ADAPTIVE SOLUTION OF ONE–DIMENSIONAL SCALAR CONSERVATION LAWS WITH CONVEX FLUX

FOLKMAR A. BORNEMANN*

Abstract. A new adaptive approach for one–dimensional scalar conservation laws with convex flux is proposed. The initial data are approximated on an adaptive grid by a problem dependent, monotone interpolation procedure in such a way, that the multivalued problem of characteristic transport can be easily and explicitly solved. The unique entropy solution is chosen by means of a selection criterion due to HOPF and LAX. For arbitrary times, the solution is represented by an adaptive monotone spline interpolation. The spatial approximation is controlled by local L^1–error estimates. As a distinctive feature of the approach, there is no discretization in time. The method is monotone on fixed grids. Numerical examples are included, to demonstrate the predicted behavior.

Key Words. method of characteristics, adaptive grids, monotone interpolation, L^1–error estimates

AMS(MOS) subject classification. 65M15, 65M25, 65M50

Introduction. A fundamental idea for supporting the development of robust, reliable, and efficient software is *adaptivity.* In the field of ordinary differential equations, and more recently, elliptic and parabolic partial differential equations much progress has been made in this direction.

For inherent structural reasons, the situation is much more difficult for hyperbolic conservation laws. However, adaptivity is of immense importance for this kind of problems, which are known to exhibit propagating waves, of both characteristic and subcharacteristic nature. In problems, for which the solution must be computed accurately in the frontal regions of the waves, such as chemical combustion or the accretion of matter to form a star, numerical computation on a fixed grid tends to be too costly, since a fine grid would be required everywhere.

Two major suggestions have been made to introduce adaptivity to hyperbolic conservation laws, cf. the survey article of HEDSTROM and RODRIQUE [5] and the discussion therein. First, the moving grid approach. This approach is essential a 1D one, for higher dimensional difficulties consider ZEGELING and BLOM [14, Example II]. Moreover, even in the 1D case, it suffers from difficulties introduced by interacting waves, or by the creation of new waves. The second approach uses static adaptive space grids and possible local time steps. Since most of the suggested algorithms in this direction use explicit methods, they have to restrict, at least locally, the time step by a CFL condition, e.g., the algorithm of LUCIER [10, p. 183, 191]. This yields time steps in the same order of magnitude as the *smallest* resolution in space, even for uniformly propagating waves, a fact, which essentially slows down these algorithms. Local time steps are a slight remedy for that problem, at least, if the occurring wave speeds differ largely.

* Fachbereich Mathematik, Freie Universität Berlin, Arnimallee 2–6, D-1000 Berlin 33, and Konrad–Zuse–Zentrum Berlin, Heilbronner Str. 10, D-1000 Berlin 31, Germany. bornemann@sc.zib-berlin.de

Implicit methods allow for a time step, which is independent of the spatial mesh, but tend to introduce too much dissipation, and are too costly. They are ideally suited for adaptive methods for parabolic problems, e.g. [2].

In this report we propose a new adaptive approach, which fully exploits the characteristic transport of (scalar) conservation laws. As an essential new feature, there is *no* discretization in time at all. However, up to now, the problem class, to which our algorithm can be applied, is quite restricted. Future research will help to clarify, whether the main ideas of the approach are capable of more complex, and more interesting problems.

The main idea of our algorithm consists in perturbing initial data, by approximating them with a certain adaptive, problem dependent interpolation procedure in space. This is done in a way, that the characteristic directions, which connect a given space–time point backwards with this new initial data, can be easily accessed. The correct value, giving the entropy solution within this multivalued solution, is obtained by a certain minimum condition due to LAX [7]. In order to represent our solution for fixed times, again, we choose an adaptive interpolation in space, e.g., a piecewise linear interpolation. Other adaptive, and monotone spline interpolation procedures could be used. The adaptive refinement of the interpolation grids is governed by local L^1–error estimates, in a way, similar to techniques known in finite element computations [1, 2, 12].

For uniform grids one could describe the approximation made by our approach as second order in smooth regions, and first order in neighborhoods of shocks. Moreover, for fixed grids, adaptive or not, the approach is monotone. Note that the whole approach is intimately connected with the concept of entropy solutions. They are the only ones which form a stable, in fact contractive, semigroup. Thus a well–aimed perturbation of initial data is only sensible for this concept of solutions.

The report is organized as follows. In Section 1 we present the necessary theoretical material to describe our algorithm. In Section 2 the algorithm is described in detail. Two numerical examples show in Section 3 the theoretically expected behavior.

1. Theoretical Preparations. We are concerned with the solution of scalar conservation laws

$$u_t + f(u)_x = 0, \qquad u(\cdot, 0) = u_0, \tag{1}$$

where $u(\cdot, t)$ is a function on $\mathbb{R}$. Our general *assumptions* on the flux f will be

- $f : \mathbb{R} \to \mathbb{R}$ is convex and C^1,
- $\alpha \equiv f' : \mathbb{R} \to]\alpha_-, \alpha_+[$ is one–one, onto and nondecreasing.

Thus, there exists $\beta \equiv \alpha^{-1} :]\alpha_-, \alpha_+[\to \mathbb{R}$. By $f^* :]\alpha_-, \alpha_+[\to \mathbb{R}$ we denote the *Legendre–Fenchel dual* of f, defined as

$$\begin{aligned} f^*(z) &= \sup_v (vz - f(v)) \\ &= u\alpha(u) - f(u) \qquad \text{with } u = \beta(z). \end{aligned}$$

Theory of convex functions (e.g., TIKHOMIROV [13]) states that

$$(f^*)' = \beta,$$

a fact, which can be easily verified if $f \in C^2$.

Our whole approach heavily relies on the following characterization of that weak solution, which satisfies the entropy condition.

THEOREM 1.1. (LAX [7]). *Let $u_0 \in L^1(\mathrm{IR})$ and $U_0(y) = \int_{-\infty}^{y} u_0(\xi) d\xi$. For $x \in \mathrm{IR}$, $t > 0$ define*

$$u(x,t) = \beta\left(\frac{x-y}{t}\right),$$

where $y = y(x,t) \in]x - t\alpha_-, x - t\alpha_+[$ is any value which minimizes

$$U_0(y) + tf^*\left(\frac{x-y}{t}\right) = \min! \,. \tag{2}$$

Then $u(\cdot,t) \in L^1(\mathrm{IR})$ is the unique entropy solution of the conservation law under consideration.

If there exist several different values y, which minimize (2) for a given x, then x is the position of a shock discontinuity. The limits $u(x-0,t)$, $u(x+0,t)$ exist, and

$$u(x-0,t) \geq \beta\left(\frac{x-y}{t}\right) \geq u(x+0,t)$$

holds for every such y.

Remark. For $u \in L^\infty(\mathrm{IR})$, $u_- \leq u_0 \leq u_+$ a.e., the solution only depends on f restricted to the interval $[u_-, u_+]$.

This Theorem has quite some history. HOPF [6] stated it for the inviscid Burgers equation $u_t + uu_x = 0$. He obtained the result in the limit $\mu \to 0$ of his explicit solution (i.e., the Cole–Hopf transformation to the heat equation) of the viscid Burgers equation $u_t + uu_x = \mu u_{xx}$. Later LAX [7] generalized the result to arbitrary convex fluxes f. A nice interpretation as Bellman's approach to the Hamilton–Jacobi equation $v_t + f(v_x) = 0$ can be found in LAX [8] or CONWAY and HOPF [3]. In fact, $U(x,t) = \int_{-\infty}^{x} u(\xi,t) d\xi$ satisfies the Hamilton–Jacobi equation and

$$U(x,t) = \min_y \left(U_0(y) + tf^*\left(\frac{x-y}{t}\right)\right),$$

if f is adjusted to $f(0) = 0$. This formula is actually connected with the modern notion of *viscosity solutions* of more general Hamilton–Jacobi equations, we refer to the book of P.-L. LIONS [9].

Our approach uses the fact that, for certain u_0, the set of values y which possibly minimize (2) can be considerably restricted. A first step in that direction is the following

COROLLARY 1.2. *Assumptions as in Theorem 1.1. Additionally let u_0 be continuous. Any value $y \in]x - t\alpha_-, x - t\alpha_+[$ which minimizes (2) satisfies*

$$y + t\alpha(u_0(y)) = x, \tag{3}$$

and allows to set $u(x,t) = u_0(y)$.

Proof. Differentiating relation (2) yields, by continuity of u_0,

$$u_0(y) - \beta\left(\frac{x-y}{t}\right) = 0.$$

This implies both assertions. □

The nonlinear equation (3) is just the one which allows to construct, for smooth u_0 and small t, by means of the implicit function theorem, a classical solution of the conservation law. For larger t, equation (3) does not have a unique solution y for whole intervals of x. Thus, the minimum condition (2) can be understood as a *selection principle* for the right y.

The following stability result allows us to change u_0 slightly, in order to obtain simpler problems of the kind (3).

Theorem 1.3. (Quinn [11]). *The entropy solutions of (1) form an L^1–contractive semigroup $\mathcal{S}_t$. Thus, for $u_0, v_0 \in L^1(\mathbb{R})$, the corresponding entropy solutions satisfy the estimate*

$$\|\mathcal{S}_t u_0 - \mathcal{S}_t v_0\|_{L^1} \le \|u_0 - v_0\|_{L^1} \tag{4}$$

for all $t \ge 0$. A proof may also be found in Lax [8].

For our further development we need some *notation*:

- The convex hull of two points u_0, u_1 will be denoted by $\mathrm{co}(u_0, u_1)$.
- For $u \in \mathrm{co}(u_0, u_1)$ the *barycentric coordinate* of u is denoted by $\lambda_{u_0,u_1}(u)$ and satisfies

$$u = (1-\lambda_{u_0,u_1}(u))u_0 + \lambda_{u_0,u_1}(u)u_1.$$

- Let $(y_0, u_0), (y_1, u_1)$ be two points in $\mathbb{R}^2$ with $y_0 < y_1$. The *β–interpolant* of these points is given as $\mu_{u_0,u_1} : [y_0, y_1] \to \mathrm{co}(u_0, u_1)$ by

$$\mu_{u_0,u_1}(y) = \beta\left((1-\lambda_{y_0,y_1}(y))\alpha(u_0) + \lambda_{y_0,y_1}(y)\alpha(u_1)\right).$$

Our assumptions imply, that this is a monoton connection of the two points.

These β–interpolants have the very nice property that (3) may be solved uniquely, as we show now.

Lemma 1.4. *For given $t > 0$ define*

$$\varphi(y) = y + t\alpha(\mu_{u_0,u_1}(y)),$$

which maps $[y_0, y_1]$ onto $\mathrm{co}(\varphi(y_0), \varphi(y_1))$. If $\varphi(y_0) \ne \varphi(y_1)$, the equation $x = \varphi(y)$ is uniquely solved by y given as

$$\lambda_{y_0,y_1}(y) = \lambda_{\varphi(y_0),\varphi(y_1)}(x).$$

Proof. Simply note that

$$\varphi(y) = (1-\lambda_{y_0,y_1}(y))(y_0 + t\alpha(u_0)) + \lambda_{y_0,y_1}(y)(y_1 + t\alpha(u_1)),$$

and that $\varphi(y_i) = (y_i + t\alpha(u_i))$ for $i = 0, 1$. □

Since integrals are involved in (2), it helps a lot that integrals of β–interpolants can be computed explicitly.

LEMMA 1.5. *We have, for $y \in [y_0, y_1]$, that*

$$\int_{y_0}^{y} \mu_{u_0,u_1}(\eta)d\eta = \begin{cases} \dfrac{y_1 - y_0}{\alpha(u_1) - \alpha(u_0)} f^* \Big|_{\alpha(u_0)}^{\alpha(\mu_{u_0,u_1}(y))} & \text{if } u_0 \neq u_1, \\ (y - y_0)u_0 & \text{if } u_0 = u_1. \end{cases}$$

Proof. Let $u_0 \neq u_1$. The substitution $\zeta = (1 - \lambda_{y_0,y_1}(\eta))\alpha(u_0) + \lambda_{y_0,y_1}(\eta)\alpha(u_1)$ gives

$$\int_{y_0}^{y} \mu_{u_0,u_1}(\eta)d\eta = \frac{y_1 - y_0}{\alpha(u_1) - \alpha(u_0)} \int_{\alpha(u_0)}^{\alpha(\mu_{u_0,u_1}(y))} \beta(\zeta)d\zeta.$$

Thus, the assertion follows from $(f^*)' = \beta$. □

Now we try to approximate u_0 by a piecewise β–interpolant $\hat{u}_0$, for which we have seen that problems (3) and (2) turn out to be fairly simple. For that purpose, let u_0 be piecewise continuous with $\operatorname{supp} u_0 \subset\subset [a, b]$. Let $\Delta : a = y_0 < y_1 < \ldots < y_n = b$ be a subdivision of that interval, with mesh–size parameter

$$h = \max_{1 \le j \le n} (y_j - y_{j-1}).$$

Denote the subintervals by $I_j = [y_{j-1}, y_j]$, $j = 1, 2, \ldots, n$. The piecewise β–interpolant $\mathcal{I}_\Delta u_0$ is now defined as

$$\mathcal{I}_\Delta u_0(y) = \begin{cases} \mu_{u_0(y_{j-1}),u_0(y_j)}(y) & \text{for } y \in I_j, j = 1, 2, \ldots, n, \\ 0 & \text{elsewhere.} \end{cases}$$

Obviously we have $\mathcal{I}_\Delta u_0 \in C^0$ and $\mathcal{I}_\Delta u_0(y_j) = u_0(y_j)$, $j = 1, 2, \ldots, n$.

LEMMA 1.6. *Let u be piecewise continuous on $[a, b]$. Then*

$$\|u - \mathcal{I}_\Delta u\|_{L^1[a,b]} \to 0 \quad \text{for } h \to 0.$$

If u is piecewise C^2, and $f \in C^3$ such that

$$M = \|f'''/f''^3\|_{L^\infty[u_-,u_+]} \|f''\|^2_{L^\infty[u_-,u_+]} < \infty,$$

where $u_- \le u(x) \le u_+$ for all $x \in [a, b]$, then there is a constant $c = c(u, M)$, such that

$$\|u - \mathcal{I}_\Delta u\|_{L^1[a,b]} \le ch, \quad \text{and} \quad \|u - \mathcal{I}_\Delta u\|_{L^1(I_c)} \le ch^2.$$

Here $I_c = \bigcup_{j \notin J_u} I_j$ with $J_u = \left\{1 \le j \le n \;\middle|\; \left(u|_{I_j}\right) \notin C^2\right\}$.

Proof. We proof the second, smooth part. The first part follows by usual density arguments. Take any $j \notin J_u$, and denote the linear interpolation operator at the nodes y_{j-1}, y_j by $\mathcal{I}_L$. Since by construction

$$\mathcal{I}_L \mathcal{I}_\Delta = \mathcal{I}_L$$

we estimate by the usual error expression for linear interpolation

$$\begin{aligned}\|u-\mathcal{I}_\Delta u\|_{L^\infty(I_j)} &\le \|u-\mathcal{I}_L u\|_{L^\infty(I_j)}+\|\mathcal{I}_\Delta u-\mathcal{I}_L\mathcal{I}_\Delta u\|_{L^\infty(I_j)}\\ &\le \frac{h^2}{8}\left(\|u''\|_{L^\infty[a,b]}+\|\mu''_{u(y_{j-1}),u(y_j)}\|_{L^\infty(I_j)}\right).\end{aligned}$$

Now we compute, for $y\in I_j$, that

$$\begin{aligned}\mu''_{u(y_{j-1}),u(y_j)}(y) &= -\frac{f'''\left(\mu_{u(y_{j-1}),u(y_j)}(y)\right)}{f''\left(\mu_{u(y_{j-1}),u(y_j)}(y)\right)^3}\left(\frac{\alpha(u(y_j))-\alpha(u(y_{j-1}))}{u(y_j)-u(y_{j-1})}\cdot\frac{u(y_j)-u(y_{j-1})}{y_j-y_{j-1}}\right)^2\\ &= -\frac{f'''\left(\mu_{u(y_{j-1}),u(y_j)}(y)\right)}{f''\left(\mu_{u(y_{j-1}),u(y_j)}(y)\right)^3}f''(\eta)^2\,u'(\zeta)^2\end{aligned}$$

for some $\eta\in\operatorname{co}(u(y_{j-1}),u(y_j))$, $\zeta\in I_j$. Hence, it is $\|\mu''_{u(y_{j-1}),u(y_j)}\|_{L^\infty(I_j)}\le M\|u'\|^2_{L^\infty[a,b]}$. For $j\in J_u$, we simply estimate

$$\|u-\mathcal{I}_\Delta u\|_{L^1(I_j)}\le 2\|u\|_{L^\infty[a,b]}h.$$

Finally, we observe that $\#J_u\le\nu$ as $h\to 0$, because we assumed that u is piecewise C^2. Thus, we obtain

$$\|u-\mathcal{I}_\Delta u\|_{L^1[a,b]}\le\frac{b-a}{8}\left(\|u''\|_{L^\infty[a,b]}+M\|u'\|^2_{L^\infty[a,b]}\right)h^2+2\nu\|u\|_{L^\infty[a,b]}h$$

and

$$\|u-\mathcal{I}_\Delta u\|_{L^1(I_c)}\le\frac{b-a}{8}\left(\|u''\|_{L^\infty[a,b]}+M\|u'\|^2_{L^\infty[a,b]}\right)h^2.$$

□

Note that the same result holds for the piecewise linear interpolation operator $\mathcal{I}_L$, as introduced in the proof.

Remark. The value of the constant M is invariant against transformations $f\mapsto\gamma f$ with $\gamma>0$.

Another important property of $\mathcal{I}_\Delta$ is monotonicity. This is a fairly simple consequence of the assumed monotonicity of α,β.

LEMMA 1.7. *Let u,v be piecewise continuous. The pointwise inequality $u\le v$ implies that pointwise $\mathcal{I}_\Delta u\le\mathcal{I}_\Delta v$. The same holds for the linear interpolation operator $\mathcal{I}_L$.*

2. The Algorithm. Our algorithm solves the following problem: Given a conservation law (1), a piecewise continuous initial u_0, compute, for an accuracy TOL and a time $t>0$, an approximation $\hat{u}(\cdot,t)$ to the solution $u(\cdot,t)$ such that

$$\|u(\cdot,t)-\hat{u}(\cdot,t)\|_{L^1}\le\mathsf{TOL}\,. \tag{5}$$

The idea behind it is described very roughly as

A. Approximate u_0 by a piecewise β–interpolant $v_0 = \mathcal{I}_\Delta u_0$, such that

$$\|u_0 - v_0\|_{L^1} \leq \mathsf{TOL}/2.$$

B. The exact solution $v(\cdot, t) = \mathcal{S}_t v_0$ to the initial v_0 can exactly and easily be pointwise evaluated.

C. Approximate $v(\cdot, t)$ by a piecewise linear interpolant $\hat{u}(\cdot, t) = \mathcal{I}_L v(\cdot, t)$, resp. a piecewise β–interpolant $\hat{u}(\cdot, t) = \mathcal{I}_{\Delta_t} v(\cdot, t)$, such that

$$\|v(\cdot, t) - \hat{u}(\cdot, t)\|_{L^1} \leq \mathsf{TOL}/2.$$

We denote our approximation operator by $\mathcal{P}_t : u_0 \mapsto \hat{u}(\cdot, t)$. If these steps can be achieved, Theorem 1.3 guarantees for the accuracy requirement (5).

The choice of the interpolant in Step C, i.e., $\mathcal{I}_L$ or $\mathcal{I}_{\Delta_t}$, is not really important. In fact, any adaptive monotone spline interpolation, which controls the L^1–approximation error, could be used to represent the solution for fixed times. The β–interpolant should be taken, if we intend to use the solution at a particular time as new initial data for another computation.

We now describe each step more closely. Note that Steps A and C are quite similar tasks.

Step A. Here, the choice of an appropriate mesh Δ is the essential problem. This will be done in an *adaptive* way, starting with a coarse mesh Δ_0. The main loop reads as:

```
while  (estimated $L^1$–error > $\mathsf{TOL}/2$)
  {
        $\Delta_{k+1} = \text{refine}(\Delta_k)$;
        $k = k + 1$;
  }
```

Let the kth mesh be $\Delta_k : a = y_0^k < y_1^k < \ldots < y_{n_k}^k = b$. For the following, we will suppress the index k. The L^1–error is composed as

$$\epsilon = \|u_0 - \mathcal{I}_\Delta u_0\|_{L^1} = \sum_{j=1}^{n} \epsilon_j,$$

with

$$\epsilon_j = \int_{y_{j-1}}^{y_j} \left| u_0(\xi) - \mu_{u_0(y_{j-1}), u_0(y_j)}(\xi) \right| d\xi.$$

The local error ϵ_j will be estimated by a trapezoidal sum, introducing the midpoint of I_j. Thus, noting the interpolation property, we obtain the local estimate

$$\epsilon_j \approx \eta_j = \frac{(y_j - y_{j-1})}{2} \left| u_0\left(\frac{y_{j-1} + y_j}{2}\right) - \mu_{u_0(y_{j-1}), u_0(y_j)}\left(\frac{y_{j-1} + y_j}{2}\right) \right|.$$

We remark that

$$\mu_{u_0(y_{j-1}), u_0(y_j)}\left(\frac{y_{j-1} + y_j}{2}\right) = \beta\left(\frac{1}{2}\left(\alpha(u_0(y_{j-1})) \quad + \alpha(u_0(y_j))\right)\right).$$

This error estimate is sensible for accuracy and complexity reasons. The global estimate is thus given as

$$\eta = \sum_{j=1}^{n} \eta_j .$$

For actual refinement we need some *refinement strategy*, which uses the local information provided by the indicators η_j. We build our strategy along the lines of an idea, which was introduced by Babuška and Rheinboldt [1] for elliptic problems. The whole refinement is governed by a single value, cut:

$$\text{bisect } I_j \text{ if } \eta_j > \text{cut}.$$

We would like to choose cut in such a way that the local errors are nearly equidistributed. For this reason, we use a simple heuristic prediction scheme to forecast, what may happen to η_j, if I_j is subdivided. We may assume

$$\eta_j = c_j h_j^{\lambda_j} \text{ as } h_j \to 0,$$

with $h_j = \text{meas}(I_j)$. Suppose I_j was generated by subdividing I_j^{old} with local error η_j^{old}. The η_j–value after subdivision of I_j will thus be approximately

$$\eta_j^{\text{new}} = \frac{\eta_j^2}{\eta_j^{\text{old}}} .$$

Clearly now, we should refine only those elements I_j, which have an η_j–value above the largest predicted new η–value in the next mesh:

$$\text{cut} = \max_j \eta_j^{\text{new}} .$$

Remark. In the case that I_j is bisected, we note that $u_0((y_{j-1} + y_j)/2)$ has already been computed for η_j. Thus, we can readily assign this value to the new node.

The actual implementation of mesh refinement can easily be done by means of packages designed for finite element computations which use tree data structures, e.g., Roitzsch [12].

After the adaptive refinement we are provided with the final mesh Δ, an error estimate $\eta < \mathsf{TOL}/2$, and each node y_j carries the interpolation information $u_0(y_j)$. For purposes of Step B, we should additionally store in each node the integral information

$$\int_{-\infty}^{y_j} \mathcal{I}_\Delta u_0(\xi) d\xi,$$

which can be computed by successive application of Lemma 1.5.

Step C. The piecewise linear approximation (piecewise β–interpolant) $\hat{u} = \mathcal{P}_t u_0$ to $v(\cdot, t) = \mathcal{S}_t v_0$ is computed in a similar fashion as $v_0 = \mathcal{I}_\Delta u_0$, creating its own mesh Δ_t, which in general will differ from Δ. This approximation procedure only demands the possibility of evaluating $v(x, t)$ for certain points x.

Step B. How to compute $v(x,t)$ for a given x? This question will be addressed now. In preparation of any evaluation, the following values are computed

$$\varphi(y_j) = y_j + t\alpha(v_0(y_j))$$

for $j = 0, 1, \ldots, n$. These are the positions of the characteristic transport of the points y_j. Given x, we first determine the set J_x of indices j, such that the equation (3), i.e.,

$$x = y + t\alpha(v_0(y)) \tag{6}$$

possesses a solution $y \in I_j$. By construction of v_0 and Lemma 1.4, this set is exactly given by

$$J_x = \{1 \leq j \leq n \mid x \in \mathrm{co}(\varphi(y_{j-1}), \varphi(y_j))\}.$$

For $j \in J_x$ with $\varphi(y_{j-1}) \neq \varphi(y_j)$, we compute the barycentric coordinate

$$\lambda_j = \lambda_{\varphi(y_{j-1}),\varphi(y_j)}(x),$$

which is, by Lemma 1.4, also the barycentric coordinate of the unique solution $\bar{y}_j \in I_j$, i.e., $\bar{y}_j = (1-\lambda_j)y_{j-1} + \lambda_j y_j$. In the exceptional case $j \in J_x$, $\varphi(y_{j-1}) = \varphi(y_j)$, all values $y_{j-1} \leq y \leq y_j$ satisfy (6). Thus, the value of the expression in (2) remains constant on the whole interval $[y_{j-1}, y_j]$. Hence, we may take $\bar{y}_j = y_{j-1}$ as representative candidate for the minimizing value of (2). Summarizing, our construction of v_0 allows us to compute a set of critical points of (2) with cardinality $\#J_x$, in which a minimizing value is included.

In view of Theorem 1.1 and its Corollary we choose the smallest value $\bar{y}_\ell$ among those $\bar{y}_j, j \in J_x$, which minimize (2). Our desired value of $v(x,t)$ is given by

$$v(x,t) = v_0(\bar{y}_\ell) = \beta\left((1-\lambda_\ell)\alpha(v_0(y_{\ell-1})) \;+ \lambda_\ell\alpha(v_0(y_\ell))\right).$$

For the evaluation of (2) it is necessary to rely on Lemma 1.5.

If there are several values $\bar{y}_k$ which minimize (2) among the $\bar{y}_j, j \in J_x$, we are allowed, due to Theorem 1.1, to take any of them: In this case, x is exactly the position of a shock of $v(\cdot,t)$. All minimizing $\bar{y}_k$ produce values $v(x,t)$ which are between the left and the right shock value. In fact, since we choose the smallest $\bar{y}_k$, which minimizes (2), we can be more specific. We obtain

$$v(x,t) = v(x-0,t) \tag{7}$$

for any shock position x. Note that the specification $\bar{y}_j = y_{j-1}$ in the case $\varphi(y_{j-1}) = \varphi(y_j)$ also served this purpose: It guarantees, that the smallest minimizing value of the $\bar{y}_j$ is really the smallest value of all minimizing values for (2).

A simple implication of the monotonicity property of $\mathcal{I}_\Delta$ and $\mathcal{I}_L$ (Lemma 1.7), together with the monotonicity of the semigroup, is the monotonicity of our algorithm, restricted to *fixed* meshes.

LEMMA 2.1. *Let u_0, w_0 be piecewise continuous, and pointwise $u_0 \leq w_0$. For fixed Δ, Δ_t, we obtain that pointwise $\mathcal{P}_t u_0 \leq \mathcal{P}_t w_0$.*

Proof. Care should be taken, if x is a shock position of both $v(\cdot, t) = \mathcal{S}_t \mathcal{I}_\Delta u_0$ and $\bar{v}(\cdot, t) = \mathcal{S}_t \mathcal{I}_\Delta w_0$. Here, one has to rely on (7). Otherwise, one would have to exclude a neighborhood of x, if this shock–position happens to be a nodal point of Δ_t. □

In order to run our algorithm, we need procedures for evaluating f, α and β. If β is not given analytically, we may compute it by Newton's method. Here, we need $f \in C^2$ and a procedure for evaluating $\alpha' = f''$.

```
function $\beta(z)$
 {
    $u = z;$
    do
      {
         $\Delta u = -(f'(u) - z)/f''(u);$
         $u = u + \Delta u;$
      }
    until $(\Delta u < \rho\,\mathsf{TOL}\,/(b - a));$
 }
```

The factor $0 < \rho < 1$ is inserted for security reasons. Note that Newton's method converges monotonely for every $z \in]\alpha_-, \alpha_+[$, because of the monotonicity of $\alpha = f'$.

3. Numerical Examples. The algorithm has been implemented in the program CFAD1, written in C. All numerical experiments of this report were made using double precision arithmetic on a SUN–SPARC 4/50 FGX.
Important: Since we can exactly evaluate $v(x, t)$ for any (x, t), $t > 0$, it should be clear, that the time–steps of the examples have been solely introduced for graphical reasons. They are completely arbitrary and independent, and we work for all times with the same v_0. *There is no discretization in time!* Once more, we remark, that any adaptive monotone spline interpolation, which controls the L^1–approximation error, could be used to represent the solution for fixed times. For simplicity, we have chosen piecewise linear interpolation.

Example 1. Here, we consider the nonlinear conservation law

$$u_t + \left(\frac{u^4}{4}\right)_x = 0$$

with initial data

$$u_0(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1, \\ 0 & \text{elsewhere.} \end{cases}$$

The inverse of the flux derivative, $\beta = (\cdot)^{1/3}$, is quite different from a linear function, giving the β–interpolant a distinguishable shape.

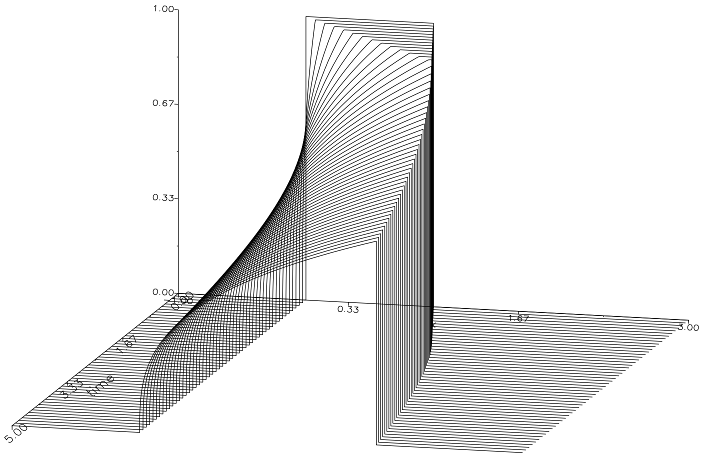

FIG. 1. *Example 1. Evolution of the solution, represented by adaptive linear interpolation.*

The exact solution is given by

$$u(x,t) = \begin{cases} \begin{cases} \left(\frac{x}{t}\right)^{1/3} & 0 \le x \le t \\ 1 & t \le x \le 1 + t/4 \\ 0 & \text{elsewhere} \end{cases} & \text{for } 0 < t \le 4/3, \\ \begin{cases} \left(\frac{x}{t}\right)^{1/3} & 0 \le x \le \left(\frac{4}{3}\right)^{3/4} t^{1/4} \\ 0 & \text{elsewhere} \end{cases} & \text{for } t > 4/3. \end{cases}$$

The computed solution for $0 \le t \le 5$, using a time–step $\tau = 0.1$ *for graphical reasons*, can be seen in Fig. 1. If not stated otherwise, we choose as accuracy $\mathsf{TOL} = 10^{-4}$. The solution was represented by the adaptive linear interpolation of Step C.

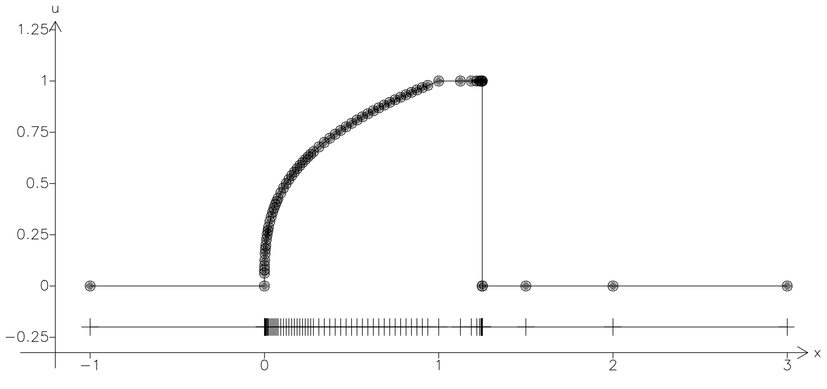

FIG. 2. *Example 1. Solution for* $t = 1.0$*, represented by adaptive linear interpolation.*

The solution for the particular time $t = 1.0$ is shown in Fig. 2, represented by the adaptive linear interpolation. We observe that, as a result of our construction (7), there is no grid point with a value between the left and the right shock value.

The development of the interpolation grid in time, here with time–step $\tau = 0.05$, can be seen in Fig. 3. We can observe nicely, how the rarefaction wave runs into the shock, and how the shock speed changes its behavior thereafter.

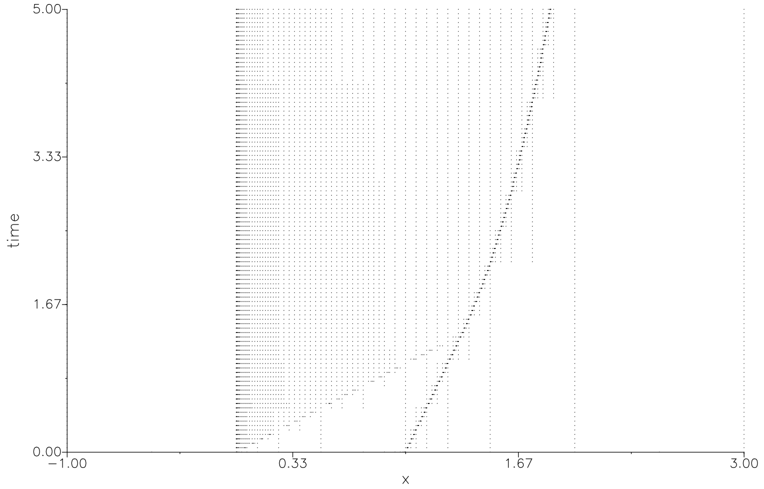

FIG. 3. *Example 1. Grid, using adaptive linear interpolation.*

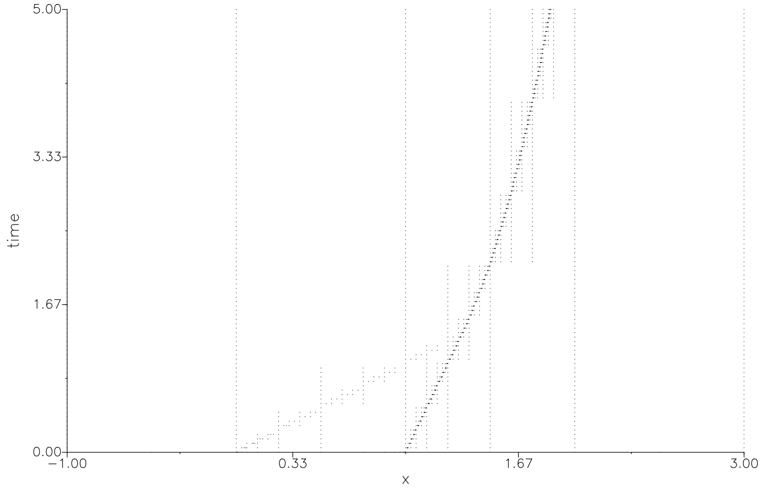

FIG. 4. *Example 1. Grid, using adaptive β–interpolation.*

Using the adaptive β–interpolation to represent the solution, we get much less grid points. This is precisely what should have been expected for this example: Rarefaction waves are exactly represented by the β–interpolant of the left and right value. For the same accuracy as above, the corresponding grid ($\tau = 0.05$) is shown in Fig. 4.

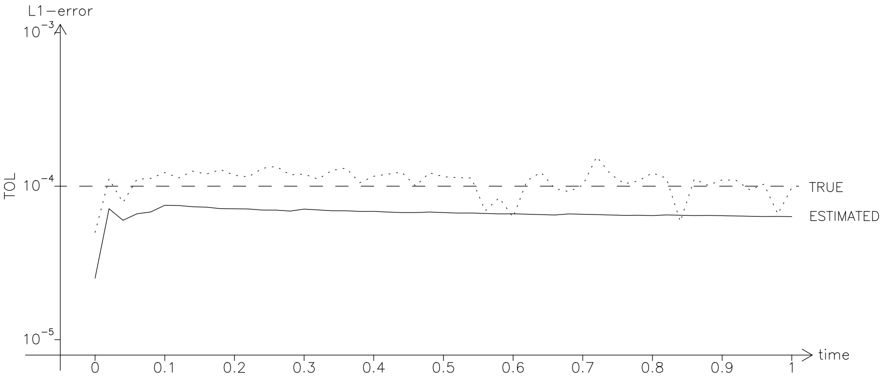

Fig. 5. *Example 1. Evolution of the error, true (...) and estimated (—).*

The quality of our error estimator can be seen in Fig. 5. We observe a slight error underestimation. Our estimated error η at time t is the *sum* of the estimated β–interpolation error η_0 of the initial data and of the estimated linear interpolation error η_t of $\hat{u}$,

$$\eta = \eta_0 + \eta_t .$$

Compared with the true L^1–error ϵ, we obtained for all of our experiments (i.e., $0 \le t \le 5$, $\tau = 0.05$, $\mathsf{TOL} = 10^{-1}, \dots, 10^{-8}$, linear as well as β–interpolation of the solutions), that

$$0.33 \le \frac{\eta}{\epsilon} \le 1.97 \; .$$

Finally, we show in Fig. 6 the dependence of the CPU–time (in seconds) on the accuracy TOL, for the case, that we represent the solution at each time by the adaptive linear interpolation. The comparison has been made using 100 time–steps of size $\tau = 0.05$ for each accuracy. The dotted line in the double–logarithmic scale has slope $-1/2$. We observe, that asymptotically

$$\text{CPU-time} \propto \mathsf{TOL}^{-1/2} \, . \tag{8}$$

This is an optimal result, since, for the set $\mathcal{S}_{2,n}$ of piecewise linear functions with not more than n breaks in the first derivative, we obtain

$$\operatorname{dist}(v(\cdot,t), \mathcal{S}_{2,n}) = \mathcal{O}\left(n^{-2}\right),$$

a result, which can be found in de Boor [4, Theorem III.2]. Thus, the behavior (8) shows two things: First, that our mesh was chosen nearly optimal, second, that we realized our algorithm with an complexity of $\mathcal{O}(\#\text{nodes})$.

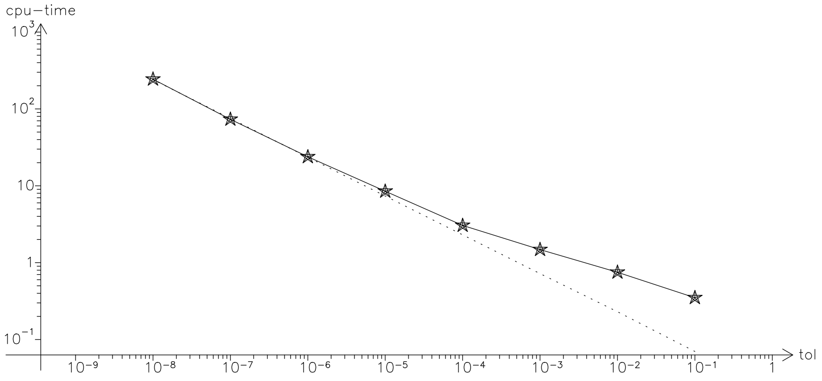

FIG. 6. *Example 1. Computing time vs.* TOL.

Example 2. The problem of this example is given by the inviscid Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0$$

with initial data

$$u_0(x) = \begin{cases} 2.4 + \sin(\pi(x - 0.5)) & \text{for } 0.5 \le x \le 2.5, \\ 2.4 & \text{elsewhere.} \end{cases}$$

This initial data does not have a compact support, but we can obviously modify our algorithm to handle this kind of problems.

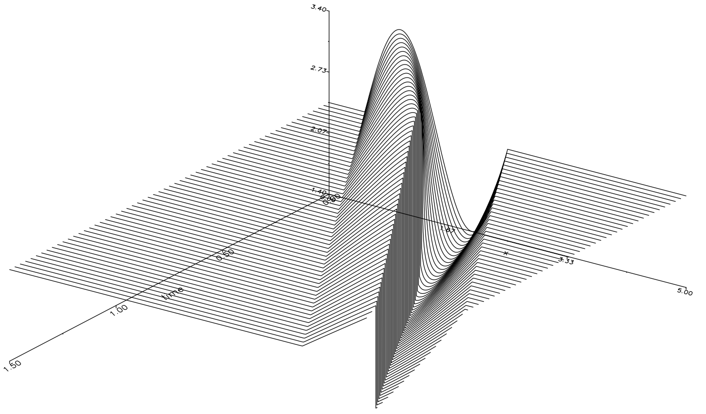

FIG. 7. *Example 2. Evolution of the solution.*

The continuous initial u_0 develops a shock at time $t = 1/\pi \approx 0.318$. The computed solution can be seen in Fig. 7. It was computed with accuracy $\mathsf{TOL} = 10^{-4}$ in the time interval $[0, 1.5]$, using a time–step $\tau = 0.025$.

The corresponding grid is shown in Fig. 8. The number of grid points varies between 330 at the beginning and 11 at the end.

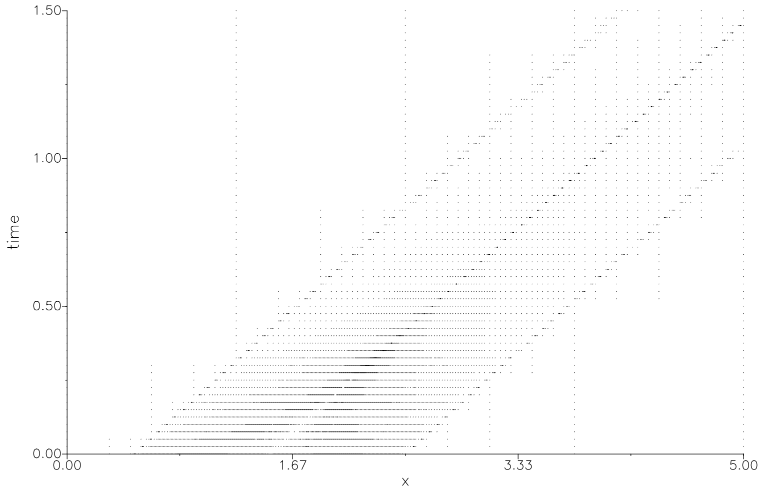

FIG. 8. *Example 2. Evolution of the grid.*

Figs. 9 and 10 show a zoom into the solution just before and just after the shock formation. In both cases we have taken the position $x_s = 1.5 + 2.4t$, and have shown the computed solution in the interval $[x_s - 0.01, x_s + 0.01]$.

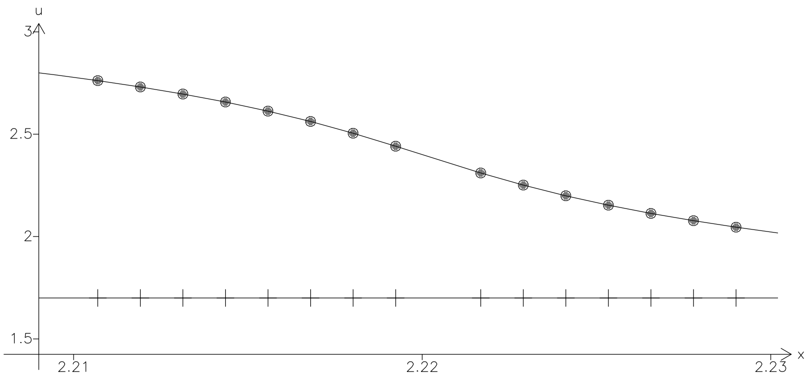

FIG. 9. *Example 2. Zoom into solution, just before the shock (t = 0.3).*

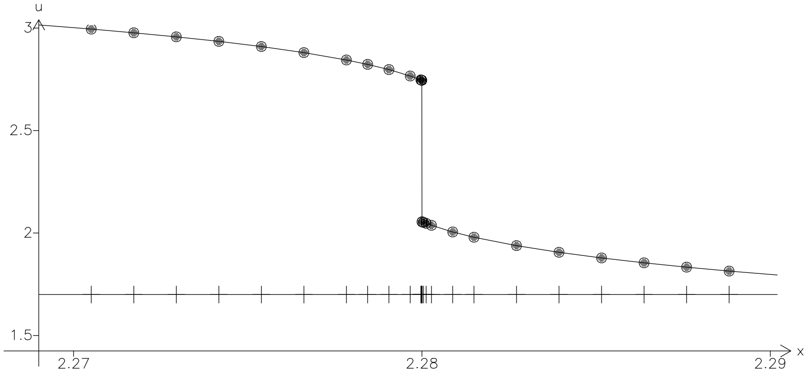

FIG. 10. *Example 2. Zoom into solution, just after the shock (t = 0.325).*

REFERENCES

[1] I. Babuška and W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., 15 (1978), pp. 736–754.

[2] F. A. Bornemann, *An adaptive multilevel approach to parabolic equations III. 2D error estimation and multilevel preconditioning*, IMPACT Comput. Sci. Engrg., 4 (1992), pp. 1–45.

[3] E. Conway and E. Hopf, *Hamilton's theory and generalized solutions of the Hamilton–Jacobi equation*, J. Math. Mech., 13 (1964), pp. 939–986.

[4] C. de Boor, *A Practical Guide to Splines*, Springer–Verlag, New York, Heidelberg, Berlin, 1978.

[5] G. W. Hedstrom and G. H. Rodrique, *Adaptive–grid methods for time–dependent partial differential equations*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer–Verlag, Berlin, Heidelberg, New York, 1982, pp. 474–484.

[6] E. Hopf, *The partial differential equation $u_t + uu_x = \mu u_{xx}$*, Comm. Pure Appl. Math., 3 (1950), pp. 201–230.

[7] P. D. Lax, *Hyperbolic systems of conservation laws II*, Comm. Pure Appl. Math., 7 (1957), pp. 537–566.

[8] ———, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, Conf. Board Math. Sci. 11, SIAM, Philadelphia, 1973.

[9] P.-L. Lions, *Generalized Solutions of Hamilton–Jacobi Equations*, Research Notes in Mathematics 69, Pitman, Boston, London, Melbourne, 1982.

[10] B. J. Lucier, *A stable adaptive numerical scheme for hyperbolic conservation laws*, SIAM J. Numer. Anal., 22 (1985), pp. 180–203.

[11] B. Quinn, *Solutions with shocks, an example of an L_1–contractive semigroup*, Comm. Pure Appl. Math., 24 (1971), pp. 125–132.

[12] R. Roitzsch, *KASKADE programmer's manual*, Konrad–Zuse–Zentrum, Berlin, Technical Report TR89–5, 1989.

[13] V. M. Tikhomirov, *Convex analysis*, in Analysis II, Encyclopaedia of Mathematical Sciences Vol. 14, R. Gamkrelidze, ed., Springer–Verlag, Berlin, Heidelberg, New York, 1990.

[14] P. A. Zegeling and J. G. Blom, *A note on the grid movement induced by MFE*, Centrum voor Wiskunde en Informatica, Amsterdam, Report NM–R9019, 1990.