A high-resolution scheme of TVD type for the Aw-Rascle-Greenberg model of traffic flow

MARCEL GURRIS DMITRI KUZMIN STEFAN TUREK

Institut of Applied Mathematics (LSIII), University of Dortmund
Vogelpothsweg 87, D-44227 Dortmund, Germany

Abstract: An approximate Riemann solver for the Aw-Rascle traffic flow model and the extension of Greenberg is constructed on the algebraic level. A discrete diffusion operator is added to the (oscillatory) high-order finite difference / element discretization to enforce a vectorial LED criterion. This yields a nonoscillatory but diffusive low-order scheme. To increase the accuracy an antidiffusion operator is added. The amount of antidiffusion is controlled using TVD flux limiters. Practical algorithms are presented for the derivation of the low-order scheme, construction of the antidiffusion operator, and the solution of the arising nonlinear algebraic system. Another option considered in this paper is based on a segregated treatment of the equations at hand. It is shown that scalar upwinding / limiting techniques are inappropriate, since even the low-order scheme produces oscillatory solutions to the coupled AR model, although it is monotone for each single equation. Hence, the strongly coupled algorithm developed in this paper yields superior results, as demonstrated by numerical examples.

1 Introduction

Various macroscopic traffic flow models have been proposed in the literature. Usually such models consist of one or two partial differential equations. The

quantities of interest are the density and velocity, which depend on space and time.

The developement of macroscopic traffic flow models started with the work of Lighthill, Whitham and Richards [28], [30]. They published a model of the form

$$\partial_t \rho + \partial_x (V(\rho)\rho) = 0, \tag{1}$$

which is simply the continuity equation with a nonlinear car velocity V depending on the density ρ.
Another model is proposed by Payne [27] and Whitham [29]. It contains a velocity equation and reads

$$\begin{aligned} \partial_t \rho + \partial_x (u\rho) &= 0 && (2) \\ \partial_t u + u\partial_x u + \frac{c_0^2}{\rho}\partial_x \rho &= \frac{V(\rho) - u}{\tau} && (3) \end{aligned}$$

with constants τ and c_0.

Both models are unable to describe traffic flow precisely. The first one yields a very strong coupling between the density and velocity and the second one admits waves traveling faster than the cars [32].

In 2000 Aw and Rascle proposed a model (AR model), which avoids the drawbacks of the above mentioned models. It is given by

$$\begin{aligned} \partial_t \rho + \partial_x (u\rho) &= 0 && (4) \\ \partial_t (u + p(\rho)) + u\partial_x (u + p(\rho)) &= 0, && (5) \end{aligned}$$

where the 'pressure' p satisfies the equation of state

$$p(\rho) = \rho^\gamma, \qquad \gamma > 0. \tag{6}$$

To describe traffic flow more precisely Greenberg [17] added a relaxation term, which corresponds to a 'velocity maximization', to the velocity equation of the AR model. He replaced the velocity equation by

$$\partial_t(u-v) + u\partial_x(u-v) = -\frac{u-v}{\delta}, \tag{7}$$

where $v = v(\rho)$ is a maximum velocity depending on the density such that $F(\rho) = \rho v(\rho)$ is concave and δ is a constant.

In this paper we present an approximate Riemann solver with algebraic flux correction of TVD type for the AR model and the extension of Greenberg. It is common knowledge, that the discretization of hyperbolic equations or systems of such equations induces numerical oscillations. The main task is to supress this wiggles. It turns out, that the techniques for scalar equations produce wiggles, if they are applied to the AR model [1]. Algebraic flux correction techniques for scalar equations can be found in [3] or [1]. Due to the strong coupling of equations (4), (5) and (6) it is necessary to discretize the system above as a whole. This can be done by means of a so called approximate Riemann solver as proposed by Roe [6]. Approximate Riemann solvers for the Euler equations can be found in [7] or [12]. In spite of their low accuracy, they produce nonoscillatory solutions and can serve as a starting point for the design of characteristic TVD schemes [7]. In this paper we apply the methodology developed by Kuzmin and Möller [4] for the Euler equations to the AR model.

After this introduction we give a short summary of analytic results. In the third section we construct an approximate Riemann solver based on an accurate high-order and a nonoscillatory low-order scheme combined in the framework of algebraic flux correction.

2 Analytical results

Before we start with the numerical methods, we summarize the results of Aw and Rascle [2]. For smooth solutions the AR model can be written as

$$\partial_t \rho + \partial_x(u\rho) = 0 \tag{8}$$

$$\partial_t u + (u - \rho\partial_\rho p(\rho))\partial_x u = 0 \tag{9}$$

using the chain rule. Substituting $U = (\rho, u)^T$ we get

$$\partial_t U + \underbrace{\begin{pmatrix} u & \rho \\ 0 & u - \rho p'(\rho) \end{pmatrix}}_{=:A(U)} \partial_x U = 0 \tag{10}$$

as a quasilinear form. The eigenvalues of the matrix A are

$$\lambda^2 = u - \rho p'(\rho) < u = \lambda^1. \tag{11}$$

Thus the system is strictly hyperbolic for $\rho > 0$ and the wave speeds are at most equal to the velocity of the cars. We do not consider the case $\rho = 0$, because a continuous model does not make sense for an empty road.

Moreover Aw and Rascle show that there is an invariant region

$$R = \{(\rho, u) | 0 \leq u \leq u_{\max} - p(\rho),\ \rho \geq 0,\ 0 \leq u \leq u_{\max}\}. \tag{12}$$

Furthermore they prove the existence of solutions for arbitrary Riemann data in R. From this follows that the solution is bounded from above and from below and remains nonnegative. The continuity equation yields the conservation of cars and there is a continuous dependence with respect to the data for positive densities.

3 An approximate Riemann solver

Now we turn to the numerical methods. The method employed in this paper is based on a nonoscillatory low-order scheme, which is derived from a suitable high-order scheme on the algebraic level. The quality of the resulting high-resolution scheme depends on the quality of the underlying low-order scheme. In particular it is necessary, that the low-order scheme is nonoscillatory.

The method proposed in this paper requires a diagonalization of the matrix A. In the case of system (10) the transformation matrices depend on $p'(\rho)$ and thus, they are variable for $\gamma \neq 1$. Therefore, special mean values in the sense of Roe [6] are necessary for the approximation on every edge / element (see section 3.1). To avoid such mean values, we transform the AR model to the primitive variables p and u. Let us consider the pressure equation

$$\partial_t p + u\partial_x p + w\partial_x u = 0 \tag{13}$$

with $p = \rho^\gamma$ and $w := \rho\partial_\rho p = \gamma p$. For smooth solutions this equation can be written as

$$\partial_\rho p\big(\partial_t \rho + \partial_x(\rho u)\big) = 0, \tag{14}$$

so that the continuity equation is recovered. Combining (13) and (9) we obtain the following system written in matrix form

$$\partial_t \begin{pmatrix} p \\ u \end{pmatrix} + \begin{pmatrix} u & w \\ 0 & u-w \end{pmatrix} \partial_x \begin{pmatrix} p \\ u \end{pmatrix} = 0, \tag{15}$$

which yields a diagonalization in terms of

$$\Lambda = R^{-1} A R. \tag{16}$$

The involved matrices are

$$R = R^{-1} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \quad A = \begin{pmatrix} u & w \\ 0 & u-w \end{pmatrix} \quad \text{and} \quad \Lambda = \begin{pmatrix} u & 0 \\ 0 & u-w \end{pmatrix}. \tag{17}$$

Hence, the transformation matrices are constant and the discrete counterparts are the same. Note that system (15) is exactly the same as system (10) in the special case $\gamma = 1$. Since the discretization process is essentially the same for $\gamma \neq 1$, it is sufficient to consider (15) with $p(\rho) = \rho$, which gives

$$\partial_t \begin{pmatrix} \rho \\ u \end{pmatrix} + \begin{pmatrix} u & \rho \\ 0 & u-\rho \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ u \end{pmatrix} = 0 \tag{18}$$

with eigenvalues

$$\lambda^1 = u, \qquad \lambda^2 = u - \rho \tag{19}$$

and

$$A(U) = \begin{pmatrix} u & \rho \\ 0 & u-\rho \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} u & 0 \\ 0 & u-\rho \end{pmatrix}. \tag{20}$$

It would be more convenient to deal with a system in conservation form like

$$\partial_t U + \partial_x F = 0 \tag{21}$$

with a flux function F, but this turns out to be impossible. Indeed, let us assume the existence of a flux function $F = (F_1, F_2)^T$ with $\partial_x F(U) = A(U)\partial_x U$. Then it follows from $\partial_\rho F_2 = 0$ that

$$F_2(U) = \int 0 d\rho + c(u) = c(u) \tag{22}$$

with an integration constant c depending only on u by the definition of A(U). The differentiation of F_2 with respect to u

$$\partial_u F_2(U) = \partial_u c(u) = u - \rho \tag{23}$$

yields a contradiction. Thus we cannot derive a flux function from A(U).

3.1 High-order scheme

The hyperbolic system to be solved is given by

$$\partial_t U + A(U)\partial_x U = 0. \tag{24}$$

We discretize this system using the nonconservative finite difference approximation

$$\begin{aligned} \frac{dU_i}{dt} &= -\frac{1}{2}\big(A_{i-\frac{1}{2}}\frac{U_i - U_{i-1}}{\Delta x} + A_{i+\frac{1}{2}}\frac{U_{i+1} - U_i}{\Delta x}\big) && (25)\\ &= \frac{1}{2\Delta x}\big(A_{i-\frac{1}{2}}(U_{i-1} - U_i) - A_{i+\frac{1}{2}}(U_{i+1} - U_i)\big) && (26)\\ &= \frac{1}{2\Delta x}\big(A_{i-\frac{1}{2}}U_{i-1} + (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})U_i - A_{i+\frac{1}{2}}U_{i+1}\big) && (27) \end{aligned}$$

with suitable approximations $A_{i\pm\frac{1}{2}}$, where Δx denotes the mesh size. To simplify notation, we will sometimes use the index I of the edge / element, which contains $x_{i\pm\frac{1}{2}}$, instead of $i \pm \frac{1}{2}$. The above discretization is equivalent to the nonconservative Galerkin discretization

$$\sum_i \int_\Omega \varphi_j \varphi_i dx \frac{dU_i}{dt} = -\sum_i \int_\Omega A(U_h)\varphi_j \partial_x \varphi_i U_i \, dx \tag{28}$$

with

$$U_h = \sum_i U_i \varphi_i \quad \text{and} \quad \partial_x U_h = \sum_i U_i \partial_x \varphi_i \tag{29}$$

using the group finite element formulation [23] (and mass lumping). We replace the consistent mass matrix by its lumped counterpart, which is given element-wise by

$$M_{LI} = \frac{\Delta x}{2} diag\{1,1,1,1\}. \tag{30}$$

The discrete transport operator can be written as

$$K_I = \frac{1}{2}\begin{pmatrix} A_{i+\frac{1}{2}} & -A_{i+\frac{1}{2}} \\ A_{i+\frac{1}{2}} & -A_{i+\frac{1}{2}} \end{pmatrix} \tag{31}$$

for the element $I = [x_i, x_{i+1}]$. The size of the matrices K_I and M_{LI} is 4×4. This yields the linear system

$$M_L \frac{dU}{dt} = KU \tag{32}$$

of size $2n \times 2n$ for n grid points.

It is still to be specified, how to determine the averages $A_{i\pm\frac{1}{2}}$. Setting $v = u - \rho$ we discretize the scalar equations of the form

$$\partial_t \rho + u\partial_x \rho + \rho\partial_x u = 0 \tag{33}$$

$$\partial_t u + v\partial_x u = 0 \tag{34}$$

by the nonconservative finite difference method

$$\begin{aligned} \frac{d\rho_i}{dt} &= \frac{u_{i-\frac{1}{2}}(\rho_{i-1} - \rho_i) - u_{i+\frac{1}{2}}(\rho_{i+1} - \rho_i)}{2\Delta x} \\ &\quad + \frac{\rho_{i-\frac{1}{2}}(u_{i-1} - u_i) - \rho_{i+\frac{1}{2}}(u_{i+1} - u_i)}{2\Delta x} \end{aligned} \tag{35}$$

$$\frac{du_i}{dt} = \frac{v_{i-\frac{1}{2}}(u_{i-1} - u_i) - v_{i+\frac{1}{2}}(u_{i+1} - u_i)}{2\Delta x}. \tag{36}$$

One easily sees, that the first equation (35) is equivalent to

$$\frac{d\rho_i}{dt} = \frac{\rho_{i-1}u_{i-1} - \rho_{i+1}u_{i+1}}{2\Delta x}, \tag{37}$$

which is the conservative Galerkin approximation of the continuity equation. Thus, the nonconservative finite difference discretization conserves mass, too.

In the case of the second equation (36), we obtain conservation only in the special case of the Burgers equation ($v = u$), i. e.

$$\partial_t u + u\partial_x u = 0. \tag{38}$$

Substitution $v_{i\pm\frac{1}{2}} = u_{i\pm\frac{1}{2}}$ yields

$$\frac{du_i}{dt} = \frac{u_{i-\frac{1}{2}}(u_{i-1} - u_i) - u_{i+\frac{1}{2}}(u_{i+1} - u_i)}{2\Delta x} = \frac{u_{i-1}^2 - u_{i+1}^2}{4\Delta x}, \tag{39}$$

which corresponds to the Galerkin discretization of the Burgers equation in conservative form

$$\partial_t u + \partial_x\left(\frac{1}{2}u^2\right) = 0. \tag{40}$$

Now we can write the system of discretized equations in vectorial form

$$\begin{aligned}\frac{dU_i}{dt} &= \frac{1}{2\Delta x}\underbrace{\begin{pmatrix} u_{i-\frac{1}{2}} & \rho_{i-\frac{1}{2}} \\ 0 & u_{i-\frac{1}{2}} - \rho_{i-\frac{1}{2}} \end{pmatrix}}_{=A_{i-\frac{1}{2}}} \begin{pmatrix} \rho_{i-1} - \rho_i \\ u_{i-1} - u_i \end{pmatrix} \\ &\quad - \frac{1}{2\Delta x}\underbrace{\begin{pmatrix} u_{i+\frac{1}{2}} & \rho_{i+\frac{1}{2}} \\ 0 & u_{i+\frac{1}{2}} - \rho_{i+\frac{1}{2}} \end{pmatrix}}_{A_{i+\frac{1}{2}}} \begin{pmatrix} \rho_{i+1} - \rho_i \\ u_{i+1} - u_i \end{pmatrix}. \end{aligned} \tag{41}$$

(42)

We define

$$A_{i\pm\frac{1}{2}} = \begin{pmatrix} u_{i\pm\frac{1}{2}} & \rho_{i\pm\frac{1}{2}} \\ 0 & u_{i\pm\frac{1}{2}} - \rho_{i\pm\frac{1}{2}} \end{pmatrix} \tag{43}$$

with

$$\rho_{i\pm\frac{1}{2}} = \frac{\rho_{i\pm1} + \rho_i}{2}, \quad u_{i\pm\frac{1}{2}} = \frac{u_{i\pm1} + u_i}{2} \quad \text{and} \quad v_i = u_i - \rho_i \tag{44}$$

and obtain equation (26). Thus, the high-order scheme, which was constructed above, is a suitable approximation of the scalar equations by the finite difference / Galerkin finite element method, which proves to be conservative in the case of the continuity equation and in the special case of the Burgers equation. The discretization is second order accurate and of course oscillatory.

3.2 Low-order scheme

To avoid numerical oscillations, we need a low-order scheme. We will construct such scheme on the algebraic level as explained in [4], [1]. An algebraic low-order scheme for scalar equations can be found in [1] or [3].

3.2.1 The LED property

A numerical scheme, which enjoys the LED property (local extremum diminishing) is nonoscillatory. This means in the scalar case, that the off-diagonal coefficients of the discrete transport operator are not negative. But in the case of our hyperbolic system, the entries of the transport operator are matrices. To enforce the LED property for systems, we need the following theorem [4]:

Theorem 3.1 *Consider the equation*

$$\frac{dU_i}{dt} = \frac{1}{m_i} \sum_{j \neq i} L_{ij}(U_j - U_i)$$

with 2×2 *matrices* L_{ij}*. They are assumed to be positive semi-definite.*

Then the scheme enjoys the LED property.

Thus, the LED property is fulfilled, if the off-diagonal blocks of the discrete transport operator contain no negative eigenvalues. We can enforce this constraint by eliminating negative eigenvalues of off-diagonal matrix blocks.

3.2.2 Artificial viscosity

The AR model is strictly hyperbolic. Hence, the matrix $A(U)$ is diagonalizable and the averages $A_{i\pm\frac{1}{2}}$ are diagonalizable by construction, too. We diagonalize the matrices using the factorization

$$\Lambda_{i\pm\frac{1}{2}} = R^{-1}_{i\pm\frac{1}{2}} A_{i\pm\frac{1}{2}} R_{i\pm\frac{1}{2}}. \tag{45}$$

Since R and R^{-1} are constant, the approximations on the discrete level are the same. Regarding (43) we choose

$$\Lambda_{i\pm\frac{1}{2}} = \begin{pmatrix} u_{i\pm\frac{1}{2}} & 0 \\ 0 & u_{i\pm\frac{1}{2}} - \rho_{i\pm\frac{1}{2}} \end{pmatrix}. \tag{46}$$

Like in the scalar case, we will enforce the LED property by adding a discrete diffusion operator [3], [1]. Discrete diffusion operators are symmetric matrices with vanishing column and row sums, which guarantees conservation. On the edge / element I we define

$$|\Lambda_I| = \begin{pmatrix} |u_I| & 0 \\ 0 & |u_I - \rho_I| \end{pmatrix} \tag{47}$$

and take

$$|A_I| = R_I |\Lambda_I| R_I^{-1} \tag{48}$$

as the off-diagonal diffusion coefficients. To fulfill the vanishing column and row sum property, we define the local diffusion matrix as

$$D_I = \frac{1}{2} \begin{pmatrix} -|A_I| & |A_I| \\ |A_I| & -|A_I| \end{pmatrix}. \tag{49}$$

This yields a local low-order operator of the form

$$L_I = \frac{1}{2}A_I + \frac{1}{2}|A_I| = \frac{1}{2}R_I\Lambda R_I^{-1} + \frac{1}{2}R_I|\Lambda_I|R_I^{-1} = \frac{1}{2}R_I(\Lambda_I + |\Lambda_I|)R_I^{-1}. \qquad (50)$$

Its global counterpart can be assembled edge by edge. It is important, that the added diffusion vanishes as the mesh is refined (in our case the diffusion is proportional to the mesh size). This rules out nonphysical solutions and yields a consistent scheme. But on the other hand, the low-order scheme is only first order accurate.

3.3 Algebraic flux correction of TVD type

After the addition of artificial diffusion, we wish to increase the accuracy of the low-order scheme. This can be done by adding antidiffusion. In order to make sure that the solution remains nonoscillatory, we introduce a flux limiter function Φ to control the amount of antidiffusion. Flux limiters of TVD type satisfying $0 \leq \Phi \leq 2$ can be found, e. g., in [25] or [13]. Usually the tools of algebraic flux correction are designed for scalar problems and the limiter functions are only applicable to scalar equations. Thus, it is necessary to decouple the equations for the algebraic flux correction.

3.3.1 Characteristic variables

Using a diagonalization of the Jacobi matrix, linear hyperbolic systems with constant coefficients can be transformed into a set of decoupled transport equations, which can be discretized and solved separately (see e. g. [14]). For arbitrary nonlinear systems this is not possible, because the transformation matrices depend on the desired quantities. In this case only a local approximate decoupling by suitable mean values (for the Euler equations see [6]) is possible. To avoid problems with the definition of such mean values, we use primitive variables, which yield

constant transformation matrices (see (17)). This enables us to apply the same transformation as in the case of constant coefficients. Moreover, the characteristic speeds given by the eigenvalues of our system are variable, so the factorization of A is to be performed locally, edge by edge. Hence, we will transform only locally [4].

We define the characteristic variables by

$$W := R^{-1}U. \tag{51}$$

This transformation will be used only to decouple the equations of the AR model and to perform algebraic flux correction. The nonlinear system will be solved in primitive variables (p, u) for the special case $p(\rho) = \rho$.

3.3.2 Algebraic flux decomposition

We consider on the edge I the local decoupled (scalar) equations

$$\partial_t W^k + \lambda^k \partial_x W^k = 0. \tag{52}$$

The fluxes defined in [3] are vectors with two entries. On the edge $I = [x_i, x_j]$ we can write the diffusive flux as

$$D_{ij}(U_j - U_i) = \frac{1}{2}|A_I|(U_j - U_i) = \frac{1}{2}R_I|\Lambda_I|R_I^{-1}(U_j - U_i) = \frac{1}{2}R_I|\Lambda_I|(W_j - W_i)$$

due to the zero row sum property of the diffusion operator. To determine the nodal correction factors we need the sums of positive and negative fluxes and the edge orientation [3]. They can be assembled from individual edge contributions, because the equations are decoupled on the edge I. For equation (52) the local

transport operator is given by

$$K_I^k = \frac{1}{2}\begin{pmatrix} \lambda_I^k & -\lambda_I^k \\ \lambda_I^k & -\lambda_I^k \end{pmatrix} \tag{53}$$

with mean values of the eigenvalues

$$\lambda_I^1 = \frac{u_i + u_j}{2} \qquad \text{and} \qquad \lambda_I^2 = \frac{u_i + u_j}{2} - \frac{\rho_i + \rho_j}{2}. \tag{54}$$

This yields the edge orientation for equation k. On the edge I, we have an upwind node i and a downwind node j=i+1, if $\lambda_I^k \geq 0$. Otherwise we obtain the opposite.

Now we determine the sums of positive and negative fluxes for the upwind node i and the downwind node j. The contribution of the edge I is given by [3]

$$P_i{}^{k^{\pm}}{}_{|_I} = -\frac{1}{2}|\lambda_I|^k \underset{\max}{\min}\{0, W_j^k - W_i^k\} \tag{55}$$

$$Q_j{}^{k^{\pm}}{}_{|_I} = \frac{1}{2}|\lambda_I|^k \underset{\min}{\max}\{0, W_i^k - W_j^k\}. \tag{56}$$

Let us consider the edge $I = [x_i, x_{i+1}]$ with upwind node i ($\lambda_I^k \geq 0$). This yields

$$P_i{}^{k^{\pm}}{}_{|_I} = -\frac{1}{2}\lambda_I^k \underset{\max}{\min}\{0, W_{i+1}^k - W_i^k\} \tag{57}$$

$$Q_{i+1}{}^{k^{\pm}}{}_{|_I} = \frac{1}{2}\lambda_I^k \underset{\min}{\max}\{0, W_i^k - W_{i+1}^k\}. \tag{58}$$

On the other hand we obtain

$$P_{i+1}{}^{k^{\pm}}{}_{|_I} = \frac{1}{2}\lambda_I^k \underset{\max}{\min}\{0, W_i^k - W_{i+1}^k\} \tag{59}$$

$$Q_i{}^{k^{\pm}}{}_{|_I} = -\frac{1}{2}\lambda_I^k \underset{\min}{\max}\{0, W_{i+1}^k - W_i^k\}, \tag{60}$$

if $i+1$ is the upwind node ($\lambda_I^k < 0$).

In an edge-based code, we can implement this as follows:
At first we define the local fluxes in characteristic variables by

$$f_I = -\frac{1}{2}\begin{pmatrix}\lambda_I^1 & 0\\ 0 & \lambda_I^2\end{pmatrix} R_I^{-1}(U_{i+1}-U_i) = -\frac{1}{2}\begin{pmatrix}\lambda_I^1 & 0\\ 0 & \lambda_I^2\end{pmatrix}(W_{i+1}-W_i). \tag{61}$$

Then we denote the upwind node by *up* and the downwind node by *dw* and update the vectors $P^{\pm}$ and $Q^{\pm}$ in a loop over all edges:

- If ${f_I}^k > 0$: Set

$$ {P_{up}^+}^k := {P_{up}^+}^k + {f_I}^k \qquad \text{and} \qquad {Q_{dw}^+}^k := {Q_{dw}^+}^k + {f_I}^k. \tag{62}$$

- If ${f_I}^k < 0$: Set

$$ {P_{up}^-}^k := {P_{up}^-}^k + {f_I}^k \qquad \text{and} \qquad {Q_{dw}^-}^k := {Q_{dw}^-}^k + {f_I}^k. \tag{63}$$

It is convenient to handle the four vectors $P^{\pm}$ and $Q^{\pm}$ as $2 \times n$ matrices, where the first row contains the nodal values for the first equation and the second one the nodal values for the second equation.

3.3.3 Determination of the antidiffusive correction factors

After the flux decomposition in the first loop over all edges, we can evaluate the limiter function and calculate the nodal correction factors in characteristic variables. Then we have to transform back to primitive variables

$$U = RW, \tag{64}$$

because we solve the algebraic system in primitive variables. For equation k let us define the nodal correction factors

$$R_i^{\pm k} = \Phi\Big(\frac{Q_i^{\pm k}}{P_i^{\pm k}}\Big) \tag{65}$$

and use them to compute the antidiffusive flux into the upwind node i [3]

$$F_{ij}^{ak} = \begin{cases} R_i^{+k} d_{ij}{}^k (W_i{}^k - W_j{}^k) & \text{falls} \quad W_i{}^k \geq W_j{}^k \\ R_i^{-k} d_{ij}{}^k (W_i{}^k - W_j{}^k) & \text{falls} \quad W_i{}^k < W_j{}^k \end{cases}. \tag{66}$$

The corresponding flux into the downwind node j is defined as

$$F_{ji}^{ak} = -F_{ij}^{ak}, \tag{67}$$

so that flux correction corresponds to adding a discrete (anti-)diffusion operator with off-diagonal entries $-R_i^{\pm} d_{ij}$. It is important to evaluate the limiter function only in the upwind node.

Recall that the transformation to the characteristic variables yields the local flux vector (61) on the edge $I = [x_i, x_{i+1}]$. Again we assume the upwind node i (i.e. $\lambda_I^k \geq 0$) and obtain the antidiffuive correction from (66) and (67):

1.) If $W_i{}^k - W_{i+1}{}^k > 0 \qquad$ (i. e. $f_I{}^k > 0$) ,

$$\begin{aligned} F_{i,i+1}^a{}^k &= 0.5 R_i^{+k} \lambda_I{}^k (W_i{}^k - W_{i+1}{}^k) \\ F_{i+1,i}^a{}^k &= 0.5 R_i^{+k} \lambda_I{}^k (W_{i+1}{}^k - W_i{}^k). \end{aligned}$$

2.) If $W_i{}^k - W_{i+1}{}^k < 0 \qquad$ (i. e. $f_I{}^k < 0$) ,

$$\begin{aligned} F_{i,i+1}^a{}^k &= 0.5 R_i^{-k} \lambda_I{}^k (W_i{}^k - W_{i+1}{}^k) \\ F_{i+1,i}^a{}^k &= 0.5 R_i^{-k} \lambda_I{}^k (W_{i+1}{}^k - W_i{}^k). \end{aligned}$$

If i+1 is the upwind node (i.e. $\lambda_I^k < 0$), we get

1.) if ${W_{i+1}}^k - {W_i}^k > 0 \qquad$ (i. e. ${f_I}^k > 0$),

$$\begin{aligned} {F^a_{i,i+1}}^k &= -0.5 {R^+_{i+1}}^k {\lambda_I}^k ({W_i}^k - {W_{i+1}}^k) \\ {F^a_{i+1,i}}^k &= -0.5 {R^+_{i+1}}^k {\lambda_I}^k ({W_{i+1}}^k - {W_i}^k). \end{aligned}$$

2.) if ${W_{i+1}}^k - {W_i}^k < 0 \qquad$ (i. e. ${f_I}^k < 0$),

$$\begin{aligned} {F_{i,i+1}}^k &= -0.5 {R^-_{i+1}}^k {\lambda_I}^k ({W_i}^k - {W_{i+1}}^k) \\ {F_{i+1,i}}^k &= -0.5 {R^-_{i+1}}^k {\lambda_I}^k ({W_{i+1}}^k - {W_i}^k). \end{aligned}$$

This can be written equivalently as

$${F^a_{i,i+1}}^k = 0.5|\lambda_I^k| \begin{cases} {R^+_{up}}^k (W_i^k - W_{i+1}^k), & {f_I}^k > 0 \\ {R^-_{up}}^k (W_i^k - W_{i+1}^k), & {f_I}^k < 0 \end{cases} \quad \text{and} \quad {F^a_{i+1,i}}^k = -{F^a_{i,i+1}}^k.$$

In the practical implementation we perform the above mentioned transformation to characteristic variables

$$\Delta W_I = R_I^{-1} (U_i - U_{i+1}) \tag{68}$$

and determine the local flux vector

$$f_I = \begin{pmatrix} \lambda_I^1 & 0 \\ 0 & \lambda_I^2 \end{pmatrix} \Delta W_I. \tag{69}$$

In the next step we update this local quantity:

- If ${f_I}^k > 0$: Set

$${f_I}^k := {R^+_{up}}^k \Delta {W_I}^k. \tag{70}$$

- If ${f_I}^k \leq 0$: Set

$$ {f_I}^k := {R_{up}^-}^k \Delta {W_I}^k. \tag{71} $$

Now that the antidiffusive flux is limited, it is necessary to transform back to primitive variables. Therefore, we update f_I once again:

$$ f_I := \frac{1}{2} R_I |\Lambda_I| f_I. \tag{72} $$

This is the limited antidiffusive flux for the edge I. We have to add it to the global vector, which contains the antidiffusive correction

$$ F_i^a := F_i^a + f_I \quad \text{and} \quad F_{i+1}^a := F_{i+1}^a - f_I. \tag{73} $$

We denote the resulting transport operator by $K^* = K + D + F^a = L + F^a$. This definition is only for theoretical reasons. In a practical implementation, we only need the matrices L and M_L, as described above. A solution computed by this algorithm is displayed in figure 1.

3.3.4 Time integration

After the space discretization described so far, we obtain a system of ordinary differential equations:

$$ M_L \frac{dU}{dt} = K^* U. \tag{74} $$

This system can be discretized in time by standard techniques for ordinary differential equations. We do not discuss time-stepping in detail and choose the well known θ-family $(0 \leq \theta \leq 1)$ of time-stepping methods (see e.g. [8])

$$ \frac{U^{n+1} - U^n}{\Delta t} = \theta \partial_t U^{n+1} + (1-\theta) \partial_t U^n + O\big((0.5-\theta)\Delta t, \Delta t\big), \tag{75} $$

where Δt denotes the length of the time step. The time step is assumed to be sufficiently small to guarantee stability and positivity preservation. Some special

cases are the explicit Euler scheme ($\theta = 0$), the implicit Euler scheme ($\theta = 1$) and the Crank-Nicolson method ($\theta = 0.5$), which is the only method of second order. To avoid step size restrictions, we can use the implicit Euler scheme, because it turns out to be unconditionally stable and positivity preserving, see [1] for details. Time discretization by a θ-scheme yields

$$\frac{U^{n+1} - U^n}{\Delta t} = \theta K^*(U^{n+1})U^{n+1} + (1-\theta)K^*(U^n)U^n. \tag{76}$$

The last open question is the solution of this nonlinear system.

3.3.5 Iterative defect correction

The high and low-order schemes are linear and they produce linear systems. However, the antidiffusive correction depends on the solution. This makes the system nonlinear and harder to solve, but it is not a drawback, because Godunov [11] showed, that a linear monotonicity preserving scheme is at most first order accurate. So we have to use a nonlinear method to increase the accuracy of the low-order scheme. We solve the nonlinear system, which is given by equation (76), by the fixed-point defect correction scheme. In what follows n denotes the time step and m is the index of the outer iteration. We avoid the assembly of the operator K^* and take $K^* = L + F^a$, where F^a contains the antidiffusive correction. This yields

$$\begin{aligned}(M_L - \Delta t\theta L^{n+1})U^{n+1} &= \Delta t\theta F^{a\,n+1} + (1-\theta)\Delta t L^n U^n \\ &\quad + (1-\theta)\Delta t F^{a\,n} + M_L U^n \end{aligned} \tag{77}$$

as an equivalent representation of equation (76). The part, which does not depend on the next time level n+1 is

$$B^n = (1-\theta)\Delta t L^n U^n + (1-\theta)\Delta t F^{a\,n} + M_L U^n. \tag{78}$$

We initialize the fixed-point iteration by

$$U_0^{n+1} := U^n, \qquad L_0^{n+1} = L^n \tag{79}$$

and use the preconditioner

$$A_m^{n+1} = M_L - \Delta t \theta L_m^{n+1}. \tag{80}$$

This matrix enjoys the M-matrix property and under a suitable step size restriction positivity and stability are preserved [1], [5] . Note that the method is unconditionally stable and positivity preserving for the implicit Euler scheme.

In the next step we solve the defect equation

$$A_m^{n+1} \Delta U_{m+1}^{n+1} = R_{m+1}^{n+1}, \qquad \text{with} \qquad \Delta U_{m+1}^{n+1} = U_{m+1}^{n+1} - U_m^{n+1} \tag{81}$$

for the residual

$$R_{m+1}^{n+1} = B^n - A_m^{n+1} U_m^{n+1} + \theta \Delta t {F^a}^{n+1}. \tag{82}$$

The updated solution is

$$U_{m+1}^{n+1} = U_m^{n+1} + \Delta U_{m+1}^{n+1}. \tag{83}$$

Figure 1 shows the exact solution, resulting from initial values

$$U_- = (6,5)^T, \quad U_+(1,2)^T \tag{84}$$

and the pressure $p(rho) = \rho$, of the AR model vs. the numerical ones produced by the high-order and low-order scheme as well as using algebraic flux correction based on the superbee limiter. The time discretization (see section 3.3.4) is performed by the implicit Euler method with step size 0.001 and the mesh size is 0.01. The low-order scheme and the superbee limiter are obviously immune to wiggles. As expected, the wiggles only occur in the solution of the high-order scheme. Fur-

thermore the solution computed by the superbee limiter is less smeared then the low-order solution.

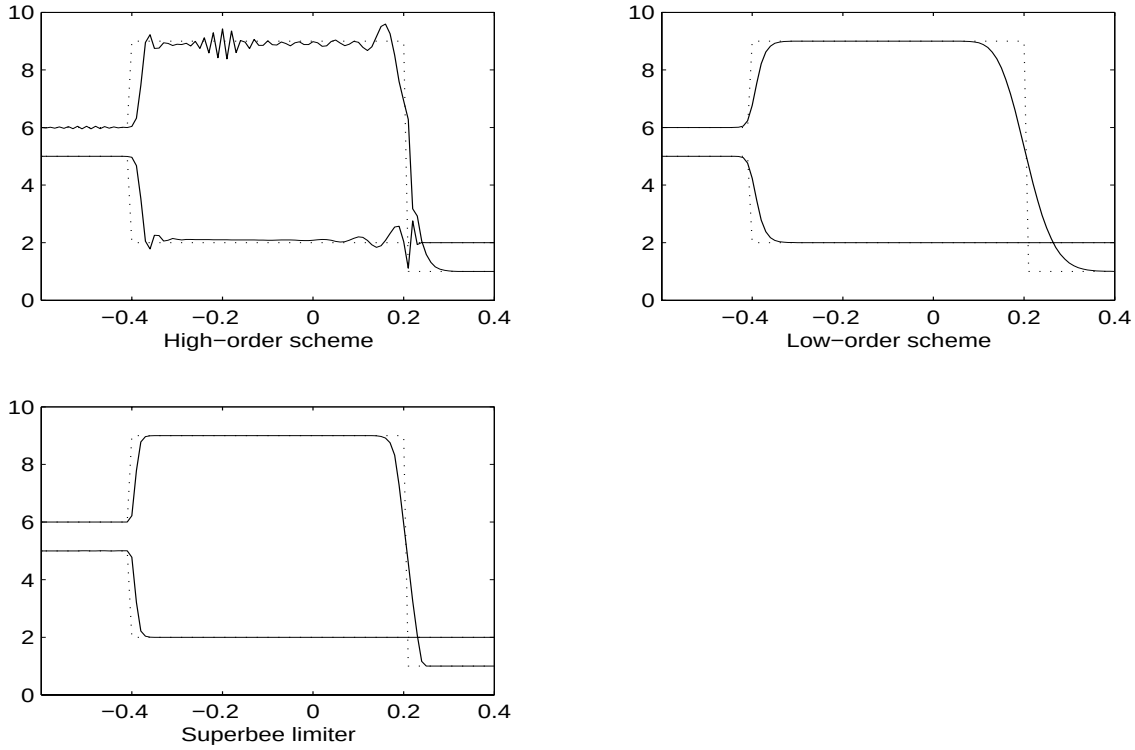

Figure 1: Comparison of numerical solutions to the AR model for the time T=1

4 Coupled vs. decoupled discretization

Another option for the numerical treatment of the AR model is to discretize the equations separately. This yields two coupled algebraic systems of equations and requires an outer iteration loop. This method is described in [1]. The main drawback of the technique are spurious wiggles produced even by the low-order scheme, although the scalar method applied to a single equation is completely

nonoscillatory. This renders the method useless for the system, because the algebraic flux correction is of antidiffusive character and increases the wiggles. They arise from slightly different speeds of error propagation in every equation. Therefore, the accuracy of the solution displayed in figure 2 (a) is corrupted by spurious wiggles. Figure 2 shows solutions of the AR model for initial data

$$U_- = (50, 200)^T \qquad \text{and} \qquad U_+ = (1, 10)^T \tag{85}$$

and the pressure function $p(\rho) = \rho$, which yields the intermediate state

$$U_0 = (240, 10)^T. \tag{86}$$

Hence, the solution produced by the decoupled solver yields a wrong prediction of U_0 (see figure 2 (a)). At the same time, the fully coupled approximate Riemann solver is nonoscillatory for the low-order scheme (b) and TVD limiters and yields the right intermediate state. Diagram (c) shows the solution computed by the superbee limiter.

5 The Greenberg model

In most traffic flow situations the drivers try to accelerate until they have reach a 'maximum velocity', which is given by v (see below) in our case. This behaviour is described by a source term. The new model (ARG model) reads [17]

$$\partial_t \rho + \partial_x (u\rho) = 0 \tag{87}$$

$$\partial_t (u - v) + u \partial_x (u - v) = -\frac{u - v}{\delta}, \tag{88}$$

where $v = v(\rho)$ is a maximum velocity depending on the density such that $F(\rho) = \rho v(\rho)$ is concave and δ is a constant. The discontinuity structure is almost the same as in the case of the AR model and there is an invariant region, too

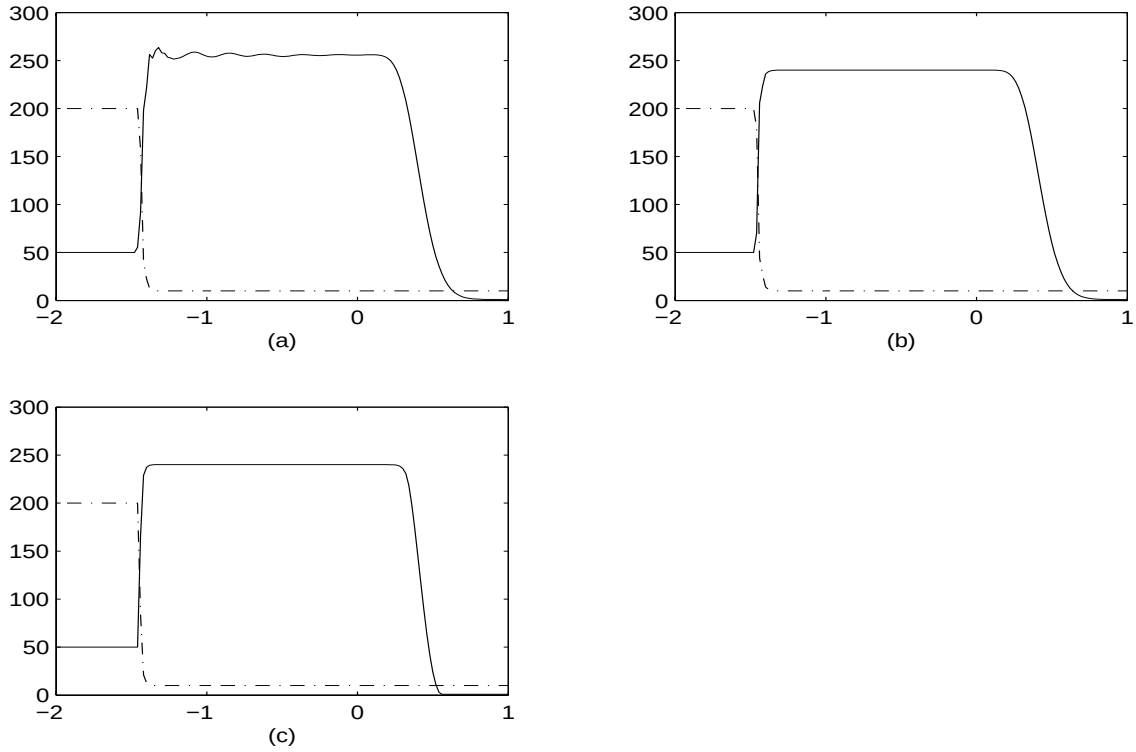

Figure 2: Numerical solutions of the AR model for $T = 0.04$

[17]. The velocity satisfies

$$0 \leq u \leq v(\rho), \tag{89}$$

if $0 \leq u_0 \leq v(\rho_0)$ holds. For smooth solutions this system can be written as

$$\partial_t \begin{pmatrix} \rho \\ u \end{pmatrix} + \begin{pmatrix} u & \rho \\ 0 & u + v'(\rho)\rho \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ u \end{pmatrix} = -\frac{1}{\delta} \begin{pmatrix} 0 \\ u - v(\rho) \end{pmatrix} \tag{90}$$

similar to the AR model.

5.1 Treatment of the source term

In contrast to convective terms, source terms are rather harmless and easy to implement. Using the group finite element formulation of Fletcher [23] the source term of the velocity equation is discretized by

$$Q_j = -\frac{1}{\delta}\sum_i (u_i - v_i)\int_\Omega \varphi_j \varphi_i \, dx. \tag{91}$$

We replace the consistent mass matrix by the lumped one and obtain

$$Q = -\frac{1}{\delta}\Delta x(u - v), \tag{92}$$

where u and v are the vectors of nodal values. At the boundary, the diagonal entries of M_L are given by $0.5\Delta x$, so that it is necessary to multiply Q by 0.5. As mentioned above we initialize R and B as $2 \times n$ matrices. Therefore, we implement the source term by updating the residual R_{m+1}^{n+1} and B^n, which yields

$$R_{m+1}^{n+1} := R_{m+1}^{n+1} + \left(0, -\frac{1}{\delta}\Delta x \Delta t \theta (u_m^{n+1} - v_m^{n+1})\right)^T \tag{93}$$

and

$$B^n := B^n + \left(0, -\frac{1}{\delta}\Delta x \Delta t (1-\theta)(u^n - v^n)\right)^T, \tag{94}$$

and multiply by 0.5 at the boundary.

Note that the source term can be assembled edge by edge, which corresponds to

$$Q_{j|_I} = -\frac{1}{\delta}(u_j - v_j)\frac{\Delta x}{2} \tag{95}$$

on the edge $I = [x_j, x_{j+1}]$.

5.2 Numerical examples

In this section we consider two different numerical examples. For both examples we choose $v(\rho) = 1 - \rho$ and $\delta = 1$. The initial data are given by

$$\rho_0 = \begin{pmatrix} \rho_- \\ \rho_+ \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0.7 \end{pmatrix} \quad \text{and} \quad u_0 = \begin{pmatrix} u_- \\ u_+ \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.1 \end{pmatrix} \tag{96}$$

for the first one and

$$\rho_0 = \begin{pmatrix} \rho_- \\ \rho_+ \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix} \quad \text{and} \quad u_0 = \begin{pmatrix} u_- \\ u_+ \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} \tag{97}$$

for the second example. Note that the initial data satisfies $0 \leq u_0 \leq v(\rho_0)$ in both cases. Figure 3 displays a comparison of the solutions of the AR and the ARG model for both examples. They are computed using the implicit Euler method for time integration, a mesh size of 0.002 and time steps of the length 0.0001. The fluxes are limited by the superbee limiter.

The first example involves a backwards moving shock wave and a forward moving contact discontinuity (for both models). The shock wave arises from breaking, which increases the density and decreases the velocity behind the breaking cars. After the shock the behavior of the solutions of the two models differs very much. In the solution of the AR model the cars maintain their velocity, while the ARG solution predicts acceleration corresponding to the low density. This is clearly observed in traffic flow situations and an advantage of the ARG model.

The second example shows a contact discontinuity followed by a rarefaction wave, which comes from fast driving cars in front of slower ones and corresponds to accelerating. Again the velocity of the AR solution remains constant after the first wave. In contrast, the ARG solution exhibits acceleration after the rarefaction wave, because of the decreasing density as expected in traffic flow. For both examples the phenomenon of 'velocity maximization' is displayed in figure 4. We

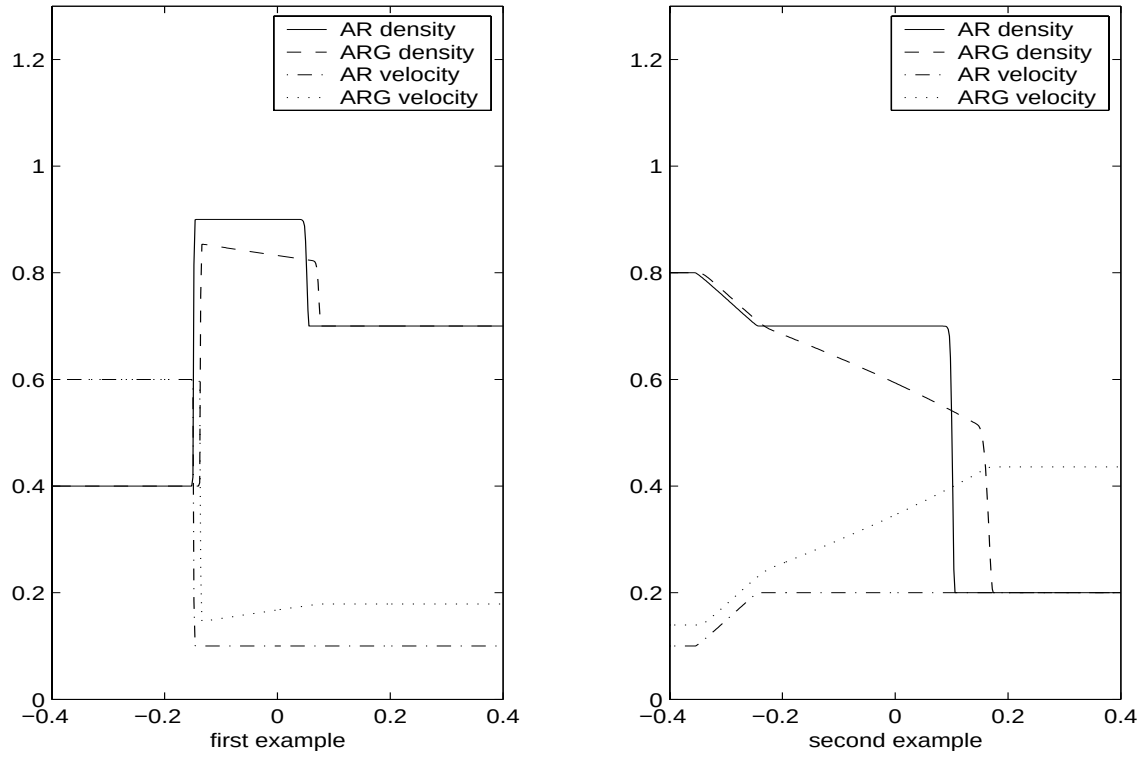

Figure 3: Comparison of numerical solutions to the AR and ARG model

observe, that 'velocity maximization' is only predicted by the ARG model, while the AR model produces solutions, which are constant out of waves. Hence, the ARG model yields a better describtion of traffic flow than the AR model.

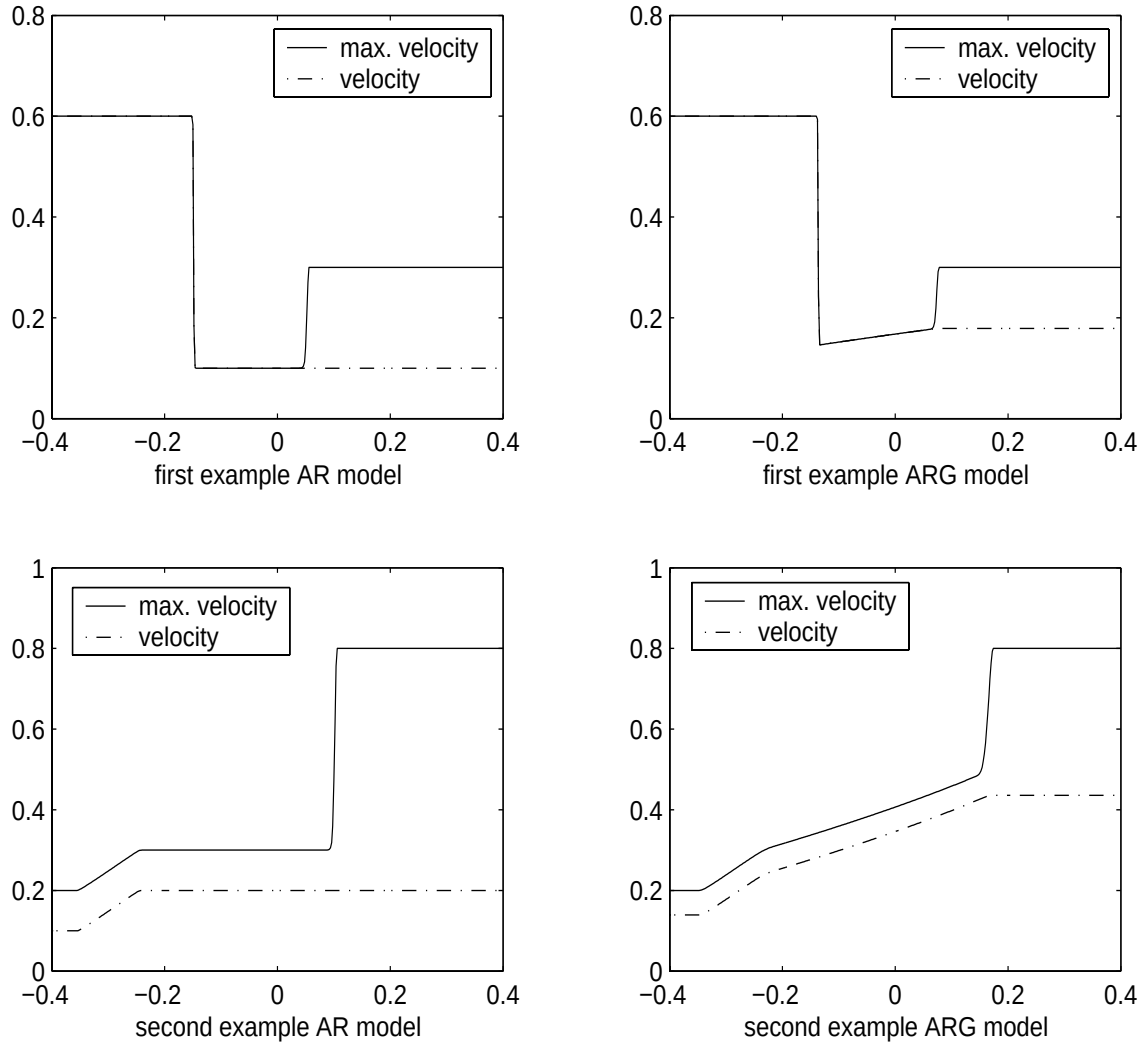

Figure 4: Car velocity vs. maximum velocity

6 Conclusions

A nonoscillatory discretization scheme for the Aw and Rascle model and the extension of Greenberg is proposed. It is based on a so called approximate Riemann solver. This class of schemes was developed by Roe [6] for the Euler equations of gas dynamics. Roe used special mean values for the flux approximation on every edge. We can aviod this, by solving the pressure equation instead of the continuity equation. Then the transformation matrices are constant and it suffices to take the arithmetic averages instead of the Roe mean values.
We tailor the scheme, proposed by Kuzmin and Möller [4] for the Euler equations, to the AR model. The underlying flux correction technique employs node-based limiters of TVD type as described in [3] and generalized to hyperbolic systems in [4]. As another option we can apply the scalar limiting techniques to the individual equations and couple the discretized equations by an outer iteration loop. However, even the resulting low-order scheme turns out to be oscillatory, which makes it useless. The AR model considered in this paper is designed for a single road without entrances, exits and traffic signals. Its generalization to road networks is feasible and will be addressed in future work.

References

[1] M. Gurris, *Makroskopische Differentialgleichungsmodelle und Diskretisierung des Modells von Aw und Rascle mit einem hochauflösenden Verfahren vom TVD-Typ*, Diploma thesis, University of Dortmund, 2006

[2] A. Aw und M. Rascle, *Resurrection of second order models of traffic flow*, SIAM J. Appl. Math., Vol. 60, p. 916-938, 2000

[3] D. Kuzmin, M. Möller, *Algebraic flux correction I. Scalar conservation laws*, Reports on findings of the Institut of Applied Mathematics, No. 249, Department of Mathematics, University of Dortmund, 2004

[4] D. Kuzmin, M. Möller, *Algebraic flux correction II. Compressible Euler equations*, Reports on findings of the Institut of Applied Mathematics, No. 250, Department of Mathematics, University of Dortmund, 2004

[5] D. Kuzmin, S. Turek, *Flux correction tools for finite elements*, J. Comput. Phys., Vol. 175, p. 525-558, 2002

[6] P. L. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comput. Phys., Vol. 43, p. 357-372, 1981

[7] A. Harten, H. C. Yee, R. F. Warming, *Implicit total variation diminishing (TVD) schemes for steady-state calculations*, J. Comput. Phys., Vol. 57, p. 327-360, 1985

[8] Jean Donea, Antonio Huerta, *Finite element method for flow problems*, Wiley 2003

[9] P. D. Lax, *Hyperbolic systems of conservation laws*, II Comm. Pure Appl. Math., Vol. 10, p. 537-566, 1957

[10] P. D. Lax, *Hyperbolic systems of conservation laws and the mathematical theory of schock waves*, SIAM Publications, Philadelphia, 1973

[11] S. K. Godunov, *Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics*, Mat. Sbornik 47, p. 271-306, 1957

[12] Randall J. LeVeque, *Numerical methods for conservation laws*, Birkhäuser, 1999

[13] A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen*, Habilitation thesis, University of Stuttgart, 2002

[14] Randall J. LeVeque, *Nonlinear conservation laws and finite volume methods for astrophysical fluid flow*, Springer, 1998

[15] J. A. Smoller, *Schock waves and reaction-diffusion equations*, Springer Verlag, Berlin 1983

[16] B. Temple, *Systems of conservation laws with coinciding shock and rarefaction curves*, Contemp. Math, Vol. 17, p. 143-151, 1983

[17] J. M. Greenberg, *Extensions and amplifications of a traffic model of Aw and Rascle*, SIAM J. Appl. Math., Vol. 62, No. 3, p. 729-745, 2001

[18] F. Siebel, W. Mauser, *On the fundamental diagram of traffic flow*, SIAM J. Appl. Math., Vol. 66, No. 4, p. 1150-1162, 2006

[19] B. Piccoli, G. Mauro, *Traffic flow on a road network using the Aw-Rascle model*, Commun. Partial Differ. Equations, No. 1-3, p. 243-275, 2006

[20] M. Herty, A. Klar, *Modeling, simulation and optimization of traffic flow networks*, SIAM J. Appl. Math., Vol. 35, No. 3, p. 1066-1087, 2002

[21] A. Klar, R. Wegener, *A hierarchy of models for multilane vehicular traffic I: Modelling*, SIAM J. Appl. Math., Vol. 59, No. 3, p. 983-1001, 1999

[22] A. Klar, R. Wegener, *A hierarchy of models for multilane vehicular traffic II: Numerical investigations*, SIAM J. Appl. Math., Vol. 59, No. 3, p. 1002-1011, 1999

[23] C. A. J. Fletcher, *The group finite element formulation*, Comput. Methods Appl. Mech. Engrg., Vol 37, p. 225-243, 1983

[24] A. Jameson, *Computational algorithms for aerodynamic analysis and design*, Appl. Numer. Math., Vol. 13, p. 383-422, 1993

[25] A. Jameson, *Analysis and design of numerical schemes for gas dynamics I. Artificial diffusion, upwinding biasing, limiters and their effect on accuracy and multigrid convergence*, Int. J. of CFD, Vol. 4, p. 171-218, 1995

[26] A. Jameson, *Positive schemes and shock modelling for compressible flows*, Int. J. Numer. Meth. Fluids, Vol. 20, p. 743-776, 1995

[27] H. J. Payne, *Models of freeway traffic and control*, Simulation Council, 1971

[28] M. J. Lighthill, J. B. Whitham, *On kinematic waves I: Flow movement in long rivers II: A theory of traffic flow on long crowded roads*, Proc. Royal Soc. London, A229, p. 281-345, 1955

[29] G. B. Whitham, *Linear and nonlinear waves*, Pure and Appl. Math., Wiley-Interscience Monographs and Tracts, New York, 1974

[30] P. I. Richards, *Shock waves on the highway*, Operations research, No. 4, p. 42-51, 1956

[31] R. Haberman, Mathematical models: *Mechanical vibrations, population dynamics and traffic flow*, Prentice-Hall, 1977

[32] C. Daganzo, *Requiem for second-order fluid approximation to traffic flow*, Transp. Res. B., Vol. 29B, No. 4, p. 277-286, 1995

[33] H. Holden, N. H. Risebro, *A mathematical model of traffic flow on a network of unidirectional roads*, SIAM J. Appl. Math. Anal., Vol. 26, No. 4, 1995

[34] J. Shi, X. Zhouping, *The relaxation schemes for systems of conservation laws in arbitrary space dimensions, Comm. on Pure and Appl. Math.*