High-speed microsimulations of traffic flow

Inaugural-Dissertation
zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Kai Nagel
aus Köln

1995

Berichterstatter: Prof. Dr. A. Bachem
Priv.-Doz. Dr. M. Schreckenberg

Tag der letzten mündlichen Prüfung: 12. Dez. 1994

Contents

Chapter 1

Introduction

Traffic jams are annoying, cost money, and pollute the environment. Meanwhile, traffic jams are only the most visible part of a far bigger problem: In the western part of Germany[1] alone, people travel 748.3 billion km per year (pkm = passenger kilometers), and goods are moved 266 billion tkm (ton kilometers) per year [171]. This is more than 2 000 000 000 pkm and 725 000 000 tkm per day.

Moreover, 13.7% of the Gross National Product of Germany is absorbed by trade, transportation, and telecommunication [57]; in the U.S. it is 14.8% by transportation alone [39]. In Germany, nearly 10% of the federal budget is spent by the department of transportation (1991/92, [57]). Thus, transportation is a huge part of the economies of industrialized countries.

The traditional answer of transportation planners and politicians to bottlenecks has been to increase the transportation supply. Yet, demand has always caught up, and during the last years it became increasingly clear that there are big problems in transportation which cannot be solved by the traditional policies.

For example in Germany, traffic produces 67.9% of all carbon monoxides, 58.4% of all NO_x, and 18.2% of all carbon dioxides [171], and it consumes 28% of all energy [57]. Car traffic is a noisy and dangerous ingredient of urban life. Airports, rail lines for high speed trains, and man-made water ways destroy the last undisturbed natural areas in the industrialized parts of Europe. Airplanes inject water vapor and pollution directly into the stratosphere.

An intuitive reaction is that there should be ways to do this without putting so much stress on the environment. And indeed, there is a large number of propositions, ranging from modern technology to changes in social habits. Examples are among others:

- fuel-saving engines,
- Advanced Traffic Information/Management Systems (ATIS/ATMS) using Advanced Transport Telematics (ATT) or the American counterpart Intelligent Vehicle Highway Systems (IVHS),
- routing optimization,
- freight distribution centers, integrated freight logistics,
- transit systems (e.g. trains),
- reducing freight transportation by making industrial production more local

---

[1] Statistical numbers in good quality are not yet available for the new parts of Germany.

- tele-commuting (i.e. working from a computer terminal at home)
- long-term traffic-reducing urban planning.

In the United States, the Clean Air Act will force communities to find policies which keep the air quality better than a given standard.

However, because of the high complexity of the problem, singular actions are often not very effective and sometimes even counter-productive. For example, addition of a new road may lead to a *de*crease of the vehicle throughput [24]. Simulations show that the introduction of a transit system may lead to *in*creased pollution [165] (private vehicles remain at home and are used by another household member for a high number of short trips with a cold engine). More than half of all passenger kilometers are not work related: In Germany, 63.2% of all pkm fall into the categories shopping (9.2%), leisure (45.3%), and vacation (8.7%) [171].

These examples should have made clear that there are no simple solutions. It will be difficult to reduce the stress on the environment when the transportation demand keeps on increasing at current rates; and reducing the transportation demand cannot be achieved only by technical solutions in freight transportation but will have to include changes in social habits. Obviously, this will need thorough discussion for finding a widely accepted balance of interests.

One of the best methodologies today to deal with complex systems like the transportation system is simulation. Fast desktop computers allow people to get a “hands-on” experience with different possible scenarios, and (parallel) supercomputers give the necessary power to deal with big problems.

As a first step, one would like to simulate “everything” which is related to transportation—from settlement and infrastructure over social habits over trip planning and execution to air pollution dynamics. It seems necessary to include all these elements since, for example, changes in the settlement structure will affect air pollution in non-foreseeable ways.

This is in fact the approach of the American TRANSIMS project at the Los Alamos National Laboratory [165]; the European project EUROTOPP [153] uses similar ideas. Large interdisciplinary groups are necessary to integrate all the different aspects of such an approach. The research presented here is part of a collaboration between the University of Cologne and the TRANSIMS project. It is planned to work with more than 30 people during the next 5 years on the project of transportation simulation, analysis, and optimization.

In this collaboration, the Cologne group concentrates so far on road traffic, with special focus on theoretical developments, on computational aspects of simulation, and on algorithmical aspects. Road traffic currently contributes more than 81% of all passenger and 52.7% of all freight transportation [171]. And despite widespread efforts, the share of road transportation is still increasing. In this sense, it makes sense to start with road traffic when dealing with transportation systems.

This work presents results of traffic simulations using very fast microscopic models. “Micro” refers to the fact that each individual vehicle is resolved, which is an inevitable requirement for, e.g., individual routing analysis. High computational speed, also on (parallel) supercomputers, is achieved by keeping the model extremely simple. This makes, on one hand, real time simulations of large road networks possible, and it allows, on the other hand, the use of dynamical systems techniques in conjunction with many repeated Monte Carlo runs to characterize the phase space of the transportation system. Last, but not least, the simplicity allows theoretical work: Critical exponents clarify the connection between fluid-dynamical and elaborate car-following models, and analytical methods help classifying the general properties of traffic flow models.

This text has two main parts: traffic flow, and network traffic. A third part contains computational results, an outlook, and the conclusion.

The first part consists of Chapters 2 to 7. Chapter 2 explains some necessary background for the traffic flow investigations. It starts with an overview of theoretical models for one-dimensional transport and for traffic as found in the literature. It is, to my knowledge, the first overview which, by using well-known results from the area of driven diffusive systems, classifies fluid-dynamical theories and particle hopping models for traffic flow into one common framework. The second part of this chapter reviews traffic flow simulation concepts and models, and argues why a fast microscopic approach is necessary. Chapter 3 presents the basic model, and gives first results on its properties. Chapters 4 and 5 discuss two different limits of the model: In the first case, all randomness is set to zero; in the second case, this is only done for the part of the randomness which leads to spontaneous initiation of traffic jams. Chapter 6 then summarizes further results and explains them in the light of the results from Chapters 4 and 5. Chapter 7 summarizes all traffic flow results, and shows how the traffic model investigated here helps closing the gap between fluid-dynamical theories, car following behavior, and other, simpler, particle-hopping models.

Chapters 8 to 10 describe traffic in networks. Chapter 8 discusses basic issues, Chapters 9 and 10 contain network examples, where road network elements such as nodes and edges can be processed and vehicles follow individual travel plans. Chapter 9 demonstrates, by using a simple example, how questions such as efficient adaptive road pricing, can be treated by using this fast microsimulation, and that these questions cannot be separated from the system dynamics. Quite generally, it turns out that high system performance is intrinsically coupled to high fluctuations and therefore low predictability. The second example (Chapter 10) is a (single lane) network implementation of the freeway network of Northrhine-Westfalia, showing that this approach makes algorithmic investigations using realistic individual travel plans possible in much faster than real time.

Chapter 11 tests computational performance by implementing the basic model on many different supercomputers. These results are compared to performance values from other projects, and it is checked how performance decreases when implementations become more realistic, e.g. when programming freeway networks. Chapter 12 is an outlook, describing indications for further work, or extensions of the model which have already been investigated by others. Chapter 13 shortly summarizes and concludes the text.

Several appendices contain, for completeness, technical details, or work which is important for the overall picture but less important for the main arguments pursued in this text.

Except for Chapter 3, it should be possible to read only parts of this text without losing too much understanding. People with a background in driven diffusive systems may want to start with the summary in Chapter 7 and just skip back when they need additional information; and the network Chapters 8 to 10 build on, but are largely independent of, the rest of the text.

Part A

TRAFFIC FLOW

Chapter 2

Background

Especially when demand exceeds supply, it makes sense to use existing transportation facilities as efficiently as possible. Naively, one would want to keep, say, a freeway in the regime of maximum vehicle throughput (= capacity). However, due to a traditional bias as well as due to a lack of tools, transportation science mostly has concentrated on traffic at densities *far below* maximum throughput—as can for example be seen from the fact that the Highway Capacity Manual [63] distinguishes five different regimes (Levels of Service A to E) for traffic at densities below capacity, but only one (Level of Service F) above.

Most of currently used traffic flow theory has been developed in the 1950's, see, e.g., [47] for a review. Steady state solutions for mathematical car-following models correspond to steady state throughput-density curves, and due to the lack of refined technology this standard was sufficient to explain measurements. Jam-waves could be explained by using the analogy between traffic and kinematic waves in, e.g., rivers for a fluid-dynamical treatment [100, 175].

However, recent and refined measurements [2, 56, 69] point out that especially the regime of maximum throughput is far from being completely understood. At the same time, extensions of the fluid-dynamical treatment using new methods from nonlinear dynamics indicate a phase transition (route to chaos) in the equations [88], consistent with theoretical developments based on measurements [1, 132]. Simulations of these equations allow for the first time systematic investigation of large scale traffic flow phenomena near capacity [80].

Nonetheless, the fluid-dynamical description is not really useful to clarify the connection between the macroscopic, emergent phenomena (e.g. jams) and the microscopic dynamics (e.g. driving behavior). In addition, simulations based on fluid-dynamical equations are known to be problematic in complex geometries [145]. In the context of traffic this would mean that each ramp or each change in the road characteristics (speed limit, number of lanes, slope, ...) could trigger spurious disturbances which stem from the discretization of the equations.[1] In fluid-dynamics, simple particle-based models (lattice gas automata, LGA) [44] are used to complement the partial differential equation (PDE) approach. LGA have proven to be extremely useful in complex geometries (e.g. for porous media [145]), where they avoid the instabilities of the discretized PDEs.

---

[1] The fact that this does not seem to happen in network implementations based on the Lighthill-Whitham theory is most probably a consequence of the fact that, together with numerical dissipation due to the discretization, the Lighthill-Whitham approach is stable for all frequencies. This is no longer true for more realistic fluid-dynamic models, which, though, have not yet been tried for road networks.

For vehicular traffic with its much higher particle granularity, an LGA approach is even more natural than for fluids. Instead of with "First Principles" as in LGA, one starts with behavioral rules, and in the same way as in LGA, one can attempt to find the hydrodynamic limit. In this way, CA traffic models are not only useful for computational applications, but also for theoretical developments (theory of traffic flow), directly placed between extensive microscopic and averaging fluid-dynamical models.

The first section of this chapter will be devoted to fluid-dynamical models for traffic. Then, particle based models for one-dimensional transport and their relations to the fluid-dynamical models will be reviewed. Section 3 treats models for car-following, and Section 4 reviews simulation models for traffic flow.

Section 1 is somewhat more comprehensive because it organizes results found in the traffic science literature and results found in the physics literature into one common framework. To my knowledge, such a review does not exist elsewhere.

2.1 Fluid-dynamical models for traffic flow

2.1.1 General equations

One might use the standard fluid-dynamical conservation equations for mass and momentum as a starting point for a fluid-dynamical description of traffic:

$$\partial_t \rho + \partial_x(\rho\, v) = 0$$

and

$$\frac{\mathrm{d}v}{\mathrm{d}t} \equiv \partial_t v + v \cdot \partial_x v = F/m \ ,$$

where ρ is the density and v the velocity.[2] $\mathrm{d}/\mathrm{d}t$ is the individual derivative, F is the force acting on mass m. The first equation (of continuity) describes mass conservation; the second one (momentum conservation) describes the fact that the momentum of a point of mass may only be changed by a force. Obviously, for traffic, F has to take care of vehicle and driving dynamics.

2.1.2 Fluctuations

A standard first step in fluid-dynamics [25] is to assume that v and ρ fluctuate statistically around average values $\langle\, v \,\rangle$ and $\langle\, \rho \,\rangle$, i.e.

$$v = \langle\, v \,\rangle + v' \ , \qquad \langle\, v' \,\rangle = 0$$

and

$$\rho = \langle\, \rho \,\rangle + \rho' \ , \qquad \langle\, \rho' \,\rangle = 0 \ .$$

In this case, one only assumes that $\langle\, v \,\rangle$ and $\langle\, \rho \,\rangle$ fluctuate *slowly* in space and time; for the general subtleties of hydro-dynamical theory see, e.g., [90].

Inserting these relations and subsequent averaging over the whole equations (e.g. $\langle\, \partial_x[(\langle\, \rho \,\rangle + \rho')\,(\langle\, v \,\rangle + v')]\,\rangle = \partial_x \langle\, \rho \,\rangle \langle\, v \,\rangle + \partial_x \langle\, \rho' v' \,\rangle$) yields

$$\partial_t \langle\, \rho \,\rangle + \partial_x \langle\, \rho \,\rangle \langle\, v \,\rangle + \partial_x \langle\, \rho' v' \,\rangle = 0$$

---

[2] The traffic literature often uses k instead of ρ.

and

$$\partial_t\langle\, v\,\rangle + \langle\, v\,\rangle\partial_x\langle\, v\,\rangle + \frac{1}{2}\partial_x\langle\, v'v'\,\rangle = \langle\, F/m\,\rangle\ .$$

One can [25] parametrize an averaged fluctuation by the corresponding gradient

$$\langle\, v'A'\,\rangle \approx -\alpha\,\partial_x\langle\, A\,\rangle\ ,$$

which leads to the set of equations

$$\boxed{\begin{aligned}\partial_t\rho + \partial_x(\rho\, v) &= D\,\partial_x^2\rho\\ \partial_t v + v\,\partial_x v &= F/m + \nu\,\partial_x^2 v\ ,\end{aligned}}$$

where, according to convention, the averaging brackets has been omitted, and the diffusion coefficient D as well as the (kinematic) viscosity ν are assumed to be independent of x and t. It should be noted that similar diffusion terms can also be obtained from other arguments.

2.1.3 Lighthill-Whitham-theory and kinematic waves

If one assumes that the velocity is a function of density only ($v = f(\rho)$), then the momentum equation is no longer necessary. This corresponds to instantaneous adaption; the particles (or cars) carry no memory. Using without loss of generality the current $q(\rho) \equiv \rho\, v(\rho)$, and setting in addition $D = 0$, one obtains

$$\boxed{\partial_t\rho + q'(\rho)\,\partial_x\rho = 0}$$

(*Lighthill-Whitham-equation*) [100, 53], where $q' = \mathrm{d}q/\mathrm{d}\rho$.[3] q' will always mean the derivative of q, even when the prime in connection with other letters often denotes fluctuations.

The equation can be solved by the ansatz $\rho(x,t) = \rho(x - ct)$ with

$$c = q'(\rho)\ .$$

This allows the solution of the *characteristics*: A region with density ρ travels with constant velocity $c = q'(\rho)$, and the resulting straight line in space-time is called characteristic. When $q(\rho)$ is convex, i.e. $q'' < 0$, then for regions of decreasing density ($\rho(x_1) > \rho(x_2)$ for $x_1 < x_2$) the characterstics separate from each other. On the other hand, in regions of *in*creasing density, the characteristics come closer and closer together and ultimately hit each other. When two characteristics hit each other, a density discontinuity appears at this place (a *front*), which moves with velocity

$$c = \frac{q(x_2) - q(x_1)}{\rho(x_2) - \rho(x_1)} = \frac{\Delta q}{\Delta\rho}\ .$$

An illustrative example is a queue, such as at a red light. When the light turns green, the outflow front quickly smoothes out, whereas the inflow front remains steep.

Leibig [97] gives results how a random initial distribution of density steps in a closed system evolves towards two single steps according to the Lighthill-Whitham-theory.

---

[3] In this text, *current* is the same as *flow* or *throughput*. The physics literature usually uses j or J instead of q.

2.1.4 Lighthill-Whitham with dissipation

Adding dissipation to the Lighthill-Whitham-equation leads to

$$\boxed{\partial_t \rho + q'(\rho)\, \partial_x \rho = D\, \partial_x^2 \rho\ .}$$

The solution of this equation is again a nondispersive wave with phase and group velocity q', but this time with a damping constant D. One way to see this is by linearizing the equation: A stationary and homogenous density $\rho \equiv const =: \rho_0$ is a solution. The ansatz $\rho = \rho_0 + \epsilon\, \rho_1 + \epsilon^2\, \rho_2 + \ldots$ leads in first order in ϵ to

$$\partial_t \rho_1 + q'(\rho_0)\, \partial_x \rho_1 = D\, \partial_x^2 \rho_1\ .$$

A Fourier-Transform (or using the ansatz $\rho_1 = \hat{\rho}_1 \exp[i(kx - \omega t)]$) yields

$$\omega = q'(\rho_0)\, k - i\, D\, k^2\ .$$

From the real part of the frequency one obtains the phase velocity c_φ and the group velocity c_g:

$$c_\varphi := \frac{\mathrm{Re}(\omega)}{k} = c_g := \frac{\partial}{\partial k} \mathrm{Re}(\omega) = q'(\rho_0)\ ,$$

which gives are the same kinematic waves as before. The difference is that this time a dissipation (damping) of the wave amplitude is given by $\mathrm{Im}(\omega) = -D\, k^2$: Any disturbance decays exponentially. The reflects the intuitively reasonable effect that traffic jams should tend to dissolve under homogeneous and stationary conditions.

2.1.5 The nonlinear diffusion (Burgers) equation

For a further development, $q(\rho)$ has to be specified. Since we are mostly interested in the behavior of traffic near maximum throughput, we start by choosing the simplest mathematical form which yields a "well-behaved" maximum:

$$q(\rho) = v_{max}\, \rho(1 - \rho)\ ,$$

which, in traffic science, is called the *Greenshields*-model (see [47]). v_{max} is, in principle, a free parameter, but it has an interpretation as the maximum average velocity for $\rho \to 0$.[4] The maximum current q_{max} is reached at $\rho(q_{max}) = 1/2$.

The result is the equation

$$\boxed{\partial_t \rho + v_{max}\, \partial_x \rho - 2 v_{max}\, \rho\, \partial_x \rho = D\, \partial_x^2 \rho\ .}$$

Introducing a linear transformation of variables [115]

$$x = v_{max}\, t' - x'\ , \qquad t = t'\ ,$$

one obtains

$$\partial_{t'} \rho + 2\, v_{max}\, \rho\, \partial_{x'} \rho = D\, \partial_{x'}^2 \rho\ ,$$

---

[4] Mathematicians would set $v_{max} = 1$; traffic scientists use $1 - \rho/\rho_{jam}$ for the term in parenthesis. ρ_{jam} is the density of vehicles in a jam.

which is the (deterministic) *Burgers equation.*

This equation has been investigated in great detail by Burgers [20] as the simplest non-linear diffusion equation. The stationary solution is a uniform density $\rho(x,t) = const.$ A single disturbance from this state evolves over time into a characteristic triangular structure with height $\sim t^{-1/2}$, width $\sim t^{1/2}$, and bent to the right such that the right side of the disturbance becomes discontinuous. This disturbance moves to the right with velocity $c = q' = 2\,\rho\, v_{max}$.

When interpreting this for traffic jams, one has to re-transform the coordinates. Jams can then move *both* to the left or to the right (with velocities between v_{max} and $-v_{max}$), and the discontinuous front develops at the upstream side of the jam, i.e. where the vehicles enter the jam. One sees that this solution is just the solution of the characteristics, with a dissipating diffusion term added—as should be expected because of $D > 0$.

Some other versions of the Burgers equation are relevant for traffic and have been investigated thoroughly [85, 18, 105]:

- **Noisy Burgers equation:** Adding a Gaussian noise term η to the equation ($\langle\, \eta(x,t)\eta(x',t')\,\rangle = \eta_0\, \delta(x-x')\, \delta(t-t')$) leads to the noisy Burgers equation

  $$\partial_t \rho + 2\, v_{max}\, \rho \partial_x \rho = D \partial_x^2 \rho + \eta\ .$$

  This equation does not reach a stationary state any more.

- **Generalized Burgers equation:** The nonlinearity of the Burgers equation can be generalized:

  $$\partial_t \rho = \sum_\beta b_\beta \partial_x \rho^\beta + D \partial_x^2 \rho\ .$$

  E.g., a current of

  $$q(\rho) = q_{max} - v_{max}\, |\rho - \rho(q_{max})|$$

  leads to a linear Burgers equation

  $$\partial_t \rho + \mathrm{sgn}[\rho(q_{max}) - \rho]\, v_{max}\, \partial_x \rho = D\, \partial_x^2 \rho\ .$$

  As we will see later, this linear ansatz is useful for traffic.

  Generalized Burgers equations with arbitrary β have been investigated [18, 85].

The Burgers equation has the same mathematical structure as a well-known equation for surface growth, the Kardar-Parisi-Zhang (KPZ) equation [18, 78, 85]. The connection is most straightforward when one interprets the *fluctuations* of the density, $\rho' = \rho - \langle\, \rho\,\rangle_L$, as a step density for a 1-dimensional surface (i.e. $\rho' = \partial_x h$). Inserting $\rho = \langle\, \rho\,\rangle_L + \rho'$ into the generalized Burgers equation yields

$$\partial_t \rho' = \sum_\beta \tilde{b}_\beta \partial_x \rho'^\beta + D\, \partial_x^2 \rho'\ ,$$

with the same structure as before, but with different coefficients $\tilde{b}_\beta$. The system here is assumed to be closed (periodic boundary conditions), in consequence $\partial_t \langle\, \rho\,\rangle_L \equiv \mathrm{d}\langle\, \rho\,\rangle_L/\mathrm{d}t = 0$. By inserting $\rho' = \partial_x h$ and integrating once, one obtains

$$\partial_t h = const + \sum_\beta \tilde{b}_\beta\, (\partial_x h)^\beta + D\, \partial_x^2 h$$

(generalized *KPZ-equation*).

2.1.6 Including momentum

The equations so far do not explain the spontaneous phase separation into relatively free and rather dense regions, which is observed in real traffic. To obtain this, one can no longer neglect the effect of momentum: One can neither accelerate instantaneously to a desired speed nor slow down without delay. It becomes necessary to include the momentum equation. Here, one has to specify the force term F/m, which describes acceleration and slowing down. At least two properties have to be incorporated: interaction with other cars, and relaxation towards desired speed.

- A first order approximation for the relaxation term is [88, 130]

  $$\frac{1}{\tau}(V(\rho) - v),$$

  where $V(\rho)$ is the desired average speed as a function of density.

  This choice yields exponential relaxation towards the desired speed. The function $V(\rho)$ has to be specified externally, e.g. from measurements.

- A commonly used interaction term [26, 80, 88, 130] is

  $$-\frac{c_0^2}{\rho}\partial_x \rho \ , \qquad (*)$$

  where c_0 is treated as constant. In traffic, a typical value for c_0 is 15 km/h [87].

  The meaning is that one tends to reduce speed when the density increases, even when the local density is still consistent with the current speed.

  Formally, this term comes from the the pressure term of compressible flow

  $$-\frac{1}{\rho}\partial_x p \ ,$$

  where p is the pressure due to thermal motion of the particles. Assuming an ideal gas ($p = \rho\, RT$) and isothermic behavior $T = const$, one obtains waves similar to sound waves as a solution of the linearized equations.[5] This leads to ($*$), where c_0 is the speed of the "sound" waves. Note that sound waves move in *both* directions from a disturbance, which means that sound waves alone are not a good explanation for freeway start-stop-waves, contrary to what is written in [14].

A possible momentum equation for traffic therefore is

$$\boxed{\partial_t v + v\partial_x v = -\frac{c_0^2}{\rho}\partial_x \rho + \frac{1}{\tau}\left(V(\rho) - v\right) + \nu \partial_x^2 v \ ,}$$

and the system is closed together with an equation of continuity

$$\boxed{\partial_t \rho + \partial_t(\rho\, v) = D\, \partial_x^2 \rho \ .}$$

Usually, D is set to zero.

---

[5] True sound waves assume the gas to behave adiabatic, i.e. $p \propto \rho^\kappa$. Then, increasing pressure results in an increased temperature.

For this equation, the homogeneous solution $(v,\rho) \equiv (v_0,\rho_0)$ is unstable for densities near maximum flow for a suitable choice of parameters. Using the methods of nonlinear dynamics [88, 143, 156] (see also [64]), one can go beyond the linear stability analysis. One finds a multitude of stable or unstable fixpoints and limit cycles which suggest that traffic near maximum flow operates on a strange attractor. This can lead to quasi-periodic behavior, exactly as is observed in traffic measurements.

Earlier work [26, 58, 130] has analyzed the same equation without diffusion ($\nu = 0$). The results are reported to be unrealistic for bottlenecks, and the equations do not reproduce the start-stop-wave behavior.

2.1.7 Discussion of fluid-dynamical approaches

Fluid-Dynamical models have been used in traffic science for a long time, with considerable success. But they have shortcomings. Some of the major points are:

- One has to give externally the relation between speed or current and density. This is unsatisfying in terms of the development of a theory. But an even more intricate problem is that there is no agreement on a functional form of the speed-density relation; it is even under discussion if this relation is at all continuous [56, 132].
- Temperature parametrizes the random fluctuations of particles around their mean speed. For gases, fluctuations and therefore temperature increase with density. For granular media, fluctuations *de*crease with density (i.e. inside a jam)—it has been claimed that exactly this inverse temperature effect is responsible for clustering [48]. In this way, assuming isothermic instead of adiabatic behavior as done for the momentum equation seems only half the way one has to go. Helbing [61] discusses this further.
- Helbing [61] also discusses *excluded volume* to take into account the spatial extension of vehicles.

Nonetheless, fluid-dynamical approaches [80, 88, 143, 156] give, for the first time, systematic insight into traffic near maximum flow beyond simple extrapolation of light and dense traffic results. Lee [96] has found the same mechanism for waves in general granular media. Putting all their results together, the overall picture is as follows: Near maximum throughput, the flow becomes unstable due to instabilities in the momentum equation (dynamic mode). These instabilities grow exponentially until the wave is outside the unstable region (i.e. too high density). Then the dynamic mode damps out, but the density discontinuity survives as a kinematic wave.

2.1.8 Kinetic theory

The heuristic introduction of the fluid-dynamical equations for traffic flow remains unsatisfactory. As in standard hydrodynamic theory [140], it should be possible to derive correct hydrodynamic equations for traffic from a kinetic theory. Indeed, such an approach has been followed by Prigogine, Herman, and coworkers [138], and in fact, the hydrodynamic equations of Lighthill and Whitham have been derived from this as a limiting case.

In short, the kinetic approach attempts to give the time development of a distribution function $f(x,v,t)$, where

$$\int_0^\infty \mathrm{d}v\, f(x,v,t) = \rho(x,t)$$

and

$$\frac{1}{\rho(x,t)} \int_0^\infty \mathrm{d}v\, v\, f(x,v,t) = \overline{v}(x,t) \; .$$

Technically similar to the fluid-dynamical theory, the dynamic equation for f is

$$\partial_t f + v\, \partial_x f = \text{force terms} \; .$$

Again similar to the fluid-dynamical theory, mainly two terms are considered for the force:

- The relaxation term

  $$-\frac{f - f_d}{T} \; ,$$

  where $f_d(x,v,t)$ is the distribution of desired speeds.

- Following standard kinetic theory, the interaction term is at first written as a master equation:

  $$\left.\frac{\partial f(x,v_i,t)}{\partial t}\right|_{interaction} = (1-P) \sum_j \left[R(v_i|v_j) - R(v_j|v_i) \right] \; .$$

  $R(v_b|v_a)$ is the rate with which cars change from velocity v_a to velocity v_b due to interaction. In gas dynamics, these rates are given by the collision rules. For traffic, it means that an upcoming car slows down to the speed of a slower car. P is the probability of passing: If a car can pass, interaction does not take place. P is itself a function at least of ρ.

  Further approximation of this term leads to the expression

  $$\partial_t f|_{interact} = (1-P)\, \rho\, (\overline{v} - v)\, f \; .$$

  $\partial_t f(v)|_{interact}$ is positive when $v < \overline{v}$ and negative when $v > \overline{v}$, which means that the interaction term itself has the tendency to slow down the traffic to velocities below $\overline{v}$.

  This approximation is only mean field, meaning that it only takes into account interactions of a car with the *average* field of all the others. Higher correlations are in principle feasible.

The problems with the kinetic approach are:

- The mathematics are relatively laborious *and* one needs a lot of specificity (assumptions about driving behavior etc.) at an early level of the derivation. The approach is therefore not useful for investigating a lot of different assumptions.

- Kinetic theory for traffic is so far only mean field: No two-point correlations have been included into the calculations. Although this is in principle feasible, it will be even more difficult mathematically. Yet, these correlations are most important near maximum flow, where we are interested in.

2.2 Particle models and their relation to fluid-dynamical models

2.2.1 Particle models for one-dimensional transport phenomena

In more recent times, investigations of partial differential equations have been complemented by investigations of discrete models [162].

They are much more straightforwardly to simulate on a computer: Local rules usually only consist of a few integer operations instead of many more floating point operations necessary for discretized partial differential equations. And since in addition no round-off errors and subsequent instabilities occur, much faster algorithms (for a given number of sites) are possible. This is especially useful for the investigation of critical phenomena (see below), where large correlation lengths and times and huge fluctuations call for large systems, long runs, and averages over many runs.

Moreover, these discrete models can be approached by analytical many-body methods from statistical physics (see later).

Out of the multitude of particle hopping models, in the context of this work, two of them are of particular interest. Both use a lattice of size L (open or periodic boundary conditions) which is filled with $N = \rho\, L$ particles.

- **Stochastic asymmetric exclusion** [74, 52, 85, 34]: In each update step, one of the particles is selected randomly (random sequential update). If the site to the right of the particle is free, the particle is moved to this site, otherwise it does not move.
- **Deterministic asymmetric exclusion** [85]: All particles are simultaneously updated according to the same rule as before.

  Biham and coworkers [17] have first used this model for traffic flow considerations. Nagatani [117] has used it for two-lane traffic. Moreover, all CA traffic models on two-dimensional grids (see Chapter 8) are based on this velocity update.

2.2.2 Damage spreading and critical exponents

A helpful concept for understanding critical phase transitions in discrete systems is the notion of "damage spreading" [159]: One simulates two identical copies of the system. At a certain point, a minimal change in one of the copies is made and then the time evolution of the *differences* between the systems is observed.

To be more precise, I will use the picture of traffic jams as an example. In Chapters 5 and 6, this picture will be verified and extended for a subcase of the CA traffic model, and Chapter 7 will explain the meaning and limit of the picture in the full model.

In our traffic system, one way of "damaging" is to change the velocity of one randomly picked car by -1. This car then can, by a chain reaction, cause a jam of a certain life-time; and downstream of this jam, the traffic pattern will be different from the undisturbed model. After this jam has dissolved, the *spatial* amount of damage extends from the disturbed car to the last car involved in the jam, and this length is proportional to the life-time of the jam. For the limit $p_{spont} \to 0$ (but $p_{spont} \neq 0$), i.e. where spontaneous initiation of a jam becomes rare (see Chapter 5), one obtains the following picture:

- For low densities, jams are usually short-lived (i.e. with an exponential cut-off in the life-time distribution). As a result, the average amount of spatial damage is small.
- For high densities, a jam caused by the disturbance will (in the average) survive forever, thus (in the average) causing infinite damage.

  However, dense traffic is characterized by the existence of many jams quasi-randomly distributed over the system. So the additional jam caused by the disturbance will not change the *statistical* properties of the system.

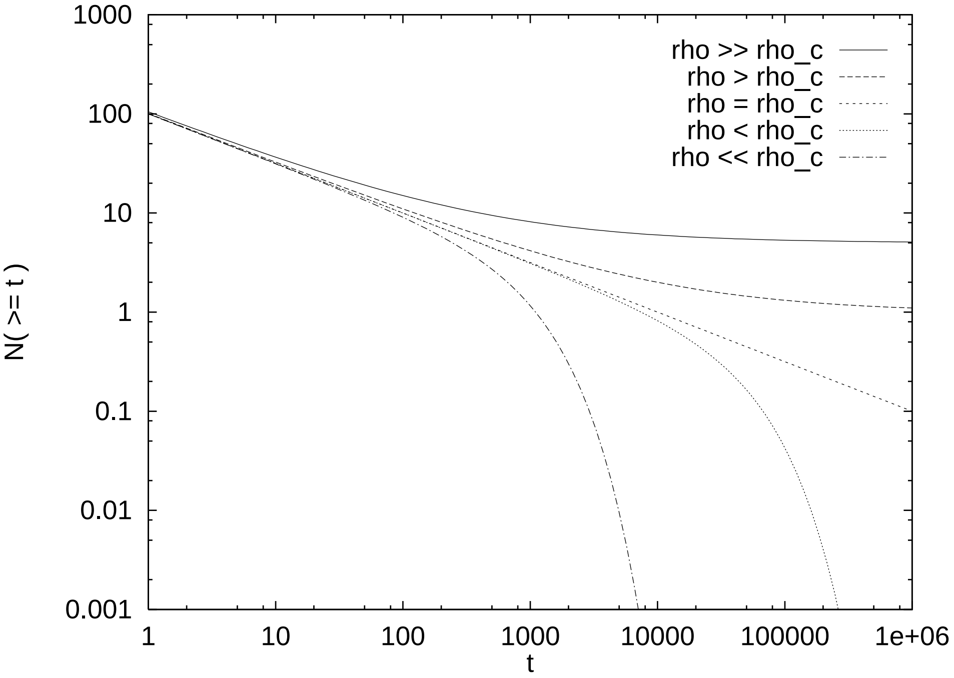


Figure 2.1: Theoretical distribution functions of life-times of traffic jams. The curves show, for different densities, the number of jams with a life-time larger or equal than t as a function of t. (y-axis arbitrary units)

- In between, there should be a density ρ_c, where, when this density is approached from below, jams become increasingly long-lived, with the result that the amount of spatial damage becomes larger and larger. Ultimately, exactly at the critical point, a damage of size infinity (in the thermodynamic limit) is possible.

All these observations are similar to conventional damage spreading observations in cellular automata (CA) [91, 159]: The damage is limited for class I and class II CA, and it *can* be infinite for class VI CA. For class III CA, the damage is practically always infinite, but does not change the statistical properties of the system.

Moreover, at a critical point a system's behavior is characterized by critical exponents. At the critical point, i.e. where jams barely survive, one would expect for the jam survival probability a scaling law:

$$P_{surv}(t) \sim t^{-\delta} ,$$

where $\sim$ means "proportional to in the limit of $t \to \infty$ and $L \to \infty$", and δ is a critical exponent.

But why should that make sense? Why should traffic take place exactly at ρ_c? And if so, should one not use a more realistic model to obtain quantitative numbers (i.e. critical exponents)? The answers to these questions are called in short *self-organized criticality* [8] and *universality* [173].

In our context, the first means that there are microscopic and macroscopic reasons that traffic flow has the tendency to operate near ρ_c. Microscopic, because the outflow from any high density region happens at ρ_c; macroscopic, because optimal use of a road network pushes ρ towards ρ_c. Both results will be explained in detail in later chapters.

Universality means that, in many cases, critical exponents are robust with respect to changes of the model, i.e. grid structure, neighborhood definitions, etc. For the traffic model, we even can explain most of the exponents by a phenomenological theory. In consequence, we expect the results being true for most if not all car following models. The details of this claim will be worked out later in this text.

Traffic practitioners have indeed encountered problems which come out of these fundamental observations. In the TRAF-NETSIM simulation package, a problem was that small control measures could lead to completely different simulation behavior, which is exactly the problem of damage spreading. Rathi and Santiago [139] resolved it by implementing "identical traffic streams", but without a theory it seems doubtful if this is a final solution.[6] Other authors [23] have investigated the variability of NETSIM simulations—an enterprise which also would benefit from more theory such as developed here. For example, it would clarify that variability is density dependent, and where one should expect the highest variability. See also Chapter 9 on variability in network simulations.

2.2.3 Hydrodynamic exponents

Critical exponents have another advantage: They are useful to compare particle hopping models with fluid-dynamical descriptions even when one does not know the hydrodynamical limit of the hopping model. In this way, we will be able to work out the connections of the traffic model proposed in the next chapter to other models in physics and to the fluid-dynamical theories of traffic flow.

For example, dimensional analysis of the KPZ equation (see above)

$$\frac{\partial h}{\partial t} = b_2 \, |\frac{\partial h}{\partial x}|^2 + D \frac{\partial^2 h}{\partial x^2}$$

suggests

$$\frac{H}{T} \sim \frac{H^2}{X^2} , \frac{H}{X^2} ,$$

where H, T, and X are "typical" values for variations in h, t, and x. Since, heuristically, the last term on the RHS is smaller than the first one on the RHS, one finds

$$X^2 \sim H \, T \ .$$

Driving the KPZ equation by noise yields in addition $H \sim X^{1/2}$ [85] and therefore

$$T \sim X^{3/2} \ ,$$

with the well known "anomalous" dynamic KPZ exponent $z = 3/2$.

More precisely, one considers correlation functions of the type

$$F_h^2(x,t) = \langle \, [h(x_0 + x, t_0 + t) - h(x_0, t_0)]^2 \, \rangle$$

[105] and checks if $F \sim x^\xi$ (roughness exponent) or $F \sim t^{\xi/z}$. And by using, e.g., renormalization group arguments, one has shown rigorously that the above exponents for KPZ are exact [18].

---

[6] Actually, in that case the situation is somewhat more complicated. In the original NETSIM simulation package, one random number sequence was used for all randomized elements, especially also for turning decisions at intersections. And since vehicles in NETSIM do not have trip plans, any small change in the random number sequence (e.g. by the addition of one single car) sends each car to completely different destinations. Using true paths will help a lot in this situation, but a fundamental issue remains: One needs theory to evaluate results of Monte Carlo simulations.

2.2.4 Correspondences between field theories and particle models

What now is left is to identify correspondences between field theories and particle models. For the two hopping models presented above, this is done in the following.

Stochastic asymmetric exclusion process

The stochastic asymmetric exclusion process corresponds [85] to the noisy Burgers equation with $\beta = 2$, and therefore corresponds to the exponents mentioned above: $X^2 \sim H\,T$ in general, and $H \sim X^{1/2}$, $T \sim X^{3/2}$ in the steady state.

For our needs, however, this statement has to be more precise. The particle process corresponds to a diffusion equation

$$\partial_t \rho + \partial_x q = D\, \partial_x^2 \rho + \eta$$

with a current $q = \rho\,(1-\rho)$ [85, 34]. Inserting the current yields

$$\partial_t \rho + \partial_x \rho - \partial_x \rho^2 = D\, \partial_x^2 \rho + \eta \ .$$

Since we are searching for fluctuations,[7] we insert $\rho = \langle\, \rho \,\rangle_L + \rho'$ and obtain

$$\partial_t \rho' + (1 - 2\langle\, \rho \,\rangle_L)\, \partial_x \rho' - 2\, \rho'\, \partial_x \rho' = D\, \partial_x^2 \rho' + \eta \ .$$

Thus, when transforming into the moving coordinate system $x' = x + (1 - 2\langle\, \rho \,\rangle_L) \cdot t$, one obtains KPZ exponents. In all other coordinate systems, the linear gradient is dominating [18]. A precise treatment of this uses correlations between tagged particles [105].

In terms of kinematic waves, the explanation of this is as follows: The particle process with a given average density $\langle\, \rho \,\rangle_L$ produces kinematic waves of wave velocity $c = q' = 1 - 2\langle\, \rho \,\rangle_L$. Only in the coordinate system of these waves, one sees the anomalous fluctuations. And there is a density, which—not coincidently [86]—corresponds to maximum flow, where these waves do not move in space. When traffic would follow a Burgers equation, then one could detect maximum traffic flow by standing on a bridge: Jam-waves moving in flow direction indicate too low density, jam-waves moving against the flow direction indicate too high density (cf. Figs. 7.1 and 7.2).

For surface physics, these spatial transformations usually do not matter. People are interested into spatially averaged quantities such as the average surface height fluctuations

$$W^2(t) = \langle\, (h(x,t) - \langle\, h \,\rangle_L)^2 \,\rangle_L \ ,$$

which shows anomalous KPZ behavior $W \sim t^{\xi/z}$ independent of spatial transformations.

Something similar is true for traffic flow. According to this theory, a (small)[8] traffic jam of length l would need a time $t \sim l^{3/2}$ until it would be no longer visible in the system. But at the same time, the jam as a whole can move forwards or backwards. Only at maximum flow, the wave velocity $c = q' = 0$, and the jams stays at the same place.

In consequence, when measuring time series of local quantities, for example the density $\rho(x_0, t)$ for fixed x_0, one finds different correlations and a different frequency spectrum at $\rho = \rho(q_{max})$ [105].

---

[7] This is why the Musha transformation (see above) is not the correct one to obtain critical exponents.

[8] When a jam becomes too big, the system is no longer in the steady state, and a crossing over to other exponents is found.

Deterministic asymmetric exclusion process

The deterministic asymmetric exclusion process corresponds to the deterministic Burgers equation with $\beta = 1$, with a current as described above,

$$q(\rho) = q_{max} - v_{max} \left| \rho - \rho(q_{max}) \right| ,$$

with $q_{max} = 1/4$, $v_{max} = 1$, and $\rho(q_{max}) = 1/2$. Further observations for this model are described in Chapter 7, together with summaries of other cases which are relevant for traffic.

2.2.5 Open systems

In an open system, the number of particles (cars) is no longer conserved. We continue to consider a one-dimensional system. Particles move from the left to the right. One imposes open boundary conditions by allowing particles to enter the system at the left end ($x = 0$) at rate ρ_0 and letting them leave the system at the right end ($x = L$) at unit rate. This is equivalent to fixing the density as $\rho(x=0) = \rho_0$, $\rho(x=L) = 0$.

For these systems, one generically has to distinguish two regimes [86]:

- $\rho_0 < \rho(q_{max})$. In this case, the density in the whole system except for the boundary regions is equal to ρ_0. Near the right boundary, ρ decays to zero [86]. The size of the boundary region goes to zero in the hydrodynamic limit [21, 34].
- $\rho_0 > \rho(q_{max})$. In this case, the density in the whole system is equal to $\rho(q_{max}) < \rho_0$, i.e. the system maximizes the flow and adjusts the density [86].

The second regime is interesting: As explained above, the case $\rho = \rho(q_{max})$ shows anomalous fluctuations for a fixed observer. By the automatic selection of this state, this supposedly rare case suddenly happens generically—which may be seen [86] as a case of self-organized criticality [8].

These observations are “generically” true for a large class of systems, but exceptions exist.

For the traffic CA, these observations give the correct ideas, but the details are distinctly different. That will put us into a position to judge where the Burgers equation and the more general Lighthill-Whitham description are incomplete for traffic.

2.3 Car following models

In socio-economic systems such as traffic, one does not have first principles for particle behavior to start from. It is therefore somewhat difficult to discuss the justification of a certain modeling approach—in many cases, one will have to argue from the usefulness of the results. Nonetheless, car following theory has been a field under development for a long time now, and common agreement has established a variety of results.

2.3.1 Mathematical car following theory

Mathematical car following theory assumes a linear relationship between the (re)action and a stimulus (cf. [47] for the following):

$$\text{action} = \text{sensitivity} \cdot \text{stimulus} \ .$$

The stimulus could, e.g., be the distance to the car ahead, or the difference of the velocities. One of the first propositions was

$$v_i(t+\tau) \propto \Delta x_i(t) \ , \qquad (*)$$

where v_i is the velocity of the ith car in a chain, τ is the time which is necessary for the adaption (this is more than the reaction time), and Δx is the distance to the next car ahead. Most of the theory treats vehicles as points. Especially in recent times, results have been obtained starting from this approximation [10, 125, 176].

Deriving $(*)$ once with respect to time leads to

$$a_i(t+\tau) \propto \Delta v_i(t) \ ,$$

where a is the acceleration of i and Δv is the velocity difference to the next car ahead. A generalisation of this ansatz, taking into account that the sensitivity depends on the distance and on the speed, is

$$a_i(t+\tau) = c_1 \cdot \frac{[v_i(t)]^m}{[\Delta x_i(t)]^l} \cdot \Delta v_i(t) \ . \qquad (**)$$

l and m are integer numbers. This equation has been analyzed in great detail in the 1960s, for example with respect to its stability for a single following car (local stability) or for an infinitely long chain of cars (asymptotic stability). On can derive speed-density and therefore flow-density relations from $(**)$. The resulting curves are, however, not extremely realistic when compared to modern measurements [1, 2, 56, 69, 132].

In any case, equation $(**)$ is problematic for numerical modelling. When Δv is zero, the equation allows an arbitrarily small distance to the car ahead, even at very high speed. In the theory, this never happens as long as one starts from realistic initial conditions, but due to numerical inaccuracy for simulations such situations could occur. It is very probable that it will occur in a model with only a few states, as is the model which will be introduced in the next chapter.

As a result, practical implementations either use distance alone as stimulus [51], or they use it to provide some bounds for the velocity-based reactions [16, 177].

2.3.2 Psycho-physiological spacing models

Another approach to car following is based on psychological and physiological observations. Wiedemann [98, 177], before describing a microscopic car-following simulation model, reviews these aspects:

A starting point is that the *angular* resolution of the human eye is limited. Even when a human can detect another car as coming closer, it can estimate its *velocity* only when the car becomes noticeably bigger in the field of vision—i.e. when the angle between left and right parts of the car increases. By using geometric arguments, one quickly sees that this observability threshold is proportional to the speed difference, and inversely proportional to the square of the distance:

$$threshold \propto \frac{\Delta v}{\Delta x^2} \ .$$

Note that this is similar to the mathematical car following model $(**)$ with $l = 2$ and $m = 0$.

Another element of psycho-physiological car following models is that the reactions are, as a first approximation, step functions instead of smooth ones. Thus, after crossing the threshold, slowing down or braking is adjusted with the goal to reach the speed of the car in front together with a certain desired spacing. Measurements, consistent with the physiological argument, indicate that the reaction threshold is largely independent of one's velocity. Thus, sometimes immediate emergency braking may be necessary.

Furthermore, the human's braking reaction usually is not very precise, so that further control action is necessary, leading to a highly stochastic element in car following.

Similar observations are true if the distance to the car in front increases, except that reactions are slower. This can be explained in part again physiologically, and in part by the weaker motivation to react (less dangerous).

More precise results here should be possible using results from ecological psychology. Ecological psychology is concerned with finding general principles as to how organisms, including humans, interact with their environment. Visual perception links the perceiver with his/her environment via an optic flow field, which provides the action relevant visual information. The goal is to find low-dimensional quantities that capture this information as is provided in the flow field.

In the context of car following, the concept of "time to contact" is relevant. Time to contact and its derivative has proven to be a comprehensive descriptor for informing the perceiver about the necessary movement behavior given a goal. This goal could be, for example, to arrive smoothly at a desired car following position [82, 95].

2.4 Simulation and the role of microscopic high speed models

Naively, simulation seems a perfect tool: One simply formulates every aspect of the system as programming code, and then runs the code (i.e. the model) on a computer. When initial conditions and the system dynamics are only realistic enough, this should lead to a good prediction.

There are, though, severe limitations: Simulations are resource-limited; and nonlinearities make predictions even more resource-demanding. The second aspect will be discussed further down; the first leads to a trade-off between resolution, fidelity, system size, simulation speed, and resources [11]:

- **Resolution:** The *level* of detail of a model. All processes of the system which lead, by *upward causation*, to the *emergence* of the phenomena of interest, may be important.
- **Fidelity:** The degree of realism in modeling each process.
- **System size:** The system size which can is covered by the simulation.
- **Simulation speed:** The speed of the simulation, often compared to real time.
- **Resources:** Often computational resources, but also limitations of the time one wants to spend on the programming.

This trade-off has to be balanced in conjunction with the problem and the question under consideration. With respect to resolution, one often distinguishes between macroscopic and microscopic models. In traffic science, macroscopic means that one averages over vehicles, whereas microscopic means that at least vehicles are resolved individually.

Another classification of models is according to their intended use: models for, e.g., signalized intersections, arterial networks, freeway corridors, or rural highways. In the context of this work, models for arterial networks and models for freeway corridors are more important than the others.

For reviews of traffic simulation models see, e.g., [47, 108, 170].

2.4.1 Macroscopic models

For practical network control problems, only macroscopic models are in use, e.g. FREQ, CORQ, NETFLO, or CONTRAM [108, 146, 170]. For CONTRAM and its traffic assignment algorithm, parallelization attempts are reported [50, 65]. TRANSYT [174] is mostly used for signal optimization. In the area of freeway simulation, models often are based on discretizations of fluid-dynamical equations (MACK, FREFLO, FRECON [170, 146], cell-transmission-model [31, 133], [26, 29]). It has been claimed [32] that many of these approximations have not been done systematically, i.e. that the hydrodynamic limit of the finite difference equations would not be consistent with the fluid-dynamical equations. However, it should be kept in mind that this does not necessarily matter because the fluid-dynamical equations themselves are not exact for traffic flow.

2.4.2 Microscopic models

Is is quite obvious that certain questions can only be answered by a microscopic model. To understand the relation between traffic flow phenomena, such as jams, and individual vehicle characteristics (e.g. driving behavior, cruise control, radar-based automatic car following), one has at least to resolve the individual vehicles. Dynamic individual routing just does not make much sense when individual vehicles are not present: For example, to react to incidents in an ATMS/ATT/IVHS environment, destination information is necessary (cf. Fig. 2.2). Other microscopic questions are modal choice or pollution (engine temperature!).

These questions quickly lead to 500 000 or more "intelligent objects" in the system and to the necessity of corresponding computational power.

There are only few well-known simulation models which are at the same time microscopic and operating on a network level. TRAFF-NETSIM [124] was developed for the Federal Highway Administration of the USA more than 20 years ago and has been continuously updated and enhanced since that time. The model is based on a stochastic, time-stepping simulation approach describing the dynamics of traffic operations in urban street networks, which consist of uni-directional links and nodes. A wide variety of options and modules is available, including signal controllers, bus operations, measures of effectiveness, or estimates of emission or fuel consumption.

However, for our needs, NETSIM is too slow, there exists no version which runs efficiently on parallel supercomputers, and it does not allow for trip plans: Vehicles at intersections are distributed according to turn counts (cf. Fig. 2.2).

PARAMICS [181, 110] is a rather new approach, specifically designed for parallel computers (PARAMICS = PARAllel MICroscopic traffic Simulator). It is, so far, the only approach comparable to the work presented here in its possible applications.

2.4.3 Frameworks

A severe shortcoming of all the models above is that they only deal with one part of the transportation system: road traffic. However, in recent times it became increasingly clear that future solutions will need an integrated traffic management, and modal split (i.e. the use of different modes of transportation) will become a necessity. The TRANSIMS project [165] is a reaction to that. It is a framework planned to include a wider spectrum of transportation activities than ever before, such as:

- Trip generation of individuals from demographic data

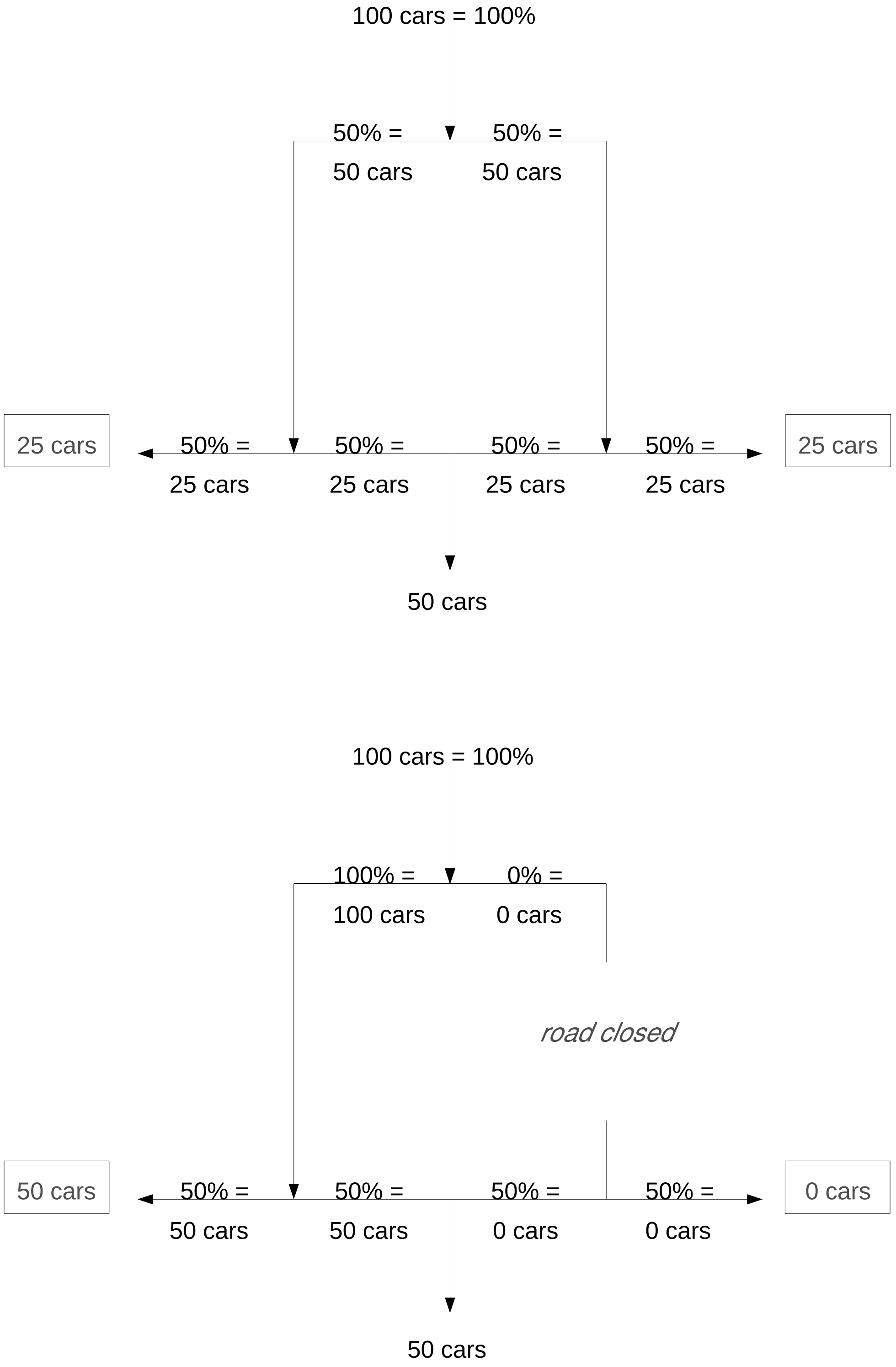


Figure 2.2: Simple example demonstrating the need of microsimulation and travel plans for incident management. *Top:* Undisturbed network. *Bottom:* One road is closed. In both networks, 100 cars are entering from the top, and at junctions they are distributed according to turn-count percentages. From the undisturbed network one sees that 25 cars exit at the right and 25 cars at the left (boxed values). With the closed road, the turn-counts simulation completely misses the correct destination pattern: 50 cars exit to the left, and none to the right (boxed values).

- Planning of trips, routing
- Execution of trips (microsimulation)
- Incident detection
- Meteorological module: air pollution
- Urban development

Current status is that a demo version shows the feasibility of the approach, and a large team of scientist and engineers currently works on its realization. This thesis, in a collaboration between Cologne and Los Alamos, is also part of the TRANSIMS project.

EUROTOPP [153] is a framework similar to TRANSIMS, but is does not have parallel supercomputing plans so far.

PARAMICS, coming from the supercomputing side of traffic microsimulation, is planning to extend towards a framework similar to TRANSIMS.

2.4.4 Nonlinear Dynamics and Statistical Physics: A dynamic systems approach to transportation

So far, together with a general overview over freeway simulation models, it has been shown that microscopic modelling is a necessity for many of the urgent questions. Another desire is the wish for simulation speed. Besides for real time applications, where the simulation has to run many times faster than reality, here another argument will be shown: The necessity of systematic analysis of traffic/transportation as a dynamic system.

Even with a perfect prediction machine, a complete traffic flow forecast would not be possible. Nonlinear Dynamics [152] tells us that tiny changes in the initial conditions can lead to completely changed long-term behavior even in deterministic systems. The best-known example for this is the butterfly effect ("A butterfly in Japan may influence tomorrow's weather in Europe.") [152].

But the situation is not completely lost. Short-term predictions are always possible, and depending on the region in phase space, such a prediction will be better or worse. And in addition it will be possible to identify "average" traffic regimes and to predict their changes under changes of parameters: In the context of the routing problem, one would like to know how different algorithms perform, e.g., in light vs. dense traffic, and if one can generalize these results.

In addition, it will turn out that the optimization process itself affects the system dynamics, in a way that optimization of average quantities (such as overall cost) will lead to high fluctuations of individual quantities: Although the overall connection between two given points may be faster as the result of the optimization, unexpected breakdowns of the whole system may happen more often than before. In short, the optimization procedure becomes part of the system dynamics and can no longer be seen separately as in other optimization problems, such as job scheduling or tour planning [93].

Quite generally, it appears that simulation does not remove the old task of science: The search for structure is as necessary as ever. Clearly, to deal with such questions, a dynamic systems approach to systems such as transportation is necessary. Barrett and Rasmussen [12] describe steps towards formalizing this search for structure; similar questions are currently discussed in the Artificial Life (alife) research community [144].

One of the currently best concepts to characterize the phase space of a spatial stochastic system is "damage spreading" [159] (see above). Investigations like this need a lot of averaging over many

Monte Carlo runs. Since, in addition, things become worse near a critical point ("critical slowing down": Large system sizes and long times are necessary to avoid finite size effects [66]), models useful for this kind of analysis have to run by orders of magnitude faster than traditional forecasting models.

2.4.5 Microscopic high speed models

In consequence, a traffic simulation model with high resolution *and* high speed would be useful. PARAMICS achieves this by plainly setting on parallel supercomputers together with a not too sophisticated car following dynamics. But according to the arguments above, ultimately one has to sacrifice fidelity. The DYNEMO model [154] achieves simulation speed by moving the individual vehicles according to a macroscopic flow, thus indeed reducing car-following fidelity to achieve computational speed.

Another approach is to reduce fidelity directly on the car following level. Similarly to the *lattice gases* in fluid dynamics [160], space, time, and the possible states are discretized. The roadway is separated into boxes, and each box is either occupied by exactly one vehicle, or it is empty. Movement of vehicles is restrained to the boxes. Occupied boxes (i.e. vehicles) can have further properties, such as speed, vehicle type, and so on. In each update step, all these properties have to be moved with the vehicle, which costs computational resources.

Already an early paper on traffic simulation [46] contains an idea which comes close to a lattice gas: Vehicles are restricted to boxes, and the only difference to a lattice gas is that the boxes are not equally spaced. Instead, spacing reflects velocity at certain points of a network: Regions of short spacing allow only small movements in each time-step and thus low velocity.

A modern version of the lattice gas method (including single bit coding) has been introduced for traffic simulation in 1986 [28]. Here, boxes are equally spaced, and vehicles move with different velocities. The model has been developed towards a complete simulation tool allowing different vehicle types and having different modules for multi-lane freeways, two-lane highways with oncoming traffic, and urban traffic [151]. The main problems of this model are:

- The employed bit-coding scheme does not vectorize. As a result, the model did not offer itself for large scale Monte-Carlo simulations on the hardware available at that time. But then, without a large-scale speed advantage, bit-coding is too inflexible for modern traffic science question.
- Vehicle movement is only from one site to the next. The model allowed for velocities up to $v_{max} = 6$, but this had to be achieved by 6 single site movements. In consequence, when the model did not use bit-coding, it was relatively slow.

In recent time and started by the independent publication of [17] and of the first results of this thesis work [120], CA models for traffic are increasingly used, this time by Statistical Physicists, partially for one-dimensional [9, 15, 92, 111, 116, 117, 147, 150, 167, 172] and partially for two-dimensional [30, 112, 119, 118, 166] investigations (i.e. traffic on a square grid). I will mention most of the approaches again in the text.

Chapter 3

A cellular automaton model for traffic flow

In this chapter, a basic model for (single lane) traffic flow is proposed, which is an adaption of the lattice gas method to traffic. Phenomenology and average behavior of the model are studied and compared to reality. The influence of the two free parameters of the model is investigated, followed by some observations concerning calibration.

3.1 This work

According to arguments given in the last chapter, for high simulation speed one ultimately has to sacrifice fidelity, and one acceptedly useful way of doing this is to use the lattice gas methodology. But how much fidelity does one really need?

Due to the lack of "first principles" in traffic science, only an empirical answer can be possible. From all possible elements of driving dynamics, one has to select the most important ones.

This work is exactly about taking this idea to the extreme: Individual cars are preserved, but driving behavior is simplified as much as possible without losing the essentials of traffic jam dynamics. And the simplification matches computer architecture considerations as much as possible.

The limits of simplification are certain properties of vehicular traffic which one wants to preserve, for example spontaneous jam formation. It will come out that the model specified in the next section is "minimal" in the sense that every further simplification leads to losing one of these essential properties.

This approach will turn out to indeed run fast enough to use averaging methods from Statistics / Statistical Physics, for real time applications in large road networks, for algorithm development for dynamical routing, or for long-term urban planning considerations.

Moreover, since a local and simultaneous update will be used, the model will formally be a cellular automaton (CA) [180]. This will simplify further analysis: Many spatial methods of statistical physics have been developed in connection with CA. Analytical treatment is possible (which is, though, laborious, and at the edge of currently available methods, see later), and it allows computational research for finding the essentials of traffic jam dynamics. This is especially useful for traffic near maximum throughput, which is the range which is most interesting for planning purposes but explained worst by conventional methods.

Many models could be used to follow this route. However, I will, by testing state-of-the-art programming techniques such as single-bit coding, argue that a more efficient use of current computer architectures will be difficult to achieve. And I will argue that traffic has measurable "universal" properties which will be reproduced by any microscopic model whatsoever, extensive or not.

3.2 The model

The basic computational model is defined on a one-dimensional array of L sites with open or periodic boundary conditions. This could, for example, be an edge in a road network. Each site is either occupied by one vehicle, or empty. Each vehicle has an integer velocity with values between zero and v_{max}. The number of empty sites in front of a vehicle is denoted by *gap*. For an arbitrary configuration, one update of the system consists of the following four consecutive steps, which are performed simultaneously for all vehicles:[1]

1.) **Acceleration:** If the velocity v of a vehicle is lower than v_{max} and if there is enough room ahead ($v \leq gap - 1$), then the speed is increased by one:

```
if $(v \leq gap - 1)$ then $v := \max[v_{max}, v + 1]$ .
```

2.) **Slowing down (due to other cars):** If the next vehicle ahead is too close ($v \geq gap + 1$), speed is reduced to *gap*:

```
if $(v \geq gap + 1)$ then $v := gap$ .
```

3.) **Randomization** (*which is applied after rules 1 & 2*): With probability p, the velocity of each vehicle (if greater than zero) is decreased by one:

```
with probab. $p$ do $v := \max[v - 1, 0]$ .
```

4.) **Car motion:** Each vehicle is advanced v sites.

Through the steps one to four very general properties of single lane traffic are modelled on the basis of integer valued probabilistic cellular automaton rules [160, 180]. Already this simple model shows nontrivial and realistic behavior.

The Monte Carlo simulations have mainly been carried out with the choice of $v_{max} = 5$ for reasons stated below. It should be noted that the term maximum speed does not reflect the physically possible maximum speed of each car. It reflects, however, a maximum speed as attained in reality, restricted, for example, by speed limits or by preferences of the driver.

Step 3 is essential in simulating realistic traffic flow since otherwise the dynamics is completely deterministic. It condenses three different behavioral patterns into one computational rule:

- **Fluctuations at maximum speed:** Assume a vehicle driving at maximum velocity $v = v_{max}$ and with $gap \gg v$ (free driving). The velocity remains unchanged in rules 1 & 2; in rule 3, it is, with probability p, reduced to $v_{max} - 1$. In the next iteration, according to rule 1, the speed $v_{max} - 1$ is algorithmically reset to v_{max}, before the randomization can again reduce it with probability p. Thus, according to the algorithm, a free vehicle moves with frequency $1 - p$ with $v = v_{max}$, and with $v = v_{max} - 1$ otherwise.

---

[1] Note that *either* rule 1 *or* rule 2 applies, but never both.

This reflects the fact that people even in light traffic do not keep their speed perfectly constant, for example due to distractions.

- **Retarded acceleration:** Assume a single vehicle with $v = 0$. Taking rules 1 & 3 together, the vehicle accelerates to $v = 1$ with probability $1 - p$. The deterministic acceleration sequence

  $$0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow v_{max}$$

  is thus replaced by a multitude of stochastic sequences, for example

  $$0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 2 \rightarrow \ldots \rightarrow v_{max} \; .$$

  It seems realistic that people often wait longer than necessary when the car ahead already moves.

- **Over-reactions at braking:** When people suddenly see slow traffic ahead, they tend to overcompensate. In the rules, instead of reducing the speed exactly to gap, with frequency p they reduce it further to $gap - 1$ (if ≥ 0).

Without this randomness, every initial configuration of vehicles and corresponding velocities reaches very quickly a stationary pattern which is shifted backwards, i.e. opposite to the vehicle motion, one site per time step (see Chapter 4).

In the framework of car-following models described in Chapter 2, the CA traffic model is difficult to classify. It only uses spacing as a stimulus; the braking rule is similar to a model where the velocity is proportional to the spacing, whereas the acceleration is constant as soon as there is enough room ahead.

When judging the realism of the model beyond these general remarks, one should keep in mind that this model is intended to be a *minimal* microscopic model. As already mentioned above, I will in the end argue that the model goes beyond all of the current fluid-dynamical theory of traffic jam dynamics, and that even for this, most of the exact microscopic properties do not matter. Only *after* having identified the important elements, it will certainly make sense to think about making them more realistic (which will usually mean slower computational speed, though).

For that same reason, this text concentrates on single-lane models. Multi-lane results obtained by other authors will be shortly mentioned in Chapter 12.

Certain choices of the model were originally done for computational reasons, especially for a bitwise implementation:

- Making the model grid-based: (a) Theory for evaluating spatial dynamic systems are much better developed for (few state) grid based models. (b) In view of a planned single-bit implementation, there is no alternative.

- Velocity of the next car ahead is not taken into account: In a bitwise implementation, just checking bit-fields for the presence of a car is much easier to implement (and thus runs much faster) than complicated calculations based on some number different from 0 or 1.

- The choice of the randomization parameter of $p = 1/2$ and the fact that is is equal for acceleration, braking, and free driving, also reflects the fact of an intended bitwise implementation: A bitwise probability of 1/2 is fast to obtain—a randomly selected integer number has its bits randomly occupied with probability 1/2. Other bitwise probabilities are more costly. Random number generation typically consumes about 20% of the computer time for simulations of this traffic flow model.

- Random sequential (cf. Section 2.2) versus simultaneous versus event driven update: Event driven update means that each object keeps on doing the same until some stimulus exceeds a certain threshold—for example that another car comes too close. The most realistic update obviously is the event driven update; and random sequential update is claimed to be a better approximation of it than parallel update because in reality it is more probable that the events occur randomly than in some ordered fashion.

  The exact consequences of the parallel choice for the traffic model will be discussed much later. Roughly, it turns out that for random sequential update to be realistic one would have to make the time scales much shorter. For the CA traffic model, it is certainly much more realistic to assume that all cars move simultaneously. Another reason is computational: For both discrete event simulation and random sequential update an efficient geometric parallelization is hard to achieve:

  For discrete event simulations one needs some more or less optimistic scheduling schemes of running ahead and rolling back [19, 45] to be efficient on parallel machines, which is much harder to implement.

  For random sequential update the problem is that a global determination of the next (randomly chosen) update site obviously is completely inefficient. Localizing this procedure, though, leads to the introduction of a new length scale (the system size between two computational nodes), which might destroy critical properties. Or one would have to implement a scheduler similar to the one for the discrete event simulation.

In an evaluation of the computational speed (Chapter 11), one finds that the bitwise implementation is not much faster on non-vectorizing parallel computers, which are currently the majority of installed supercomputer hardware. Moreover, for a hybrid model together with a more realistic car following logic, which is planned in the TRANSIMS project, the bitwise implementation is useless; the same is already true for the trip plans implementation of Chapter 10. Both arguments together make the use of a bitwise implementation of the model for the future improbable; some of the restrictions above could therefore be relaxed in future work.

3.3 Model vs. reality

Modeling traffic flow as particles which jump from one box to another seems like a quite crude approximation of reality. Can one expect realism from the simulation of such a model? As a first answer to this question, I will make a phenomenological and a quantitative comparison to reality which will show that the model reproduces some aspects of reality rather well. Theoretic arguments supporting this claim will be given later in the text.

3.3.1 Trajectories of cars

A useful way to visualize traffic and traffic jams are space-time diagrams (Fig. 3.1). One can follow the trajectory of each car from top left to bottom right; fast cars have trajectories close to horizontal, slow cars have trajectories close to vertical. An exactly vertical trajectory means that the vehicle does not move at all. In Fig. 3.1 one clearly sees a jam wave, recognizable by the vertical trajectories denoting slow vehicles, and by the high density in that region (trajectories close together). Note that the jam wave moves backwards, i.e. *against* the traffic.

Figure 3.1: Space-time-lines (trajectories) for cars from Aerial Photography (after [168]). Each line represents the movement of one vehicle in the space-time-domain.

Figs. 3.2 and 3.3 show the corresponding plots from the cellular automaton. These plots are standard for cellular automata outputs. Lines are configurations at consecutive time-steps, integer numbers show velocities of cars just before movement. For traffic flow, the plots are at the first view somewhat unusual. Nonetheless, one can follow a car individually from left to right, and if one would connect the positions of one car in space-time, one would get the same trajectories as in Fig. 3.1.

Figs. 3.2 and 3.3 show typical situations at low and high densities. Whereas we find laminar traffic at low densities, there are congestion clusters (small jams) at higher densities, which are formed randomly due to velocity-fluctuations of the cars. If one follows (in Fig. 3.3) one individual car coming from the left, one sees that the car comes in with a speed varying between four and five and then has to stop due to the congestion cluster. There it stays stuck in the queue for a certain time with some small moves, and can re-accelerate to full speed after having left the cluster at its end. So the cluster represents a typical start-stop-wave as found in freeway traffic, and compared to Fig. 3.1, it looks rather realistic.

Figs. 3.4 and 3.5 show larger portions of the system. Vehicles are reduced to dots, i.e. the plot just shows the space-time-position of each vehicle during the evolution.

```
.........................4................................5...........5
.............................4.................................5.......
.5...............................5..................................4..
5.....5...............................5................................
.....4.....4...............................5...........................
.........5.....4................................5......................
.5............3....4.................................4.................
......5..........3.....4.................................4.............
...........5........3......5.................................5.........
................5......4........4.................................4....
.....................4.....5........5.................................4
.........................5......4........4.............................
..............................5.....5........5.........................
```

Figure 3.2: Simulated traffic at a (low) density of 0.03 cars per site. Each new line shows the traffic lane after one further complete velocity-update and just before the car motion. Empty sites are represented by a dot, sites which are occupied by a car are represented by the integer number of its velocity. At low densities, we see undisturbed motion.

```
........4................5......5......5........00.1.......4..........5
..4.........5.................4......5......2...00..2..........5.......
......4..........5................4.......3...0.01....2.............5..
4.........4...........4...............5......01.1.2.....3..............
....5.........5...........5................0.1.0.2..3......4...........
.........5.........4...........4...........1..00...3...3.......5.......
.4............4........4...........5........1.00......3...3.........5..
.....4............5........4............3....000.........2...3.........
.........4.............5.......4...........0.001...........3....3......
.............4..............5......4.......0.00.2.............4....4...
4................4...............5.....2...1.01...2...............3....
....4................4................2..1..01.2....3................4.
........5................5..............0.1.0.2..2.....3...............
.............5................5.........0..00...1..2......3............
..4...............4................3....1..00....2...3.......3.........
......5...............4...............2..0.00......2....3.......3......
..5........4..............4.............01.00........3.....4.......4...
.5.....4.......5..............5.........0.000...........3......4.......
......4....5........5..............4....1.001..............3.......4...
4.........5.....5........4.............1.000.1................4........
....5..........4.....5.......5..........0000..1...................5....
.........4.........5......4.......4.....0000...2.......................
.............5..........4.....5.......0.0001.....2.....................
..................4.........4......1..0.000.2......2...................
.5....................4.........2...0.1.000...3......3.................
......5...................5.......1.0..0001......4......4..............
..5........5...................2...01..001.2.........4......5..........
.......5........4................1.0.0.01.1..3...........5.......4.....
....5.......5.......5.............00.0.0.1.1....3.............4......5.
.........4.......5.......5........01.1.0..0.1......3..............4....
.............4........5.......2...1.0.00..1..2........4...............5
.................5.........4....2..01.01...2...3..........5............
5.....................4........2..01.00.2....3....3............4.......
.....4....................5......00.000...3.....3....4.............5...
```

Figure 3.3: Same picture as Figure 1, but at a higher density of 0.1 cars per site. Note the backward motion of the traffic jam.

3.3.2 Fundamental diagrams

For the next reality check, more aggregated quantities are used. We start with systems with periodic boundary conditions (thus simulating traffic in a closed loop as in car races but only on a single lane).

Figure 3.4: Evolution of the model from random initial conditions. Each black pixel represents a vehicle. Space direction is horizontal, time is pointing downwards, vehicles move to the right. The simulation was of a system of size $L = 10000$ with density $\rho \approx \rho(q_{max}) \approx 0.08$; the figure shows the first 1000 iterations in a window of $l = 1000$.

As the total number N of cars in the loop cannot change during the dynamics, it is possible to define a constant system density

$$\langle \rho \rangle_L = \frac{N}{L} = \frac{\text{number of cars in the loop}}{\text{number of sites of the loop}} \; .$$

Figure 3.5: Evolution of the model from random initial conditions. Each black pixel represents a vehicle. Space direction is horizontal, time is pointing downwards, vehicles move to the right. The simulation was of a system of size $L = 10000$ with density $\rho \approx \rho(q_{max}) \approx 0.08$; the figure shows the first 1000 iterations in a window of $l = 1000$.

Other system-averaged quantities are the "travel velocity" $\langle\, v^{trav} \,\rangle_N$ and the flow $\langle\, q \,\rangle_L$:

$$\langle\, v^{trav} \,\rangle_N = \frac{1}{N} \sum_{i=1}^{N} v_i \ , \qquad\qquad \langle\, q \,\rangle_L = \frac{1}{L} \sum_{i=1}^{N} v_i \ ,$$

where v_i is the velocity of the ith car and the sum runs over all cars. v^{trav} is in the average identical to the velocity each driver measures for himself when traveling, therefore the name.

For practical reasons, traffic engineers prefer quantities which can be measured at a fixed position. Thus, in order to mimic real conditions, we measure[2] densities (= occupancies) $\langle \rho \rangle_T$ on a fixed site i averaged over a time period T:

$$\langle \rho \rangle_T = \frac{1}{T} \sum_{t=t_0+1}^{t_0+T} \delta_i(t) \ ,$$

where $\delta_i(t) = 0$ (1) if site i is empty (occupied) at time step t.

The time-averaged flow $\langle q \rangle_T$ between sites i and $i+1$ is defined by

$$\langle q \rangle_T = \frac{1}{T} \sum_{t=t_0+1}^{t_0+T} \delta_{i,i+1}(t)$$

where $\delta_{i,i+1}(t) = 1$ if a car motion is detected crossing the boundary between sites i and $i+1$.

For large T, large L, and periodic boundaries one has $\langle \rho \rangle_T = \langle \rho \rangle_L$ and $\langle q \rangle_T = \langle q \rangle_L$.

The *local velocity* is defined as

$$\langle v^{loc} \rangle_T := \frac{1}{n(T)} \sum_{j=1}^{n(T)} v_j$$

where $n(T)$ is the number of vehicles which cross the boundary $i \rightarrow i+1$ during the time interval T, and v_j are the vehicles' velocities. It should be noted that v^{trav} generally is lower than v^{loc} due to differences in the statistical sampling. For example, vehicles of velocity $v = 0$ are included for v^{trav} but excluded for v^{loc} (see, e.g., [183]).

With these definitions, it is easy to perform many simulations with different vehicle densities, thus—after relaxation to equilibrium—getting data for the commonly used fundamental diagrams flow $\langle q \rangle_T$ vs. density $\langle \rho \rangle_T$, or $\langle v^{loc} \rangle_T$ vs. $\langle \rho \rangle_T$, etc.

Fig. 3.6 was obtained starting with a random initial configuration of cars with density ρ and velocity 0 and beginning the collection of data after the first t_0 time steps, typically $t_0 = 10^4$. Whereas the broken line indicates the results of averaging over 10^6 time steps, the scattered points represent averages over only 10^2 time steps. The latter should be compared to data from real traffic (Fig. 3.7). For low densities, flow increases linearly with density, reflecting the fact that in this regime adding vehicles does not decrease the velocity and thus just adds to the flow: $q_{low.density} = \rho \, v_{max}$.[3] High densities are similarly easy to explain: Many cars have few holes between them. When a car moves forward, the hole effectively moves backwards. In that sense, flow is due the movement of the holes between jammed cars.

In between, no easy explanation is available. For the long-time averages the maximum flow $q_{max} = 0.318 \pm 0.0005$ is found at the density $\rho = 0.085 \pm 0.004$.

Further simulations show that the position and the form of the maximum of $q(\rho)$ depend on the system size (Fig. 3.6, see [92] for a more detailed analysis). Simulations without randomization, i.e. $p = 0$, do not show a dependence on the system size.

---

[2] I use the term "measurement" both for reality and simulation, somewhat similar to a "measurement operator" in quantum mechanics or in statistical physics which extracts averaged information from a microscopic system. When it is not clear from the context, I will try to specify if I am referring to reality or to a simulation. E.g., "traffic measurement" refers to measurements of real traffic.

[3] This relation is no longer linear in ρ as soon as one includes passing as well as vehicles of different speed [141].

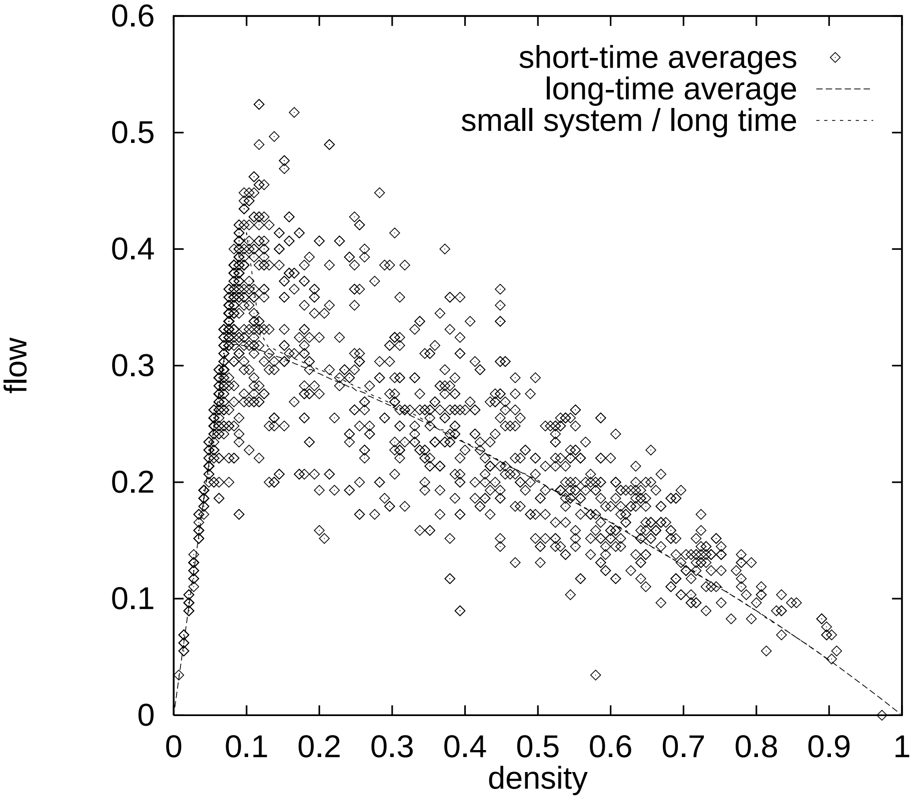


Figure 3.6: Fundamental diagram of the model (throughput versus density). Points: Averages over short times (100 iterations) in a sufficiently large system ($L = 10,000$). Broken line: Long time averages (10^6 iterations) in a large system ($L = 10,000$). Dotted line: Long time averages in a small system ($L = 100$).

3.3.3 Fluctuations

The variance of v^{loc} is defined as

$$\sigma(v^{loc}) = \sqrt{\frac{1}{N}\sum_i v_i^2 - (v^{loc})^2} \ .$$

This quantity may be used for detecting instabilities in the traffic flow and has in fact been used in Germany to install adaptive speed limits [183]. Fig. 3.8 shows that also in the model $\sigma(v^{loc})$ drastically increases near maximum flow.

3.4 Varying the parameters of the model

In order to get an overview of the influence of the model's two parameters v_{max} (the maximum integer speed) and p (the amount of fluctuation included by the pseudo-random numbers), simulations with different values of these parameters were performed. A relatively large closed system ($L = 10^6$ sites) was used, where it was known from previous tests that this is large enough to prevent visible finite size effects for the measured quantities. Starting from an initial random distribution of vehicles, the system run for 10^4 time-steps in order to let the transients die out. Then, every 10^3 iterations one measurement step was inserted, and the average of 100 such measurements gives one data point. Each

Figure 3.7: Fundamental diagram as found in reality from five-minute averages. Data from a Canadian expressway [56].

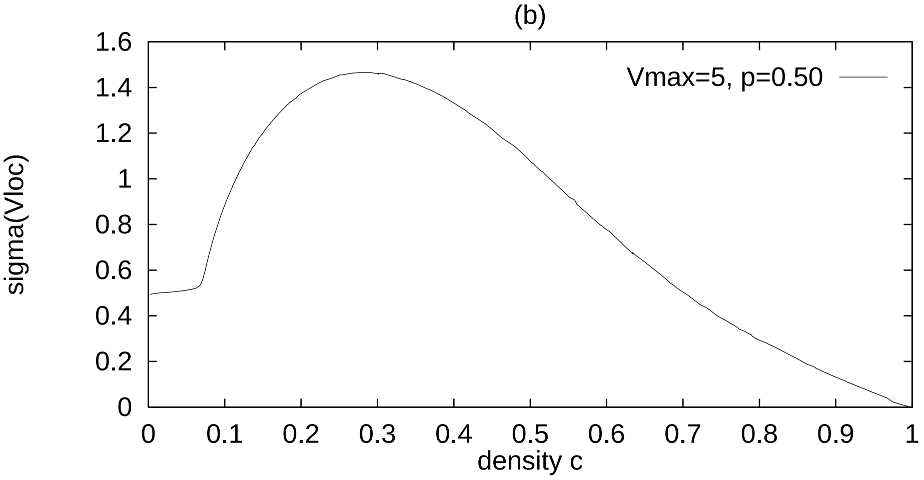


Figure 3.8: Variance of the local velocity.

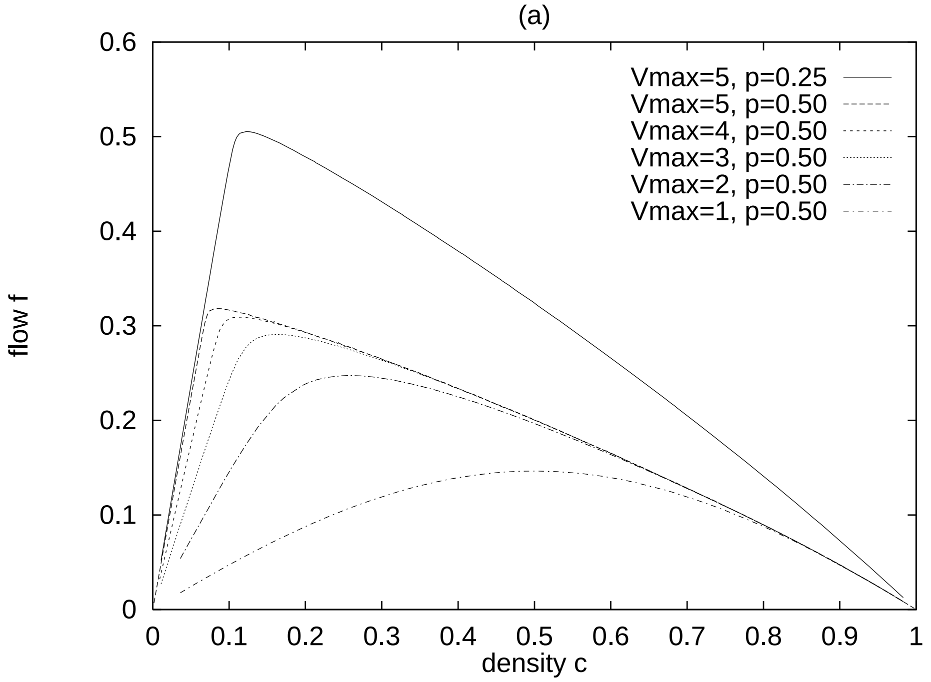


Figure 3.9: Different fundamental diagrams obtained by variations of the two principal parameters of the model p and v_{max} (the latter denoted as `V` in the legends).

curve for a given pair of parameters (v_{max}, p) contains about 100 such data points, which corresponds to about 6 hours on the NEC or 12 hours on the Parsytec (see Chapter 11 for computational details).

The results are summarized in Figs. 3.9, 3.10, and 3.11, where the three standard fundamental diagrams are given for the model for $v_{max} = 1, ..., 5$ (all with $p = 0.5$) and for $p = 0.25$ and $v_{max} = 5$. Clearly, the model may be adapted, by only varying these two parameters, to a wide range of circumstances (e.g., the influence of bad weather, slopes, ...) [63].

3.5 Quantitative comparison with realistic traffic

In this section, some rough arguments concerning the length scale and time scale of the simulation model are made. The easiest approach to calibrate the model is the claim that in a dense jam each car occupies about $7.5\,m$ of space, which is thus the length of one site, in agreement with [151]. Since the average velocity in free traffic of 4.5 sites per time step should correspond to a velocity of about 120 km/h (in Germany), one arrives naturally at a time for one iteration of

$$7.5\,\frac{\mathrm{m}}{\mathrm{site}} \times 4.5\,\frac{\mathrm{sites}}{\mathrm{time\text{-}step}}/(120/3.6)\,\frac{\mathrm{s}}{\mathrm{m}}$$

$\approx$ (1 second per time-step) which again agrees with [151].

A second possibility is to scale the model by the position of the maximum in the fundamental diagram. From traffic measurements, this maximum is found at about $\rho \simeq$ (30 vehicles per lane and kilometer) = (0.225 vehicles/7.5 m), which is by a factor of about 2 higher than the position of the maximum in the scatter-plot for our model. However, this changes when two-lane traffic with slower vehicles is simulated [141].

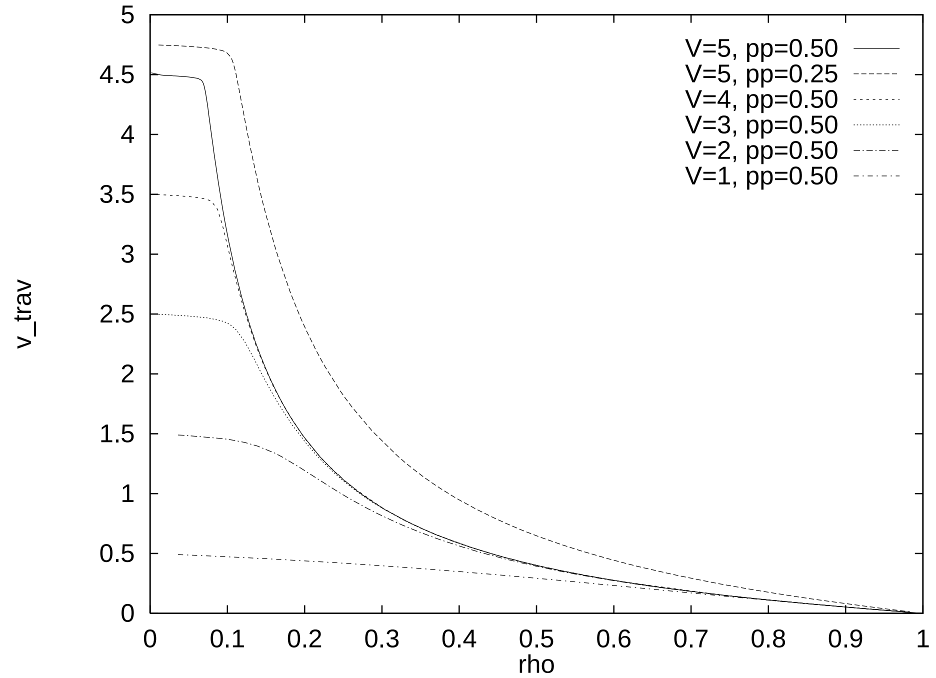


Figure 3.10: Travel velocity v^{trav} as a function of the density. The model very well describes the nearly constant regime for $\rho < \rho(q_{max})$ and the strong decrease of the velocity near maximum flow.

Similarly, freeways have a maximum capacity of about (2000 vehicles per hour and lane) = (0.56 vehicles per second). As our maximum of the flow is only 0.32 vehicles per time step, our model time step should correspond to $0.32/0.56 \approx 0.5$ seconds, thus being by a factor of two lower from the value presented above. However, the definition of "maximum flow" is somewhat unclear. One would think that averaging the reality measurements in Fig. 3.7 in the same way as the simulations would also lead to a much lower flow than 2000 veh/h. In other words: In the simulation, we have an obvious definition for an average maximum flow. It seems however that this definition is not consistent with the more heuristic definition of traffic practitioners.

A fourth possibility for a calibration uses the value of the velocity of the back-travelling start-stop-waves, where a value of about $15\,km/h \approx 4.2\,m/s$ has been measured on freeways. As the maximum capacity of our simulated system is about 0.3 cars per time step, the maximum speed of the back-propagating wave, assuming $\rho = 1$ inside jams, is 0.3 sites ($\approx 2.25\,m$) per time step (i.e., about every third time step a new car arrives at the back of the traffic jam). This would fix one model time step at $2.25/4.2 \approx 0.7$ seconds, thus yielding a value between those of the first and of the third method. However, due to the inexact braking, the density inside jams is smaller than one, thus yielding a higher speed of the backpropagating wave in the model.

In any case, all estimates agree in order of magnitude: One iteration roughly correspond to one second, and one box roughly corresponds to 7.5 m. Since calibration of fundamental diagrams is, as mentioned, different for multi-lane simulations [141], and since the more theoretical results do not depend on the calibration, no further effort for calibration will be done in this work. Calibration for practical applications is the topic of future research.

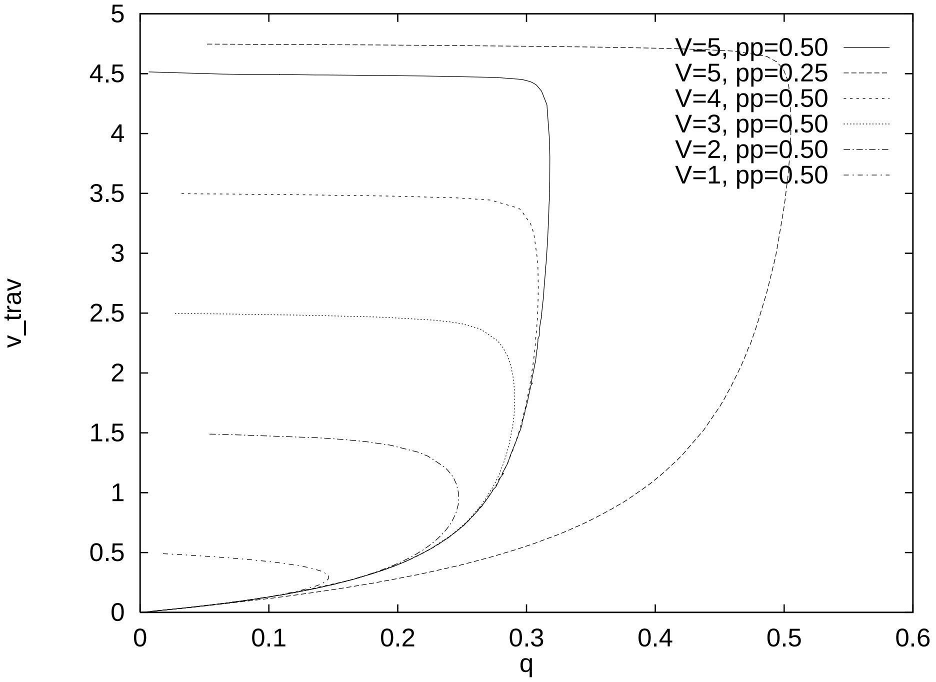


Figure 3.11: Relation between travel velocity v^{trav} and throughput q.

One should note that, contrarily to intuition, the parameters for a discrete traffic flow model are rather fixed once one accepts "one car per lattice site in a jam". The parameters used for the simulations seem relatively reasonable.

Chapter 4

The deterministic limit $p \to 0$

As a first step to understand the model introduced in the last chapter, one can start with its deterministic limit, i.e. $p \to 0$, which simply amounts to taking the randomization rule out of the velocity update. It will turn out that this limit already contains many of the properties of the complete CA model, albeit in a deterministic way. This is not realistic, but easy to work with. Most traffic models used in the Statistical Physics literature so far do not go beyond this model [15, 17, 116, 172, 182], although it is often used in two dimensions [17, 30, 112, 118, 119, 166].

Further results with the deterministic model, for example on different update schemes, and more details of the following can be found in [121].

4.1 Fundamental diagram and phenomenological behavior

The fundamental diagram for the deterministic limit simply consists of two straight lines (Fig. 4.1), which intersect at $\rho_{c,det} = \rho(q_{max}) = 1/6$ and $q_{c,det} = q_{max,det} = 5/6$. They will be explained further down. The intersection point divides two phenomenological regimes: light traffic ($\rho < \rho_{c,det}$) and dense traffic ($\rho > \rho_{c,det}$). A typical situation for light traffic is shown in Fig. 4.2. After starting from a random initial condition, the traffic relaxes to a steady state, where the whole pattern just moves $v_{max} = 5$ positions to the right in each iteration. Cars clearly have a tendency of keeping a *gap* of $\geq v_{max} = 5$ between each other. As a result, the current q in this regime is

$$q_< = \rho \cdot v_{max} \ .$$

The velocity of the kinematic waves in this regime is $c_< = q'_< = v_{max}$. This means that disturbances, such as holes, just move with the traffic, as can also be seen in Fig. 4.2.

Dense traffic is different (Fig. 4.3). Again starting from a random initial configuration, the simulation relaxes to a steady state where the whole pattern moves one position to the *left* in each iteration. Note that cars still move to the right; if one follows the trajectory of one individual vehicle, for this car regions of relatively free movement are alternating with regions of high density and slow speed. Although in a too static way, this captures some of the features of start-stop-traffic. The average speed in the steady state equals the number of empty sites divided by the number of particles: $\langle v \rangle_L = (L-N)/N$; the current is $q_> = \rho \cdot \langle v \rangle_L$, or, with $\rho = N/L$,

$$q_> = 1 - \rho \ .$$

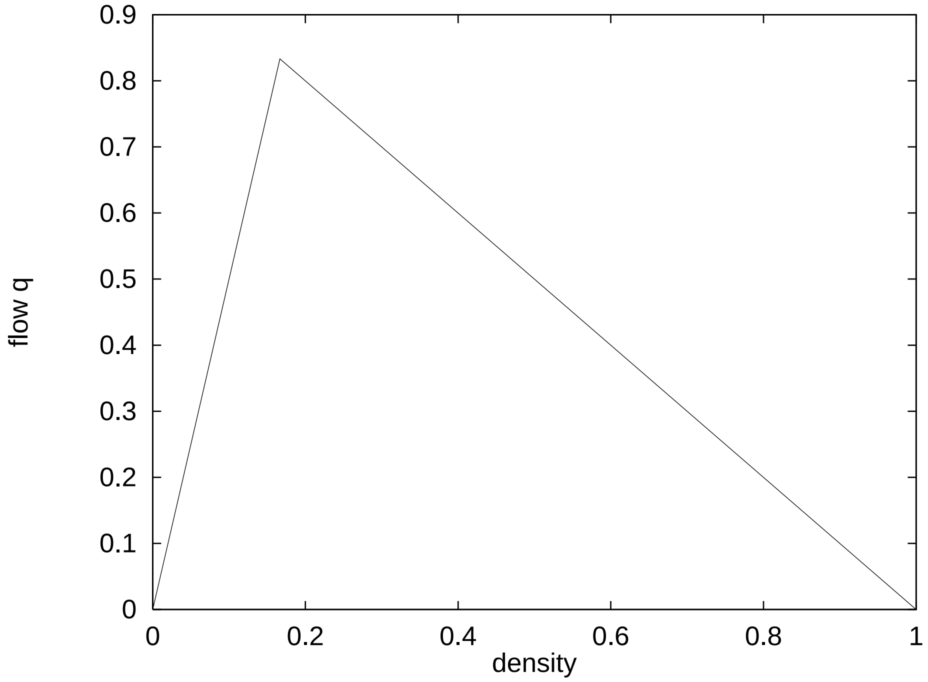


Figure 4.1: Flow-density relation for the deterministic limit.

```
...........5.....1.1.1.............4....5............4.....3..........
................1.1.1.2................5.....5...........4....4.......
.................1.1.2..3...................5.....5..........4....5...
.5................1.2..3...4.....................5.....5.........5....
5.....5............2..3...4....5......................5.....5.........
.....5.....5.........3...4....5.....5......................5.....5....
5.........5.....5.......4....5.....5.....5......................5.....
                              now steady
.....5.........5.....5......5.....5.....5.....5......................5
....5.....5.........5.....5......5.....5.....5.....5..................
.........5.....5.........5.....5......5.....5.....5.....5.............
..............5.....5.........5.....5......5.....5.....5.....5........
...................5.....5.........5.....5......5.....5.....5.....5...
.5......................5.....5.........5.....5......5.....5.....5....
5.....5......................5.....5.........5.....5......5.....5.....
.....5.....5......................5.....5.........5.....5......5.....5
....5.....5.....5......................5.....5.........5.....5......5.
...5.....5.....5.....5......................5.....5.........5.....5...
```

Figure 4.2: Evolution of the deterministic model from random initial conditions for low density ($\rho < \rho(q_{max})$). Note the quick relaxation towards a steady state.

This straight line intersects with the one from light traffic at $\rho_{c,det} = 1/(1 + v_{max})$, which is therefore the density corresponding to maximum throughput.

The velocity of the kinematic waves in the dense regime is $q'_{>} = -1$, which corresponds to the backwards moving pattern in Fig. 4.3.

```
..........1.1.4....03...2..4....04.......4....1.1.4....00..5.....01.1.
...........1.2....01...2..3....01....5.......1.1.2....001.......01.1.2
.3..........2..2..1.2....3...1.1.2........3...1.2..2..01.2......1.1.2.
3...4.........2..1.2..3.....1.1.2..3.........1.2..2..01.2..3.....1.2..
...4....5.......1.2..3...3...1.2..3...4.......2..2..01.2..3...3...2..3
..4....5.....3...2..3...3...1.2..3...4....5.....2..01.2..3...3...2..3.
                             now steady
.4....5.....3...2..3...3...1.2..3...4....5.....2..01.2..3...3...2..3..
4....5.....3...2..3...3...1.2..3...4....5.....2..01.2..3...3...2..3...
....5.....3...2..3...3...1.2..3...4....5.....2..01.2..3...3...2..3...4
...5.....3...2..3...3...1.2..3...4....5.....2..01.2..3...3...2..3...4.
..5.....3...2..3...3...1.2..3...4....5.....2..01.2..3...3...2..3...4..
disturbing here:        x
.5.....3...2..3...3...1.1..3...4....5.....2..01.2..3...3...2..3...4...
5.....3...2..3...3...1.1.2....4....5.....2..01.2..3...3...2..3...4....
.....3...2..3...3...1.1.2..3......5.....2..01.2..3...3...2..3...4....5
....3...2..3...3...1.1.2..3...4........2..01.2..3...3...2..3...4....5.
...3...2..3...3...1.1.2..3...4....5......01.2..3...3...2..3...4....5..
..3...2..3...3...1.1.2..3...4....5.....1.1.2..3...3...2..3...4....5...
                  now steady again, damage time:  5
.3...2..3...3...1.1.2..3...4....5.....1.1.2..3...3...2..3...4....5....
3...2..3...3...1.1.2..3...4....5.....1.1.2..3...3...2..3...4....5.....
```

Figure 4.3: Evolution of the deterministic model from random initial conditions for high density ($\rho > \rho(q_{max})$). Note the quick relaxation towards a steady state, which, though, looks different than for light traffic (Fig. 4.2). The disturbance changes the wave where it happens, and causes a hole moving downstream which is absorbed by the next wave.

4.2 Damage times

In order to estimate the duration of a disturbance, a "damage time" τ_d [159] was measured in the simulation model. To do so, after reaching the steady state, the velocity of one randomly chosen particle is reduced by the smallest possible amount (i.e. by one); the damage time τ_d then is the time the system needs to reach again a new steady state. As explained, the steady state is reached when the entire configuration of the system at time $t+1$ either is a right (v_{max} positions) or a left (1 position) shift of the configuration at time t.

Numerical results for τ_d as a function of the density $\rho = N/L$ are given in Fig. 4.4. At the density $\rho_{c,det} = 1/6$ the damage time τ_d shows a remarkable peak which grows with system size; finite size scaling analysis confirms that it diverges as

$$\tau_d(\rho_c) \propto L \ .$$

The phenomenological reason for the divergence at $\rho_c = 1/6$ for the parallel update is as follows. At first, note that each disturbance can produce both a "hole", which moves with v_{max}, and a "front", which moves backwards as a kinematic wave.

- At low densities, the front travels upstream and is absorbed by a gap; the relevant length scale obviously depends on ρ and not on L. The "hole" will survive forever, thus having no influence on stationarity.
- For high densities (compare Fig. 4.3), the situation is inverse: The new front survives forever, whereas the hole is absorbed by the next wave. The length scale for the wave similarly depends on ρ.
- In between the high and the low density regimes, a transition takes place; and the critical point is a state where particles move with maximum velocity $v_{max} = 5$ and minimum gap

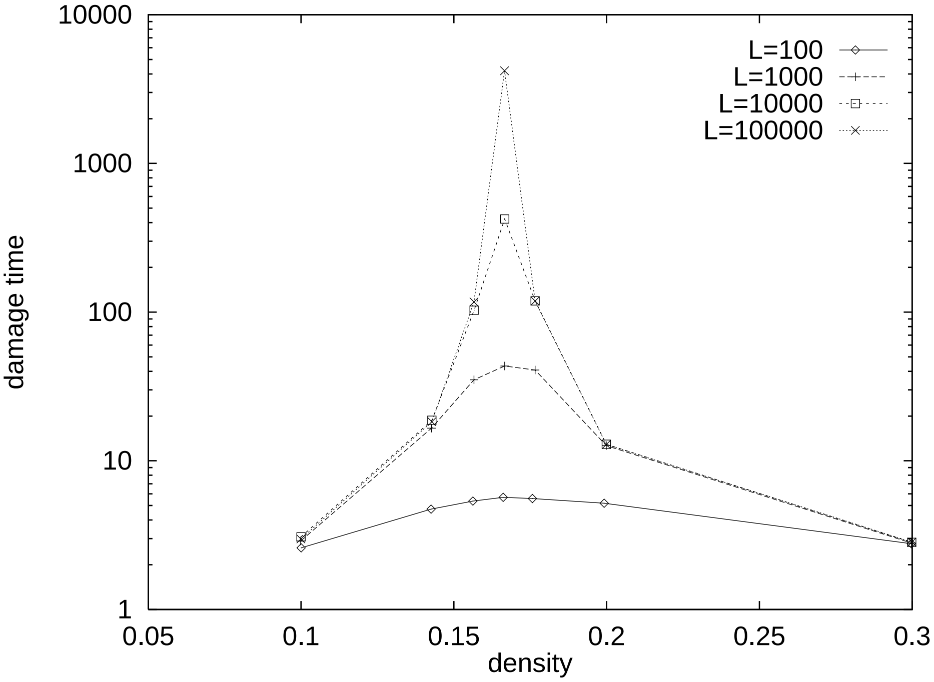


Figure 4.4: Relaxation times after disturbing for traffic in a closed system using the deterministic model, for different densities and different system sizes L. Note that the system size is only relevant for densites near $\rho(q_{max}) = 1/6$.

$gap_{min} = v_{max} = 5$ (and therefore $\rho_c = 1/(gap_{min} + 1) = 1/6$). If one introduces a disturbance into this "critical" state, front and hole move in opposite direction until they meet again (periodic boundary conditions!); hence the L-dependance of τ_d at ρ_c.

At the same time, ρ_c is the density which corresponds to maximum particle throughput.

4.3 Self-organization of the critical state

If one assumes a very dense jam as in Fig. 4.5, then the outflow from this jam assumes exactly the critical configuration of particles moving at maximum speed 5 and with $gap = 5$. This is no longer true for an open boundary which is fixed in space (Fig. 4.6); but it can be restored, e.g., by forcing a higher acceleration for the first particle (Fig. 4.7). In this last case, a disturbance (as in Fig. 4.7) causes a front which travels with constant speed to the left boundary, whereas the hole travels with constant speed to the right boundary. Conceptually, this case of self-organized criticality therefore belongs to Bak, Tank, and Wiesenfeld's one-dimensional sandpile model [8], with the difference that our model selects a state of non-trivial density (i.e. $\rho_c \neq 0$).

With respect to reality, this means that the outflow of the traffic jam adjusts itself exactly at the critical density and therefore at maximum capacity, and due to the criticality, new disturbances have long-ranged effects through density waves traveling with constant speed. This is no longer true for a bottleneck situation (e.g. a two-lane directional road merging into only one lane): The open boundary described in Fig. 4.6 leads to lower density and throughput, and the same is true similarly for the full CA model (see later) and for reality.

```
00000000000000000000000001.2..3...4....5.....5.....5.....5.....5.....5
0000000000000000000000001.2..3...4....5.....5.....5.....5.....5.....5.
000000000000000000000001.2..3...4....5.....5.....5.....5.....5.....5..
00000000000000000000001.2..3...4....5.....5.....5.....5.....5.....5...
0000000000000000000001.2..3...4....5.....5.....5.....5.....5.....5....
000000000000000000001.2..3...4....5.....5.....5.....5.....5.....5.....
00000000000000000001.2..3...4....5.....5.....5.....5.....5.....5.....5
0000000000000000001.2..3...4....5.....5.....5.....5.....5.....5.....5.
```

Figure 4.5: Deterministic outflow from a traffic jam.

```
02....4........5.........5.........5.........5.........5.........5....
1..3......5.........5.........5.........5.........5.........5.........
02....4........5.........5.........5.........5.........5.........5....
1..3......5.........5.........5.........5.........5.........5.........
02....4........5.........5.........5.........5.........5.........5....
1..3......5.........5.........5.........5.........5.........5.........
02....4........5.........5.........5.........5.........5.........5....
1..3......5.........5.........5.........5.........5.........5.........
02....4........5.........5.........5.........5.........5.........5....
```

Figure 4.6: Acceleration from a dense situation (open system). Particles are inserted with maximum speed at the left boundary whenever there is space, but due to the other particles ahead they are immediately slowed down. In consequence, the system is not able to reach its maximum throughput.

4.4 Summary

Already a deterministic CA model for traffic, which is a subcase of the full model, shows some features which are realistic for traffic. Light traffic is characterized by holes moving with the traffic; dense traffic is characterized by kinematic jam waves moving backwards against the traffic. The regimes are separated by a relatively obvious phase transition, showing diverging relaxation times after disturbances.

This transition point gets additional importance because it is automatically selected as output from a jam.

As will be shown in the next chapters, this model already captures some of the features of traffic jam dynamics. It has, though, the property that outflow traffic is maximally dense, in the sense that at any higher density, cars could no longer move at maximum speed. Yet, outflow traffic which is not “maximally” dense is the additional necessary ingredient for traffic jams becoming the singular clusters which they are in traffic measurements (Fig. 3.1).

```
05.....5.....5.....5.....5.....5.....5.....5.....5.....5.....5.....5..
5.....5.....5.....5.....5.....5.....5.....5.....5.....5.....5.....5...
4....5.....5.....5.....5.....5.....5.....5.....5.....5.....5....
3...5.....5.....5.....5.....5.....5.....5.....5.....5.....5.....
         x   (disturbance)
2..5.....4.....5.....5.....5.....5.....5.....5.....5.....5.....5
1.5.....4....5......5.....5.....5.....5.....5.....5.....5.....5.
05.....4....5.....5......5.....5.....5.....5.....5.....5.....5..
5.....4....5.....5.....5......5.....5.....5.....5.....5.....5...
4....4....5.....5.....5.....5......5.....5.....5.....5.....5....
3...4....5.....5.....5.....5.....5......5.....5.....5.....5.....
```

Figure 4.7: Acceleration from a dense situation (open system) with a forced acceleration of the leftmost particle. This means that at this position, a particle instantly can accelerate from zero to, e.g., the maximum speed five (see, e.g., from the first to the second line).

Chapter 5

Cruise control limit and self-organized criticality

Already in Chapter 3 it became clear that start-stop-waves play an important role, as well in the model behavior as in reality. Chapter 4 then established that dense traffic can be distinguished from light traffic by the direction of the wave speed, and that the transition point between the regimes is the point of maximum flow.

This chapter now continues to work out the relation between the waves and the flow maximum. It actually starts with observations of the full model of Chapter 3, but then moves to a "cruise control limit", where the picture is clearer. Nevertheless, this cruise control limit correctly captures the main features of the dynamics of the full model. The exact relation between the cruise control limit and the full model will then be described in Chapters 6 & 7.

5.1 Density waves

Contrary to the deterministic model of the last chapter, the original model reaches its much lower maximum throughput $q_{max} = 0.318 \pm 0.001$ already at a density of $\rho(q_{max}) = 0.086 \pm 0.002$ (Fig. 3.6). What is the deeper reason behind this capacity limitation?

As a first step, one can look at space-time-plots of systems slightly below and above the threshold density $\rho(q_{max})$ (first row of Fig. 5.1). As before, in these pictures horizontal lines are configurations at consecutive time steps, time evolving downwards. Black pixels stand for occupied sites. Vehicles are moving from left to right, and by following the pixels, one can discern the trajectories of the vehicles.

These pictures show marked shock waves, and they occur more often for the higher density. The waves form at arbitrary times and positions due to a "bad" superposition of the disturbances caused by the random element in the velocity update. They are clearly visible as clusters of cars of low velocity, with more interior structure inside the clusters. Once such a disturbance has formed, it is maintained as long as there are more vehicles arriving at the end of the queue than vehicles leaving the queue at its head. These disturbances appear already well *below* the regime of maximum traffic flow, but they are rare. They start to dominate the system's appearance at densities above the regime of maximum flow. This leads to the idea that the regime of maximum traffic flow might be reached when there are, for the first time, waves with a "very long" lifetime, similar to a percolation transition [163]. See also Ref. [167] for a similar analysis of a deterministic model.

Figure 5.1: (Previous page) Plots of simulated single lane freeway traffic in the space-time-domain with resolutions (a) 1:1, (b) 1:4, (c) 1:16. *Left column:* Density $\rho = 0.07$, slightly below the regime of maximum flow. *Right column:* Density $\rho = 0.1$, slightly above the regime of maximum flow. Each black pixel corresponds to a site occupied by a vehicle at a certain place (x-direction) and at a certain time (y-direction). A trajectory of an undisturbed vehicle goes therefore diagonally downwards and to the right. The pictures of the first row cover 500 sites and 500 time steps. The pictures of each row are contained (as indicated by the boxes) in the pictures of the row underneath.

To get a better overview, the second and third row of Fig. 5.1 show the same system at lower resolutions obtained by averaging, therefore showing a larger part of the system and more time steps. A striking feature of these pictures is that they look in some way self-similar [41, 106], i.e., large jams are composed of many smaller ones which look like large ones at a higher resolution. This supports the idea of a critical transition at maximum flow.

5.2 The cruise control limit

To analyse these jam-clusters, it is extremely helpful to construct a system where at most one such cluster exists. This is achieved by taking the "cruise control limit" of the model: Fluctuations at free driving are set to zero. The resulting model reads as follows: For every configuration of the model, one iteration consists of the following steps, which are each performed simultaneously for all vehicles (the quantity gap again equals the number of empty sites in front of a vehicle):

- A vehicle is stationary when it travels at maximum velocity v_{max} and has free headway ($gap \geq v_{max}$). Such a vehicle just maintains its velocity.
- If a vehicle is not stationary, it is jammed. The following two rules are applied to jammed vehicles:
  - **Acceleration of free vehicles:** If a vehicle has $gap \geq v + 1$, then: With probability 1/2, it accelerates to $v + 1$, otherwise it keeps the velocity v.
  - **Slowing down due to other cars:** Each vehicle with $gap \leq v$ slows down to gap: $v := gap$. With probability 1/2, it over-reacts and slows down even further: $v := \max[gap - 1, 0]$.
- **Movement:** Each vehicle advances v sites.

For clarity, a formal version of the velocity update is given in the appendix.

While in the original model (Chapter 3), vehicles at v_{max} slow down randomly with probability $p_{free} = p$, here only the jammed vehicles move nondeterministically. This corresponds to the $p_{free} \to 0$ limit, or the "cruise control limit", of the original model and completely separates the time scales for driving (i.e. perturbing) the system and the system's response.

The fundamental diagram, or current-density relation, $q(\rho)$, was determined numerically as shown in Fig. 5.2 for a closed system of size $L = 30\,000$. Starting with a random initial condition with N cars (i.e. $\rho = N/L$) and after discarding a transient period of $5 \cdot 10^5$ iterations, we measured $\langle q \rangle_L(t) = \sum_{i=1}^{N} v/L$ every 2500 time-steps up to the $3 \cdot 10^6$th iteration. Each data point corresponds to the average over current measurements for a single initial condition, with the following exception: When a run becomes stationary (i.e. no more jammed cars in the sense of the definition above), then the future behavior is predictable. In this case, the run is stopped, and the current will be equal to $q_{det} = v_{max} \cdot \rho$, see below.

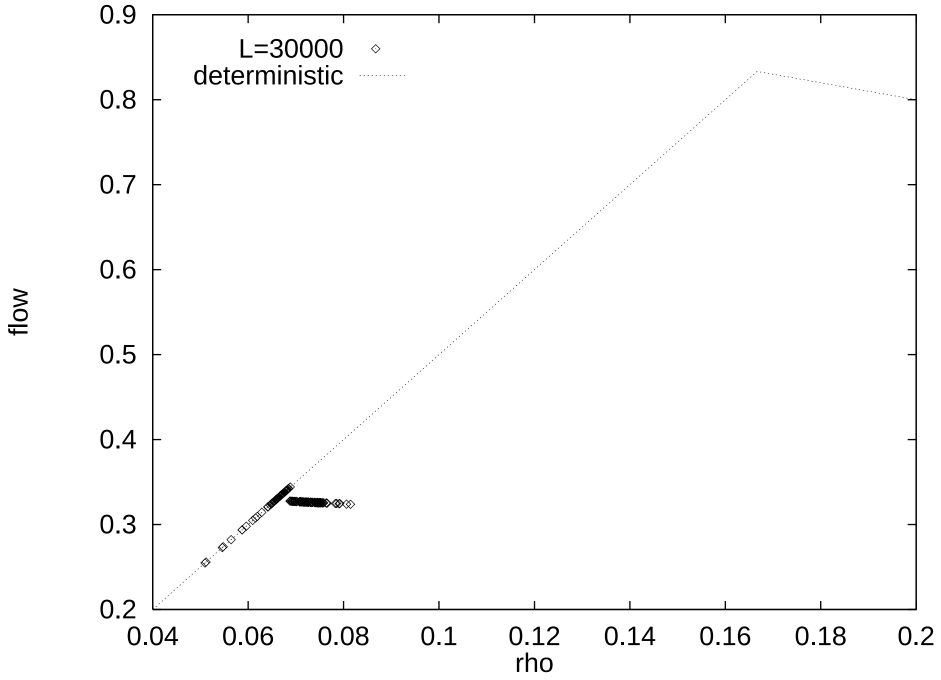


Figure 5.2: The fundamental diagram, $q(\rho)$, for $v_{max} = 5$. The dotted line is valid for deterministic traffic, i.e. when the initial state is prepared such that for each car $gap > v_{max}$ and $v = v_{max}$. The points are measurement results starting from random initial conditions; each point corresponds to one run of a closed system of length $L = 30\,000$ and an average over $2.5 \cdot 10^6$ iterations. When a run relaxed to the deterministic state (no more jammed cars), it was stopped and the deterministic current was taken as the result (points on the dotted line).

For a spatially infinite system, the following results hold: For $\rho < \rho_c$, jams present in the initial configuration are eventually sorted out and the stationary deterministic state is jam free with every vehicle moving at maximum velocity. Thus in the lamellar regime the current is a linear function of density with slope $v_{max} = 5$. Lamellar behavior is observed up to a maximum current $q_c(\rho_c)$. For $\rho > \rho_c$, and $\rho < \rho_{det,max}$ (defined below) the system is bistable. Starting from an initial configuration which has many jams, the jams in this case are never sorted out (infinite system size). The steady state is an inhomogeneous mixture of jam free regions and higher density jammed regions. Clearly, these jammed regions decrease the average current in the system. It is possible, nevertheless, to prepare initial configurations that have no jams. Since all motion is deterministic in this state, the steady state will also have no jams and the current will still be a linear increasing function of ρ (the dotted line in Fig. 5.2). This is possible up to densities of

$$\rho_{det,max} = \frac{1}{v_{max} + 1} ,$$

leading to a maximum current of

$$q_{det,max} = \frac{v_{max}}{v_{max} + 1} .$$

This clearly is much higher than the current q_c for random initial conditions. It is in this sense that our system is bi-stable (cf. also [148, 167]). This effect allows us to produce outflows with densities above ρ_c.

Above ρ_c, the current-density relation can be derived by assuming that the system phase separates into jammed regions separated by jam free gaps. The jam free gaps are the outflow of a jam and thus have current $q_c(\rho_c)$, as argued in the next section. Conservation of the number of cars and of volume leads to $N_f + N_j = N$, $L_f + L_j = L$, $N_j/L_j = 1/a$, $N_f/L_f = \rho_c$, where N_f,L_f, and ρ_c are the number, length, and density, respectively, in the free flowing phase, and N_j, L_j, and $1/a$ are the corresponding quantities in the jammed phase. After some algebra one obtains

$$q = q_c - \frac{(\rho - \rho_c)(aq_c - v_j)}{1 - a\rho_c} \quad ,$$

where a is the average number of lattice sites per jammed vehicle, and v_j is the average velocity ($< v_{max}$) of a jammed vehicle (see [10] for a similar calculation). Thus, the current-density relation is linear both above and below the critical point, as demonstrated in Fig. 5.2.

The discontinuity in the current at the critical point, as seen in the figure, is a finite size effect due to the fact that each point in the figure represents a single initial configuration. In a finite system, there is a finite probability that even a system with supercritical density $\rho > \rho_c$ finds the deterministic state, and then has a current of $q_{det} > q_c$.

5.2.1 The outflow from a jam occurs at maximum throughput

A striking feature of the model is that maximum throughput is selected automatically when the left boundary condition is an infinitely large jam and the right boundary is open. An intuitive explanation is that maximum throughput cannot be any higher than the intrinsic flow rate out of a jam, because then jams become stable in the long time limit and reduce the overall current. By definition, of course, maximum throughput cannot be lower than this intrinsic flow rate.

In Fig. 5.3, the cars on the left flow out from a region of high density where they move with zero velocity. This high density region is not plotted here; only the interface or front separating the high density region and its deterministic outflow is plotted. This is the branched structure on the left hand side of the figure. The vehicles flowing out of the large jam ultimately relax to the deterministic state when they have moved sufficiently far away from the jam.

This feature of maximum throughput selection is a general feature of driven diffusive systems [21, 34, 86]. However, in our case the left boundary condition is unusual: if the left boundary is fixed in space as in [21, 34, 86], then the outflow from a jam cannot reach maximum throughput (cf. bottleneck situation in the last and in the next chapter).

5.2.2 Traffic jams in the outflow show self-organized criticality

The outflow situation, as described above, produces deterministic flow asymptotically at large distances. This means that sufficiently far downstream from the large jam, the jam flow has sorted itself out into deterministic flow. In the deterministic region, one car is randomly perturbed by reducing its velocity to zero. Many different choices for the local perturbation, however, give rise to the same large scale behavior. The perturbed car eventually re-accelerates to maximum velocity. In the meantime, though, a following car may have come too close to the disturbed car and has to slow down. This initiates a chain reaction – the emergent traffic jam.

Fig. 5.3 shows the first 1400 time steps of such an emergent jam (the right jam in the figure). Qualitatively, the jam clearly shows a tendency to branch with complex internal structure and a fractal appearance. The emergent traffic jams drift backwards; so it is possible for a sufficiently long lived emergent jam to eventually intersect with the outflow jam interface that is itself becoming broader

Figure 5.3: Outflow from a dense region (left); this outflow relaxes precisely to the critical density. In the outflow region, a jam is triggered by a small disturbance. Note the branching behavior of this jam. — "Deterministic" vehicles to the right of the jam are not plotted.

with time. It is likely that the branching behavior of the emergent jams is the same as the branching behavior of the original jam interface. In this work, however, we do not explicitly study the interface. Contrary to the figure, in the computer code, the interface region to the left and the jam to the right are kept completely separate using methods described in the appendix.

A jam is sorted out when the number of jammed cars is zero. This defines the lifetime, t, of an emergent traffic jam. In order to obtain statistics for the distribution of traffic jam lifetimes, $P(t)$, for example, the deterministic outflow is then disturbed again. Other cluster statistics such as the spatial extent w of the jam or overall space-time size s (mass) of the jam are obtained as well. These distributions are analogous to similar distributions for other branching processes such as directed percolation [49], branching annihilating random walks [76], or in nonequilibrium lattice models [75] although the precise behaviors are different.

Fig. 5.4 shows 1400 time steps in the middle of the life of a larger jam. Here, vehicles that are stationary are no longer shown; the plot only shows the "particles", or jammed vehicles, that propagate the disturbance.

For a quantitative treatment, we start by measuring the probability distribution of jams as a function of their lifetime t. Fig. 5.5 shows that for $t >\sim 100$ this distribution follows a power law

$$P(t) \sim t^{-(\delta+1)} \qquad \text{with} \qquad (\delta+1) = 1.5 \pm 0.01 \ ,$$

very close to $\delta = 1/2$. Figure 5.5 represents averaged results of more than 60 000 avalanches.

Here scaling is observed over almost four orders of magnitude as determined by our *numerically imposed* cutoff: For this figure, if jams survive longer than 10^6 time steps, they are removed from the data base. It is very important to note that these emergent jams are precisely critical. Their power law scaling persists up to any arbitrarily large numerically imposed cutoff. The lifetime distribution is related to the survival probability $P_{surv}(t)$ by

$$P_{surv}(t) = \int_t^\infty \mathrm{d}t' P(t') \sim t^{-\delta} \qquad \text{for} \qquad \delta > 0 \ .$$

We again emphasize that no external tuning is necessary to observe this scaling behavior. The outflow from the infinite jam self-organizes to the critical state.

5.3 Random walk arguments

Paczuski [123] has developed a phenomenological theory for the branching waves. The arguments are given in this and the next section.

It is, perhaps, surprising that such a seemingly complicated structure as shown in Fig. 5.3 is described by such a simple apparent exponent. Numerically, the exponent $\delta + 1$ is conspicuously close to 3/2, the first return time exponent for a one-dimensional random walk. In fact, for $v_{max} = 1$ this random walk picture is exact, as shown below.

Let us consider a single jam in a large system with $v_{max} = 1$. The vehicles in the jam form a queue, and all of these cars have velocity zero. When the vehicle at the front of the jam accelerates to velocity one, it leaves the jam forever. The rate at which vehicles leave the jam is determined by the probabilistic rule for acceleration. Vehicles, of course, can be added to the jam at the back end. These vehicles come in at a rate which depends on the density and velocity of cars behind the jam. Given the rules for deceleration, the spacing between the jammed cars is zero so that the number of cars in

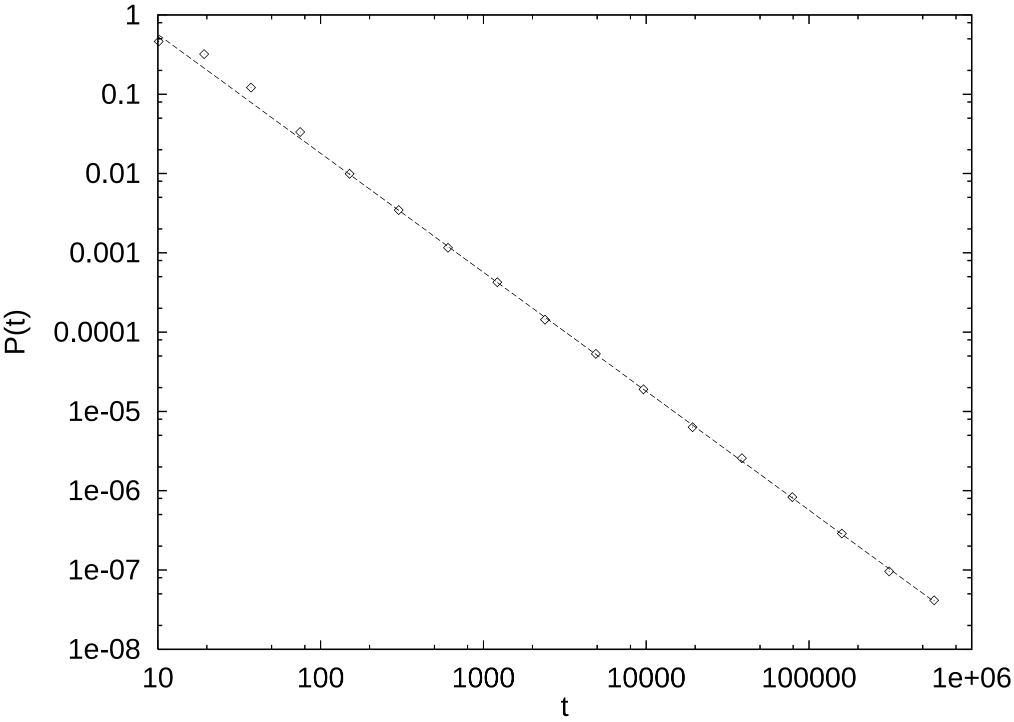


Figure 5.5: Lifetime distribution $P(t)$ for emergent jams in the outflow region; average over more than 65 000 clusters (avalanches). The straight line has slope 3/2. Without external tuning, the lifetime distribution is precisely critical up to the numerically imposed cut-off at $t = 10^6$.

the jam, n, is equal to the spatial extent of the jam, w. This contrasts with the branching behavior for $v_{max} > 1$. The probability distribution, $P(n,t)$, for the number of cars in the jam n at time t is determined by the following equation:

$$P(n,t+1) = (1 - r_{in} - r_{out})P(n,t) + r_{in}P(n-1,t) + r_{out}P(n+1,t) \quad . \qquad (*) \qquad (5.1)$$

Here, the quantities r_{in} and r_{out} are phenomenological parameters that depend on the density behind the jam and the rate at which cars leave a jam. They are independent of the number of cars in the jam. For large n and t, one can take the continuum limit of Eqn. $(*)$ and expand to lowest order

$$\frac{\partial P}{\partial t} = (r_{out} - r_{in})\frac{\partial P}{\partial n} + \frac{r_{out} + r_{in}}{2}\frac{\partial^2 P}{\partial n^2} \quad . \qquad (5.2)$$

When the density behind the jam is such that the rate of cars entering the jam is equal to the intrinsic rate that cars leave the jam, then the first term on the right hand side vanishes. Then, the jam queue is formally equivalent to an unbiased random walk in one dimension [42]. The first return time of the walk then corresponds to the lifetime of a jam. This leads immediately to the result $P(t) \sim t^{-3/2}$ for the lifetime distribution.

This argument shows that the outflow from an infinite jam is in fact self-organized critical. This can be seen immediately by noting that the outflow from a large jam occurs at the same rate as the outflow from an emergent jam created by a perturbation. This also shows that maximum throughput corresponds to the percolative transition for the traffic jams. Starting from random initial conditions in a closed system, the current at long times is determined by the outflow of the longest-lived jam in the system.

When $r_{in} = r_{out}$, one also finds from Eqn. $(*)$ that $n \sim t^{1/2}$ and the size of the jam $s \sim nt \sim t^{3/2}$. If the density in the deterministic state is below the critical density ρ_c, then the jams will have a characteristic lifetime, t_{co}, size s_{co}, number n_{co}, etc. From Eqn. $(*)$, $t_{co} \sim n_{co}(r_{out} - r_{in})^{-1}$. Assuming that near the critical point $r_{out} - r_{in} \sim \rho_c - \rho$, then using $n_{co} \sim t_{co}^{1/2}$ leads to

$$t_{co} \sim (\rho_c - \rho)^{-2} \quad .$$

For $\rho > \rho_c$, vehicles on average enter the jam at a faster rate than they leave. In this case, there is a finite probability to have an infinite jam, P_∞, which vanishes as $\rho \rightarrow \rho_c$ as

$$P_\infty \sim (\rho - \rho_c)^{\tilde{\beta}} \quad .$$

The steady-state density of jammed cars, $\rho_j = (\rho - \rho_c)^1$, so that the order parameter exponent $\tilde{\beta} = 1$.

From the random walk Eqn. $(*)$, and in analogy with other branching processes such as directed percolation [49], P_{surv} follows a scaling form

$$P_{surv}(t, \Delta) \sim t^{-\delta} f(t \, \Delta^{\nu_t}) \ ,$$

near the critical point. Here $\Delta \equiv |\rho - \rho_c|$ and $t_{co} \sim \Delta^{-\nu_t}$. From this scaling relation, $\tilde{\beta} = \delta \nu_t$. For $v_{max} = 1$, $\delta = 1/2$, $\nu_t = 2$, and $\tilde{\beta} = \beta = 1$.

The number of jammed vehicles, $\bar{n}$, averaged over all jams, including those that die out, has the scaling form

$$\bar{n} \sim t^{\eta} g(t \, \Delta^{\nu_t}) \ .$$

The number of jammed vehicles averaged over surviving jams, scales with a different exponent

$$n(t) = \bar{n}(t)/P(t) \sim t^{\eta+\delta} \ .$$

The mapping to the random walk gives $\eta = 0$.

The cluster width, averaged over surviving clusters, scales as $\bar{w} \sim t^{1/z}$, and the mapping to the random walk gives $z = 2$. The average cluster size $\bar{s} \sim t^{\eta+\delta+1}$; $\bar{s} \sim t^{3/2}$ in the random walk case.

In the numerical measurements, we averaged the quantities t = lifetime of the cluster, w = maximum width of cluster during cluster life, n = maximum number of simultaneously jammed vehicles during cluster life, s = total number of jammed vehicles during cluster life.

Our theoretical results should describe the emergent traffic jams not only at $v_{max} = 1$ but also for any $v_{max} > 1$ as long as the traffic jam itself remains dense. If this is the case, then the dynamical evolution is determined solely by the balance of incoming and outgoing vehicles as described by Eqn. $(*)$. The ratio $\bar{w}/n$ should go to a finite constant at large times if the theory is valid. If the emergent jams break up into a fractal structure, and $\bar{w}/n$ diverges, internal dynamics must also be included. Since the jams displayed in Figs. 5.3 and 5.4 appears branched and at least qualitatively fractal, one might doubt that such a simple theory could describe this behavior. Nevertheless, the close numerical agreement of the lifetime distribution exponent for the SOC behavior suggests the possibility that the random walk theory is a valid description of the branching jam waves.

5.4 A cascade equation for the branching jams

We now analyze the branching behavior of jams with $v_{max} > 1$ in terms of a phenomological cascade equation. A very large emergent jam, at a fixed point in time, consists of small dense regions of jammed cars, which we call subjams, separated by intervals, "holes", where all cars move at maximum velocity. If the jam is dense, then the holes have a finite average size. Otherwise, the jammed vehicles comprise a fractal with dimension $d_f < 1$. We will consider the subjams to have size one.

Holes between the subjams are created at small scales by the probabilistic rules for acceleration. Each subjam can create small holes in front of it. We will ignore the details of the injection mechanism, and assume that there is a steady rate at which small holes are created in the interior of a very long lived jam. We also assume that the interior region of a long-lived jam reaches a steady state distribution of hole sizes.

In order to determine the asymptotic scaling of the large holes in the interior of a long-lived jam, it is necessary to isolate the dominant mechanism in the cascade process for hole generation. The microscopic mechanism that connects holes at different scales is the dissolution of one subjam. When one subjam dissolves because the cars in it accelerate to maximum velocity, the two holes on either side of it merge to form one larger hole. Holes at any large scale are created and destroyed by this same process. In the steady state, the creation and destruction of large holes must balance. This leads to a cascade equation for holes of size x:

$$\sum_{u=x+1} < h(x)h(u-x) >= \sum_{x'=1}^{x-2} < h(x')h(x-x'-1) > \quad . \qquad (**)$$

Here, the angular brackets denote an ensemble average over all holes in the jam, and the quantity $h(x)h(u-x)$ denotes a configuration where a hole of size x follows a hole of size $u-x$. The right hand side of this equation represents the rate at which holes of size x are created, and the left hand side represents the rate at which holes of size x are destroyed.

Now, we make an additional ansatz; namely, for large x, $< h(x')h(x-x'-1) >= G(x)$, independent of x' to leading order. That is, to leading order the probability to have two adjacent holes, whose sizes sum to x is independent of the size of either hole. $G(x)$ then also scales the same as $P_h(x)$, the probability to have a hole of size x. Thus Eqn. $(**)$, to leading order can be written

$$xG(x) \sim \sum_{u=x} G(u) \quad .$$

Differentiating leads to

$$x\frac{\partial G(x)}{\partial x} = -2G(x) \quad ; \quad G(x) \sim \frac{1}{x^2} \quad .$$

Thus the distribution of hole sizes decays as

$$P_h(x) \sim x^{-\tau_h} \quad ; \quad \mathrm{with} \tau_h = 2 \quad .$$

It is interesting to note that the cascade equation $(**)$ is identical to the dominant mechanism in the exact cascade equation for forests in the one-dimensional forest fire model [128]. In addition, the result $\tau_h = 2$ equals the distribution exponent for the forests, which has been obtained exactly [35, 128].

The exponent τ_h is related to the fractal dimension d_f of jammed vehicles by

$$\tau_h = 1 + d_f \ ,$$

as long as $\tau_h \leq 2$ [107]. Thus, $\tau_h < 2$ implies that the equal time cut of the jam clusters is fractal, otherwise not. The point $\tau_h = 2$ is the boundary between fractal and dense behavior. At this special point, the random walk theory can still be expected to apply, although with logarithmic corrections.

The width of an emergent jam, at a given point in time, $w(t)$, can be expressed as

$$w(t) = \frac{n(t)}{w_j} \left(w_j + \int \mathrm{d}x \, x \, P_h(x,t) \right) \ .$$

Here, w_j is the average width of a subjam, it is $\mathcal{O}(1)$. The quantity $P_h(x,t)$ is the probability distribution to have a hole of size x in a jam that has survived to time t. It is natural to assume that this distribution corresponds to $P_h(x)$ up to a cutoff which grows with t. Inserting the expression for $P_h(x)$ gives

$$w(t) \sim n(t) \left(1 + \int_1^{x_h^*} \mathrm{d}x \, x^{1-\tau_h} \right) \ ,$$

where the upper bound x_h^* represents a time-dependent cutoff. Using $\tau_h = 2$, $n \sim t^{\delta+\eta}$, and assuming $x_h^* \sim t^c$ gives

$$w(t) \sim t^{\delta+\eta} \left(1 + c \log t\right) \qquad \text{for } \tau_h = 2 \quad .$$

In other words, if $\tau_h = 2$, as the above arguments suggest, spatial quantities such as $w(t)$ will exhibit logarithmic corrections to the random walk results. In the following section, we test these theoretical predictions with further numerical studies.

5.5 Simulation results

In this section, we present the rest of our numerical results. Unless otherwise noted, these results were obtained for systems with $v_{max} = 5$.

5.5.1 At the self-organized critical point

We study the critical properties of the outflow of a large jam by driving it with slow random perturbations as described in Sec. II. Numerically, we find (Fig. 5.6)

$$n(t) \equiv \langle \, n \, \rangle_{surv}(t) \sim t^{\eta+\delta} \qquad \eta + \delta = 0.5 \pm 0.1$$

and (Fig. 5.7)

$$s(t) \sim n(t) t \sim t^{1+\eta+\delta} \qquad 1 + \eta + \delta = 1.5 \pm 0.1$$

in agreement with the random walk predictions. However, the simulations do not converge to power law scaling before $t \simeq 3 \cdot 10^4$, and since the simulation is cut off at $t = 10^6$, the exponents are obtained from less than two orders of magnitude in t. Figs. 5.6 and 5.7 contain the averaged results of more than 160 000 avalanches, typically corresponding to approximately 200 workstation hours (see Appendix and figure captions for further information).

5.5.2 Off criticality

By changing the left boundary condition (i.e. the inflow condition) of the open system, simulations were performed both above and below the critical point. This is achieved by replacing the mega jam by the following mechanism: Vehicles are inserted at a fixed left boundary so that in all cases v_{max}

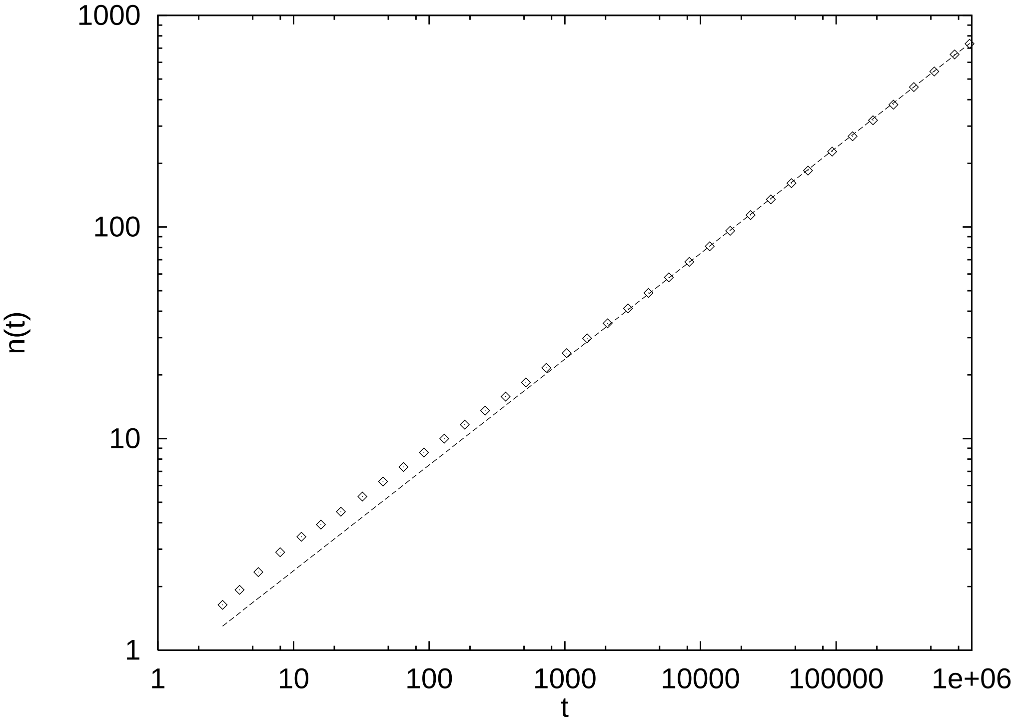


Figure 5.6: Number of jammed particles at time t, $n(t)$, averaged over surviving clusters, in the outflow situation. More than 165 000 clusters were simulated. The straight line has slope 1/2. Note that the measurement only converges for $t >\sim 5000$.

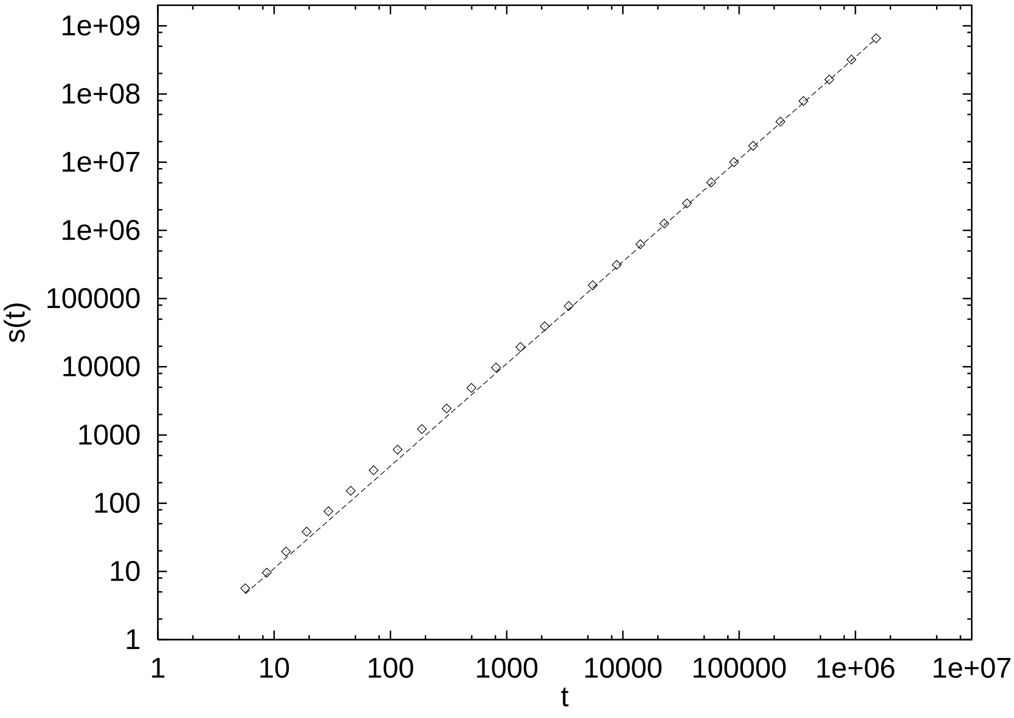


Figure 5.7: Mass of jam in space-time, $s(T)$, in the outflow situation, for the same clusters os in Fig. 5.6. Jams of similar lifetime T were averaged. The straight line has slope 3/2. Note that the measurement again only converges for $t >\sim 5000$.

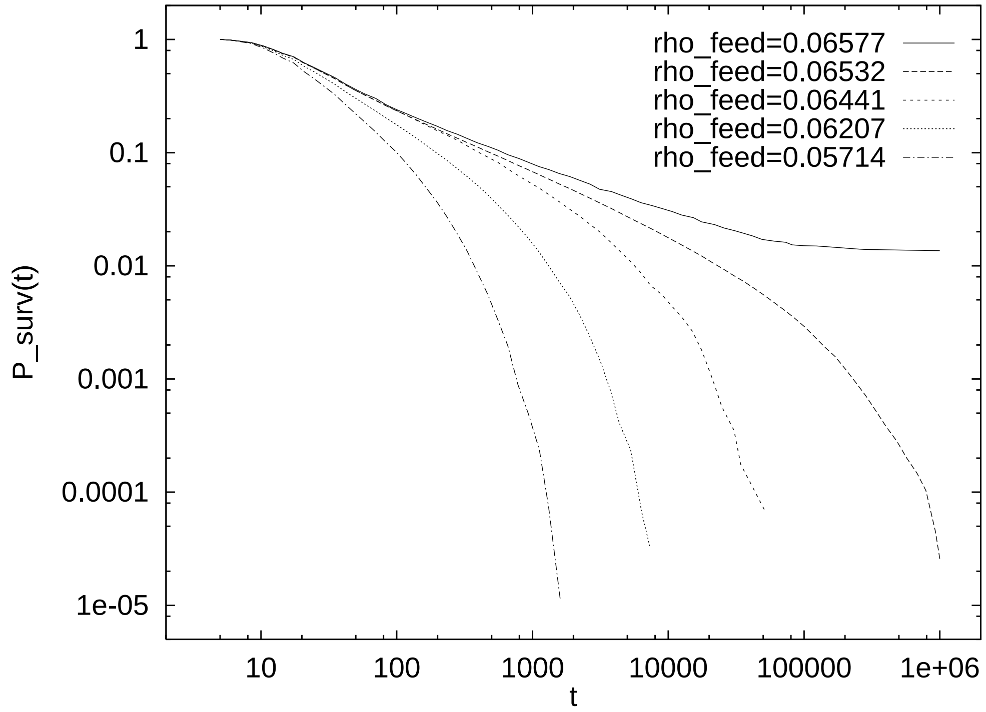


Figure 5.8: Survival probability for jam-clusters, $P_{surv}(t)$, for different inflows. This picture corresponds exactly to the theoretical picture, cf. Fig. 2.1. Note that this distribution is highly sensitive to the inflow, re-confirming that the outflow is indeed precisely critical.

sites are left empty and then the following sites are attempted to be occupied with probability p_{insert} until a site is occupied. The rate p_{insert} determines an average density ρ by

$$\rho = \frac{1}{v_{max} + 1/p_{insert}} \ ,$$

which can go as high as $\rho = \rho_{det,max} = 1/6 = 0.16666\ldots$ for $v_{max} = 5$, much higher than the critical density of $\rho_c \approx 0.0702$.

We have performed data collapse for the survival probability $P_{surv}(t)$ on varying the density, as shown in Fig. 5.9. By plotting $P_{surv}/t^{-\delta}$ vs. $t\,\Delta^{\nu_t}$ with the exponents $\delta = 0.5$, $\nu_t = 2$ was determined by the qualitatively best collapse. The accuracy of this method is not very high, though, so that the conclusion from the numerical results is no better than

$$\nu_t = 2 \pm 0.2 \ ,$$

which is again in agreement with our random walk predictions.

5.5.3 Spatial behavior

So far, we have only shown simulation results for exponents describing the evolution of the number of vehicles, but not their distribution in space. Here, our simulation results are less conclusive. The width $\bar{w}(t)$ vs. t (Fig. 5.10) is, besides the convergence problems already described, best approximated by an exponent

$$\frac{1}{z} = 0.58 \pm 0.04$$

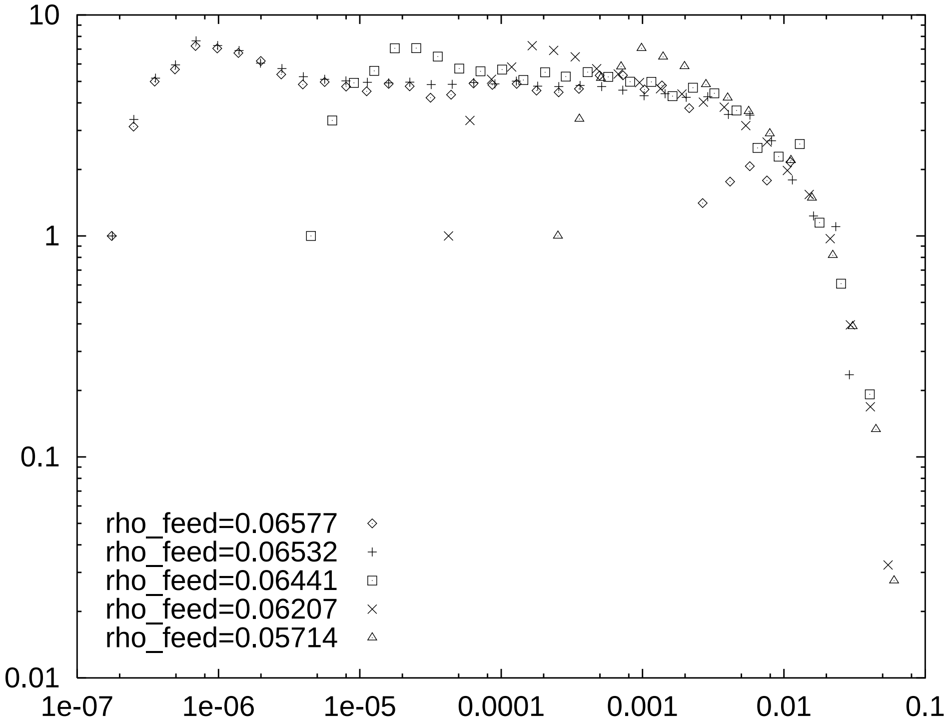


Figure 5.9: Data collaps for the survival probability of jams for the same data as for Fig. 5.8 with $\delta = 0.5$ and $\nu_t = 2$.

instead of 1/2. Measurements of other relations (e.g. w vs. n; not shown) confirm these discrepancies for the spatial behavior for branching jam clusters with $v_{max} > 1$. However, the form $w(t) \sim t^{1/2} \ln t$ vs. t (Fig. 5.10) is also consistent with the numerics.

In an effort to resolve this question, we analyzed large jam configurations. We ran simulations with $v_{max} = 2$ until a cluster reached a width of, say, $2^{13} = 8192$, and stored the configuration of this time-step. About 30 configurations of the same size were used. Box-counting analysis of these configurations was not conclusive, but measuring the distribution of holes inside the configurations is consistent with the results from the cascade equation, presented earlier.

Fig. 5.11 shows a plot of the probability distribution for hole sizes, $P_h(x)$ vs. x, obtained from these configurations. We find

$$P_h(x_h) \sim x_h^{-\tau_h} \qquad \tau_h = 1.96 \pm 0.1 \ ,$$

which is indeed consistent with the prediction $\tau_h = 2$ from the cascade equation.

Nevertheless, our numerical results are not precise enough to distinguish $\tau_h = 2$ from $\tau_h < 2$ for the holes exponent, or the power law fit with exponent 0.58 from the theoretically plausible fit with exponent 1/2 and logarithmic corrections for the width exponent.

5.6 Applications to real traffic

With respect to real world traffic, much of this discussion appears rather abstract. A configuration of size $2^{13} = 8192$, as analyzed in this work, corresponds to more than 100 km of undisturbed roadway, a situation that rarely occurs in reality. However, the following results should be general enough to be important for traffic:

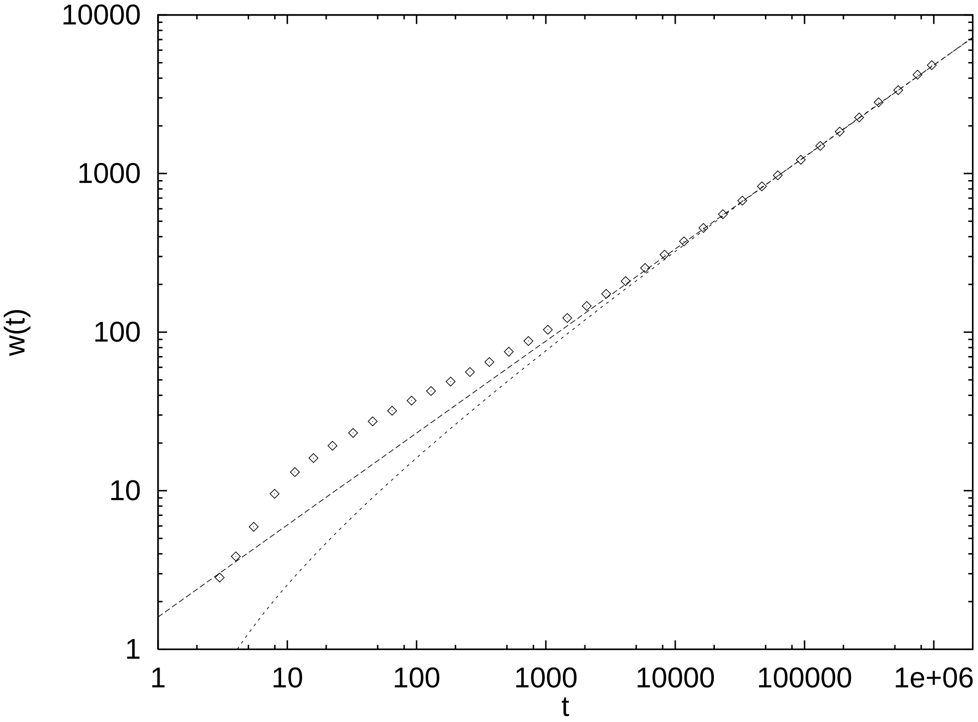


Figure 5.10: Averaged maximum width of clusters, w, as a function of their life-time, t. The straight line has slope 0.58; the dotted line is a logarithmic fit $A \cdot t^{1/2} \cdot \log(t)$ where A is a free parameter. This measurement only converges for life-times $t >\sim 50\,000$, and then, the data material (from the same clusters as for Figs. 5.6 and 5.7 is not enough to distinguish between the scaling law and the theoretically plausible fit with a logarithmic correction.

- The concept of critical phase transitions is helpful for characterizing real traffic behavior. Open systems will tend to go close to a critical state that is determined by the outflow from large jams. This underlying self-organized critical state corresponds to a percolative transition for the jams; i.e. spontaneous small fluctuations can lead to large emergent traffic jams.

- Interestingly, planned or already installed technological advancements such as cruise-control or radar-based driving support will tend to reduce the fluctuations at maximum speed similar to our limit, thus increasing the regime of validity of our results. One unintended consequence of these flow control technologies is that, if they work, they will in fact push the traffic system closer to its underlying critical point; thereby making prediction, planning, and control more difficult.

- The fact that traffic jams are close to the border of fractal behavior means that, from a single "snapshot" of a traffic system, one will not be able to judge which traffic jams come for the same 'reason'. Concepts like point queues [157] or single waves do not make sense when traffic is close to criticality. 'Phantom' traffic jams emerge spontaneously from the dynamics of branching jam waves.

- The fact that holes scale with an exponent around -2 means that, at criticality, the jammed cars are close to not carrying any measure at all. The regime near maximum throughput thus corresponds to large "holes" operating practically at ρ_c and q_{max}, plus a network of branched jam-clusters, which do not change ρ and q very much. The fluctuations found in the 5-minute-measurements of traffic at capacity [56, 132, 69, 2] therefore reflect the fact that traffic flow

Figure 5.11: Probability distribution P_h for hole-sizes x_h. The straight line has slope -2. The average is over 60 configurations, which all have width $w = 2^13 = 8192$. Contrary to all other figures in this chapter, these results were obtained with $v_{max} = 2$, thus confirming that $v_{max} = 2$ is already sufficient to obtain the described branching behavior.

is inhomogeneous with essentially two states (jammed and maximum throughput). The result of each 5-minute-measurement depends on how many jam-branches are measured during this period.

Chapter 6

Further results

This chapter contains further results for the full model and for variations of it. Chronologically, most of this work was done before the work on the cruise control limit. But after having more general insight into the model, these results are now a lot easier to explain.

The chapter starts with repeating the idea of the jam lifetimes, now for the full model. Next, open systems are considered, i.e. systems where cars are inserted at the upstream end and move out of the system at the downstream end. Then, some simple changes in the driving behavior are investigated, and the results can be interpreted as tentative answers to questions such as “What would happen if everybody had a Porsche?” or “What influence have radar-based automatic braking systems?”. Last, I shortly report from an analytical approach.

6.1 Lifetimes in the original model

In the previous chapter, a central quantity was $P_{surv}(t)$, the probability that a jam cluster survives until time t. This quantity was analysed under different conditions, especially for different densities.

A similar analysis is now presented for the complete model of Chapter 3. In that model, several jam clusters can coexist in a system at the same time. One therefore computationally has to distinguish between these clusters. This is done using ideas from cluster labeling in percolation [68].

The technical details of this investigation, especially how the cluster labeling is done on a parallel computer, can be found in the Appendix.

6.1.1 Results of lifetime measurements

Fig. 6.1 (lower branch) shows the results for the lifetime distribution of our traffic model. The figure shows the (normalized) number $P(T)$ of traffic jams of lifetime T; as the data is collected in “logarithmic bins”, the y-axis is proportional to $T \cdot P(T)$. Closed systems (periodic boundaries) of sizes of $L = 10^4$ and $L = 10^5$ were used. The simulations were started from random initial conditions for densities $\rho = N/L = 0.09$, 0.08, 0.07, and 0.06. The first $2 \cdot 10^5$ iterations were discarded to let transients die out; data then was collected over typically 10^7 iterations.

For a density of $\rho = 0.08$ (near the capacity threshold density $\rho(q_{max}) = 0.085$), there is a region where $P(T) \propto T^{-\alpha_1}$ ($\alpha_1 = 3.1 \pm 0.3$) for T approximately between 5 and 50, and another region where $P(T) \propto T^{-\alpha_2}$ ($\alpha_2 = 1.65 \pm 0.08$) for T approximately between 100 and 5000. For a higher density,

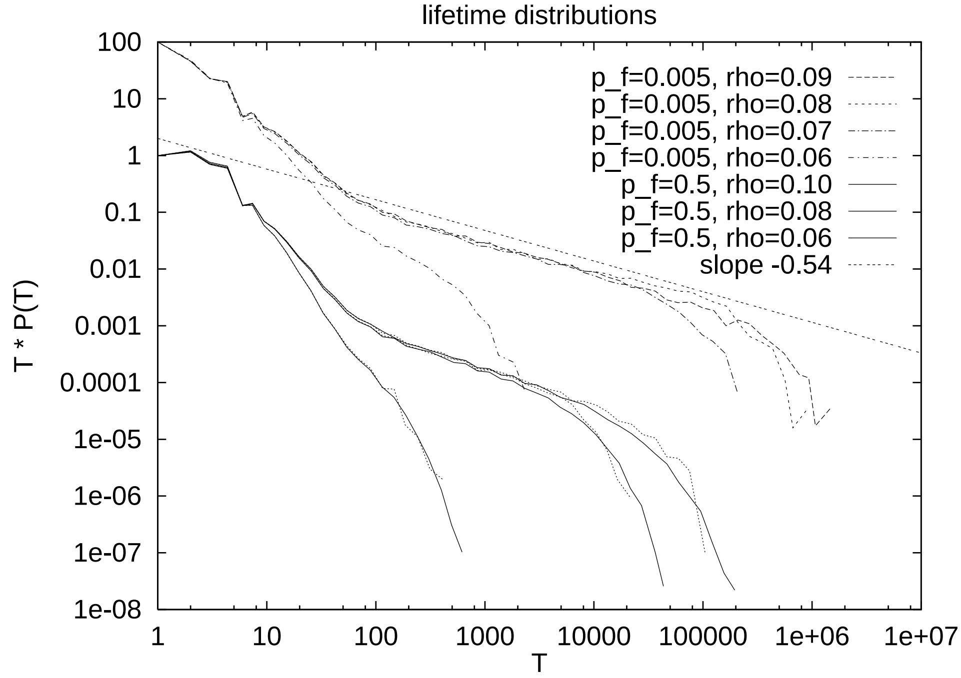


Figure 6.1: Comparison of lifetime distributions $P(T)$ for the traffic jams between the "standard" model and the "model with cruise control" (upper branch). — The data is collected in logarithmic bins, therefore the y-axis is proportional to $T \cdot P(T)$, and it has been normalized such that $P(T=1)=1$ for the lower and $P(T=1)=100$ for the upper branch. — *Lower branch*: standard model, i.e. $p_{free}=0.5$. Straight lines, from left to right: Results for system size $L=10^5$ and densities $\rho=0.06$, 0.08, and 0.10, i.e., below, near, and above the threshold density $\rho(q_{max})$. Dotted lines: Results for same densities, but smaller system size $L=10^4$. — *Upper branch*: including "cruise control", i.e. $p_{free}=0.005$. System size $L=10^5$, densities $\rho=0.09$, 0.08, 0.07, and 0.06, as noted in the legend.

$\rho=0.1$, the second regime gets slightly longer. It vanishes totally for a lower density of $\rho=0.06$. In other words, the change-over from the light traffic regime ($\rho<\rho(q_{max})$) to the heavy traffic regime ($\rho>\rho(q_{max})$) is accompanied by a qualitative change in the lifetime distribution, i.e., the emergence of a regime with $P(T)\sim T^{-\alpha_2}$, but the lifetime distribution does not show critical behavior in the sense of a percolation transition because of the upper cut-off.

This cut-off of the lifetime distribution near $T=50000$ is not a finite size effect. Since we analyze clusters in a space-time-domain, finite size effects can be caused by space or by time. For the space direction, in Fig. 6.1 the results for system sizes $L=10^4$ and $L=10^5$ are superimposed. The scaling region is not any longer for the larger system. Similarly, it was checked that the cut-off is no finite time effect [122].

6.1.2 Reducing the fluctuations at maximum speed

A reduction of only the fluctuations at high speed should converge towards the cruise control limit. For this purpose, the "randomization" step of the update algorithm was replaced by the following rule:

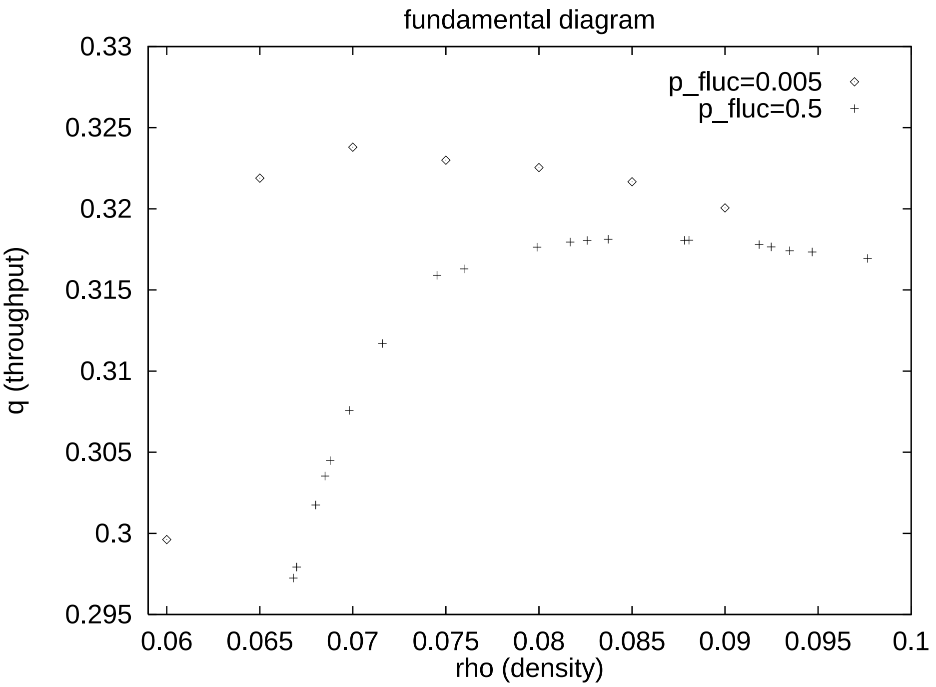


Figure 6.2: Parts of the fundamental diagrams (i.e. throughput q versus density ρ) near the capacity maximum for the original model (+) and for the model with reduced fluctuations (squares) presented later in this text.

- **New randomization:** If a vehicle has maximum speed v_{max} and $gap > v_{max}$, then it reduces its speed by one with a much lower probability $p_{free} = 0.005$. Otherwise, it reduces its speed by one with probability 0.5 (as before).[1]

By this rule, only the fluctuations at $v = v_{max}$ are changed, whereas the slowing down or the acceleration remain the same. For $p_{free} \to 0$ this gives the cruise control limit.

The part of the fundamental diagram (throughput versus density) near the throughput maximum is included in Fig. 6.2. The maximum throughput becomes slightly higher for this new model and is found at a somewhat lower density, but the change in throughput is only 2%.

In the scaling plot of the lifetime distribution (Fig. 6.1), the scaling region of the "second" regime clearly gets longer and extends now over about three orders of magnitude from $T = 200$ to $T = 200000$. In this region, $P(T) \sim T^{-\alpha_2'}$ with $\alpha_2' = 1.55 \pm 0.05$, i.e. converging towards $\delta + 1 = 3/2$ from the cruise control limit.

6.1.3 Discussion

One can attribute the cutoff to the non-separation of the time scales between disturbances and the emergent traffic jams. As soon as p_{free} is different from zero, the spontaneous initiation of a new jam can terminate another one. Obviously, this happens more often when p_{free} is high, which explains why the scaling region gets longer when one reduces p_{free}. Dimensional arguments suggest that the cutoff in the space-time volume, $V \sim wt$, should scale as $V_{co} p_{free} \sim 1$ (for $p_{free} \ll 1$) since this implies

---

[1] The definition of "free driving" is, for "historical reasons", slightly inconsistent throughout this thesis. This influences some of the absolute values (e.g. maximum flow or density at maximum flow), but does not change the overall results, especially not the critical exponents.

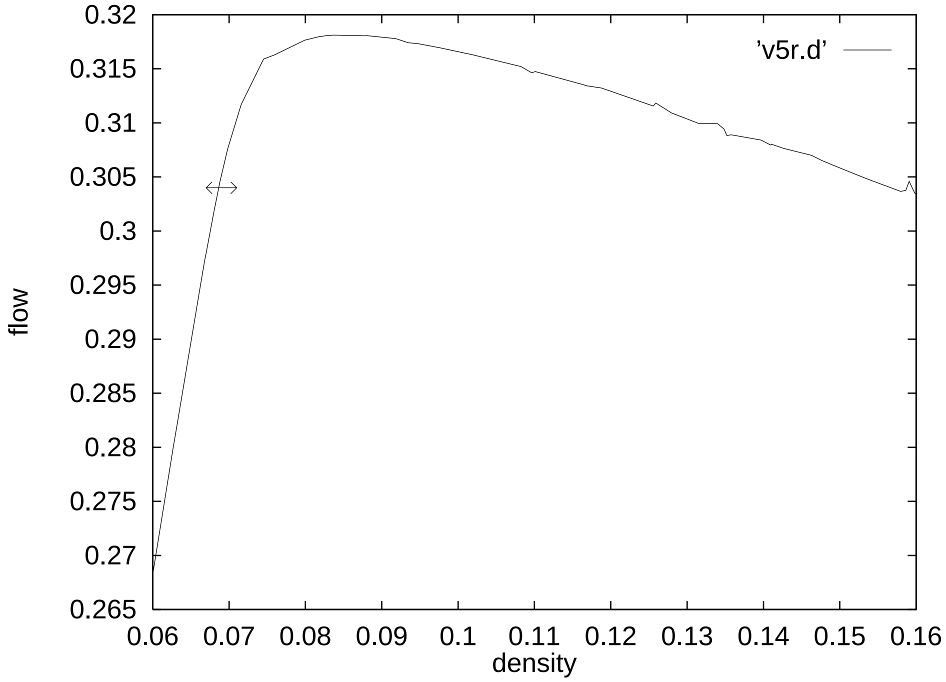


Figure 6.3: Position of the flow-density value from the bottleneck situation; tips of the arrow indicate the error of ρ. The error in the q-measurement is too small to be visible on this scale. The simulation clearly indicates that the self-organizing bottleneck state does not correspond to the maximum of the flow.

that a new jam is initiated in a space-time volume occupied by a previously initiated jam. According to the random walk picture $V \sim s$, so that $s_{co} \sim p_{free}^{-1}$ and $t_{co} \sim p_{free}^{-2/3}$. Measuring these correlation lengths, however, is outside of the scope of the present study.

6.2 Open systems

6.2.1 Traffic in a bottleneck situation

For this section, different boundary conditions are used, leaving the rest of the model unchanged from Chapter 3:

- When the leftmost site (site 1) becomes empty, a new car with velocity $v_{insert} = 0$ is inserted at this position. As our traffic is going from left to right, one may imagine a bottleneck situation where a saturated two-lane-street feeds a street of only one lane (which is simulated).

  The results are independent of the choice of v_{insert}.

- At the right side (i.e. the end of the street), cars on the rightmost six sites are deleted, resulting an open boundary. This simulates the beginning of a more expanded freeway section.

The simulations included grid length up to 10000 sites with durations up to $5 \cdot 10^6$ time steps. After relaxation, the model shows traffic at a density of $\rho_{bn} = 0.069 \pm 0.002$ and a flow of $q_{bn} = 0.304 \pm 0.001$ (Fig. 6.3). This is definitely less than maximum flow, $q_{max} = 0.318 \pm 0.0005$ at $\rho(q_{max}) = 0.085 \pm 0.004$, confirming the observation from Chapter 4 for the deterministic case, that an open boundary on the inflow side is not able to produce maximum flow.

Baldus [9] reports systematic investigations of bottlenecks of this kind, with different inflow and outflow rates, analogous to theoretical investigations of the stochastic asymmetric exclusion model [34].

6.2.2 Self-organization of maximum throughput in the outflow situation

Meanwhile, the outflow *from a jam* self-organizes into maximum flow, so that the outflow behavior is exactly as for the simpler cases of Chapters 4 and 5. In order to see this, in a system of length $L = 10^6$, the left half is filled with density $\rho_{left} = 1$; the right half is left empty (cf. Fig. 6.4). The right boundary is open, i.e., vehicles on sites $L - v_{max}, \ldots, L$ are deleted. The left boundary is closed.

The system is run according to the update rules. After $t_0 = 2 \cdot 10^5$ time steps to let transients die out, we start to count the vehicles which left the system at the right boundary. Fig. 6.5 shows the average throughput

$$q_{open} = \frac{n(t, t_0)}{t - t_0} \ ,$$

where $n(t, t_0)$ is the number of vehicles which left the system between times t_0 and t. We find $q_{open} = 0.318 \pm 0.01$ for large times, which is, within errors, exactly the value of maximum throughput q_{max} for the closed system. In addition, even when filling up the left half of the system randomly only with a much smaller density $\rho_{left} = 0.1$, the outflow is the same. We conjecture therefore that the outflow from a high density regime selects by itself the state of maximum average throughput; and "high density" means an average density above the threshold density $\rho(q_{max})$.

As said before, this is comparable to the case of boundary-induced state selection for asymmetric exclusion models [86], with one difference: Our model does *not* select this state of maximum throughput when adding as many particles as possible at a fixed left boundary (see "Bottleneck situation"). In Chapter 4 it was shown for a simpler model that this can be overcome by some artificial update rules for a few sites at the left boundary; the same is true for the model here.

This point warrants further investigation, since it corresponds to the real world observation that disturbances which are fixed in space, such as bottlenecks or on-ramps, lead to much lower throughput downstream than would be possible theoretically [77]. And one indeed has a similar problem when programming on-ramps (Chapter 9).

6.3 Varying driving behavior

As the next issue, we show how far capacity can be enhanced by changing characteristics of the vehicles or of the driving behavior. We analyze the influence of a "cruise control", of quicker acceleration, of braking "to the point", and of a better car following behavior in the "dead zone" where neither acceleration nor deceleration are necessary. Technically, we define different "randomization probabilities" p_{acc}, p_{sld}, p_{free}, and p_{ptn} for acceleration noise, noise during slowing down, noise at free driving, and noise for platoon behavior, respectively. The definitions will be in a way that $p_{acc} = p_{sld} = p_{free} = p_{ptn} = 0.5$ reduce to the "old" model with $p_{general} = 0.5$. These new noise parameters were integrated into the velocity update in the following way:

- *acceleration:* If the distance to the next vehicle ahead is large enough ($v \leq gap - 1$) *and* maximum velocity is not yet reached ($v \leq v_{max} - 1$), then accelerate with probability $1 - p_{acc}$ by one. Fast acceleration corresponds therefore to a small value of p_{acc}.

Figure 6.4: Space-time plot of the outflow from a jam. As usual, the horizontal direction is the space direction and time is running downwards. The system size is $L = 500$, much smaller than the systems used for Fig. 6.5.

- *slowing down:* If the next car ahead is too close ($gap \leq v - 1$), then reduce the velocity to gap, with a probability of p_{sld} to over-react.
- *free driving:* If the car has maximum speed ($v = v_{max}$) and drives freely ($gap \geq v_{max} + 1$), then introduce with probability p_{free} a fluctuation.

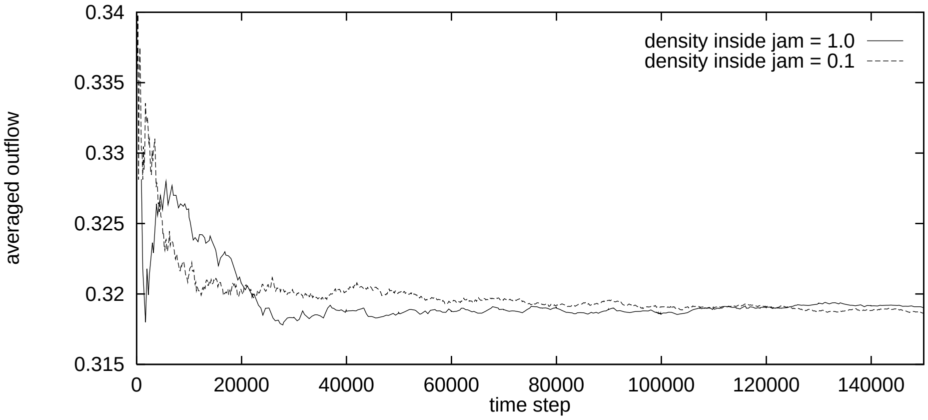


Figure 6.5: Average outflow (see text) from a high density region as a function of time. The straight line shows the outflow with $\rho_{left}(t=0)=1.0$, the broken line shows the outflow from a region with $\rho_{left}(t=0)=0.1$. In both cases, the system self-organizes towards the state of maximum throughput.

- *driving in a platoon:* If the car is driving in a platoon ($v = gap$), then reduce speed with probability p_{ptn} by one.

In order to make this clear, the rules are once again given in pseudo-code:

```
if ($v \le gap - 1$ & $v \le v_{max} - 1$) then
    with probab. ($1 - p_{acc}$) do $v := v + 1$ else $v := v$
else if ($gap \le v - 1$) then
    $v := gap$ ; with probab. $p_{sld}$ do $v := \max[gap - 1, 0]$
else if ($v = v_{max}$ & $gap \ge v_{max} + 1$) then
    $v := v$ ; with probab. $p_{free}$ do $v := v - 1$
else
    $v := gap$ ; with probab. $p_{ptn}$ do $v := \max[gap - 1, 0]$
endif
```

As usual, after the velocity update the vehicle propagation is done.

Fig. 6.6 contains the averaged fundamental diagrams when

- p_{acc} is reduced from 0.5 to 0.005 (better acceleration), or when
- p_{sld} is reduced from 0.5 to 0.005 (reduced over-reaction for slowing down), or when
- p_{free} is reduced from 0.5 to 0.005 (reduced fluctuations at free driving: “cruise control”), or when
- $p_{ptn}$is reduced from 0.5 to 0.005 (reduced fluctuations during platoon driving).

A cruise control gives a capacity of 0.324 vehicles per iteration (2% better than the control case), better braking gives 0.327 vehicles per iteration (2%), and better platoon behavior leads to an increase to 0.380 vehicles per iteration (about 20%). But the remarkable result of these simulations is that an enhancement of the acceleration (p_{acc} reduced) nearly doubles the throughput from 0.318 to 0.623 vehicles per iteration (cf. [135] for a similar prediction). In addition, the space-time diagram for this

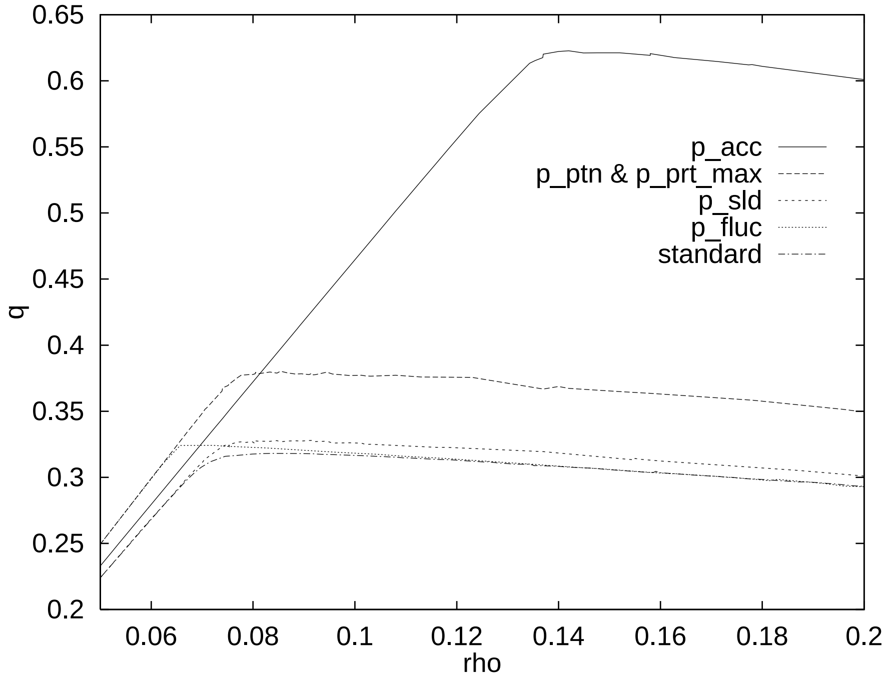


Figure 6.6: Throughput versus density for different sets of parameters (see text). The legends gives the parameter(s) which is/are reduced versus the "standard" model. Note the high increase in possible throughput when p_{acc} is reduced to 0.005 (vehicles accelerate more quickly).

system near capacity (Fig. 6.7) looks qualitatively different from all the others, which look, at least at this resolution, similar to Fig. 3.4.

The reason for this behavior is that, for (infinitely) large systems, it is impossible to suppress jams. And then, the flow is entirely dominated by the outflow, which means that everything else except acceleration does not matter. A more detailed interpretation will be given in the next chapter.

6.4 Analytical results

Analytical calculations were an integral part of the research on the model. These calculations were mostly done by M. Schreckenberg and A. Schadschneider [120, 147, 150].

The idea is to find equations for subsections of the road array. Let the configuration of a subsection of length n (n-sites approximation) be denoted by $\sigma_1, \ldots, \sigma_n$, where σ_i are the state variables, e.g. -1 for empty and $0, \ldots, v_{max}$ for a vehicle with the corresponding velocities, i.e. $v_{max}+2$ states.[2] The probability to find this configuration at time $t+1$ from a given state at time t is given by a master equation:

$$P_n^{(t+1)}(\sigma_1, \ldots, \sigma_n) = \sum_{\{\tau_j\}} W(\sigma_1, \ldots, \sigma_n | \tau_{-v_{max}+1}, \ldots, \tau_{n+v_{max}}) \times$$

$$\times P_{2v_{max}+n}^{(t)}(\tau_{-v_{max}+1}, \ldots, \tau_{n+v_{max}}) \; .$$

---

[2] It is possible to reduce the technically necessary number of possible states by one.

Figure 6.7: Evolution of the model from random initial conditions for a reduced $p_{acc} = 0.005$. Apart from that the figure is the same as Fig. 3.4.

In the simplest case with $n = 1$ and $v_{max} = 1$ this reduces to

$$P_1^{(t+1)}(\sigma_1) = \sum_{\{\tau_j\}} W(\sigma_1|\tau_0, \tau_1, \tau_2) P_3^{(t)}(\tau_0, \tau_1, \tau_2)\ .$$

The next two steps are:

- Find all transition probabilities W. They can be enumerated: For $v_{max} = 1$ each variable has three possible states (empty, occupied with $v = 0$, and occupied with $v = 1$), leading to a table with 3^4 entries.
- Express $P_{2v_{max}+n}$ in terms of P_n. In the case of $n = 1$, one obtains

  $$P_3(\sigma_0, \sigma_1, \sigma_2) = P_1(\sigma_0) \cdot P_1(\sigma_1) \cdot P_1(\sigma_2) \ ,$$

  i.e. spatial correlations are no longer considered (mean field).

Since (i) this did not belong to the work of this thesis, (ii) the technical details are not imperial for understanding the conclusions, and since (iii) the results are relatively easy to find in [120, 147, 150], I will not go any further. The results of this analysis are, however, quite interesting, and they will support conclusions presented in the next chapter:

- For random sequential update and $v_{max} = 1$, already the 1-site approximation (mean field) is exact.
- For parallel update and $v_{max} = 1$, the 1-site approximation is not exact, but the 2-sites approximation is.
- For parallel update and $v_{max} \geq 2$ there does not seem to be an exact solution. However, one can plot 1-, 2-, ..., n-sites approximations, and they converge towards the results obtained from simulation. For example, for $v_{max} = 2$ the 5-sites approximation is only a few percent different from both the 4-sites approximation and from the simulation result.

6.5 Summary

This chapter described four seemingly rather unrelated further observations for the traffic CA.

First, it was shown that, in the complete CA model, there is a finite cut-off for the jam life-time distribution, which is due to interactions between jams. Remember that the cruise control limit allowed only one jam in the system and therefore no interactions.

Next, open systems were discussed. Analogous to the results in previous chapters, also the full model shows self-organisation to maximum flow in the outflow situation, but less than maximum flow in the bottleneck situation.

Third, variations of the individual driving rules were tested. It turned out that only a higher acceleration has a big impact for increasing the flow, whereas, e.g., more controlled braking increases flow by only 2%.

Last, analytical results from other researchers were reported. They are especially interesting because the difficulty to obtain correct analytical results corresponds directly to the different phenomenological complexities for the different subcases of the CA traffic model. This will be discussed in the next chapter.

Chapter 7

Discussion of traffic flow results and meaning for reality

As already described in Chapter 2, for many particle hopping models one knows fluid-dynamical limits. It turns out that these fluid-dynamical descriptions correspond to well-known relations of traffic flow theory. More specifically: Some particle hopping processes are equivalent to the Lighthill-Whitham-theory when specific flow-density-relations are used. Yet, the behavior of the full CA traffic model of Chapter 3 goes beyond the Lighthill-Whitham-theory as it includes spontaneous initiation of jams, and fluctuations. Newer fluid-dynamical models for traffic flow include spontaneous initiation of jams in a similar way, but still no fluctuations. Phenomenologically, taking the *average* over different realisations of the CA starting from macroscopically identical initial conditions leads to the fluid-dynamical description of [80, 87, 88, 96].

The details of these connections and the contributions of the CA model are the topic of this chapter.

7.1 Models with random sequential update

One of the best-investigated particle hopping models is the asymmetric exclusion process with random sequential update [34]. To recall, the rule is "choose one particle randomly and move the particle forward if the next site is free".

In the steady state, already the mean field approximation is exact [120].

This model corresponds, in the hydrodynamic limit, to a noisy and diffusive equation of continuity, i.e.

$$\partial_t \rho + q' \, \partial_x \rho = D \, \partial_x^2 \rho + \eta$$

with a current of

$$q = v_{max} \, \rho \, (1 - \rho) \tag{7.1}$$

(in traffic science, this is Greenshields' relation, see [47]), which leads to the noisy Burgers equation with an additional linear term, i.e.

$$\partial_t \rho + v_{max} \, \partial_x \rho - 2 \, v_{max} \, \rho \, \partial_x \rho = D \, \partial_x^2 \rho + \eta \ .$$

In other words, this exclusion process and the Lighthill-Whitham-theory with the Greenshields flow-density relation (plus noise plus diffusion) describe the same behavior.

Musha and coworkers [115] have proposed the noisy Burgers equation for traffic flow without mentioning the connection to the Lighthill/Whitham theory.

This model class (universality class) has a roughness exponent (see 2.2.3) of $\zeta = 1/2$ and a dynamic exponent of $z = 3/2$ at *all* densities; remember that $z = 3/2$ has the meaning that the time to dissolve a jam of length l is proportional to $l^{3/2}$.

In the steady state, this model shows kinematic waves (= small jams), which are produced by the noise and damped by diffusion (Fig. 7.1). These non-dispersive waves move forwards ($c > 0$) at low densities (Fig. 7.1) and backwards ($c < 0$) at high densities (Fig. 7.2). Somewhere in between, the wave velocity is exactly zero ($c = 0$), and this is the point of maximum throughput [86]. Moreover, this point is selected automatically in the outflow situation (see Chapter 6).

The drawback of this theory with respect to traffic flow is that it does neither have a regime of lamellar flow nor “real”, big jams. Because of the random sequential update, vehicles with average speed $\overline{v}$ fluctuate severely around their average position given by $\overline{v}\,t$. As a result, they always “collide” with their neighbors, even at very low densities, leading to “mini-jams” everywhere. This is clearly unrealistic for light traffic.

Actually, this fact is also visible in the speed-density-diagram. Using Eqn. 7.1, one obtains

$$v = \frac{q}{\rho} \propto 1 - \rho \ ,$$

which is in contrast to the observed result that, at low densities, speed is nearly independent of density.

Judging from Figs. 7.1 and 7.2, changing the maximum velocity in the update does not change the universality class. I.e., picking one particle randomly and then applying the update rules of the CA traffic model only skews the flow-density-relation towards lower densities, but does not lead to other phenomenological behavior and not to another universality class.

7.2 Deterministic models

At first, deterministic models look like subcases of the last section. The prototype model for $v_{max} = 1$ is “for all particles simultaneously, if a particle has a free site in front of it at time t, then it moves one site ahead during this iteration”.

But taking away the noise from the particle update completely changes the universality class [85]. The model now corresponds to the non-diffusive, non-noisy equation of continuity with a linear flow (cf. Chapter 2)

$$q(\rho) = \frac{1}{2} - |\rho - \frac{1}{2}|.$$

This yields a linear Burgers equation:

$$\partial_t \rho + \mathrm{sgn}[\frac{1}{2} - \rho]\, \partial_x \rho = 0$$

Here, the dynamic exponent z is equal to 1; the roughness exponent ξ depends on the initial conditions [85].

In some sense, this model is the complementing ingredient for the final traffic model. It lacks the mechanisms for spontaneous formation and dissipation of jams, but the traffic jam dynamics itself is a useful approximation of reality:

Figure 7.1: Space-time plot for random sequential update, $v_{max} = 1$, and $\rho = 0.3$. Clearly, the kinematic waves are moving forwards.

- For $\rho < \rho(q_{max})$, all jams eventually dissipate. A dynamic exponent of $z = 1$ means that a jam of length $2 \cdot l$ needs twice as long to dissipate than one of length l, $t(2l) = const \cdot (2l)^z = 2\,t(l)$, which intuitively makes more sense for reality than the exponent $z = 3/2$ of the random sequential update. When one disturbs, the disturbance decays deterministically, also with $z = 1$. The wave

Figure 7.2: Space-time plot for random sequential update, $v_{max} = 5$, and $\rho = 0.5$. Clearly, the kinematic waves are moving backwards, i.e. for $v_{max} = 5$ the density $\rho = 0.5$ is above $\rho(q_{max})$.

velocity for $\rho < \rho(q_{max})$ is $c = v_{max}$, meaning that correlations travel with the cars. This is clear: As soon as all disturbances are sorted out, the density profile travels with v_{max} (Fig. 4.2).

- Above $\rho(q_{max})$, the situation is different. Some jams are never sorted out, and they survive forever while traveling backwards with the kinematic wave velocity $c = -1$. A disturbance, however, makes the model non-stationary only for a short time: Usually, it just shifts another wave or part of it to a new wave originated by the disturbance (Fig. 4.3). Here, as also for $\rho < \rho(q_{max})$, the average transient time after a damage scales as $T_{trans} \sim |\rho - \rho(q_{max})|$.

- *Near* $\rho(q_{max})$, relaxation of the model needs more and more time. In the case of $\rho = \rho(q_{max})$ and periodic boundaries, a disturbance causes a hole in the traffic flow, which travels with $c_< = v_{max}$, and a wave, which travels backwards with $c_> = -1$. In a closed system, the hole and the wave ultimately meet again and cancel each other. As a result, the transient time here scales as $T_{trans} \sim L$, where L is the system size.

In short, this model already possesses the ingredients for the critical behavior in the cruise control limit, albeit in an overly deterministic way.

Meanwhile, this model does not have spontaneous wave formation; and adding noise in the usual way just leads back to the model of the previous section.

Using a maximum velocity higher than one does not change these results (Chapter 4).

As said before, most work using CA models for traffic is based on this model. Biham and coworkers [17] have introduced it for traffic flow, with $v_{max} = 1$. Other authors base further results on it [15, 116, 167, 182], also for two-lane traffic [117]. Vilar and coworkers [172] use it with $v_{max} = 5$. It is also the basis of the two-dimensional CA models for traffic [30, 112, 118, 119, 166].

7.3 Stochastic parallel update with $v_{max} = 1$

There is, however, another way of adding noise, which is used in the CA traffic model introduced in Chapter 3. There, all vehicles/particles simultaneously evaluate their state, but their resulting action is randomized. From an algorithmical point of view, this is an obvious choice, but for analytical work, it is much more complicated.

Judging from space-time-plots, stochastic parallel update with maximum velocity $v_{max} = 1$ (Fig. 7.3) does not seem much different from the standard stochastic asymmetric exclusion process (Fig. 7.1, where, though, due to a different density the movement direction of the kinematic waves is different). Apart from the same particle-hole-symmetry, it also shows the same kind of kinematic waves, moving forwards for low densities ($\rho < \rho(q_{max})$) and backwards for high densities; and the waves itself do not look much different from the random sequential update. However, two facts indicate that one enters a different dynamical regime: The mean-field approximation is no longer exact, and it is possible to work with the cruise control limit.

Analytic results

Analytic calculations [147, 150] show that the mean-field approximation is no longer exact. Compared to the asymmetric stochastic exclusion model, particles have a tendency to remain equidistant (effective anti-ferromagnetic potential), which is clearly realistic for cars. This results in a higher maximum flow compared to random sequential update.

The next approximation beyond mean field, i.e. the 2-cluster approximation which takes two-point correlations into account, turns out to be exact for $v_{max} = 1$.

Figure 7.3: Space-time plot for parallel update, $v_{max} = 1$, and $\rho = 0.5$. This time, the waves do not move; $\rho = 0.5$ is the density of maximum throughput. Apart from the different velocity of the waves, the picture looks quite similar than random sequential update (Fig. 7.1).

Cruise control limit

Because of the parallel update, it is possible to define the cruise control limit exactly as in Chapter 5. For $v_{max} = 1$ this means that all particles with speed $v = 1$ and $gap \geq 1$ move deterministically with rate one. The model loses its particle-hole symmetry, and the fundamental diagram is skewed to the left. Meanwhile, it is now possible to completely separate jams and lamellar flow. A system with density $\rho < \rho(q_{max.det}) = 0.5$ can be prepared to behave completely deterministically (Fig. 7.4). Then, a single disturbance initiates, via a chain reaction, a jam, and depending if one has super-critical, sub-critical, or exactly critical density, the emergent jam is, in the average, growing, shrinking, or "undecided".

Since $v_{max} = 1$ means that the jam is dense, this completely corresponds to a random walk description (Chapter 5), i.e. the number of cars in the jam, $n(t)$, behaves as the trajectory of a random walk. If the inflow is higher than the outflow (middle part of Fig. 7.4), then cars enter the jam with a higher rate than they leave, the average jam grows linearly in time, and the corresponding random walk is biased. If the inflow is lower than the outflow, the jam quickly dissolves. If the inflow exactly matches the outflow (which happens generically, because of the outflow behavior; see, e.g., in the lower part of Fig. 7.4), then the random walk is unbiased and shows its typical fluctuations and exponents, as explained earlier.[1]

This instability definitely goes beyond a Burgers equation, since the Burgers equation can never be instable. One has to include the effect of momentum into the partial differential equations.

It is, however, from plots such as Fig. 7.3, not clear why the cruise control limit behavior, i.e. singular jams and the phase separation, should play an important role. This only becomes clear at higher maximum velocities. For that reason, it is worthwhile to spend some more time describing the $v_{max} = 1$ cruise control case, because the following remarks remain valid for arbitrary v_{max}.

It is useful to compare the CA cruise control case results with results from partial differential equation (PDE) models for traffic flow [80, 87, 143, 156]. This is easiest when starting from homogeneous initial conditions, as is done in all figures of this chapter.

- For low densities, i.e. $\rho < \rho(q_{max})$, a disturbance decays, although some survive longer than others. The *average* behavior corresponds to the decay of the wave amplitude of the PDE.
- For high densities, i.e. $\rho > \rho(q_{max})$ but $\rho < \rho(q_{max.det})$, a homogeneous initial state is a meta-stable state. Here, a disturbance *in the average* grows linearly in time (growth of the wave amplitude). However, an *individual* disturbance can still die out.

The picture for the *average* CA wave is this similar to the linear stability analysis of the PDEs for traffic flow, where disturbances become instable when the density is higher than a critical value. There is, however, a difference in the growth rate: PDE waves grow exponentially, CA waves grow linearly in time.

Thus, the flow restriction is not caused by an intrinsic density limit for lamellar flow, but by the fact that there is a density above which the *waves* no longer decay. The rate of inflow which a wave can barely support without growing (in the average) is just the rate of release on the other, the outflow, side. And this rate of release is given by the acceleration characteristics. This is why, in Chapter 6, only the change in the acceleration had a remarkable influence on throughput.

In consequence, a higher flow than the outflow from a jam can only be sustained for a limited time. Measures for stabilizing traffic flow would thus only be useful if the disturbance could be suppressed during the whole rush hour. Otherwise, a disturbance would eventually happen, it would grow because the inflow would be supercritical, and the outflow from the jam would reset the traffic system to the same lower flow as before the introduction of the new technology. From traffic science measurements, this phenomenon is known as hysteresis [169].

---

[1] Actually, for Fig. 7.4 this is not precisely true. There is a correlation via the periodic boundary conditions.

Figure 7.4: Space-time plot for parallel update (cruise control limit) and $v_{max} = 1$. The first 50 time steps show deterministic supercritical flow; after the disturbance, the jam clearly grows; when the outflow reaches the jam again via the periodic boundary conditions, both jam fronts describe a random walk (but see footnote in text).—In order to be clearer, the figure uses a lower acceleration probability than normal.

In the cruise control limit, the flow-density-relation is non-smooth at q_{max}. This just reflects the phase separation:

- Below $\rho(q_{max})$, in the steady state there are only cars with maximum velocity v_{max}, leading to a linear flow-density relation $q_< = \rho \cdot v_{max}$. The velocity of the kinematic waves $c_< = q' = v_{max}$, i.e., disturbances move with the traffic.
- Above $\rho(q_{max})$, traffic separates into laminar regions operating at $(\rho(q_{max}), q_{max})$ and jammed regions operating at $(\rho = 1, q = 0)$. The fundamental diagram is just the linear interpolation between both: $q_> = q_{max}\,(1 - \rho)/(1 - \rho(q_{max}))$. The velocity of the kinematic waves

$$c_> = \frac{\Delta q}{\Delta \rho} = -\frac{q_{max}}{1 - \rho(q_{max})} \; .$$

No other wave velocity is possible in the steady state. The effect is similar (although smeared out) for higher velocities, which explains why the waves in the model all move backwards with similar velocities.

Takayasu and Takayasu [167] have shown that the bi-stability can also be obtained by a deterministic model—although in this case one replaces the random walk picture by its deterministic average.

7.4 The full CA model for traffic flow

For higher maximum velocities, e.g. $v_{max} = 5$, there are three new observations: The cruise control limit dynamics now visibly plays a role, i.e., singular jams are an important feature of the dynamics; jams can branch; and analytic work is more difficult.

- **Cruise control dynamics.** The cruise control regime now visibly plays a role, i.e. simulated traffic flow separates into laminar and jammed regions, as in reality. However, for this, there is no *dynamical* difference to the $v_{max} = 1$ case, it is only that the natural parameters are closer to the cruise control case.

  Together with the knowledge that the $v_{max} = 1$ case looks like random sequential update, and the knowledge of the cut-off in the lifetime measurements (see Chapter 6), one comes to the conjecture that a correct description of traffic contains a cross-over from cruise-control dynamics to Burgers/KPZ dynamics. An associated correlation length ξ describes up to which length the cruise control description is valid; there is also a similar correlation time. Judging from Treiterers space-time-plots (see Fig. 3.1), for traffic this length scale has to be larger than 10 km.

  It is this effect which is responsible for the special shape of the fundamental diagram. If ξ were infinity (cruise control limit), the fundamental diagram would be non-smooth at $\rho(q_{max})$. However, a $\xi < \infty$ smoothes out the peak region.

- **Jams can branch.** Completely new for $v_{max} > 1$ is the effect that jams now can branch. This behavior enters *naturally* into the model when one attempts to obtain a more realistic fundamental diagram by using a maximum speed v_{max} larger than one. The branching is due to the memory effect of $v_{max} > 1$, where cars leaving a jam need some time to accelerate to v_{max}. During this acceleration phase, they follow different rules than when driving at maximum speed, which may lead to a sub-branch of the jam (Fig. 7.5).

  I am not aware of any traffic measurements which directly support this picture, but it seems intuitively reasonable—after an accident site has been cleared, one usually drives through many start-stop-waves with free traffic in between—, and it is actually very difficult to suppress it in the simulations as soon as one allows fluctuations in the acceleration.

Figure 7.5: Space-time plot for parallel update (cruise control limit), $v_{max} = 5$, $\rho = 0.09$, i.e. slightly above critical. The flow is started in a deterministic, supercritical configuration, but from a single disturbance separates into a jam and a region of exactly critical density.—This is phenomenologically the same plot as Fig. 7.4 except that $v_{max} = 5$.

Chapter 9 will contain an example how branching will make the density estimation for congestion pricing much harder than one could expect.

- **Analytic results.** The fact that the internal jam dynamics becomes more complicated is reflected by the fact that the analytic approximation becomes more complicated [147, 150]. The

n-sites approximations only converge towards the correct solution for $n \to \infty$, but do not seem to become exact.

Even with its natural parameters, the model behaves very similar to the fluid-dynamical results of the Daimler-Benz research group [80, 88, 143, 156]: Starting from ordered initial conditions (Fig. 7.6), dynamic instabilities develop when $\rho >\sim \rho_c$, which transform into kinematic waves, which survive for a long time. See Ref. [96] for a thorough treatment of the differences and connections between dynamic and kinematic waves. These models use a smooth flow-density relation; the correct behavior comes out due to the nonlinear effects in the equations.

It is interesting to note that both the fluid-dynamical and the particle hopping approach go in the same way beyond the Lighthill-Whitham theory: They replace instantaneous velocity adaption with momentum (an additional momentum equation for the fluid-dynamical models, the acceleration rules for the hopping model), which leads to the instability of the lamellar flow of high densities.

7.5 Random sequential vs. parallel update

As mentioned in Chapter 3, one usually assumes that random sequential updating is a better approximation of reality than parallel updating, and that one chooses parallel updating only for computational reasons. Why is for traffic flow parallel updating more realistic?

The problem is that for the random sequential update, the *fluctuations* of the speed are proportional to v_{max}, whereas in the parallel update, they are independent of v_{max}. In consequence, the *relative* fluctuations around $\overline{v}$ are constant for random sequential update, but decrease as $\propto 1/v_{max}$ for parallel update. This is what makes the cruise control limit relevant for higher velocities: Fluctuations at free driving are decreased.

In order to "save" the random sequential update, one would thus have to reduce the fluctuations around the mean velocity for free driving. This could, e.g., be achieved by moving smaller spatial steps and making the probability of moving proportional to the velocity of the car. But then, during one cycle a vehicle would have to be moved several times, which would make the simulation much slower. Noise-reduction techniques for surface growth models work similarly [81, 179].

7.6 Summary for traffic flow theory

As one sees, once one has an overview over connections between the different hopping models and to the fluid-dynamical models, there is a wide variety of traffic flow phenomena which could be readily explained. This will be, together with discussions with traffic engineering people and with data evaluations, the topic of further work. Here, I just want to summarize the—in my view—most important results which have come out of this research in the area of traffic flow:

- There is strong evidence from this work, from fluid-dynamical results, and from car-following simulations that, on a sufficiently small scale, stable traffic flow regimes are only: light traffic $\rho < \rho_c$, outflow traffic $\rho = \rho_c$, and dense traffic $\rho \approx 1$. All other traffic with $\rho > \rho_c$ undergoes a spontaneous phase separation into outflow regions of ρ_c and jammed regions with densities close to one. Many traffic measurements can be explained using this clear picture.

Figure 7.6: Space-time plot for parallel update, $v_{max} = 5$, $\rho = 0.09$ (i.e. slightly above $\rho(q_{max})$), starting from ordered initial conditions. The ordered state is meta-stable, i.e. "survives" for about 300 iterations until is spontaneously separates into jammed regions and into regions with $\rho = \rho(q_{max})$.

- Since all traffic above ρ_c eventually separates into the two regions explained above, the maximum flow is given by the outflow from a jam, and this is given by the acceleration rules. Faster acceleration at the outflow of a start-stop-wave greatly enhances the throughput.

This situation is described in the average by a *linear* Burgers equation. The fluctuations around this average picture (apart from the branching complications) are described by a random walk.

- The opposite picture would be that traffic flow is mostly constrained by disturbances "all over the place".

  This picture corresponds to the noisy nonlinear Burgers equation and the standard KPZ universality class.

- All results indicate that in reality there is a crossover between both descriptions, where the "cruise control limit"-description is valid on short scales, and the Burgers description on large scales. Both descriptions are thus correct, but for different length scales, and judging from traffic observations, the crossover length is larger than 10 km.

  The length and time scale of this crossover increase with *de*creasing p_{free}, the fluctuations at maximum speed.

- Traffic jams out of nowhere are caused by fluctuations at maximum speed. In the fluid-dynamical description, they are caused as instabilities in the equations for *dynamic* waves, but survive as *kinematic* waves [88, 80, 96]. The CA model supports this picture.

- Reducing the fluctuations at maximum speed does not increase maximum throughput very much. Therefore, installing technology for better controlling maximum speed will not be very useful without coupling it to the acceleration (see above).

  The reason why that does not help is that maximum flow is dominated by the outflow from a jam, and as long as it is not possible to keep traffic *completely* free of jams, it does not make a big difference if there are many or few jams in the system. However, reducing fluctuations at maximum speed will largely enhance throughput when used in bottlenecks of a shorter length: Then, it is feasible to keep the traffic along this length completely free of jams.

- Jam waves branch. Between the branches, one finds holes "of all sizes" (see Chapter 5). This makes the concept of queues difficult for state estimation problems: From a spatial configuration at a given time step alone, one cannot decide if two jams have the same dynamical origin or not—one needs the history.

The results about the universality classes mean that, to a large extent and in the way described here, all car following models will lead to the same macroscopic behavior. Since all major simplifications of the the CA traffic model lead either to losing the effect of spontaneous jam formation or to losing the lamellar flow regime, it is improbable that a much simpler model can be found which gives the same macroscopic behavior. On the other hand, using a more complicated model is only justified if one is interested in effects which go beyond the ones identified in this work. This is certainly the case for effects like the design of ramps or intersections, but seems—at least in first approximation—useless for most studies on the freeway network level. In other words: If one is satisfied with a numeric version (i.e. including diffusion and noise) of the Lighthill-Whitham (LW) model for certain questions but needs a microscopic version, then the approach presented here should be more than enough. It is even better than discrete LW in at least one respect: It correctly reproduces the instability above $\rho(q_{max})$. For example, whereas discrete LW will always predict the same average queue before a bottleneck, the particle model will allow both days without breakdowns and days with very severe breakdowns. This can make a difference when, e.g., only the queue from the severe breakdown reaches back into, say, another intersection.

Moreover, together with our implementation results on different supercomputers (Chapter 11) this means that, on current supercomputers, it will be difficult to find a faster microscopic traffic simulation model with the same phenomenology.

Part B

NETWORK TRAFFIC

Chapter 8

Some issues of traffic in networks

The text now moves away from single lane traffic flow considerations towards road networks with ramps connecting different segments. Such a road network could be the freeway network of the German land Northrhine Westfalia, but it could also be a simple "toy" example. This chapter, the first of three, reviews other work on network traffic. The first section explains the conflict User Optimum vs. System Optimum. Section 8.2 explains traffic management systems and congestion pricing theories. In Section 8.3, it is reviewed how simulation models handle traffic streams, which is related to Section 8.4, about traffic assignment methods, and Section 8.5, about origin-destination information. Section 8.6 gives an overview of how two-dimensional cellular automata techniques have been used for traffic networks.

8.1 Nash Equilibrium vs. System Optimum

Again, as a first approximation, one would like to use the network infrastructure as efficiently as possible. Even if one discards environmental concerns, practicability, personal preferences, etc., there is still the problem that "efficient use" means something different for users than for a global observer [3, 22]:

- A *Nash Equilibrium* (NE; also called *User Optimum*) is defined as a state where no user can reduce his/here costs by unilaterally changing his/her behavior.
- A *System Optimum* (SO) is defined as a state where the sum of all individual costs is globally minimal.[1]

These system states are often in conflict, because, as is common knowledge, individual interests rarely coincide with global welfare. An example will be given in Chapter 9.

8.2 Traffic management

Different projects, as well in the U.S. as in Europe, such as PROMETHEUS, DRIVE, SCOPE, etc., attempt to make road traffic safer and more efficient (see, e.g., [54, 55, 70, 127]). I will shortly outline

---

[1] Whereas all SOs need to have the same sum of costs, there can be more than one SO. However, one should take this point from the pragmatic side, since in real, fluctuating road traffic, it is infeasible to find an exact optimum.

the part of these projects which are most important here. This is the traffic management part. The idea of traffic management is to guide vehicles in an efficient way through the network, i.e.

- to give them routing information if they are in an unfamiliar area,
- to warn them of incidents and to guide them around incident locations in an efficient way, and
- to redistribute traffic such that the network is used more efficiently.

The necessary data collection and data transmission are parts of the projects Advanced Transport Telematics (ATT, in Europe) or Intelligent Vehicle Highway Systems (IVHS, in the U.S.). Advanced Traffic Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) are intended for driver information and traffic management.

Providing routing information for unfamiliar drivers does not pose any big problems. For incident management at least an improvement (if not an optimal solution) is foreseeable. For the third point, however, the benefits are unclear. It depends (i) on in how far today's commuter traffic is already operating at a Nash Equilibrium, or could benefit from better driver information, and (ii) on how much one assumes a System Optimum to be better than a Nash Equilibrium.

Mahmassani and Peeta [102] report from simulation results a 15% advantage of the SO over the NE, whereas Zallmann and König [184] found practically no difference. Hall [55] expects possible benefits of IVHS mostly in non-recurrent (i.e. caused by an incident) congestion. Emmerink et al [38] evaluate, based on simulations using behavioral rules for travelers, possible benefits of ATIS systems as a function of market penetration. Their results with respect to recurrent vs. non-recurrent congestion are similar to Hall's.

Economists do not believe that it will be possible to push people towards an SO if that means a disadvantage for the user [3]. They propose congestion pricing as a solution, and prove for a bottleneck situation that, when the social cost of using the bottleneck equals the toll, then the SO is reached, and each user pays exactly the amount of money he/she is ready to pay for the time he/she saves. Social cost is defined as the sum of all additional costs one additional user imposes on all users.

MacKie and Varian [101] propose a similar method for the Internet. The problem is very similar indeed since each longer message when submitted to the Internet is broken into pieces which are transmitted in separate packets. And each packet searches its own path through the network.

Note that all these policies propose *congestion* pricing, not road pricing. That means that it is not the use itself of the infrastructure which is priced, but the social costs one imposes to others (and himself) by over-using it.

A problem with these policies so far is that it is unclear how to implement them in a spatial, fluctuating, and unpredictable environment. MacKie and Varian propose that packets carry a certain amount of money which they use to pay congested parts of their travel. But what if, shortly before they reach their target, there is a prohibitively expensive bottleneck? Something similar is true for road traffic, where it cannot be expected that commuters want to live with a toll which is highly fluctuating from day to day and from time slice to time slice just because congestion toll is supposed to match social costs.

Thus, one has to develop this line of thinking further. Chapter 9 will present a situation where even in a highly fluctuating environment a slowly self-adjusting congestion toll leads to an acceptable result.

If people do not like pricing, other coercive measures might lead to success. In Chapter 9, one will find an example where a simple traffic light would already help considerably.

8.3 Modeling traffic streams in road networks

Since many traffic assignment methods are directly coupled to simulation models, it is necessary to review some basic features of network traffic models. Contrary to Chapter 2 with the focus on traffic flow, this section will concentrate on issues which are related to traffic stream management.

Two issues are of particular importance here: the implementation of directional information (trip plans vs. turn counts), and the handling of queues or jams. Both issues are interrelated, as will be seen.

8.3.1 Trip plans vs. turn counts

Most traffic simulation models to date, even microscopic, use turn counts as data input. This means that, for each intersection and each time slice, one either has the percentage or the number of cars relating incoming to outgoing travel, i.e. the number of left turns, right turns, cars going straight, …, for each incoming road.The opposite, basing the simulation model on trip plans (also called paths), means that each car individually knows at each intersection where it wants to continue its trip.[2]

The decision to use turn counts in the past was due to pragmatic reasons: Turn count data is much easier available. Moreover, on a certain level of modeling, it does not make a difference. For example, a macroscopic model based on the Lighthill-Whitham equations [31, 29, 133] produces an average over possible traffic situations as output [134]. Thus, as long as one uses *average* turn counts (for example monthly averages of all weekdays) as input, then simulation and input data are compatible.

The situation becomes more problematic when unexpected incidents occur. Assume, for example, that a road is suddenly blocked, e.g. by an accident (Fig. 2.2). Then, a model based on turn-counts has no means to predict the resulting reorganization of the traffic patterns. A model based on trip-plans, however, can, based on behavioral rules or on a traffic management strategy, re-plan the affected trips.

8.3.2 Handling of queues

All simulation models obey mass conservation and therefore handle queues in one way or the other: If a segment cannot deal with a certain flow demand, and if vehicles have no alternative, the vehicles have to wait somewhere. There are, however, differences in how realistic queues are handled:

- Macroscopic models [129] often handle flow-density-relationships on a link basis. If, for example, an intersection at the end of a link is blocked, this immediately increases the average density of the whole link.
- Point queue models [157] have queues before bottlenecks (e.g. intersections) which have no spatial extension. Back-spill of jams, which produces grid-lock, will not occur.
- Deterministic models (e.g. fluid-dynamical models based on the Lighthill-Whitham theory [130, 31, 29, 133], but also deterministic microscopic models [17, 116, 117] or combined approaches [154]), lead to *average* queues. Such models will give an deterministic answer to a question if a certain incident will cause, say, a network breakdown or not.
- *Stochastic* microscopic models can produce different scenarios from the same input data. For example, sometimes a queue can spill back to another intersection and cause a network break-

---

[2] I will, when that makes writing easier, personalize the car. DVU (driver-vehicle-unit) is used in some texts.

down, whereas in the average, the queue does not grow that long. For obtaining interpretable results, averaging over many different scenarios may be necessary.

This method is the only one available if one is interested in fluctuations, e.g. the *probability* that a certain incident can lead to a network breakdown.

Note that the average itself can be different from the averaging models, if, for example, the network breakdown has enough weight in the averaging procedure. If one wants, for example, obtain measures of efficiency for a certain traffic management strategy, then one has to take network breakdowns into account, and if they do in fact happen, than deterministic models will give a wrong answer.

The connection to the turn counts / trip plans issue is that, as soon as either the simulation or the turn counts data or both are no longer averaged (for example in real time applications, where the turn counts are not averaged), driving the model by turn counts is inconsistent as soon as fluctuations become large, e.g. by traffic jams: For example, suppose a situation where the deterministic model predicts, say, 500 vehicles at a specific intersection in a certain time slice. Meanwhile, in reality all vehicles which want to make a left turn are caught in some jam somewhere, and there are only 300 vehicles, which all go straight or to the right. When using these turn-counts percentages, the simulation will send too few cars to the left in this time slice. Research is necessary to clarify how much averaging is necessary to avoid such situations.

There will be data assimilation methods to handle this, especially in real time applications, when one does not want to give up the higher simulation speed of a macroscopic model. This is, though, not a conceptually satisfying solution. A better solution is to directly drive the model by trip plans. This does not ensure compliance with reality, but at least one can be sure that the simulation follows *one* of the possible trajectories of the system in state space.

There are a number of models in the literature which could easily support trip plans. Nonetheless, this is usually not the case. For example, both NETSIM and FREESIM (based on INTRAS) use turn counts [146].

Meanwhile, there are some notable exceptions. CONTRAM moves *packets* of vehicles with identical origin-destination pairs through the network. Emmerink and coworkers [38] use a model where 300 packets are used for each simulation run. Yet, the use of packets is only possible as long as there are enough vehicles with the same path in each space-time unit of the simulation. CONTRAM achieves this by using time-steps of 15-30 minutes and by taking links as the basic unit of space. Emmerink and coworkers use a corridor network, i.e. all vehicles start from the same node and drive to the same node. In such a case, the number of possible paths is much smaller than in a full network. Cremer and Meissner [29] consider only two routing alternatives and model them, in a fluid-dynamical model, as portions of the flow.

In terms of microscopic models, DYNASMART is a path-based microscopic simulation model used for urban traffic assignment [102]. Zallmann and König [184] use a traditional turn-count based model for the traffic assignment, but completely vehicle-based models for the microsimulations. The demo version of the TRANSIMS microsimulation [165] also uses trip plans. PARAMICS [181] includes behavioral rules and therefore trip plans for travelers.

8.3.3 Origin-destination information

Turn counts data itself does not give any information on *correlations* between turns, for example the information if a vehicle making a left turn at a certain intersection is likely to make a right turn at the next intersection.

Yet, turn counts data contains temporal correlations: Assume that one has, at a certain time and at a certain intersection A, many cars making a left turn. Assume further that one has, at the next intersection B and after the time one needs to drive from A to B, many cars making a right turn. Than it is *likely* that these cars are the same. These are the kind of correlations which are used by current methods of O-D estimation [27].

It is clear though that this method cannot reveal long distance correlations. Here, it seems more promising to estimate O-D relations from demographic data, where one has, for example, information on where people live and where they work [165, 153]. And then, evidently, one can run microsimulations based on that data and validate the simulation results with turn counts or other observational data.

As mentioned above, a major reason that one currently uses turn counts to drive the simulation models is that this data is, at least in principle, relatively easy to obtain: One just counts the cars. Origin-destination (O-D) data is much more difficult to obtain. It is though open which is the more economic method when dealing with large systems. Reliable automatic counting systems seem difficult to develop [134], and it might be better in the long run to develop a method method based on demographic data and simulation which can do with fewer sensors for validation and incident detection.

8.4 Traffic assignment methods

8.4.1 Static equilibrium methods

A commonly used method in traffic science is *static equilibrium assignment*, usually only called equilibrium assignment [164, 184, 155]. The problem is equivalent to a static multi-commodity network flow problem.

Commonly, people use a relaxation algorithm which works as follows:

(i) Assume that a network of nodes and links/arcs with link cost functions and a matrix of origin-destination-streams (O/D-matrix) are given. Choose one of the streams.

(ii) For the O-D-stream, find the cheapest path through the network.

(iii) Calculate the resulting density on each link. Calculate link costs (for example travel time on that link) from this density.

(iv) Take the next O-D-stream and go back to step (ii).

This algorithm is known in the traffic science literature as *All-or-nothing* assignment method because is assigns each traffic stream at once. It collapses, for example, when there is an O-D-demand where no path exists which carries the whole demand.

There are several variations of this general scheme in the literature, e.g.:

- In order to avoid the "all-or-nothing" problem, one can replace (*ii*) by [184, 155]

  (ii') For the O-D-stream, calculate the shortest path through the network. *Assign only* $1/k$ *of the demand to it.*

- More recent implementations often use a simulation model instead of step (iii) [164].
- The above scheme may only be taken as a first guess, which is followed by a relaxation algorithm where small parts of the streams are exchanged [184].

Since in the assignment, each stream optimizes its own cost function, the result corresponds to a Nash Equilibrium.

Note that the equilibrium assignment assumes a steady state solution, i.e. a solution where the traffic streams according to the O-D-matrix are stationary.

8.4.2 Queues

Static equilibrium assignment is obviously problematic for time-varying demands. But it completely breaks down above capacity: Queues develop at the entrances of the overloaded links, and equilibrium solutions become history dependent. In other words: Static equilibrium assignment assumes static traffic streams. Yet, under static *and* congested circumstances, queues would be infinitely long.

Arnott and coworkers [3] treat the simplest case of this problem: A single bottleneck during the rush period. It is assumed that, for a certain time, transportation demand through the bottleneck exceeds supply, such that a queue develops at the entrance of the bottleneck. After the rush hour, the queue decays due to reduced demand. One-dimensional bottleneck cases can be treated analytically.

Kuwahara and Newell [89] extend the idea to a single core city: It is assumed that workers want to reach a point-like central business district (CBD) during the morning rush hour, and that queues develop on all arterials leading to the CBD.

One option to deal with the queuing problem would be to make the assignment procedure time-dependent, and then to use a simulation model to calculate the cost functions. However, as long as the simulation model is driven by turn counts, the non-queueing planning process and the simulation are incompatible. It is again the same problem as above, that, for example, all people who are supposed to make a left turn at a certain intersection according to the assignment can be caught in a queue. Meanwhile, the turn-counts keep on sending the predetermined percentage of all arriving people to the left, which then is wrong. And the error is likely to amplify during later time-steps. Indeed, differences between the simulation-based assignment and reality are reported [164].

One way out is to use a simulation model driven by individual trip plans in order to calculate the cost function. The main reason why this is not done is that conventional microscopic models, when used with trip plans, are computationally prohibitive, especially in an iterative assignment loop.

On the algorithmical side, it is only recently that congestion/queueing has been taken into account. See the collection of articles in [126]. However, each of these papers has at least one of the following restrictions: (i) No algorithmical solution is proposed. (ii) The algorithm is only valid if one has a single destination. (iii) The algorithm does not take into account that the traffic pattern will vary during the driving time; in other words: A path is assigned to a driver when she enters the network, and this path is optimized *based on a snapshot of the current traffic situation.* Each of these restrictions prevents the practical application for the dynamic congested traffic problem. See [129] for a a framework for a dynamic treatment of multidestination traffic networks.

8.5 Cellular automata models for traffic in road networks

8.5.1 Two-dimensional cellular automata models on a grid

Two-dimensional cellular automata models [30, 112, 119, 166] are useful to understand the *grid-lock transition* (also called jamming transition by several authors) occurring on overcrowded road networks: A blocked intersection leads to a queue, which blocks an intersection somewhere else, etc., until a large portion of the network is in grid-lock.

The CA models represent traffic on a two-dimensional grid where vehicles can move to adjacent grid-points in each iteration. In all the models so far, the vehicle dynamics follows the deterministic asymmetric exclusion process, i.e. $v_{max} = 1$ and $p = 0$ in the language of Chapter 3. Meanwhile, they employ more sophisticated rules for making turns at intersections.

The models show that a grid-lock transition occurs at a certain vehicle density ρ_c; above that density, most or all vehicles are caught in a few very large grid-lock clusters; below that density, no stationary grid-lock clusters exists. Yet, in these models the maximum flow is given entirely by the capacities of the intersections, which is not always realistic, e.g. for arterials. Nagatani [118] investigates a case where a portion of the intersections is two-level.

8.5.2 Road networks composed of cellular automata segments

A more realistic, but more laborious, approach is to connect one-dimensional CA traffic flow segments to road networks. This is similar to conventional microscopic traffic flow simulation [124], except that the CA model will run much faster (see Chapter 11).

As mentioned in Chapter 2, Schütt [151] describes a tool-box of urban and rural roads and freeways plus the corresponding intersections in order to construct road networks.

Rickert [141] reports an implementation, based on this work here, with two lanes in each direction for arbitrary freeway networks. He describes results of extensive simulations with a network implementation of Northrhine-Westfalia.

Both models do not, however, implement trip plans, the missing of which have been identified as one of the major shortcomings in current methodologies.

I will, in the next two chapters, describe results obtained with a single-lane network implementation which includes trip plans.

Chapter 9

Predictability vs. network performance

This chapter is the first of two dealing with traffic in road networks. It considers a simple road network where, using a basic model of human decision behavior, issues such as User Equilibrium and Congestion Pricing are investigated in a simulation. An important results will be that traffic management makes traffic indeed more efficient, but also more fluctuating in certain quantities and therefore less predictable.

The chapter starts with the introduction of a new quantity, the vehicle-to-vehicle fluctuations for trip times. Then it introduces the specific network example which will be used, discusses traffic management aspects and the resulting set-up for the simulations. Section 9.3 presents the simulations results, followed by a summary.

9.1 Variability and predictability of travel times

Measuring the life-time distribution of traffic jams as in Chapters 5 and 6 is convenient for a theoretical understanding, but it is not very useful for everyday traffic. The probably most important reason for this is that life-times of jam-clusters are practically not amenable to measurements in reality.

A quantity which is much easier to measure and which is extremely relevant in the context of transportation management is the individual travel-time and its variation from vehicle to vehicle using the same route. For the following simulations, we still use a closed loop of size L. We define a subsegment of length $l < L$ and measure, for each car, the time t_l between entry and exit of this subsegment.

The relative variation of travel-times is defined as

$$\sigma(t_l) := \frac{\sqrt{\langle (t_l - \langle t_l \rangle)^2 \rangle}}{t_l} .$$

$\langle \ldots \rangle$ denotes the average over all cars during the simulation; $\langle t_l \rangle$ therefore is the average travel-time for all cars during the simulation.

This can be seen as a measure of the predictability of the trip time. When a vehicle enters the measurement section, one would like to tell the driver how much time he will need. One would tell him $\langle t_l \rangle$ as expected time; and for the expected relative error of this prediction one would use σ.

Results of these measurements as a function of density are shown in Fig. 9.1. A system of length $L = 10^3$ is used where trips along a designated subsegment of $l = 100$ are measured. The simulation runs, after 10^4 iterations to let the transients die out, for 10^5 time steps, and every time a car finishes

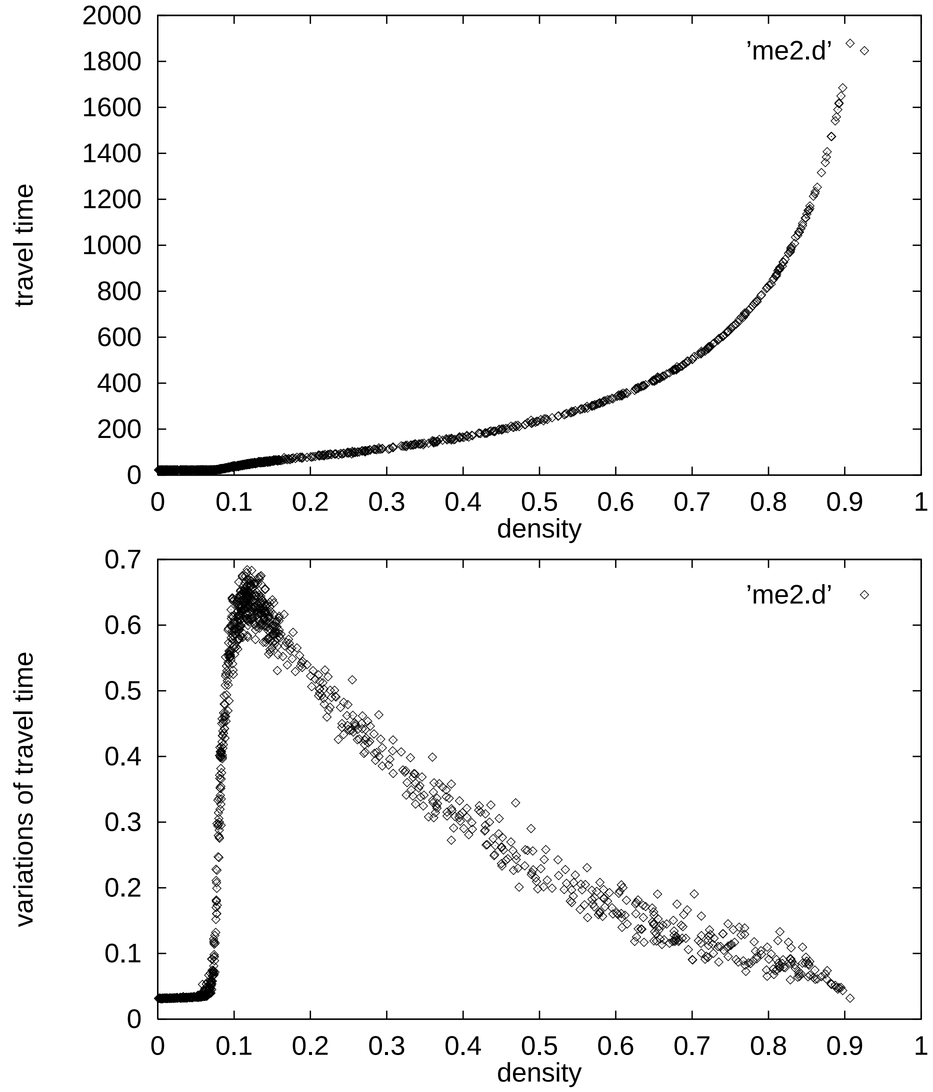


Figure 9.1: Travel time and variations of travel time as a function of density. System size $L = 10^3$, length of traveled subsection $l = 10^2$, measured time $T = 10^5$ time-steps.

a complete travel along the measurement subsegment, its travel time is taken into account for the average.

One clearly sees that both the travel time and the vehicle-to-vehicle fluctuations are approximately constant up to a density around 0.09. There, the travel time starts to rise as a function of density. Meanwhile, the fluctuations go up very steeply and reach a maximum near $\rho = 0.11$. In other words, one can not only show that the region of maximum throughput shows near-critical behavior in a theoretical sense, but also that this behavior has practical consequences: It implies that, passing from slightly below to slightly above capacity, one comes from a regime where the travel time is predictable with an accuracy of approx. ±3% to a regime where the error climbs up to more than ±65%.

9.2 A simple transport network

Let us now move away from the single lane closed loop system to a single lane highway network with ramps connecting the different segments. The travelers on this network have route plans so that they know which ramps they need to exit to reach their individual destinations. We assume that each traveler always has the same origin/destination pair. Each traveler remembers the last travel time for each alternative route between his or her origin and destination. The network may have traffic density sensors at specified locations which can be used to identify congested areas and perhaps introduce toll for the use of such links. The travelers are able to re-plan depending on their aggregated transportation costs which is their remembered travel time plus eventual toll. Such a sensor setup is an example of a (Advanced) Transportation Informational System (ATIS), and the introduction of toll for the use of highly congested links is a simple example of an (Advanced) Transportation *Management* System (ATMS) [55, 70]. The rationale behind such a toll policy is to make the highway traffic more efficient by pushing a larger part of the system towards the density corresponding to maximum flow. Interestingly, this implies that more traffic intentionally will be moved into the critical regime as defined above which in turn will increase the fluctuations of the travel times as well as the non-predictability of transportation system dynamics. This effect is the topic of this section.

9.2.1 Ramps

In order to simulate a network, one first needs a reasonable algorithms for transferring vehicles from one road to another at junctions. This involves two parts: Including the vehicle into the traffic stream on the target road; and then deleting it from the source road.

Unfortunately, introducing an additional car into a given traffic stream can cause problems. Just adding the inflow to the traffic on the main road easily leads to disturbances which (i) block the traffic on the main road, and (ii) lead to an outflow, downstream from the ramp, which is *below* capacity. For this reason, we chose an algorithm where access to the main road is only possibly when there is sufficient space between vehicles. We believe that this is realistic enough to represent metered ramps (i.e. ramps with regulated access), and since we are often concerned with the analysis of future traffic systems, it seems, in a first step, appropriate to model a technically advanced traffic control system here.

The algorithm works as follows. Imagine a ramp, as in common experience, as two parallel stretches of road; these parallel stretches have a length of 5 sites in the model. The target stretch is part of a longer road and therefore is connected at both ends, whereas the source stretch is only backwards connected. If there is a vehicle (velocity v) on the source stretch, then

- it looks, on the target stretch, for the next car ahead (which may be its neighbor; $\leadsto gap_{forward}$); and
- it looks for the closest car behind on the target stretch ($\leadsto gap_{backwards}$).

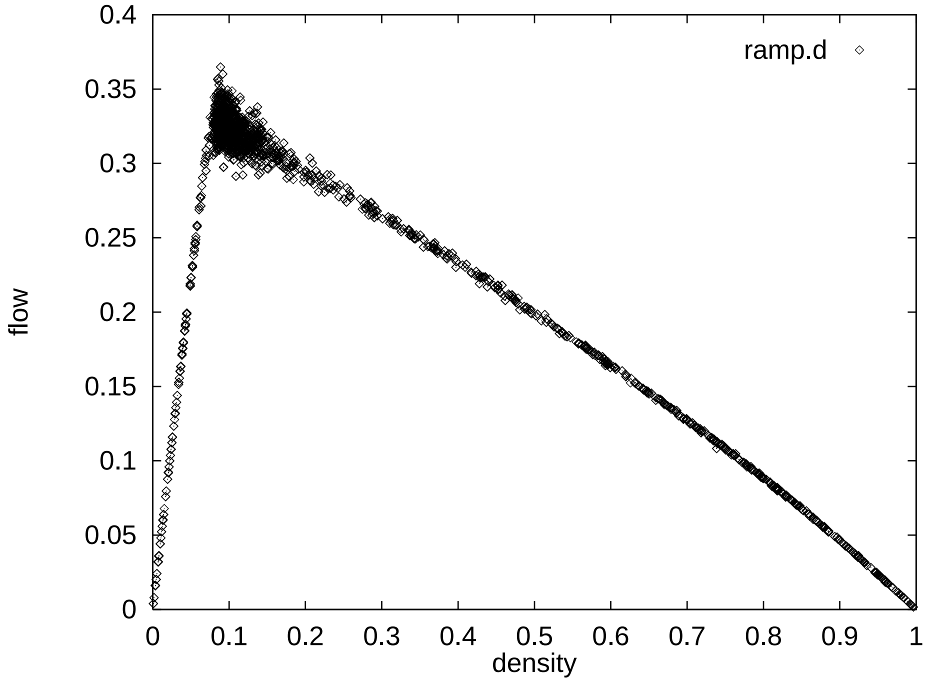


Figure 9.2: Fundamental diagram for ramp. A circular segment of length $L = 10^3$ may be partially bypassed by a second segment. 50% of the traffic uses the bypass; at the end of the bypass, it again merges with the main stream of the traffic. The measurements were taken at the part of the main segment where no bypass exists.

- Then the following rules are applied:

```
if ($gap_{forward} > v$ & $gap_{backwards} > v_{max}$)
    change_lane
    $v = \max(v_{max}, gap_{forward})$ on new lane
else
    $v = 0$
    $(*)$
endif
```

One may imagine that this is emulating a ramp metering system, where a technical device upstream of the ramp determines where to fit in a car. The car then gets a green light and arrives at maximum speed, in between two other cars on the target road.

In line $(*)$ one has to take some precautions that the vehicle really does not move, which depends on the algorithmical structure.

The details of this algorithm will probably not matter for our results, as long as it allows maximum flow downstream from the ramp. That this indeed is the case is shown in Fig. 9.2, which may be compared to Fig. 3.6. It gives the fundamental diagram for a system with two road segments where one is a closed loop and the other one provides an alternative route for a certain length, connected to the main road by one exit and one entry ramp. Half of the vehicles use this alternate route. Density and throughput are measured on the undivided part.

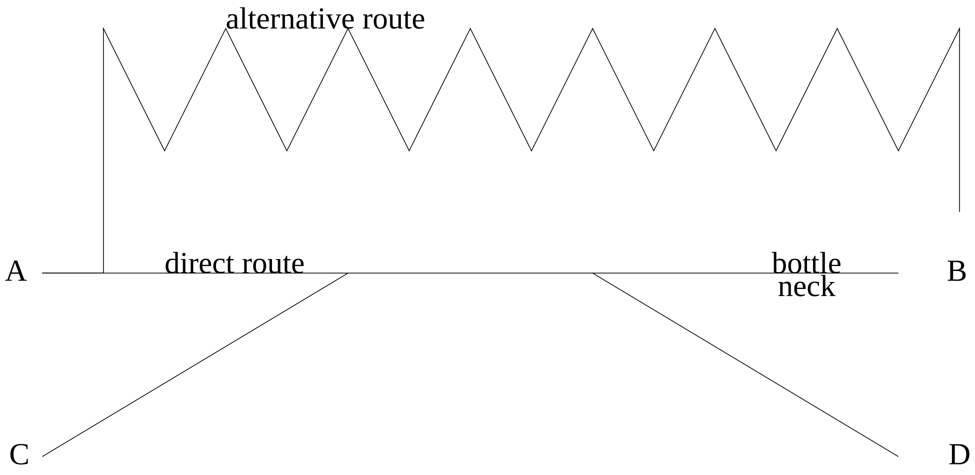


Figure 9.3: Schematic sketch of the network used for the simulations. Vehicles drive from A to B and can choose between the direct route and the much longer alternative route. On the direct route, the encounter a bottleneck. Other vehicles drive from C to D.

9.2.2 Nash Equilibrium versus System Optimum: An example

The conflict between Nash Equilibrium (NE) and System Optimum (SO) can perhaps best be illustrated in terms of a simple transport network example (a variation of [22]), especially as we will use the same example for simulation experiments later on.

Imagine a road from A to B with capacity q_{max}, with a bottleneck with capacity $q_{bn} < q_{max}$ before B (Fig. 9.3). Further imagine that there exists an alternative, but longer route between A and B. On the direct route from A to B additional travelers from C have to go to destination D. First assume that there are no travelers with origin in C.

If many drivers are heading from A to B, they will, without knowing anything about the overall traffic situation, all enter the direct road. In consequence, a queue builds up from the bottleneck.

A Nash-Equilibrium is defined as a situation where no agent (= driver) can lower his or her cost (= decrease travel time) by unilaterally changing behavior. Assuming that the drivers have complete information, this implies that, in the NE, their waiting time in the queue exactly compensates for the additional driving time on the alternative route.

Now assume that there are additional travel demands from C to D (see Fig. 9.4), with the exit to D lying shortly before the bottleneck. Obviously, this traffic is suffering from the bottleneck queue upstream (= left) of the bottleneck, and from these travelers' point of view it would be much better if the queue were located to the left of the ramp that the travelers from C use to enter the link. Note that moving the queue further upstream does not make any difference for the NE of the drivers originating in A.

This could, for example, be achieved by a traffic light near the entrance ramp from C. This light would restrict traffic flow from A to the bottleneck capacity. As a result, traffic between the traffic light and the bottleneck would flow freely, and the traffic from C to D would be much less disturbed.

Note, however, that in more complicated situations such actions are no longer possible. In our case, already some travel demand from A to D would make this traffic light a doubtful solution. And note, in addition, that the *steady state* optimum solution for our system would be to send all A-B travel via

```
10: ======================================================================================================================================
                   ........................................ // ..................................................................
                   .                             (alternative route)                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
 (A)          #.....                               (direct route)                                          V1V1V1V1V1            (B)
#####04.....5.....3....3.....4.....1.4......5......5............................................................................#####
                                  .....#                                                #.....
                                  .                                                          .
                                  0                                                          .
                                  0                                                          .
                                  .                                                          .
       (C)                        .                                                          .                             (D)
      #####0.3....4.....4.....2...1                                                          .............................#####
30: ======================================================================================================================================
                   ................4....................... // ..................................................................
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
 (A)          #.....                                                                                       V1V1V1V1V1            (B)
#####1.2...1.01..2..1...2..3.....5.......4.......5......5...........5.......4.......4.....4......4...........0..1...1...........#####
                                  .....#                                                #.....
                                  .                                                          .
                                  .                                                          .
                                  0                                                          .
                                  0                                                          .
       (C)                        0                                                          .                             (D)
      #####0.01.1..0.01.01.00.00000                                                          ...................5.........#####
210: =======================================================================================================================================
                   .......5................................ // ............4.........5...........................................
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   .                                                                                                            .
                   4                                                                                                            .
 (A)          #4....                                              (*)                                      V1V1V1V1V1            (B)
#####0.4....4.........4................4......2...2...1.2.....000.000000.1...3.......4.........4........3.....1..1..2...........#####
                                  .....#                                                #.....
                                  .                                                          .
                                  .                                                          .
                                  2                                                          .
                                  .                                                          .
       (C)                        1                                                          .                             (D)
      #####03....01.1.1.01.0000001.                                                          ........................4....#####
```

Figure 9.4: Schematic network representation with traffic, showing time-steps 10, 30, and 210 of a particular simulation. Traffic entering at (A) is bound for (B) and may use the "direct route" or the "alternative route". Traffic entering at (C) is bound for (D). The bottleneck is denoted by `V1V1V1V1V1` (maximum speed 1). One observes that the traffic coming from (C) has difficulties entering into the main stream; and—in time-step 210—a disturbance denoted by (*) has traveled backwards from the bottleneck.

the long route: In this way, throughput is maximized. However, for rush hour traffic, such a solution is no longer optimal.

A rather general way to push a traffic system from an NE towards an SO is to keep the density on each road at or below $\rho(q_{max})$, the density of maximum throughput. Then there would not exist queues anywhere in the system, since queues are regions of high density, This would ensure that additional traffic which is not using the bottleneck could proceed undisturbed. Note that this could for instance imply (in the limit of a perfect implementation) that drivers have to wait to enter the road network until sufficient capacity is available for them.

9.2.3 Egoistic drivers: Travel plans and individual decision logic

In our simple network, there are only two different types of travelers: Travelers from A to B, and travelers from C to D. Travelers from A to B can choose between the direct and the longer, alternate route. In order to make decisions, each AB-driver remembers his or her last travel-time on each of the two routes.

A traveler calculates expected costs [3] according to

$$cost_{direct} = toll + \alpha \cdot t_{direct}$$

and

$$cost_{alt} = \alpha \cdot t_{alt}$$

where $cost_{direct}$ and $cost_{alt}$ are the expected costs for the two route choices, *toll* is the toll for the current day (see below), t_{direct} and t_{alt} are the remembered travel times for each route, and α is a conversion factor which reflects trade-off between time and money. α could be different for each driver, but is uniformly equal to one in this work. α reflects "standard values of time", VOT, which can be looked up for traffic systems [6].

Then, each driver chooses the cheaper route, except that there is a 5% probability of error (which gives each driver a chance from time to time to update her information about the other possibility).

As long as the traffic dynamics is deterministic and completely uniform, this scheme leads to a Nash equilibrium [3]. However, in our case of stochastic traffic dynamics, this is no longer true: There is no easy way to define complete rationality because the interactions become too complicated. It is therefore better to speak of 'bounded rationality', which means that individual decisions are plausible instead of mathematically optimal. By dealing with stochastic traffic dynamics, the notions of economic equilibrium theory have to be used with care.

9.2.4 Space-time dynamics

Before we discuss how to determine the toll, we shortly turn to a space-time plot of the direct route from A to B (Fig. 9.5). As before, vehicle movement is to the right and time is downward. The figure contains the first 300 time-steps, and then time-steps 2000 to 2950.

The major dimensions of the system are:

- direct route from A to B: 1021 sites (full size of plot)
- cars leave for the alternative route at (a)
- cars coming from C enter at (b) and leave again at (c)
- the bottleneck ($v_{max} = 1$) is at (d).

20% of the A-B vehicles are preselected to leave at the junction for the longer route, as can be seen in the picture by a change of the gray shading at (a). The entry-point of the C-D (b) vehicles is marked by the permanent existence of a disturbance, which is very often connected to other disturbances which travel "backwards" through the system.

The point of exit for the C-D (c) vehicles is covered by dense traffic most of the time, but it may be seen near the top right of the figure as a change in gray shading and as a sudden stop of some trajectories.

The bottleneck (d) is visible at the very right edge of the figure, where the trajectories of the vehicles are diagonally pointing downwards to the right.

The striking feature of this picture is the graphic illustration of the highly dynamic and nonlinear structure of traffic patterns. Vehicles do *not* wait orderly in front of the bottleneck, but instead self-organize into backwards moving jam waves. If one of these waves reaches back into an area with higher density (in our case upstream, i.e. to the left, of (c)), then the survival probability of this jam wave suddenly becomes much larger, and it may move deeply back into the system. A single snap-shot of such a traffic situation (see time-step 210 in Fig. 9.4) cannot uncover the origin of such a wave (cf. the arguments of Chapters 5 and 7). The implications to traffic measurement and modeling are important.

Figure 9.5: Space-time plot of the main segment (A-B) of the network. The cars are injected at the left. At (a), a change in gray indicates the junction where vehicles to the alternative route leave. (b) marks the position of the on-ramp for travel from C. At (c), another change in gray indicates the off-ramp to D. Very close to the right (d) is the bottleneck, together with jams emerging from this region and traveling backwards into the system.

Something similar is true for the region where the C-D traffic stream enters the main road. It is not a process where both traffic streams line up to wait until they can jointly proceed. Instead, it is often possible that the additional traffic enters into the main stream without causing a major disturbance right away. But the locally enhanced density is unstable and leads sooner or later to the initiation of a disturbance, which then travels backwards to the junction (and often beyond).

Both effects cannot be described by deterministic models, neither by microscopic [116, 117, 172] nor by fluid-dynamical [31, 133, 80] ones.

9.2.5 Congestion detection and congestion pricing

We now come back to a further determination of the tolling scheme. For simplicity, assume that the toll is based on some traffic observation of the last period (day). Let us further assume that each driver only drives this trip once in each period (day). Note that this is over-simplistic, and further investigations are needed to make it work for, e.g., workdays versus weekend.

Algorithmically, we proceed as follows: (i) The traffic microsimulation is executed for one period. Each driver updates her travel time information just after arrival at the destination. (ii) After all cars have reached their destinations, the toll is adapted according to the average value provided by the sensor. (iii) Each A-B driver makes her route choice. (iv) The next microsimulation period starts. — This results in a day-to-day evolution of the decision pattern [104]. The procedure is actually very similar to standard multi-period game-theory [5], except that we obtain the pay-off from the microsimulation and not from a predefined matrix.

A critical question remains: Where should one place the traffic sensor(s) for the determination of the toll? Placing it *inside* the bottleneck is not very useful, because traffic there is always at or below the "efficient" density (i.e. at or below the density corresponding to maximum throughput).

Intuitively, it would make sense to measure the length of the queue before of the bottleneck. However, as we showed in the last section, the dynamics of the traffic does not lead to the built-up of a regular queue but to a system of back-traveling jams instead, which makes this approach infeasible.

Since there is no obvious solution, it makes sense to start with something simple and local. It was therefore decided to use the temporal and spatial average density on the segment upstream of the bottleneck, i.e. between the exit to D and the bottleneck. A local as opposed to a global strategy is for example discussed by Faieta and Huberman in the context of traffic light control [40].

Then, the next question is, which should be the target density for the control algorithm? When one is measuring traffic upstream of a bottleneck, then stationary traffic can never reach maximum throughput: Either traffic operates at densities corresponding to flow rates lower than the bottleneck capacity, or dense traffic builds up. Traffic can only "use" the part of the fundamental diagram which is below the capacity of the bottleneck; in consequence, densities are either far below or far above the ones corresponding to maximum throughput [56, 132, 69, 172].

However, having some knowledge about the bottleneck is not really helpful: In a more complicated traffic network, it may be the case that further downstream of one bottleneck there is another one, which has even lower capacity. Or the bottleneck may be the on-ramp to a crowded major road: Here the performance of the bottleneck depends on the time-dependent and fluctuating load on the main road.

We therefore again follow a simplistic and completely local approach, which will nevertheless prove to be quite effective. Assume that the toll is operated by a local "toll agent", who does not have any global knowledge. However, she knows the fundamental diagram (flow as function of density) of *her* sensor area. If she wants to keep the traffic at maximum flow, she has to keep the density in the correct range, i.e. near $\rho = 0.08$. We implement this by the following rules:

```
if ( $\rho < 0.06$ ) then
    $toll = toll - delta$
else if ( $\rho > 0.10$ ) then
    $toll = toll + delta$
endif ,
```

where *delta* is an external parameter.

According to our arguments above, is not obvious that this approach will produce meaningful decision behavior: The toll agent tries to keep the traffic at a density regime which is dynamically impossible because of the bottleneck downstream. It is not clear, a-priori, what effect this will have, and it was one of our main points of interest how this control mechanism would work.

9.3 Simulation results: How to play traffic games

9.3.1 Technical set-up

In the following, we describe one particular simulation run in more detail. We used a network of overall size 1962 sites, composed of the following parts:

- direct route from A to B: 241 sites (smaller than for Fig. 9.5 to reduce computational demand)
- alternate route from A to B: 1570 sites (*much* longer than direct)
- connection from C to main route: 103 sites
- connection from main route to D: 48 sites
- length of the section shared by A-B and C-D-travelers: 101 sites
- thus, overall length from C to D: 252 sites
- length of bottleneck (with maximum velocity reduced to one): 10 sites
- position of the bottleneck: starts 20 sites before reaching B

The density ρ_{toll} for the update of the toll is measured between the junction where the vehicles heading for D leave the main route, and the start of the bottleneck.

We have $N_{AB} = 16000$ vehicles which want to travel from A to B, and the same amount $N_{CD} = 16000$ which want to travel from C to D. At each "day" of the simulation, they are lined up outside the simulated system in the same sequence; and they enter the system at their respective entry points as soon as the simulated traffic allows it (cf. Fig. 9.4).

When the vehicles enter the system, they already have decided on their travel plans, so they just execute these plans. The simulation runs until all vehicles have reached their respective destinations. Then the toll is updated and drivers decide on their route for the next day, as described above.

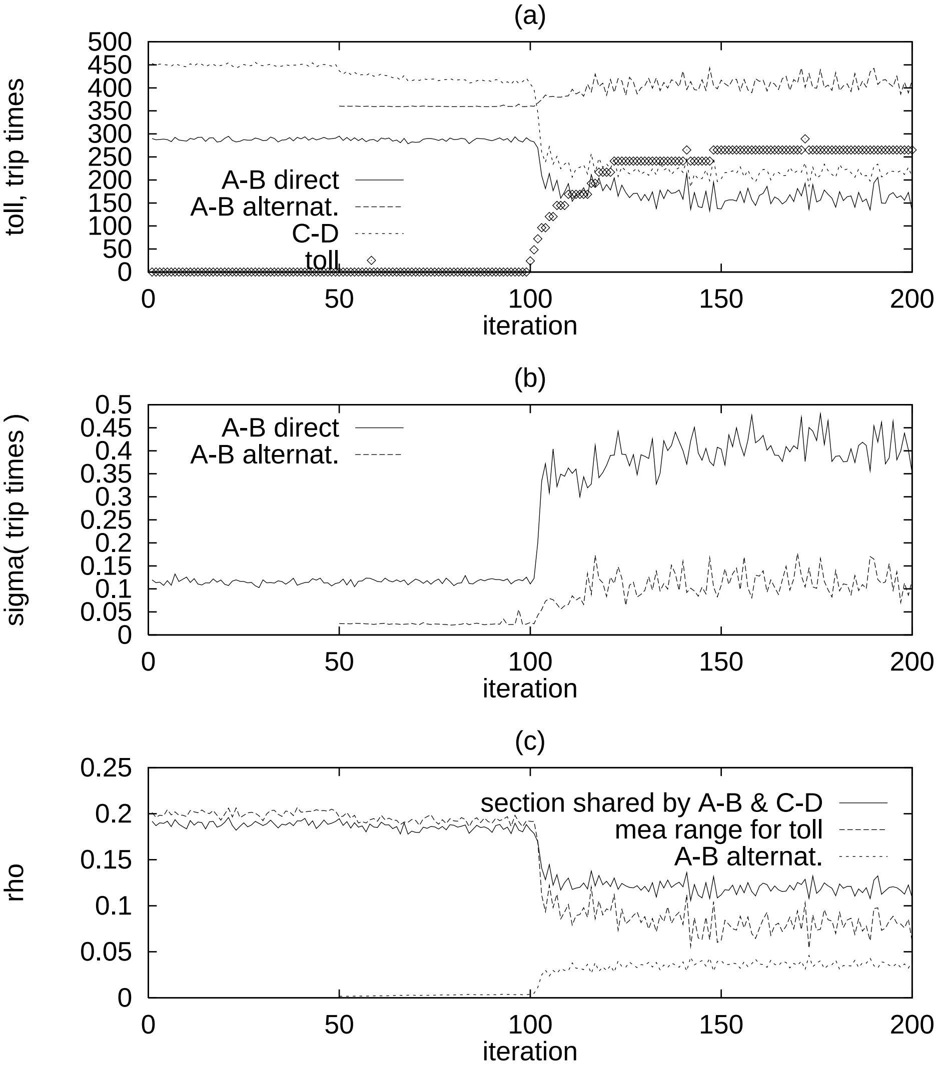


Figure 9.6: Simulation output for 200 iteration of the simple corridor network model. Time-steps 1-50: No adaption; 51-100: drivers can choose alternative route; 101-200: drivers can choose alternative route, and the toll adapts in order to keep the density at the specified level. *Top:* Average trip times for the direct and for the alternative route from A to B as well as for the route from C to D, and toll for the direct route from A to B. *Middle:* Vehicle-to-vehicle fluctuations of trip time for the direct and for the alternative route from A to B. *Bottom:* Densities on the segment shared by A-B-direct travelers and C-D-travelers, on the segment shortly before the bottleneck used for determination of the toll, and on the alternative route from A to B.

9.3.2 A simulation of 200 periods (days)

We describe 200 days of a simulation where the toll was kept at zero during the first 100 days, and in addition all A-B-travelers were forced to use the direct route during the first 50 days.

Fig. 9.6(a) shows results for the trip times and the adaptive toll, Fig. 9.6(b) the vehicle-to-vehicle variations of the trip time (as defined earlier), and Fig. 9.6(c) the day-averaged density, on selected road sections. These road sections are: (i) the section where the density for the toll adaption is measured, (ii) the section of the main road between the on-ramp from C and the off-ramp from D, and (iii) the alternative route.

Even when allowed (i.e. after day 50), not many of the A-B drivers use the new option of the alternative route. This is to be expected, since it is more than six times longer than the direct one. In consequence, travel times and fluctuations do not change much.

After day 100, the adaptive tolling starts and fairly quickly reaches a stationary value around 260 (Fig. 9.6(a)). As the "toll" line in Fig. 9.6(c) indicates, this keeps indeed ρ_{toll} near the specified range between $\rho = 0.06$ and 0.10. In addition, the density on the main segment (used by both A-B and C-D travelers) drops to around 0.11, above, but close, to the density of maximum throughput.

Travel times for C-D and for A-B-direct travelers decrease (Fig. 9.6(a)); and the toll just offsets the time gain for use of the direct route: $time_{direct} + \alpha \cdot toll \approx time_{alternat.}$; remember that $\alpha = 1$.

Vehicle-to-vehicle fluctuations (Fig. 9.6(b)) for the use of the alternative road increase from ca. 2% to around 12%, and for the use of the direct road from ca. 11% to around 42%. Moreover, the *day-to-day* fluctuations also seem to go up in all measurements.

All this is in agreement with our intuition that traffic management can indeed make traffic more efficient, but may in addition lead to higher fluctuations and, as a consequence, lower predictability, since the system is driven closer to capacity and thus to the edge of chaos.

One should distinguish between two different kinds of fluctuations: Fluctuations due to the dynamics, and fluctuations due to the learning. The fluctuations in the latter might be due to the specifics of the chosen learning scheme, especially the lack of historic information beyond the last try. More realistic assumptions about the learning and en-route information are claimed to avoid that [6, 38]. However, the results for the vehicle-to-vehicle fluctuations (i.e. the σ as defined in the text) only depend on the fact that the traffic density is driven towards the critical value. A less fluctuating learning scheme should therefore even *increase* our values for σ.

It should be emphasized that the quantity of the vehicle-to-vehicle and therefore the prediction error as defined above can only be found in a *microscopic* model.

9.4 Summary and discussion

The predictability of travel times sharply decreases when the density increases above the density of maximum throughput (near-critical regime). Therefore, traffic management will not only make traffic flow more efficient, but will also drive large portions of the system towards the critical regime. The main reason for this is that the most efficient use of a traffic system takes place when all parts operate at densities near capacity flow. Systems designed for the management of traffic flows will reroute traffic from overcrowded roads to under-crowded ones, thus driving both closer to criticality.[1] Once traffic is near the critical region, further control inputs will have unpredictable consequences.

This implies that the approximation of deterministic, predictable traffic patterns will be less and less correct the more one approaches high performance of the traffic system. In consequence, traffic assignment methods based on relaxation to (time dependent) equilibrium would no longer be meaningful: The changes in the traffic patterns due to one relaxation step would get lost in the changes due to the inherently fluctuating dynamics, and the algorithm would never converge. An open question is in how far one can replace the equilibrium quantities by statistical averages (e.g. many Monte-Carlo runs); this is a topic of future research.

---

[1] Technically, it should be remembered that traffic at maximum flow is not exactly critical, but shows critical properties up to a certain cut-off, provided by the fluctuations at free driving, as discussed in Chapter 7.

One envisaged way [102] of pushing drivers towards system-optimal behavior is to give each driver only individual route guidance instead of complete traffic information. If one doubts that this will lead to high user acceptance, then congestion pricing seems to be the only alternative. Our simulation results support the idea that already locally operating agents can achieve this in an efficient way.

In the text, the case of tolling on a specific road segment upstream of "the" bottleneck is discussed. However, this demands prior knowledge about the system. Yet, one can imagine a completely local algorithm in the following way (see also [101]): Assume that *every* road segment in the system is operated by a simple economic agent. This agent wants to keep the operation of the segment as efficient as possible, and the only measure she has is to go up or down with the toll. The agent knows the performance characteristics (i.e. throughput q as a function of density ρ) of her segment, and from this she obtains the density which corresponds to maximum flow and therefore to maximum road performance. The agent then tries to keep the density on her segment at this particular density, increasing the toll when the density becomes too high, and else decreasing it. In a real network, we would expect that the toll for most segments turns out to be zero.

This tolling scheme gives the impression that every agent locally drives her segment towards criticality (= maximum flow), but the situation is more complicated. In most cases, it is not the traffic *inside* bottlenecks which is tolled, but the overcrowded segments *upstream of* the bottlenecks. But because of the bottleneck, these upstream segments usually cannot operate at maximum throughput: As soon as the incoming flow is more than the bottleneck capacity, dense traffic builds up, and the segment switches from operation far below to far above the critical point (see above). Nevertheless, our results show that this still leads to having more parts of the network near criticality, as a result of collective effects.

In an economic context, we therefore have a local aiming for high performance, which happens to coincide with criticality. But even though the criticality very often cannot be reached locally by this mechanism, it drives the *global* system closer towards criticality: Local maximization of efficiency leads to global criticality [8].

Or in short: The fact that, in a complex system, high performance often has the downside of high variability seems also to be true in transportation systems.

Chapter 10

A path-based simulation of the freeway network of Northrhine-Westfalia

This chapter describes simulation experiments using the freeway network of the German Land Northrhine-Westfalia (NRW). Except for the different underlying network, the structure of the simulations is very similar to the last chapter.

This time, there are many travelers with different origin-destination pairs. Travelers have route plans (paths) so that they know on which intersections they have to make turns in order to reach their destinations. Each traveler has a choice between 10 different paths. Each traveler chooses a path, the microsimulation is executed according to the plans of each traveler (no replanning during the trip), and each traveler remembers the performance of his/her option.

Each traveler tries each option once. Afterwards, she usually chooses the option which performed best in the past, except that, with a small probability p_{other}, another option is chosen randomly, in order to update the information about other options.

This approach—giving each agent a set of options and let each agent act on the basis of the performances of these options—is a simplified version of Holland's classifier systems [67]. See [4] for an application of these ideas in an economic context.

In contrast to the last chapter, no congestion pricing will be examined.

10.1 Network

The following simulations are based on a digital version of the freeway network of NRW, where some lower level highways (Bundesstraßen) are added in order to prevent free ends inside the network. The code is written for parallel computers using message passing, in principle for an arbitrary numbers of computational nodes (CPNs). In practice, two Sparc10 workstations, coupled via optical link and using PVM 3.2, were used. That gives the idea that experiments such as the following are already possible with a still modest amount of hardware, and that the consistent use of parallel supercomputers will allow systematic analysis of much larger systems.

The network data which is used comes from Rickert [141] as an intermediate step of his input data preparation. The original data is a list of nodes and a list of edges, where the list of nodes contains all ramps, junctions, and intersections, and edges are the connecting segments. In a first step, Rickert has connected edges to *superedges*, where all nodes of degree two (e.g. ramps) are omitted. In a next step, these superedges are distributed across CPNs, according to a heuristic which yields plausible but not optimal load-balancing and boundary minimization:

- Sort all supernodes according to their y coordinate.
- Cut the network at a certain coordinate y_0 such that all nodes with $y < y_0$ are in one part, and all other nodes are in the other part. Choose y_0 such that the computational load is as balanced as possible. Segments which are cut by this procedure are cut in the middle, and each CPN is responsible for the half of the segment which is adjacent to the local node. By this, one can calculate the length of the network in each new part, and y_0 is chosen such that both new parts have as equal length as possible under the constraints. This is necessary because most of the computational time is needed to perform the dynamical update on the road segments.
- Repeat the procedure with the largest network part which is currently there, using the x coordinates when y ones were used last time (and vice versa). Stop when enough distributed network parts are available.

More details can be found in [141]; since the following simulations were done on two coupled workstations, effectively the result after the first step was used.

Two files are constructed from the result of the network distribution algorithm:

- A list of *supernodes*, which are the nodes which "survived" the above pruning process, i.e. which are junctions or intersections, together with geographical coordinates, and the number of the CPN where they belong to.
- A list of the *superedges*, each superedge together with the numbers of the start and end node, the length of the superedge in traffic model units, and the CPN numbers of both nodes at the ends.

This data is the network basis of the following simulations.

Apart from the network and the individual trip plans, the simulation is kept as simple and straightforward as possible. This includes the following:

- Single directional lanes, i.e. one lane in each direction. The multiplication of the computational load by using multiple directional lanes seems—apart from showing the principal feasibility, which has been done by Rickert—only be justified when one has information or assumptions which need multiple lanes. Examples for this would be the knowledge of the exact number of lanes at all locations, or differences between truck and passenger car trip demand. Since, for this example, no realistic data was available beyond the network data, and since this is the first test of this approach using trip plans inside a realistic network, the use of single lanes seems reasonable. Besides, there are currently, to my knowledge, no results of microscopic simulations of networks of this size with travel plans in the literature at all.
- Oversimplified ramps. The ramp algorithm is the same as in Chapter 9. This means specifically that there are no connecting segments between freeways: A car attempts to change from one freeway segment directly to the other according to the ramp algorithm, and if this is not possible, it stays where it is. It is obvious that this will be overestimating disturbances near intersections. However, in view of the traffic flow results, it is expected that the difference to more realistic intersections is small: Usually, traffic is either sub-critical, where disturbances decay quickly, or supercritical, where disturbances grow.

It might only make a difference exactly at maximum flow. However, it was argued in Chapters 6 and 7 that fluctuations at maximum speed result in occasional disturbances, which leads to an upper cut-off for the critical behavior. At least as a first approximation, it should not make a difference if these disturbances are provided by the driving dynamics or by the intersection dynamics.

10.2 Specific simulation setups

The specific simulation set-up is described in the following. Since no origin-destination data was available, the simulation is a plausible approximation of long distance traffic through NRW. Let us denote by *boundary segments* all segments which lead through the border of NRW and which are thus connected to the rest of the network only at one end. Let us also denote by *boundary nodes* the nodes which connect the boundary segments to the rest of the network.

For the simulation set-up, the boundary segments are replaced by *inflow segments* of equal length $l_{inflow} = 2000$. Then, each site of each inflow segment is occupied by a car. This situation looks like a mega-jam on each inflow segment, similar to Fig. 6.4, but, as we know, it has the same effect as letting enter cars through the boundaries at the maximum flow rate. Each car randomly chooses a destination, which is one of the other boundary segments. The probability to chose a certain destination depends on the air line distance between the origin and the destination segment; three distributions were used:

$$P(destination) = const \ , \qquad (\text{“random”})$$

$$P(destination) \propto distance \ , \qquad (\text{“linear”})$$

and

$$P(destination) \propto (distance)^4 \ . \qquad (\text{“fourth order”})$$

Obviously, “fourth order” is heavily biased towards long trips.

Then, each vehicle gets a list of 10 different paths to reach its destination. These lists have been pre-calculated for all used O-D-pairs, and contain the 10 geometrically shortest paths which do not use the same node twice ([113], see Appendix). In cases where this is not possible, the options are filled up by replications of a possible path. Each vehicle individually now decides which path to use. In the first day (= period), each vehicle uses the shortest path; during the subsequent 9 days, each vehicle randomly selects one of the not yet tested options. Afterwards, it selects, as mentioned in the introduction, the option with the best performance, apart from the probability $p_{other} = 5\%$ to re-test one of the other options, which is then chosen randomly.

After these preparations, the microsimulation starts. Vehicles are updated according to the standard dynamical rules, and they change segments when they are at an intersection. Thus, they enter the network via the the boundary node, where they make their first route decision according to their plans. Each vehicle then follows its plan until it reaches its destination segment, and when it reaches the end of this segment, it notes the arrival time t_{arriv}, i.e. the current iteration step of the simulation, which is used as performance criteria for this specific path. When this is the first time this specific path has been used, this is used as the first guess; for repeated trials of the same route, two schemes are used:

- **Myopic:** Each traveler remembers only the last instance, i.e.

$$t_{remembered}(option) = t_{arriv} \ ,$$

where $t_{remembered}(option)$ is the arrival time this car keeps in memory for this option, and where it bases subsequent decisions on.

- **Averaging:** An exponentially decaying memory:

  $$t_{remembered}(option) = const \cdot t_{remembered} + (1 - const) \cdot t_{arriv} \ .$$

  A value of $const = 0.7$ was used; $const = 0$ reduces to the myopic scheme.

In practice and on the level of the descriptions below, no differences between both learning schemes were observed.

After all vehicles have reached their destinations and recorded the above information, the simulation is restarted, where all vehicles have the same initial position and the same destination as before, but may choose a different path according to the rules described above.

10.3 Simulation runs

Figs. 10.1 shows an example of a simulation after 400 and 1400 iterations. Iterations are roughly equivalent to seconds (cf. Chapter 3). One clearly sees how the initially empty network is filled by vehicles coming in from the boundary nodes. Each pixel in the plot corresponds to a small region of 30 sites (225 m). Green dots denote that there is at least one car in the region, yellow stars mark slightly over-critical regions, where at least one car has velocity zero, and red triangles mark jams: The density here is above 0.4.

Figs. 10.2 demonstrates the result of the learning algorithm. Both the top and the bottom figure uses exactly the same initial configuration of cars with their individual targets. Targets have been selected according to the "fourth order" probability distribution, thus there are practically only long trips in the simulation. Both figures are snapshots of the situation after 6000 iterations (100 minutes). The top figure shows the situation when every driver follows the geometrically shortest path. Meanwhile, in the bottom figure, drivers act according to their previous experiences, i.e. they usually use the path where they were fastest in the past. Note some important differences between the figures:

- Drivers learn to use A43 from Wuppertal through the Ruhrgebiet. A43 is not contained in any shortest long-distance path.
- At the Kölner Ring, problems are after the learning not only on the eastern but also on the western side. The Kreuz Köln-West is jammed.
- A1 between Wuppertal and the Kamener Kreuz is much more crowded after the adaption, with densities above 0.4 at many places.
- At the Kamener Kreuz, there is now not only a jam for people coming from Hannover, but also for people coming from the Ruhrgebiet.

Generally speaking, people "learn" according to the programmed rules to equilibrate the jams.

For comparison, the same situation is shown one "day" later (Fig. 10.3 top). Moreover, the bottom figure shows a situation (after "learning") for a *linear* distance distribution. Clearly, the overall structure of the jams in both cases remains similar. The fact that there are less jams in the interior of the network for the linear distance distribution is to be expected: The share of short trips is higher, and these trips do not go through the interior of the network.

From these two examples and from further results (not shown), one can conclude that in spite of the stochasticity of the model many of the results are robust, as well from day to day as between different distance distributions.

Obviously, it will be necessary to replace the arbitrary O-D-pattern of these simulations by more realistic data. Yet, some of the network bottlenecks seem generic with respect to transit traffic through NRW: The jams between Wuppertal and Kreuz Kamen are well known, and, as one sees, a consequence of the missing extension of the freeway A4 beyond Olpe. This extension has since long been planned; but it leads through environmentally sensitive areas, and it is thus under discussion if it will ever be built. Note that the simulation methodology presented here can be used to evaluate the utility of such an extension, or what is needed to replace it by improvements along existing paths. Or, which traffic streams have to be reduced in order to manage with the currently existing infrastructure, and how this can be achieved.

The problems near Krefeld are due to the same bottleneck in North-East/South-West direction. It is also known that the Kölner Ring presents a bottleneck.

10.4 Network throughput

As a quantitative measure, the accumulated number of cars which have reached their destination is counted as a function of time. Fig. 10.4 shows one result, for the fourth order distance distribution. Interestingly, the network performance *de*creases after drivers have learned. The probable explanation is that the network becomes crowded in a "balanced" way after drivers learn, whereas before, some parts are overcrowded and some are rather empty. It is, for example, reasonable to believe that, in Fig. 10.2 (top), a path Kreuz Meckenheim - Kölner Ring West - Wuppertal - Kamener Kreuz - Hannover has higher throughput than in Fig. 10.2 (bottom). This is exactly the problem of the C-D drivers of chapter 9, which are trapped by other traffic.

10.5 Discussion

The relaxation algorithm leads to a reasonable distribution of the traffic streams, given the initial origin-destination-assumptions. Thus, this method is capable to do the equivalent of the static equilibrium assignment for a dynamic and congested situation.

It cannot be expected that the simple assumptions yield a completely realistic picture of traffic streams; and for an exact comparison with reality no data was readily available. This will (and has to be) a topic of further research. However, the result should be judged against the input. And in this respect it is perhaps astonishing that already such a simple input leads to a reasonable distribution of traffic streams. Moreover, the traffic patterns after adaption are robust against different statistical distributions for the origin-destination pairs, which supports the expectation that already relatively few information on realistic trip generation will yield rather realistic results.

The simulation experiments show that it is feasible to do large scale traffic simulation from lower level microscopic principles. If one sees the two extremes of modelling as, on one hand, interpolating between known or measured data points from reality, and, on the other hand, as getting data from "first principles", i.e. from structural information, then the approach here views itself more in the second category. By investing enough structural information, such as demographic data and larger network sizes, it should be possible to reach a level of fidelity which is equivalent to the level of fidelity of measurement sensors. This result is well known from simulations of physical, such as fluid-dynamical, systems. However, there is no reason why it should not also hold for socio-economic systems. A result might be that, in the long run, reality measurements should be run through a simulation model for a "consistency check" before they are "believed". If this sounds unusual, one only has to look into the area of weather prediction, where this method has been a standard for many years now [37].

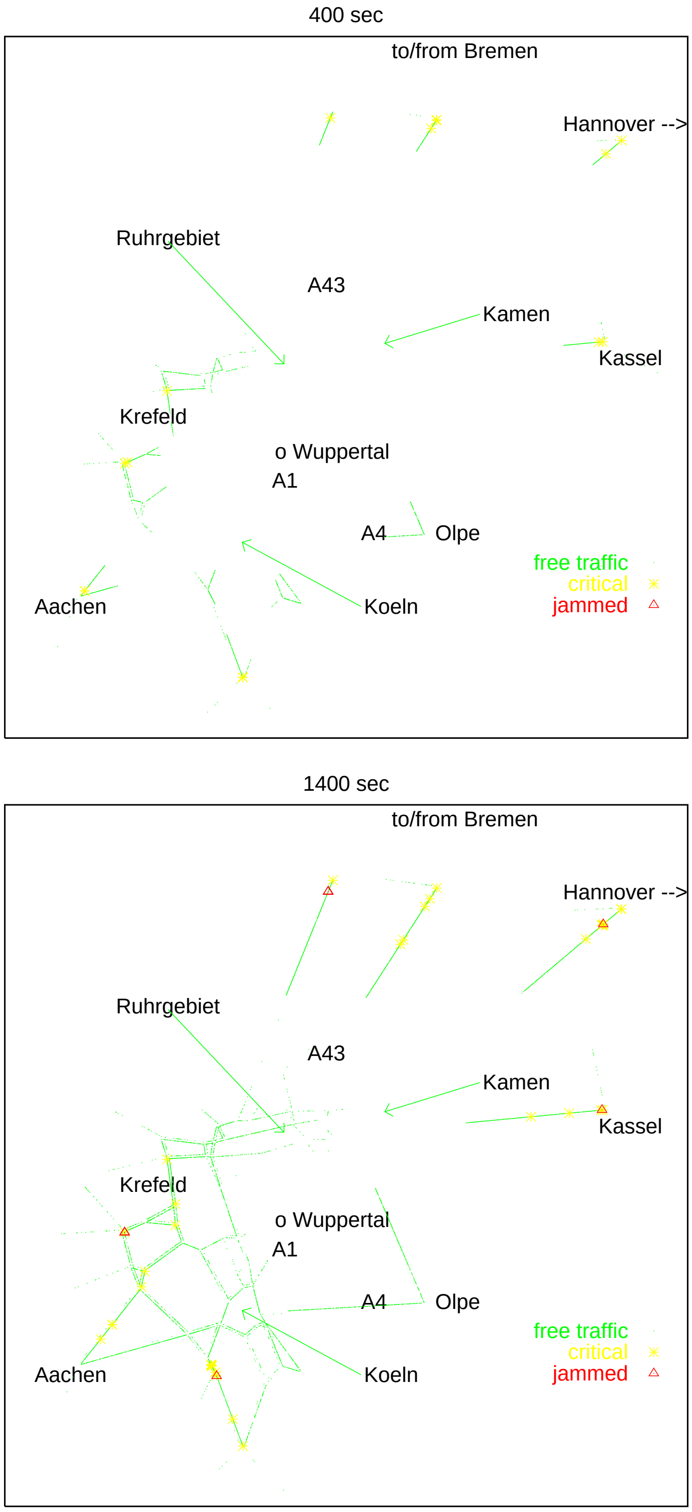
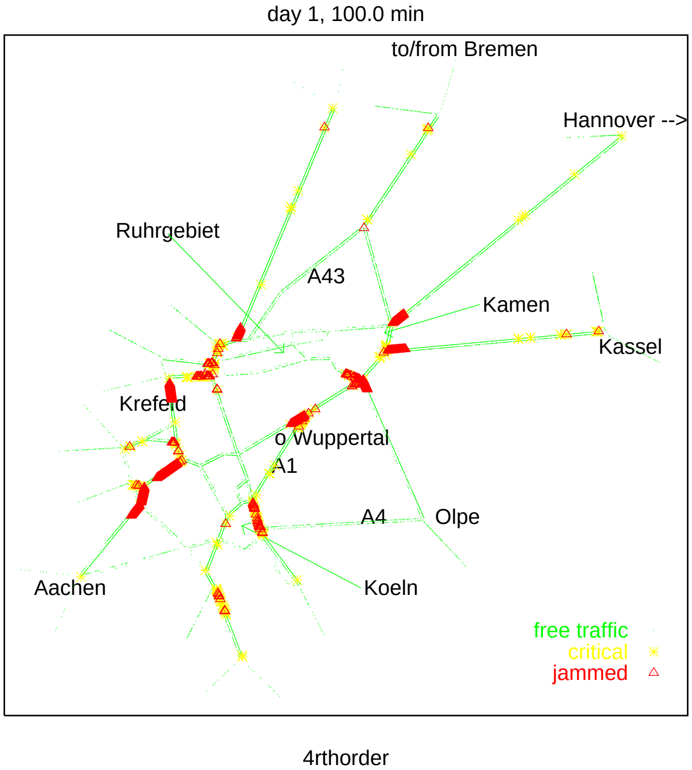


Figure 10.1: Situation after 400 iterations (top) and after 1400 iterations (bottom).



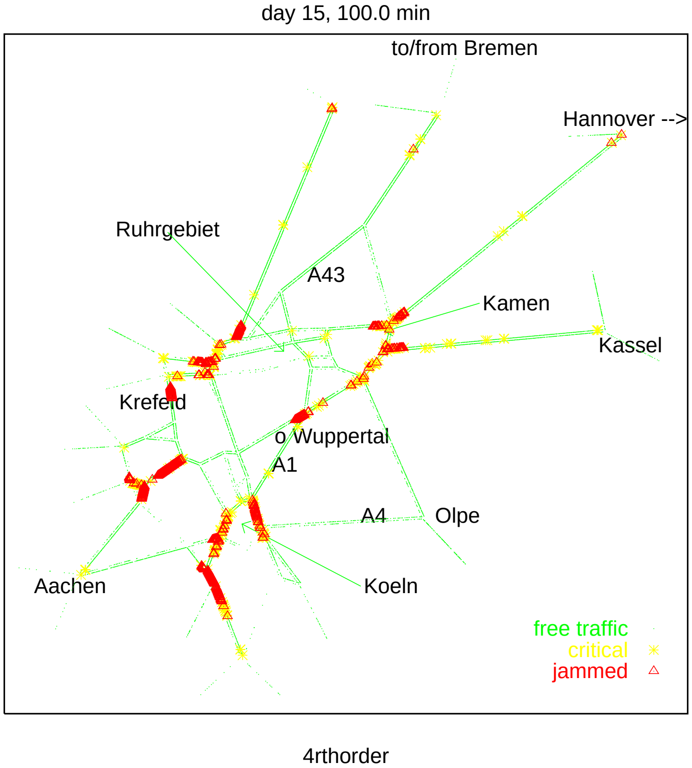
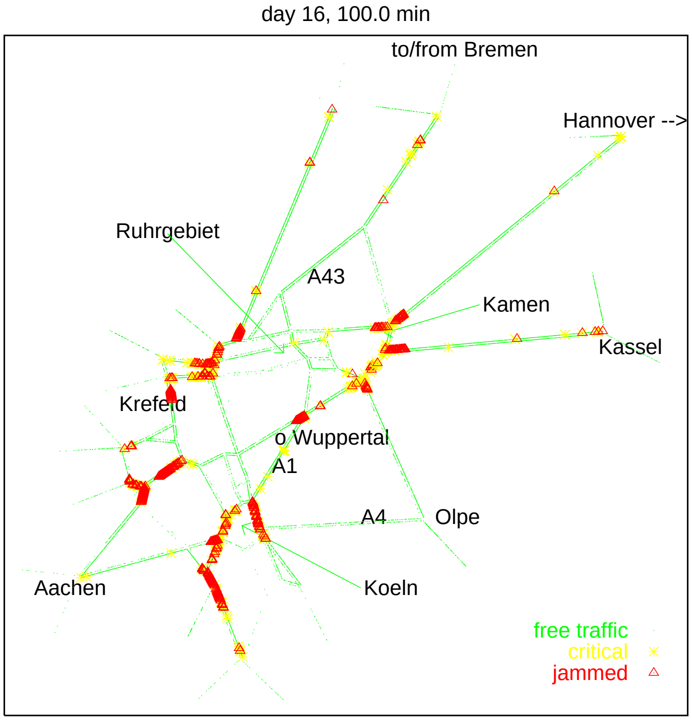
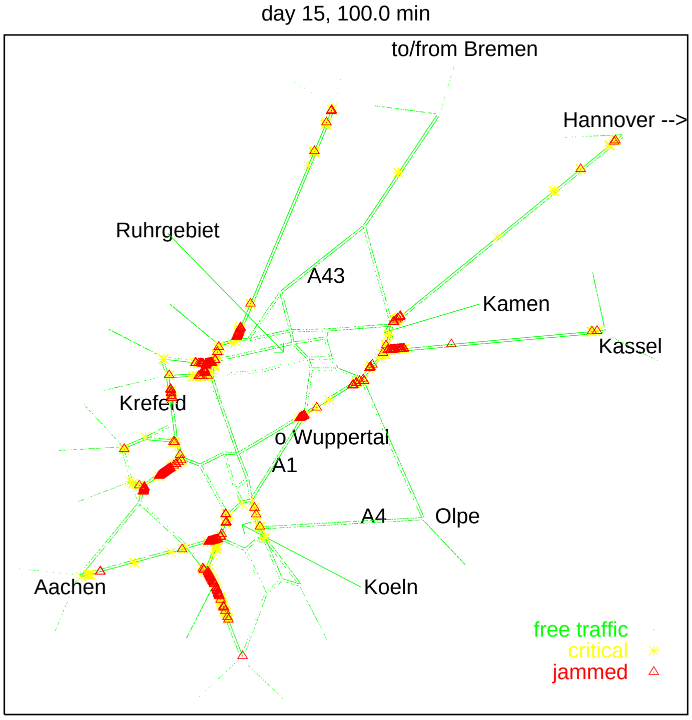


Figure 10.2: *Top:* Situation at the "first day" after 6000 iterations (100 minutes), when trips through the network are chosen with a fourth order preference for long trips, and when all drivers follow the geometrically shortest path. — *Bottom:* Situation at "day 15" after 6000 iterations (100 minutes), for the same initial conditions as for the top figure, but where drivers have "learned".



4rthorder



linear

Figure 10.3: *Top:* Same as Fig. 10.2 (bottom), except that it is one "day" later. — *Bottom:* Same as Fig. 10.2 (bottom), except that the distance distribution is linear.

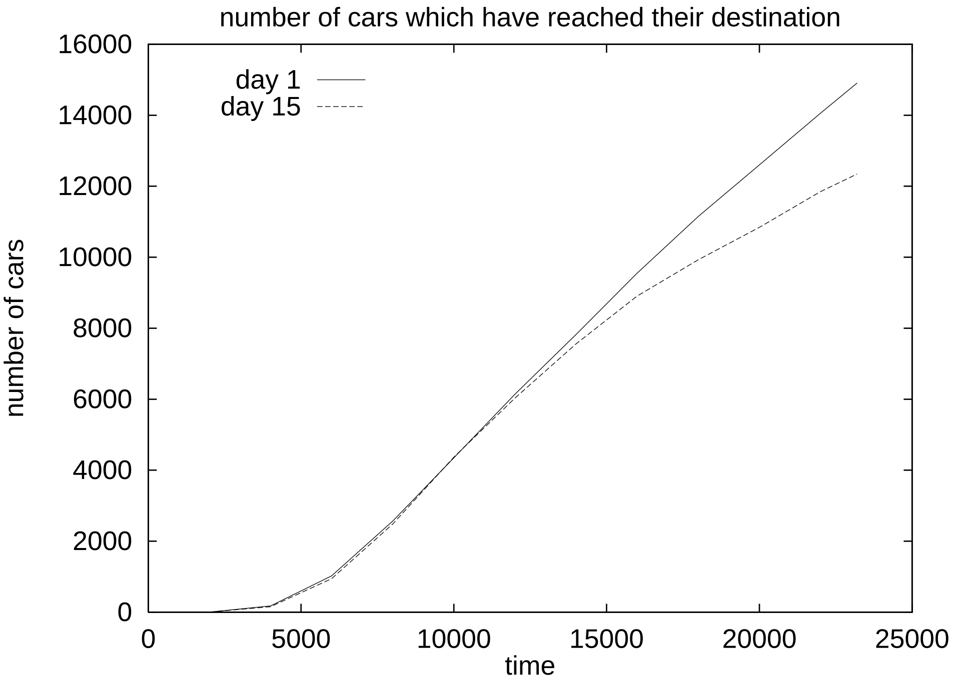


Figure 10.4: Accumulated number of cars which have reached their destination. "Days" 1 and 15 are shown. In day 1, all vehicles drive according to their geometrically shortest path, whereas in day 15 everybody has some knowledge of the travel time on different paths and usually chooses the best one. — Interestingly, the network throughput *de*creases during the relaxation, indicating that indeed something like grid-lock occurs not only in urban traffic, but can also occur in a freeway network.

Part C

COMPUTATIONAL CONSIDERATIONS, OUTLOOK, AND CONCLUSION

Chapter 11

Computational considerations

Much of the research presented in this text relies on computer simulations. Moreover, the ultimate goal is an ultrafast microscopic simulation model for large traffic networks. In such a situation, it makes sense to spend some time thinking about implementation issues. The model is developed in a way that it contains a minimal version of car following logic, and that it would run very fast on vectorizing or parallel supercomputers. To test this, we implemented the model using different algorithms on several different supercomputers.

A question is if one should do such an evaluation at such an early state of the research. Should one not first develop a realistic network model and evaluate the performance of *that* model? The answer for our project is that the early performance evaluation was helpful. The code was still short (20 lines, vs. at least 1200 for a network) which largely simplified the task of adapting to the different computers, the test provided guidelines for which techniques to use in the long run of the project, and it helped keeping the focus on computational speed because it provided benchmarks against which all more extensive models had to compare. For example, a two-lane network implementation of the German freeway network had a network overhead of less than 50%, thus reducing the computational speed by less than a factor of two. Last but not least, it allowed the evaluation against other models with regard of the question if it would make sense to continue this line of research.

This chapter first discusses details of different implementation schemes for the CA model for traffic. Section 2 contains the most important results of implementation tests on different supercomputers. Sections 3 and 4 give more detailed information for different system sizes, followed by a section about issues which came up during a parallelization attempt in the American TRANSIMS project. The last section of this chapter discusses consequences.

11.1 Implementations of the CA traffic model

For the practical coding, we considered three different approaches: site oriented, particle oriented, and an intermediate scheme. *Site oriented* directly implements the CA: A street is represented by a chain $\vec{v}$ of integers with values between -1 and v_{max}. -1 means that there is no particle at this site, whereas the other values denote a particle and its velocity. In contrast, *particle oriented* means that two lists $(x_i)_{i=1,\dots,N}$ and $(v_i)_{i=1,\dots,N}$ contain position x_i and velocity v_i of each particle i ($i = 1, \dots, N$). This is similar to a molecular dynamics algorithm, except that particles are constrained to integer positions and velocities.

Obviously, the particle-oriented approach will always be faster than the site-oriented one for sufficiently low vehicle densities. The particle-oriented approach is more flexible, and since in single-lane simulations passing of vehicles is not possible, the particle lists are always ordered, making efficient codes for all kinds of computers easy to write. In addition, an extension to continuous position and velocity is straightforward (Appendix D).

On the other hand, for the site-oriented (cellular automaton) approach single-bit coding [160] is possible. This means that the model is formulated in logical variables, which may be stored bitwise into computer words. Logical operations on computer words treat all bits of the word simultaneously, giving a theoretical speedup of b, where b is the number of bits per word (usually 32 or 64). However, the *practical* gain for traffic simulations on a workstation is much lower because the bit-oriented approach cannot take advantage of the fact that only a fraction of all sites is occupied by a particle. Nevertheless, we found that, on a workstation, the single-bit algorithm is faster for densities above 0.05 (for $v_{max} = 5$). In addition, the single-bit code runs very efficiently on a Thinking Machines CM-5 using data parallel CM-Fortran and on a NEC-SX/3 traditional vector computer.

Once passing of vehicles is allowed (multi-lane traffic), for the particle oriented approach efficient memory allocation for parallel and/or vector processors becomes more difficult, and single-bit coding for the site oriented approach becomes tiresome. These observations led to a third, *intermediate* approach. As in the site oriented approach, each site is in one of $(v_{max} + 2)$ states, but for the update only the relevant sites are considered. It turns out (see below) that on parallel but not vectorizing computers this algorithm is about as fast as the single-bit version.

In all cases, the parallelization was done geometrically, i.e. dividing the one-dimensional system of size L into p pieces of length $l = L/p$, where p is the number of CPNs (computational nodes). For single-bit coding this is a bit tricky: The standard trick of having the parallel direction(s) different from the bit-coding direction does not work any longer in one dimension. More details can be found in the appendix.

11.2 Computational speeds on different supercomputers

Table 11.1 gives an overview of the computational speeds on selected computers. All values are valid for a vehicle density of $\rho = 0.1$ and system size of $L = 1\,333\,333$ sites, corresponding to 10 000 single lane kilometers. MUPS corresponds to Mega-Updates Per Second, i.e., the number of sites updated per second divided by 10^6. These values are useful in order to compare with other implementations of similar cellular automata or particle hopping systems. The third number is the extrapolated real time system size, i.e. the extrapolated system size (assuming linear speed-up) where the computation would be just as fast as reality. The last number also is an extrapolated real time value, but if gives vehicle seconds per second, i.e. the number of vehicles which could be simulated in real time speed.

The most noteworthy features of the table are emphasized in boldface:

- Although traffic dynamics is quite different, our computational speeds are comparable to those of Ising models: Our fastest value of 533 MUPS on the NEC-SX3/11 may be compared with the value of 1050 MUPS of a very fast implementation of the (two-dimensional) Ising model on the same computer [71].
- On vectorizing computers such as the SX-3 and the CM-5, the single-bit algorithm has a notable advantage over the other algorithms. On all other machines, the intermediate algorithm is never more than a factor 2.5 slower than the single-bit algorithm.

	s.bit (F77)	particle (F77)	intermed. (C)	netw. (C)
Sparc10	0.33 sec		0.71 sec	
	4.0 MUPS	3.4 MUPS	1.9 MUPS	1.2 MUPS
	30 000 km		14 000 km	8 800 km
	0.4 e 6 veh		0.19 e 6 veh	
PVM	0.07 sec		0.15 sec	
(5× Sp10)	19.0 MUPS		8.9 MUPS	
	140 000 km		65 000 km	
	1.9 e 6 veh		0.87 e 6 veh	
SX-3/11$^{(1)}$	0.0025 sec		0.48 sec	
1 CPN	**533 MUPS**		**2.8 MUPS**	
	4 000 000 km		21 000 km	
	53 e 6 veh		0.28 e 6 veh	
GCel-3	0.013 sec		0.011 sec	
1024 CPNs	102 MUPS	211 MUPS	121 MUPS	
	750 000 km	1 550 000 km	900 000 km	
	10 e 6 veh		12 e 6 veh	
iPSC	0.016 sec		0.038 sec	
32 CPNs	83 MUPS		35 MUPS	
	630 000 km		260 000 km	
	8 e 6 veh		3.5 e 6 veh	
Paragon				
64 CPNs				
				290 000 km
CM-5$^{(1)}$	0.0077 sec$^{(2)}$		0.045 sec$^{(3)}$	
32 CPNs	**173 MUPS**		**30 MUPS**	
	1 300 300 km		220 000 km	
	17 e 6 veh		2.9 e 6 veh	
CM-5$^{(1)}$				
1024 CPNs				
				[> **1.7 e 6 km**]

(1) CPN(s) has/have vector units (SIMD instruction set)
(2) using data parallel Fortran (CMF)
(3) using message passing (CMMD)

Table 11.1: Computing speed of different algorithms on different computer architectures. "s(ingle) bit", "particle", "intermed(iate)", and "netw(ork)" mean the corresponding algorithms described in the text. For each machine and algorithm, the first table entry gives the time each computer needed to simulate a system of size 10, 000 *km*. From this figure, we derive the other three entries. The second entry is the computational speed in Mega Updates Per Second, and the third and fourth are real time limits in km and in vehicle sec / sec. For further details see text. The entries for network traffic are due to M. Rickert (personal communication).

- Already on a relatively modest machine such as an Intel Paragon with 64 nodes, our real time limit including network handling overhead (intersections etc.) is 280 000 single lane kilometers. Since, e.g., the whole freeway network of Germany is about 60 000 single lane kilometers long

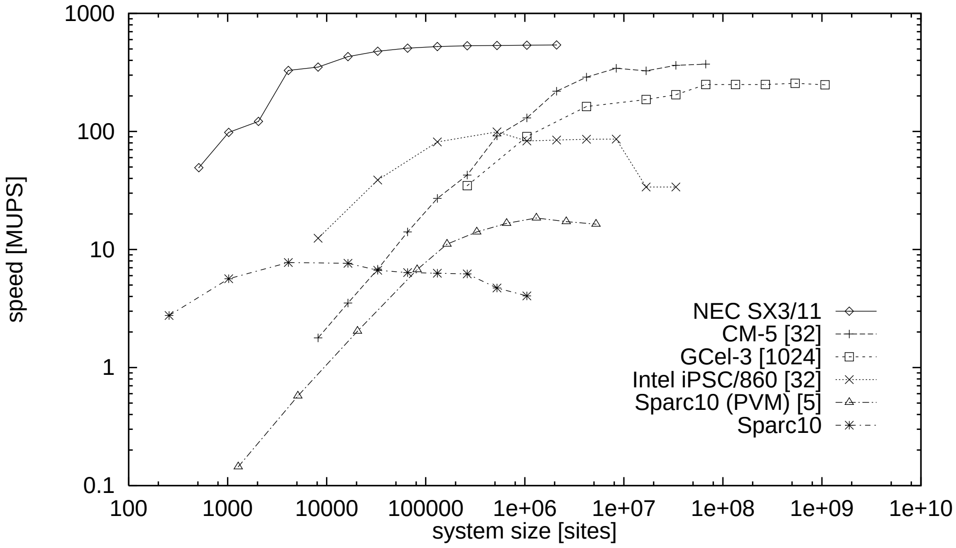


Figure 11.1: Simulation Speeds for different computers as a function of the system size. The computational speed is given in Mega Updates Per Second, i.e. the number of sites which can be updated in one second divided by 10^6. If not noted explicitly, we refer to the computational speed of the whole machine, i.e. all CPNs running simultaneously. Our fastest value of 540 MUPS on the NEC-SX3/11 may be compared with the value of 1050 MUPS of a very fast implementation of the Ising model on the same computer [71].

(12 000 km × 2 directions × 2.5 lanes), the real use of this computational speed will be (i) real time applications, where the traffic forecast has to be computed before the situation arrives; and (ii) Monte Carlo simulations, where many runs are necessary.

- The by far fastest "realistic" traffic microsimulation world-wide to date is the PARAMICS microsimulation project [181]. Their real time limit is ≈ 20 000 km on 16k CM-200, a machine which is in terms of peak performance slightly faster than a 128-node-Paragon. In other words, it seems that our not completely realistic car following logic buys us about an order of magnitude in performance.

11.3 Different system sizes

The above observations were made for one fixed system size (10 000 km). If one wants to make predictions for different system or computer sizes, one needs more results. This subsection shows some results of systematic measurements.

Fig. 11.1 gives the best computational speed in MUPS (Mega Updates Per Second) we reached for each system size on all the accessed machines.

Fig. 11.2 gives the same information in terms of the "real to simulation time ratio", which is the factor the computer model runs faster than the simulated reality. It is obvious from this figure that for fast simulations of small systems single CPN machines are far better than parallel machines. For example, a system of roads of 100 km could be simulated in 1/40000 of the real time on the NEC vector computer. For slower simulations of large systems, parallel machines become equivalent or even better due to their larger overall memory. Our largest system (on the Parsytec GC-el/3 with 1024 CPNs) had a length of nearly 10 million km single-lane road, which would correspond to 1.6 million km of

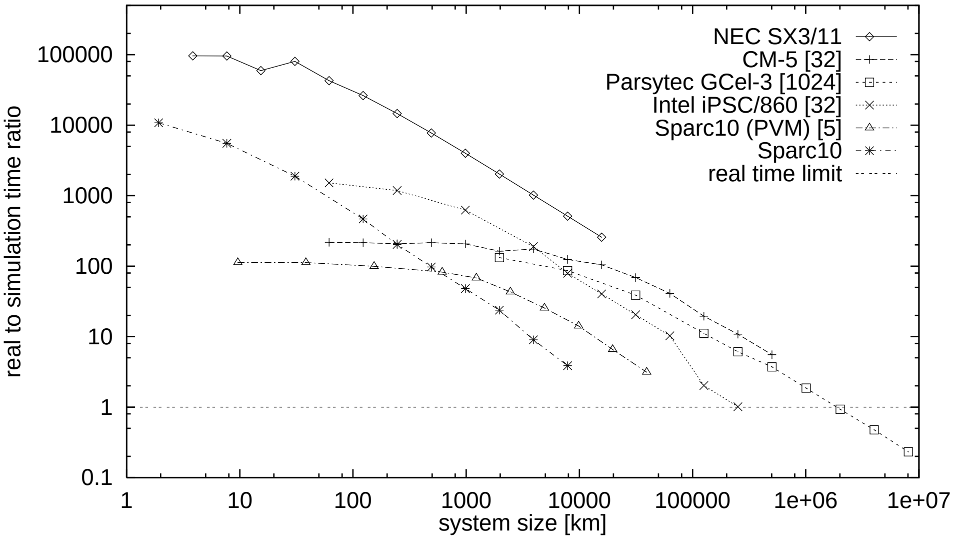


Figure 11.2: Comparison between real time and computer time for different system sizes, i.e. the factor r the computer model runs faster than reality. The system sizes where the curves cross $r = 1$ are the so-called real time limits. Real time limits for current traffic simulation models are near 1000 km ([103, 151]).

freeway with six lanes (three in each direction). In view of practical applications, this result indicates that memory consumption is not a critical issue for microscopic traffic simulations with our approach.

An efficient system size is reached when doubling of the system size demands an approximate doubling of the computer time, which is visible as a slope of -1 in the double-logarithmic plot of Fig. 11.2. For some computer architectures one needs quite large systems in order to make efficient use of the computer. In these cases, adding more processing CPNs to the parallel machine only allows processing of larger systems in the same time (scale-up), but not the same system size in shorter time (i.e., no speed-up). An especially bad case in this respect is the use of coupled workstations ("Sparc10 * 5 [PVM]"), where the single Sparc10-processor outperforms the coupled workstations up to system sizes up to more than 400 km.

The performance gain of the single-bit algorithm against the intermediate one can be seen in Fig. 11.3. The gain is relatively low (between a factor of 1 and 4) on workstation-like CPNs (SUN Sparc 10, Intel iPSC/860) and is completely lost (between 0.4 and 2) on the massively parallel Parsytec GCel–3 with 1024 CPNs. On the other hand, this gain is considerable on the vectorizing machines: We found a factor of about 200 on the traditional vector computer NEC–SX3/11 and still a factor of more than 10 on the CM–5 (the latter only for large system sizes).

11.4 Complexity and scaling

In order to make reliable predictions on scaling behavior, some theory is helpful. Results on a model for the parallel time complexity of the algorithm are found in the Appendix.

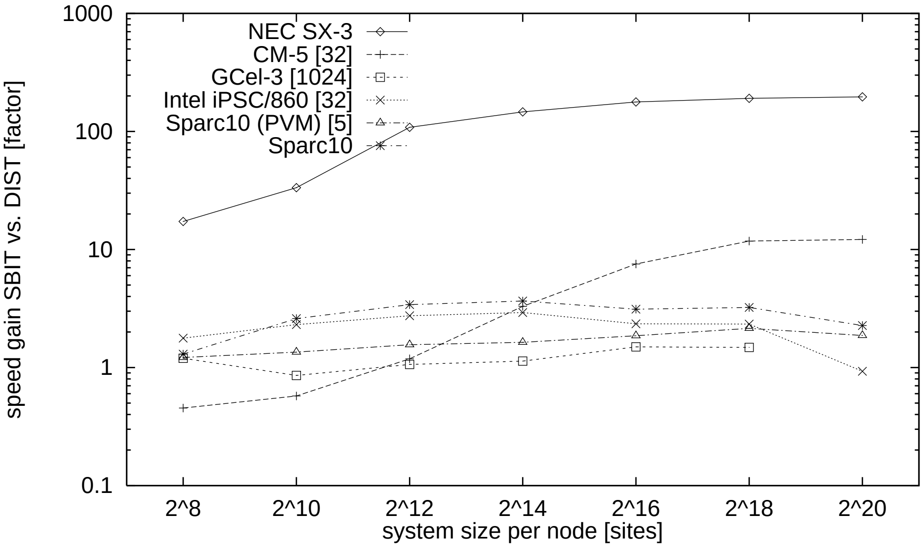


Figure 11.3: Computing speed ratio (SBIT divided by DIST) between the two main coding schemes used for our traffic model as a function of system size. Note the low gain on the massively parallel GCel–3 and the huge gains on the vectorizing computers NEC–SX3/11 and the CM–5 (the latter only for large system sizes).

11.5 A first attempt of parallelizing the TRANSIMS model

During summer 1993, we made an attempt to parallelize an extensive but preliminary microscopic network simulation model of the TRANSIMS project. Without recalling the whole procedure, I want to summarize some of the important results:

- The TRANSIMS demo version was written without planning for a parallel machine. At least in this case it turned out that the structure of the model was not compatible with a parallel implementation. Current status of software tools seems to suggest that *efficient* computer use is only possible when already during the definition of the model and the data structures some hardware realities are taken into account.
- Two technical sub-points of this observation are the following: (i) One should avoid pointers to global objects. The key example here were pointers to network segments, which were undefined as soon as one distributed the network on several CPNs.

  (ii) One should avoid sub-pointers for data which will have to be communicated between CPNs. The problem here was that each vehicle had a pointer to a data field, and some these data fields were again pointers, such as the one to the beginning of the linked list representing the travel plans. Since message passing operations only work on contiguous parts of memory, for the transfer of a vehicle from one CPN to another one had to follow all the pointers and to copy the results into a contiguous memory field, than to send it the the other CPN, and then to inverse the procedure. It looks like a better alternative to keep, e.g., travel plans in arrays of fixed length, even when most of the entries are empty because most travel plans are much shorter.
- As a result of the collaboration, it was decided to try a hybrid simulation model which can switch back and forth between the cellular automaton and a more extensive driving logic. In praxis, this means that even in the CA model, the vehicles have "to carry around" a lot of additional information such as on the travelers in the car, on the car itself, on routing plans, etc. The

most efficient implementation of this is that each site in addition to occupation and velocity information also has room for a pointer. When the site is empty, this is a null pointer, when the site is occupied, it is just the starting address of the vehicle data field. And every time when a vehicle moves from a site i to another site j, the address is copied from i to j. It turns out that this slows down the intermediate approach not more than 5%.

- Large programming projects with intelligent objects are only to handle with an object-oriented programming language. Programming large *grid-based* problems (such as fluid-dynamics) in Fortran77 is possible because of the regular program flow, but this is no longer true for other geometries such as road networks. In view of existing languages, C++ is the only portable choice.

  This is even true in view of the fact that the development of C++-compilers and tools lag one to two years behind Fortran on many parallel machines.

11.6 Consequences

Regarding all the information obtained during the implementation experiences, it seems that a bitwise approach will not be useful in the foreseeable future. Vectorizing CPUs are found less and less, and in view of the trend to use standard desktop workstation products together with a more or less powerful communication network to build parallel machines, one would expect that parallel CPNs will only reappear when they become standard in desktop workstations. But without vectorizing CPNs, the bitwise approach is at most four times faster than the intermediate approach.

In addition, the bitwise approach is tiresome for multi-lane traffic, and it does not allow for the additional address-pointer on each site which will be necessary for a hybrid approach. Moreover, for large "intelligent objects" (as opposed to simple grid-based) problems, people will turn to object-oriented programming languages, i.e. C++. One would not expect efficient vectorizing compilers for C++ in the near future even if the hardware existed.

In consequence, most future work in this area in our group will be based on the intermediate approach using C/C++. Throughout this thesis, however, a wider variety of methods was used. Some quantitative data has been obtained using the bit-oriented approach. Most of the simulations in the context of the cruise-control-limit were done using a particle oriented approach, mostly because while not being slower than the bitwise approach the data analysis was easier to code. The network examples at the end use the intermediate approach.

Chapter 12

Outlook

This outlook describes indications for further work, and extensions of the model which have already been investigated by others.

12.1 Validation

12.1.1 2-lane traffic

Rickert [141] has extensively tested a 2-lane model, including algorithms for passing, and including slower vehicles, of maximum velocity $v_{max} = 3$. A typical fundamental diagram is shown in Fig. 12.1. The density of maximum flow has clearly moved to higher densities, as is realistic. Moreover, even for low densities ($\rho < \rho(q_{max})$), the flow q is no longer a simple linear function of ρ. Instead, the following is seen: (i) Very light traffic shows free movement of all vehicles. Average speed $\bar{v}$ here is the average over the maximum speeds of slow and fast vehicles, and flow in this regime therefore $q = \rho \cdot \bar{v}$. (ii) At higher densities, but still below $\rho(q_{max})$, traffic is dominated by the slower vehicles. q therefore grows slower with ρ than in the very light traffic regime.

Moreover, Rickert has shown in simulations that the self-organization of maximum throughput in the outflow situation also holds for 2-lane traffic. Since this is one of the major ingredients for the traffic flow results of Chapters 4 to 7, this supports the conjecture that the traffic flow results are general results independent of the number of lanes.

Latour [92] reports a comprehensive investigation of different algorithms for 2-lane models.

Nagatani [117] also reports two-lane simulations. He uses a model with $v_{max} = 1$ where the velocity update is deterministic.

12.1.2 Acceleration

A remaining problem is the *flow* value at capacity, which is 0.32 vehicles per second. Realistic values are higher (cf. the discussion at the end of Chapter 3). From the traffic flow results, especially in Chapter 6, it is known that the value of q_{max} practically only depends on the acceleration behavior. Therefore, by coding a faster acceleration, it is possible to obtain a higher flow. Seemingly, a problem here is that the acceleration is already rather high: In the average, cars accelerate from 0 to 100 km/h in 10 seconds. However, it turns out that only the acceleration at low velocities really matters.

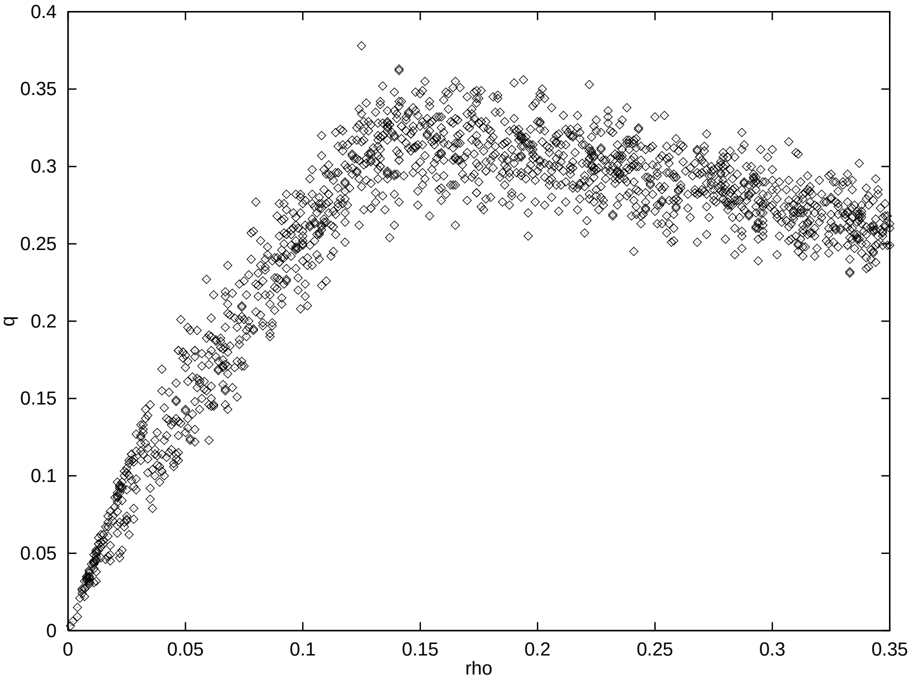


Figure 12.1: Fundamental diagram of simulated two-lane traffic, where 20% of all vehicles are slow (i.e. $v_{max} = 3$). From Ref. [141]

Preliminary results show that, by making the acceleration from $v = 0$ to $v = 2$ faster and making in slower for higher velocities, one can obtain higher flows and more realistic acceleration at the same time. Nonetheless, some more research is needed here. For example, just reducing the probability to accelerate from, say, $v_{max} - 1$ to v_{max}, leads to a wider distribution of possible acceleration times and in consequence to some undesired effects.

Migowsky and coworkers discuss an approach where the car following characteristics of each driver depends on the traffic situation around him/her [111].

12.2 Granular flow

In recent times, granular media have received a lot of attention by physicists [72]. Of main interest for traffic flow is the one-dimensional flow of granular media, e.g. sand. Poeschel [136] reports an experiment and subsequent molecular dynamics simulations of sand falling through a narrow pipe (diameter $\simeq 10 - 50$ particle diameters). Already in a tube of a length of 1.4 meters, clogging occurs and waves develop which are similar to traffic jam waves. Also the mechanism is very similar: Interaction with the rough boundary leads to small fluctuations at maximum speed, which can initiate a chain reaction, leading to a "jam". And the jam is meta-stable when the inflow rate of a jam matches the outflow rate, which is also in the pipe generically true when there are several jams in the system. Lee [96] has done more extensive simulations of the same system and has developed a theory (already mentioned in Chapter 2). Peng and Herrmann [131] use a Lattice Boltzmann simulation method for the system and find in the simulations $1/f^{\alpha}$-scaling of the power spectrum of a particle number time series, which is consistent with experimental measurements [149, 13].

12.3 $1/f$ noise

$1/f$-noise is an ubiquitous phenomenon in nature [36], and in many cases no satisfying explanation is known. Also for traffic flow, $1/f$ noise was observed [115]. The authors analyzed time series of vehicles passing at a given point, i.e. a signal which produced a delta spike every time when a vehicle passed. Pieces of this time-series are combined in a way that the average is removed from the signal, and then the power spectrum (i.e. the square of the Fourier transformation) $S_J(f)$ is calculated. It shows scaling behavior $S_J(f) \sim 1/f$ for low frequencies, and a crossing over to white noise ($S_J(f) \sim const$) for high frequencies.

Such time series analysis should receive further attention in future work: On one hand, time series measurement data is relatively easy to record, and on the other hand, scaling exponents as in the power spectrum are a better starting point for comparisons with CA-like models than most other observables.

The authors' explanation for the $1/f$ measurement is that traffic would follow the noisy Burgers equation, as shown in Chapter 2. The analysis is, however, not consistent with Krug [86], who predicts $S_J(f) \sim f^{-2}$ for the noisy Burgers equation at densities different from $\rho(q_{max})$. Moreover, in view of the results of this work, it seems that the Burgers equation is only valid for rather large length scales. It therefore makes sense to look for other mechanisms which could yield the same result. $1/f$-like behavior was found in two cases, indicating that further work in this area looks promising.

12.3.1 In the cruise control limit

For the CA model, the power spectrum was measured. This was done by first recording the time series for the number of jammed vehicles, $N_l(t)$, in a small segment of length l of a closed system, and then taking the square of the Fourier transform:

$$S_N(f) = |N_l(f)|^2 = |FT[N_l(t)]|^2 .$$

In practice, the routine `spctrm.c` of the "Numerical Recipes" [137] was used.

Since the jams have a finite drift velocity, the distribution of hole sizes in space, which has been explained in Chapter 5, is translated into the same distribution of time intervals for the activity. In particular, the hole size distribution in space translates to the first return time for jammed vehicles when sitting at a fixed position in space. It has been shown [107] that given a distribution of first return times of activity $P_{first}(t) \sim t^{-\tau_{FIRST}}$, the power spectrum scales as $S(f) \sim 1/f^{\tau_{FIRST}-1}$. Using the result $\tau_h = \tau_{FIRST} = 2$ this gives precisely a $1/f$ power spectrum for the noise. The power spectrum for the CA model with the parameter $p_{free} = 0.5, 0.005, 0.00005$ and system sizes $L = 10^3$, $L = 10^4$, and $L = 10^5$ was measured in a closed system near the critical density. As shown in Fig. 12.2 for $p_{free} = 0.00005$, $L = 10^5$, and $\rho = 0.072$, the numerical results are in general agreement with this prediction.

12.3.2 In a continuous case

In the appendix, a variant of the CA traffic model using continuous variables is described. This model displays, in a follow-the-leader situation, behavior similarly complex as the CA traffic model. Here, the first vehicle has a lower maximum velocity than all following vehicles. Following vehicles catch

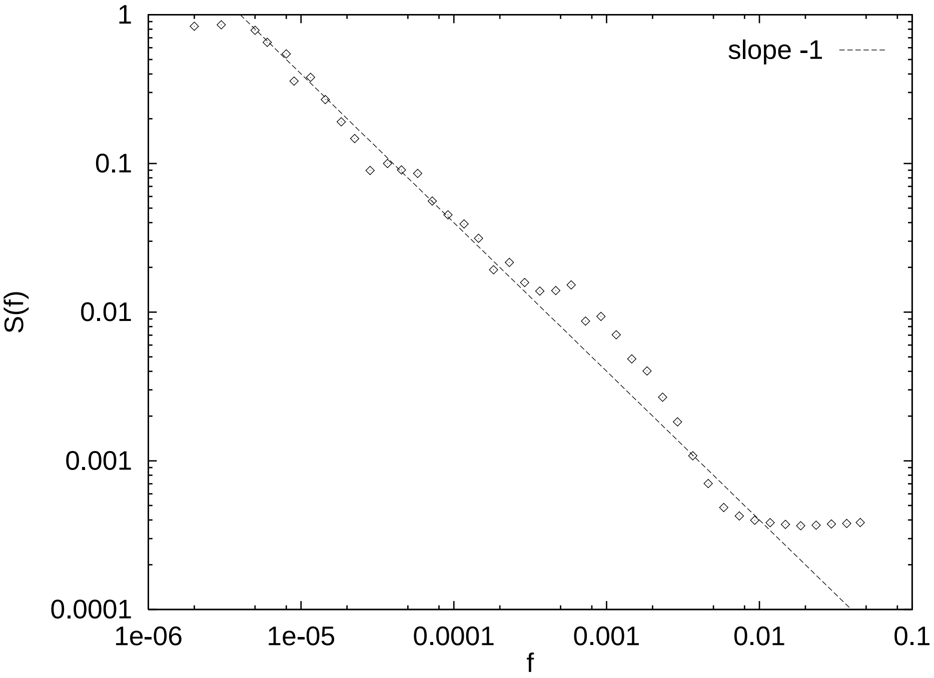


Figure 12.2: Power spectrum, $S_N(f)$, smoothed by averaging, for a closed system of length $L = 10^5$ and with $p_{free} = 0.00005$. Straight line has slope -1.

more and more up. Then, spontaneously and unpredictably, one vehicle slows down and initiates a chain reaction, which looks similar to the traffic jams produced by the CA model.

In a simulation of this model using $N = 3\,000$ vehicles, in every time step the number of cars in a small segment of length l at a position 1000 sites behind the leading car was recorded:

$$N_l(t) = \sum_{i=i_0-1000}^{i_0-1000-l} \delta_i \ ,$$

where i_0 is the position of the leading vehicle, and δ_i is 0 (1) when site i is empty (occupied). Then, the power spectrum of this time series is taken as above.

Also this measurement (Fig. 12.3) shows, similarly to the measurement above, non-trivial behavior, i.e. neither white noise ($S(f) \sim const$) nor $1/f^2$ scaling.

12.4 Phase separation

It is in some sense obvious [114] to try a description of the traffic flow results in terms of a phase separation [99]. Traffic flow at all densities has a tendency to separate into jammed regions and free flow regions, *but only above* $\rho(q_{max})$ *the jammed regions are stable.* The equilibrium is dynamic, meaning that jams are only maintained because inflow match outflow rates. The transition should then be similar to a gas-liquid transition, except that, at the transition point, the "liquid" (jammed) clusters in the CA traffic model have no preferred size, which makes the transition critical.

This is consistent with the 2-fluid approach [62, 102] which works rather well for traffic flow. In the 2-fluid approach, it is assumed that traffic flow can be approximated by having one portion moving with

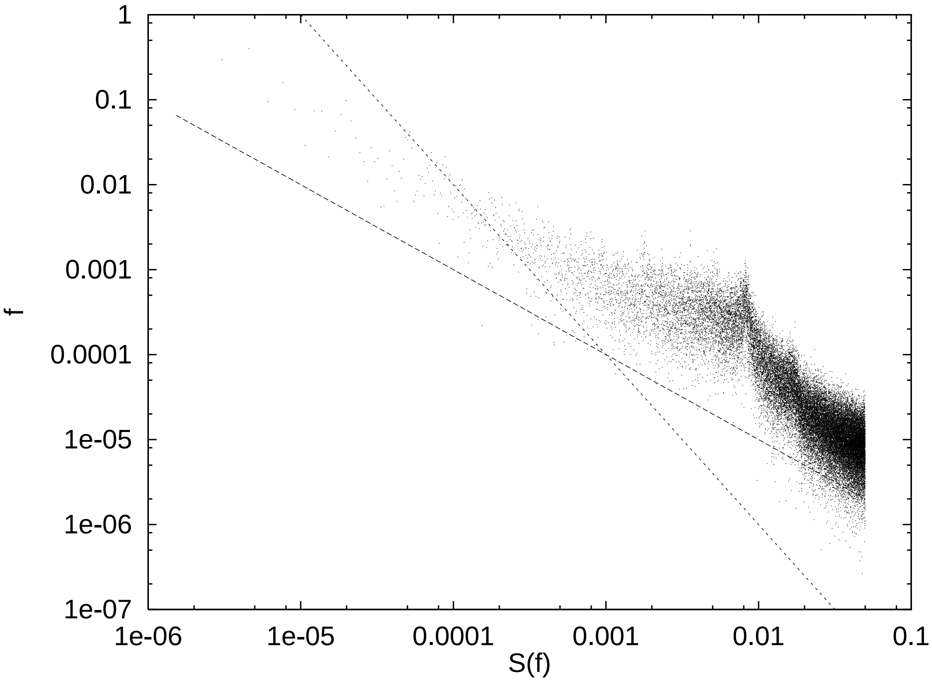


Figure 12.3: Power spectrum, $S_N(f)$, for the continuous model, at a fixed distance of 1000 sites behind the leading vehicle. For comparison, the lines have slope -1 (broken) and -2 (dotted).

maximum velocity and the rest not moving at all, i.e. that all densities above $\rho(q_{max})$ spontaneously separate into portions moving at v_{max} and other portions not moving at all. Note that the 2-fluid approach uses only the average of the phase separation dynamics.

12.5 A dynamic systems approach to transportation simulation and optimization

A general reason for a simulation to be *microscopic* is the following: As soon as simulations are no longer only used as interpolations between known regimes but as a tool to predict new situations (such as the addition or the closing of a road), *structural* (i.e. lower level) information, such as behavioral rules of intelligent objects, is much more robust than aggregated information, such as turn counts. This experience is also known from the simulation of other socio-economic systems, such as of the savings and loans dynamics of building societies [7, 79, 178].

A major reason for doing *high speed* simulations is that it makes a dynamic systems approach easier. One can analyze many different scenarios, or, in more technical terms, systematically investigate the state space of the model (which usually leads to a better understanding of the real system).

Thus, fast microsimulations provide an efficient and robust testbed for the transportation system. This is especially important because experiments with the real system are costly or infeasible. Some examples for testbed applications are:

- Development of efficient optimization techniques for traffic assignment in the non-static case. Fast computational speed means that the model can be used in an iterative loop with methods of (combinatorial or other) optimization, thus providing cost/fitness functions from a relatively realistic simulation instead as from often rather unrealistic mathematical function approximations.

- Testing scenarios such as for adding or closing roads or for the introduction of traffic information or management systems on a large scale basis. This is important for cost-benefit evaluations of such actions, and is equally useful for public service as for private companies which want, for example, invest into individual routing services.
- Evaluation of O-D estimation methods. The simulation can produce dynamic scenarios, one extracts time-dependent turn-count data, the O-D method gives *its* estimate of the O-D matrix, which can then be checked against the simulation input.
- Incident detection algorithms. Incident detection is not possible from single sensors, but only when data input from multiple sensors does not match expectations. For example, a blockage by an accident leads to too few vehicles at sensors downstream of the accident. Such algorithms can be pre-tested in a microsimulation test-bed [151, 165].

In addition, there is another reason why robust microsimulations become important when optimizing the traffic system: Especially near efficient performance, the traffic system (and supposedly many other systems) shows high fluctuations which can be treated using the formalism of critical phenomena (Chapter 9). These fluctuations and the resulting unpredictability provide an effective limitation for optimization attempts. The fluctuations can only be seen when one uses stochastic microscopic simulations, and they can only be represented correctly when the simulation is robust with respect to these fluctuations.

In addition, when fluctuations become important (i.e. when the system becomes more and more efficient), deterministic simulations start to give wrong answers. To recall an earlier example: When the fluctuating back-propagation of a jam triggers a network breakdown with a probability of 5%, then the effect of the network breakdown is only considered in the average when a stochastic model is used.

12.6 A dynamic systems approach to simulation

It is only a small step beyond this to see a simulation itself as a dynamical system [11, 12]. The question is in how far the effects one sees are caused by the dynamics of the simulated system or by the simulation itself.

The traffic dynamics in this text are (hopefully) an image of the traffic dynamics happening in reality. Meanwhile, in future work, a hybrid approach between the CA model and a more extensive microscopic model are planned as part of the TRANSIMS project. This hybrid model is planned to run on parallel machines, and it is planned to independently allocate resources between CA and extensive simulation and between computational nodes of the parallel machine, depending on the question which the simulation is supposed to answer. In this situation, the simulation definitely gets a dynamics of its own, and it is highly probable that even for the same underlying traffic system, two different questions can lead to two different system evolutions because the resources are differently allocated. It will be necessary to identify which of these effects are due to the simulation dynamics and which are due to the traffic system dynamics; and in which cases the answers are reliable and when not.

Chapter 13

Conclusion

Here, I suppose, everybody wants to read the final reason for traffic jams and how to avoid them. This will not be possible; all results indicate that the opposite is true.

This work investigates the usefulness of very fast, hardware-oriented microscopic models for traffic simulation. For this purpose, one of the best available techniques for fast microscopic simulation—the methodology of cellular automata (CA)—is adapted to the needs of traffic simulation. Streets are broken into boxes, and each box can either be empty, or occupied by exactly one car. Movement happens by "jumping" an integer number of boxes, where $v_{max} = 5$ is the maximally possible jump. Rules for acceleration and for braking constitute the dynamics. This is a fairly simple stochastic particle model, which is, especially in view of its simplicity, astonishingly realistic. It correctly reproduces start-stop-waves and fundamental diagrams (relation between average density ρ, flow q, and velocity v). This model is single-lane; Rickert [141] shows that the method is extensible to multi-lane traffic.

Two limits of the model are investigated in detail—the deterministic limit, where all randomness is taken out of the model; and the "cruise control limit", where only the randomness at full speed is set to zero.

The completely deterministic limit is equivalent to a linear, deterministic, and non-diffusive generalized Burgers equation, which is the same as the Lighthill-Whitham theory for kinematic waves with a piecewise linear "inverse V" flow-density relation. Discretizations of the Lighthill-Whitham equations are, together with piecewise linear flow-density relations, indeed in use for traffic simulation.

The cruise control limit allows stable homogeneous flows up to a certain density $\rho(q_{max,det})$. Yet, there is another important density $\rho_c < \rho(q_{max,det})$ with the following property: (i) For $\rho < \rho_c$, a disturbance decays in the average. (ii) For $\rho > \rho_c$, a disturbance (jam) grows in the average. (iii) For $\rho = \rho_c$, in the average, a disturbance does neither grow nor shrink; individual jams fluctuate according to an unbiased random walk. In consequence (Chapter 5), jams at $\rho = \rho_c$ are critical clusters in the sense of critical phenonema. ρ_c is also the density of maximum flow *when starting from random initial conditions*, i.e. $\rho_c = \rho(q_{max,rnd.ini})$. Homogeneous traffic between ρ_c and $\rho(q_{max,det})$ is just *barely* stable: The slightest disturbance leads to a growing jam (according to (ii)); and the outflow from this jam is *exactly at* ρ_c (self-organized criticality).

A third special case is to set the maximum velocity v_{max} to one. This turns out to be comparable (but not equivalent) to the well-known stochastic asymmetric exclusion process (SAEP). In the SAEP, the update is random sequential, which means that randomness enters differently than in the CA traffic model. Nonetheless, the case $v_{max} = 1$ of the CA traffic model behaves similarly to the SAEP, including the effect that kinematic waves move forward for densities below the point of maximum flow $\rho < \rho(q_{max})$, and backward for higher densities $\rho > \rho(q_{max})$. This similarity vanishes with higher v_{max}, where forward moving waves do not exist. The SAEP is, in the hydrodynamic limit, equivalent to a generalized Burgers equation with a linear plus a nonlinear term, which is in turn, after discarding noise and diffusion, equivalent to the Lighthill-Whitham theory using a quadratic flow-density relation (called Greenshields relation in traffic science).

The complete CA model behaves similarly as the cruise control limit, except that *all* configurations above $\rho(q_{max,rnd.ini})$ are only meta-stable, eventually and spontaneously separating into jammed regions and regions of $\rho(q_{max,rnd.ini})$. The latter are critical in the sense described above ("cruise control limit"). However, they are no longer *exactly* critical: There is an upper cut-off in space and time, and the conjecture from all results, especially from the $v_{max} = 1$ case, is that above this cut-off Burgers-like behavior takes over. Judging from the knowledge that single jams definitely exist in reality, and comparing with simulation results, this length scale has to be above 10 km.

This (phase) separation cannot be described by the Lighthill-Whitham theory; fluid-dynamic theories for traffic capture it only after including a momentum equation. These fluid-dynamical theories, contrary to stochastic microscopic models such as the CA, still do not capture the fluctuations around the average behavior.

For all particle models described above it is true that the outflow from a jam automatically selects the state of maximum flow, i.e. $\rho = \rho(q_{max,rnd.ini})$, which means that the critical state is selected generically.

In addition to the effects summarized so far, traffic jams in the model branch. This leads to a fractal appearance; measurements of the distribution of the holes between jams show that the structure is *barely* non-fractal. The structure could explain $1/f^{\alpha}$ noise in measurements, especially since it also shows up under rather different circumstances in a car-following model with continuous variables.

A big advantage of having invested into understanding the exact behavior of the particle model is that one can now argue in how far the model is equivalent to the well-known Lighthill-Whitham model for traffic flow. That is, the argument that the car-following logic used for the model would be too rudimentary to describe realistic traffic, is, besides by the results, also set off by the fact that the macroscopic behavior is similar to, but goes beyond, a well-accepted theory.

Moreover, the critical behavior as indicated above can be described by critical exponents. Since, in addition, there is a phenomenological theory for these exponents, it can be expected that these exponents are universal for microscopic traffic models, i.e. that they do not depend on the exact properties of the models. It can therefore be expected that more complicated models will not show a different behavior in these properties. — Yet, the model presented here is minimal in the sense that further simplifications lead to the loss of essential features such as stochastic jam dynamics.

The second part of this text describes experiments with traffic in networks. In Chapter 9, a simple network with a bottleneck and a longer route around the bottleneck is analyzed. In these simulation experiments, congestion pricing turns out to be useful for a fairer allocation of the resources (road capacity). Although this is generally to be expected, the situation is unclear in the heavily fluctuating environment which traffic near maximum flow is: The toll was only allowed to adapt much slower than the fluctuations, meaning that economic theory would be valid at best in the average; and holes in queues make state estimation problems, e.g. for setting the toll, difficult.

In addition, the simulation serves as an example of a more general problem: The traffic flow results say that maximum flow is close to a critical point, meaning that fluctuations are large. This is confirmed by measuring (in the simulation) also a more practical quantity, the relative prediction error for trip times, which is maximal near maximum flow. Since all traffic management (ATMS-Advanced Traffic Management Systems), such as congestion pricing, will push traffic near maximum flow, predictions and therefore control will become harder and harder. This general conjecture is confirmed by the simulation with the simple example network.

Chapter 10 describes simulations of the freeway network of the German Land Northrhine-Westfalia, with more than 60 000 vehicles simultaneously in the simulation. Building in part on the work of Rickert, travel plans are implemented, meaning that every car knows, when entering the system, which route it wants to take. In the simulations, cars are initially lined up on all outside nodes; they randomly select another outside node as target, and they get the set of the 10 shortest paths to their target to operate on. As with the simple network, the simulation is repeated over and over again, and after having tried out all 10 paths, during the following repetitions each individual tries to find the best path for him/herself.

The traffic then quickly relaxes towards a relatively robust pattern, where bottlenecks, such as the Kölner Ring, are equally congested on all options. Interestingly, the *performance* of the network visibly decreases during this “bottleneck equilibration process”. The result indicates that it is indeed more efficient to “lock up” some portion of the traffic somewhere outside the main network.

Despite the stochastic model, the results are relatively robust as well against day-to-day variations as against changes in the distance distribution. This robustness supports the claim that, to a certain extent, input data from reality measurements such as turn counts can indeed be replaced by structural data, such as plausible assumptions about paths.

Chapter 11 gives an overview of computational performance: Single loop, single lane implementations of the CA traffic model on many different supercomputers give indications for future work and expectations, which are confirmed or exceeded by Rickert’s 2-lane network implementations.

It is hoped that this work contributes to simulation-based dynamics system approaches to socio-economic systems, using microscopic resolution, Monte-Carlo methods, and high performance computing, and combining combinatorial and adaptive methods for modeling human behavior in the system and for eventually optimizing the performance of the system.

Part D

APPENDIX

Appendix A

Computational details for life-time measurements in the Cruise Control Limit

For the results in Chapter 5, a “vehicle-oriented” technique was used. Vehicles are maintained in an ordered list, and each vehicle has an integer position and an integer velocity. For single lane traffic, passing is impossible, and the list always remains ordered.

The following is an algorithmical version of the velocity update of the model:

```
for all vehicles do in parallel
    if ($v < v_{max}$ or $gap < v$) then
        if ($gap > v$) then
            $v := v + \mathrm{int}(0.5 + \mathrm{rnd}())$
        else if ($gap \leq v$) then
            $v := \max(gap - \mathrm{int}(0.5 + \mathrm{rnd}()), 0)$
        endif
    endif
enddo ,
```

where rnd() gives a random number between 0 and 1, and int(a) gives the integer part of a. Note that, because of integer arithmetic, expressions like $gap > v$ and $gap \geq v + 1$ are equivalent.

The simulated system is, for all practical purposes, infinite in space. The idea is comparable to a Leath-Algorithm in percolation [94], which also only remembers the active part of the cluster.

As described, a jam-cluster is surrounded by deterministic traffic. Let us assume that the leftmost car of this jam has the number i_{left} and is at position x_{left} (similar for the rightmost car). Cars are numbered from left to right; traffic is flowing from left to right.

To the right of car i_{right}, everything is deterministic and at maximum speed, and, in consequence, nothing happens which can influence the jam. Therefore, not simulating these cars does not change the properties of the jam. Moreover, as soon as car i_{right} becomes deterministic, it can never re-enter the non-deterministic regime. Therefore, car number $i_{right} - 1$ becomes the new rightmost car, and car number i_{right} is no longer considered for the simulation.

To the left of car i_{left}, the situation is similar. The only information that is needed is the sequence of the gaps $(gap_i)_i$ of the incoming cars. Just before car $i_{left} - 1$ enters the jam, one more car is added at the left, with $gap_{i_{left}-2}$.

It is obvious that, with this computational technique, the only restriction for the spatial size is given by the memory of the computer. Since our model is one-dimensional, this has never been a problem.

A remaining question is how to obtain the sequence of gaps $(gap_i)_i$ of the incoming cars, especially for the outflow situation. One possibility would be to first run another simulation of the outflow from a mega-jam. Cars leave this mega-jam, drive through a regime of decreasing density, and eventually relax to the deterministic state. One then records the gaps between these cars, writes them to a file, and reads this file during the other simulation. Apart from technicalities (avoiding the intermediate file), this is the technique we adopted in our simulations.

Our program runs with approximately 270 000 vehicle updates per second on a SUN Sparc10 workstation; and since the critical density is $\rho_c \approx 0.0655$, for $v_{max} = 5$ this corresponds to $270\,000/0.0655 = 4.1 \cdot 10^6$ site updates per second. This includes all time for measurements and for the production of the gaps of the incoming cars.

The numerical results cannot resolve the question between logarithmic corrections for the width $w(t)$ or an exponent different from $1/2$, in spite of data obtained over six orders of magnitude in time. The reason for this is a large "bump" in the measurements of the width. Simulations of larger systems would have been helpful. The time complexity for our questions is $O(t)$: As shown above, when averaging over all *started* clusters, the number of active sites, $\langle\, n \,\rangle_{started}$, is constant in time: $\langle\, n \,\rangle_{started}(t) \sim t^0$. When t_{co} is the numerically imposed cut-off, then one performs for each started cluster in the average $\alpha\, t_{co}$ updates of a vehicle. From experience, $\alpha \approx 5$ for $v_{max} = 5$.

In other words, in order to add another order of magnitude in time, with the same statistical quality as before, one would need a factor of 10 more computational power. However, each of our graphs already stems from runs using 4 or more Sparc10 workstations for 10 days or more. And using a parallel supercomputer seems difficult: Standard geometric parallelization is ineffective because most of the time the jam-clusters are quite small, and in consequence all the CPUs responsible for cars further away "from the middle of the jam" would be idle. More sophisticated load-balancing methods might be a solution.

Appendix B

Labeling of jam clusters on a massively parallel computer

This appendix contains the technical details of the cluster labeling techniques which were used for the lifetime measurements in Section 6.1.

At first, "free" cars and "slow" cars are defined: After the deterministic part of the update and before the randomization step, all "free" cars have velocity $v = v_{max}$. One therefore defines all cars with $v < v_{max}$ at this point as "slow". One then looks for "clusters" of slow cars in the model and measured the lifetime of these clusters. Different jams are marked by different labels, and the jam of each label lbl is active between $t_{start}(lbl)$ and $t_{end}(lbl)$. Initially, $t_{end} = 0$ and $t_{start} = t_{max}$ are set for all lbl. (t_{max} is the total number of iterations of the simulation run.) Then, at each time step after the deterministic and before the random part of the velocity update, the following is done:

- All "fast" cars get a very high label number lbl_{max}, with $t_{start}(lbl_{max}) = 0$ and $t_{end}(lbl_{max}) = t_{max}$.
- Being "slow" (in the sense of the above definition) can in the model only be caused by two reasons: Either the car n had to slow down because the next one ahead $n + 1$ was too close, or the car has not yet accelerated to full speed due to a jam which it just has left. Therefore,

  $$t_{start}(n,t) = \min[t_{start}(n+1,t-1), t_{start}(n,t-1), t] \; .$$

  In words, this means that if two different jams may be the origin of n's slowness, then the algorithm selects the older one.
- Then the label is set to the one of the selected jam:

  ```
  IF $t_{start}(n,t) = t_{start}(n+1,t-1)$ THEN
      $lbl(n,t) = lbl(n+1,t-1)$
  ELSE IF $t_{start}(n,t) = t_{start}(n,t-1)$ THEN
      $lbl(n,t) = lbl(n,t-1)$
  ELSE
      $lbl(n,t) = newlbl$
  ENDIF
  ```

  where $newlbl$ is a new label not yet used.
- Next, t_{end} is updated: $t_{end}(lbl(n,t)) = t$.

The overall result of this labeling is that every vehicle which becomes “slow” without another “slow” car as a cause originates a new jam with an associated lifetime. When one jam splits up into several branches, they all obtain the same label because they have the same origin. In consequence, only the branch which stays “alive” the longest time determines the lifetime of this specific jam. When two branches *completely* merge together, the “older” one takes over. The younger one then no longer exists, but it is counted for the statistics because it had its own independent origin.

This algorithm is implemented on a Parsytec GCel-3 parallel computer, where up to 1024 processors could be used. The dynamics itself is implemented in a “vehicle-oriented” way as already explained earlier. Since, in single-lane traffic, the list of particles remains ordered, one can distribute the model by placing N/p consecutive vehicles on each of the p processors. This results, for large system sizes, in a computational speed of $8.5 \cdot 10^6$ *particle*-updates per second on 512 processors, compared to $0.34 \cdot 10^6$ on a Sparc10 workstation. (At a density of $\rho = 0.08$, this corresponds to $106 \cdot 10^6$ and $4.25 \cdot 10^6$ *site*-updates per second, respectively.) But for a smaller system size of $L = 10^5$ ($\rho = 0.08$) the computational speed on 512 nodes decreases to $3.1 \cdot 10^6$ particle-updates ($= 39 \cdot 10^6$ site-updates) per second.

For the parallel cluster labeling, the implementation follows the idea of Ref. [43]. That means that labels are assigned locally on the processors, and only labels that touch boundaries are exchanged with the neighbors. After the labeling, information on “active” labels is exchanged by a relaxation method (see Ref. [43]) to the leftmost processor which has this label in use. By this method one keeps track of “active” jams, and lifetimes of “dead” jams (i.e. cluster labels which were no longer in use) could be recorded.

For sufficiently large system sizes, the computational speed decreases by a factor of four due to the labeling; but for smaller systems of size $L = 10^5$ and $\rho \approx 0.08$, 512 processors are inefficient. The following table shows computational speeds (in MUPS = Mega Updates Per Second = 10^6 site-updates per second) for these parameters ($\rho = 0.08$):

number of computational nodes	32	128	512
speed w/o labeling	6.8	27	106
speed with labeling	1.5	5.5	5.7

In consequence, usually 128 processors per job were used. About five days of computing time on 512 processors (4 × 128) were needed for Fig. 6.1.

Appendix C

Enumeration of the 10 shortest network-paths

C. Moll has developed a short recursive procedure to enumerate the 10 shortest paths between any set of nodes in the freeway network of NRW. For completeness, the main procedure of this algorithm is given in the following. Note that the algorithm was developed for a specific problem (the network data was given in a certain form; the number of nodes and edges was known; the algorithm had only to run once to produce the simulation input; etc.), so that neither high computational speed nor general usefulness were an issue.

The algorithmical version is as follows; a more textual description is given below. The algorithm works on a directed graph $G(E, V)$, where E is the set of the edges and V is the set of the nodes or vertices.

```
global $Length(1 : |E|)$ /* contains the lengths of all edges */
global $Dist(v, w)$ /* distance from $v$ to $w$ via shortest path */
global $bound$ /* maximum length of the 10 shortest paths found so far */
global $CurrentPath(1 : |V|)$ /* list which contains all edges of the current path so far*/
global $Touched(1 : |V|)$ /* flag to mark nodes which have been touched */

subroutine FindAllPaths ( $v, CurrentLength, CurrentDepth$ )
    $Touched(v) :=$ true
    if ( $v = t$ ) SavePath ( $CurrentLength$ )
    for all edges $e$ adjacent to $v$ do
        $w := target(e)$
        if ( not $Touched(w)$ & $CurrentLength + Length(e) + Dist(w, t) < bound$ ) then
            $CurrentPath(CurrentDepth) := e$
            FindAllPaths ( $w, CurrentLength + Length(e), CurrentDepth + 1$ )
            $CurrentPath(CurrentDepth) := NULL$
        endif
    endfor
    $Touched(v) :=$ false
return
```

The algorithm is started for each origin-destination pair (s, t) at s. Then, FindAllPaths is called with the arguments s, $CurrentLength = 0$, and $CurrentDepth = 0$.

Now assume we are in the middle of the recursive enumeration. v is the current node, $CurrentPath$ is the list of edges we have used to reach v, $CurrentDepth$ is the number of edges in this list (minus one), *bound* is the maximum of the 10 shortest paths from s to t found so far. FindAllPaths now marks v as touched; because of the recursive algorithm, all nodes which are in $CurrentPath$ are marked as touched. For all outgoing edges from v it is checked if the target nodes are untouched, and if the path from v via this node and then using the shortest distance to t is shorter than *bound*. If both is true, the outgoing edge is added to $CurrentPath$, and FindAllPaths is called again, now starting at the new node. If this new node happens to be t, $CurrentPath$ is saved in the list of the so far known 10 shortest paths, and *bound* is adjusted.

$Dist(w, t)$ is pre-calculated, using a Dijkstra algorithm starting at t.

Thus, the algorithm recursively tries all paths from s to t, discarding those which have no chance to belong to the 10 shortest. It is good to have an initial guess for *bound*; in consequence, the algorithm starts with a very low initial *bound* and repeatedly increases it until 10 paths are found.

Appendix D

A continuous version

In order to investigate if there is a deterministic model which shows more complex (and therefore more realistic) behavior than the case of Chapter 4, a continuous version of the model is considered here. The one-dimensional system has, as usual in the traffic flow chapters, length L with periodic boundary conditions; but velocity v_i and position x_i of a vehicle i are now continuous variables. The update rule is as follows:

- If the velocity is high with respect to the gap, then the car slows down:

  $$v > gap + 1 - \alpha \quad \rightsquigarrow \quad v \rightarrow \max(\,0\,,\,gap\,)\ ;$$

  (the "max" is only necessary to prevent negative velocities);

- else if the velocity is low with respect to the gap *and* slower than five, then the car accelerates:

  $$v < gap + 1 - \beta \ \ \& \ \ v < v_{max} \quad \rightsquigarrow \quad v \rightarrow v + \min(\,1\,,\,\gamma * (gap + 1)\,)\ .$$

  Note that this rule allows maximum speeds up to nearly six.

- After the velocity has been updated for all vehicles according to the last two rules, we move all vehicles simultaneously according to their velocities.

In our simulation we used $\alpha = 0.5$, $\beta = 3.0$, and $\gamma = 0.1$.

The only new feature of this model with respect to the integer version is that the acceleration is weaker when the distance is still small. A distinct feature of this model which it shares with the integer model is that there is a "dead zone" in the distance to the next car ahead where a driver neither accelerates nor slows down. This is in accordance with psychological investigations insofar as it stresses the importance of physiological *thresholds* in order to make human drivers react (see Chapter 2 for further information).

With this model, simulations of different setups are performed. Whereas the normal closed systems shows a behavior similar to the integer model (i.e., settling down to an "imitate your leader"-state), already the introduction of one slightly slower vehicle leads to complex and unpredictable behavior. To be specific, the setup is as follows. In a system of length L and with periodic boundary conditions, initially, $N = [\rho \cdot L]$ vehicles are placed on sites 1 to N, all with velocity zero, where ρ is chosen small enough to prevent any effect of the last car of the platoon on the first one through the periodic boundary conditions. (A platoon is an ensemble of vehicles travelling together.) Starting from this totally ordered initial state, the system is allowed to evolve according to the above rules, with the exception of the first vehicle, the speed of which was kept fixed at $v_{lead} = 4.99999$ after its initial acceleration. This is a simplification of the well known situation where a number of fast vehicles has to follow a slower one which they cannot pass, and besides its obvious single-lane applications this situation also occurs on freeways when many passenger cars want to pass a group of trucks.

Fig. D.1 shows a section of the evolution of the system. For a certain time, the density behind the first car gets larger because all vehicles close up. But at seemingly random instances, many of the followers have to slow down in one large collective event, which redistributes the vehicles with a lower density. As the first vehicle moves freely, the simulation represents, in spite of the periodic boundary conditions, the situation of a platoon moving infinitely in space.

In Fig. D.2 we see the time evolution of a system which has been transformed to the coordinates of the first vehicle, i.e. the positions of all cars are given relative to the first car. One sees that equally spaced cars rapidly evolve into a fluctuating state (right hand side). In this new state density increases give rise to very short bursts (traffic jams) of very different sizes which redistribute the cars backwards such that in some cases they even start again in equally spaced patterns.

Although the model itself is totally deterministic, small perturbations may lead to totally different trajectories in phase space due to the chaotic dynamics. One notices this effect when simulating the same number of vehicles in systems of slightly different size: After a certain time, the development of the systems diverges, due to small differences in the transfer of cars through the periodic boundary. In order to clarify that this divergence is an intrinsic consequence of the dynamics and not just the enhancement of numerical round-off errors, single precision were compared with double precision calculations. The overall result of these tests is that noise enters the system with the same "speed" for all cases, which is a strong indication that complex behavior originates from a chaotic dynamics and is not driven by the limited precision of the floating point numbers.

In addition, the principal behavior of the model (i.e. the formation of the collective shocks) is robust with respect to parameter changes. More precisely, no qualitatively different behavior is found for changes of the parameters α, β, γ, and v_{lead} within the following range $0.1 \leq \alpha \leq 0.6$, $2.0 \leq \beta \leq 5.0$, $0.08 \leq \gamma \leq 0.12$, $4.5 \leq v_{lead} \leq 4.99999$.

In order to quantify these observations, the distribution of times τ between consecutive "braking" events for the last vehicle (τ is the time from the end of one braking to the beginning of the next) was measured. Braking is defined here as a slowing down according to the rules for the velocity update. We performed simulations on a Parsytec GCel-3, replicating a system with a fixed number of vehicles but different system sizes on up to 512 processors and averaging the results. For instance for $N = 1900$ vehicles, after $3 \cdot 10^5$ time steps to let the transients die out, the distribution of τ measured during about $1.1 \cdot 10^6$ further time steps. This specific simulation took about 33 hours on 256 processors.

According to Fig. D.3, the distribution displays a remarkable straight line on a log-log-plot, fulfilling the power law

$$n(\tau) \propto \tau^{-\alpha}$$

with $\alpha = -2.2 \pm 0.1$. This non-trivial exponent is a strong indication for the existence of self-organized criticality[8] for this model.

Many aspects of this model are reminiscent of the so-called train model for earthquake dynamics [158]. Instead of pulling at one end, the slower car may be seen as *pushing* against the other cars which want to move faster. This leads to a slowly increasing average density, and at some time this density locally exceeds a critical threshold. The reaction is a more or less drastic slowing down of the corresponding

Figure D.1: Evolution of the continuous model (5600 time steps), $N = 61$, $L = 1024$. As before, consecutive lines represent configurations at consecutive time steps; vehicles are denoted by black squares, empty spaces are left blank.

Figure D.2: (Previous page) First 4200 time-steps of the continuous model, transformed to the coordinates of the leading (rightmost) vehicle. Only every third time-step is plotted, everything else is as in Fig. D.1. The jam-waves are now clearly visible as nearly horizontal lines. The flow is from left to right.

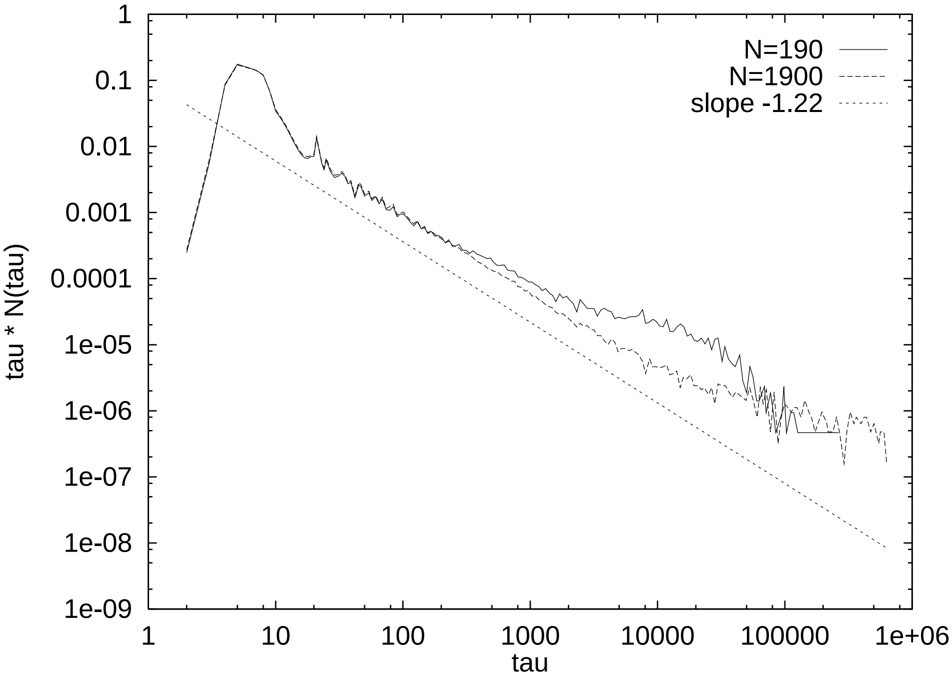


Figure D.3: Distribution $n(\tau)$ of times τ between consecutive braking events for $N = 190$ and $N = 1900$ cars. (The times are collected in "logarithmic bins", so that the y-axis is proportional to $\tau\, n(\tau)$.)

vehicle, which may or may not force the next vehicle to slow down as well. By this mechanism, avalanches of all sizes are generated, which may propagate through the entire platoon of vehicles.

Appendix E

Coding

This chapter discusses single-bit coding and compares it to the “intermediate algorithm”.

E.1 Single-bit coding: SBIT-algorithm

One iteration consists for each car of a deterministic velocity update, which depends on the neighbors, a randomized velocity update, and the vehicle movement. Since each site can only have a very low number of states, this is very much like Ising-model programming [59]. Accordingly, single bit coding is possible, which means that all 32 bits of a computer word carry information for a different site, which can be treated simultaneously with one integer-operation. For the NEC–SX3 and the CM–5 more efficient 64-bit integers are used, even when not explicitly mentioned in the following.

The technique of efficient single-bit coding is described in detail elsewhere [160], but it will be explained for our algorithm. The first step is to code the different vehicle velocities by Boolean variables (e.g. 0 and 1). Certain choices are possible for doing this, but one can follow [28, 151] in taking the simplest approach: Each velocity is denoted by a velocity “level” which means that one has arrays `iv0`, …, `iv5`; and a `1` in a level means that the vehicle at this place has (at least) this velocity. As an example, accelerating all vehicles is done by the following steps:

```
DO ii=1,max_x
  iv5(ii) = iv5(ii) .OR. iv4(ii)
  iv4(ii) = iv3(ii)
  iv3(ii) = iv2(ii)
  iv2(ii) = iv1(ii)
  iv1(ii) = iv0(ii)
  iv0(ii) = 0
ENDDO                                  .
```

These operations can as well be performed bitwise on all 32 bits of each computer word `iv0, ... , iv5` (single-bit coding). In theory, this technique can therefore yield a gain of a factor of 32 in computational speed as well as in memory consumption. In practice, one can program our traffic dynamics more efficiently when not intending to use single-bit coding, see below. Thus, the practical gain is much lower. But the single-bit coded program (“SBIT”) is fully vectorizable, which makes it far superior on the traditional vector computer NEC–SX3/11 and is still better on the vector CPNs of the CM–5 (CPN: **Com**~~P~~**P**utational **N**ode). This is a clear advantage over the work of Cremer, Ludwig and Schütt, whose bit-oriented coding scheme is not vectorizable. (See [151], where tests on a Convex C1-XP vector processor gave a speedup of only 2 against an IBM-80286-AT with 10 MHz). In order to get good vector performance, Fortran is used for this approach (CM-Fortran on the CM–5).

E.2 Touching only the occupied sites: DIST-algorithm

The CA updating approach as described above is not very efficient for traffic flow, because it needs for an empty site as many operations as for an occupied one. An approach which is better in this respect is, on an array where vehicles move from left to right, to go through the array from left to right, and to perform the following steps:

If you are at an occupied site, then

- *search for the next car ahead (*`distance`*);*
- *adapt the velocity according to that distance taking into account your own velocity;*
- *move car according to velocity;*
- *proceed to the next car ahead according to* `distance`

("algorithm DIST"). As this algorithm does not vectorize, C is used as programming language.

E.3 Parallelization and vectorization

Geometric parallelization of an array of length L is straightforward: Each of the N CPNs gets L/N sites of the array, and the boundary information is passed on by messages after each time step. On each architecture the fastest message passing method available was used, i.e. asynchronous communication where possible, and the boundary information is sent away at the earliest time possible, which is as soon as the edges of the array have been processed.

This method ensures that one will reach linear scale-up if one only makes the computational part of each CPN large enough: The part of communication at the total time goes to zero. Some machines allow overlapping of computation and communication, and then this works even better.

The situation is more complicated for the parallelization of the single-bit coded program. The standard technique for single-bit coding is to use "staggered" computer words, which means in an example with 32 bits (0 to 31) and a system size of $32 \times L$ that the leftmost L sites of a road are stored in bit number 31 of words 1 to L, sites $L+1$ to $2L$ are stored in the next bit (number 30) of the same words 1 to L, etc., until the whole system size of $32L$ is stored (cf. Fig. E.1). Now a bit-coded site in word number m finds the states of its two neighbors in words $m-1$ and $m+1$ at the same bit position ($m = 2 \ldots L-1$).

In order to provide the correct boundary information, one needs in our case at each end five ($= v_{max}$) additional boundary words with numbers -4 to 0 and $L+1$ to $L+5$. On a sequential machine and with periodic boundary conditions, the left neighbor of bit 31 in word 1 is bit 0 in word L, and the left neighbors of bits 30 to 0 in word 1 are bits 31 to 1 in word L. In general, words -4 to 0 contain a circular right shift of words $L-5$ to L, and words $L+1$ to $L+5$ contain a circular *left* shift of words 1 to 5.

w1b31	w2b31	w3b31	w4b31	w5b31	w6b31	w1b30	w2b30	w3b30	
$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	...

Figure E.1: Visualisation of single-bit coding. For example, "w4b31" means "bit 31 of word number 4". What can be seen here is the left end of the "road"-array on one CPN. In this example we use only 6 different words, so the length of this "road" is $6 \times 32 = 192$ sites. As one has to pass the states of the leftmost 5 bits to the next CPN, bit number 31 has to be extracted from words number 1 to 5 each. This information has to be sent to the next CPN to the left, where it has to be inserted into the correct words (the "boundary words"). — Note that words are counted from left to right whereas, according to the standard notation, bits are numbered from right to left.

Since boundary conditions need the information on five ($= v_{max}$) sites to be passed on to the neighbor CPN, this implies that this information has to be extracted bitwise from five words, to be passed on, and then to be inserted bitwise into five other words. When, say, sending information to the left neighbor CPN, then

- the states of bit number 31 in each of the words 1 to 5 have to be extracted,
- these states have to be sent to the left neighbor CPN,
- and on this CPN, bit number 0 in each of the words $L + 1$ to $L + 5$ has to be overwritten by these states.

And this has to be done for each variable, which are seven in our coding (`iv0, ..., iv5`, and one for the information if a site is occupied at all).

The situation is again different on the CM–5, where, at the time of this work, only the data parallel Fortran compiler was able to produce code for the four vector processors on each CPN. Translating the single-CPN code was straightforward, but tuning needed a lot of time. For example, the `WHERE ... ELSEWHERE ... ENDWHERE` construct turned out to be very slow and had to be replaced by masks.

Referring to vectorization, both codes were tested on a NEC-SX3/11 vector computer. As expected, the automatic vectorizer vectorizes all loops in the main part of SBIT completely, whereas the C compiler does not find any vectorizable loops at all in DIST. (The C-compiler of the NEC is supposed to find vectorizable loops, although experience shows that it does much worse than for Fortran loops. In addition, the DIST-algorithm is structured in a way that still would not vectorize efficiently even when written in Fortran.)

Appendix F

Practical computational complexity of geometric parallelization

In order to make reliable predictions on scaling behavior, some theory is helpful. Here results on a simplified model for the parallel time complexity of the algorithm are given. The predictions of the model will be compared with experimental results. Similar analysis for the geometric parallelization method have been presented by Heermann and Burkitt [60] and Jakobs and Gerling [73].

We begin by some basic assumptions for our one-dimensional system:

- The computing time t_{step} per time-step consists of calculation and of communication parts. It depends on the number of CPNs n and on the system size l per CPN. The overall system size L is always $L = l \cdot n$.
- Communication consists of a fixed amount $t_{sync}(n)$ of work for filling buffers, calling communication routines etc. and of the time required for transportation of the message, $t_{async}(n)$. A power law [73]

  $$t_{async}(n) = \begin{cases} 0 & : \quad n = 1 \\ c_1 \cdot n^{\epsilon} & : \quad n > 1 \end{cases}$$

  gives a good fit of the data for the one-dimensional traffic flow case. It should be noted that the power law is only used for the asynchronous part of the communication time, in contrast to [73].

  $t_{sync}(n)$ is given by

  $$t_{sync}(n) = \begin{cases} 0 & : \quad n = 1 \\ t^*_{sync} & : \quad n > 1 \end{cases}$$

- Computation times consist of a fixed amount t_{bnd} of work for dealing with boundaries and sending away the required messages and of the time t_{calc} for the update of l sites:

  $$t_{calc} = c_2 \cdot l$$

- Because of overlapping computation and communication, the message transportation time only becomes visible if it takes longer than computation:

  $$t_{step}(n) = t_{sync}(n) + t_{bnd} + max(t_{async}(n), t_{calc}(l))$$

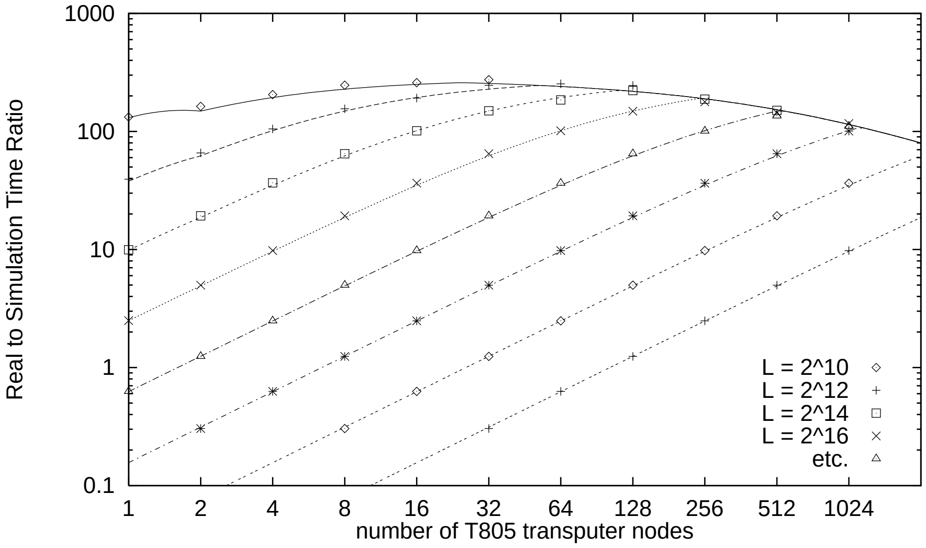


Figure F.1: Real to Simulation Time Ratio (measurements and approximating theoretical curves) as a function of the number of CPNs for different system sizes.

Thus

$$t_{step}(n) = \begin{cases} t_{bnd} + c_2 \cdot l & : \quad n = 1 \\ t^*_{sync} + t_{bnd} + max(c_1 \cdot n^{\epsilon}, c_2 \cdot l) & : \quad n > 1 \end{cases}$$

To verify the assumptions, computation times on a Parsytec GCel-3 with 1024 CPNs are measured. Systems of size L=$n \cdot 2^i$ on partitions of $n = 2^j$ CPNs ($i = 5 \ldots 19, j = 0 \ldots 10$) are used. From these data the following diagrams are calculated:

- Real to Simulation Time Ratio

$$realsim = \frac{1}{t_{step}}$$

- Speedup

$$speedup = \frac{t_{step}(1)}{t_{step}(n)}$$

- Efficiency

$$eff = speedup/n$$

given in Figs. F.1, F.2, and F.3. The approximating curves refer to parameter values $t^*_{sync} = 2.2 \cdot 10^{-3} sec$, $t_{bnd} = 1.4 \cdot 10^{-3} sec$, $c_1 = 2.0 \cdot 10^{-5} sec$, $c_2 = 6.1 \cdot 10^{-6} sec/site$, and $\epsilon = 0.8$. t_{bnd} and c_2 are found by approximating the values for one CPN ($n = 1$), t^*_{sync} by approximating all curves for small n, and $c1$ and ϵ by approximating all curves for large n. The approximations are visibly worse for $\epsilon = 0.5$ and for $\epsilon = 1.0$ (not shown).

The above definition for ϵ is not directly comparable to [73], which is due to explicitly taking care of overlapping computation and communication by only using the maximum of both times. But especially in Fig. F.1 one observes that this is necessary in order to approximate the abrupt change of the curves near their maxima. This is probably due to the fact that in this one-dimensional system the whole

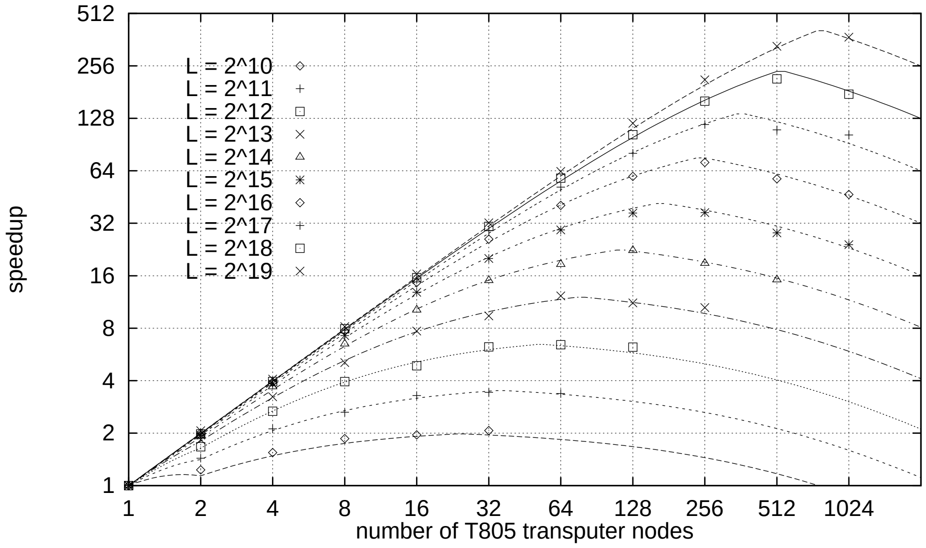


Figure F.2: Speedup (measurements and approximating theoretical curves) as a function of the number of CPNs for different system sizes.

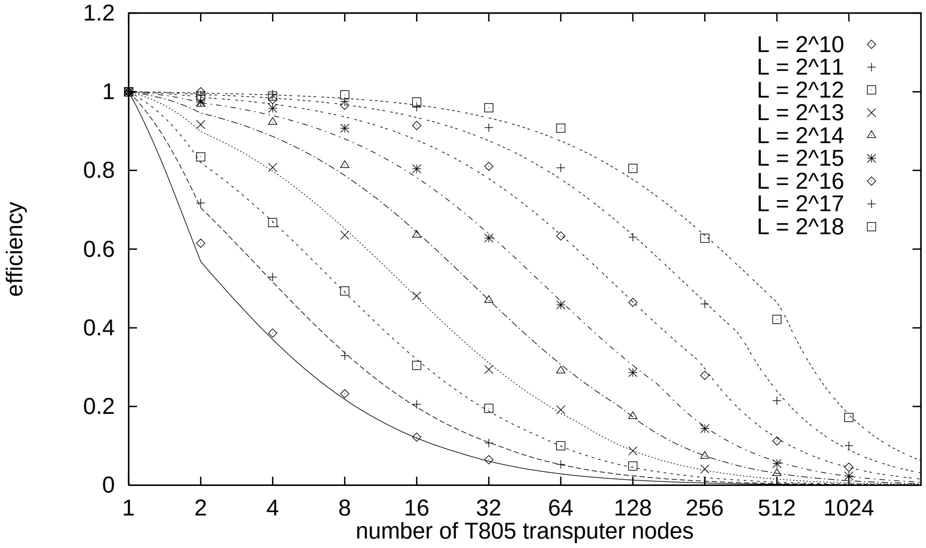


Figure F.3: Efficiency (measurements and approximating theoretical curves) as a function of the number of CPNs for different system sizes.

communication time does not depend on the system size, which makes the explicit formulation of the overlap more important.

With these parameters, the computational speed per CPN s may be written as

$$\begin{aligned} s &= \frac{l}{t_{step}} \\ &= \frac{l}{t^*_{sync} + t_{bnd} + max(t_{async}, t_{calc})} \end{aligned}$$

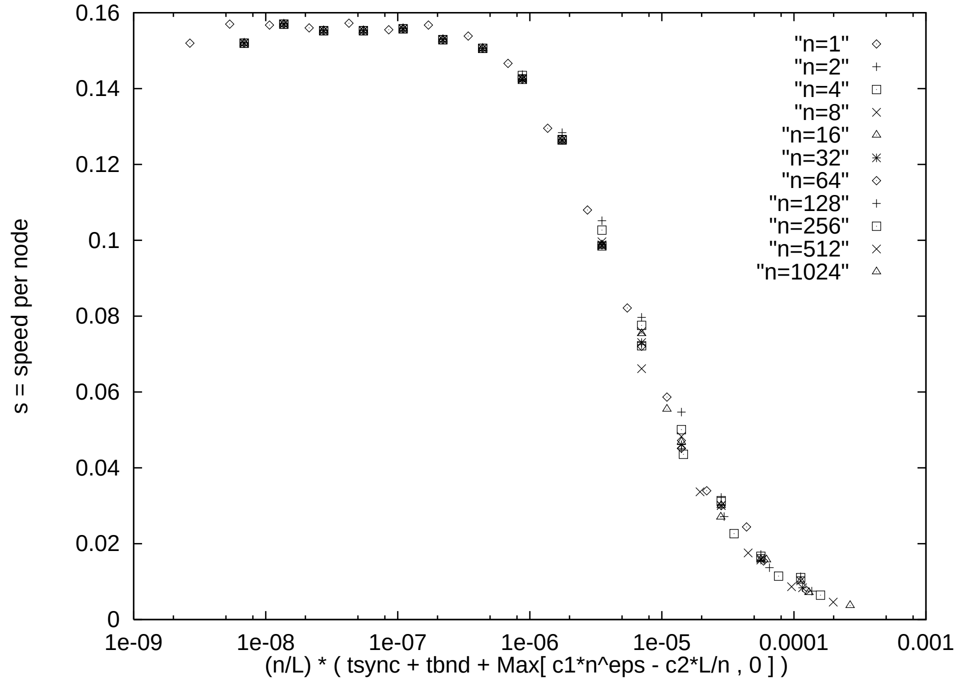


Figure F.4: Scaling Plot. This is the rescaled version of Fig. F.3.

$$
\begin{aligned}
&= \frac{1}{c_2 + (n/L)(t_{sync} + t_{bnd} + max(t_{async} - c_2 \cdot L/n, 0))} \\
&=: \frac{1}{c_2 + n_{scaled}} \; .
\end{aligned}
$$

In order to verify our approximation, the speed per CPN s is plotted against n_{scaled} (Fig. F.4) and one obtains a nice data collapse.

Bibliography

[1] Acha-Daza J A, Hall F L, Graphical comparison of predictions for speed given by catastrophe theory and some classic models, *Transportation Research Record* **1398** (1993) 119.

[2] Agyemang-Duah K, Hall F L, Some issues regarding the numerical value of freeway capacity, in: Brannolte (ed.), *Highway Capacity and Level of Service*, Balkema, Rotterdam, 1991.

[3] Arnott R, De Palma A, Lindsay R, A structural model of peak-period congestion: A traffic bottleneck with elastic demand, *The American Economic Review* **83(1)** (1993) 161.

[4] Arthur B, Inductive reasoning, bounded rationality, and the bar problem, *American Economic Review* (Papers and Proceedings) **84** (1994) .

[5] Axelrod R, *The Evolution of Cooperation*, Basic Books, NY, 1984.

[6] Axhausen K, personal communication.

[7] Bachem A, Korte B, Schrader R, Mathematische Modelle für Bausparkollektive, in: Rudolph B, Wilhelm J (eds.), *Festschrift für H.-J. Krümmel*, Dunker & Humblot, Berlin, 1988.

[8] Bak P, Tang C, Wiesenfeld K, Self-organized criticality, *Phys. Rev. A* **38** (1988) 368.

[9] Baldus A, *Simulation von Zellular-Automaten-Modellen für Verkehrsfluß mit offenen Randbedingungen*, Master Thesis, University of Cologne, 1993.

[10] Bando M, Hasebe K, Nakayama A, Shibata A, Sugiyama Y, Structure stability of congestion in traffic dynamics, *Japan Journal of Industrial and Applied Mathematics* **11(2)** (1994) 203.

[11] Barrett C, personal communication.

[12] Barrett C, Rasmussen S, Towards a theoretical foundation of simulation, in preparation.

[13] Baxter G W, Behringer R P, Fagert T, Johnson G A, Pattern formation in flowing sand, *Phys. Rev. Lett.* **62** (1989) 2825.

[14] Begley S, Why does traffic jam, *Newsweek*, August 1, 1994, 53.

[15] Ben-Naim E, Krapivsky P L, Redner S, Kinetics of clustering in traffic flows, cond-mat preprint 1994.

[16] Benz T, The microscopic traffic simulator AS (Autobahn Simulator), in: Pave A (ed.), *Proc. 1993 European Simulation Multiconference*, 1993, 486.

[17] Biham O, Middleton A, Levine D, Self-organization and a dynamical transition in traffic-flow models, *Phys. Rev. A* **46** (1992) R6124.

[18] Binder P-M, Paczuski M, Barma M, Scaling of fluctuations in one-dimensional interface and hopping models, *Phys. Rev. E* **49(2)** (1994) 1174.

[19] Blanchard T, Lake T, Conservative spatial simulation, in: Pave A (ed.), *Proc. 1993 European Simulation Multiconference*, 1993, 515, and other papers in the same proceedings.

[20] Burgers J M, *The nonlinear diffusion equation*, Riedel, Boston, 1974.

[21] Carlson J M, Chayes J T, Grannan E R, Swindle G H, Self-organized criticality and singular diffusion, *Phys. Rev. L* **65** (1990) 2547.

Carlson J M, Grannan E R, Singh C, Swindle G H, Fluctuations in self-organizing systems, preprint 1993.

[22] Catoni S, Pallottino S, Traffic Equilibrium Paradoxes, *Transpn. Sc.* **25** (1991) 240.

[23] Chang G-L, Kanaan A, Variability assessment for TRAF-NETSIM, *J. Transportation Engineering* **116(5)** (1990) 636.

[24] Cohen J, Kelly F, A paradox of congestion in a queueing network, *J. Appl. Probability* **27** (1990) 730.

[25] Cotton W R, Anthes R A, *Storm and cloud dynamics*, Academic Press, San Diego, CA, 1989.

[26] Cremer M, *Der Verkehrsfluß auf Schnellstraßen: Modelle, Überwachung, Regelung*, Fachberichte Messen-Steuern-Regeln, Springer, Berlin, 1979.

[27] Cremer M, O-D estimation: Dynamic methods, in: Papageorgiou M (ed.), *Concise encyclopedia of traffic and transportation systems*, Pergamon Press, Oxford, 1990.

[28] Cremer M, Ludwig J, A fast simulation model for traffic flow on the basis of Boolean operations, *Mathematics and Computers in Simulation* **28** (1986) 297.

[29] Cremer M, Meißner F, Traffic prediction and optimization using an efficient macroscopic simulation tool, in: Pave A (ed.), *Proc. 1993 European Simulation Multiconference*, 1993, 515.

[30] Cuesta J A, Martínez F C, Molera J M, Sánchez A, Phase transitions in two-dimensional traffic flow models, *Phys. Rev. E* **48(N6)** (1993) R4175-R4178.

[31] Daganzo C F, The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, California PATH, UCB-ITS-PRR-93-7 (1993).

[32] Daganzo C F, A finite difference approximation of the kinematic wave model, California PATH, UCB-ITS-RR-93-11 (1993).

[33] Daganzo C F, Lin W-H, The spatial evolution of queues during the morning commute in a single corridor, California PATH UCB-ITS-PWP-93-7 (1993).

[34] Derrida B, Domany E, Mukamel D, An exact solution of a one dimensional asymmetric exclusion model with open boundaries, *J. Stat. Phys.* **69** (1992) 667.

Derrida B, Evans M R, Exact correlation functions in an asymmetric exclusion model with open boundaries, *J. Phys. I France* **3** (1993) 311.

[35] Drossel B, Clar S, Schwabl F, Exact results for the one-dimensional self-organized critical forest-fire model, *Phys. Rev. Lett.* **71(N23)** (1993) 3739.

[36] Dutta P, Horn P M, Low-Frequency fluctuations in solids: $1/f$ noise, *Rev. Mod. Phys.* **53** (1981) 497.

[37] ECMWF, Daily global analyses, Operational data assimilation system, European Center for Medium Range Weather Forecast, every year.

[38] Emmerink R H M, Axhausen K W, Nijkamp P, Rietveld P, The potential of information provision in road transport networks with non-recurrent congestion, Paper TI 94-30 (1994), Tinbergen Institute, Roetersstraat 11, 1018 WB Amsterdam, The Netherlands.

Emmerink R H M, Axhausen K W, Nijkamp P, Rietveld P, Effects of information in road transport networks with recurrent congestion, TI 93-229 (1993), Tinbergen Institute.

[39] Eno Foundation for Transportation, Transportation in America, published by Transportation Policy Associates, in: National Transportation Statistics, D.O.T. Report DOT-VNTFC-RSPA-92-1 Annual Report, June 1992.

[40] Faieta B, Huberman B A, Firefly: A synchronization strategy for urban traffic control, preprint 1993.

[41] Feder J, *Fractals*, Plenum Press, 1988.

[42] Feller W, *An introduction to probability theory and its applications*, Vol. 1, Wiley, New York, 1968.

[43] Flanigan M, Tamayo P, A parallel cluster labeling method for Monte Carlo dynamics, *Int. J. Mod. Phys. C*, 1993.

[44] Frisch U, Hasslacher B, Pomeau Y, Lattice-gas automata for Navier-Stokes equations, *Phys. Rev. Lett.* **56** (1986) 1505.

[45] Fujimoto R M, Parallel discrete event simulation, *Commun. ACM* **33** (1990) 10.

[46] Gerlough D L, Simulation of freeway traffic by an electronic computer, *Proc. 35th Annual Meting*, Highway Research Board, Washington, D.C., 1956.

[47] Gerlough D L, Huber M J, *Traffic Flow Theory*, Special Report 165 (1975), Transportation Research Board, National Research Council, Washington, D.C.

[48] Goldhirsch I, Zanetti G, Clustering instability in dissipative gases, *Phys. Rev. Lett.* **70(11)** (1993) 1619.

[49] Grassberger P, de la Torre A, Reggeon field theory (Schlögl's First Model) on a lattice: Monte Carlo calculations of critical behaviour, *Ann. Phys. (NY)* **122** (1979) 373.

[50] Greenwood D G, Taylor N B, Parallelising the CONTRAM traffic assignment model, preprint 1992.

[51] Gynnerstedt G F, Introductory description of the Swedish two-way road traffic simulation model, preprint 1992.

Gynnerstedt G F, Two-way highway traffic simulation model system, in: Pave A (ed.), *Proc. 1993 European Simulation Multiconference*, 1993, 474.

[52] Gwa L-H, Spohn H, Bethe solution for the dynamical-scaling exponent of the noisy Burgers equation, *Phys. Rev. A* **46** (1992) 844.

[53] Haberman R, *Mathematical models in mechanical vibrations, population dynamics, and traffic flow*, Prentice-Hall, 1977.

[54] Häußermann P, Informatik und Straßenverkehr heute: Stand in den großen nationalen und europäischen Projekten, *Informationstechnik und Technische Informatik* **36** (1994) 67.

[55] Hall R W, Non-recurrent congestion: How big is the problem? Are traveler information systems the solution?, *Transpn. Res. C* **1(1)** (1993) 89.

[56] Hall F L, Allen B L, Gunter M A, Empirical analysis of freeway flow-density relationships, *Transpn. Res. A* **20A** (1986(3)) 197.

Hall F L, Pushkar A, Shi Y, Some observations on speed-flow and flow-occupancy relationships under congested conditions, *Transportation Research Record* **1398** (1993) 24.

[57] *Aktuell '93. Das Lexikon der Gegenwart*, Harenberg Lexikon-Verlag, Dortmund 1992.

[58] Hauer E, Hurdle V F, Discussion of the Freeway Traffic Model 'Freflo', *Transpn. Res. Rec.* **722** (1979) 75.

[59] Heermann D W, *Computer simulation methods in theoretical physics*, Springer, Heidelberg, 1986.

[60] D. W. Heermann and A. N. Burkitt, Parallelization of the Ising model and its performance evaluation, *Parallel Comput.* **13** (1990) 345.

[61] Helbing D, An improved fluid-dynamic model for vehicular traffic, *Phys. Rev. E*, submitted 1994.

[62] Herman R, Prigogine I, A two-fluid approach to town traffic, *Science* **204** (1979) 148.

[63] *Highway Capacity Manual*, Special Report 209 (1985), Transportation Res. Board, National Research Council, Washington, DC.

[64] Hilliges M, Reiner R, Weidlich W, A simulation model of dynamic traffic flow in networks, in: Pave A (ed.), *Proc. 1993 European Simulation Multiconference*, 1993, 505.

[65] Hislop A, McDonald M, Hounsell N, The application of parallel processing to traffic assignment for use with route guidance, *Traffic Engineering + Control* November 1991, 510.

[66] Hohenberg P C, Halperin B I, Theory of dynamic critical phenomena, *Rev. Mod. Phys.* **49(3)** (1977) 435.

[67] Holland J H, Using classifier systems to study adaptive nonlinear networks, in: Stein D (ed.), *Lectures in the sciences of complexity*, SFI studies in the sciences of complexity, Addison-Wesley Longman, 1989, 463.

[68] Hoshen J, Kopelman R, Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm, *Phys. Rev. B* **14** (1976) 3438.

[69] Hsu P, Banks J H, Effects of location on congested-regime flow-concentration relationships for freeways, *Transportation Research Record* **1398** (1993) 17.

[70] IVHS AMERICA, Surface transportation: Mobility, technology, and society, *Proc. IVHS AMERICA 1993 annual meeting*, Washington, D.C., 1993.

[71] Ito N, Non-equilibrium critical relaxation and interface energy of the Ising model, *Physica A* **196** (1993) 1.

[72] Jaeger H M, Nagel S R, Physics of the granular state, *Science* **255** (1992) 1523.

[73] Jakobs A, Gerling R W, Scaling properties of geometric parallelization, *Physica A* **180** (1992) 407.

Jakobs A, Gerling R W, Scaling results for parallelism on the Intel i860 hypercube, *Int. J. Mod. Phys. C* **4(N5)** (1993) 983.

Jakobs A, Gerling R W, Scaling aspects for the performance of parallel algorithms, *Parallel Computing* **19(N9)** (1993) 1063.

[74] Janowsky S A, Lebowitz J L, Finite-size effects and shock fluctuations in the asymmetric simple-exclusion process, *Phys. Rev. A* **45** (1992) 618.

[75] Jensen I, Dickman R, Nonequilibrium phase transitions in systems with infinitely many absorbing states, *Phys. Rev. E* **48** (1993) 1710.

[76] Jensen I, Critical exponents for branching annihilating random walks with an even number of offspring, cond-mat preprint 1994.

[77] Kanaan A, personal communication.

[78] Kardar M, Parisi G, Zhang Y, *Phys. Rev. Lett.* **56** (1986) 889.

[79] Kellershohn I, Mathematische Simulationen von Bausparkollektiven mit Hilfe von empirischen Verteilungen in monothetischen hierarchischen Kollektivclusterungen, Ph.D.-thesis, University of Cologne, 1992.

[80] Kerner B S, Konhäuser P, Cluster effect in initially homogenous traffic flow, *Phys. Rev. E* **48(4)** (1993) R2335-R2338.

Kerner B S, Konhäuser P, Structure and parameters of clusters in traffic flow, *Phys. Rev. E* **50(1)** (1994) 54.

[81] Kertész J, Wolf D E, Noise reduction in Eden models: II. Surface structure and intrinsic width, *J. Phys. A: Math. Gen.* **21** (1988) 747.

[82] Kim N-G, Turvey M T, Carello C, Optical information about the severity of upcoming events, *Journal of Experimental Psychology, Human Perception, and Performance* **19(1)** (1993) 179.

[83] Kinzel W, Directed percolation, in: Deutscher G, Zallen R, Adler J (ed.), *Percolation structures and processes*, A. Hilger, 1983, 425.

[84] König R, Langbein R, Simulation of traffic management strategies, *Proceedings of ISATA '93* (1993).

König R, Langbein R, PROROAD—Simulation of the consequences of traffic management strategies on the road traffic, in: Pave A (ed.), *Proc. 1993 European Simulation Multiconference*, 1993.

[85] Krug J, Spohn H, Universality classes for deterministic surface growth, *Phys. Rev. A* **83(8)** (1988) 4271.

[86] Krug J, Steady state selection in driven diffusive systems, in: Riste T, Sherrington D (eds.), *Spontaneous formation of space-time structures and criticality*, Kluwer Academic Publishers, Netherlands, 1991, 37.

[87] Kühne R, Verkehrsablauf auf Fernstraßen, *Physikalische Blätter* **47(3)** (1991) 201.

[88] Kühne R, Traffic patterns in unstable traffic flow on freeways, in: Brannolte U (ed.), *Highway Capacity and Level of Service*, Balkema, Rotterdam, 1991.

Kühne R, Freeway Speed Distribution and Acceleration Noise - Calculations from a stochastic continuum theory and comparison with measurements, in: *Proc. Int. Symp. on Transpn. and Traffic Theory*, MIT, Cambridge, 1987.

Kühne R, Macroscopic freeway model for dense traffic stop-start waves and incident detection, *9th Int. Symp. Transpn. Traffic Theory*, VNU Science Press, 1984.

[89] Kuwahara M, Newell G F, Queue evolution on freeways leading to a single core city during the morning peak, in: Gartner N H, Wilson N H M (eds.), *Transportation and traffic theory*, Elsevier, 1987.

[90] Landau L D, Lifschitz E M, *Hydrodynamik*, Volume 6 of: Landau L D, Lifschitz E M, Lehrbuch der theoretischen Physik, Akademie-Verlag, Berlin, 1966.

[91] Langton C G, Life at the edge of chaos, in: Langton C G et al (eds.), *Artificial Life II*, Santa Fe Institute Studies in the Science of Complexity, Vol. 10, Addison-Wesley, Redwood City, CA, 1992.

[92] Latour A, *Simulation von Zellularautomaten-Modellen für Mehrspurverkehr*, Master Thesis, University of Cologne, 1993.

[93] Lawler E L, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[94] Leath P L, Cluster size and boundary distribution near percolation threshold, *Phys. Rev. B* **14** (1976) 5046.

[95] Lee D N, A theory of visual control of braking based on information about time to collision, *Perception* **5** (1976) 437.

Lee D N, Young D S, Reddish P E, Lough S, Clayton T M H, Visual timing in hitting an accelerating ball, *Quarterly Journal of Experimental Psychology* **J35A** (1983) 333.

[96] Lee J, Density waves in the flows of granular media, *Phys. Rev. E* **49** (1994) 281.

[97] Leibig M, Pattern-formation characteristics of interacting kinematic waves, *Phys. Rev. E*, **94(1)** (1994) 184.

[98] Leutzbach W, *Introduction to the theory of traffic flow*, Springer, Berlin, 1988.

[99] Lifschitz E M, Pitajewski L P, *Physikalische Kinetik*, Volume 10 of: Landau L D, Lifschitz E M, Lehrbuch der theoretischen Physik, Akademie-Verlag, Berlin, 1983.

[100] Lighthill M J, Whitham G B, On kinematic waves: I. Flood movement in long rivers, *Proc. R. Soc. Lond.* **A229** (1955) 281.

Lighthill M J, Whitham G B, On kinematic waves: II. A theory of traffic flow on long crowded roads, *Proc. R. Soc. Lond.* **A229** (1955) 317.

Lighthill M J, Whitham G B, On kinematic waves: II. A theory of traffic flow on long crowded roads, Special Report **79** (1964) 8, Highway Research Board, National Research Council, Washington, D.C.

[101] MacKie-Mason J K, Varian H R, Pricing the Internet, University of Michigan preprint 1994.

MacKie-Mason J K, Varian H R, Some economics of the Internet, University of Michigan preprint 1992.

[102] Mahmassani H S, Peeta S, Network performance under system optimal and user equilibrium assignments: Implications for Advanced Traveler Information Systems, *Transportation Research Record* **1408** (1993) 83.

[103] Mahmassani H S, Jayakrishnan R, Herman R, Network traffic flow theory: Microscopic simulation experiments on supercomputers, *Transpn. Res. A* **24A(2)** (1990) 149.

[104] Mahmassani H S, Chang G-L, Herman R, Individual Decisions and Collective Effects in a Simulated Traffic System, *Transpn. Sc.* **20(4)** (1986) 258.

[105] Majumdar S N, Barma M, Tag diffusion in driven systems, growing interfaces, and anomalous fluctuations, *Phys. Rev. B* **44(10)** (1991) 5306.

[106] Mandelbrot B B, *The fractal geometry of nature*, Freeman, 1982.

[107] Maslov S, Paczuski M, Bak P, *Phys. Rev. Lett.*, to be published.

[108] May A D, *Traffic flow fundamentals*, Prentice Hall, Englewood Cliffs, NJ, 1990.

[109] McArthur D, Rule-based representation of driver behaviour in traffic modelling, Paper presented at *International Conference on Applications of AI*, June 1993.

[110] McArthur D, The PARAMICS model: Present and future directions, preprint 1994.

[111] Migowsky S, Wanschura T, Ruján P, Competition and cooperation on a toy Autobahn model, preprint 1994.

[112] Molera J M, Martínez F C, Cuesta J A, Theoretical approach to two-dimensional traffic flow models, preprint 1994.

[113] Moll C, personal communication.

[114] Montroll E W, Theory and observations of the dynamics and statistics of traffic on an open road, 1962, 231.

[115] Musha T and Higuchi H, The 1/f fluctuation of traffic current on an expressway, *Jap. J. Appl. Phys.* **15** (1976) 1271.

Musha T, Higuchi H, *Jap. Jrn. Appl. Phys.* **17(5)** (1978) 811.

[116] Nagatani T, Bunching of cars in asymmetric exclusion models for freeway traffic, submitted PRE.

[117] Nagatani T, Self-organization and phase transition in traffic-flow model of a two-lane roadway, *J. Phys. A* **26** (1993) L781-L787.

Nagatani T, Traffic jam and shock formation in stochastic traffic-flow model of a two-lane roadway, *J. Phys. Soc. Japan* **63(1)** (1994) 52.

Nagatani T, Dynamical jamming transition induced by a car accident in traffic-flow model of a two-lane roadway, *Physica A* **202** (1994) 449.

[118] Nagatani T, Jamming transition in the traffic-flow model with two-level crossings, *Phys. Rev. E* **48(5)** (1993) 3290.

[119] Nagatani T, Spreading of traffic jam in a traffic flow model, *J. Phys. Soc. Japan* **62** (1993) 1085.

Nagatani T, Power-law distribution and 1/f noise of waiting time near traffic-jam threshold, *J. Phys. Soc. Japan* **62** (1993) 2533.

Nagatani T, Anisotropic effect on jamming transition in traffic-flow model, *J. Phys. Soc. Japan* **62** (1993) 2656.

Nagatani T, Jamming transition induced by a stagnant street in a traffic-flow model, *Physica A* **198** (1993) 108.

Nagatani T, Clustering of cars in cellular-automaton model of freeway traffic, *J. Phys. Soc. Japan* **62(N11)** (1993) 3837.

Nagatani T, Traffic jam and phase transitions in two-dimensional traffic-flow model with three components, preprint 1994.

Nagatani T, Effect of jam-avoiding turn on jamming transition in two-dimensional traffic flow model, *J. Phys. Soc. Japan* **63(4)** (1994) 1228.

Nagatani T, Traffic jam induced by a crosscut road in a traffic-flow model, *Physica A* **207** (1994) 574.

Nagatani T, Shock formation and traffic jam induced by a crossing in the 1d asymmetric exclusion model, *J. Phys. A: Math. Gen.* **26** (1993) 6625.

[120] Nagel K, Schreckenberg M, A cellular automaton model for freeway traffic, *J. Phys. I France* **2** (1992) 2221.

[121] Nagel K, Herrmann H J, Deterministic models for traffic jams, *Physica A* **199** (1993) 254.

[122] Nagel K, Life-times of simulated traffic jams, *Int. J. Mod. Physics C* **5(3)** (1994) 567.

[123] Nagel K, Paczuski M, Emergent Traffic Jams, submitted.

[124] Traffic Network Analysis with NETSIM—A User Guide, Federal Highway Administration, Washington, D.C., 1980.

Rathi A K, Santiago A J, The new NETSIM simulation program, *Traffic Engineering + Control* May 1990, 317.

[125] Newell G F, Theories of instability in dense highway traffic, *J. Oper. Res. Soc. Jap.* **5** (1963) 9.

Newell G F, Instability in dense highway traffic, a review, *Proc. 2nd Int. Symp. Theory Road Traffic Flow*, OECD, 1965, 73.

[126] Stochastic and dynamic models in transportation, Special issue, *Operations Research* **41(1)** (1993).

[127] *PROMETHEUS, das europäische Forschungsprogramm zur Optimierung des Straßenverkehrssystem in Europa*, Bundesminister für Forschung—Pressereferat (ed.), 1989.

[128] Paczuski M, Bak P, Theory of the one-dimensional forest-fire model, *Phys. Rev. E* **48** (1993) R3214.

[129] Papageorgiou M, Banos J C M, Messmer A, Optimal control of multidestination traffic networks, in: *Proc. 29th conf. on decision and Control*, Honolulu, Hawaii, 1990.

[130] Payne H J, A critical review of a macroscopic freeway model, in: *Am. Soc. Cov. Eng.*, New York, 1979, 251.

[131] Peng G, Herrmann H J, Density waves of granular flow in a pipe using lattice-gas automata, *Phys. Rev. E* **48** (1994) R1796.

Peng G, Herrmann H J, Density waves and $1/f$ density fluctuations in granular flow, preprint HLRZ 11/94 (1994).

[132] Persaud B N, Hall F L, Catastrophe theory and patterns in 30-second freeway traffic data—Implications for incident detection, *Transpn. Res. A* **23A(2)** (1989) 103.

[133] Pfenning J-T, Experiences with the Mether-NFS virtual shared memory system, *Proc. High-Performance Computing and Networking conference*, Springer, 1994, 316.

[134] Pfenning J-T, personal communication.

[135] Piper H P, Stauvorgänge auf voll ausgelasteten Autobahnen, *Internationales Verkehrswesen* **43(11)** (1991) 489.

[136] Pöschel T, Recurrent clogging and density waves in granular material flowing through a narrow pipe, *J. Phys. I France* **4** (1992) 499.

[137] Press W H, Flannery B P, Teukolsky S A, Vetterling W T, *Numerical Receipes in C*, Cambridge University Press, 1988.

[138] Prigogine I, Herman R, *Kinetic theory of vehicular traffic*, Elsevier, New York, 1971.

[139] Rathi A K, Santiago A J, Identical traffic streams in the TRAF-NETSIM simulation program, *Traffic Engineering and Control* June 1990, 351.

[140] Reif F, *Fundamentals of statistical and thermal physics*, McGraw Hill, New York, 1965.

[141] Rickert M, *Simulationen zweispurigen Autobahnverkehrs mit Zellularautomaten*, Master Thesis, Univ. of Cologne, 1994.

[142] Ristow G, Herrmann H J, Density patterns in two-dimensional hoppers, *Phys. Rev. E* **50** (1994) R5.

[143] Rödiger M, *Chaotische Lösungen nichtlinearer Wellengleichungen mit Anwendungen in der Verkehrsflußtheorie*, Master Thesis, University of Münster, 1990.

[144] SWARM workshop, Santa Fe Institute, June 16-19, 1994.

[145] Sahimi M, Flow phenomena in rocks: From continuum models to fractals, percolation, cellular automata, and simulated annealing, *Rev. Mod. Phys.* **65(4)** (1993) 1393.

[146] Santiago A J, Chen H, Traffic modeling software for IVHS applications, in: Balci O, Sadowski R P, Nance R E (eds.), *Proceedings of the 1990 winter simulation conference*, 759.

[147] Schadschneider A, Schreckenberg M, Cellular automaton models and traffic flow, *J. Phys. A* **26** (1993) L679.

[148] Schäfer J, Wolf D E, *Phys. Rev. E*, in press.

[149] Schick K L, Verveen A A, *Nature* **251** (1974) 599.

[150] Schreckenberg M, Schadschneider A, Nagel K, Ito N, Discrete stochastic models for traffic flow, submitted.

[151] Schütt H, *Entwicklung und Erprobung eines sehr schnellen, bitorientierten Verkehrssimulationssystems für Straßennetze*, Schriftenreihe der AG Automatisierungstechnik TU Hamburg-Harburg 6 (1991).

[152] Schuster H G, *Deterministic Chaos*, Physik Verlag, Weinheim, 1984.

[153] Schwarzmann R, Das EUROTOPP Modell, SCOPE/VIKTORIA-Bericht, Institut für Verkehrswesen, TH Karlsruhe, Juli 1992.

[154] Schwerdtfeger T, *Makroskopisches Simulationsmodell für Schnellstraßennetze mit Berücksichtigung von Einzelfahrzeugen (DYNEMO)*, Ph.D. thesis, TH Karlsruhe, 1987.

[155] Sheffi Y, *Urban transportation networks: Equilibrium analysis with mathematical programming methods*, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[156] Sick B, *Dynamische Effekte bei nichtlinearen Wellengleichungen mit Anwendungen in der Verkehrsflußtheorie*, Master Thesis, University of Ulm, 1989.

[157] Simão H P, Powell W B, Numerical methods for simulating transient, stochastic queueing networks, *Transportation Science* **26** (1992) 296.

[158] Sousa Vieira M de, Self-organized criticality in a deterministic mechanical model, *Phys. Rew. A* **46** (1992) 6288.

[159] Stanley E H, Stauffer D, Kertész J, Herrmann H J, Dynamics of spreading phenomena in two-dimensional Ising models, *Phys. Rev. Lett.* **95(20)** (1987) 2326.

[160] Stauffer D, Computer simulations of cellular automata, *J. Phys. A* **24** (1991) 909.

[161] Stauffer D, personal communication.

[162] Stauffer D, Hehl F W, Winkelmann V, Zabolitzky J G, *Computer simulation and computer algebra*, Second edition, Springer, 1989.

[163] Stauffer D, Aharony A, *Introduction to percolation theory*, Taylor & Francis, London, 1992.

[164] Stephanedes Y J, Argiropoulos I Z, Michalopoulos P, On-line traffic assignment for optimal freeway corridor control, *J. Transportation Engineering* **116(6)** (1990) 744.

[165] TRANSIMS—The **TR**ansportation **AN**alysis and **SIM**ulation **S**ystem project, Los Alamos National Laboratory.

[166] Tadaki S, Kikuchi M, Jam phases in two-dimensional cellular automata model of traffic flow, patt-sol preprint 1994.

[167] Takayasu M, Takayasu H, Phase transition and $1/f$ type noise in one dimensional asymmetric particle dynamics, *Fractals* **1(4)** (1993) 860-866.

[168] Treiterer J, Investigation of Traffic Dynamics by Aerial Photogrammetry Techniques, *Techn. Rep. PB246 094*, Columbus, Ohoi, 1975.

[169] Treiterer I, Myers J A, The hysteresis phenomenon in traffic flow, in: Buckley D J (ed.), *6th Int. Symp. Transpn. Traffic Theory*, A.H. & A.W. Reed Pty Ltd., Artarmon, New South Wales, 1974.

[170] Van Aerde M, Yagar S, Ugge A, Case E R, A review of candidate freeway-arterial corridor traffic models, *Transpn. Res. Rec.* **1132** (1990) 53.

[171] Verkehr in Zahlen, Der Bundesminister für Verkehr (ed.), 1993.

[172] Vilar L C Q, de Souza A M C, Cellular automata models for general traffic conditions on a line, *Physica A* **211(1)** (1994) 84.

[173] Vicsek T, *Fractal growth phenomena*, World Scientific, 1989.

[174] Wallace C E, At last—A TRANSYT model designed for American traffic engineers, *ITE Journal*, August 1983, 28.

[175] Walker J, The amateur scientist: How to analyze the shock waves that sweep through expressway traffic, *Scientific American*, August 1989, 84.

[176] Whitham G B, Exact solutions for a discrete system arising in traffic flow, *Proc. Royal Society London* **A428** (1990) 49.

[177] Wiedemann R, *Simulation des Straßenverkehrsflusses*, Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Heft 8 (1974).

[178] Windmöller I, Clusteranalyse zur Steuerung von Bausparkollektiven, talk at *Operations Research 1994*, International conference on Operations Research, TU Berlin, 1994.

[179] Wolf D E, Kertesz J, Noise reduction in Eden models: I, *J. Phys. A: Math. Gen.* **20** (1987) L257.

[180] Wolfram S, *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986.

[181] Wylie B J N, McArthur D, Brown M D, *PARAMICS parallelisation schemes*, EPPC-PARAMICS-CT.10.

[182] Yukawa S, Kikuchi M, Tadaki S, Dynamical phase transition in one dimensional traffic flow model with blockage, cond-mat preprint 1994.

[183] Zackor H, Kühne R, Balz W, *Untersuchungen des Verkehrsablaufs im Bereich der Leistungsfähigkeit und bei instabilem Fluß*, Forschung Straßenbau und Straßenverkehrstechnik 524 (1988), Bundesminister für Verkehr, Bonn–Bad Godesberg.

[184] Zallmann A, König R, A tool for defining realistic traffic scenarios for an integrated traffic management system, *Proceedings of ISATA 1993*, 1993.

Acknowledgements

Work such as this would never be possible as a single-person enterprise: Too diverse and interdisciplinary are the topics which have to be covered, ranging, in this case, from Traffic Engineering via Fluid-Dynamics and Simulation to Statistical Physics and Combinatorial Optimization.

First of all, I thank Professor A. Bachem, who introduced me to the problem. He was always open-minded towards new ideas, yet forced me in numerous discussions to clarify why and in how far they would be helpful for attacking the problem.

Next, I thank my collaborators. Chris Barrett, Hans Herrmann, Christoph Moll, Maya Paczuski, Steen Rasmussen, Marcus Rickert, Doug Roberts, Axel Schleicher, and Michael Schreckenberg accompanied different parts of the work, knew about the details which are always more complicated than they look, and searched with me for solutions. Most of them are co-authors of the papers which have been written during this thesis work, MS is in addition the co-referent of the thesis.

Chris Barrett and Steen Rasmussen were also my hosts during 6 months at the Los Alamos National Laboratory and at the Santa Fe Institute, where they initiated me to thinking about this work in a more general simulation context. Per Bak hosted me for another 6 months at Brookhaven National Laboratory, and I enjoyed (not only there) sharing many ideas with him, about this thesis and about other subjects.

Dietrich Stauffer gave pragmatic and helpful advice.

Thomas Pfenning explained to me how computers really work.

I had enjoyful and helpful discussions with Kay Axhausen, Dante Chialvo (about similarities between traffic and nerve cells), Kim Christensen, Sergei Esipov, Heinrich Jaeger, Janos Kertesz, Greg Kohring, Joachim Krug, Chris Langton, Jysoo Lee, Michael Leibig, Sergej Maslov, Darryl Morgueson, Syd Nagel, Gongweng Peng, Thorsten Poeschel, Harald Puhl, Mike Stein, Nick Vriend, and Dietrich Wolf.

R. Kühne and P. Konhäuser gave necessary help at an early stage, and they pointed me to the important references [28] and [151].

I am obliged to my colleagues at the Zentrum für Paralleles Rechnen ZPR in Cologne, in the Simulation Applications group (TSA-DO) of the Los Alamos National Laboratory, at the Sante Fe Institute, and at the Physics department of the Brookhaven National Laboratory.

Special thanks to Corinna, Dagmar, Marc, Martijn, Stefan, and Una-May, for reminding me regularly that there is also (a) life outside the computer.

Dagmar Sternad also provided Refs. [95, 82].

Last, but not least, I am grateful to my parents, who, by always supporting me generously and critically, built in me the confidence which was necessary to accomplish this work.

And I am sure I still forgot many others!

This work was supported in part by the "Graduiertenkolleg Scientific Computing Köln/St. Augustin" and by the GRA program of the Los Alamos National Laboratory.

Computing time on the Parsytec GCel-3 and on the workstation cluster of the Zentrum für Paralleles Rechnen Köln, on the Intel iPSC/860 of the Forschungszentrum Jülich, on the CM–5 of the Institut für Angewandte Informatik Wuppertal, and on the NEC–SX3 of the Regionales Rechenzentrum Köln is gratefully acknowledged, as well as the corresponding support.

Deutsche Zusammenfassung

In dieser Arbeit wird die Eignung eines sehr schnellen, hardware-orientierten, *mikroskopischen* Modelles für Verkehrssimulationen untersucht.

Nach einem einführenden Kapitel über Verkehrsfluß-Theorie, Verkehrssimulationsmodelle sowie über kritische Phänomene und ihre Bedeutung für die Staudynamik wird eine der besten verfügbaren Methoden für schnelle mikroskopische Simulationen—die Methode der Zellularautomaten (CA) [160]—für Verkehrssimulationen variiert. Straßen werden in Kästchen gleicher Länge unterteilt, und ein Kästchen ist entweder leer, oder von genau einem Auto besetzt. Dieses Auto kann Integer-Geschwindigkeiten v zwischen 0 und v_{max} annehmen. Bewegung des Fahrzeuges erfolgt durch Springen auf dem Kästchenraster; vorher gewährleisten (stochastische) Brems- und Beschleunigungsregeln, daß keine "Unfälle" passieren.

Dies ist ein ziemlich einfaches Teilchen-Modell; um so erstaunlicher ist sein wirklichkeitsnahes Verhalten, so z.B. das spontane Entstehen realistischer Stop-and-Go Wellen [87] sowie realistische Fundamentaldiagramme [56], z.B. Durchfluß q gegen Dichte ρ oder Geschwindigkeit v gegen Dichte.

Dieser Ansatz ist einspurig; Rickert [141] und Latour [92] beschreiben, wie die Methode auf zweispurigen Verkehr erweiterbar ist.

Zwei Grenzfälle des Modelles werden im vorliegenden Text detailliert untersucht—der komplett deterministische Grenzfall, sowie ein "Speedomat"-Grenzfall, bei dem nur die Fluktuationen bei freiem Fahren (= bei Höchstgeschwindigkeit) auf Null gesetzt werden.

Der *komplett deterministische Grenzfall* ist, im hydrodynamischen Limes, äquivalent [85] zu einer linearen, deterministischen und nicht-diffusiven generalisierten Burgers-Gleichung. Diese wiederum stimmt mit der Lighthill-Whitham Theorie für kinematische Wellen überein [100], wenn man eine stückweise lineare, zeltförmige Fluß-Dichte-Beziehung einsetzt. Es gilt

$$\partial_t \rho + \mathrm{sgn}[\rho(q_{max}) - \rho]\, v_{max}\, \partial_x \rho = D\, \partial_x^2 \rho \ .$$

Der *Speedomat-Grenzfall* erlaubt stabile homogene Verkehrsflüsse bis zu einer gewissen Dichte $\rho(q_{max,det})$. Diese Flüsse kann man deterministisch konstruieren, da ja bei maximaler Geschwindigkeit keine Fluktuationen existieren. Allerdings gibt es noch eine weitere wichtige Dichte ρ_c mit folgender Eigenschaft: (i) Für $\rho < \rho_c$ lösen sich Störungen (Staus, z.B. nach einem Unfall) im Mittel (Ensemblemittel) auf. (ii) Für $\rho > \rho_c$ wachsen Störungen im Mittel immer weiter. (iii) Wenn $\rho = \rho_c$ ist, dann wachsen Störungen im Mittel nicht, schrumpfen aber auch nicht. Für einen *individuellen* Stau bei dieser Dichte hat das zur Folge, daß die Anzahl der Fahrzeuge im Stau entsprechend einem Random Walk fluktuiert. Als Folge sind Staus $\rho = \rho_c$ kritische Cluster im Sinne der Theorie der Kritischen Phänomene. ρ_c ist gleichzeitig die Dichte bei maximalen Flusses, *wenn man von zufälligen Anfangsbedingungen startet* ($\rho_c = \rho_{max,rnd.ini}$).

Speedomat-Verkehr zwischen ρ_c und $\rho_{max,rnd.ini}$ ist *marginal* stabil: Schon die geringste Störung führt zu einem immer weiter wachsenden Stau (entsprechend (ii)); und der Ausfluß von diesem Stau erfolgt mit Dichte ρ_c, also mit der kritischen Dichte (Selbstorganisierte Kritikalität [8]).

Ein dritter Spezialfall ergibt sich, wenn man die Maximalgeschwindigkeit des Modelles v_{max} auf eins setzt. Dies benimmt sich sehr ähnlich wie der bekannte Stochastische Asymmetrische Exclusions-Prozess (SAEP) [74, 34]. Beim SAEP ist die Reihenfolge des Teilchenupdates zufällig, d.h. daß jeweils ein Teilchen zufällig ausgewählt und dann entsprechend der Regeln einer *deterministischen* Dynamik bewegt wird. Beim CA-Modell hingegen werden alle Teilchen gleichzeitig angefasst, aber aufgrund einer stochastischen Dynamik bewegt. Nichtsdestotrotz benehmen sich bei $v_{max} = 1$ beide Modelle ähnlich, so z.B. in den kinematischen Wellen, die bei kleinen Dichten $\rho < \rho(q_{max})$ vorwärts und bei großen Dichten $\rho > \rho(q_{max})$ rückwärts laufen. Diese Ähnlichkeit zwischen den beiden Modellen verschwindet mit höherem v_{max}; in dem CA-Modell für Verkehr existieren z.B. vorwärts laufende Wellen überhaupt nicht mehr, unabhängig von der Dichte.

Der SAEP ist, im hydrodynamischen Limes, äquivalent zu einer generalisierten stochastischen Burgers-Gleichung mit einem linearen plus einem nicht-linearen Term [18]:

$$\partial_t \rho + \partial_x \rho - 2\,\rho\,\partial_x \rho = D\,\partial_x^2 \rho + \eta\ ,$$

wobei η ein Rauschterm ist. Dies entspricht, wenn man $D = \eta = 0$ setzt, wiederum der Lighthill-Whitham-Theorie, diesmal mit einer quadratischen Fluß-Dichte-Beziehung, die in den Verkehrswissenschaften Greenshields-Beziehung genannt wird.

Das vollständige CA-Modell benimmt sich nun ähnlich wie der Speedomat-Grenzfall. Allerdings sind jetzt *alle* Konfigurationen mit Dichten $\rho > \rho(q_{max,rnd.ini})$ nur noch metastabil, zerfallen also irgendwann spontan, und zwar in Stauwellen und Ausfluß-Regionen. Letztere sind wieder kritisch im oben beschriebenen Sinne (Fall (iii) im Speedomat-Grenzfall), allerdings nicht mehr *exakt* kritisch: Es exisitiert ein oberer Cut-off, der dadurch entsteht, daß spontan neu entstehende Staus alte austrocknen können; die Zeitskalen zwischen Staudynamik und Antrieb der Clusterdynamik sind also nicht mehr getrennt.

Man kann, insbesondere im Zusammenhang mit den Beobachtungen für $v_{max} = 1$, begründet vermuten, daß oberhalb dieses Cut-offs wieder die Burgers-Theorie gültig ist. Vergleich zwischen Stau-Messungen in der Realität und Simulationsergebnissen legen nahe, daß die Längenskala des Cut-off oberhalb von 10 km liegt.

Die spontane Phasentrennung kann durch die Lighthill-Whitham-Theorie nicht mehr beschrieben werden; fluid-dynamische Theorien für Verkehrsfluß können diesen Vorgang erst (und auch dann nur gemittelt) beschreiben, wenn eine Gleichung für den Impuls der Fahrzeuge eingeführt wird.

Für alle diese Teilchenmodelle gilt, daß der Ausfluß aus einer Region hoher Dichte automatisch den Zustand maximalen Flusses ($\rho = \rho(q_{max,rnd.ini})$) wählt. Interessant dabei ist vor allem, daß damit der kritische Zustand, also der Zustand hoher Fluktuationen, generisch auftritt.

Über die bisher resümierten Effekte hinaus können sich Staus noch *teilen*. Mangels genauer Meßdaten läßt sich dieses Verhalten für die Realität weder belegen noch widerlegen; es erscheint aber plausibel, und es tritt im Modell auf natürliche Weise auf, sobald $v_{max} > 1$, und läßt sich kaum wieder beseitigen. Dies führt zu “Löchern” zwischen verschiedenen Zweigen eines Staus, in denen ein Autofahrer wieder Höchstgeschwindigkeit fahren kann. Aus diesem Grunde haben Weg-Zeit-Plots der Stau-Dynamik ein fraktales Aussehen. Messungen der Verteilungsfunktion für die Löcher ergeben aber, daß die Staus *gerade so* nicht fraktal sind. Diese Struktur kann $1/f^{\alpha}$-Rauschen von Messungen erklären, insbesondere da es auch unter recht verschiedenen Umständen in einem Fahrzeugfolgemodell mit kontinuierlichen Variablen auftritt.

Zusammenfassend für diesen Teil der Arbeit, den "Verkehrsfluß-Teil", läßt sich sagen, daß die gründliche Untersuchung des Teilchenmodelles zu einem genauen Verständnis der auftretenden Phänomene geführt hat. Das Modell läßt sich in den Kontext existierender Teilchen-Modelle einbetten, wodurch sich ein direkter Zusammenhang mit fluid-dynamischen Theorien für Verkehrsfluß ergibt. Allerdings geht bereits dieses einfache CA-Modell über sämtliche derzeitigen fluid-dynamischen Theorien für Verkehrsfluß hinaus, da es nicht nur mittlere Staus beschreibt, sondern auch die möglichen Fluktuationen um diesen mittleren Zustand.

Weiterhin kann das kritische Verhalten des Modelles bei ρ_c durch kritische Exponenten beschrieben werden. Insbesondere da für diese Exponenten auch eine phänomenologische Theorie vorliegt, kann man annehmen, daß diese Exponenten *universal* sind und damit relativ unabhängig vom mikroskopsichen Modell gelten. Man kann also annehmen, daß auch wesentlich kompliziertere mikroskopische Verkehrsmodelle sich hierbei nicht anders verhalten werden. — Andererseits ist das in dieser Arbeit verwendete Modell minimal in dem Sinne, daß jede weitere Vereinfachung zu einem Verlust an gewünschten Eigenschaften führt.

Der zweite Teil der Arbeit beschreibt Simulations-Experimente in Straßennetzen. In einem einführenden Kapitel wird insbesondere herausgearbeitet, daß robuste Verkehrssimulationsmodelle aufgrund von individuellen Routenplänen funktionieren sollten (im Gegensatz zu aggregierten Abbiegewahrscheinlichkeiten). Dies wird intuitiv relativ schnell klar, wenn man einmal die Reaktionen beider Ansätze auf eine Straßensperrung untersucht (Abb. 2.2).

In Kapitel 9 wird dann ein einfaches Verkehrsnetz, bestehend aus einer Direktverbindung mit einem Engpaß, einer Ausweichstrecke sowie weiteren Verkehrsteilnehmern untersucht. Für diesen Fall stellt sich insbesondere heraus, daß das derzeit in der öffentlichen Diskussion befindliche Congestion Pricing in der Tat zu einer faireren Allokation der Resourcen (Straßenkapazität) führt. Dies ist zwar im Prinzip zu erwarten, allerdings ist unklar, inwieweit die bekannten ökonomischen Untersuchungen noch in der Nähe maximalen Durchflusses gültig sind, bei dem Verkehr nach den oben aufgezeigten Resultaten sehr start fluktuiert. Darüberhinaus ist die Bestimmung des Verkehrszustandes, die ja einer Preis-Entscheidung zugrunde liegen sollte, durch die Löcher zwischen den Staus stark erschwert.

Weiterhin kann die Simulation als Beispiel eines allgemeineren Phänomens dienen: Neben dem theoretischen Wissen über die Fluktuationen, das sich aus den Verkehrsfluß-Untersuchungen ergibt, wurde auch eine praktischere Größe gemessen: die Fluktuationen von Reisezeiten zwischen Fahrzeugen, die zu verschiedenen Zeiten, aber unter gleichen mittleren Bedingungen, die gleiche Route fahren. Interpretiert ist dies die Fehlerbreite einer Reisezeit-Vorhersage. Es stellt sich heraus, daß diese Fehlerbreite bei Dichten knapp oberhalb des Regimes maximalen Flusses stark anwächst. Statt einer Vorhersageungenauigkeit von ±3% bekommt man im Modell plötzlich mehr als ±65%; eine vorhergesagte mittlere Reisezeit von 2 Stunden kann also leicht mehr als 3 Stunden dauern, und eine bessere Vorhersage ist auch aus *prinzipiellen* Gründen nicht möglich.

Daraus ergibt sich das Problem, daß sich das System einer weiteren Optimierung entzieht, da man ohne verläßliche Vorhersage nicht optimieren kann. Verkehrsmanagement (ATMS) wird also über diesen Punkt hinaus nicht möglich sein. Daraus ergibt sich die Notwendigkeit, diejenigen Zustände im Verkehrssystem zu finden, die einer Optimierung tatsächlich zugänglich sind.

In Kapitel 10 werden dann Simulationen des Autobahn-Netzes von Nordrhein-Westfalen vorgestellt. Aufbauend auf Arbeiten von Rickert [141], werden individuelle Routenpläne implementiert. Jedes

Fahrzeug weiß also an jeder Kreuzung individuell, welchen Weg es einschlagen will. Für die Simulationen werden alle Fahrzeuge an Randknoten des Netzes aufgereiht. Jedes Fahrzeug wählt einen anderen Randknoten als Ziel; daraufhin erhält es eine Liste der 10 geometrisch kürzesten Wege zu diesem Ziel. Die Fahrzeuge fahren in der Simulation entsprechend ihrer Aufreihung los, wobei sie zufällig einen Wege aus ihren 10 Möglichkeiten wählen. Nachdem alle Fahrzeuge angekommen sind, wird die Simulation wieder gestartet. Auf diese Weise sammeln alle Fahrzeuge Erfahrungen über die Performances der verschiedenen Routenvorschläge; und nach einiger Zeit wählen sie im allgemeinen die schnellste Route.

Es stellt sich heraus, daß man mit diesem Verfahren die Fahrzeugströme im Straßennetz erfolgreich equilibrieren kann—und zwar auch im Nichtgleichgewicht eines überbelasteten Straßennetzes, wo konventionelle Methoden versagen. Darüberhinaus sind die Resultate erstaunlich robust, und zwar sowohl von einem simulierten "Tag" auf den nächsten, als auch unter verschiedenen Annahmen für die Zielknoten-Auswahl.

Die Simulation behandelt 60 000 Fahrzeugen gleichzeitig, und läuft auf 2 gekoppelten Sparc10 Workstations etwa 5 mal schneller als Echtzeit. Ein Simulationslauf bis zur Relaxation des Lernverhaltens dauert damit etwa 36 Stunden. Damit ist die Methode schon auf relativ moderater Hardware schnell genug zur Durchrechnung von Szenarien. Durch gezielte Untersuchungen auf zahlreichen Höchstleistungsrechnern wurde darüberhinaus nachgeprüft, daß der Ansatz dort effizient läuft, so daß auch aufwendigere Monte-Carlo-Untersuchungen und eventuell Echtzeit-Anwendungen möglich sind.

LEBENSLAUF: KAI NAGEL

Geburtsdatum:	17. September 1965
Geburtsort:	Köln
Eltern:	Katrin Nagel geb. Köhler, Hellmut Nagel
1971 - 1975:	Grundschule
1975 - 1984:	Gymnasium Kreuzgasse, Köln
1984:	Abitur
1984 - 1985:	Wehrdienst
1985 - 1991:	Studium der Physik und der Meteorologie an der Universität zu Köln
1987:	Vordiplom in Physik (Köln) Vordiplom in Meteorologie (Köln)
1988 - 1989:	Studium in Paris (Frankreich). Abschluß mit 'DEA' (≈ Diplom) in Meteorologie
Sept. 1991:	Diplom in Physik (Köln)
Okt. 1991 - Juni 1993:	Wissenschaftlicher Mitarbeiter an der Universität zu Köln (Mathematisches Institut, Prof. A. Bachem)
seit Juli 1993:	Stipendiat des Graduiertenkolleg Scientific Computing Köln / St. Augustin
Juli 1993 - Juli 1994:	Los Alamos National Laboratory, Santa Fe Institute, und Brookhaven National Laboratory (P. Bak, C. Barrett, S. Rasmussen)
Seit Juli 1994:	Zentrum für Paralleles Rechnen an der Universität zu Köln (Prof. Dr. A. Bachem)

ERKLÄRUNG

Ich versichere, daß ich die von mir vorgelegte Dissertation selbstständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; daß diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; daß sie abgesehen von unten angegebenen Teilpublikationen noch nicht veröffentlicht worden ist sowie, daß ich eine solche Veröffentlichung vor Abschluß des Promotionsverfahrens nicht vornehmend werde. Die Bestimmungen dieser Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. A. Bachem betreut worden.

Teilveröffentlichungen:

- Nagel K, Schreckenberg M, A cellular automaton model for freeway traffic, *J. Phys. I France* **2** (1992) 2221.
- Nagel K, Schleicher A, Microscopic traffic modeling on parallel high performance computers, *Parallel Comput.* **20** (1994) 125-146.
- Nagel K, Herrmann H J, Deterministic models for traffic jams, *Physica A* **199** (1993) 254.
- Nagel K, Life-times of simulated traffic jams, *Int. J. Mod. Physics C* **5(3)** (1994) 567.
- Nagel K, Rasmussen S, Traffic at the edge of chaos, Brooks R, Maes P (eds.), Proceedings of the Alife 4 meeting, MIT press, Cambridge, MA, 1994.
- Nagel K, Paczuski M, Emergent Traffic Jams, *Phys. Rev. E*, in press.
- Schreckenberg M, Schadschneider A, Nagel K, Ito N, Discrete stochastic models for traffic flow, *Phys. Rev. E*, in press.