

**NaSch模型编程及数值模拟**

**学 院： 交通运输学院**

**专 业： 系统科学**

**姓 名： 马智傲**

**学 号： 20120747**

**指导教师： 贾斌教授**

**一、作业要求**

* 模型参数取值: Lroad=1000， p=0.3, Vmax=5
* 边界条件:周期性边界
* 数据统计:扔掉前10000个时间步，对后1000个时间步进行统计需给出的结果
* 基本图(流量-密度关系) :需整个密度范围内的
* 时空图(横坐标为空间，纵坐标为时间，密度分别取0.1,0.2和0.3，画500个时间步即可)

**二、编程设计思想**

**1.设置相关参数**

这些参数包括最大速度vmax、随机慢化概率p、元胞矩阵road、速度矩阵velocities、仿真时间simulation\_steps、采样点数量num\_samples、保存流量-密度采样值的矩阵sample、密度变化的步长density\_step、存放道路状态的矩阵history、存放速度状态的矩阵velocity\_history。

**2. 随机分配车辆**

车辆数等于本次循环的密度乘以道路长度。随机分配车辆的算法是，生成 1000 个不重复的随机整数并提取前 n 个（n为车辆数）赋值给z，z即为这些车对应的元胞矩阵中的位置，从第一辆车开始依次将每一辆车放入对应元胞。判断元胞的取值，非零的即为有车，设置元胞取值为 1。从而得到 0-1 取值的元胞矩阵，1 代表随机分布的车。

**3.速度和位置更新**

（1）计算速度更新。从第一个元胞到最后一个元胞循环，先检测有车的元胞位置，按照 NaSch 模型的更新规则执行加速、减速、随机慢化，当末车位于

最后一个元胞时，按照周期性边界跟新，末车变为首车。

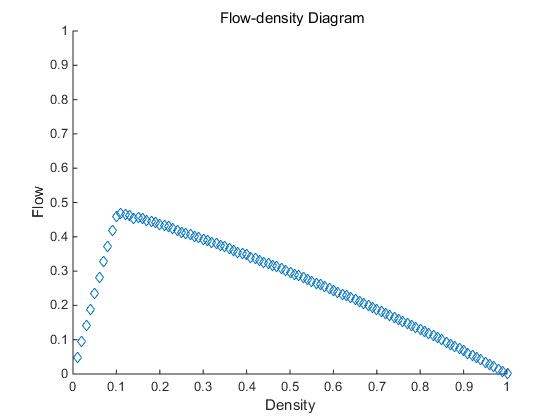
（2）计算位置更新。新位置等于旧位置加速度，遇到边界同上执行。

**4.记录每次采样的速度、流量**

将速度记录到 velocity\_history 变量中，密度等于车辆数除以道路长，流量等于总速度除以总车数乘以总平均车数，将密度和流量点对记录到 samples 变量中。

**三、数值模拟实例**

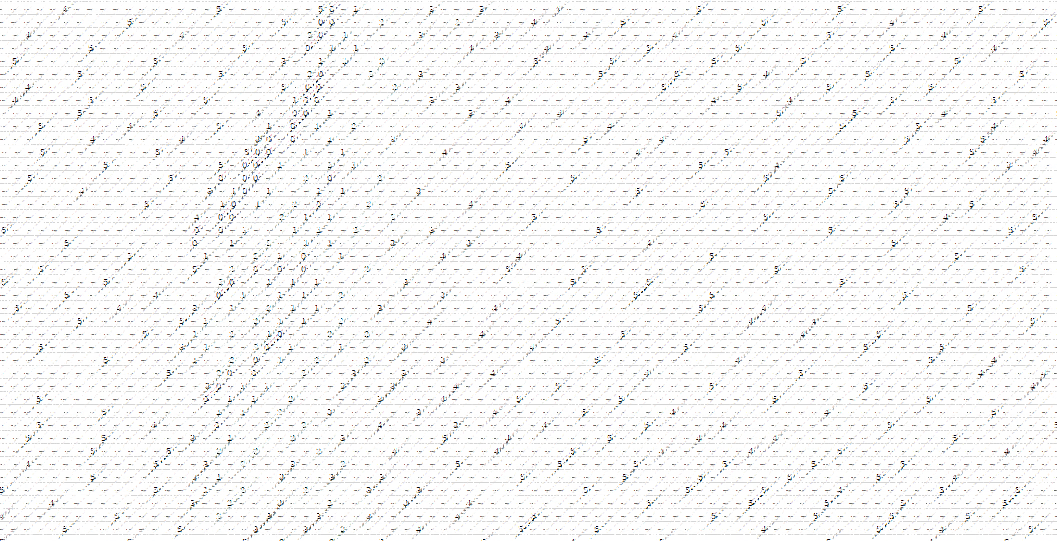
**1.流量密度关系基本图**



**图1 NaSch模型流量密度基本图**

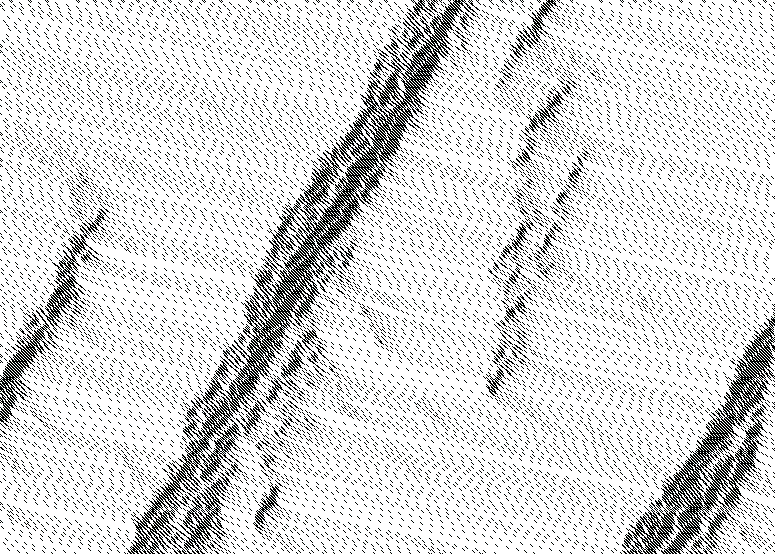
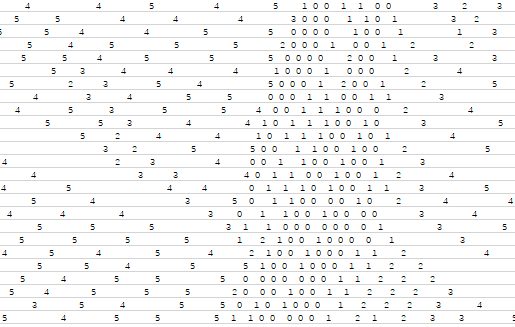
由图1可知，在密度介于0-0.12左右时，流量随密度的增加而不断增大，但当达到最大流量0.48左右后，随着密度的增长流量呈现出相反的变化趋势。

**2.时空图（后500时间步）**



**图2 p=0.1的时空图**

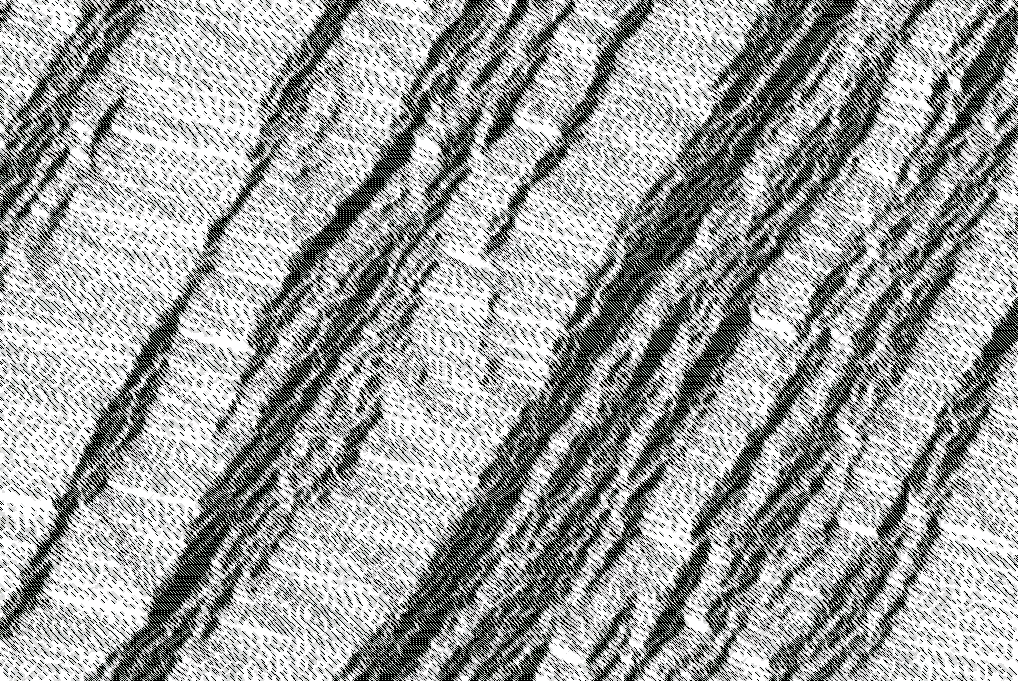
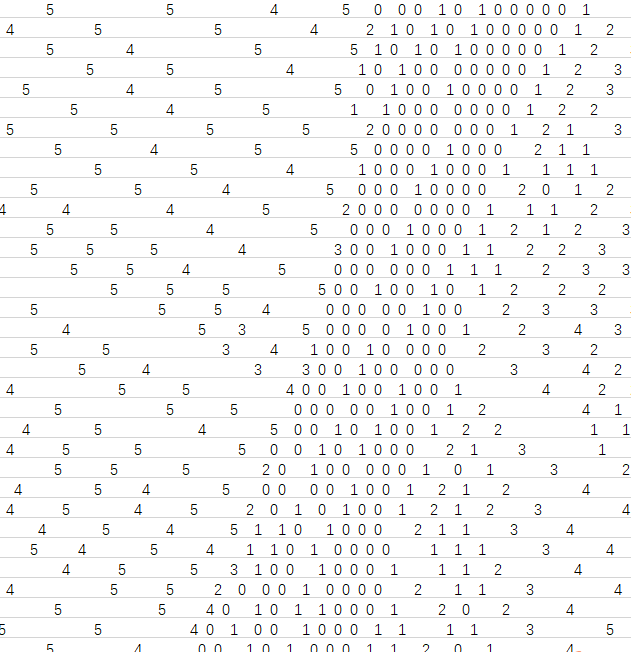
图2为由MATLAB导出的数据在EXCEL中的部分时空图展示，由图2可知，当车道密度取0.1时，整个道路上的车辆绝大多数处于自由流状态，只有极少数的车辆速度达到零，导致道路堵塞。

**(a) (b)**

**图3 p=0.2的时空图 (a)总体时空图 (b)局部时空放大图**

图3为由MATLAB导出的数据在EXCEL中的部分时空图展示，由图3可知，当车道密度取0.2时，道路上的车辆开始出现一些拥堵，时走时停波开始出现，相较于p=0.1时交通状况变差。

**(a) (b)**

**图4 p=0.3的时空图 (a)总体时空图 (b)局部时空放大图**

图4为由MATLAB导出的数据在EXCEL中的部分时空图展示，由图4可知，当车道密度取0.3时，车道上的拥堵情况变得十分严重，时走时停波也变得十分明显，说明数值模拟情况良好。

**附录 MATLAB代码**

%设置参数

vmax = 5;

p = 0.3;

road\_length = 1000;

simulation\_steps = 20000;

render\_on = 0;

pause\_on = 0;

delay\_on = 0;

delay\_length = 0.01; %10 FPS

road = zeros(1,road\_length); %设置道路元胞矩阵

road\_next = road;

velocities = zeros(1,road\_length); %设置速度矩阵

velocities\_next = velocities;

%采样点

num\_samples = 100;

samples = zeros(2,num\_samples); %流量、密度点对应矩阵

density\_step = 1/num\_samples;%密度步长

history = zeros(simulation\_steps, road\_length);%记录所有仿真时间步的道路情况

velocity\_history = zeros(simulation\_steps, road\_length);%记录所有仿真时间步的速度情况

figure

%画时空分布图时候设置成所需要的密度，画基本图时设置num\_samples

for g=1:num\_samples

%道路状态，初始车辆状态

road = zeros(1,road\_length);%完成一轮计算后重新初始化道路

road\_next = road;

density = g/num\_samples;

car=density\*road\_length;

%随机分配车辆

rdm=randperm(road\_length);

z=rdm(1:car);

for i=1:car

j=z(i);

if j==0

road(j)=0;

else

road(j)=1;

end

end

if render\_on

imshow(road);

drawnow

end

%开始仿真

for i=1:simulation\_steps

history(i, :) = road;

velocity\_history(i,:) = velocities;

%--------------------速度更新------------------------%

for j=1:road\_length

if road(j) == 1

distance = 0;

%检测最大加速距离

bf = 0;

for k=1:vmax

distance = k;

if j+k <= road\_length %周期性边界

index = j+k;

else

index = j+k-road\_length; %周期性边界

end

if road(index) == 1

bf = 1;%有车

end

if bf == 1, break, end

end

if velocities(j) < vmax %加速

velocities(j) = min(velocities(j) + 1,vmax);

end

if (velocities(j) > distance - 1) && bf == 1 %减速

velocities(j) = distance - 1;

end

if rand < p && velocities(j) > 0 %随机慢化

velocities(j) = max(velocities(j) - 1,0);

end

end

end

%--------------------位置更新-------------------------------%

for j=1:road\_length

if road(j) ==1

if j+velocities(j) <= road\_length

index = j+velocities(j);

else

index = j+velocities(j) - road\_length; %周期性边界

end

%碰撞检测

if road\_next(index) == 1

disp('Collision detected')

end

road\_next(index) = 1;

velocities\_next(index) = velocities(j);

end

end

velocities = velocities\_next;

road = road\_next;

road\_next = zeros(1,road\_length);

if render\_on

imshow(road);

drawnow

end

if pause\_on

pause

end

if delay\_on

pause(delay\_length)

end

end

%记录每次采样的速度、流量数据

velocity\_history = velocity\_history.\*history;

samples(:,g) = [mean(history(:))

(sum(velocity\_history(:))/sum(history(:)))\*mean(history(:))];%记录密度，流量

disp('Sample step:')

g

end

%绘制基本图

scatter(samples(1,:), samples(2,:),'d');%1为密度横坐标，2为流量纵坐标

axis([0 1 0 1]);

xlabel('Density')

ylabel('Flow')

title(' Flow-density Diagram')

%绘制时空图

space=history+velocity\_history;

for i=0:6

space(find(space==i))=i-1;

end

timespace=num2str(space);

for m=simulation\_steps-499:simulation\_steps

timespace2=strrep(timespace(m,:),'-1',' ~');

disp(timespace2);%显示图形

end