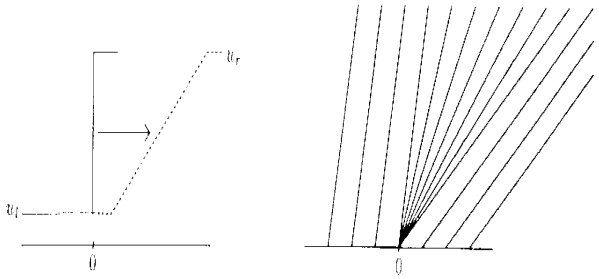

Figure 5: Rarefaction wave

is the **shock** speed, the speed at which the discontiunity travels. A general expression for the shock speed will be derived below. Note that characteristics in each of the region where u is constant go into the shock (see 3) as time advances.

Case II. $u_l < u_r$
In this case there are infinitely many weak solutions. One of these is again (17) in which the discontinuity propagates with speed (18). Note that characteristics now go out of the shock (see 4) and that this solution is not stable to perturbations. If the data is smeared out slightly, or if a small amount of viscosity is added to the equation, the solution changes completely. Another weak soltution is the **rarefaction wafe**

$$u(x,t) = \left\{ \begin{array}{ll} u_l & x < u_l t \\ x/t & u_l t \leq x \leq u_r t \\ u_r & x > u_r t \end{array} \right\} \tag{19}$$

This solution is stable to perturbations and is in fact the vanishing viscosity generalized solution (see 5).

The propagating shock solution (17) is a weak soltution to Burger's equation only if the speed of propagation is given by (18). The same discontinuity

propagating at a different speed would not be a weak solution.

The speed of propagation can be determined by conservation. If M is large compared to st then $\int_{-M}^{M} u(x,t)\,dx$ must increase at the rate

$$\frac{d}{dt}\int_{-M}^{M} u(x,t)\,dx = f(u_l) - f(u_r) = \frac{1}{2}(u_l + u_r)(u_l - u_r) \tag{20}$$

for Burgers' equation. On the other hand, the solution (17) clearly has

$$\int_{-M}^{M} u(x,t)\,dx = (M + st)u_l + (M - st)u_r \tag{21}$$

so that

$$\frac{d}{dt}\int_{-M}^{M} u(x,t)\,dx = s(u_l - u_r) \tag{22}$$

Comparing (22) and (20) gives (18). More generally, for arbitrary flux functions $f(u)$ this same argument gives the following relation between the shock speed s and the states u_l and u_r, called the Rankine-Hugoniot jump condition

$$f(u_l) - f(u_r) = s(u_l - u_r) \tag{23}$$

For scalar problems this gives simply

$$s = \frac{f(u_l) - f(u_r)}{u_l - u_r} = \frac{[f]}{[u]} \tag{24}$$

where [.] indicates the jump in some quantity across the discontinuity.

There are situations in which the weak solution is not unique and an additional condition is required to pick out the physically relevant vanishing viscosity solution. The condition which defines this solution is that it should be the limiting solution of the viscous equation as $\epsilon \to 0$, but this is not easy to work with. We want to find a simpler condition.

For scalar equations there is an obvious condition suggested by 3 and 5. A shock sould have characteristics going into the shock, as time advances. A propagating discontinuity with characteristics coming out of it, as in 4, is unstable to perturbations. Either smearing out the initial profile a little, or adding some viscosity to the system will cause this to be replaced by a rarefaction fan of characteristics, as in 5. This gives our first version of the entropy condition.

Entropy Condition (Version I). A discontinuity porpagating with speed s given by (23) satisfies the entropy condition if

$$f'(u_l) > s > f'(u_r) \tag{25}$$

Note that $f'(u)$ is the characteristic speed. For convex f, the Rankine-Hugoniot speed s from (24) must be between $f'(u_l)$ and $f'(u_r)$, so that (25) reduces to the requirement that $f'(u_l) < f'(u_r)$, which again by convexity requires $u_l < u_r$.

Another approach to the entropy conditon is to define an entropy function $\eta(u)$ for which an additional conservation law holds for smooth solutions and that becomes an inequality for discontinuous solutions. In gas dynamics, there exists a physical quantity called the entropy that is known to be constant along particle paths in smooth flow and to jump to a higher value as the gas crosses a shock. It can never jump to a lower value, and this gives the physical entropy condition that picks out the correct weak solution in gas dynamics.

Suppose some function $\eta(u)$ satisfies a conservation law of the form

$$\eta(u)_t + \Psi(u)_x = 0 \tag{26}$$

for some entropy flux $\Psi(u)$. Then we can obtain from this, for smooth u

$$\eta'(u)u_t + \Psi'(u)u_x = 0 \tag{27}$$

The conservation law $u_t + f(u)_x = 0$ can be written as $u_t + f'(u)u_x = 0$. Multiply this by $\eta'(u)$ and compare with (27) to obtain

$$\Psi'(u) = \eta'(u)f'(u) \tag{28}$$

For scalar conservation law this equation admits many solutions $\eta(u), \Psi(u)$. An additional condition we place on the entropy function is that it be convex, $\eta''(u) > 0$, for reasons that will be seen below.

The entorpy $\eta(u)$ is conserved for smooth flows by its definition. For discontinuous solutions, however, the manipulation performed above are not valid. Since we are particularly interested in how the entropy behaves for the vanishing viscosity weak solution, we look at the related viscous problem and will then let the viscosity tend to zero. The viscous equation is

$$u_t + f(u)_x = \epsilon u_{xx} \tag{29}$$

Since solutions to this equation always are smooth, we can derive the corresponding evolution equation for the entropy following the same manipulations we used for smooth solutions of the inviscid equation, multiplying (29) by $\eta'(u)$ to obtain

$$\eta(u)_t + \Psi(u)_x = \epsilon\eta'(u)u_{xx} \tag{30}$$

We can now rewrite the right hand side to obtain

$$\eta(u)_t + \Psi(u)_x = \epsilon \left(\eta'(u)u_x\right)_x - \epsilon\eta''(u)u_x^2 \tag{31}$$

Integration this over the rectangle $[x_1, x_2] \times [t_1, t_2]$ gives

$$\int_{t_1}^{t_2}\int_{x_1}^{x_2} \eta(u)_t + \Psi(u)_x \, dxdt = \tag{32}$$

$$\epsilon \int_{t_1}^{t_2} \left[\eta'\left(u(x_2,t)\right) u_x(x_2,t) - \eta'\left(u(x_1,t)\right) u_x(x_1,t)\right] dt$$

$$-\epsilon \int_{t_1}^{t_2}\int_{x_1}^{x_2} \eta''(u)u_x^2 \, dxdt$$

As $\epsilon \to 0$ the first term on the right hand side vanishes. (This is clearly true if u is smooth at x_1 and x_2 and can be shown more generally). The other term, however, involves integrating u_x^2 over $[x_1, x_2] \times [t_1, t_2]$. If the limiting weak solution is discontinuous along a curve in this rectangle, then the term will not vanisch at the limit. However, since $\epsilon > 0,\ u_x^2 > 0\ and\ \eta'' > 0$ (by our assumptions), we can conclude that the right hand side is nonpositive in the limit and hence the vanishing viscosity weak solution satisfies

$$\int_{t_1}^{t_2} \int_{x_1}^{x_2} \eta(u)_t + \Psi(u)_x \, dxdt \leq 0 \tag{33}$$

for all x_1, x_2, t_1 and t_2. Alternatively, in integral form,

$$\int_{x_1}^{x_2} \eta\left(u(x,t)\right) \, dx \mid_{t_1}^{t_2} + \int_{t_1}^{t_2} \Psi\left(u(x,t)\right) \, dt \mid_{x_1}^{x_2} \leq 0 \tag{34}$$

ie.

$$\begin{aligned} \int_{x_1}^{x_2} \eta\left(u(x,t_2)\right) \, dx \quad \leq \quad & \int_{x_1}^{x_2} \eta\left(u(x,t_1)\right) \, dx - \\ & \left(\int_{t_1}^{t_2} \Psi\left(u(x_2,t)\right) \, dt - \int_{t_1}^{t_2} \Psi\left(u(x_1,t)\right) \, dt \right) \end{aligned} \tag{35}$$

Consequently, the total integral of η is not necessarily conserved, but can only decrease. (Note that our mathematical assumption of convexity leads to an "entorpy function" that decreases, whereas the physical entropy in gas dynamics increases.) The fact that (33) holds for all x_1, x_2, t_1 and t_2 is summerized by saying that $\eta(u)_x + \Psi(u)_x \leq 0$ in the weak sence. This gives our final form of the entropy condition, called the entropy inequality.

Entropy Condition (Version II). The function $u(x,t)$ is the entropy solution of (10) if, for all convex entropy functions and corresponding entropy fluxes, the inequality

$$\eta(u)_t + \Psi(u)_x \leq 0 \tag{36}$$

is satisfied in the weak sence.

This formulation is also useful in analyzing numerical methods. If a discrete form of this entropy inequatlity is known to hold for some numerical method, then it can be shown that the method converges to the entropy solution.

The weak form of the entropy inequality is

$$\begin{aligned} \int_0^{\infty} \int_{-\infty}^{\infty} \left(\phi_t(x,t)\eta\left(u(x,t)\right) + \phi_x(x,t)\Psi\left(u(x,t)\right)\right) \, dxdt \\ \leq - \int_{-\infty}^{\infty} \phi(x,0)\eta\left(u(x,0)\right) \, dx \end{aligned} \tag{37}$$

for all $\phi \in C_0^1(R \times R)$ with $\phi(x,t) \geq 0$ for all x, t.

3 Numerical Methods

3.1 Numerical Methodes for the inviscid Burgers' Equation

We consider the inviscid Burgers' equation

$$u_t + f(u)_x = 0 \qquad f(u) = \frac{1}{2}u^2,\ x \in R,\ t \geq 0 \qquad (38)$$
$$u(x,0) = u_0(x)$$

We discretize the x-t-plane by choosing a mesh with $h \equiv \Delta x$ and a time step $k \equiv \Delta t$ and define the discrete mesh points (x_j, t_n) by

$$x_j = jk \qquad j = \ldots, -1, 0, 1, 2, \ldots$$
$$t_n = nk \qquad n = 0, 1, 2, \ldots$$

It will also be useful to define

$$x_{j+1/2} = x_j + \frac{h}{2} = \left(j + \frac{1}{2}\right) h$$

For simplicity we take a uniform mesh, with h and k constant.

The finite difference methods we will develop produce approximations $U_j^n \in R$ to the solution $u(x_j, t_n)$ at the discrete grid points. The pointwise values of the true solution will be benotet by

$$u_j^n = u(x_j, t_n)$$

In developing methods for conservation laws it is often preferable to view U_j^n as an approximation to a cell average of $u(x, t_n)$ defined by

$$\bar{u}_j^n \equiv \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_n)\, dx \qquad (39)$$

rather than as an approximation to the pointwise value u_j^n.

As initial data for the numerical method we use $u_0(x)$ to define U^0 by pointwise values, $U_j^0 = u_j^0$ or preferable by cell averages $U_j^0 = \bar{u}_j^0$.

We will study explicit 2-level-methods, and introduce some special notation for such methods, writing

$$U^{n+1} = H_k(U^n)$$

Here U^{n+1} represents the vector of approximations U_j^{n+1} at time t_{n+1}. The value U_j^{n+1} at a particular point j typically depends on several values from the vector U_n (depending on how large the stencil is) , and so we write

$$U_j^{n+1} = H_k(U^n; j) \qquad (40)$$

to signify that U_j^{n+1} depends on the full vector U^n. For example, the Backward Euler method for the linear equation $u_t + u_x = 0$ is

$$U_j^{n+1} = U_j^n - \frac{k}{2h}(U_{j+1}^n - U_{j-1}^n) \tag{41}$$

The H_k-Operator takes the form

$$H_k(U^n; j) = U_j^n - \frac{k}{2h}(U_{j+1}^n - U_{j-1}^n) \tag{42}$$

The finite difference operator H_k can be extended to apply to functions of x rather than to discrete grid functions in a very natural way. If $v(x)$ is an arbitrary function of x then we define $H_k(v)$ to be the new function of x obtained by applying the difference scheme to v at each point x. Pointwise values of $H_k(v)$ might be denoted by $[H_k(v)](x)$, or, for consistency with (40), by $H_k(v; x)$ For example, the method (41) to $v(x)$ gives $H_k(v)$ defined by

$$H_k(v; x) = [H_k(v)](x) = v(x) - \frac{k}{2h}\left(v(x+h) - v(x-h)\right) \tag{43}$$

$U_k(., t)$ denotes the function of x alone obtained by fixing t. The operator H_k applies only to functions of one variable.

We define a piecewise constant function $U_k(x, t)$ for all x and from the discrete values U_j^n. We assign this function the value U_j^n in the (j, n) grid cell, ie.

$$U_k(x, t) = U_j^n \quad for\ (x, t) \in [x_{j-1/2}, x_{j+1/2}] \times [t_n, t_{n+1}] \tag{44}$$

We index this function U_k by the time step k, and assume that the mesh width h and the time step k are related in some fixed way, so that the choice of k defines a unique mesh.

Notice in particular that we apply this operator $H_k(v; x)$ to the piecewise constant function $U_k(., t)$ at any time t, we obtain the piecewise constant function $U_k(., t + k)$, so

$$U_k(x, t + k) = H_k\left(U_k(., t); x\right) \tag{45}$$

This demonstrates a certain consistency in our notation, we will use the same symbol H_k to denote both the discrete and continuous operators.

Convergence. We are interested in how well U_j^n approximates the true solution, and we define the global error to be the difference between the true and computed solution. In studying smooth solutions it is ussually most convenient to consider the pointwise error

$$E_j^n = U_j^n - u_j^n \tag{46}$$

For conservation laws it is sometimes preferable to consider the error relative to the cell average of the true solution

$$E_j^n = U_j^n - \bar{u}_j^n \tag{47}$$

This can be unified to some extend by introducing the error function

$$E_k(x,t) = U_k(x,t) - u(x,t) \tag{48}$$

Then E_j^n is the pointwise value $E_k(x_j, t_n)$ while $\bar{E}_j^n$ is the cell average of E_k at time t_n.

With these definitions, we say that a method is **convergent** in some particular norm $\| \, . \, \|$ if

$$\| \, E_k(.,t) \, \| \to 0 \; as \; k \to 0 \tag{49}$$

for any fixed $t \geq 0$ and for all initial data u_0 in some class.

Local Truncation Error. The local truncation error $L_k(x,t)$ is a measure of how well the difference equation models the differential equation locally. It is defined by replacing the approximate solution U_j^n in the difference equation by the true solution $u(x_j, tn)$. Of cource, this true solution of the PDE is only an approximate solution of the difference equation, and how well it satisfies the difference equations gives an indication of how well the exact solution of the difference equation satisfies the differential equation.

DEFINITION 2. For a general 2-level method, we define the **local truncation error** by

$$L_k(x,t) = \frac{1}{k} \left[u(x, t+k) - H_k \left(u(.,t) \right) \right] \tag{50}$$

DEFINITION 3. The method is **consistent**, if

$$\| \, L_k(.,t) \, \| \to 0 \; as \; k \to 0 \tag{51}$$

DEFINITION 4. The method is of **order** p, if for all sufficiently smooth initial data with compact support, there is a constant C_L such that

$$\| \, L_k(.,t) \, \| \leq C_L k^p \quad for \; all \; k < k_0, t \leq T \tag{52}$$

DEFINITION 5. The method is **stable** if for each time T there is a constant C_S and a value $k_0 > 0$ such that

$$\| \, H_k^n \, \| \leq C_S \quad for \; all \; nk \leq T, k < k_0 \tag{53}$$

Here superscripts on H_k represent powers of the (linear) operator H_k.

CFL Condition. A necessary stability condition for any numerical method is that the domain of dependence of the finite difference method should include the domain of dependence of the PDE, at least in the limit $k, h \to \infty$. The condition is known as the CFL condition after Courant, Friedrich and Lewy.

The solution $u(x,t)$ of Burgers' equation (38) at any point $(\bar{x}, \bar{t})$ depends only on the initial data u_0 at a single point, namely the point $\bar{x}_0$ such that $(\bar{x}, \bar{t})$ lies on the characteristic through $\bar{x}_0$. The set $D(\bar{x}, \bar{t}) = \{\bar{x}_0\}$ is called the domain of dependence of the PDE.

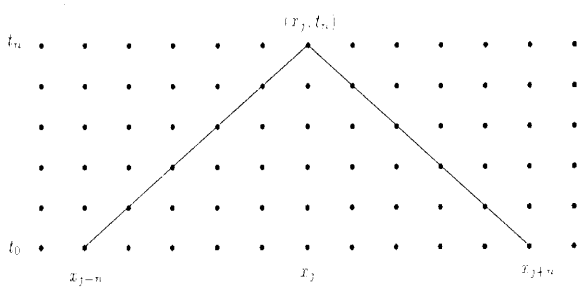

Figure 6: Computational triangle for a three-point stencil. The CFL condition requires that characteristics lie within this triangle

The numerical domain of dependence $D_k(\bar{x}, \bar{t})$, for a particular method is similarly defined. It is the set of points x for which initial data $u_0(x)$ could possibly effect the numerical solution at $(\bar{x}, \bar{t})$

The CFL condition is only a necessary condition for stability, nut sufficient.

Conservative Methods for Nonlinear Problems. If we write Burger' equation (38) in the quasilinear form

$$u_t + uu_x = 0 \tag{54}$$

then a natural finite difference method is

$$U_j^{n+1} = U_j^n - \frac{k}{h}\left(U_j^n - U_{j-1}^n\right) \tag{55}$$

Method (55) is adequate for smooth solutions but will not, in general, converge to a discontinuous weak solution of Burgers' equation (54) as the grid is refined. 7 shows the true and computed soltution of Burgers' equation at time $t = 1$ with Riemann data $u_l = 1.2$ and $u_r = 0.4$. We get a nice looking solution propagating at entirely the wrong speed. It can be shown that method (55) is consistent with (54).

It turns out to be a very simple riquirement we can impose on our numerical method which will guarantee that we do not converge to non-solutions. This is the riquirement that the method be in **conservation form**, which means it has the form

$$U_j^{n+1} = U_j^n - \frac{k}{h}\left[F(U_{j-p}^n, U_{j-p+1}^n, \ldots, U_{j+q}^n) - F(F(U_{j-p-1}^n, U_{j-p}^n, \ldots, U_{j+q-1}^n)\right] \tag{56}$$

for some function F of $p + q + 1$ arguments. F is called the **numerical flux function**. In the simplest case, $p = 0$ and $q = 0$, so that F is a function of only two variables and so (56) becomes

$$U_j^{n+1} = U_j^n - \frac{k}{h}\left[F(U_j^n, U_{j+1}^n) - F(U_{j-1}^n, U_j^n)\right] \tag{57}$$

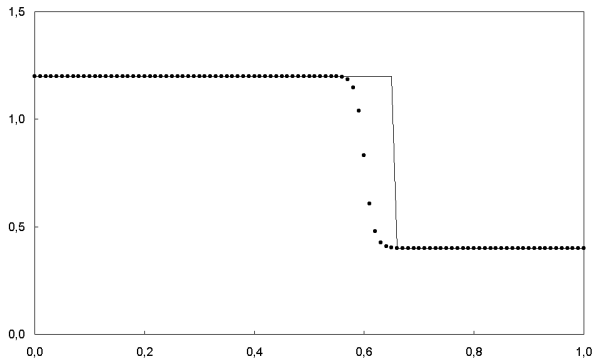

Figure 7: True and computed solution fo Burgers' equation using a nonconservative method

This form is very natural if we view U_j^n as an approximation to the cell average $\bar{u}_j^n$ defined by (39).

One way to derive methods in conservation form is to use standard finite difference discretisations but to start with the conservation form of the PDE rather that the quasilinear form. For example, if we generalize the upwind method to Burgers' equation using the form $u_t + \left(\frac{1}{2}u^2\right)_x = 0$, we obtain

$$U_j^{n+1} = U_j^n - \frac{k}{h}\left[\frac{1}{2}(U_j^n)^2 - \frac{1}{2}(U_{j-1}^n)^2\right] \tag{58}$$

This is of the form (57) with

$$F(u,v) = \frac{1}{2}v^2 \tag{59}$$

We assume that $U_j^n \geq 0$ for all j, n, so that the "upwind" direction is always to the left. More generally, for a nonlinear system $u_t + f(u)_x = 0$ for which the derivative $f'(U_j^n)$ has nonnegative values for all U_j^n, the upwind method is of the form (57) with

$$F(v,w) = f(v) \tag{60}$$

Lax-Friedrich's method to nonlinear equations takes the form

$$U_j^{n+1} = \frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right) - \frac{k}{2h}\left(f(U_{j+1}^n) - f(U_{j-1}^n)\right) \tag{61}$$

This method can be written in conservative form (57) by taking

$$F(U_j, U_{j+1}) = \frac{k}{2h}(U_j - U_{j+1}) + \frac{1}{2}\left(f(U_j) + f(U_{j+1})\right) \tag{62}$$

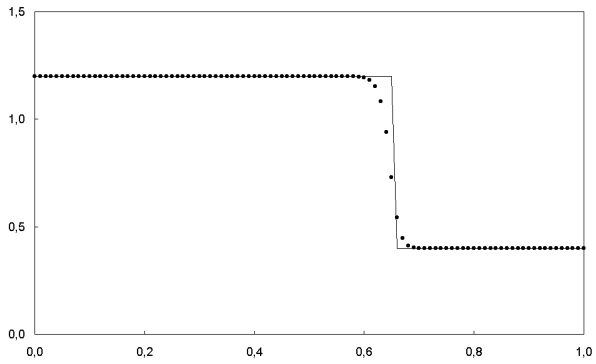

Figure 8: True and computed solution to Burgers' equation using the conservative upwind method

Method (57) is **consistent** with the original conservation law if the numerical flux function F reduces to the true flux f for the case of constant flow. We expect

$$F(\bar{u}, \bar{u}) = f(\bar{u}) \quad \forall \bar{u} \in R \tag{63}$$

Some smoothness is also required, so that as the two arguments of F approach some common value $\bar{u}$, the value of F approaches $f(\bar{u})$ smoothly. For consistency it sufficies to have F a Lipschitz continuous function of each variable. We say that F is Lipschitz at $\bar{u}$ if there is a constant $K \geq 0$ (which may depend on $\bar{u}$ such that

$$\mid F(v, w) - f(\bar{u}) \mid \leq K \ \max(\mid v - \bar{u} \mid, \mid w - \bar{u} \mid) \tag{64}$$

for all v, w with $\mid v - \bar{u} \mid$ and $\mid w - \bar{u} \mid$ sufficiently small. We say that F is a Lipschitz continuous function if it is Lipschitz at every point.

The upwind flux (58) is consistent since it clearly satisfies (63) and is Lipschitz continuous provided f is Lipschitz. It also can be shown that Lax-Friedrich's flux (61) is consistent.

The methods considered above all are first order. The Lax-Wendroff method

$$\begin{aligned} U_j^{n+1} = U_j^n \quad &- \quad \frac{k}{2h}\left(f(U_{j+1}^n) - f(U_{j-1}^n)\right) \\ &+ \quad \frac{k^2}{2h^2}\left[A_{j+1/2}\left(f(U_{j+1}^n) - f(U_j^n)\right) - A_{j-1/2}\left(f(U_j^n) - f(U_{j-1}^n)\right)\right] \end{aligned} \tag{65}$$

where $A_{j\pm 1/2}$ is the derivative of u evaluated at $\frac{1}{2}(U_j^n + U_{j\pm 1}^n)$. The difficulty with this form is that it requires evaluating the derivative, and is more expensive to use than other forms that only use the function $f(u)$.

Lax Wendroff Theorem. The above discussion suggests that we can hope to correctly approximate discontinuous weak solutions to the conservation law

by using a conservative method.

THEOREM 1 (LAX AND WENDROFF). *Consider a sequence of grids indexed by $l = 1, 2, \ldots$, with mesh paramters $k_l, h_l \to 0$ as $l \to \infty$. Let $U_l(x,t)$ denote the numerical approximation computed with a consistent and conservative method on the lth grid. Suppose that U_l converges to a function u as $l \to \infty$, in the sence made precise below. Then $u(x,t)$ is a weak solution of the conservation law.*

We will asumme that we have convergence of U_l to u in the following sence:
1. Over every boundet set $\Omega = [a,b] \times [0,T]$ in $x - t$ space,

$$\int_0^T \int_a^b \mid U_l(x,t) - u(x,t) \mid dxdt \to 0 \quad as\ l \to \infty \tag{66}$$

This is the 1-norm over the set Ω, so we can simply write

$$\parallel U_l - u \parallel_{1,\Omega} \to 0 \quad as\ l \to \infty \tag{67}$$

2. We also assume that for each T there is an $M > 0$ such that

$$TV\left(U_l(.,t)\right) < M \quad fo\ all\ 0 \le t \le T, \quad l = 1, 2, \ldots \tag{68}$$

Here TV denotes the total variation function,

$$TV(v) = \sup \sum_{i=1}^{N} \mid v(\xi_i) - v(\xi_{i-1}) \mid \tag{69}$$

where the supremum is taken over all subdivisions of the real line $-\infty = \xi_0 < \xi_1 < \ldots < \xi_N = \infty$. Note that for the total variaton to be defined v must approach constant values $v_{\pm\infty}$ as $x \to \pm\infty$.
Another possible defintion is

$$TV(v) = \limsup_{\epsilon \to 0} \frac{1}{\epsilon} \int_{-\infty}^{\infty} \mid v(x) - v(x - \epsilon) \mid dx \tag{70}$$

If v is differentialbe then this reduces to

$$TV(v) = \int_{-\infty}^{\infty} \mid v'(x) \mid dx \tag{71}$$

PROOF. Can be found in [1]

This theorem does not guarantee that weak solutions obtained in this manner satisfy the entropy condition. For some numerical methods, it is possible to show that this can never happen: We use the numerical solution U^n to define a piecewise constant function $\tilde{u}^n(u, t_n)$ for $t_n \le t \le t_{n+1}$. The equation can

be solved exactly over a short time interval because the initial data $\tilde{u}^n(u, t_n)$ is piecewise constant, and hence defines a sequence of Riemann problems. The exact solution, up to the time when waves from neighbouring Riemann problems begin to interact, is obtained by simply piecing together these Riemann solutions. This will be true if $k \cdot v \leq h/2$, where $v = \sup_{|\xi| \leq |u|} \mid f'(\xi) \mid$. This yields

$$\frac{k}{h} f'(U_j^n) \leq \frac{1}{2} \Longleftrightarrow f'(U_j^k) \leq \frac{h}{2k} \tag{72}$$

After obtaining this solution over the interval $[t_n, t_{n+1}]$ we define the approximate solution U^{n+1} at time t_{n+1} by averaging this exact solution at time t_{n+1}

$$U_j^{n+1} = \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} \tilde{u}^n(x, t_{n+1})\, dx \tag{73}$$

These values are then used to define new piecewise constant data $\tilde{u}^{n+1}(x, t_{n+1})$ and the process repeats.

Since $\tilde{u}^n$ is assumed to be an exact weak solution, we know that

$$\begin{aligned} \int_{x_{j-1/2}}^{x_{j-1/2}} \tilde{u}^n(x, t_{n+1})\, dx \quad &= \quad \int_{x_{j-1/2}}^{x_{j-1/2}} \tilde{u}^n(x, t_n)\, dx + \\ &\qquad \int_{t_n}^{t_{n+1}} f\left(\tilde{u}^n(x_{j-1/2}, t)\right) dt \\ - \int_{t_n}^{t_{n+1}} f\left(\tilde{u}^n(x_{j+1/2}, t)\right) dt & \end{aligned} \tag{74}$$

Dividing bei h, using (73), and noting that $\tilde{u}^n(x, t_n) \equiv U_j^n$ over the cell $(x_{j-1/2}, x_{j+1/2})$, this equation reduces to

$$U_j^{n+1} = U_j^n - \frac{k}{h} \left[F(U_j^n, U_{j+1}^n) - F(U_{j-1}^n, U_j^n)\right] \tag{75}$$

where the numerical flux function F is given by

$$F(U_j^n, U_{j+1}^n) = \frac{1}{k} \int_{t_n}^{t_{n+1}} f\left(\tilde{u}^n(x_{j+1/2}, t)\right) dt \tag{76}$$

This shows that Godunov's method can be written in conservative form. Moreover, note that the integral (76) we need to compute is trivial because $\tilde{u}^n$ is constant at the point $x_{j+1/2}$ over the time interval (t_n, t_{n+1}). This follows from the fact, that the solution of the Riemann problem at $x_{j+1/2}$ is a similarity solution[1] constant along each ray $(x - x_{j+1/2})/t = constant$.

[1] If we put the initial discontinuity of a Riemann problem at $x = 0$, then the resulting solution $u(x, t)$ is a "similarity solution" in the variable x/t, meaning that $u(x, t)$ can be expressed as a function of x/t alone, say $u(x, t) = w(x/t)$. If follows that $u(x, t) = u(\alpha x, \alpha t)$ for any $\alpha > 0$, so the solution at two different times t and αt look the same if we rescale the x-axis. This also means that the waves move at a constant speed and the solution $u(x, t)$ is constant along any ray $x/t = constant$ in the $x - t$ plane.

The constant value of $\tilde{u}^n$ along the line $x = x_{j+1/2}$ depends only on the data U_j^n and U_{j+1}^n for this Riemann problem. If we denote this value by $u^*(U_j^n, U_{j+1}^n)$, then the flux (75) reduces to

$$F(U_j^n, U_{j+1}^n) = f\left(u^*(U_j^n, U_{j+1}^n)\right) \tag{77}$$

and Godunov's method becomes

$$U_j^{n+1} = U_j^n - \frac{k}{h}\left[f\left(u^*(U_j^n, U_{j+1}^n)\right] - f\left(u^*(U_{j-1}^n, U_j^n)\right)\right] \tag{78}$$

Note that the flux (77) is consistent with f since if $U_j^n = U_{j+1}^n \equiv \bar{u}$ then $u^*(U_j^n, U_{j+1}^n) = \bar{u}$ as well. Lipschitz continuty follows from smoothness of f.

Recall the Riemann problem with data u_l, u_r.

$$F(u_l, u_r) = f(u^*(u_l, u_r) \tag{79}$$

In the convex case there are four cases that must be considered:

1. $f'(u_l), f'(u_r) \geq 0 \quad \Longrightarrow \quad u^* = u_l$
2. $f'(u_l), f'(u_r) \leq 0 \quad \Longrightarrow \quad u^* = u_r$
3. $f'(u_l) \geq 0 \geq f'(u_r) \quad \Longrightarrow \quad u^* = u_l$ if $[f]/[u] > 0$ or $u^* = u_r$ if $[f]/[u] < 0$
4. $f'(u_l) < 0 < f'(u_r) \quad \Longrightarrow \quad u^* = u_s$ (transonic rarefaction)

Note in particular that in Cases 1 and 2 it is irrelevant whether the solution is a shock or a rarefaction, since the value of u^* is the same in either case. This shows that using Godunov's method with entropy-violating Riemann solutions does not necessarily converge to entropy-violating numerical solutions.

It is only in Case 4, the transonic rarefaction, u^* is neither u_l nor u_r, but is the intermediate value u_s, with the property that $f'(u_s) = 0$. This is the value of u for which the characteristic speed is zero, and is called the sonic point.

The function $\tilde{u}^n(x,t)$ for $t_n \leq t \leq t_{n+1}$ is assumed to be a weak solution of the conservation law. In situations where this weak solution is not unique, there may be several choices for $\tilde{u}^n(x,t)$. Different choices may give different values of $u^*(U_j^n, u_{j+1}^n)$ and hence different numerical solutions. Godunov's method is conservative and consistent regardless of what choice we make. It makes sence to use the entropy-satisfying weak solution for $\tilde{u}^n$ in each time step. In [1] it is shown that Godunov's method satisfies a discrete version of the entropy condition. Weak solutions obtained by Godunov's method satisfy the entropy condition, provided we use entropy-satisfying Riemann solvers.

Nonlinear Stability. The Lax-Wendroff Theorem does not say anything about whether the method converges, only if a sequence of approximations converges then the limit is a weak solution. To guarantee convergence, we need some form of stability.

The global error $U_k(x,t) - u(x,t)$ is not well defined when the weak solution u is not unique. Instead, we measure the global error in our approximation by the distance from $U_k(x,t)$ to the set of *all* weak solutions W,

$$W = \{w : \ w(x,t) \textit{ is a weak solution to the conservation law}\} \tag{80}$$

To measure this distance we need a norm, for example the 1-norm over some finite time interval $[0,T]$, denoted by

$$\begin{aligned} \| \ v \ \|_{1,T} \qquad &= \int_0^T \| \ v(.,t) \ \|_1 \ dt \\ &= \int_0^T \int_{-\infty}^{\infty} \mid v(x,t) \mid dxdt \end{aligned} \tag{81}$$

The global error is then defined by

$$dist(U_k, W) = \inf_{w \in W} \| \ U_k - w \ \|_{1,T} \tag{82}$$

The convergence result we would like to prove takes the following form: *If U_k is generated by a numerical method in conservation form, consistent with the conservation law, and if the method is stable in some appropriate sence, then*

$$dist(U_k, W) \to 0 \quad as \ k \to 0$$

In order to prove a convergence result of the type formulated above, we must define an appropriate notion of "stability". For nonlinear problems the primary tool used to prove convergence is *compactness*. The most important property of compact sets, in relation to our goals of defining stability and proving convergence, is the following.

PROPOSITION 1. If K is a compact set in some normed space, then any infinite sequence of elements of K, $\{k_1, k_2, k_3, \ldots\}$, contains a subsequence which converges to an element of K.

The fact that compactness guarantees the existence of convergent subsequences, combined with the Lax-Wendroff Theorem, will give us a convergence proof of the type formulated above.

Since we are interested in proving the convergence of a sequence of functions $U_k(x,t)$, our definition of stability will require that all the functions lie within some compact set in some normed function space. Restricting our attention to the time interval $[0,T]$, the natural function space is the space $L_{1,T}$, consisting of all functions of x and t for which the $1,T$-norm (81) is finite.

$$L_{1,T} = \{v : \| \ v \ \|_{1,t} < \infty\}$$

This is an infinite dimensional space and so it is not immediately clear if this space includes a compact subset.

In order to obtain a compact set in L_1, we will put a bound on the total variation of the functions, a quantity already defined in (69) through (71). The set

$$\{v \in L_1 : \ TV(v) \leq R \textit{ and } Supp(v) \subset [-M, M]\} \tag{83}$$

is a compact set, and any sequence of functions with uniformly bounded total variation and support must contain convergent subsequences.

Since our numerical approximations U_k are functions of x and t, we need to bound the total variation in both space and time. We define the total variation over $[0,T]$ by

$$\begin{aligned} TV_T(u) &= \limsup_{\epsilon\to 0} \frac{1}{\epsilon}\int_0^T \int_{-\infty}^{\infty} \mid u(x+\epsilon,t) - u(x,t) \mid dxdt \\ &\quad + \limsup_{\epsilon\to 0} \frac{1}{\epsilon}\int_0^T \int_{-\infty}^{\infty} \mid u(x,t+\epsilon) - u(x,t) \mid dxdt \end{aligned} \tag{84}$$

It can be shown that the set

$$K = \{u \in L_{1,T} : \ TV_T(u) \le R \ and \ Supp(u(.,t)) \subset [-M,M] \ \forall t \in [0,T]\} \tag{85}$$

is a compact set in $L_{1,T}$.

DEFINITION 6. We will say that a numerical method is total variation stable, or simply **TV-stable**, if all approximations U_k for $k < k_0$ lie in some fixed set of the form (83) (where R and M may depend on the initial data u_0 and the flux function $f(u)$, but not on k).

Note that our requirement in (83) that $Supp(u)$ be uniformly bounded over $[0,T]$ is always satisfied for any explicit method if the initial data u_0 has compact support and k/h is constant as $k \to 0$. This follows from the finite speed of propagation under such a method.

The other requirement for TV-stability can be simplified considerably by noting the follwing theorem. This says that for the special case of functions generated by conservative numerical methods, it sufficies to insure that the one-dimenisional total varation at each time T_n is uniformly bounded (independent of n). Uniform boundedness of TV_T then follows.

THEOREM 2. Consider a conservative method with a Lipschitz continuous numerical flux $F(U;j)$ and suppose that for each initial data u_0 there exists some $k_0, R > 0$ such that

$$TV(U^n) \le R \quad \forall n,k \ with \ k < k_0, \ nk \le T \tag{86}$$

Then the method is TV-stable.

PROOF. A proof can be found in [1].

We are now able to prove our convergence theorem, which requires total variation stability along with consistency.

THEOREM 3. Suppose U_k is generated by a numerical method in conservation form with a Lipschitz continuous numerical flux, consistent with some

scalar conservation law. If the method is TV-stable, ie. if $TV(U^n)$ is uniformly bounded for all n, k with $k < k_0$, $nk \leq T$, then the method is convergent, ie. $dist(U_k, W) \to 0$ as $k \to 0$.

PROOF. A proof can be found in [1].

We have just seen that TV-stability of a consistent and conservative numerical method is enough to guarantee convergence, in the sence that $dist(U_k, W) \to 0$ as $k \to 0$.

One easy way to ensure TV-stability is to require that the total variation be nonincreasing as time evolves, so that the total variation at any time is uniformly bounded by the total variation of the initial data. The requirement gives rise to the very important class of TVD methods.

DEFINITION 6. The numerical method $U_j^{n+1} = H(U^n; j)$ is called total variation diminishing (abbreviated **TVD**) if

$$TV(U^{n+1}) \leq TV(U^n) \tag{87}$$

for all grid functions U^n.

It can be shown that the true solution to a scalar conservation law has this TVD property, ie. any weak solution $u(x, t)$ satisfies

$$TV(u(., t_2)) \leq TV(u(., t_1)) \quad for\ t_2 \geq t_1 \tag{88}$$

If this were not the case then it would be impossible to develope a TVD numerical method. However, since true solutions are TVD, it is reasonalbe to impose this requirement on the numerical solution as well, yielding a TV-stable and hence convergent method. A number of very successful numerical methods have been developed using this requirement.

One difficulty associated with numerical approximations of discontinuous solutions is that oscillations may appear near the discontinuity. In an attempt to eliminate this possibility, one natural requirement we might place on a numerical method is that it be monotonicity preserving. This means that if the initial data U_j^0 is monotone as a function of j, then the solution U_j^n should have the same property for all n. This means in particular that oscillations cannot arise near isolated propagating discontinuity, since Riemann initial data is monotone. For TVD methods we habe this property.

THEOREM 4. Any TVD method is monotonicity preserving.

PROOF. A proof can be found in [1].

Any weak solution of a scalar consevation law satisfies

$$\| u(., t_2) \|_1 \leq \| u(., t_1 \|_1 \quad for\ t_2 \geq t_1) \tag{89}$$

In particular, if u_0 is the initial data at time $t = 0$, then

$$\| u(.,t) \|_1 \leq \| u_0 \|_1 \quad \forall t \geq 0 \tag{90}$$

The result (89) is true for any weak solution to a scalar conservation law. If we restrict our attention to the entropy solution (which is unique) then (89) is a special case of a more general result. If $u(x,t)$ and $v(x,t)$ are both entropy solutions of the same scalar conservation law (but with possibly different initial data), and if $u_0 - v_0$ has compact support (so that $\| u(.,t) - v(.,t) \|_1 < \infty$ for all t) then

$$\| u(.,t_2) - v(.,t_2) \|_1 \leq \| u(.,t_1) - v(.,t_1) \|_1 \quad for\ t_2 \geq t_1 \tag{91}$$

This property is called L_1-**contraction**: $u - v$ is contracing in the 1-norm as time evolves. We can derive (88) von (90) by taking $v(x,t) \equiv 0$, shich is an entropy solution to any conservation law.

For grid functions $U = \{U_j\}$ (at fixed time, for example) we define the 1-norm by

$$\| U \|_1 = h \sum_{j=-\infty}^{\infty} | U_j | \tag{92}$$

The space l_1 consists of all grid functions for which the 1-norm is finite:

$$l_1 = \{U : \| U \|_1 < \infty\} \tag{93}$$

Note that if we extend the grid function U to a piecewise constant function $\tilde{u} = U_j$ for all $x_{j-1/2} \leq x \leq x_{j+1/2}$, then

$$\| U \|_1 = \| \tilde{u} \|_1 \tag{94}$$

Conversely, if we take a function $u(x)$ and restrict it to a grid function U by setting $U_j = \bar{u}_j$, the cell average , then

$$\| U \|_1 \leq \| u \|_1 \tag{95}$$

In analogy to the L_1-contracton property of he true solution operator, we say that a numerical method

$$U_j^{n+1} = H(U^n; j) \tag{96}$$

is l_1-contracing if, for any two grid functions U^n and V^n for which $U^n - V^n$ has compact support, the grid functions U^{n+1} and V^{n+1} defined by (96) and $V_j^{n+1} = H(V^n; j)$ satisfy

$$\| U^{n+1} - V^{n+1} \|_1 \leq \| U^n - V^n \|_1 \tag{97}$$

The fact that l_1-contracing methods are convergent follows from the next theorem and our previous results.

THEOREM 5. Any l_1-contracting numerical method is TVD.

PROOF. A proof can be found in [1].

Another useful property of the entropy-satisfying weak solution is the following: If we take two sets of initial data u_0 and v_0, with

$$v_0(x) \geq u_0(x) \quad \forall x \tag{98}$$

then the respective entropy solution $u(x,t)$ and $v(x,t)$ satisfy

$$v(x,t) \geq u(x,t) \quad \forall x,t \tag{99}$$

The numerical method $U_j^{n+1} = H(U^n; j)$ is called a **monotone method** if the analogous property holds:

$$V_j^n \geq U_j^n \quad \forall j \qquad \Longrightarrow \qquad V_j^{n+1} \geq U_j^{n+1} \quad \forall j \tag{100}$$

To prove that a method is monotone, it sufficies to check that

$$\frac{\partial}{\partial U_j^n} H(U^n; j) \geq 0 \quad for\ all\ i, j, U^n \tag{101}$$

This means that if we increase the value of any U_j^n then the value of U_j^{n+1} cannot decrease as a result.

THEOREM 6. Any monotone method is l_1-contracing.

PROOF. A proof can be found in [1].

To summerize the relation between the different types of methods considered above, we have:
monotone $\Longrightarrow$ l_1-contracing $\Longrightarrow$ TVD $\Longrightarrow$ monotoniciy preserving

High Resolutions Methods It can be shown that monotone methods are at most first order accurate. Now, we will study some "high resolution" methods. This term applies to methods that are at least second order accurate on smooth solutions and give well resolved, nonoscillatory discontinuities. The main idea behind any high resolution method is to attempt to use a high order method, but to modify the method and increase the amount of numerical dissipation in the neighbourhood of a discontinuity.

The basic idea of **slope-limiter methods** is to generlize Godunov's method by replacing the piecewise constant representation of the solution by some more accurate representation, say piecewise linear.

Recall that Godunov's method can be viewed as consisting of the following three steps (although it is not typically implemented this way)

ALGORITHM 1.

1. **Reconstruct.** Given data $\{U_j^n\}$, construct a function $\tilde{u}^n(x,t_n)$. (Piecewise constant in Godunov's method.)
2. **Solve.** Solve the vonservation law exactly with this data to obtain $\tilde{u}^n(x,t_{n+1})$.
3. **Average.** Compute cell averages of the resulting solution to obtain U_j^{n+1}

To generalize this procedure, we replace Step 1 by more accurate reconstruction, taking for example the piecewise linear function

$$\tilde{u}^n(x,t_n) = U_j^n + \sigma_j^n(x-x_j) \quad on\ the\ cell\ [x_{j-1/2}, x_{j+1/2}] \tag{102}$$

Here σ_j^n is a slope on the jth cell which is based on the data U^n. Note that taking $\sigma_j^n = 0$ for all j and n recovers Godunov's method.

The cell average of $\tilde{u}^n(x,t_n)$ from (102) over $[x_{j-1/2}, x_{j+1/2}]$ is equal fo U_j^n for any choice of σ_j^n. Since Steps 2 and 3 are also conservative, the overall method is conservative for any choice of σ_j^n.

For nonlinear problems we will generally not be able to perform Step 2 exactly. The constuction of the exact solution $\tilde{u}^n(x,t_n)$ based on solving Riemann problems no longer works when $\tilde{u}^n(x,t_n)$ is piecewise linear. I will show a suitable way to approximate the solution later.

The most interesting question is how do we choose the slopes σ_j^n?
One simple choice of slopes satisfying

$$TV(\tilde{u}^n(.,t_n)) \le TV(U^n) \tag{103}$$

is the so-called **minmod slope**,

$$\sigma_j = \frac{1}{h} minmod(U_{j+1} - U_j, U_j - U_{j-1}) \tag{104}$$

where the minmod function is defined by

$$minmod(a,b) \ = \ \left\{ \begin{array}{ll} a & if \ \mid a \mid < \mid b \mid \ and\ ab > 0 \\ b & if \mid b \mid < \mid a \mid \ and\ ab > 0 \\ 0 & if\ ab \le 0 \end{array} \right\} \tag{105}$$

In attempting to apply Algorithm 1 to a nonlinear problem, the principle difficulty is in Step 2, since we typically cannot compute the exact solution to the nonlinear equation with piecewise constant initial data. However there are various ways to obtain approximate solutions which are sufficiently accurate that second order accurancy can be maintained.

Here I will present the main idea in a simplified form, under the assumption that the data is monotone and that $f'(u)$ does not change sign over the range of data. Moreover we will impose the time step restriction

$$\frac{k}{h} \max \mid f'(U_j^n) \mid < \frac{1}{2} \tag{106}$$

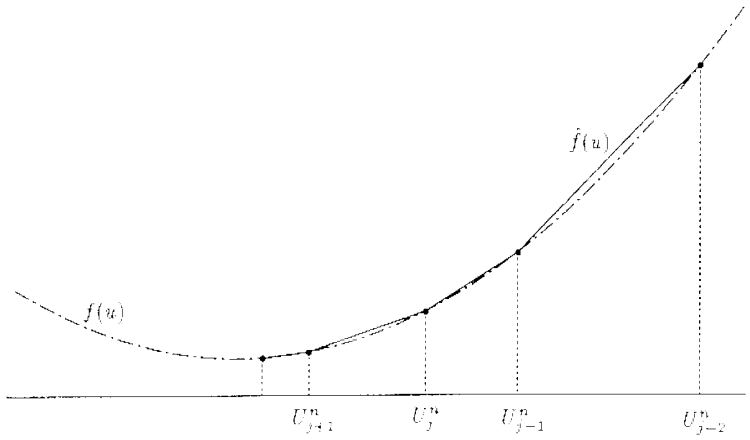

Figure 9: $\hat{f}(u)$ is a piecewise linear approximation of the flux obtained by interpolation at the grid values

With the above assumptions on the data, we can define a piecewise linear function $\hat{f}(u)$ by interpolation the values $(U_j^n, f(U_j^n))$ as in 9. We nowe define $\hat{u}^n(x,t)$ by solving the conservation law

$$\hat{u}_t + \hat{f}(u)_x = 0 \tag{107}$$

for $t_n \leq t \leq t_{n+1}$, with the piecewise linear data (102). The evolution of $\hat{u}^n(x,t)$ is indicated in Figure 10. The flux is still nonlinear, but the nonlinearity has been concentrated at the points U_j^n. Shock form immediately at the points x_j, but because of the time step restriction (106), these shocks do not reach the cell boundary during the time step. Hence we can easily compute the numerical flux

$$F(U^n; j) = \frac{1}{k} \int_{t_n}^{t_{n+1}} \hat{f}\left(\hat{u}^n(x_{j+1/2}, t)\right)\, dt \tag{108}$$

At the cell boundary $x_{j+1/2}$, the solution values lie between U_j^n and U_{j+1}^n for $t_n \leq t \leq t_{n+1}$ and hence

$$\hat{f}\left(\hat{u}^n(x_{j+1/2}, t)\right) = f(U_j^n) + \left(\hat{u}^n(x_{j+1/2}, t) - U_j^n\right) \hat{a}_j \tag{109}$$

where

$$\hat{a}_j = \frac{f(U_{j+1}) - f(U_j)}{U_{j+1} - U_j} \tag{110}$$

The conservation law $\hat{u}_t + \hat{f}(u)_x = 0$ reduces to the advection equation $\hat{u}_t + \hat{a}_j \hat{u}_x = 0$ near $x_{j+1/2}$ and so

$$\begin{aligned} \hat{u}(x_{j+1/2}, t) &= \hat{u}(x_{j+1/2} - (t - t_n)\hat{a}_j, t_n) \\ &= U_j^n + \left(\frac{h}{2} - (t - t_n)\hat{a}_j\right) \sigma_j^n \end{aligned} \tag{111}$$

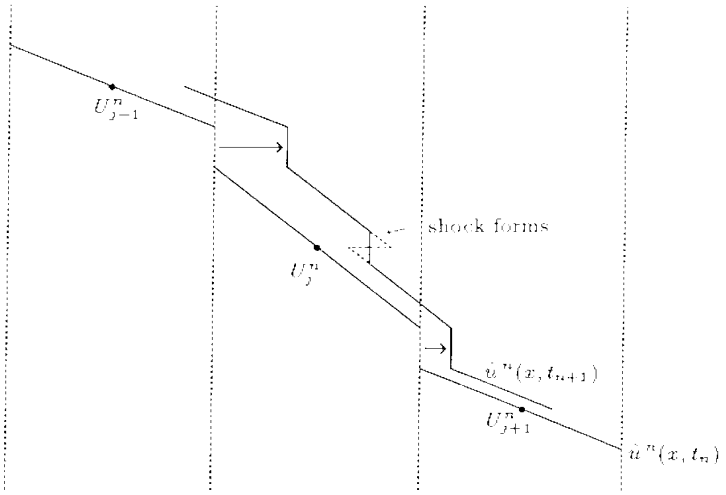

Figure 10: Evolution of the piecewise linear data with the conservation law with piecewise linear flux function $\hat{f}(u)$

Finally, again using (109), we compute the numerical flux (108) to be

$$\begin{aligned} F(U^n; j) &= \frac{1}{h}\int_{t_n}^{t_{n+1}} f(U^n_j) + \left(\frac{h}{2} - (t - t_n)\hat{a}_j\right)\sigma^n_j \, dt \\ &= f(U^n_j) + \frac{1}{2}\hat{a}_j\left(1 - \frac{k}{h}\hat{a}_j\right)h\sigma^n_j \end{aligned} \tag{112}$$

Note that although we do not solve our original conservation law exactly in Step 2 of Algorithm 1, we do obtain $\hat{u}^n$ as the exact solution to a modified conservation law and hence $\hat{u}^n$ is total variation diminishing. By using slope limiter that enforces (103) we obtain an overall method for the nonlinear scalar problem that is TVD.

3.2 Numerical Methodes for the viscid Burgers' Equation

The viscid Burgers' equation

$$u_t + f(u)_x = Du_{xx} \quad f(u) = \frac{1}{2}u^2, \; t > 0, \; D > 0 \tag{113}$$

is a parabolic PDE. Solutions to (113) always are smooth.

If we formally integrate (113) from $x_{j-1/2}$ to $x_{j+1/2}$ and rewrite the equation, we obtain

$$\int_{x_{j-1/2}}^{x_{j+1/2}} u_t \, dx - [Du_x]_{x_{j-1/2}}^{x_{j+1/2}} = -[f(u)]_{x_{j-1/2}}^{x_{j+1/2}} \tag{114}$$

If u_{tx} is continuous on $[x_{j-1/2}, x_{j+1/2}]$ then by the mean value theorem

$$u_t = u_t(x_j, t) + u_{tx}\left(\xi(x), t\right)(x - x_j) \tag{115}$$

for some function $\xi(x)$, $x_{j-1/2} \leq \xi(x) \leq x_{j+1/2}$. But since u_{tx} is continuous, there exists an M such that $\mid u_{tx}\left(\xi(x), t\right) \mid \leq M$ on $[x_{j-1/2}, x_{j+1/2}]$. Therefore

$$\int_{x_{j-1/2}}^{x_{j+1/2}} u_t \, dx = \frac{du}{dt}(x_j, t)\Delta x + O\left((\Delta x)^2\right)$$

If we assume that u_{xxx} is continuous, then

$$u_x(x_{j+1/2}) = \frac{u(x_{j+1}, t) - u(x_j, t)}{\Delta x} + O\left((\Delta x)^2\right)$$

The right side of equation (114) is

$$= -[f(u)]_{x_{j-1/2}}^{x_{j+1/2}} = -f(u(x_{j+1/2}, t)) + f(u(x_{j-1/2}, t))$$

Substituting these expressons into (114) and dividing by Δx yields

$$\begin{aligned} \frac{du}{dt}(x_j, t) &- D(x_{j+1/2}, t)\frac{u(x_{j+1}, t) - u(x_j, t)}{(\Delta x)^2} \\ &+ D(x_{j-1/2}, t)\frac{u(x_j, t) - u(x_{j-1}, t)}{(\Delta x)^2} \\ &= \frac{-f(u(x_{j+1/2}, t)) + f(u(x_j - 1/2, t))}{\Delta x} + O(\Delta x) \end{aligned} \tag{116}$$

Now let $\{U_j(t)\}$ denote the set of approximations to the functions $\{u(x_j, t)\}$, obtained by dropping the $O(\Delta x)$ term in (116). This means that the functions $\{U_j(t)\}$ are required to satisfy the system of ordinary differential equations

$$\begin{aligned} \frac{dU_j}{dt} &- \frac{D(x_{j+1/2}, t)\left(U_{j+1} - U_j\right) + D(x_{j-1/2}, t)\left(U_j - U_{j-1}\right)}{(\Delta x)^2} \\ &= -f(U_{j+1/2}) + f(U_{j-1/2}) \end{aligned} \tag{117}$$

As a computational method we discretize the time derivative in (117) by a forward difference to obtain

$$\begin{aligned} \frac{U_j^{n+1} - U_j^n}{\Delta t} &- \frac{D_{j+1/2}^n\left(U(x_{j+1}^n - U_j^n\right) + D_{j-1/2}\left(U_j^n - U_{j-1}^n\right)}{(\Delta x)^2} \\ &= \frac{-f_{j+1/2}^n + f_{j-1/2}^n}{\Delta x} \end{aligned} \tag{118}$$

The variable U_j^n approximates $u(x_j, t_n)$. $D_{j+1/2}^n = D(x_{j+1/2}, t_n)$, $f_{j+1/2}^n = f(u_{j+1/2}, t_n)$ and so on.

The explicit method takes the form

$$\begin{aligned} U_j^{n+1} &= U_j^n \\ &+ \Delta t \left(\frac{D_{j+1/2}^n \left(U(x_{j+1}^n - U_j^n \right) + D_{j-1/2} \left(U_j^n - U_{j-1}^n \right)}{(\Delta x)^2} - \frac{-f_{j+1/2}^n + f_{j-1/2}^n}{\Delta x} \right) \end{aligned} \tag{119}$$

For a constant function D equation (119) reduces to

$$U_j^{n+1} = U_j^n + \Delta t \left(\frac{D \left(U_{j+1}^n - 2U_j^n + U_{j-1}^n \right)}{(\Delta x)^2} - \frac{-f_{j+1/2}^n + f_{j-1/2}^n}{\Delta x} \right) \tag{120}$$

4 Results

EXAMPLE 1. Consider the inviscid Burgers' Equation (38) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 0.25 & x < 0.25 \\ -x + 0.5 & 0.25 \leq x \leq 0.5 \\ 0 & x > 0.5 \end{array} \right\} \tag{121}$$

Example 1 shows one characteristic property of a hyperbolic conservation law. Even starting with continuous (or smooth) initial data, a shock forms.

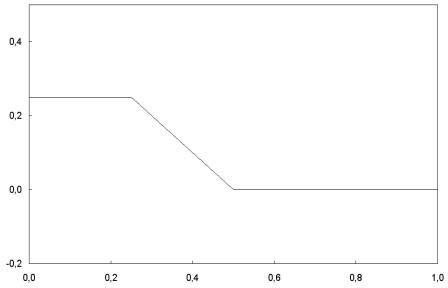

Figure 11: Initial data

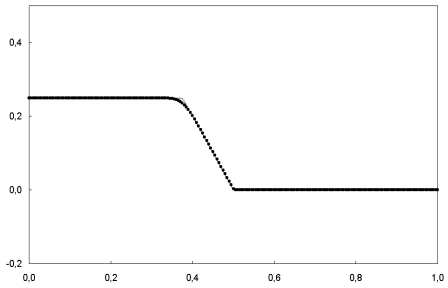

Figure 12: True and computed solution at time $t = 0.5$ using the conservative upwind method (58)

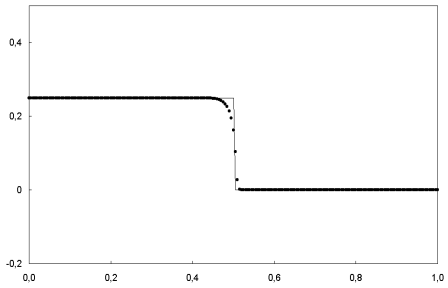

Figure 13: True and computed solution at time $t = 1$ using the conservative upwind method (58)

EXAMPLE 2. Consider the inviscid Burgers' Equation (38) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 1.2 & x \leq 0.25 \\ 0.4 & x > 0.25 \end{array} \right\} \tag{122}$$

Example 2 shows the true and computed solution to this Riemann problem. The natural finite different method (55) computes a solution propagating with a wrong speed. The conservative upwind method (58) computes an approximation to the true solution.

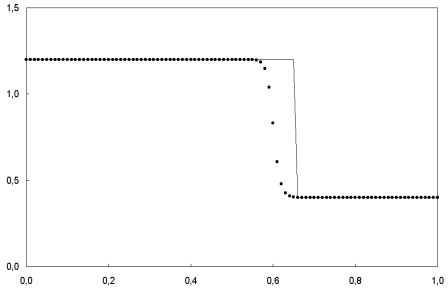

Figure 14: True and computed solution using a natural finite difference method (55)

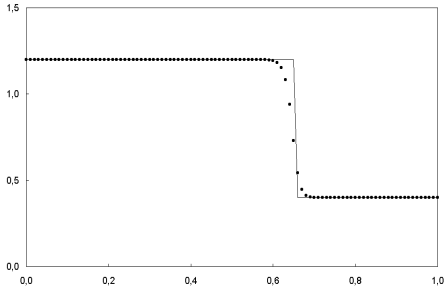

Figure 15: True and computed solution using the conservative upwind method (58)

EXAMPLE 3. Consider the inviscid Burgers' Equation (38) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 1.2 & x \leq 0.25 \\ 0.4 & x > 0.25 \end{array} \right\} \tag{123}$$

Example 3 compares different numerical methods solving this Riemann problem.

The upwind method $U_j^{n+1} = U_j^n - \frac{k}{h}\left[\frac{1}{2}(U_j^n)^2 - \frac{1}{2}(U_{j-1}^n)^2\right]$ is the appropriate one for our initial value problem, because our shock travels to the right. There is no influence on the values U_j^n for $x_j \leq 0.25$, because $\frac{1}{2}(U_j^n)^2 - \frac{1}{2}(U_{j-1}^n)^2 \equiv 0$ for all $x_j \leq 0.25$. If a shock would travel to the left, one should use a downwind method, ie. $U_j^{n+1} = U_j^n - \frac{k}{h}\left[\frac{1}{2}(U_{j+1}^n)^2 - \frac{1}{2}(U_j^n)^2\right]$.

The upwind method uses two values (U_{j-1}^n and U_j^n) for the caluculation of U_j^{n+1}. Therefore the upwind method has got a two point stencil. Lax-Friedrich's Method has got an three point stencil, because for the caluclation of U_j^{n+1}, U_{j-1}^n and U_{j+1}^n are used. While calculating a travelling shock, there occur two problems: First, if we compare Lax-Friedrich with the upwind method, the first part is $\frac{1}{2}(U_{j-1}^n + U_{j+1}^n)$ rather then U_j^n. This is good for smooth solutions, but leads to more smearing out for discontinuities. Second, because of the three point stencil also U_j^n left of the shock are changed. In the first time step the first point left of the shock will change to a lower value. In the next time step the second value left of the initial shock changes to a lower value. Summing up this two facts it can be understood that Lax-Friedrich leads to more smearing out right of the shock, and that the values left of the shock also change, compared with the upwind method.

Godunov's Method also has got a three point stencil. Considering the accurancy there is almost no difference to the upwind method, which is clear if we look on the definition of Godunov's method.

Second order methods, such as Lax-Wendroff give oscillatory approximations to discontious solutions. This can be easily understood using the interpretation of Algorithm 1. Reconstructing a piecewise linear function $\tilde{u}^n(x, t_n) = U_j^n + \sigma_j^n(x - \bar{x}_j)$ for $x_j \leq x < x_{j+1}$ with slope $\sigma_j^n = \frac{U_{j+1}^n - U_j^n}{h}$ in the j^{th} cell, solving the hyperbolic conservation law to obtain $\tilde{u}^n(x, t_{n+1})$ a time Δt later. Finally we average this function over each grid cell to obtain $U_j^{n+1} = \frac{1}{h}\int_{C_j} \tilde{u}^n(x, t_{n+1})dx$. Use [3] to see that this procedure is equivalent to Lax-Wendroff method. Consider Lax-Wendroff method applied to piecewise constant data. Choosing slopes in each grid cell gives the piecewise linear function shown in 16(a). The function $\tilde{u}^n(x, t_n)$ has an overshoot. When we advance this profile and then compute the average over the J^{th} cell, we will get a value that is greater then the initial value. In the next time step this overshoot will be accentuated, while in cell $J - 1$ we will now have a positive slope, leading to a value that is less than 1. This oscillation then grows with time.

The slope proposed before was based on the assumption that the solution is smooth. Near a discontinuity there is no reason to believe that introducing this slope will improve the accurancy. If one of our goals is to avouid nonphysical oscillations, then we must modify the slope in the J^{th} cell. Our high resolution

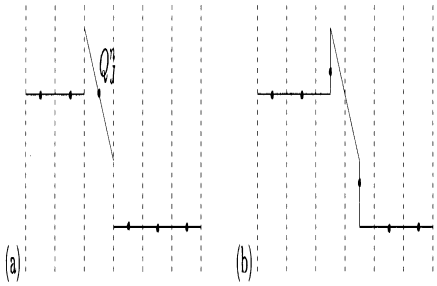

Figure 16: (a) Grid values U^n and reconstructed $\tilde{u}^n(., t_n)$. (b) After advancing in time the dots show the new cell averages U^{n+1}. Note the overshoot.

mehtod, the slope limiter mehtod shows the improvement which can be acchived using a modified slope.

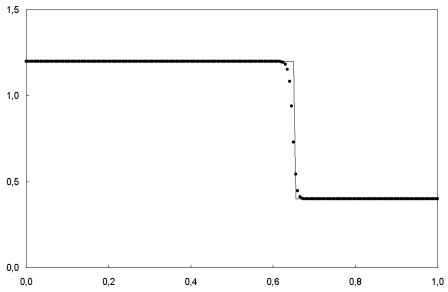

Figure 17: True and computed solution using the upwind method (58)

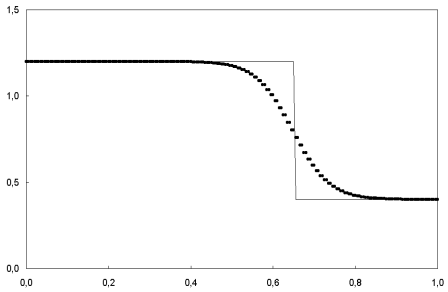

Figure 18: True and computed solution using Lax-Friedrichs method (60)

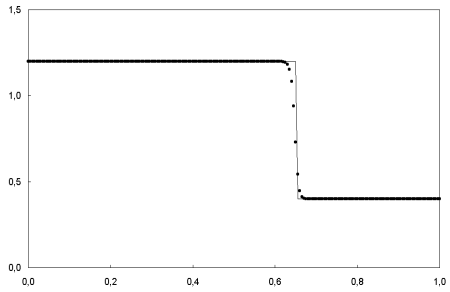

Figure 19: True and computed solution using Godunov's method

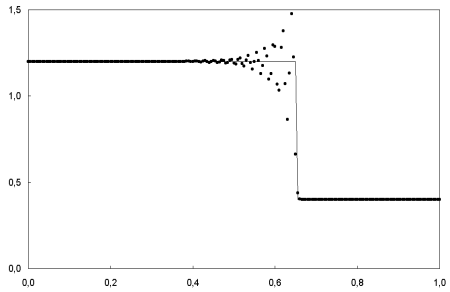

Figure 20: True and computed solution using Lay-Wendroff method (64)

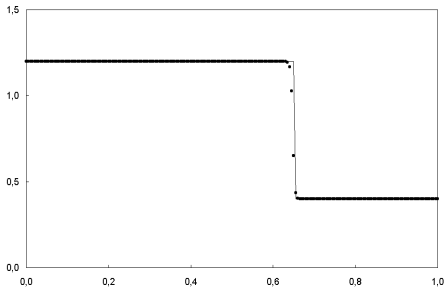

Figure 21: True and computed solution using a High Resolution Method - Slope limiter method

EXAMPLE 4. Consider the inviscid Burgers' Equation (38) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 0 & 0 \leq x < 0.25 \\ 4x - 1 & 0.25 \leq x < 0.5 \\ -4x + 3 & 0.5 \leq x < 0.75 \\ 0 & 0.75 \leq x \leq 0.25 \end{array} \right\} \tag{124}$$

Example 4 shows the evolution in time of the inital data using Godunov's method.

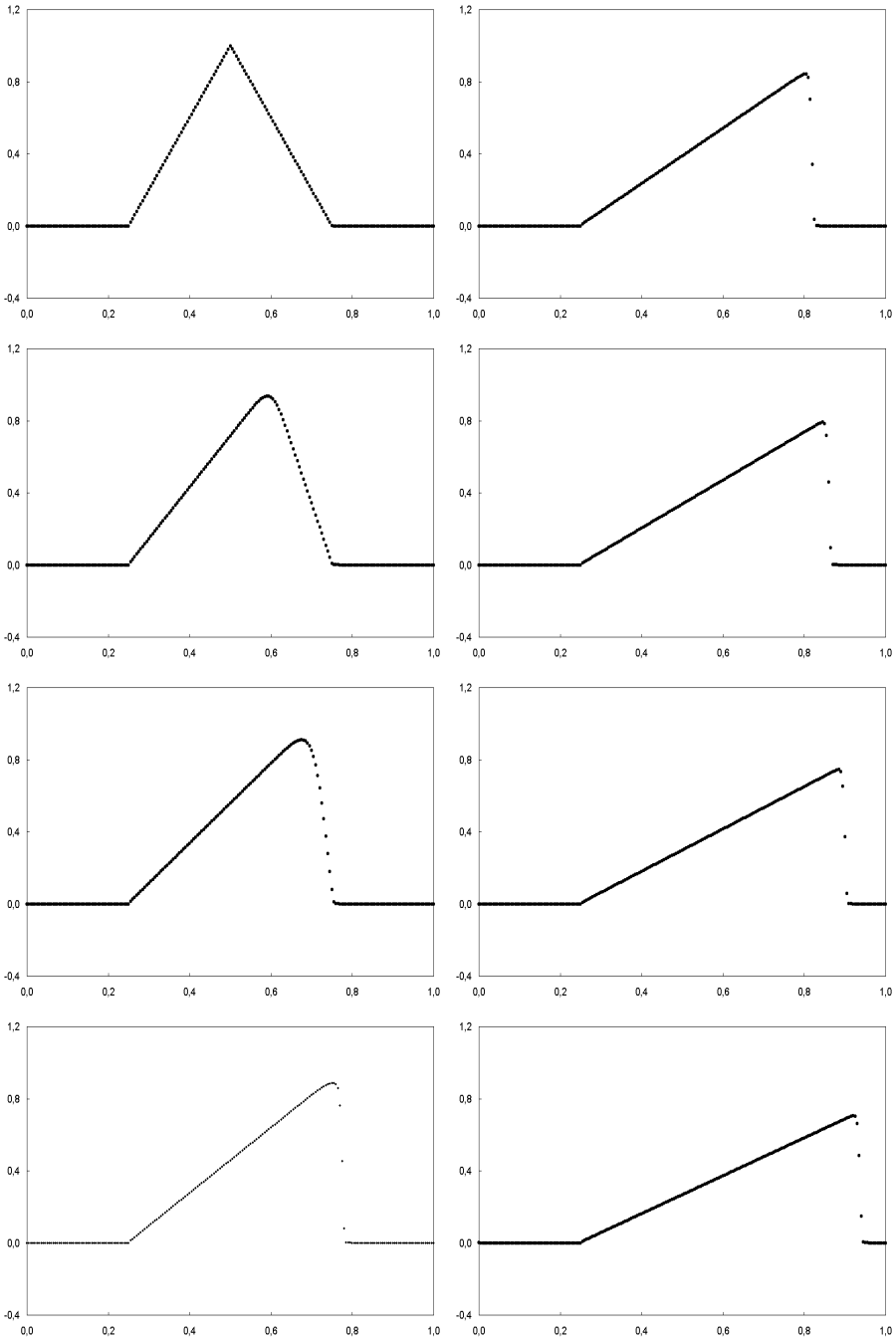

Figure 22: Evolution in time using Godunov's method

EXAMPLE 5. Consider the inviscid Burgers' Equation (38) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 1 & 0 \leq x \leq 1 \\ 0 & else \end{array} \right\} \tag{125}$$

Example 5 shows the evolution in time of the inital data using Godunov's method.

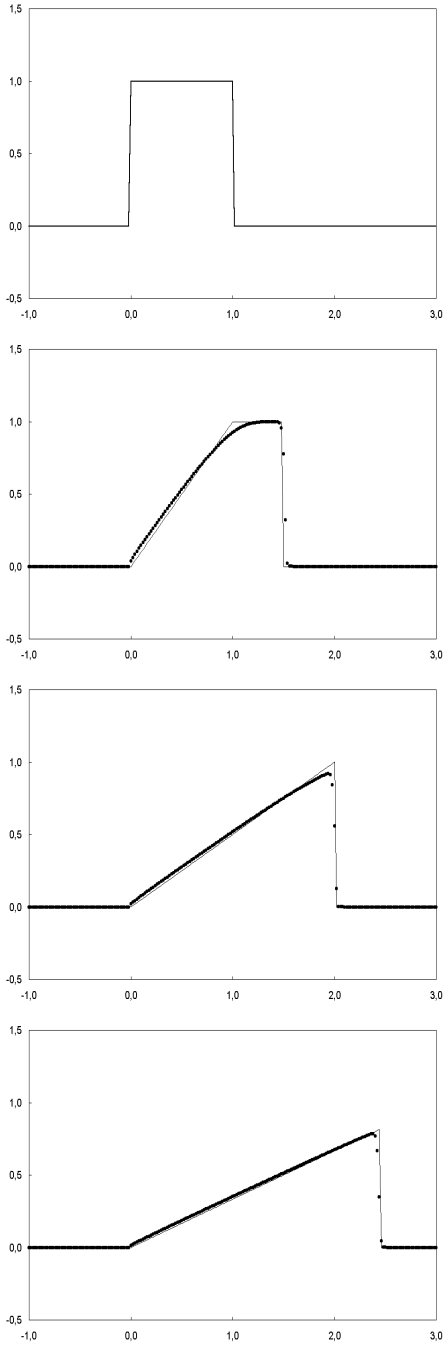

Figure 23: Evolution in time using Godunov's method

EXAMPLE 6. Consider the inviscid Burgers' Equation (38) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 0.4 & x \leq 0.25 \\ 1.2 & x > 0.25 \end{array} \right\} \tag{126}$$

Example 6 shows the developed rarefaction wafe at time t=0.5 computed with different methods.

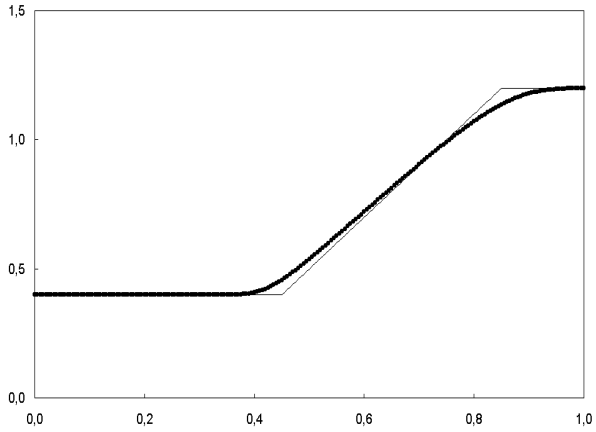

Figure 24: Rarefaction wafe computed with Godunovs method

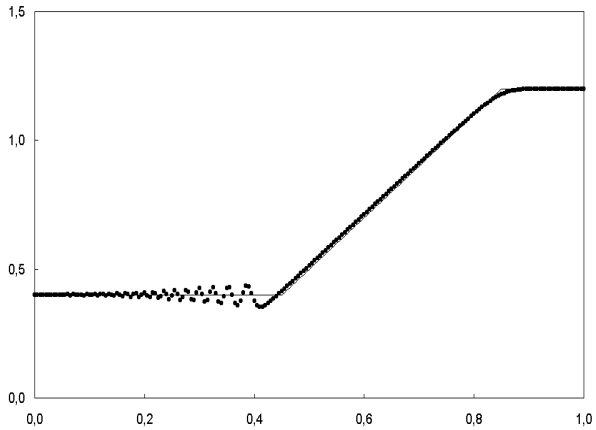

Figure 25: Rarefaction wafe computed with Lax-Wendroff's method

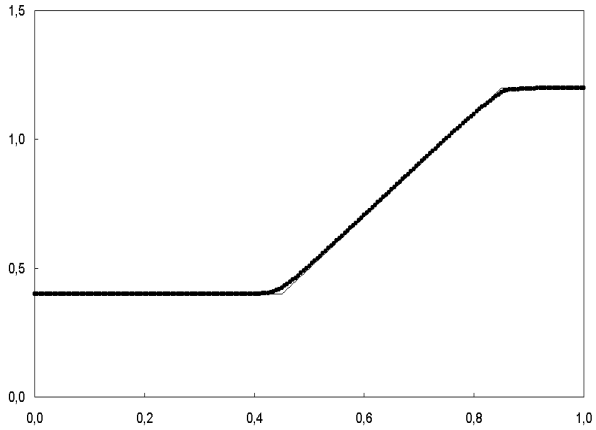

Figure 26: Rarefaction wafe computed with a high resolution method - slope limiter method

EXAMPLE 7. Consider the viscid Burgers' Equation (112) with initial data

$$u_0(x) = \left\{ \begin{array}{ll} 1.2 & x \leq 0.25 \\ 0.4 & x > 0.25 \end{array} \right\} \tag{127}$$

Example 7 shows the computed solutions ($D = 0.01$ and $D = 0.005$) using a numerical method for parabolic PDEs. The smaller D the better is the approximation to the inviscid Burgers' equation (black line).

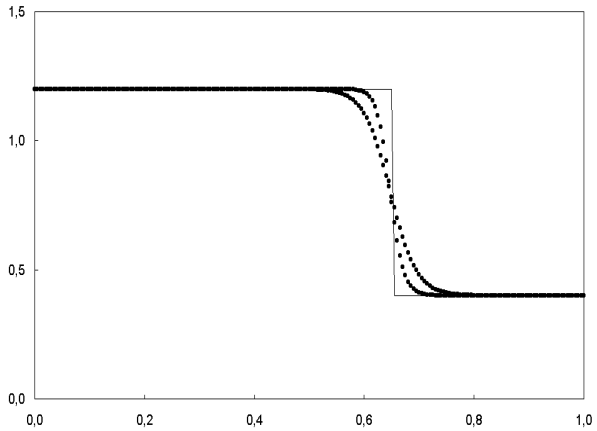

Figure 27: Solution to the viscous Burgers' equation for two different values of D (0.01 and 0.005)

A Source code

```
module globals
  implicit none
  real,allocatable :: u(:),x(:),h(:),u_old(:),numerical_flux(:)
  real :: a,b,D,u_l,u_r,u_star
  integer :: n !number of gridpoints
  character(len=1) :: what_burgers_equation,initial_data,what_continuous
  integer :: what_method
  integer :: output=81,input=80
  integer :: margin
  integer :: n_intv
  real :: n_intv_i
  integer :: i,j,k,quater,half
  real,parameter :: pi=3.1415927
  real :: courant_number,delta_t,delta_x,time,t_max
  logical :: l_per
  logical :: first_time=.true.
  real :: f_plus,f_minus
  real :: sigma,a_roof,u1,u2,minmod

  contains
    function f(u)
      real :: f,u
      f=0.5*u**2
    end function
    function f_der(u)
      real :: f_der,u
      f_der=u
    end function

end module globals


program burgers_equation
  !This program computes numerical soulutions to the inviscid and viscid Burgers' equation
  !using different numerical methods

  use globals

  write(*,*) 'Numerical methods for BURGERS EQUATION'
  write(*,*) 'Inviscid (i) or viscid (v) Burgers equation'
  read(*,*) what_burgers_equation

  if (what_burgers_equation .EQ. 'v') then
    what_method=7
  elseif (what_burgers_equation .EQ. 'i') then
    write(*,*) 'What numerical method should be used'
    write(*,*) '  1  natural finite difference method'
    write(*,*) '  2  upwind method'
    write(*,*) '  3  Lax Friedrich method'
    write(*,*) '  4  Godunovs method'
    write(*,*) '  5  Lax Wendroff method'
    write(*,*) '  6  High resolution - slope limiter method'
    read(*,*) what_method
    if (what_method .NE. 1 .AND. what_method .NE. 2 .AND. what_method .NE. 3 .AND. &
        & what_method .NE. 4 .AND. what_method .NE. 5 .AND. what_method .NE. 6) then
      write(*,*) 'bad value for what_method'
      stop
    endif
  else
    write(*,*) 'bad value for what_burgers_equation'
    stop
  endif

  call init ()

  call init_data ()
```

```
  call calculation ()

  call put_out ()


end program burgers_equation



subroutine init ()
  use globals
  implicit none

  open(unit=input,file='input.a')
  read(input,*) a
  read(input,*) b
  read(input,*) n
  read(input,*) margin
  read(input,*) courant_number
  read(input,*) t_max
  if (what_method .EQ. 7) read(input,*) D
  close(input)

  allocate(x(1:n))
  allocate(h(1:n-1))
  allocate(u(1:n))
  allocate(u_old(1:n))
  allocate(numerical_flux(1:n-1))

  n_intv=n-1
  n_intv_i=1.0/n_intv

  do i=1,n
    x(i)=(i-1)*(b-a)*n_intv_i
  enddo
  do i=1,n_intv
    h(i)=x(i+1)-x(i)
  enddo

  delta_x=(b-a)*n_intv_i
  delta_t=delta_x*courant_number

end subroutine init


subroutine init_data ()
  use globals
  implicit none
  write(*,*) 'Initial data'
  write(*,*) '  (h) roof initial data'
  write(*,*) '  (s) shock initial data'
  write(*,*) '  (r) rarefaction initial data'
  write(*,*) '  (f) shock forming inital data'
  write(*,*) '  (c) a complicated example'
  read(*,*) initial_data

  if (initial_data .EQ. 'h') then
    l_per=.true.
    quater=n/4
    half=n/2
    do i=1,quater
      u(i)=0
    enddo
    do i=n-quater+1,n
      u(i)=0
    enddo
    do i=quater+1,half+1
```

```
      u(i)= (x(i)-x(quater+1)) / (x(half+1)-x(quater+1))
    enddo
    do i=half+1,n-quater
      u(i)= 1- (x(i)-x(half+1)) / (x(n-quater)-x(half+1))
    enddo
  elseif (initial_data .EQ. 's') then
    l_per=.false.
    quater=n/4
    u_l=1.2
    u_r=0.4
    do i=1,quater
      u(i)=u_l
    enddo
    do i=quater+1,n
      u(i)=u_r
    enddo
  elseif (initial_data .EQ. 'r') then
    l_per=.false.
    quater=n/4
    u_l=0.4
    u_r=1.2
    do i=1,quater
      u(i)=u_l
    enddo
    do i=quater+1,n
      u(i)=u_r
    enddo
  elseif (initial_data .EQ. 'f') then
    l_per=.false.
    quater=n/4
    half=n/2
    do i=1,quater
      u(i)=0.25
    enddo
    do i=quater+1,half+1
      u(i)=0.25 - (x(i)-x(quater+1))*0.25/(x(half+1)-x(quater+1))
    enddo
    do i=half+2,n
      u(i)=0
    enddo
  elseif (initial_data .EQ. 'c') then
    l_per=.true.
    quater=n/4
    half=n/2
    do i=1,quater
      u(i)=0.
    enddo
    do i=quater+1,half+1
      u(i)=1
    enddo
    do i=half+2,n
      u(i)=0
    enddo
  else
    write(*,*) 'bad value for inital_data'
    stop
  endif
end subroutine init_data


subroutine put_out ()
  use globals
  implicit none

  open(unit=output,file='output.xls',form='formatted',position='rewind',status='unknown')
  do i=1,n
    write(output,*) u(i)
  enddo
```

```
end subroutine put_out


subroutine calculation ()
  use globals
  implicit none
  do time=0,t_max,delta_t
    if (first_time .EQ. .false.) then
      if (l_per) then
        u(1:margin)=u(n-margin-margin+1:n-margin)
        u(n-margin+1:n)=u(margin+1:margin+margin)
      else
        if (what_method .NE. 6) then
          u(1)=u(2)
          u(n)=u(n-1)
        endif
        if (what_method .EQ. 6) then
          u(1)=u(3)
          u(2)=u(3)
          u(n)=u(n-2)
          u(n-1)=u(n-2)
        endif
      endif
    endif
    first_time=.false.

    u_old=u

    if (what_method .EQ. 1) call finite_difference ()
    if (what_method .EQ. 2) call upwind ()
    if (what_method .EQ. 3) call lax_friedrich ()
    if (what_method .EQ. 4) call godunov ()
    if (what_method .EQ. 5) call lax_wendroff ()
    if (what_method .EQ. 6) call high_resolution ()
    if (what_method .EQ. 7) call parabolic ()
  enddo
end subroutine calculation


subroutine finite_difference ()
  use globals
  implicit none
  do i=2,n
    u(i)=u_old(i)-courant_number*u_old(i)*(u_old(i)-u_old(i-1))
  enddo
end subroutine finite_difference


subroutine upwind ()
  use globals
  implicit none
  do i=2,n
    u(i)=u_old(i)-courant_number*(0.5*(u_old(i)**2)-0.5*(u_old(i-1)**2))
  enddo
end subroutine upwind


subroutine lax_friedrich
  use globals
  implicit none
  do i=2,n-1
    u(i)=0.5*(u_old(i-1) + u_old(i+1)) - 0.5*courant_number* (f (u_old(i+1)) - f(u_old(i-1)) )
  enddo
end subroutine lax_friedrich


subroutine godunov
  use globals
```

```
  implicit none

  !caluclating numerical flux
  do i=1,n-1
    if (f_der(u_old(i)) .GE. 0 .AND. f_der(u_old(i+1)) .GE. 0) u_star=u_old(i)
    if (f_der(u_old(i)) .LE. 0 .AND. f_der(u_old(i+1)) .LE. 0) u_star=u_old(i+1)
    if (f_der(u_old(i)) .GE. 0 .AND. f_der(u_old(i+1)) .LE. 0 .AND. &
        & (f(u_old(i+1))-f(u_old(i)))/(u_old(i+1)-u_old(i)) .GT. 0) u_star=u_old(i)
    if (f_der(u_old(i)) .GE. 0 .AND. f_der(u_old(i+1)) .LE. 0 .AND. &
        & (f(u_old(i+1))-f(u_old(i)))/(u_old(i+1)-u_old(i)) .LT. 0) u_star=u_old(i+1)
    if (f_der(u_old(i)) .LT. 0 .AND. f_der(u_old(i+1)) .GT. 0) u_star=0
    numerical_flux(i)=f(u_star)
  enddo

  do i=2,n-1
    u(i)=u_old(i)-courant_number*(numerical_flux(i)-numerical_flux(i-1))
  enddo
end subroutine godunov


subroutine parabolic
  use globals
  implicit none
  do i=2,n-1
    f_minus=0.5*(f(u_old(i))+f(u_old(i-1)))
    f_plus=0.5*(f(u_old(i))+f(u_old(i+1)))
    u(i)=u_old(i)+courant_number*( D*(u_old(i+1)-2*u_old(i)+u_old(i-1))/delta_x - &
         & (f_plus-f_minus) )
!         & (f(u_old(i))-f(u_old(i-1))) )
  enddo
end subroutine parabolic


subroutine lax_wendroff ()
  use globals
  implicit none
  do i=2,n-1
    u(i)=u_old(i)-0.5*courant_number*( f( u_old(i+1) ) - f( u_old(i-1)) ) +0.5*courant_number**2* &
        & ( f_der( 0.5*(u_old(i)+u_old(i+1)) ) * (f(u_old(i+1)) -f(u_old(i  ))) - &
        &   f_der( 0.5*(u_old(i)+u_old(i-1)) ) * (f(u_old(i)  ) -f(u_old(i-1))) )
  enddo
end subroutine lax_wendroff


subroutine high_resolution ()
  use globals
  implicit none

  !calculation numerical flux
  do i=2,n-1
    u1=u_old(i+1)-u_old(i)
    u2=u_old(i)-u_old(i-1)
    if ( abs(u1) .LT. abs(u2) .AND. (u1*u2) .GT. 0) minmod=u1
    if ( abs(u2) .LT. abs(u1) .AND. (u1*u2) .GT. 0) minmod=u2
    if ( (u1*u2) .LE. 0) minmod=0
    sigma=1/delta_x*minmod
    if ( abs(u_old(i+1)-u_old(i)) .GT. 0.001) then
      a_roof=(f(u_old(i+1))-f(u_old(i)))/(u_old(i+1)-u_old(i))
    else
      a_roof=0
    endif
    numerical_flux(i)=f(u_old(i)) + 0.5*a_roof*(1-courant_number*a_roof)*delta_x*sigma
  enddo

  do i=3,n-2
    u(i)= u_old(i) -courant_number*(numerical_flux(i)-numerical_flux(i-1))
  enddo
end subroutine high_resolution
```

References

[1] Randall J. LeVeque, Numerical Methods of Conservation Laws, Birkhäuser, 1990

[2] Charles A. Hall & Thomas A. Porsching, Numerical Analysis of Partial Differential Equations, Prentice-Hall, 1990

[3] Randall J. LeVeque, Nonlinear Conservation Laws and Finite Volume Methods for Astrophysical Fluid Flow, 27^{th} Saas-Fee Advanced Cource Lecture Notes, Springer-Verlag, 1998