PARALLEL TRAFFIC FLOW SIMULATION OF FREEWAY NETWORKS, PHASE 1

FINAL REPORT

Principal Investigator: Anthony Chronopoulos

Graduate Students: Choong-Yul Rhee and Gang Wang

Computer Science Department
University of Minnesota

March 1994

Parallel Traffic Flow Simulation of Freeway Networks (Phase 1 Report)

A. Chronopoulos *

Abstract

Numerical methods for solving simple macroscopic traffic flow continuum models have been studied and efficiently implemented in traffic simulation codes. The Lax method is an explicit method which has been implemented to solve simple continuum model in the traffic simulation package KRONOS. In this part of the project we studied the implementation of the Lax method for solving a high-order flow conservation traffic model on Parallel Computers. We wrote an experimental code in C to simulate a freeway traffic flow. Tests with real data collected from the I-35 W freeway in Minneapolis were conducted on a workstation computer. We then implemented the high order Lax method on a parallel machine and run tests with real data collected from an 18-mile stretch of the I-494 freeway. The parallel implementation demonstrated significant execution time speedup.

*Department of Computer Science, University of Minnesota. The Project was funded through the Center of Transportation Studies. The Computer Science graduate students Choong-Yul Rhee and Gang Wang worked on this project. Dr. Ping Yi of MnDot/GuideStar provided advice for this Project. Mr. David Berg of MnDot provided help with the freeway traffic data.

The financial support of oil overcharge funds distributed through the Minnesota Department of Administration is acknowledged, but the authors assume complete responsibility for the contents herein.

1 Introduction

Macroscopic continuum traffic models flow based on traffic density, flow and velocity have been proposed and analyzed in the past. Examples include Lighthill and Whitham's (1955) flow conservation model,

Payne's momentum model conservation model and Michalopoulos's momentum model [8], [12], [11]. These models involve partial differential equations (PDEs) defined on appropriate domains with suitable boundary conditions which describe various traffic phenomena and road geometries.

The improvement of computational efficiency in the continuum traffic models has been the focal point in the development of traffic simulation programs. It is understood that the computer execution time to solve traffic flow problems depends not only on the size of the freeway and the complexity of roadway geometries, but also on the model equations and numerical schemes used in their discretization.

Explicit numerical methods (for example Lax, Upwind) have been used by Michalopoulos and Lin and Leo and Pretty to compute the solution of traffic flow continuum models [10], [7]. In these explicit schemes the space and time mesh sizes are restricted both by accuracy and numerical stability requirements. In order to reduce the computer execution time and maintain good accuracy, the total number of computations must be reduced. This can be achieved by using larger values of time and space mesh sizes. Implicit numerical methods provide the same accuracy as explicit methods and allow changes in the mesh sizes while maintaining numerical stability (see [3], [4], [5]).

In this work we use the Lax method to solve more efficiently the momentum conservation model on a parallel computer. We wrote an experimental code in C simulating a freeway traffic flow. Tests with real data collected from the I-35 W freeway in Minneapolis were conducted. These data have been collected by the Minnesota Department of Transportation. Using these data we tested (for accuracy and efficiency) our code on a Sun Sparc1 workstation computer. We then implemented efficiently the Lax-Momentum method on the (16 processor) NCUBE2 parallel computer located at the Department of Computer. Each processor of the NCUBE2 is as powerful as a SUN 3/50 workstation. We run tests with real data from the I-494 freeway in Minneapolis. On the NCUBE2, the parallel Lax-Momentum method on the 16 processors run 13 times faster than on the one processor.

The outline of this article is as follows. In section 2, we review the momentum conservation continuum traffic model. In section 3, we review the the Lax method. In section 4, we describe the various theoretical and empirical curves relating the traffic flow and density. In section 5, we describe the freeway model. In section 6, we describe the parallel implementation of the Lax method. In section 7, we present the numerical results. Section 8 contains concluding remarks.

2 A Simple and High-Order Continuum Model of Traffic Flow

The following conservation equation has been proposed by Lighthill and Whitham (1995) [8] as a simple continuum traffic model:

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = g(x,t), \tag{1}$$

where $k(x,t)$ and $q(x,t)$ are the traffic density and flow respectively at the space-time point (x,t). The generation term $g(x,t)$ represents the number of cars entering or leaving the traffic flow in a freeway with entries/exits. The traffic flow, density and speed are related by the equation:

$$q = ku, \tag{2}$$

where the equilibrium speed $u(x,t) = u(k)$ must be provided by a theoretical or empirical u-k model. The theoretical u-k model, equation of state, can take the general form.

$$u_e = u_f[1 - (k/k_{jam})^{\alpha}]^{\beta}, \tag{3}$$

where u_f is the free flow speed and k_{jam} the jam density [2]. For instance, for $\alpha = 1$ and $\beta = 1$, one obtains the Greenshield's(1934) equation of state. More information on this and other forms of the u-k relationships can be found elsewhere (Mcshane and Roess 1990) (see [9]). Since the simple continuum model does not consider acceleration and inertia effects, it does not faithfully describe non-equilibrium traffic flow dynamics.

The high-order continuum formulation takes into account acceleration and inertia effects by replacing Equation (3) with the momentum equation (Eq. 4).

$$\frac{du}{dt} = \frac{1}{T}[u_f(x) - u] - G\frac{\partial u}{\partial t} - \nu k^{\beta}\frac{\partial k}{\partial x} \tag{4}$$

where $\frac{du}{dt}$ is the acceleration of an observer moving with the traffic stream and is related to the acceleration $\frac{\partial u}{\partial t}$ of the traffic stream as seen by an observer at a fixed point of the road, i.e.

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} \tag{5}$$

The first term on the right hand side of Eq. 4, $\frac{1}{T}[u_f(x) - u]$, represents the relaxation term, the tendency of traffic flow to adjust speeds due to changes in $u_f(x)$ along the roadway, where relaxation time T is assumed to vary with density k according to

$$T = t_0(1 + \frac{rk}{k_{jam} - rk}) \tag{6}$$

where $t_0 > 0$ and $0 < r < 1$ are constants. The second term, $G\frac{\partial u}{\partial t}$, addresses the traffic friction at freeway ramp junctions due to ramp flows. G is the friction parameter. It is a function of both roadway conditions and the ramp volume entering or leaving the freeway and is derived experimentally as $G = \mu k^{\epsilon} g$, where μ is a geometry parameter depending on the type of road geometry, ϵ is a dimensionless constant, and g is the generation term. The third term, $\nu k^{\beta}\frac{\partial k}{\partial x}$, represents the anticipation term which is the effect of drivers reacting to downstream traffic conditions. In this term ν is the anticipation paramter. As implied in this example, if downstream density is higher due to congestion, speed has to be decreased accordingly. Conversely, if downstream density is lower, speed can be increased. From equations (4) - (6) one derives a momentum model for the traffic flow described by the following system of PDEs.

$$\frac{\partial \vec{\mathbf{U}}}{\partial \mathrm{t}} + \frac{\partial \vec{\mathbf{E}}}{\partial \mathrm{x}} = \vec{\mathbf{Z}} \tag{7}$$

where $\vec{U}$, $\vec{E}$, and $\vec{Z}$ are the following vectors:

$$\vec{U} = \begin{pmatrix} k \\ q \end{pmatrix}$$

$$\vec{E} = \begin{pmatrix} ku \\ u^2 k + \frac{\nu}{\beta+2} k^{\beta+2} \end{pmatrix}$$

$$\vec{Z} = \begin{pmatrix} g \\ \frac{k}{T}[u_f(x) - u] - Gk\frac{\partial u}{\partial t} + gu \end{pmatrix}$$

We note that the momentum conservation model does not require a $q-k$ curve as in the case of the simple continuum model. However speed data are not available from the the real traffic data then a $q-k$ curve is used to generate the speed data.

3 Numerical Methods

We consider one high-order explicit method (Lax) and one high-order implicit method (Euler implicit) which are used in computational fluid dynamics [3]. For each traffic model the road section (the space dimension) is discretized using uniform mesh for all numerical methods; only the time stepsizes differ between methods. We use the following notation:

Δt = time stepsize.

Δx = space stepsize.

k_j^n = density (vehicles/mile/lane) at space node $j\Delta x$ and at time $n\Delta t$.

q_j^n = flow (vehicles/hour/lane) at space node $j\Delta x$ and at time $n\Delta t$.

u_j^n = speed (mile/hour) at space node $j\Delta x$ and at time $n\Delta t$.

3.1 Lax Method

The high-order Lax method is an explicit method. The new density value k_j^{n+1} and flow value q_j^{n+1} are computed directly from the density and flow at the preceding time step n:

$$\vec{U}_j^{n+1} = \frac{\vec{U}_{j+1}^n + \vec{U}_{j-1}^n}{2} - \frac{\Delta t}{\Delta x}\frac{\vec{E}_{j+1}^n - \vec{E}_{j-1}^n}{2} + \frac{\Delta t}{2}(\vec{Z}_{j+1}^n + \vec{Z}_{j-1}^n). \qquad (8)$$

The method is of first order accuracy with respect to Δt, i.e. the error is $O(\Delta t)$. To maintain numerical stability time and space stepsizes must satisfy the CFL condition $\frac{\Delta x}{\Delta t} > u_f$, where u_f is the free flow speed. For example in the KRONOS traffic simulation code (using Lax) $\Delta x = 100 feet$ and $\Delta t = 1 sec$ are recommended.

4 Volume-Density (q-k) Model Curves

A $q - k$ model curve is an indispensable part of the simple continuum model. This relation can be used to express the flow rate as a function of the flow density i.e. $q = q(k)$. This function is a nonlinear function which must satisfy some general requirements. The equations that define the $q - k$ curve are used in the programs to convert from density to flow and from flow to density. The Momentum Traffic Flow Model does not require a $q - k$ curve. However, if the speed data are not available from the traffic data then a $q - k$ curve is used to compute the traffic speed.

These general requirements on the $q - k$ curve can be derived from the following observations on traffic flow modeling [9].

- For uncongested flow an increase in density corresponds to an increase in flow, up to a critical density k_c, where the flow becomes congested.
- Maximum flow occurs at the critical density: $q_{max} = q(k_c)$.
- For congested flow an increase in density corresponds to a decrease in flow, up to the jam density k_{jam}, where flow stops.

A q-k model curve must also be adapted to characteristics of the freeway section which it represents. Theoretical q-k model curves can not be adapted to the special roadway characteristics and so such a model function must be constructed from empirical data. Greenshields q-k curve is derived from equations (2) and (3) and appropriate choices for the free flow speed u_f and jam density $k_{jam} = k_0$. In our applications we chose u_f =(60 miles/hour) and k_0 =(180 vehicles/mile). The Greenshields curve has the basic features described above but can not be tuned to local characteristics of a freeway section. However, we used Greenshield's q-k curve in the initial development of the programs and as a baseline for comparisons.

4.1 Experimental q-k Model Curves

Field data for constructing the q-k model curve were collected in I-35 W in Minneapolis. With these discrete data a piecewise linear q-k curve was derived [11]. Such a curve must have parameter ranges reflecting the road characteristics of the freeway section it represents [?]. With our discrete data the experimental q-k curve must have following parameter ranges:

- The critical density k_c should be about 70 to 75 vehicles/mile/lane.
- The maximum flow q_{max} should be less than 2500 to 2700 vehicles/hour/lane.
- The slope of the curve at $k = 0$, which represents the free-flow speed u_f, should be approximately 65 to 75 miles/hour.

We have used several curve fitting methods to construct continuous q-k curves from the set of (k, q) discrete data points available. Our objective was to find a general method that produces a curve which is based on the discrete data, has the basic features of a q-k curve, has the parameter ranges (described above), and also works well in the numerical methods for solving (1). We used three different methods **piecewise linear**, **cubic spline**, and **least squares** to approximate q-k curves from field data.

The simplest method consists of connecting the q-k data points with straight line segments, yielding a **piecewise linear** q-k curve. This is a continuous curve that passes through all data points but the slope of the curve (which is used in the implicit methods) is discontinuous at the line segments intersections.

In an effort to find a curve that interpolated all of the q-k data points and that also had a continuous first derivative, we constructed a **cubic spline.** The cubic spline is a collection of third-degree polynomials, one polynomial for each interval between q-k data points. We tested both clamped (slope at endpoints is specified) and natural (slope at endpoints is unspecified) splines and found that for our field data set the splines were nearly identical. All cubic spline programs used the natural cubic spline.

Finally, several **least squares** approximations were tried. In this method the data points (k_i, q_i) are used to construct a rectangular matrix with row i composed of powers of k_i and a right-hand-side vector containing the q_i. Then the matrix is reduced using the singular value decomposition method (SVD) available in the LINPACK package or the Matlab package [1]. The reduced matrix is then used to find the coefficients of the curve that minimizes the total squared error between the data points and the curve. This method will produce curves of any degree up to the number of data points. Quadratic, cubic and quartic least-squares polynomial curves were found using the Matlab's (SVD). The quartic curve

$$q = -1.7156 \times 10^{-5} k^4 + 7.1802 \times 10^{-3} k^3 - 1.2514 k^2 + 94.8463 k - 69.1588$$

appeared to be the lowest-degree least-squares approximation to the discrete data that satisfies the q-k curve criteria. This quartic polynomial must be evaluated at each node for each time step so it is important to use a polynomial of the least degree. The choice of q-k curve was found to have a large effect on the stepsize selection of the implicit methods. Implicit methods using the smooth q-k curves generated by the least squares and Greenshield's methods were able to use larger time steps than the programs using cubic spline or piecewise linear curves.

4.2 Density estimation using occupancy

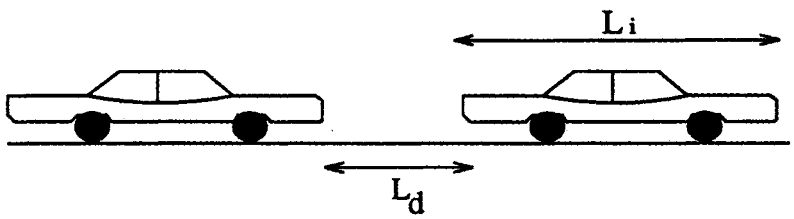


Figure 1. Density/Occupancy calculation from vehicle relative positions

Lane occupancy is defined as the time that detector is on divided by the measured time and multiplied by 100. The value of lane occupancy is available from the single detector installed under the freeway pavement. If we assume that the speed of the vehicle is constant during measurement time and each vehicle's length is the same, we can derive the relationship between density and occupancy.

$k = 52.8 \times \frac{\phi}{L_e}$, where effective length $L_e = \frac{\sum_{i=1}^{N}(L_i+L_d)}{N}$, ϕ=lane occupancy, L_i=vehicle length, L_d=intervehicle distance.

5 Freeway Model with Multiple Entries/Exits

We considered two multiple entry/exit freeways

- A section of I-35W Northbound for the workstation implementation
- A section of I-494 Eastbound for the parallel computer implementation

The I-35W roadway geometry is presented in Figure (2). The upstream and downstream boundaries were set at the location of the 86th street and the 63th street. It has two weaving sections at the first two entry/exit zones as shown in Figure(2). Data were collected by the Minnesota Department of Transportation on November 7, 1989 for I-35W. We have used two schemes to add merge/diverge traffic volumes to the mainlane traffic flow and density. 1) Ramp volumes are assumed to merge into(diverge from) the mainlane freeway at a single node. This treatment is necessary to simplify the modelling and reduce computation time at such mainlane nodes. 2) Ramp volumes are assumed to merge evenly into(diverge from) the mainlane freeway at the acceleration/deceleration nodes. The road geometry of I-494 is similar. The I-494 Eastbound section extends from the Carlson Pwy to Portland Avenue.

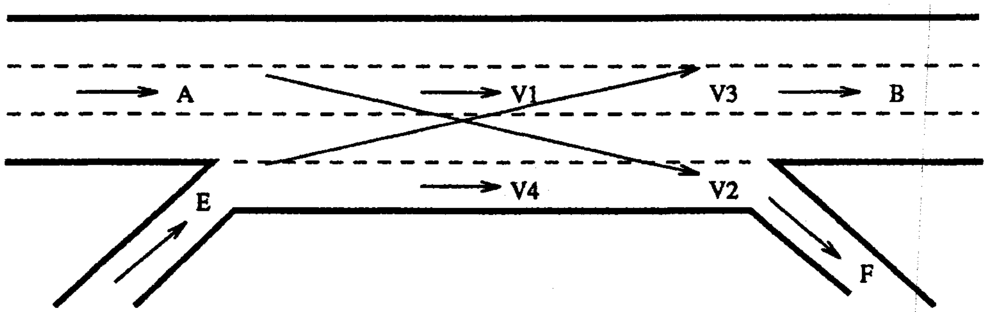


Figure 2. Weaving flows in a freeway

In Figure (2), flow v_1 represents the through traffic stream flow in from link A to link B and flow v_2 represents the diverging stream from link A to link F, where $q_A = v_1 + v_2$; v_3 is the merging stream from link E to link B and flow v_4 is the through stream from link E to link F, and $q_E = v_3 + v_4$. It is obvious that $q_F = v_2 + v_4$ and $q_B = v_1 + v_3$. Because there are interchanges of v_2 and v_3, traffic friction at link B and link E in this case is greater than the case of a single entrance ramp or exit ramp. Likewise, merging dynamics at an entrance ramp should be employed if $v_2 = 0$.

When L is less than 600ft, merging and diverging movements must be completed within a short distance. In such a case a net value of the merging and exiting flows is sought for flow conservation, $g = q_{net} / \Delta x = (q_{in}^n - q_{out}^n) / \Delta x$, where q_{in}^n and q_{out}^n are the merging and exiting flows. If $g > 0$, the short weaving section is treated as a single on ramp, if $g < 0$, it is treated as a single on-ramp. However, since both q_{in}^n and q_{out}^n require lane changing at the same limited length of roadway at the same time, the sum of q_{in}^n and q_{out}^n should be included in the generation term.

Operation	Time	Comm/Comp
8 Byte transfer	111 μ sec	-
8 Byte Add	1.23 μsec	90 times
8 Byte Multiply	1.28 μsec	86 times

Table 1: Computation and Communication times on the NCUBE2

6 Parallel Lax

We have implemented the Lax - Momentum method on the NCUBE2 parallel computer at the Department of Computer Science of the University of Minnesota. The NCUBE2 has 16 processors connected in a hypercube network and a host (Sun 3/50) computer for interaction with the user. The number of (N) processors to be active is chosen by the user, but must be a power of 2. The host computer allocates N processors arranged in an m-dimensional hypercube, where $m = \log_2 N$. Each of the N processors is directly connected to m other processors. In table 1 we show a summary of inter processor communication times and basic floating point operation times [6]. We see that communication even between neighbor processors is several times slower than floating point operations. Programs run most efficiently when inter processor communication is minimized and when all communication occurs between neighbor processors.

In the parallel implementation of the Lax-Momentum method we partition a freeway section into N (equal) segments and assign each segment to one processor, for $N = 1$ and 16. Boundary data, upstream and downstream traffic volume and speeds are stored in processor 0 and processor 15 respectively. At each time step the values for density and flow at the segment boundary points must be exchanged between processors. Road boundary conditions are handled by the processors assigned the upstream and downstream segments. When the computations are completed the processors with check station send their output to screen or outfile.

We have tested this implementation with eighteen mile stretch of interstate highway I-494. This I-494 East Bound data has twenty one on-ramps and eighteen off-ramps and twenty six test sites. Each processor only has 17 nodes to simulate. We used $\Delta t = 1sec$, total simulation time = 7200 sec, and $\Delta x = 200$ ft.

7 Results

The following statistics are used to measure the effectiveness of the simulation quantitively.

$$Mean\ \ Absolute\ \ Error = \frac{1}{N}\sum_{i=1}^{N}|Observed - Simulated| \qquad (9)$$

$$Mean\ \ Relative\ \ Error = \frac{1}{N}\sum_{i=1}^{N}\frac{|Observed - Simulated|}{Observed} \qquad (10)$$

$$Maximum\ \ Relative\ \ Error\ \ with\ \ 2\text{--}Norm = [\frac{\sum_{i=1}^{N}(Observed - Simulated)^2}{\sum_{i=1}^{N}Observed^2}]^{1/2} \qquad (11)$$

where N is the number of observations.

$$Standard\ \ Deviation = \frac{1}{N}[\sum_{i=1}^{N}(Observed - Simulated)^2]^{1/2} \qquad (12)$$

The arrival and departure pattern for the tests with I-35W North and are shown in Figures (3), (4) and for the tests with I-494 East in Figures (5) and (6) respectively. To test the program, the time stepsize selection was made as follows. For the Lax method we set $\Delta t = 1sec$. This is required to maintain numerical stability. We measured the error of the simulation on I-35W with weaving for the freeway weaving ramps and with average and point merging schemes. The error statistics are summarized in tables (2),(3), (4) and (5). The errors are very reasonnable for the volume but are higher for the speed measurements. From the tables (2) and (4), we can see small improvement at the test site 1 and 2 when we apply the point weaving at the weaving area. The overall average relative error in traffic volume is about 10 percent. There is little difference between using Average merge/exit scheme and point merge/exit scheme in traffic volume error as we see in test sites 3, 4, and 5 of tables (2) and (4).

The table (6) shows each processor's execution time (in secs) for simulating 18 mile's interstate highway I-494 using 16 processors. The Sixteen processors' execution time is about 13 times faster than that of single processor's execution time since single processor's execution time is about 20 secs.

8 Conclusions

We studied a high-order continuum models using an explicit (Lax) method. We wrote an experimental code in C simulating a freeway (un)congested pipeline and freeway entry/exit traffic flow. Tests with real data collected from the I-35 W freeway in Minneapolis were conducted on a workstation computer. The Lax explicit method were used to simulate the eight mile stretch of I-35W freeway section for which we have real data. We have also implemented efficiently the Lax-Momentum method on a (16 processor) NCUBE2 parallel computer. The single processor of the NCUBE2 is as powerful as a workstation processor. The parallel Lax-Momentum method was 13 times faster than the single processor one. This implementation shows there is a lot of potential in applying parallel processing to carry out real time traffic simulations.

References

[1] Dongarra J.J. et al.(1979), "LINPACK user's guide. " *SIAM publications.*

[2] Greenshields, B. D. (1934), "A study of traffic capacity ", *Proc. Highway Res. Board, Vol. 14, pp. 448-477.*

[3] Hirsch C. (1988), "Numerical Computation of Internal and External Flows ", *Vol.2, John Wiley and Sons.*

[4] A. T. Chronopoulos, P. Michalopoulos and J. Donohoe (1992), "Efficient Traffic Flow Simulation Computations", *Mathematical and Computer Modelling, Vol. 16, No. 5, pp. 107-120*;

[5] A.T. Chronopoulos, P. Michalopoulos and C. Rhee (1993), "Traffic Flow Simulation Through High Order Traffic Modelling", *Mathematical Computing Modelling, Vol. 17, No. 8, pp. 11-22*

[6] S. K. Kim and A. T. Chronopoulos, " A Class of Lanczos-like Algorithms Implemented on Parallel Computers", Parallel Computing 17, pp. 763-778, 1991.

[7] C. J. Leo and R. L. Pretty (1990), "Some Comments and Numerical Tests on Upwind Finite Difference Schemes for Macroscopic Traffic Modeling", *Dept. of Civil Eng., Univ. of Queensland, Australia.*

[8] Lighthill, M. H., G. B. Witham (1955), "On Kinematic waves: II A theory of traffic flow on long crowded roads", *Proceed. R. Soc. Ser. A 229, No. 1178, pp. 317-345.*

[9] McShane W. and Roess R. (1990), "Traffic Engineering ", *Prentice Hall.*

[10] Michalopoulos P. G., J. Lin (1985), " A freeway simulation program for microcomputers ", *Proceed., 1st National Conf. on Microcomputers in Urban Transp., ASCE, California, pp. 330-341.*

[11] Michalopoulos P. G., Ping Yi, D.E. Beskos and A.S. Lyrintzis (1991), " Continuum Modeling of Traffic Dynamics ", *Proc. of the 2nd Int. Conf. on Appl. of Advanced Tech. in Transportation Eng., Aug. 18-21, Minneapolis, Minnesota, pp. 36-40*

[12] Payne, H. J. (1971), "Models of freeway traffic and control", In: *Proceed. Math. of Publ. Syst. published by Simul. Council, Vol. 1, No. 1, pp. 51-61.*

[13] Ping Yi (1992), "Development and Field Testing of Advanced Continuum Traffic Flow Models for Congested Freeways", *Unpublished Doctoral Dissertation, University of Minnesota*

1-2 sites with weaving, 3-5 sites with Average Entry/Exit scheme. Lax $dt = 1sec$, Volume error $(veh/5min)$						
Sites	*Maximum*	*Max. Rel.*	*2 - Norm*	*Average*	*Average Rel.*	Std. Dev.
1	56.7	0.17	0.10	30.4	0.10	32.5
2	64.8	0.24	0.10	20.6	0.08	26.7
3	40.0	0.14	0.09	22.7	0.08	24.5
4	48.4	0.17	0.11	26.9	0.10	29.4
5	47.1	0.14	0.09	21.6	0.09	24.4

Table 2. Error statistics for Traffic flow

1-2 sites with weaving, 3-5 sites with Average Entry/Exit scheme. Lax $dt = 1sec$, Speed error $(mile/hour)$						
Sites	*Maximum*	*Max. Rel.*	*2 - Norm*	*Average*	*Average Rel.*	Std. Dev.
1	17.4	0.31	0.26	12.8	0.25	13.0
2	17.0	0.32	0.24	10.3	0.22	11.0
3	14.3	0.23	0.15	8.2	0.15	8.53
4	13.5	0.25	0.23	12.4	0.23	12.4
5	19.9	0.36	0.34	18.1	0.33	18.2

Table 3. Error statistics for Traffic speed

1-2 sites without weaving, 3-5 sites with Point Entry/Exit scheme. Lax $dt = 1sec$, Volume error $(veh/5min)$						
Sites	*Maximum*	*Max. Rel.*	*2 - Norm*	*Average*	*Average Rel.*	Std. Dev.
1	58.0	0.18	0.11	32.1	0.11	33.86
2	58.5	0.21	0.09	18.8	0.07	24.10
3	40.1	0.14	0.09	22.7	0.08	24.6
4	48.6	0.17	0.11	27.0	0.10	29.6
5	47.3	0.14	0.09	21.8	0.09	24.6

Table 4. Error statistics for Traffic flow

1-2 sites with weaving, Lax $dt = 1sec$,	3-5 sites with Point Entry/Exit scheme. Speed error $(mile/hour)$					
Sites	*Maximum*	*Max. Rel.*	*2 - Norm*	*Average*	*Average Rel.*	Std. Dev.
1	17.3	0.31	0.26	12.8	0.25	13.01
2	17.0	0.32	0.24	10.2	0.22	10.98
3	14.3	0.23	0.15	8.18	0.15	8.52
4	13.5	0.25	0.23	12.4	0.23	12.42
5	19.9	0.36	0.34	18.1	0.33	18.22

Table 5. Error statistics for Traffic speed

Processor	Parallel Execution time
0	21.34
1	22.65
2	22.65
3	22.77
4	22.61
5	22.69
6	22.65
7	22.77
8	22.66
9	22.75
10	22.69
11	22.88
12	22.76
13	22.75
14	22.83
15	21.25

Table 6. Computation times (in secs) on the NCUBE2 processors

Figure 6. I-494 speedup on NCUBE

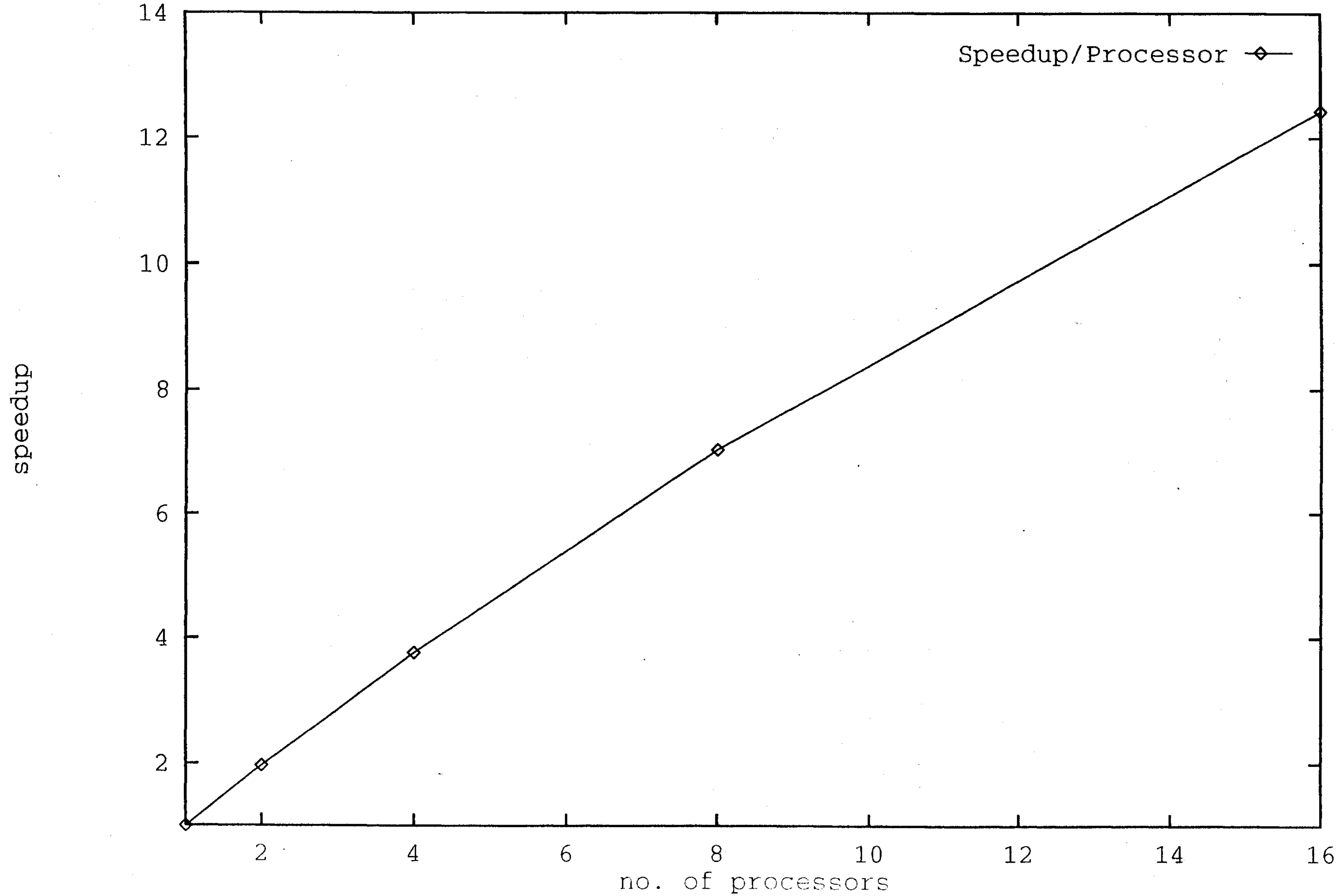
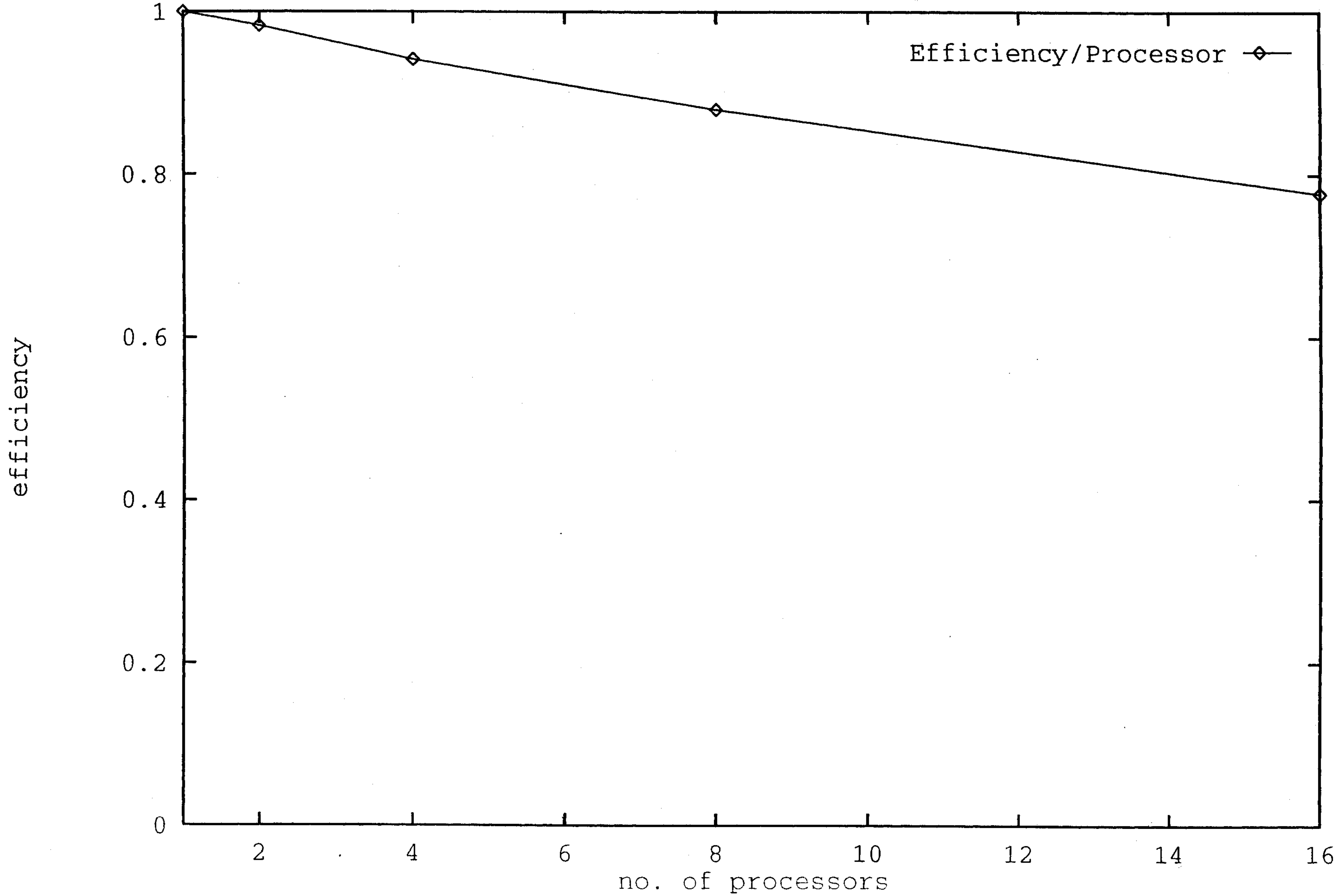


Figure 7. I-494 efficiency on NCUBE



CTS

Traffic Flow Simulation Through Parallel Processing

Final Research Report

September 26. 1991

Abstract

Explicit numerical methods for solving macroscopic traffic flow continuum models have been studied and efficiently implemented in traffic simulation codes. We studied and implemented implicit numerical methods for solving the flow conservation traffic model. We then wrote an experimental code in C simulating a freeway (un)congested pipeline and freeway entry/exit traffic flow. Tests with real data collected from the I-35 W freeway in Minneapolis were conducted on a workstation computer. The implicit methods gave the same (and in some cases better) accuracy as the Lax method. The implicit methods were (more than twice) faster than the Lax method. We also implemented the Lax method on a parallel machine and obtained significant execution time reduction.

Chronopoulos/Michalopoulos

1 Introduction

Macroscopic continuum traffic models flow based on traffic density, flow and velocity have been proposed and analyzed. Examples include Lighthill and Whitham's (1955) flow conservation and Payne's momentum conservation models [6], [11]. These models involve partial differential equations (PDEs) defined on appropriate domains with suitable boundary conditions which describe various traffic phenomena and road geometries.

The improvement of computational efficiency in the continuum traffic models has been the focal point in the development of traffic simulation programs. It is understood that the computer execution time to solve traffic flow problems depends not only on the size of the freeway and the complexity of roadway geometries, but also on the model equations and numerical schemes used in their discretization.

Explicit numerical methods (for example Lax, Upwind) have been used by Michalopoulos and Lin and Leo and Pretty to compute the solution of traffic flow continuum models [8], [5]. In these explicit schemes the space and time mesh sizes are restricted both by accuracy and numerical stability requirements. In order to reduce the computer execution time and maintain good accuracy, the total number of computations must be reduced. This can be achieved by using larger values of time and space mesh sizes. Implicit numerical methods provide the same accuracy as explicit methods and allow changes in the mesh sizes while maintaining numerical stability [3].

In this work we use implicit numerical methods (Backward Euler, Trapezoid) to solve more efficiently the flow conservation model. We wrote an experimental code in C simulating a freeway (un)congested pipeline and freeway entry/exit traffic flow. Tests with real data collected from the I-35 W freeway in Minneapolis were conducted. These data have been collected by the Department of Civil Engineering at the University of Minnesota and the Minnesota Department of Transportation. Using these data we tested (for accuracy and efficiency) the implicit methods against the Lax method on a Sun Sparc1 workstation computer. The implicit methods yielded the same (or better accuracy) as the Lax method and they were (more than twice) faster than the Lax method. We have also implemented efficiently the Lax method on the (64 processor) NCUBE/7 parallel computer located at the Department of Computer. Each processor of the NCUBE/7 is as powerful as a SUN 3/50 workstation. On the NCUBE/7, the parallel Lax method on the 64 processors run 25 times faster than on the one processor.

The outline of this article is as follows. In section 2, we review the flow conservation continuum traffic model. In section 3, we review the Euler implicit, Trapezoidal and Lax methods. In section 4, we describe the various theoretical and empirical curves relating the traffic flow and density. In section 5, we describe the congested/uncongested and entry/exit freeway models. In section 6, we describe a parallel implementation of the Lax method. In section 7, we present the numerical results. Section 8 contains concluding remarks.

2 A Continuum Model of Traffic Flow

The following conservation equation has been proposed by Lighthill and Whitham (1995) [6] as a continuum traffic model:

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = g(x,t). \tag{1}$$

where $k(x,t)$ and $q(x,t)$ are the traffic density and flow respectively at the space-time point (x,t). The traffic flow, density and speed are related by the equation:

$$q = ku, \tag{2}$$

where the equilibrium speed $u(x,t) = u(k)$ must be provided by a theoretical or empirical u-k model. For the Greenshields u-k model

$$u(k) = u_f(1 - k/k_0), \tag{3}$$

where u_f is the free flow speed and k_0 the jam density [2]. The generation term $g(x,t)$ represents the number of cars entering or leaving the traffic flow in a freeway with entries/exits.

3 Numerical Methods

We consider one explicit method (Lax) and two implicit methods (Euler implicit and Trapezoidal) which are used in computational fluid dynamics [3].

For each traffic model the road section (the space dimension) is discretized using uniform mesh for all numerical methods; only the time stepsizes differ between methods. We use the following notation:

Δt = time stepsize.

Δx = space stepsize.

k_j^n = density (vehicles/mile/lane) at space node $j\Delta x$ and at time $n\Delta t$.

q_j^n = flow (vehicles/hour/lane) at space node $j\Delta x$ and at time $n\Delta t$.

3.1 Lax Method

The Lax method is an explicit method. The new density value k_j^{n+1} is computed directly from the density and flow at the preceding time step n:

$$k_j^{n+1} = \frac{k_{j+1}^n + k_{j-1}^n}{2} - \frac{\Delta t}{\Delta x}\frac{q_{j+1}^n - q_{j-1}^n}{2} + \frac{\Delta t}{2}(g_{j+1}^n - g_{j-1}^n). \tag{4}$$

The method is of first order accuracy with respect to Δt, i.e. the error is $O(\Delta t)$. To maintain numerical stability time and space stepsizes must satisfy the CFL condition $\frac{\Delta x}{\Delta t} > u_f$, where u_f is the free flow speed. For example in the KRONOS traffic simulation code (using Lax) $\Delta x = 100 feet$ and $\Delta t = 1 sec$ are recommended.

3.2 Euler Implicit

The Euler implicit method applied to the nonlinear PDE (1) generates a nonlinear recursion involving all space nodes at each time step. To solve numerically this recursion Beam and Warming have suggested using one Newton linearization steps [3]. Each Newton step constructs a tridiagonal linear system with unknowns $\Delta k_j = k_j^{n+1} - k_j^n$.

$$-\frac{\Delta t}{2\Delta x}\left(\frac{dq}{dk}\right)_{j-1}^n \Delta k_{j-1} + \Delta k_j + \frac{\Delta t}{2\Delta x}\left(\frac{dq}{dk}\right)_{j+1}^n \Delta k_{j+1} =$$
$$-\frac{\Delta t}{2\Delta x}\left(q_{j-1}^n - q_{j-1}^n\right) + \frac{\Delta t}{2}(g_{j+1}^n - g_{j-1}^n).$$

This tridiagonal linear system is solved by a variant of the Gaussian elimination called the Thomas algorithm. The solution is then advanced to the next time step simultaneously at all space nodes by computing $k_j^{n+1} = k_j^n + \Delta k_j$. This method is of first order accuracy with respect to Δt and it is unconditionally stable.

Artificial smoothing is often added to reduce oscillatory behavior in the numerical solution. This is achieved by adding a fourth order damping term d_j to each term k_j

$$d_j = -\frac{\omega}{8}\left(k_{j-2} - 4k_{j-1} + 6k_j - 4k_{j+1} + k_{j+2}\right)$$

We have tested several damping coefficients from $\omega = 0$ (no damping) to $\omega = 1$. The choice $\omega = 1$ gave the best results.

3.3 Implicit Trapezoidal Method

The Trapezoidal method is identical to the Euler implicit method except for the constants used in the tridiagonal linear system equations.

$$-\frac{\Delta t}{4\Delta x}\left(\frac{dq}{dk}\right)^n_{j-1}\Delta k_{j-1} + \Delta k_j + \frac{\Delta t}{4\Delta x}\left(\frac{dq}{dk}\right)^n_{j+1}\Delta k_{j+1} = -\frac{\Delta t}{2\Delta x}\left(q_{j+1}^n - q_{j-1}^n\right) + \frac{\Delta t}{2}\left(g_{j+1}^n - g_{j-1}^n\right)$$

It is of second order accuracy with respect to Δt and unconditionally stable. However, for discontinuous problems Euler Implicit may yield more accurate results. As with the Euler method, explicit damping is added at each time step. We note that the Trapezoidal and the Euler implicit methods require also the computation of the Jacobian dq/dk. It is clear that these methods involve more computations per time step than Lax. However, they allow much larger stepsizes which may make them overall faster than Lax.

4 Flow Rate-Flow Density (q-k) Models

A u-k model (see (2) and (3) above) is an indispensable part of the flow conservation model. This relation can be used to express the flow rate as a function of the flow density i.e. $q = q(k)$. This function is a nonlinear function which must satisfy some general requirements. The equations that define the q-k curve are used in the programs to convert from density to flow and from flow to density.

These general requirements on the q-k curve can be derived from the following observations on traffic flow modeling [7].

- For uncongested flow an increase in density corresponds to an increase in flow. up to a critical density k_c, where the flow becomes congested.
- Maximum flow occurs at the critical density: $q_{max} = q(k_c)$.
- For congested flow an increase in density corresponds to a decrease in flow. up to the jam density k_{jam}, where flow stops.

A q-k relation model must also be adapted to characteristics of the freeway section which it represents. Theoretical q-k models can not be adapted to the special roadway characteristics and so such a model function must be constructed from empirical data. Greenshields q-k curve is derived from equations (2) and (3) and appropriate choices for the free flow speed u_f and jam density $k_{jam} = k_0$. In our applications we chose u_f =(60 miles/hour) and k_0 =(180 vehicles/mile). The Greenshields curve has the basic features described above but can not be tuned to local characteristics of a freeway section. However, we used Greenshield's q-k curve in the initial development of the programs and as a baseline for comparisons.

4.1 Experimental q-k Models

Field data for constructing the q-k model were collected in I-35 W in Minneapolis. With these discrete data a q-k curve was derived [9],[10]. Such a curve has the following parameter ranges:

- The critical density k_c should be about 40 to 45 vehicles/mile/lane.
- The maximum flow q_{max} should be less than 2500 to 2700 vehicles/hour/lane.
- The slope of the curve at $k = 0$, which represents the free-flow speed u_f, should be approximately 65 to 75 miles/hour.

We have used several curve fitting methods to construct continuous q-k curves from the set of (k, q) discrete data points available. Our objective was to find a general method that produces a curve which is based on the discrete data. has the basic features of a q-k curve, has the parameter ranges (described above), and also works well in the numerical methods for solving (1). We used three different methods **piecewise linear, cubic spline,** and **least squares** to approximate q-k curves from field data.

The simplest method consists of connecting the q-k data points with straight line segments, yielding a **piecewise linear** q-k curve. This is a continuous curve that passes through all data points but the slope of the curve (which is used in the implicit methods) is discontinuous at the line segments intersections.

In an effort to find a curve that interpolated all of the q-k data points and that also had a continuous first derivative, we constructed a **cubic spline.** The cubic spline is a collection of third-degree polynomials, one polynomial for each interval between q-k data points. We tested both clamped (slope at endpoints is specified) and natural (slope at endpoints is unspecified) splines and found that for our field data set the splines were nearly identical. All cubic spline programs used the natural cubic spline.

Finally, several **least squares** approximations were tried. In this method the data points (k_i, q_i) are used to construct a rectangular matrix with row i composed of powers of k_i and a right-hand-side vector containing the q_i. Then the matrix is reduced using the singular value decomposition method (SVD) available in the LINPACK package or the Matlab package [1]. The reduced matrix is then used to find the coefficients of the curve that minimizes the total squared error between the data points and the curve. This method will produce curves of any degree up to the number of data points. Quadratic. cubic and quartic least-squares polynomial curves were found using the Matlab's (SVD). The quartic curve

$$q = -1.7156 \times 10^{-5} k^4 + 7.1802 \times 10^{-3} k^3 - 1.2514k^2 + 94.8463k - 69.1588$$

appeared to be the lowest-degree least-squares approximation to the discrete data that satisfies the q-k curve criteria. This quartic polynomial must be evaluated at each node for each time step so it is important to use a polynomial of the least degree.

The choice of q-k curve was found to have a large effect on the stepsize selection of the implicit methods. Implicit methods using the smooth q-k

curves generated by the least squares and Greenshield's methods were able to use larger time steps than the programs using cubic spline or piecewise linear curves.

5 Freeway Traffic Models

Three models of freeway traffic flow were used to test the numerical methods described above. Each model consists of a section of a single-lane freeway. The data sets used with each model give the number of vehicles counted crossing each boundary (all lanes) during each 5-minute interval. In all models these numbers, vehicles/5 minutes, are then multiplied by 12 and divided by the number of lanes to yield an average single-lane flow rate in vehicles/hour/lane. The field data are collected as follows: A detector is placed at a *check station* which counts the volume of cars passing that road point every 5 minutes. Check stations are set at the upstream and downstream boundaries and at one more locations in the freeway stretch in between. These measurements provide a flow-time function. This function has the form of a step function. The flow at the boundaries is used to set up the boundary conditions of the PDE (1). The flow at intermediate points is used to compute the deviation of the computed model solution from the field data.

In our modeling we distributed linearly the flow within the 5 minute intervals. The resulting boundary flow-time function is piecewise linear. In the programs boundary flow rates are converted to density (vehicles/mile/lane) using a q-k relation. This gives boundary density-time functions which are piecewise linear. This is implemented as follows. At each time step the boundary densities must be assigned to the boundary nodes (upstream and downstream), and then the boundary flow is determined from the density. In these programs the boundary density at each time step is found by linear interpolation between the known density values at $t = 5$ min, 10 min, etc. Then boundary flow is found at the point on the q-k curve corresponding to the boundary density.

5.1 Uncongested Pipeline

The uncongested pipeline freeway traffic model consisted of a 4000-ft segment of a 2-lane freeway. In this model the road segment contains no entry or exit ramps and the traffic flow is always uncongested. This was the Minneapolis I-35 W northbound between the 76th and the 70th streets with a check station located at the 73rd street. The field data table 2 contains traffic flow measurements (vehicles/5 minutes/2 lanes) made at the upstream and downstream boundaries and at a check station point 2000 ft from the upstream boundary. Observations were recorded at 5-minute intervals over a span of 2 hours. These field data have also been used in [9],[10].

5.2 Congested Pipeline

The congested pipeline model allows both congested and uncongested flow in a road segment without entry or exit ramps. The data used in this model was taken from observations of 3600-ft segment of 4-lane freeway. This was the Minneapolis I-35 W southbound between the 26th and the 31th streets with a check station at the 28th street. Traffic flow measurements (vehicles/5 minutes/4 lanes) were made at the upstream and downstream boundaries and at a check station point 1600 ft from the upstream boundary. Along with the flow measurements, the state of congestion (uncongested or congested) at the boundaries was also recorded. Observations were recorded at 5-minute intervals over a span of 2 hours and 40 minutes. These field data have also been used in [9],[10].

The field data table 3 for the congested model represents a road section that changes from uncongested to congested flow and remains congested for approximately 2 hours, then changes from congested to uncongested flow. For each numerical method that we tested the largest error occurred in the second *congestion-change* interval, where flow changes from congested to uncongested. The next largest error occurred at the first congestion-change interval.

In the implicit methods programs we used a *large* time step in the intervals where congestion remains constant and we used a *small* time step in congestion-change intervals, to minimize the error in those intervals. In addition to the decrease in the time step we used repeated iterations (extra Newton steps) in the congestion-change intervals. The implicit methods yielded considerably smaller maximum errors after these improvements were made to the codes.

5.3 Entry/Exit

The entry/exit freeway is a section of I 35-W northbound in Minneapolis. The upstream/downstream boundaries were set at the location of the 55th street and the 46th street, respectively. The uncongested entry/exit model consists of a 6400-ft section of 3-lane freeway with one entrance ramp and one exit ramp. The entrance ramp is located 1400 ft below the upstream boundary and the exit is located 5600 ft below the upstream boundary. The first check station is located 2000 ft below the upstream boundary; the second check station is located 3800 ft below the upstream boundary.

Data were collected by the Department of Civil Engineering and the Minnesota Department of Transportation on November 8, 1989. The data consists of flow measurements (vehicles/5 minutes/3 lanes) made at the boundaries and at the check stations and ramp flow measurements (in vehicles/5 minutes) made at each ramp. Observations were recorded at 5-minute intervals during one morning from 6:05am to 9:30am. The initial conditions were not specified so we have assumed the initial flow to be the average of the flow values at the check stations at 6:05am, and constant along the road section. For this report, only the measurements made at the first check station were used in comparisons with the numerical program output.

In this simplified model the flow is assumed to be uncongested at all times. Merging flow from the entry ramp is added to the flow at the first node downstream of the ramp. If merging flow plus mainstream flow exceeds the maximum flow q_{max}, the flow value at the entry node is set to q_{max} and any excess merging flow is not used in the calculations. Exiting flow is subtracted from the first node downstream of the exit ramp.

6 Parallel Lax

We have implemented the Lax method on the NCUBE/7 parallel computer at the Department of Computer Science of the University of Minnesota. The NCUBE has 64 processors connected in a hypercube network (see fig. 13) and a host (Sun 3/50) computer for interaction with the user. The number of (N) processors to be active is chosen by the user, but must be a power of 2. The host computer allocates N processors arranged in an m-dimensional hypercube, where $m = \log_2 N$. Each of the N processors is directly connected

Operation	Time	Comm/Comp
8 Byte transfer	470 μ sec	-
8 Byte Add	11.2 μsec	42 times
8 Byte Multiply	14.7 μsec	32 times

Table 1: Computation and Communication times on the NCUBE/7

to m other processors. In table 1 we show a summary of inter processor communication times and basic floating point operation times [4]. We see that communication even between neighbor processors is several times slower than floating point operations. Programs run most efficiently when inter processor communication is minimized and when all communication occurs between neighbor processors.

In the parallel implementation of the Lax method we partition a freeway section into N (equal) segments and assign each segment to one processor, for $N = 1, 2, 4, 8, 16, 32$, and 64. At each time step the values for density at the segment boundary points must be exchanged between adjacent segments. Adjacent road segments are assigned to adjacent processors, so all data exchanges occur between directly connected processors. The way these simultaneous data exchanges between adjacent processors are carried out is shown in figure 14. At each time step, after all segment interior values have been determined, two exchange steps must be performed. Each step exchanges the segment boundary values for half of the segment boundaries. Road boundary conditions are handled by the processors assigned the upstream and downstream segments. When the computations are completed the processors send their output back to the host, where the output data is arranged in the proper order and printed.

We have tested this implementation with an arbitrary set of upstream boundary densities. We used $\Delta t = 1 sec$, total time = 3600 sec, and $\Delta x =$ 200 ft. In order to test the program on a large number of processors we used a road model with 512 space nodes, for a total length of 102,400 ft..

7 Results

Table 2 contains the data for the empirical q-k curve. These data were used in constructing the piecewise linear, cubic spline and least squares approximations shown in figure 1. Tables 3,4,5 contain the field data for the uncongested/congested and entry/exit freeway traffic flow tests.

For the tests the time stepsize selection was made as follows. For the Lax method we set $\Delta t = 1 sec$. This is required in order to maintain numerical stability. For the implicit methods we increased the time stepsize subject to the restriction that the maximum error does not exceed that of the Lax method. For the uncongested and entry/exit flow cases a single time stepsize was selected. For the congested flow case two different time stepsizes were used. One *small* stepsize was used in the 5 min time intervals of change from congested to uncongested (or vice versa) and another *large* stepsize was in other time intervals.

The tests for selecting the best empirical q-k curve pointed to the quartic least squares approximation. This allows the largest stepsize combinations in the implicit methods yielding the smallest maximum error. The results for the congested case are contained in table 6. The largest of the stepsizes was found to give the smallest maximum error in the uncongested and entry/exit flow cases. Table 7 shows the results for accuracy and execution time obtained on the Sun SPARCstation 1, using a time step size $\Delta t = 15$ sec for the implicit methods programs on the uncongested field data. Table 8 shows the results obtained on the same machine using congested field data and $\Delta t = 15$ sec during regular intervals and $\Delta t = 3$ sec during congestion-change intervals. Table 9 shows the results obtained on the same machine using the entry/exit field data and $\Delta t = 15$ sec.

The best performance in accuracy and execution time was obtained with the Euler implicit method using three (Newton) iterations per time step in congestion-change intervals. This method showed a large improvement over the Lax method in both error and time required. In the uncongested and entry exit cases (tables 7 and 9) the maximum errors in all three methods are of the same magnitude. In the congested case (table 8) the maximum error produced by the Euler method was about one fourth of the maximum error produced by the Lax method. The implicit methods are more than twice as fast the Lax method (tables 7,8,9 last column) for the congested case and more than three times in the uncongested and entry/exit cases.

The 3-D figures show the Lax, Euler and Trapezoid solution using the empirical quartic least squares approximation. These solutions plots look very close to each other in all cases except the congested flow case. The high oscillations in the congested/uncongested change intervals appear in the Lax method more than the Euler method.

The parallel implementation of the Lax method on the 64 processor NCUBE/7 for a pipeline (artificially lengthened) freeway section shows an execution time speedup of 25 versus the single processor execution time.

8 Conclusions

We have studied the use of implicit numerical methods solve the flow conservation continuum model. We have written an experimental code in C simulating a freeway (un)congested pipeline and freeway entry/exit traffic flow. Tests with real data collected from the I-35 W freeway in Minneapolis were conducted on a workstation computer. Our tests show that the implicit methods are more efficient than the Lax method and provide the same or better accuracy. This could increase if iterative methods are used instead of the Gaussian elimination in solving the tridiagonal linear systems required by the implicit methods. We have also implemented efficiently the Lax method on a (64 processor) NCUBE/7 parallel computer. The single processor of the NCUBE/7 is as powerful as a workstation processor. The parallel Lax method was 25 times faster than the single processor one. This could increase substantially if larger freeway sections (involving entry/exits) are simulated. This implementation shows there is a lot of potential in applying parallel processing to carry out real time traffic simulations.

References

[1] Dongarra J.J. et al.(1979), "LINPACK user's guide. " *SIAM publications.*

[2] Greenshields, B. D. (1934), "A study of traffic capacity ", In: *Proc. Highway Res. Board, Vol. 14, pp. 448-477.*

[3] Hirsch C. (1988), "Numerical Computation of Internal and External Flows ", *Vol.2, John Wiley and Sons.*

[4] Kim S. K., A. T. Chronopoulos (1991), "A Class of Lanczos-like algorithms implemented on parallel computers", To appear in the Journal: *Parallel Computing.*

[5] C. J. Leo and R. L. Pretty (1990), "Some Comments and Numerical Tests on Upwind Finite Difference Schemes for Macroscopic Traffic Modeling", *Dept. of Civil Eng., Univ. of Queensland, Australia.*

[6] Lighthill, M. H., G. B. Witham (1955), "On Kinematic waves: II A theory of traffic flow on crowded roads", In: *Proceed. R. Soc. Ser. A 229, No. 1178, pp. 448-447.*

[7] McShane W. and Roess R. (1990), "Traffic Engineering ", *Prentice Hall.*

[8] Michalopoulos P. G., J. Lin (1985), " A freeway simulation program for microcomputers ", In: *Proceed., 1st National Conf. on Microcomputers in Urban Transp., ASCE. California, pp. 330-341.*

[9] Michalopoulos P. G., Eil Kwon and J. Kang (1991), " Enhancements and field testing of a dynamic freeway simulation program. " To Appear in the Journal: *Transportation Research Records.*

[10] Michalopoulos P. G., Ping Yi, D.E. Beskos and A.S. Lyrintzis (1991), " Continuum Modeling of Traffic Dynamics ", To Appear in: *Proc. of the 2nd Int. Conf. on Appl. of Advanced Tech. in Transportation Eng., Aug. 18-21, Minneapolis, Minnesota.*

[11] Payne, H. J. (1971), "Models of freeway traffic and control", In: *Proceed. Math. of Publ. Syst. published by Simul. Council, Vol. 1, No. 1, pp. 51-61.*

k = density (veh/mile/lane) q = flow (veh/hour/lane)			
k	q	k	q
0	0	66	2376
10	650	76	2432
20	1260	98	2352
30	1860	124	2232
32	1952	150	1500
35	2100	175	525
36	2124	186	0

Table 2: Empirical q-k Data

Initial condition: 271.67 veh/5 min/2 lanes uncongested flow at all nodes q_u = upstream flow. q_d = downstream flow. q_c = flow at check station							
Time (min)	Flow (veh/5 min/2 lanes)			Time (min)	Flow (veh/5 min/2 lanes)		
	q_u	q_d	q_c		q_u	q_d	q_c
5	272	277	270	65	275	267	277
10	285	284	288	70	283	289	285
15	287	283	279	75	253	261	255
20	301	293	296	80	262	259	260
25	285	286	284	85	245	240	237
30	318	310	315	90	259	257	260
35	321	319	318	95	294	287	291
40	344	356	348	100	232	241	236
45	322	322	330	105	227	234	229
50	300	303	295	110	259	247	250
55	329	319	328	115	272	273	279
60	313	327	317	120	249	240	243

Table 3: Uncongested Flow Field Data

Initial condition: 575 veh/5 min/4 lanes uncongested flow at all nodes q_u = upstream flow, q_d = downstream flow, q_c = flow at check station u = uncongested flow, c = congested flow												
Time (min)	Flow (veh/5 min/4 lanes)					Time (min)	Flow (veh/5 min/4 lanes)					
	q_u		q_d		q_c		q_u		q_d		q_c
5	579	u	572	u	564	85	404	u	484	u	437
10	580	u	547	c	579	90	465	u	492	c	467
15	574	c	562	c	570	95	465	c	485	c	473
20	549	c	558	c	541	100	460	c	482	c	463
25	576	c	550	c	561	105	440	c	454	c	437
30	574	c	489	c	529	110	469	c	455	c	475
35	483	c	503	c	472	115	523	c	519	c	536
40	545	c	563	c	555	120	527	c	519	c	497
45	548	c	554	c	542	125	513	c	540	c	532
50	545	c	503	c	527	130	555	c	551	c	559
55	512	c	507	c	510	135	522	c	526	c	516
60	501	c	506	c	500	140	503	c	537	c	515
65	531	c	502	c	523	145	489	c	495	c	493
70	492	c	511	c	496	150	430	c	441	c	431
75	508	c	470	c	497	155	447	c	434	c	432
80	487	c	422	c	436	160	444	c	453	c	453

Table 4: Congested Flow Field Data

q_u = upstream flow, q_m = entry flow, q_x = exit flow

q_c = flow at check point, q_d = downstream flow

All flow is uncongested. q_m and q_x are given in veh/5 min/ramp

All mainstream flow is given in veh/5 min/3 lanes

Time	Flow (veh/5 min)					Time	Flow (veh/5 min)				
	q_u	q_m	q_c	q_x	q_d		q_u	q_m	q_c	q_x	q_d
6:05	219	7	205	6	208	7:50	509	26	556	6	566
6:10	253	12	255	5	267	7:55	526	30	546	6	593
6:15	302	11	294	6	275	8:00	535	25	536	9	578
6:20	348	18	355	4	365	8:05	507	25	518	13	535
6:25	378	26	388	7	390	8:10	507	27	494	9	525
6:30	473	17	466	4	483	8:15	513	24	530	8	528
6:35	536	24	568	13	532	8:20	550	33	551	6	581
6:40	588	23	565	8	607	8:25	552	26	574	11	584
6:45	603	20	598	9	583	8:30	528	23	530	22	556
6:50	684	22	583	5	624	8:35	545	28	550	12	554
6:55	599	25	621	12	612	8:40	557	38	558	17	600
7:00	577	24	566	13	600	8:45	530	44	537	15	572
7:05	581	26	601	7	598	8:50	573	40	580	14	598
7:10	612	27	605	6	621	8:55	538	40	540	22	575
7:15	588	26	611	6	625	9:00	461	38	467	15	549
7:20	588	28	577	5	611	9:05	432	33	433	24	427
7:25	606	26	605	12	609	9:10	428	31	440	13	488
7:30	584	23	593	4	637	9:15	425	17	446	17	408
7:35	556	29	563	4	574	9:20	427	27	416	10	470
7:40	599	24	591	5	604	9:25	478	46	496	15	479
7:45	534	21	526	3	553	9:30	417	28	423	19	474

Table 5: Entry/Exit Field Data

Comparison of numerical methods using different q-k curves Lax method: dt = 1 s for all curves				
	Max Error (veh/5 min/4 lanes)			
	Greenshield	Linear	Cubic Spline	Least Squares
Method	dt = 15 : 3 s	dt = 6 : 3 s	dt = 6 : 3 s	dt = 15 : 3 s
Lax	205.86	261.98	278.56	273.56
Euler	45.35	302.12	317.73	77.32
Trapezoid	40.62	300.93	318.52	106.77

Table 6: q-k Curve Comparison Results for Congested Flow

Quartic Least Squares q-k Curve dt = 15 s Time = exec time on SPARCstation 1			
	Error (veh/5 min/2 lanes)		Time
Method	maximum	average	(s)
Lax	9.61	3.93	2.6
Euler	9.84	4.01	0.6
Trapezoid	9.83	4.03	0.6

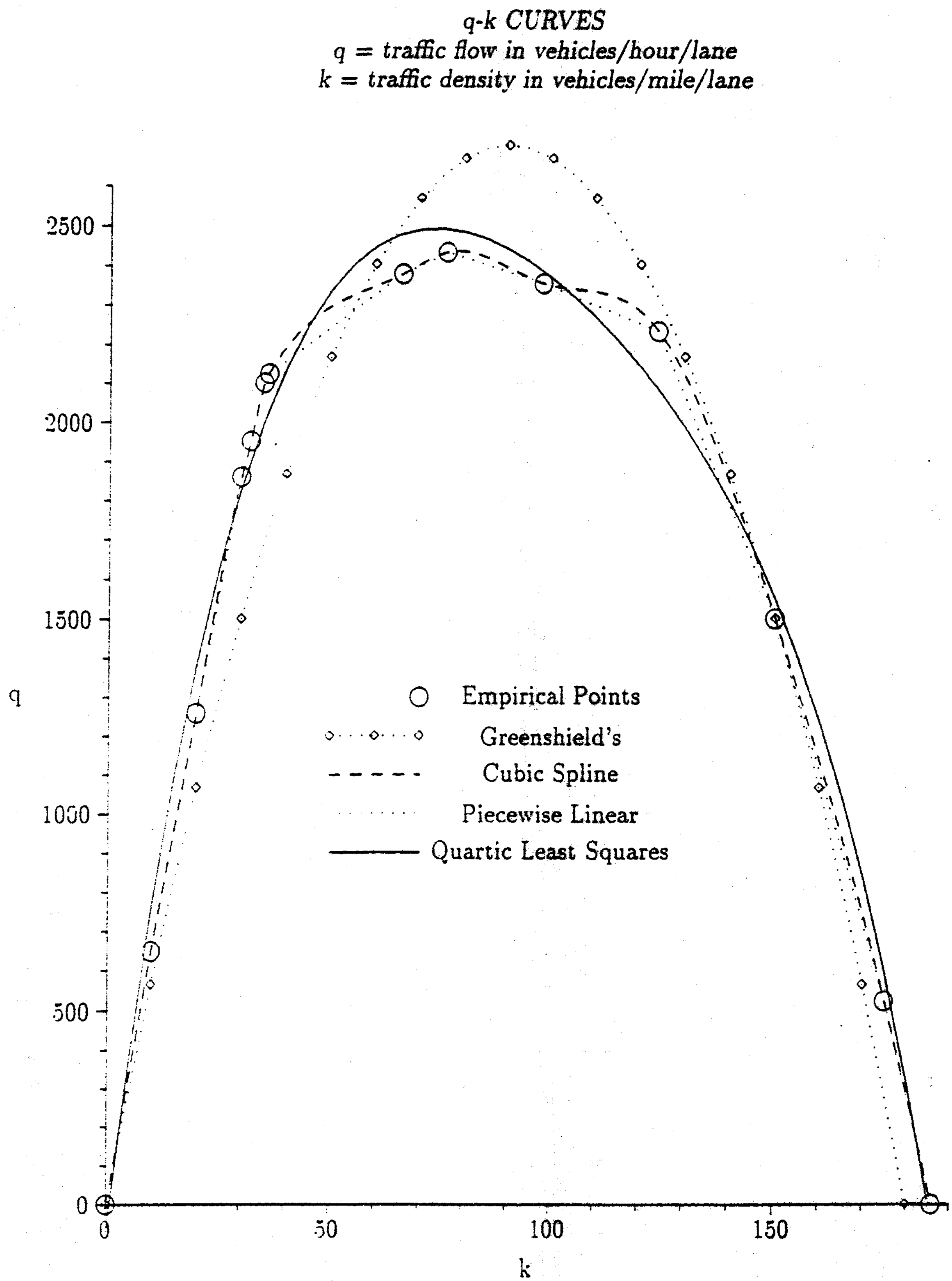
Table 7: Uncongested Flow Results

Quartic Least Squares q-k Curve dt = 15 : 3 s Time = exec time on SPARCstation 1			
	Error (veh/5 min/4 lanes)		Time
Method	maximum	average	(s)
Lax	273.56	24.99	3.2
Euler	77.32	17.33	1.6
Trapezoid	106.77	20.88	1.6

Table 8: Congested Flow Results

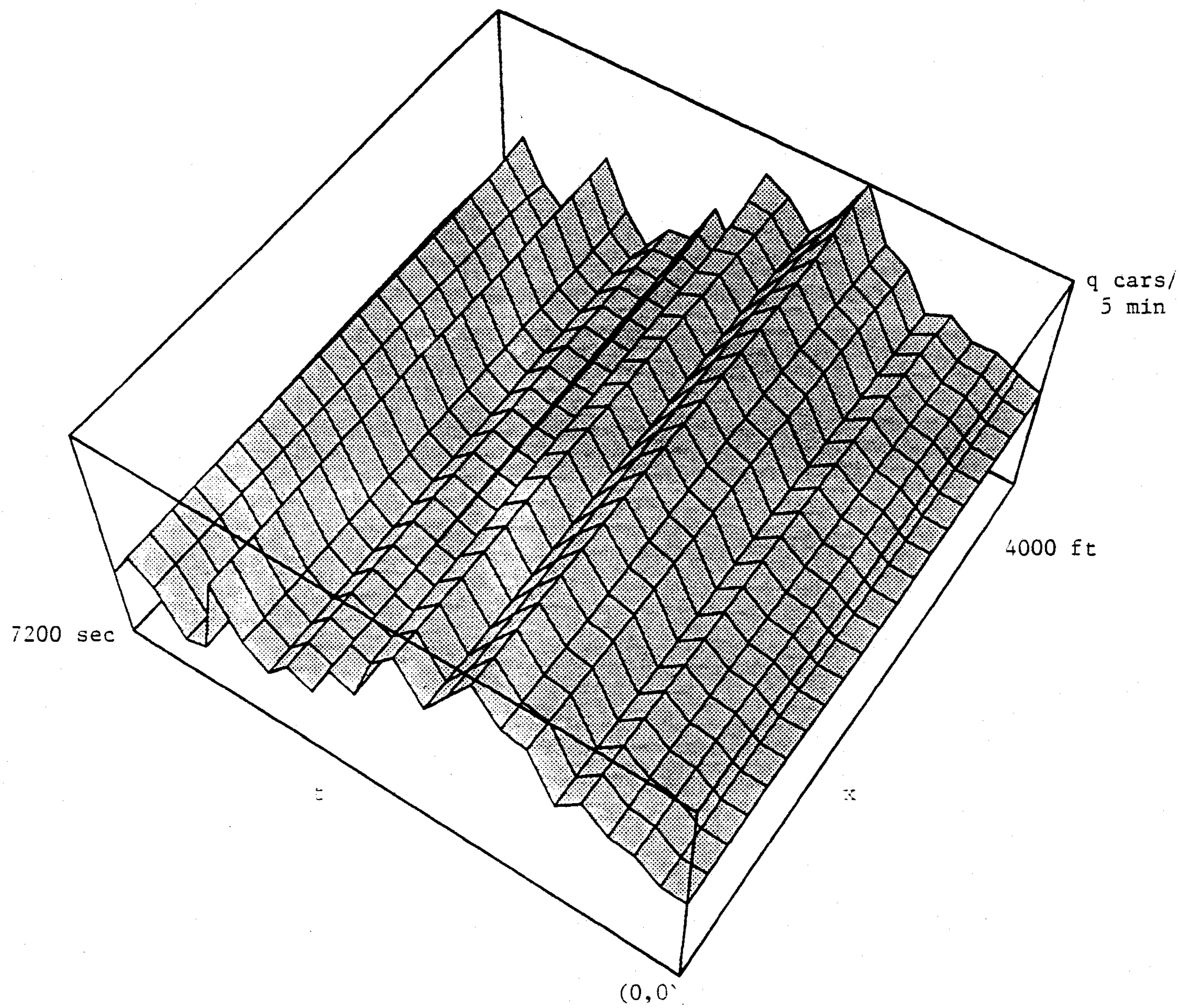
Quartic Least Squares q-k Curve dt = 15 s Time = exec time on SPARCstation 1			
	Error (veh/5 min/3 lanes)		Time
Method	maximum	average	(s)
Lax	42.66	11.58	7.2
Euler	42.51	11.38	1.6
Trapezoid	42.52	11.50	1.6

Table 9: Entry/Exit Flow Results



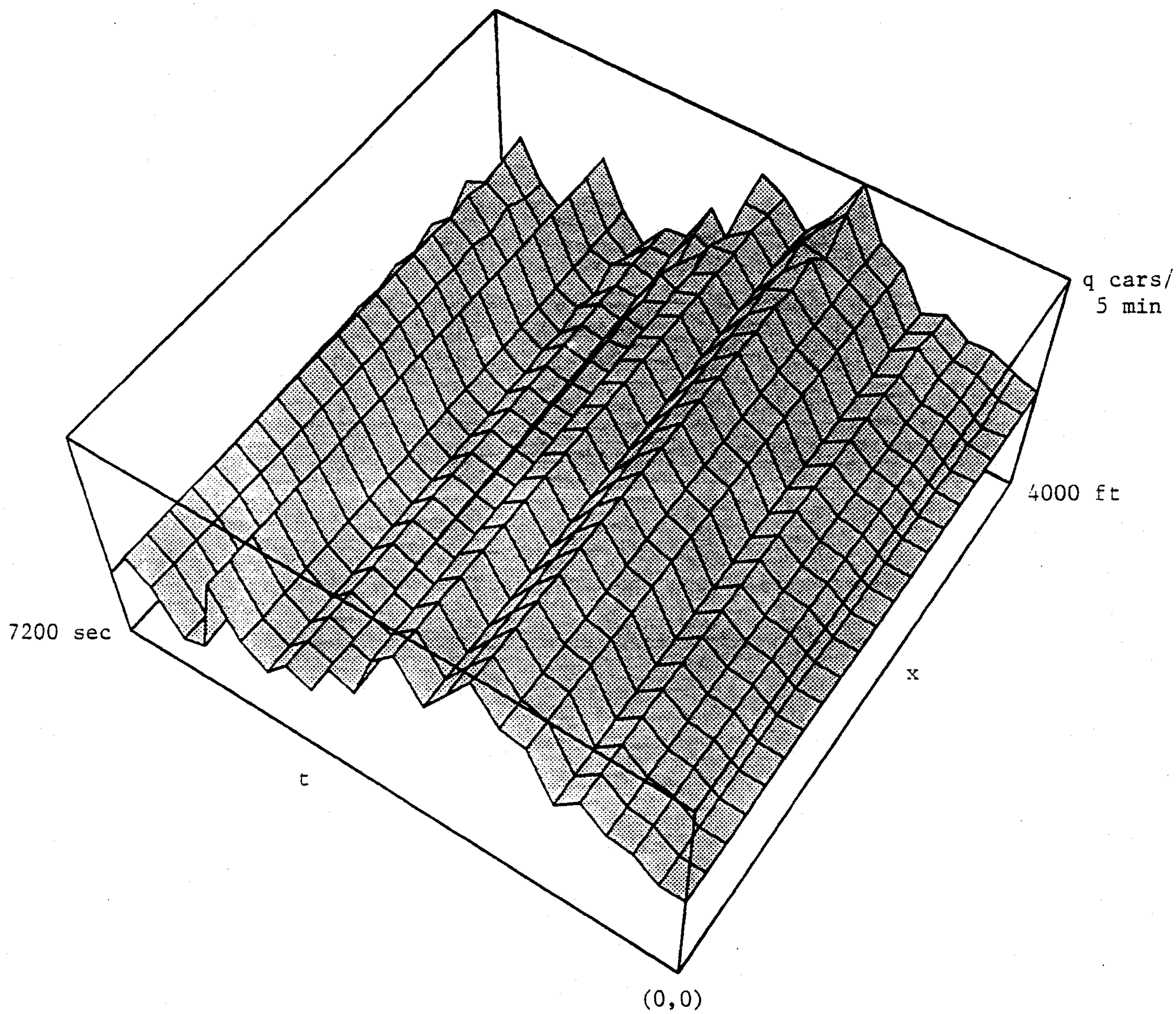
q-k CURVES
q = traffic flow in vehicles/hour/lane
k = traffic density in vehicles/mile/lane
2500
2000
1500
1000
500
0
q
0
50
100
150
k
Empirical Points
Greenshield's
Cubic Spline
Piecewise Linear
Quartic Least Squares

Fig. 1



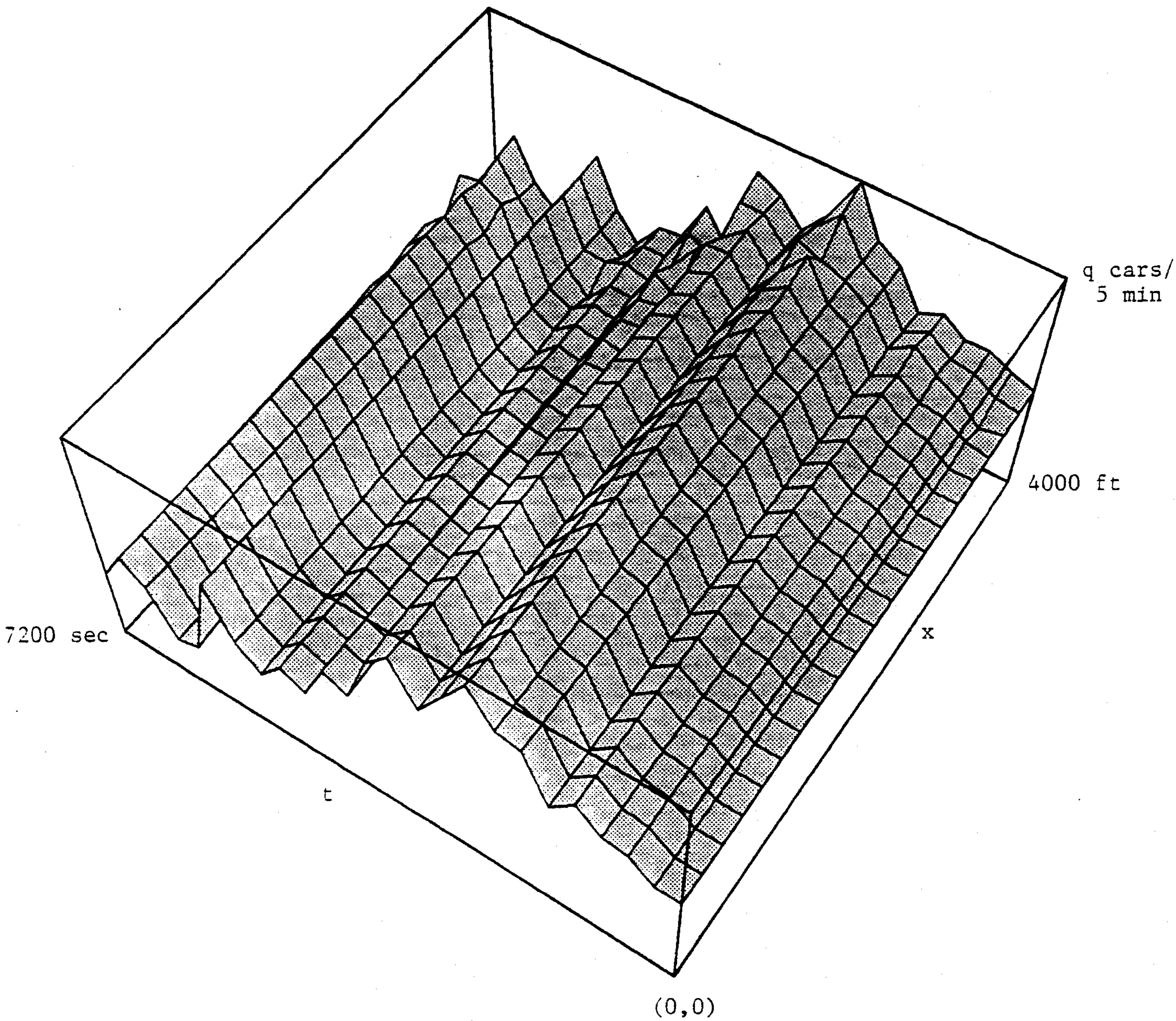
Lax Method
Uncongested Pipeline Flow (veh/5 min/2 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 1 s

Fig. 2



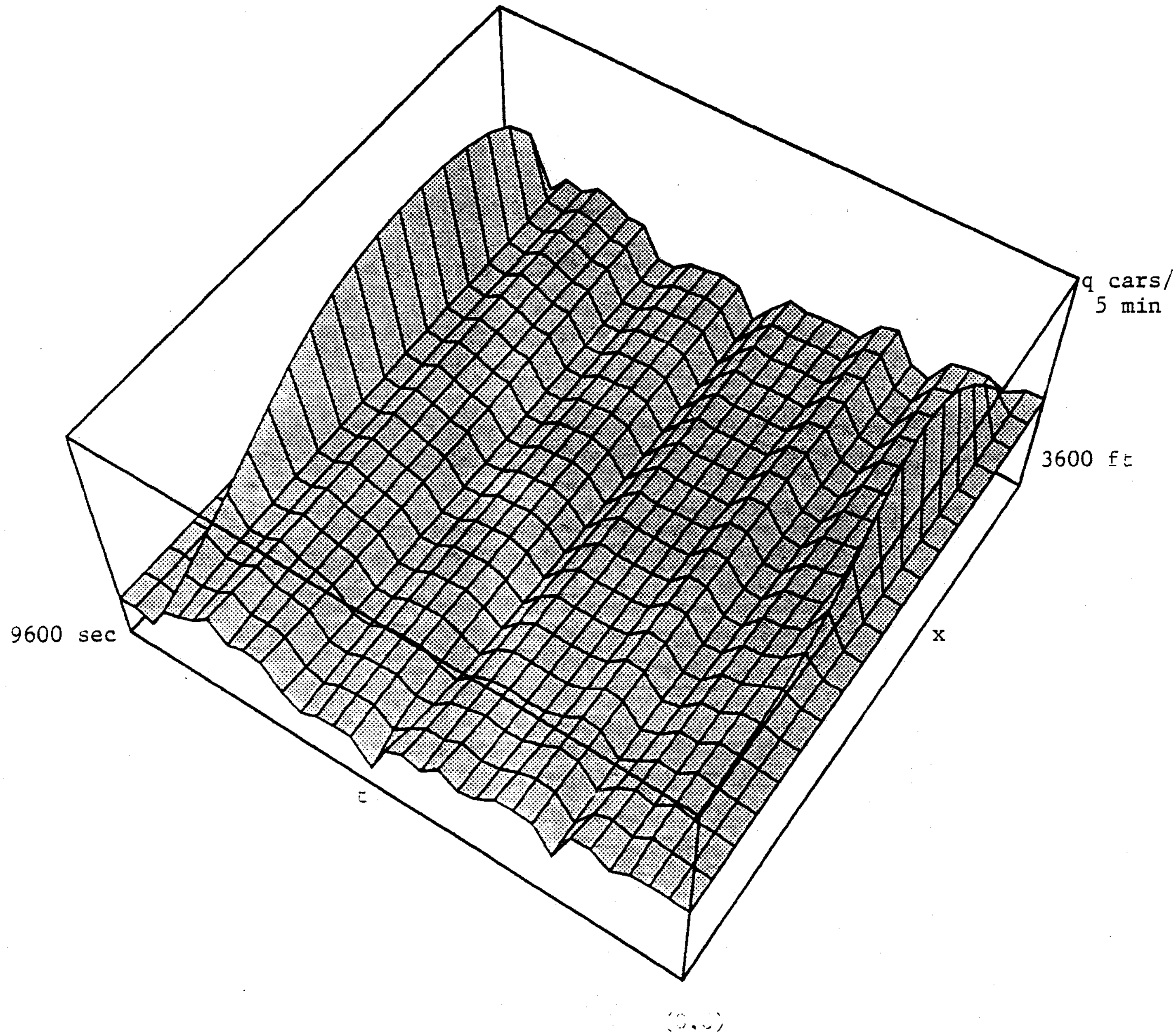
Implicit Euler Method
Uncongested Pipeline Flow (veh/5 min/2 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 15 s
omega = 1.0

Fig. 3



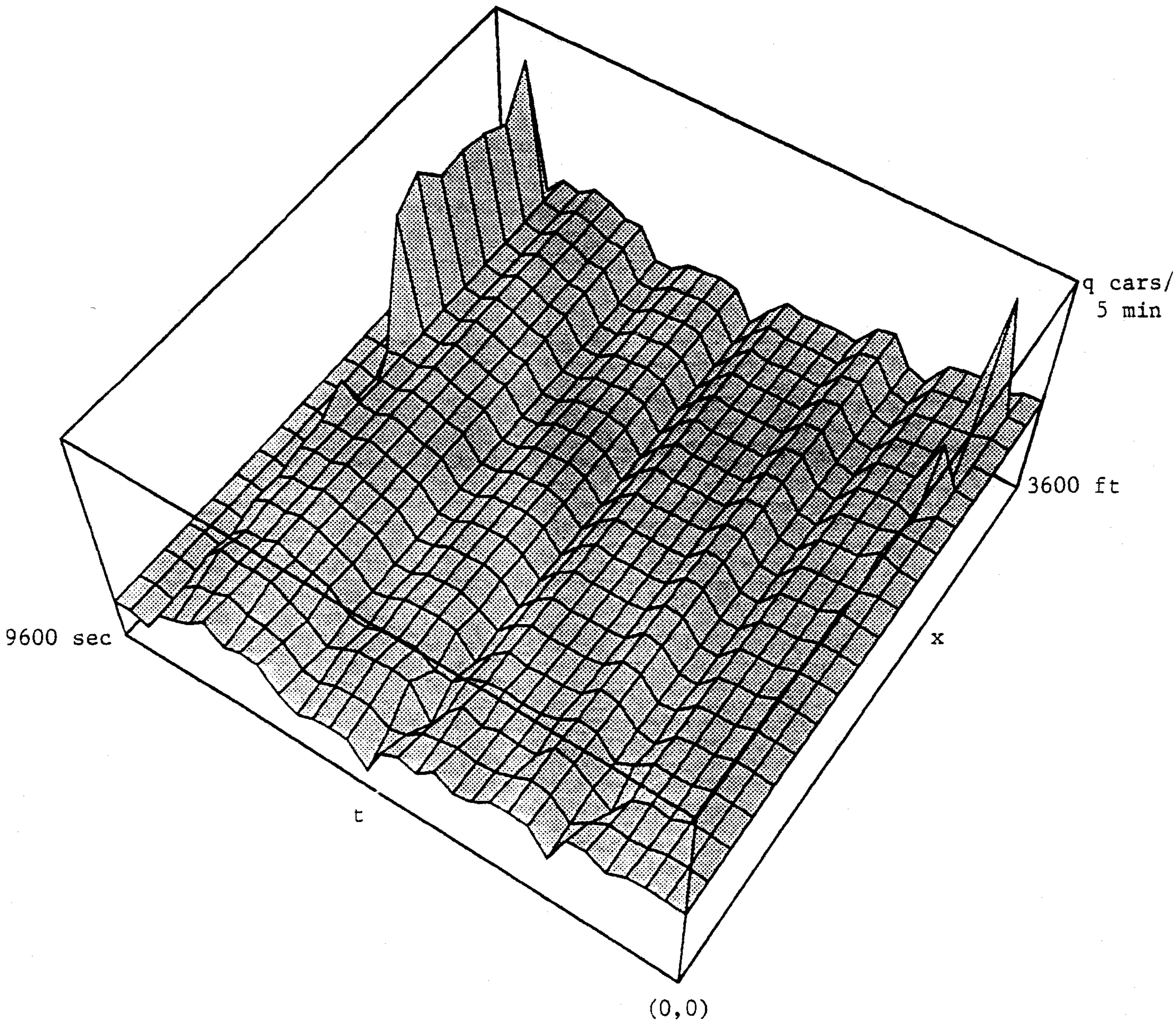
Trapezoid Method
Uncongested Pipeline Flow (veh/5 min/2 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 15 s
omega = 1.0

Fig. 4



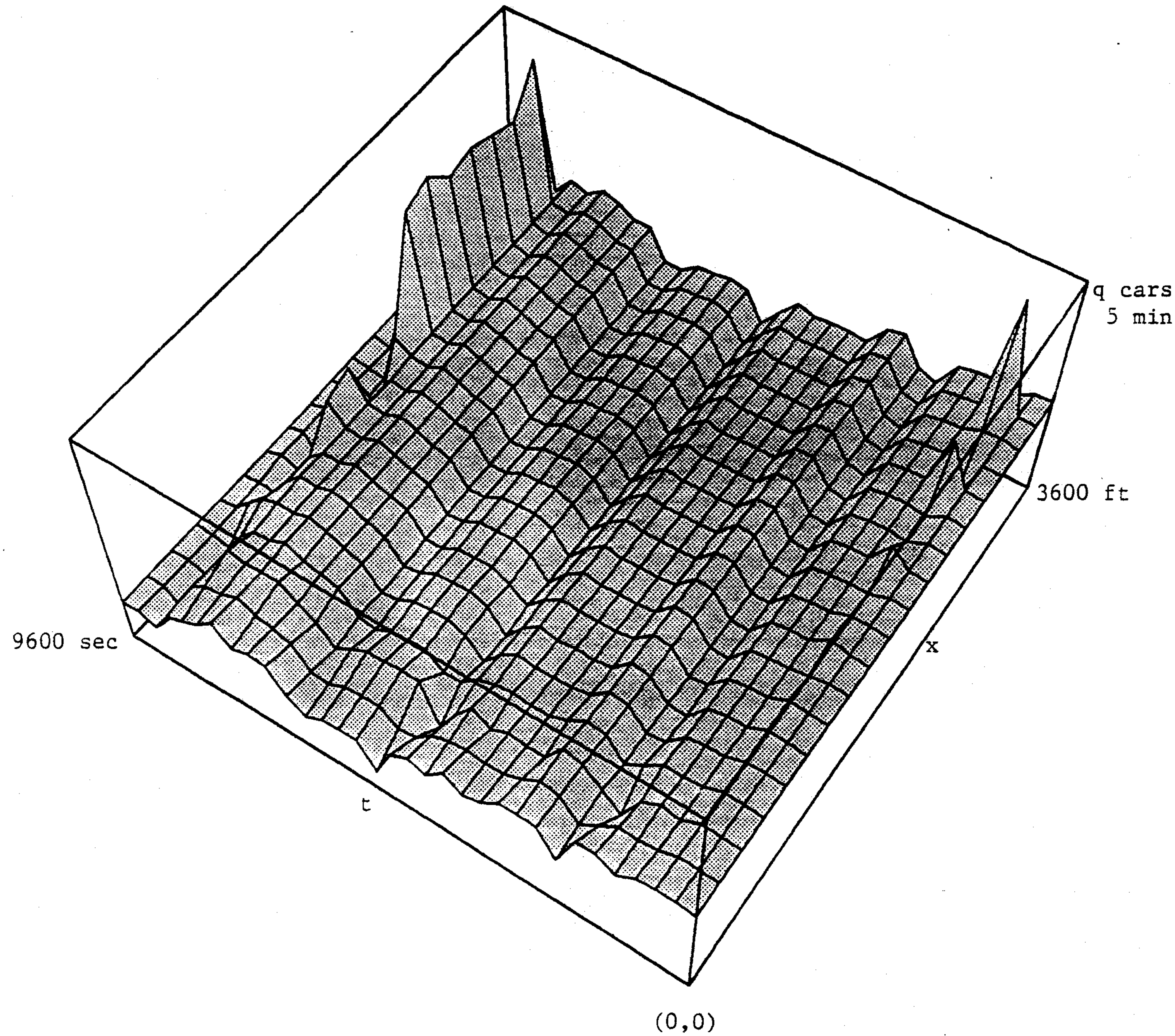
Lax Method
Congested Pipeline Flow (veh/5 min/4 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 1 s

Fig. 5



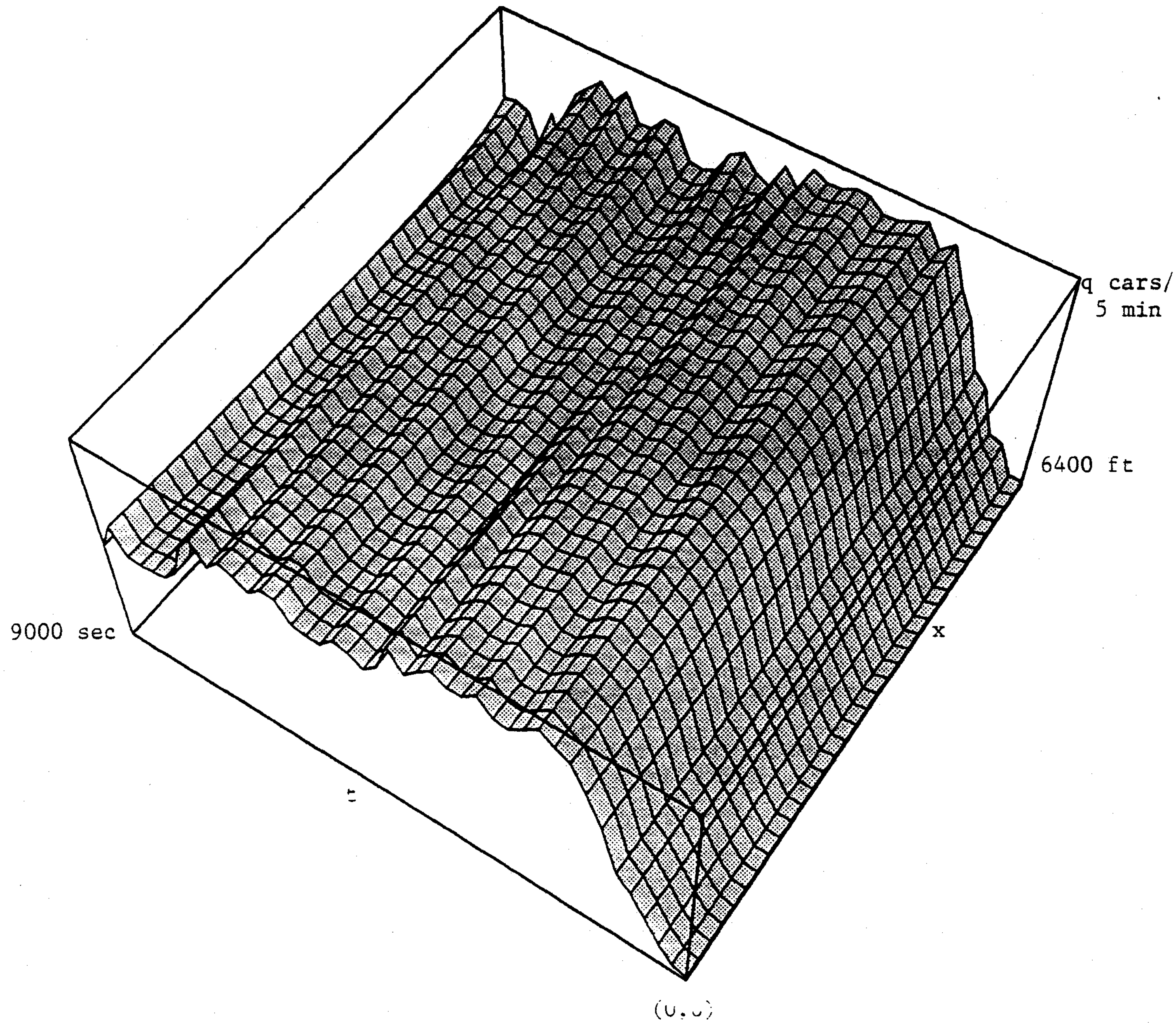
Implicit Euler Method
Congested Pipeline Flow (veh/5 min/4 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 15 s in regular regions
3 s in congestion-change regions
omega = 1.0

Fig. 6



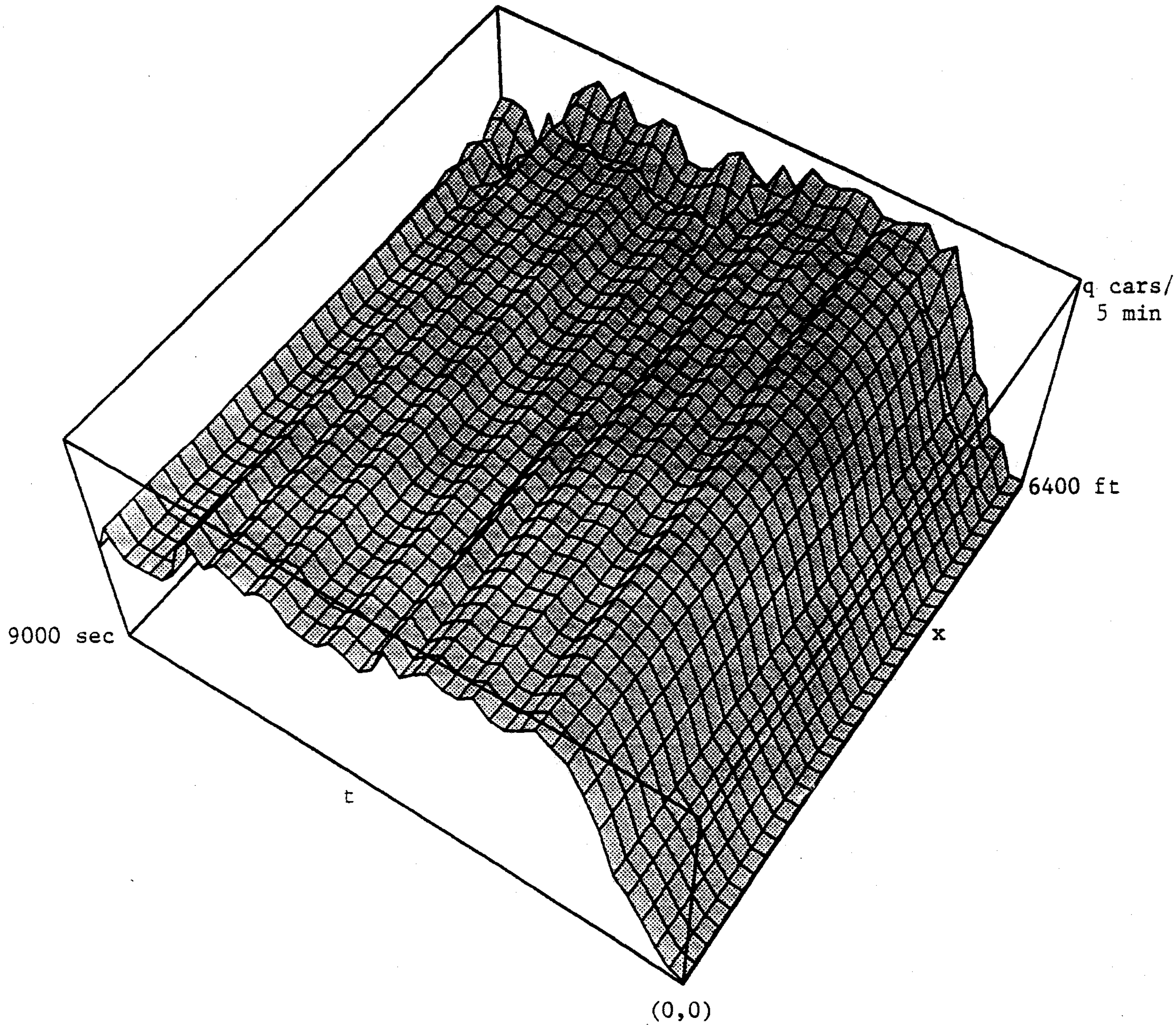
Trapezoid Method
Congested Pipeline Flow (veh/5 min/4 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 15 s in regular regions
3 s in congestion-change regions
omega = 1.0

Fig. 7



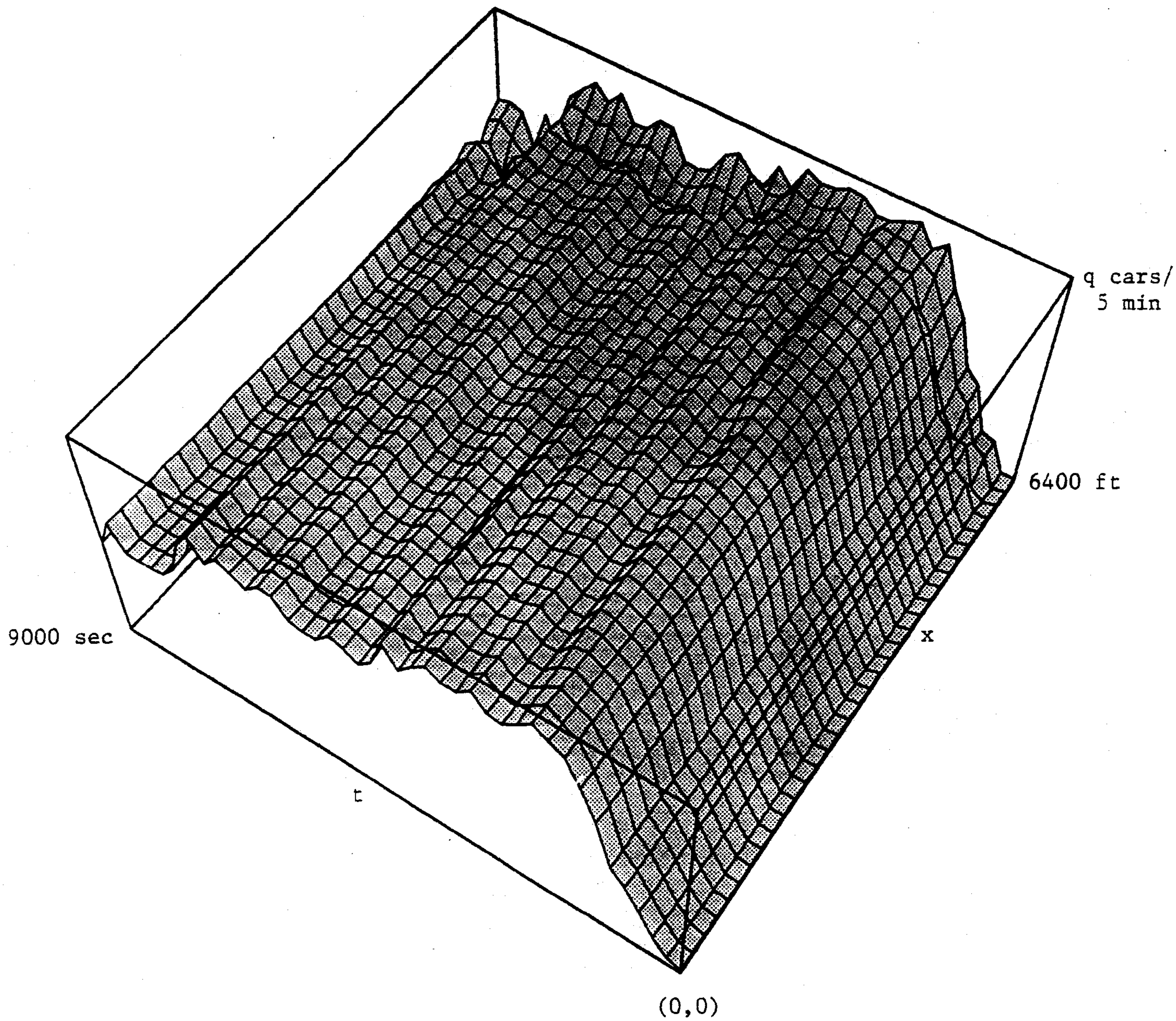
Lax Method
Uncongested Entry/Exit Flow (veh/5 min/3 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 1 s

Fig. 8



Implicit Euler Method
Uncongested Entry/Exit Flow (veh/5 min/3 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 15 s
omega = 1.0

Fig. 9



Trapezoid Method
Uncongested Entry/Exit Flow (veh/5 min/3 lanes)
Quartic Least Squares q-k Curve
dx = 200 ft
dt = 15 s
omega = 1.0

Fig. 10

Lax method on the NCUBE for problem with 512 space points.
Number of processors used: p = 1, 2, 4, 8, 16, 32, or 64.
Time given is maximum single node computation time.

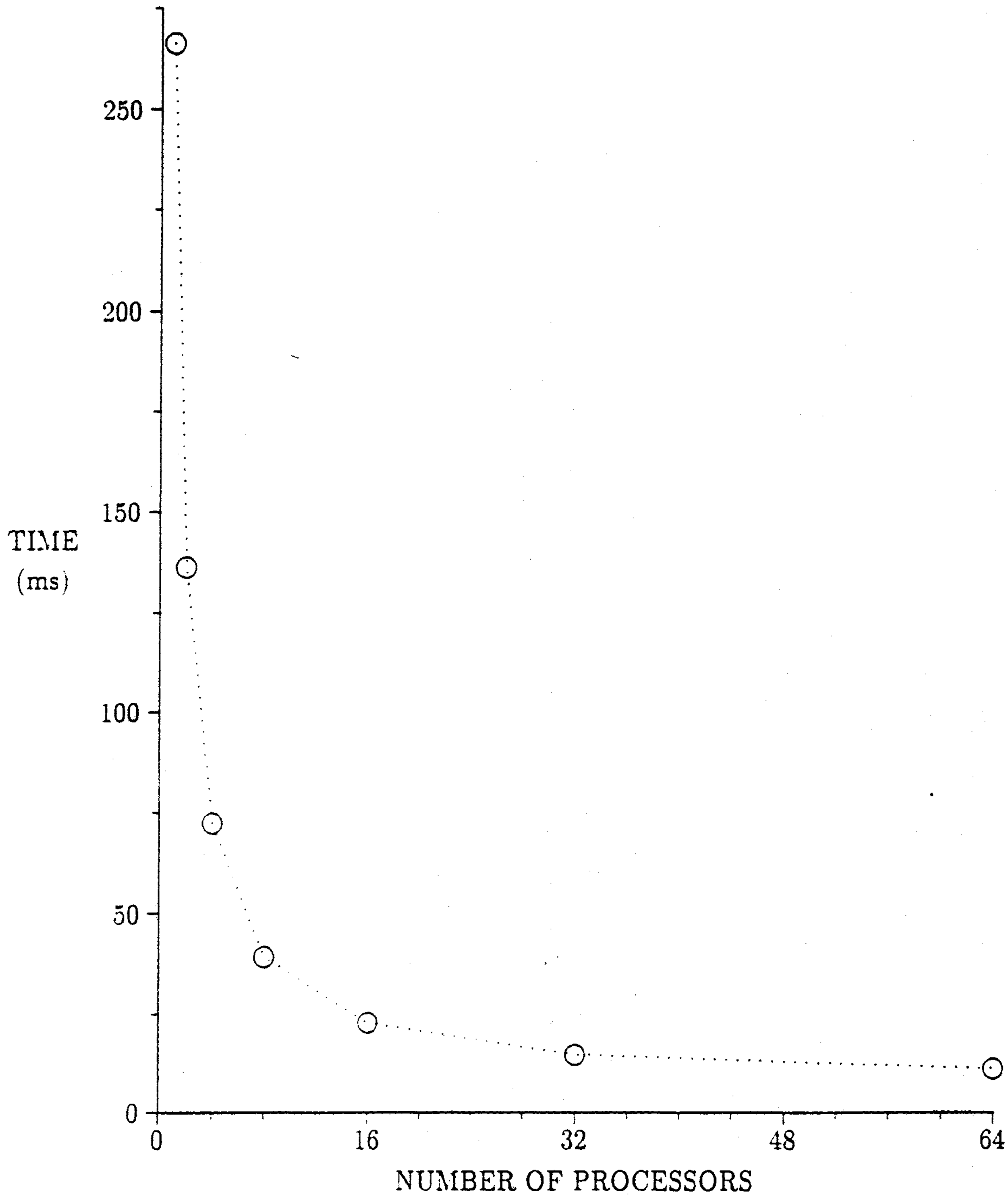


Fig. 11

Speedup achieved for Lax method on the NCUBE.
Problem size: 512 space points.
Number of processors used: p = 1, 2, 4, 8, 16, 32, or 64.
T1 = computation time on single processor.
Tp = maximum node computation time for program on p processors.

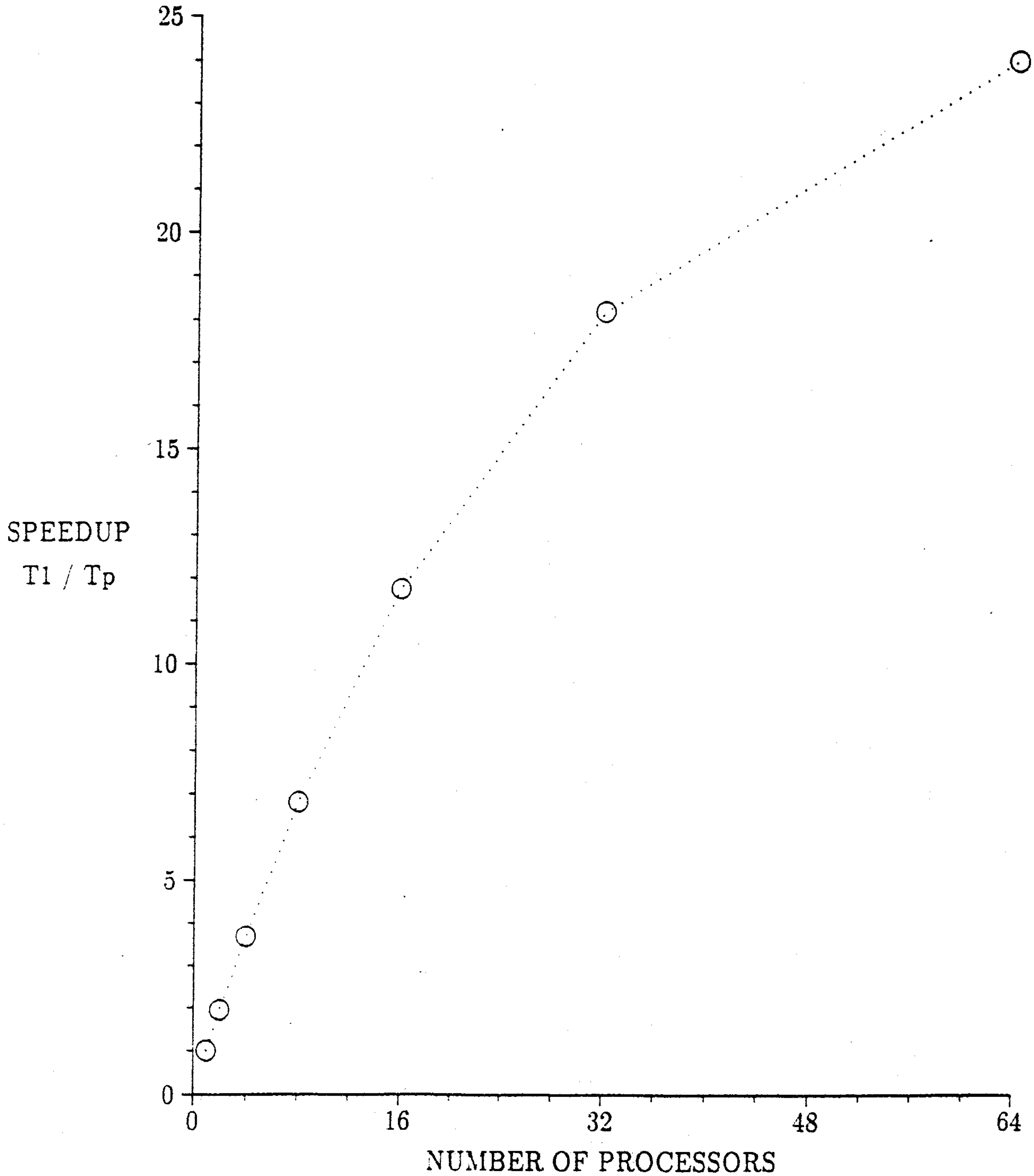


Fig. 12

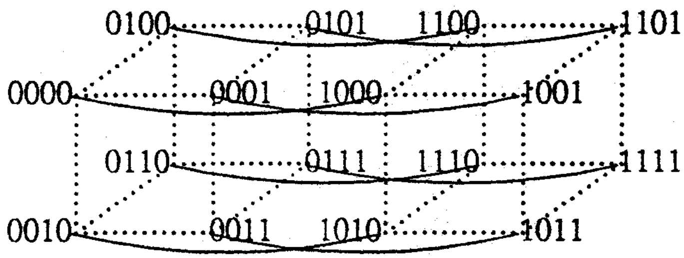


Fig. 13. Hypercube Network of dimension 4.

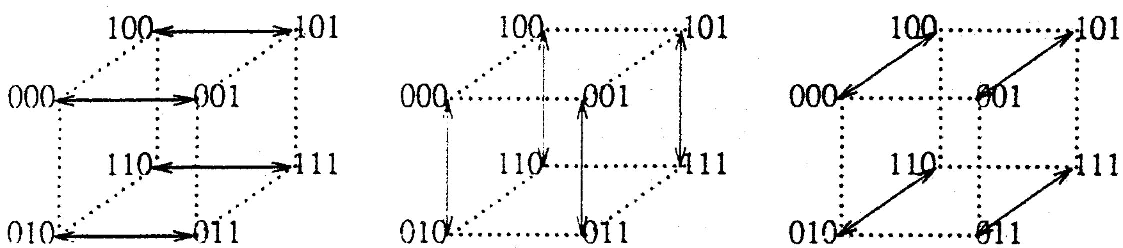


Fig. 14. Examples of data exchanges between adjacent processors.