# 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**
      $v := \max(gap - \mathrm{int}(0.5 + \mathrm{rnd}()), 0)$
    **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:

  $$\begin{array}{l} \text{IF } t_{start}(n,t) = t_{start}(n+1,t-1) \text{ THEN} \\ \quad\quad lbl(n,t) = lbl(n+1,t-1) \\ \text{ELSE IF } t_{start}(n,t) = t_{start}(n,t-1) \text{ THEN} \\ \quad\quad lbl(n,t) = lbl(n,t-1) \\ \text{ELSE} \\ \quad\quad lbl(n,t) = newlbl \\ \text{ENDIF} \end{array}$$

  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 ($4 \times 128$) 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 $\rho$ 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 $\alpha$, $\beta$, $\gamma$, and $v_{lead}$ within the following range $0.1 \le \alpha \le 0.6$, $2.0 \le \beta \le 5.0$, $0.08 \le \gamma \le 0.12$, $4.5 \le v_{lead} \le 4.99999$.

In order to quantify these observations, the distribution of times $\tau$ between consecutive "braking" events for the last vehicle ($\tau$ 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 $\tau$ 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

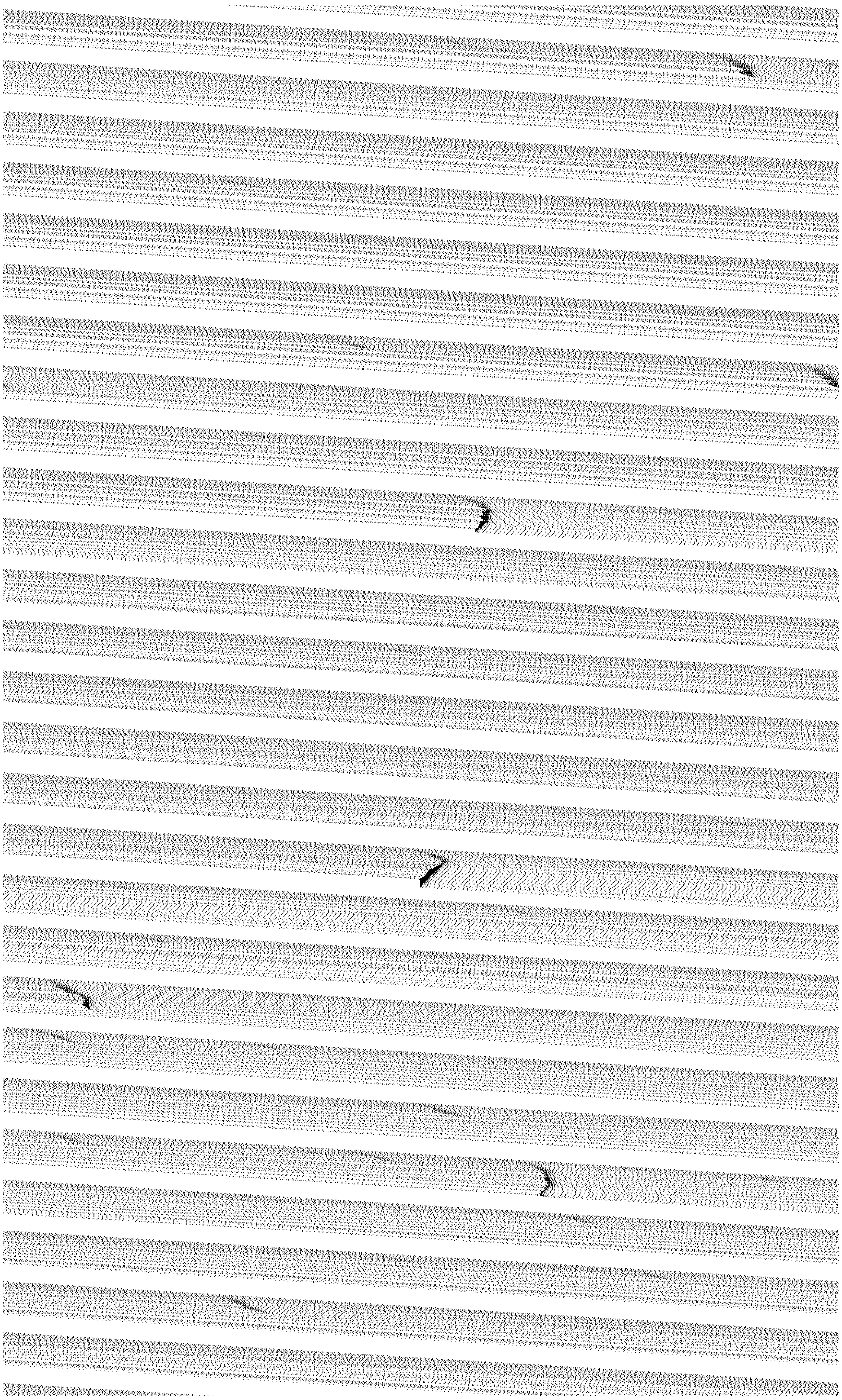

Figure D.1: Evolution of the continuous model (5600 time steps), $N = 61$, $L = 1024$. As before, consecutive lines represent configurations at consecutive time steps; vehicles are denoted by black squares, empty spaces are left blank.

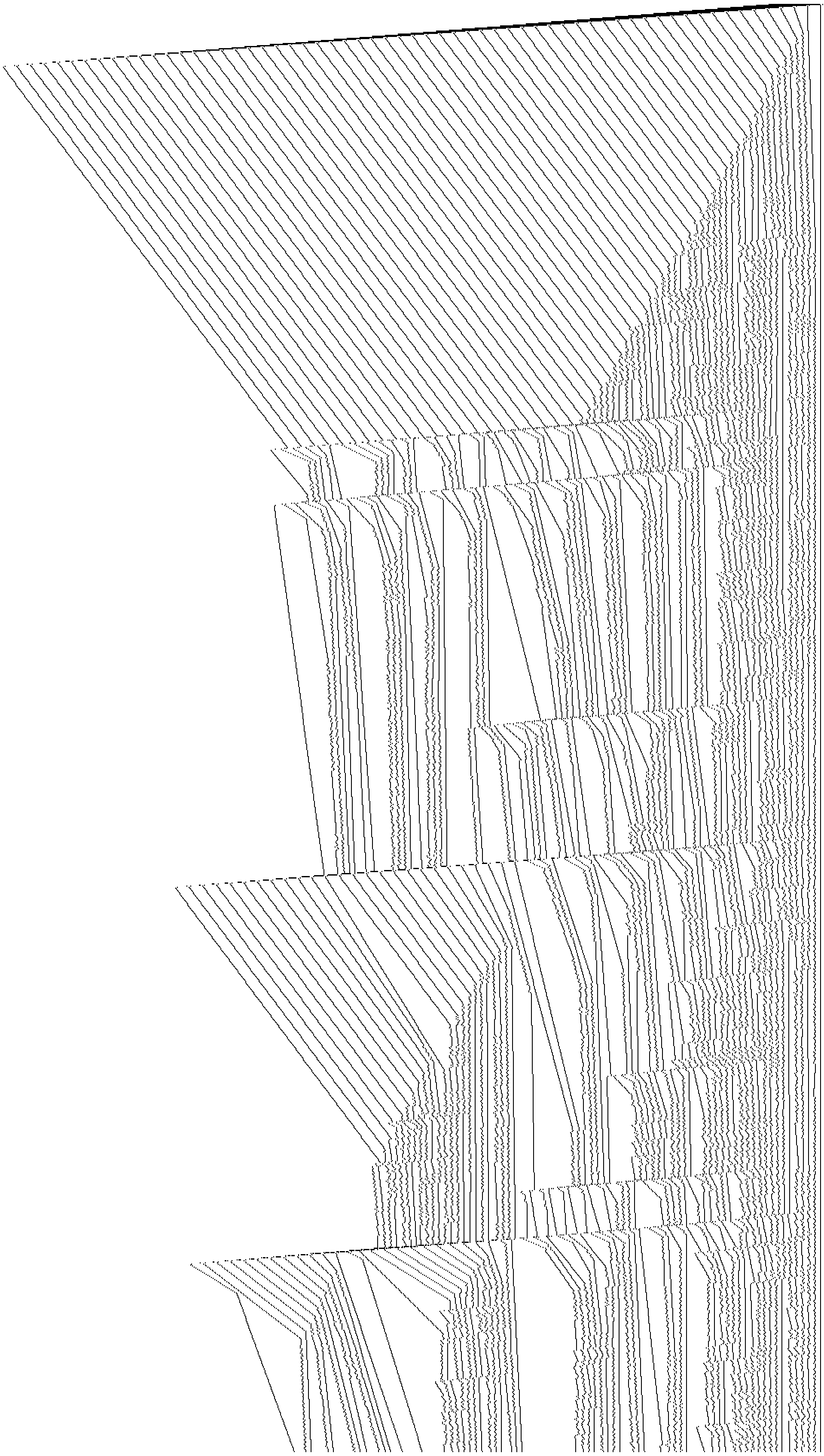

Figure D.2: (Previous page) First 4200 time-steps of the continuous model, transformed to the coordinates of the leading (rightmost) vehicle. Only every third time-step is plotted, everything else is as in Fig. D.1. The jam-waves are now clearly visible as nearly horizontal lines. The flow is from left to right.

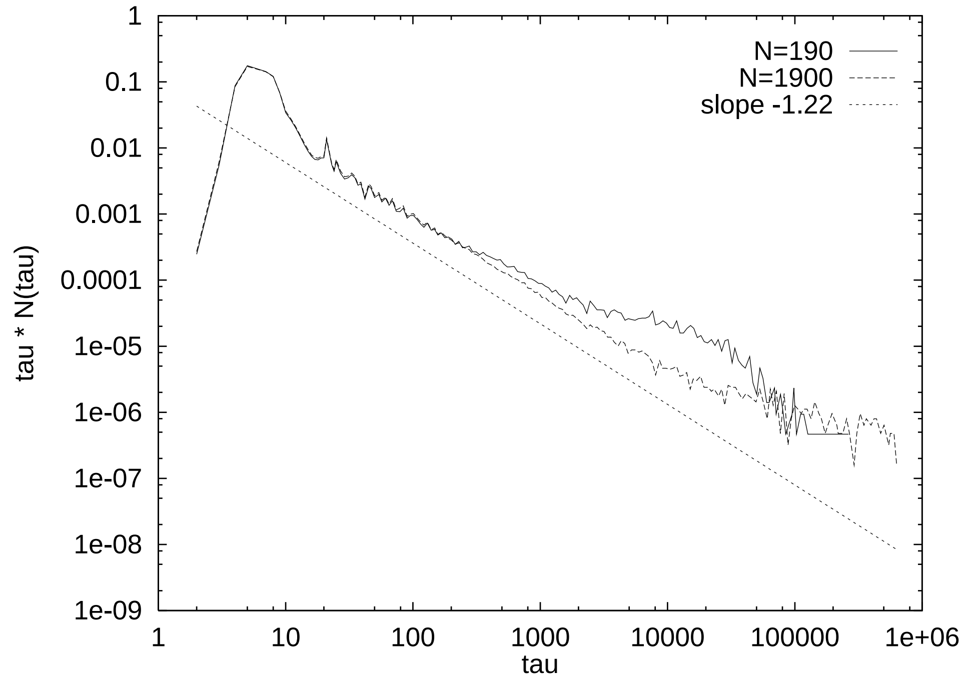


Figure D.3: Distribution $n(\tau)$ of times $\tau$ between consecutive braking events for $N = 190$ and $N = 1900$ cars. (The times are collected in "logarithmic bins", so that the $y$-axis is proportional to $\tau\, n(\tau)$.)

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**Putational **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 ($m = 2 \dots L-1$).

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:
$$t_{step}(n) = t_{sync}(n) + t_{bnd} + max(t_{async}(n), t_{calc}(l))$$

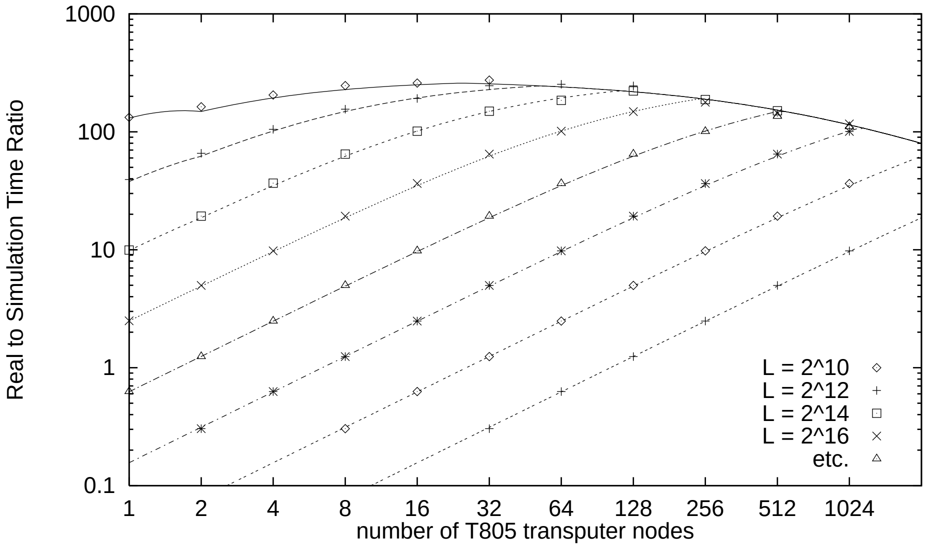


Figure F.1: Real to Simulation Time Ratio (measurements and approximating theoretical curves) as a function of the number of CPNs for different system sizes.

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 $\epsilon$ 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 $\epsilon$ 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

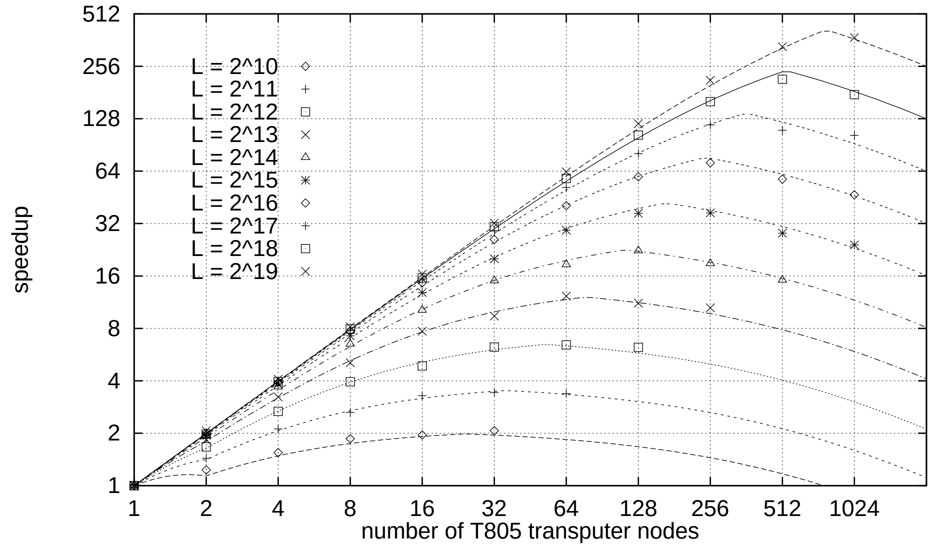


Figure F.2: Speedup (measurements and approximating theoretical curves) as a function of the number of CPNs for different system sizes.

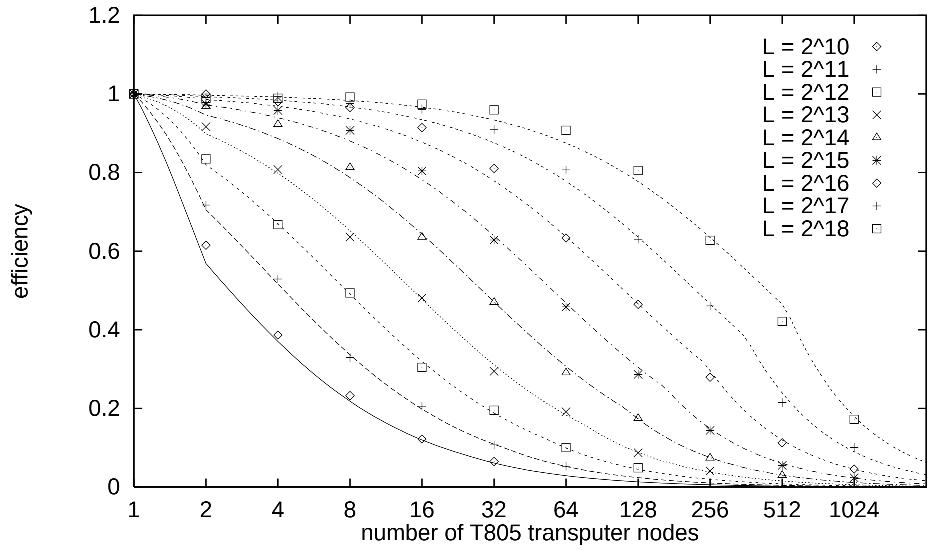


Figure F.3: Efficiency (measurements and approximating theoretical curves) as a function of the number of CPNs for different system sizes.

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

$$
\begin{aligned}
s &= \frac{l}{t_{step}} \\
&= \frac{l}{t^*_{sync} + t_{bnd} + max(t_{async}, t_{calc})}
\end{aligned}
$$

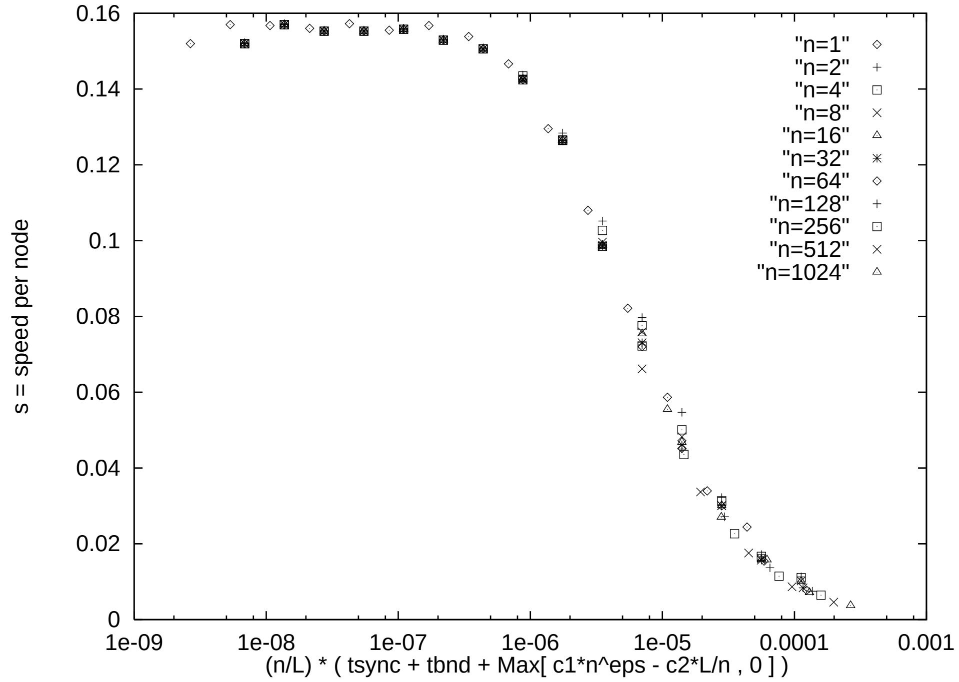


Figure F.4: Scaling Plot. This is the rescaled version of Fig. F.3.

$$
\begin{aligned}
&= \frac{1}{c_2 + (n/L)(t_{sync} + t_{bnd} + max(t_{async} - c_2 \cdot L/n, 0))} \\
&=: \frac{1}{c_2 + n_{scaled}} \,.
\end{aligned}
$$

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, Rują́n 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. Rev. 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 $\rho$ 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 $\rho_c$ 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. $\rho_c$ ist gleichzeitig die Dichte bei maximalen Flusses, *wenn man von zufälligen Anfangsbedingungen startet* ($\rho_c = \rho_{max,rnd.ini}$).

Speedomat-Verkehr zwischen $\rho_c$ 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 $\rho_c$, 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 $\eta$ 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 $\rho_c$ 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 $\pm 3\%$ bekommt man im Modell plötzlich mehr als $\pm 65\%$; 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' ($\approx$ 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.