Implicit and explicit numerical methods for macroscopic traffic flow models
Draft November 13, 2008

Submission date: November 13, 2008

Word count
5324 words + 7 figures ×250 + 1 table ×250 → 7324 'words'

Authors:
Femke van Wageningen-Kessels
TRAIL Research School
Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands
f.l.m.vanwageningen-kessels@tudelft.nl, gphone: +31 15 27 84066

Hans van Lint
Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands
j.w.c.vanlint@tudelft.nl, phone: +31 15 27 85061

Serge P. Hoogendoorn
Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands,
s.p.hoogendoorn@tudelft.nl, phone: +31 15 27 85475

Kees Vuik
Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
c.vuik@tudelft.nl, phone: +31 15 27 85530

Abstract Macroscopic models are used to describe traffic flow. In a simulation the model equations are discretized in both space and time. Time discretization, despite its importance to numerical stability and accuracy, has not received much attention. Most current implementations apply explicit time integration methods, which need to obey to strict stability conditions. These may result in large computing times and prevent the implementation of macroscopic traffic flow models to applications where computational efficiency is crucial, such as real time applications.

We describe and study implicit time integration methods which have less strict stability conditions, can be used with larger time steps and reduce the computing time.

Furthermore, we study the accuracy of the numerical methods. The widely used root mean square error does not take into account the nature of the error. We propose two accuracy measures which take into account errors that are important to applications in traffic flow. The phase error measures a shift of the solution over time. Numerical diffusion indicates that the solution is too 'smooth'.

We compare explicit and implicit schemes with simple test problems for computing time and accuracy. We found that implicit schemes, in these cases, result in 9 to 15 times smaller computing times. However, the accuracy and the nature of the error depends on the time integration method, but can be kept at an acceptable level for computationally efficient schemes. This shows that implicit time integration methods can play a key role in applications where small computing times are crucial.

1 INTRODUCTION

Macroscopic traffic flow models have been widely used since the 1950's. The first model by Lighthill and Whitham and, independently, Richards (*1, 2*) was based on the conservation of vehicles and the relation between the flow and the density. Since then so-called higher-order models have been developed, and many generalizations have been proposed (*3*).

After the mathematical model has been defined, a numerical simulation is used to find the solution to the model equations. To solve these equations numerically, they somehow have to be discretized both in space and time. For the spatial discretization several methods have been proposed, one of the best known is the minimum supply demand scheme (also known as Godunov's scheme) as described by Lebacque (*4*). In this scheme the spatial derivative of the flow is approximated as follows:

$$\frac{dq_i}{dx} \approx \frac{q_{i\to i+1} - q_{i-1\to i}}{\Delta x}, \tag{1}$$

with $q_{i\to i+1} = \min(d_i, s_{i+1})$ the flux from cell $i-1$ to i and Δx the spatial grid cell size and the demand d_i and supply s_i of cell i expressed as functions of the flow q_i and density ρ_i of cell i. The demand and supply functions, which can be seen as flux limiter functions, are determined as follows:

$$d_i = \begin{cases} q_i, \\ q_{\max,i}, \end{cases} \quad s_i = \begin{cases} q_{\max,i}, & \text{for } \rho_i < \rho_{\text{critical}} \text{ (no congestion)} \\ q_i & \text{for } \rho_i, \geq \rho_{\text{critical}} \text{ (congestion)} \end{cases} \tag{2}$$

This means that $q_{i-1\to i}$ is either q_{i-1}, q_i or $q_{\max}$ depending on whether the cells $i-1$ and i are congested or not. This method, in a modified form, is for example used in Fastlane (*5*). Alternatively, a Lax-Wendroff scheme can be applied. However, this is usually applied with explicit time stepping schemes (see Section 2) and the absence of a flux-limiter function can lead to oscillations.

However, only few researchers have considered time discretization. In most cases, simple 'explicit' schemes are used (*6*), which place a severe constraint on the size of the time step for stability of the numerical method, which in turn leads to large computing times. Adaptive discretization methods for both spatial as temporal discretization has been applied by Babcock et al (*7*) and Cho et al (*8*). These methods result in smaller computing times, but the present implementations are limited to explicit schemes, and the time step size is restricted by a stability condition. In the early 1990's Chronopoulos et al applied 'implicit' time schemes on macroscopic traffic flow models (*9, 10*). They found that a reduction of the computing time by a factor 2 to 3 can be achieved by using *larger time steps*. Because the time steps are larger *less time steps* are necessary for the *same computed (simulated) time*. Computational efficiency is crucial for applications such as real time traffic flow modelling. Application of alternative numerical time integration methods may hence play a key role in the necessary reduction of computing times.

In this paper we further study the implementation of implicit time schemes on macroscopic traffic flow models. We find that the computational efficiency can even be improved more than was indicated in previous work, namely 9 to 15 times. In contrast to the work of Chronopoulos et al we focus on the accuracy of the solution and the kind of errors being made. Usually the root mean square error (RMSE) is used to assess the quality of numerical solutions. It averages all kinds of errors, while the location and nature of the error can be important. For example in applications used for traffic control the exact location of the congested area is of great importance. The error in the predicted location of this area can be measured by the 'phase error'. We also propose a third error measure that measures the diffusion ('smoothening') of the solution introduced by the numerical method.

2 EXPLICIT AND IMPLICIT TIME INTEGRATION

In this paper we show the results for simple test problems with the equation for conservation of vehicles:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0, \tag{3}$$

with ρ the density and

$$q(\rho) = v(\rho)\,\rho, \tag{4}$$

the flow as a function of the density. The system of equations is closed by the 'fundamental relationship': the equilibrium relation between the velocity v and the density.

If we apply an explicit method to (3), this results in the semi-discretized system

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} + \frac{dq^n}{dx} = 0, \quad \Rightarrow \tag{5}$$

$$\rho^{n+1} = \rho^n + \Delta t \frac{d(q\rho^n)}{dx}, \tag{6}$$

with Δt the time step size and $\rho^n = \rho(n\Delta t)$ the density at the n-th time step.

However, if we use 'implicit' time integration, the size of the time step does not affect the stability of the numerical method. This means that we can take a much larger time step than with an explicit method. With an implicit scheme the semi-discretized version of (3) becomes:

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} + \frac{dq^{n+1}}{dx} = 0, \Rightarrow \tag{7}$$

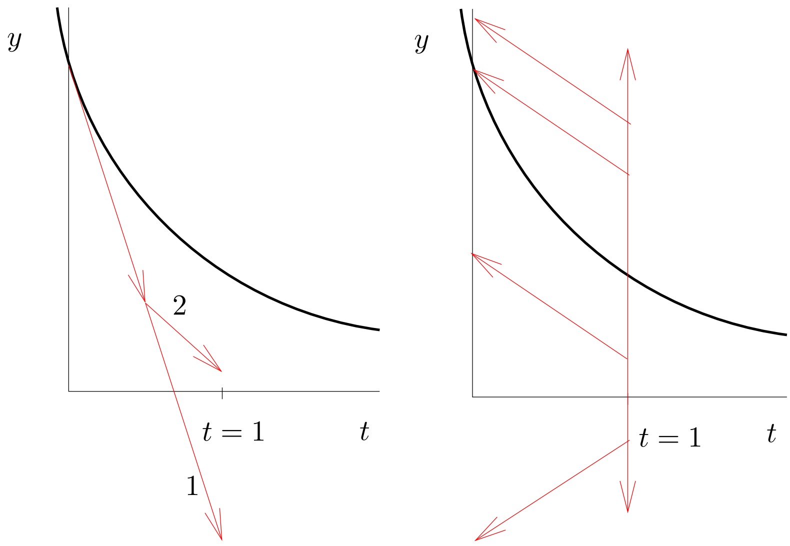
$$\rho^{n+1} + \Delta t \frac{dq(\rho^{n+1})}{dx} = \rho^n. \tag{8}$$

Note that in the above equations (5)–(8) we only show the time discretization; we do not consider the spatial discretization of dq/dx.

The main difference between the explicit and implicit scheme is the time instance at which the flow is computed. In the explicit scheme, first the flow at the present time step (q^n) is computed, subsequently the resulting density at the next time step (ρ^{n+1}) is determined. In contrast, in the implicit scheme the flow and density at the next time step (q^{n+1} and ρ^{n+1}) are computed simultaneously. The difference between an explicit scheme and an implicit scheme is illustrated in Figure 1. Note that over the period $[t, t + \Delta t]$ flows are considered constant. Therefore, there is no principle preference in using either dq^n/dx or dq^{n+1}/dx.

Implicit schemes are not often used for hyperbolic systems such as (3) (*11*). However, we will show that in some cases they can be advantageous. The main drawback of implicit schemes is the introduction of non-linearity. Clearly the left-hand side of (6) is linear, while the left hand side of (8) is not, since $q(\rho)$ is not linear and the spatial discretization of dq/dx might be non-linear. Non-linearity can be overcome in several ways. First we can use an IMEX method (*12*, Section IV.4). Second we can linearize (8) for example with Newton linearization (*13*, Section 10.2). Below we will shortly explain the IMEX method and Newton linearization.

IMEX method IMEX stands for implicit-explicit; it is a method which combines an implicit with an explicit scheme. By taking the appropriate terms either explicit or implicit, the equations



(a) (b)

FIGURE 1 Illustration of explicit time scheme (a) and implicit time scheme (b). The heavy black line denotes the exact solution. The goal is to find the solution at $t = 1$, starting from the solution at $t = 0$. To find the solution with the explicit time scheme we use the slope at $t = 0$. If we take one large time step (arrow 1) the solution becomes negative. This can be the start of an oscillation. For a more accurate solution we need to take two small time steps (arrow 2). An implicit method, however, searches for the solution at $t = 1$ directly, using the slope at $t = 1$. Therefore, several values of y at $t = 1$ are tried until the right one is found.

can become linear again. In our case IMEX is applied to make the left-hand side of (8) linear in ρ^{n+1}. Since by definition $q = v\rho$ we can use $q^{n+1} \approx v^n \rho^{n+1}$. Substituting this into (8) results in

$$\rho^{n+1} + \Delta t \frac{d(v^n \rho^{n+1})}{dx} \quad = \quad \rho^n. \tag{9}$$

By choosing an appropriate spatial discretization scheme we find a matrix A such that the discretized version of (9) is linear and we can write:

$$A\rho^{n+1} = \rho^n. \tag{10}$$

Note that we still use the velocity at the old time step, which might result in instabilities for large time step sizes.

If, for example, we use the minimum supply demand scheme (1)–(2) on a homogeneous road and if there is no congestion ($\rho_i < \rho_{\text{critical}}, \ \forall i$) we have the spatial discretization $\frac{d(v\rho)}{dx} \approx \frac{(v\rho)_i - (v\rho)_{i-1}}{\Delta x}$. This results in the matrix

$$A = \begin{pmatrix} 1 + \frac{\Delta t}{\Delta x} v_1^n & & & \emptyset \\ -\frac{\Delta t}{\Delta x} v_1^n & 1 + \frac{\Delta t}{\Delta x} v_2^n & & \\ & \ddots & \ddots & \\ \emptyset & & -\frac{\Delta t}{\Delta x} v_{I-1}^n & 1 + \frac{\Delta t}{\Delta x} v_I^n \end{pmatrix}. \tag{11}$$

If there is congestion for the whole domain, the diagonal just below the main diagonal will become zero, and the diagonal just above the main diagonal will contain entries $-\frac{\Delta t}{\Delta x} v_{i+1}^n$.

Newton linearization Newton linearization (a gradient method) is one of the most simple and easy-to-implement linearization methods. Within one time step we have several consecutive Newton iterations that lead us to the approximate solution of (8). In every Newton iteration we determine in which direction the exact solution of (8) can be found and a step in that direction is made. This can also be written more mathematically. Assume know $x_0 = x^{\text{old}}$ and we want to find $x = x^{\text{new}}$ such that $F(x) = 0$. Newton's method comes down to applying the following algorithm:

1. Initialize: $k = 0$.
2. Compute (approximate) $F'(x_k)$, the Jacobian of F at x_k.
3. Compute $x_{k+1} = x_k - (F'(x_k))^{-1}F(x_k)$.
4. If $F(x_k)$ is small enough, stop, else $k := k + 1$ and return to step 2.

This way every time step the new ρ^{k+1} can be approximated. In every Newton iteration we need the Jacobian of the left hand side of (7) with respect to the new desities ρ^{n+1}. It is a costly operation to determine (or approximate) the Jacobian. However, this can be compensated by using larger time steps.

3 STABILITY OF NUMERICAL METHODS

Any numerical method introduces approximation errors. A numerical method is called *stable* if these approximation errors do not magnify over time.

Numerical stability is illustrated in Figure 2. The solution to an initial value problem is shown. Initially we have a high density left of $x = 0$ and a low density right of it. We apply the LWR-model (3), (4) with Greenshields fundamental relation (16). We expect that the boundary between the high and the low density moves to left (acceleration fan), as can be seen from the exact solution (red dotted line). For the discretization we use a fixed size for the spatial grid cells. An explicit method is used to compute the numerical solution. If we chose a small time step (see Figure 2(a)) the computational solution is too 'smooth', but the approximation error does not magnify; the method is stable. This smoothening is further discussed in Section 4. With a medium sized time step (see Figure 2(b)) there is no approximation error and the method is stable. If the time step is large (see Figure 2(c)) the approximation error magnifies and 'wiggles' or oscillations are developed. This is due to numerical instability of the method. It is clear that this method can not be used with this time step size.

Von Neumann stability analysis (*12*, Chapter 6) can be used to derive the stability conditions. The derivation itself is beyond the scope of this paper. For explicit methods we find that the method is stable if it satifies the CFL-condition

$$\frac{\Delta t}{\Delta x} v \leq 1. \tag{12}$$

This means that the method is unstable if a vehicle can cross more than one grid cell within one time step. We furthermore find that implicit methods are unconditionally stable. This means that the time step is not restricted by any stability conditions.

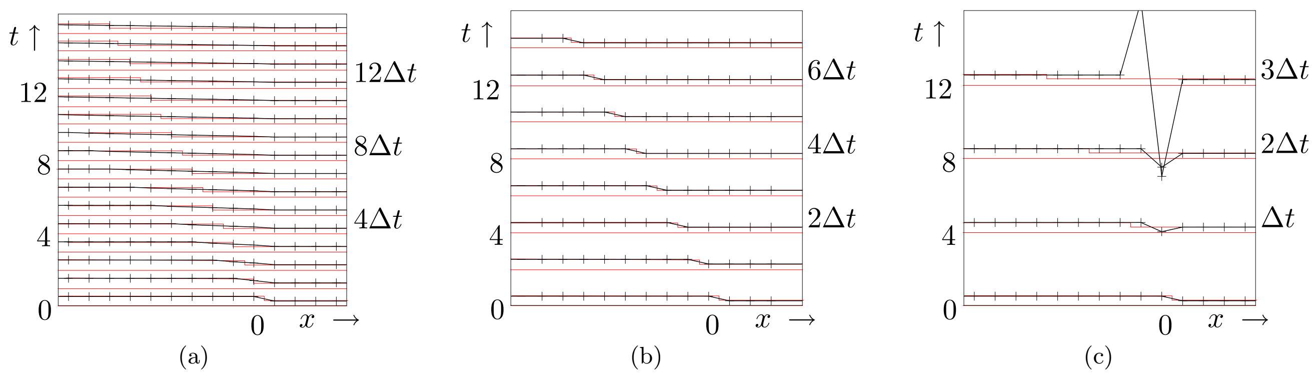


FIGURE 2 Exact (red, dottes line) and numerical (black, heavy line) solutions to initial value problem with different time step sizes.

4 ACCURACY OF NUMERICAL METHODS

The accuracy of a numerical method can be considered in several ways. In evaluation studies of traffic flow models the most commonly used measure is the root mean square error (RMSE):

$$\text{RMSE} = \left\| \frac{y - \tilde{y}}{\sqrt{I}} \right\|_2 = \sqrt{\frac{\sum_i (y_i - \tilde{y}_i)^2}{I}}, \tag{13}$$

with y the 'exact' solution (from data or from the analytical solution) $\tilde{y}$ the computed solution and I the total number of grid cells. In the solutions y and $\tilde{y}$ for example the density or the flow can be used. Note that any information on the location or the nature of the error is lost, as illustrated in Figure 3. Here the exact and two numerical solutions to a shock wave problem are shown. Note that even though the nature of the errors in the numerical solutions is very different, the RMSE is equal for both solutions.

The RMSE measures the error in 2-norm, alternatively, the error can be measured in 1-norm or in maximum-norm, which have the same drawbacks as sketched above. Therefore, we propose to consider two new accuracy measures: the phase error of the numerical method and the numerical diffusion.

Phase error A phase error indicates that the solution has shifted over space, see Figure 3(b). In traffic flow this means for example that congestion is predicted to develop at the wrong location and/or time, or that the location of a shock is computed wrongly. The optimal control measures in traffic flow depends on the exact location of congested areas and shocks, especially when they arise near ramps, bifurcations, merges or intersections. Therefore, in practical applications phase errors can cause severe problems.

The phase error is more complex to compute than the RMSE. In our test code we compute the phase error introduced by the numerical method by comparing the computed location of a shock or a fan to its location in the analytical solution:

$$\text{phase error} = \frac{\text{computed location of shock/fan - exact location of shock/fan}}{\text{time since start of simulation}}. \tag{14}$$

Note that the phase error is expressed as a velocity. It can be interpreted as the error in the characteristic velocity. In Figure 4(a) is illustrated how a value is assigned to the phase error.

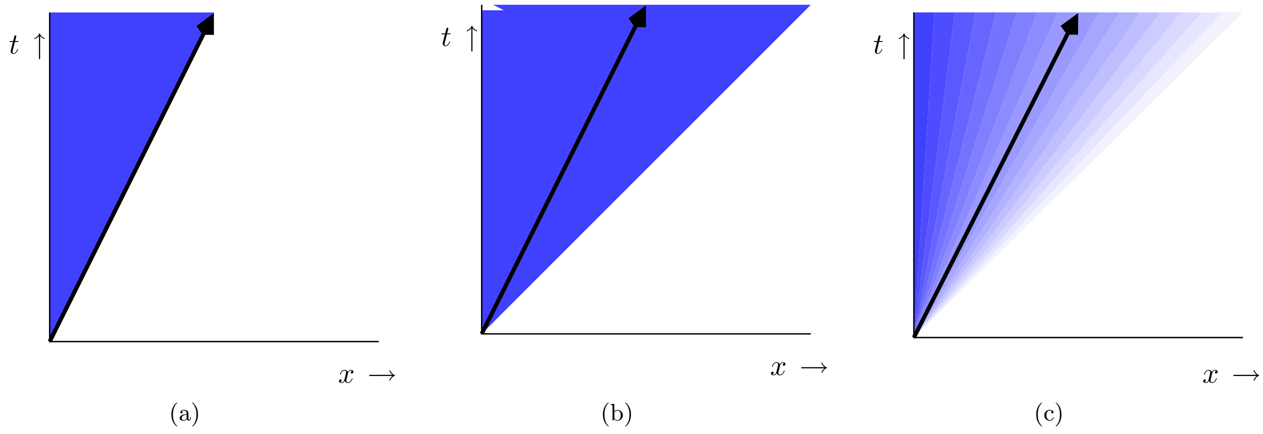


FIGURE 3 Illustration of errors in the solution of a shock wave problem. (a) Exact solution. (b) A numerical solution that shows a large phase error, and no diffusion. (c) An other numerical solution that shows much numerical diffusion, and no phase error. Note that for solution (b) and (c) the RMSE would be equal.

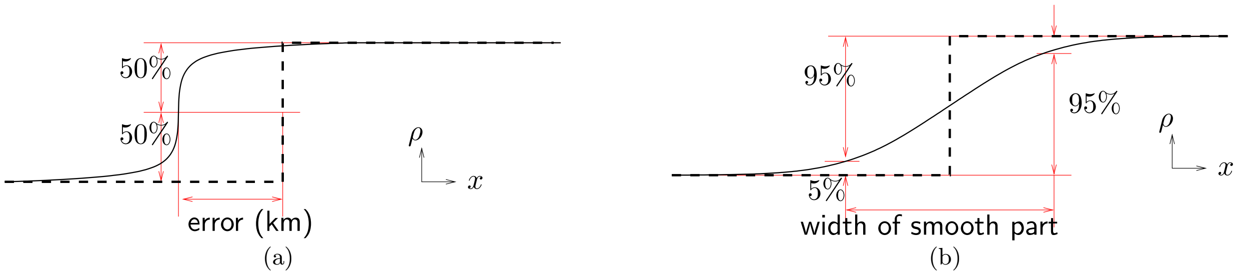


FIGURE 4 Phase error (a) and numerical diffusion (b), broken line: analytical solution, solid line: computed solution.

Numerical diffusion Numerical diffusion results in a solution that is too 'smooth', see Figure 3(c). A sudden increase or decrease in traffic densities (or velocities or flows) will disappear and shocks might not develop.

Similar to the phase error, the numerical diffusion is more complex to compute than the RMSE. In our test code we compute it by comparing the width of the 'smooth' part in the exact solution (which can be zero) to its width in the computed solution (which is nonzero if there is any numerical diffusion):

$$\text{numerical diffusion} = \frac{\text{computed width - exact width}}{\text{time since start of simulation}}. \tag{15}$$

Note that the numerical diffusion is expressed as the ratio of a length over a time, which can be seen as a velocity. Therefore, its evolution can be followed over time. In Figure 4(b) is illustrated how a value is assigned to it.

It can be argued that some numerical diffusion does not pose a great problem when the LWR model is used. In contrast to higher order models, which are supposed to be more realistic in this sense, there is no diffusion in the LWR model. Some numerical schemes do introduce numerical diffusion, which can make the results more realistic. However, one should be aware that the diffusion is not in the mathematical (LWR) model itself, but stems from the numerical method and is influenced by it.

Note that this simple methods to measure the phase error and numerical diffusion are only valid for our simple test problems described hereafter, and will not be valid if, for example the characteristic speed of the shock is not constant. Therefore, the phase error and numerical diffusion are more complex to analyse in computational experiments than the RMSE. There are no off-shelf methods available to identify these errors. This means that when real data is used the results will have to be inspected visually to identify phase errors and numerical diffusion, or methods will have to be developed. Moreover, it might not be clear whether phase errors or non-realistic smoothening are due to modelling inaccuracies or numerical approximation errors. In the field of computational fluid dynamics many methods are known to measure accuracy of data as well as of mathematical models and numerical methods (*14*). The newly introduced accuracy measures can also be applied to calibrate and evaluate traffic flow models. However, this is beyond the scope of this paper.

5 EXPERIMENTAL SETUP

We implement the IMEX method and the implicit method with Newton linearization and compare them with the explicit method on accuracy of their solution and computing times. In this paper we concentrate on comparison of the numerical time stepping methods by implementing them for two simple test problems on a linear link. A third problem is added to verify the results in congested conditions.

For illustration purposes we use a Greenshields fundamental relation (*15*, Section 7.1),

$$q(\rho) = \rho\, v_{\max} \left(1 - \frac{\rho}{\rho_{\max}}\right), \text{ for } 0 \leq \rho \leq \rho_{\max}. \tag{16}$$

In our test problems we use the maximum velocity $v_{\max} = 100$ km/h and critical density $\rho_{\text{critical}} = 25$ vehicles/km, resulting in a maximum flow $q_{\max} = 1250$ veh/km.

For the space discretization we use the minimum supply demand scheme (1)–(2). The spatial grid size Δx of the test problems is 200 meter. Using the CFL-condition (12) we find a maximum time step size of $\Delta t_{\max} = \frac{\Delta x}{v_{\max}} = 7.2$ seconds for the explicit method to be stable (*12*, Chapter 1). In our tests we use time step sizes of 5, 30 and 180 seconds.

The test problems only vary in their initial condition (see Figure 5). With these initial conditions we have free flow and and acceleration fan in test problem 1, free flow and a shock wave in test problem 2 and congestion and a deceleration fan in test problem 3.

We computed the solutions for the first 30 minutes over a spatial domain of 200 kilometer: 100 kilometer before the location of the jump in the initial condition and 100 kilometer after it. For comparison purposes we also run simulations with a simulated time of 12 minutes.

In our current implementation we did not apply implicit time schemes on problems in which a transition from free flow to congestion or vice versa occurs. This transition gives rise to some difficulties in implementing implicit schemes. This is due to the non-linearity that is created when the minimum supply demand scheme (1–2) is applied for spatial discretization.

We will explain this using an example with Greenshields fundamental relation (16) on a homogeneous road. Assume that there is congestion at the $i+1$-th grid cell. Therefore, the supply of this cell is $s_{i+1} = q_{i+1}$. In (the approximation of) the Jacobian we need $\frac{dq_{i\to i+1}}{d\rho_i}$. If $\rho_i = \rho_{\max} - \rho_{i+1}$ there is no congestion in the i-th cell and the demand is $d_i = q_i$. With the minimum supply demand

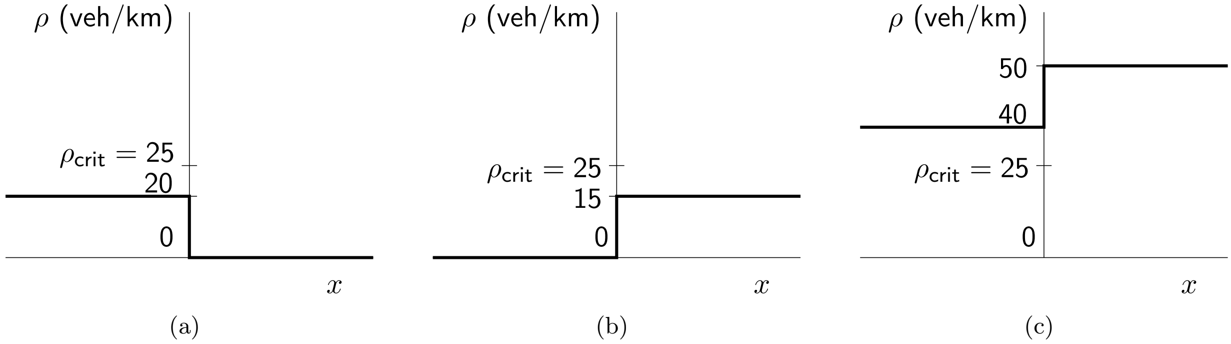


FIGURE 5 Initial conditions for the test problems. (a) Test problem 1: acceleration fan. (b) Test problem 2: shock wave. (c) Test problem 3: deceleration fan.

scheme we find $q_{i\to i+1} = q_i = q_{i+1}$. Now we have

$$\frac{dq_{i\to i+1}}{d\rho_i} = \begin{cases} \frac{dq_i}{d\rho_i} & = 1 - 2\rho_i & \text{if } \rho_i \uparrow \rho_{\max} - \rho_{i+1}, \\ \frac{dq_{i+1}}{d\rho_i} & = 0 & \text{if } \rho_i \downarrow \rho_{\max} - \rho_{i+1}. \end{cases} \tag{17}$$

Clearly $\frac{dq_{i\to i+1}}{d\rho_i}$ is discontinuous around $\rho_i = \rho_{\max} - \rho_{i+1}$ and the Jacobian can not be approximated accurately enough. Note that the simulation of congestion itself does not give difficulties: only the transition from uncongested to congested state (and vice versa) is complex.

We can deal with this problem in three distinct ways. First we can use an alternative spatial discretization. This new spatial discretization method should both introduce only small approximation errors and should not introduce non-linearities which are difficult to deal with in the time disretization. Second we can use alternative linearization methods to improve or replace Newton linearization (*11, 16, 12, 17*). Third we can apply shock wave theory to predict the locations where difficulties might arrise and use an other, more appropriate, method only at this location.

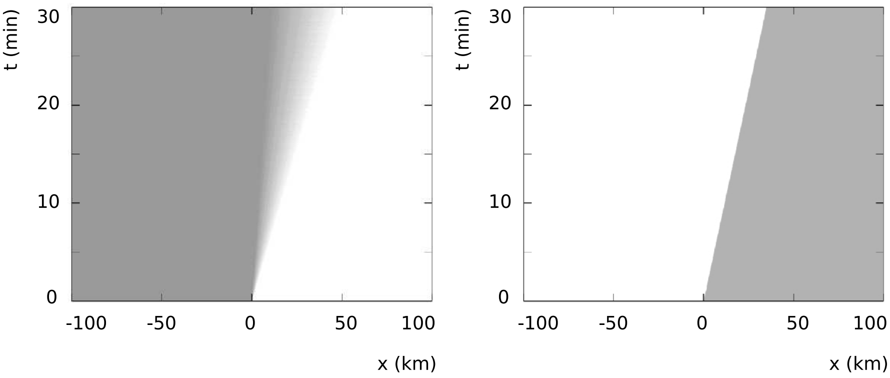
We expect that similar problems with non-linearities will occur when we apply other, more realistic, fundamental diagrams. This will for example happen when the fundamental relationship is not continuously differentiable. Furthermore, Chronopulos et al (*10*) have indicated that implicit methods can be applied on so called higher order methods as well.

The test problems and the solution methods as described above and in Section 2 have been implemented using Octave, a high-level language for numerical computations. The results of the computations are described in the next section.

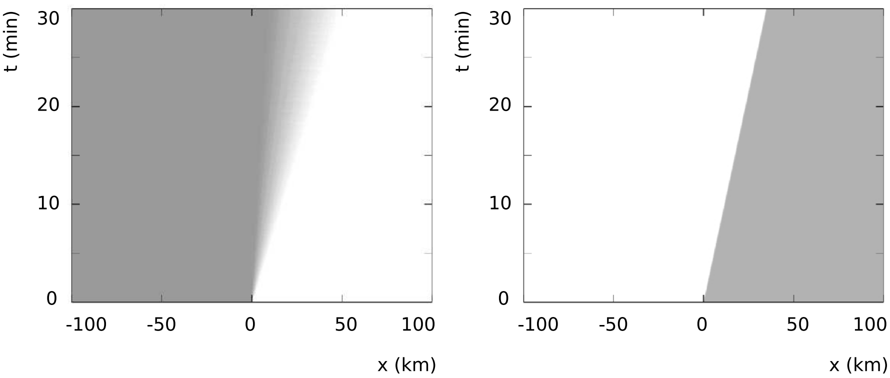
6 RESULTS AND DISCUSSION

The analytical solutions of the test problems and some computed solutions are shown in Figure 6. We show the numerical solutions with the explicit method and a small time step ($\Delta t = 5$ seconds), for the other two methods the solution with a large time step ($\Delta t = 30$ seconds) is shown. The results with respect to accuracy are summarized in Figure 7 and Table 1, in which also the total computational times are shown.

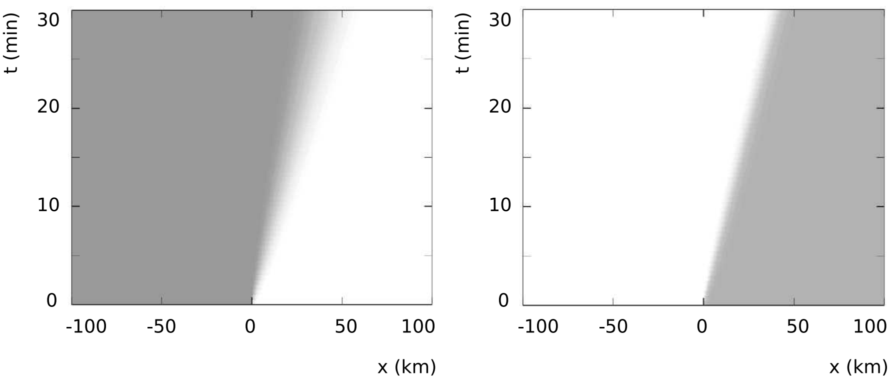
Accuracy In Figure 7 the RMSE of the densities is plotted against the computed time. With Newton linearization the RMSE is constant over time, however, it grows for the other two methods. This can also be seen from Table 1 when the results after 12 minutes are compared with those after



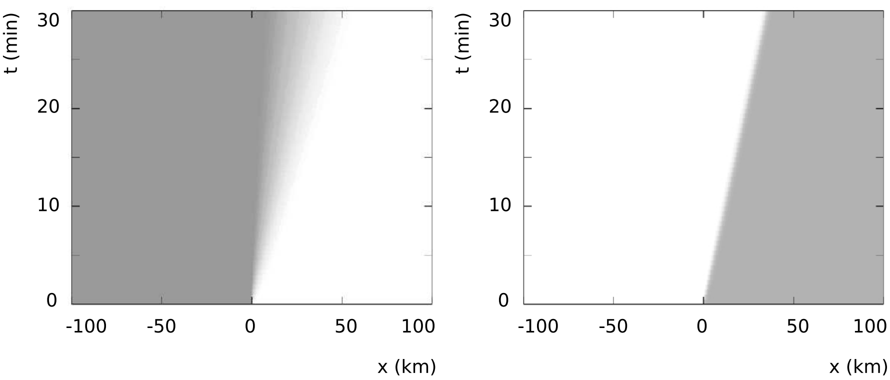
Analytical solution



Explicit method, $\Delta t = 5$ seconds



IMEX method, $\Delta t = 30$ seconds



Newton's method, $\Delta t = 30$ seconds

FIGURE 6 Exact and computed densities for test problem 1 (left) and 2 (right).

TABLE 1 Numerical accuracy and computing time

Method	Δt (s)	RMSE		Phase error (km/h)		Num Diffusion) (km/h)		CPU-time (s)
		12 min	30 min	12 min	30 min	12 min	30 min	
test problem 1 (acceleration fan)								
Explicit	5	0.005	0.004	-1	-0.4	8	4.0	1.728
	30	0.162	0.253	*	*	*	*	*
	180	0.465	0.682	*	*	*	*	*
IMEX	5	0.031	0.050	20	20.0	-8	-20.4	2.028
	30	0.029	0.048	21	20.4	17	-5.2	0.112
	180	0.031	0.046	20	21.2	99	42.0	0.016
Newton	5	0.009	0.009	-3	-1.2	19	8.8	2.460
	30	0.017	0.016	-5	-2.8	33	18.0	0.200
	180	0.042	0.040	-15	-8.0	88	50.0	0.028
test problem 2 (shock)								
Explicit	5	0.008	0.008	1	0.4	4	1.6	1.700
	30	0.072	0.110	*	*	*	*	*
	180	0.104	0.145	*	*	*	*	*
IMEX	5	0.028	0.050	13	13.6	22	10.0	1.956
	30	0.025	0.041	11	12.0	46	24.0	0.116
	180	0.043	0.047	2	7.6	114	68.8	0.016
Newton	5	0.012	0.012	1	0.4	10	4.0	2.460
	30	0.018	0.019	1	0.4	24	10.0	0.196
	180	0.041	0.040	-2	0.0	65	36.0	0.028
test problem 3 (congestion)								
Explicit	5	0.006	0.006	1	0.4	6	2.4	2.120
	30	0.053	0.081	*	*	*	*	*
	180	0.074	0.103	*	*	*	*	*
Newton	5	0.010	0.010	0	0.0	17	7.2	3.076
	30	0.015	0.016	0	0.0	38	18.4	0.284
	180	0.030	0.031	4	0.8	58	58.0	0.044

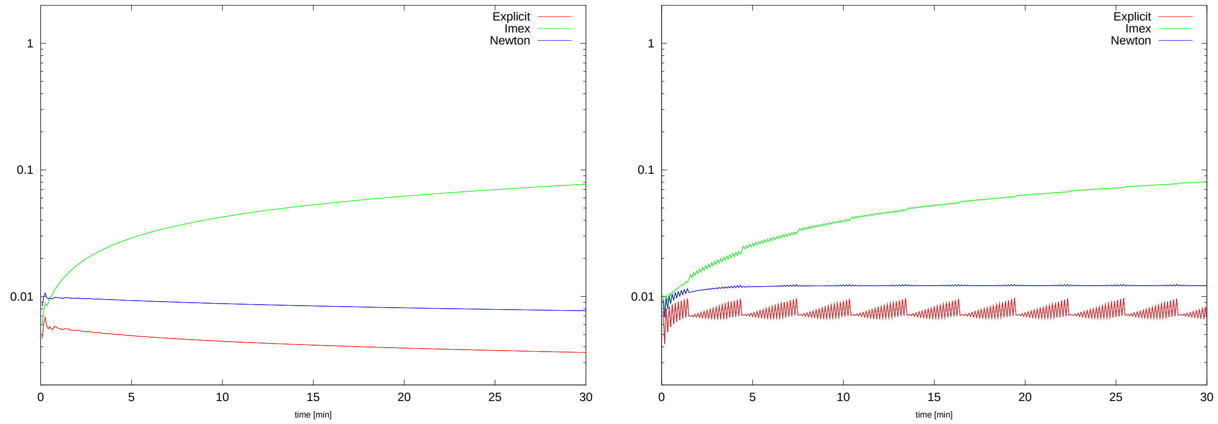
* Can not be determined because of instability.

30 minutes. In this table we furthermore see that the phase error is constant over time for the IMEX method. With the implicit method and Newton linearization, however, both the phase error and the numerical diffusion decrease over time. This indicates that Newton linearization is especially well suited for simulations over a longer time.

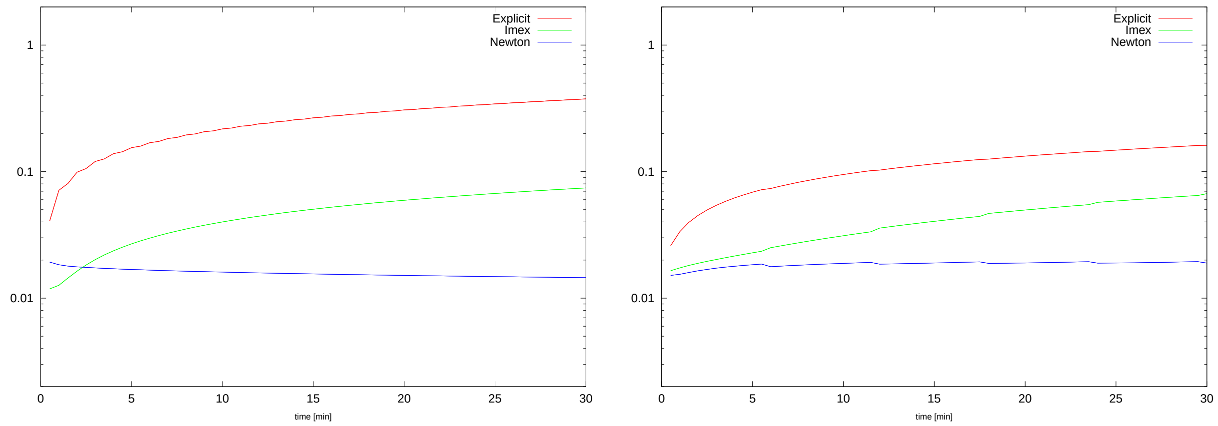
With the explicit method the errors are very large for large time steps ($\Delta t = 30$ or 180 seconds), because of the instability of the method. For the same reason, we can not determine the phase error and the numerical diffusion for the explicit method with large time steps.

The IMEX method leads to both phase errors and numerical diffusion. Note that the RMSE is similar for all cases, while the phase error and the numerical diffusion are not. This shows that the RMSE is not a proper accuracy measure if one wants to know the nature of the error. The phase error introduced by the implicit scheme with Newton linearization is small. However, here we see much numerical diffusion, especially for large time steps.

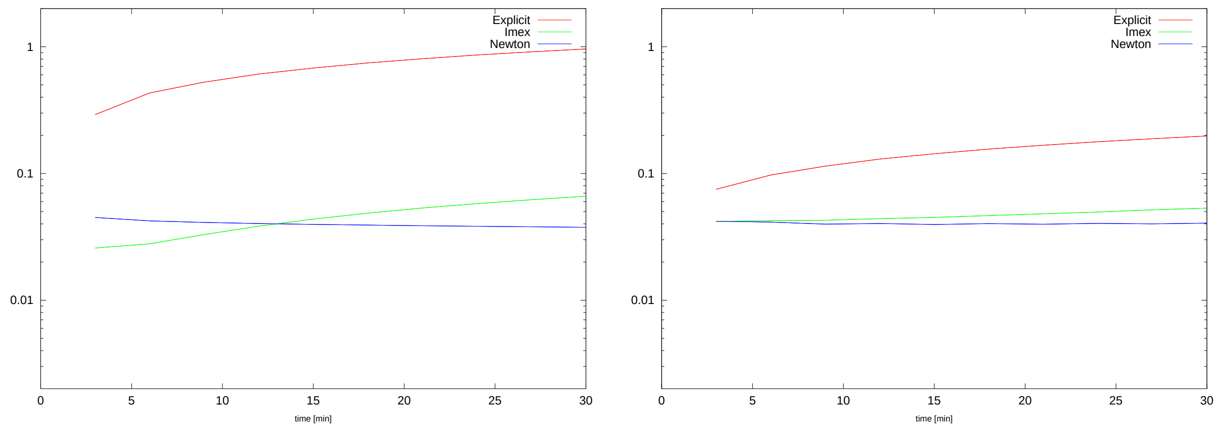
In both the IMEX method and the implicit method with Newton linearization we see that



$\Delta t = 5$ seconds



$\Delta t = 30$ seconds



$\Delta t = 180$ seconds

FIGURE 7 The RMSE plotted against the simulated time for testproblem 1 (left) and testproblem 2 (right). The oscillations in the upper right graph are a result of discretisation error.

the phase error does not depend greatly on the time step size. This means that phase errors have to be considered for any time step size. The numerical diffusion, however, is larger for larger time step sizes.

Computational efficiency In Table 1 it is shown that the computational efficiency can be improved greatly by using the IMEX method or the implicit method with Newton linearization. Only when the time step is small ($\Delta t = 5$ seconds) the explicit method is a little more efficient. For medium and large time steps the other methods are more efficient; 9 to 15 times faster with $\Delta t = 30$ and 60 to 100 times faster with $\Delta t = 180$. These results are similar for all test problems.

The IMEX method is slightly more efficient than the implicit method with Newton linearization. However, the accuracy is a little worse for the IMEX method, as discussed above. This is partly due to the explicit part of the IMEX method: the velocity at the old time step in the discretization of the space derivative of the flow. Note that the computational efficiency with large time steps is very high, but this results in much lower accuracies. Therefore, medium-sized time steps are prefered.

7 CONCLUSION AND FURTHER RESEARCH

The above results show that implicit time integration for macroscopic traffic flow models is very promising when compared to well-known explicit time integration. This is because larger time steps are possible, which offers the opportunity to use less time steps in the simulation. We have shown that a factor 9 to 15 can be achieved in some cases. Even higher computational efficiency can be achieved, but this is at the expense of lower accuracy. This computational efficiency plays a key role in the application of real time traffic flow modelling.

We have introduced two new methods to measure the accuracy of numerical methods applied on traffic flow models. These, or similar, methods can also be used to evaluate the mathematical model itself. We recommend further study of this application of the phase error and numerical diffusion as accuracy measures.

We have analyzed our results on accuracy using the RMSE and measures for the phase error and numerical diffusion. We found that implicit time schemes introduce numerical diffusion. Furthermore, the IMEX method introduces phase errors. The numerical diffusion can be reduced by using smaller time steps, but this results in larger computing times. Therefore we suggest further research into methods for reducing numerical diffusion such as Lax-Wendroff methods for spatial discretization (*16*).

In future research we will further analyze the optimal implementation of an implicit scheme, both with respect to accuracy and computing time. We will do this for a broader range of problems, including 'higher-order' models, and cases in which a transition from free flow to congestion and vice versa occurs.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their useful comments. Femke van Wageningen-Kessels is a PhD-student at the ITS Edulab, a joint collaboration between TUDelft and Rijkswaterstaat. This research is part of the project FastLane Online, partly carried out within the TRAIL research school.

References

[1] M.J. Lighthill and G.B. Witham. On kinematic waves I: Flood movement in long rivers *and* On kinematic waves II: A theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London series A, Mathematical and Physical Sciences*, volume 229, pages 281–345, London, May 1955. JSTOR.

[2] P.I. Richards. Shockwaves on the highway. *Operations Research; The journal of the Operations Research Society of America*, 4:42–51, 1956.

[3] S.P. Hoogendoorn and P.H.L. Bovy. State-of-the-art of vehicular traffic flow modeling. In *Proceedings Institution of Mechanical Engineers*, 215, pages 283–303, 2001.

[4] J.P. Lebacque. The Godunov scheme and what it means for first order traffic flow models. In J.P. Lesort, editor, *Transportation and traffic theory, Proceedings of the 13th international symposium on transportation and traffic theory*, pages 647–677. Pergamon, July 1996.

[5] J.W.C. van Lint, S.P. Hoogendoorn, and M. Schreuder. Fastlane; A new multi-class first order traffic flow model. In *TRB 87th Annual Meeting Compendium of Papers DVD*, 2007.

[6] D. Helbing and M. Treiber. Numerical simulation of macroscopic traffic equations. *Computing in Science & Engineering*, 1(5):89–98, September-October 1999.

[7] P.S. Babcock, D.M. Auslander, M. Tomizuka, and A.D. May. Role of adaptive discretization in a freeway simulation model. *Transportation Research Record*, 971:80–92, 1984.

[8] H. Cho, M. Hwang, and T. Jou. A finite difference method with adaptive time mesh for hyperbolic traffic flow. In *The IEEE 5-th international conference on intelligent transportation systems*, pages 779–784, Singapore, 2002.

[9] A. Chronopoulos, P Michapoulos, and J. Donohoe. Efficient traffic flow simulation computations. *Mathematical and Computer Modelling*, 16(5):107–120, 1992.

[10] A. Chronopoulos, A Lyrintzis, P Michapoulos, C. Rhee, and P. Yi. Traffic flow simulation through high order traffic modelling. *Mathematical and Computer Modelling*, 17(8):11–22, 1993.

[11] R.J. LeVeque. *Numerical methods for Conservation laws.* Birkhäuser, Basel, 1992.

[12] W. Hundsdorfer and J. Verwer. *Numerical solution of time-dependent advection- diffusion-reaction equations.* Springer series in computational mathematics. Springer, Berlin, 2003.

[13] R.L. Burden and J.D. Faires. *Numerical Analysis.* Brooks/Cole, Pacific Gove, CA, seventh edition, 2001.

[14] P. Wesseling. *Principles of computational fluid dynamics*, volume 29 of *Springer series in computational mathematics.* Springer-Verlag, Berlin, 2001.

[15] A.D. May. *Traffic flow fundamentals.* Prentice Hall, New Jersey, 1990.

[16] R.J. LeVeque. *Finite volume methods for hyperbolic problems.* Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, 2002.

[17] Kelley. *Iterative Methods for Linear and Nonlinear Equations.* Number 16 in Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1995.