

# タートルグラフィックスを用いた交通流・渋滞のリアルタイム シュミレーションの描画 (J6 版)

Masato SHIMURA  
jcd02773@nifty.ne.jp

2007年9月19日

## 目次

1	セル・オートマン	1
2	ASEP と ZRP を用いた交通流のシュミレーション	2
2.1	ASEP . . . . .	2
2.2	ZRP . . . . .	3
3	タートルグラフィックスのデモの解説	3
4	タートルグラフィックスの作成	5
5	Reference	7

## 概要

ASEP と ZRP を用いたセル・オートマンによる交通流と渋滞のシュミレーションのリアルタイム表示にタートルグラフィックスを用いる。

## 1 セル・オートマン

フォン・ノイマン フォン・ノイマンの (1903-1957) の遺した数学や物理学の周辺の功績について思い当たるところを列挙すると

\*1

1. ノイマン型コンピュータ
2. ヤコビ法
3. モンテカルロ法
4. ゲーム理論 (モルゲンシュタインと共同)
5. セル・オートマン

---

\*1 貴族の称号は父親が 1913 年にオーストリーで購入した

- 6. 一般経済均衡モデル
- 7. 不完全性定理の第2定理 (ゲーデルが先着した)  
 など、広範囲にわたっている。

ブオルフラム *mathematica* を創ったブオルフラムはセル・オートマンの研究家としても知られている。

このように数学を極めた巨人達が数学を補うものとして、セル・オートマンを用いたコンピュータシュミレーションに傾注している。

セル・オートマンの早期の応用例はコンウエイの「ライフゲーム」で、コンピュータウイルスがまだ誕生していなかった頃、多くのコンピュータで生息していたらしい。JのLABにグラフィック付きで入っている。

## 2 ASEP と ZRP を用いた交通流のシュミレーション

ASEP と ZRP を用いた交通流と渋滞のシュミレーションモデルが活躍し始めている。詳細は別稿とし、簡単に紹介する。

### 2.1 ASEP

最近非平衡系の可解モデルとして注目されている非対象単純排他過程・ASEP (Asymmetric Simple Exclusion Process) はセル・オートマンを用いたモデルである。これの交通流の渋滞モデルへの応用が進んでいる。

ASEP は、生物学で細胞内のたんぱく質生成過程を研究していたマクドナルドとギブスによって 1968 年に考案されたモデルでタンパク質の伝送での渋滞の有無を検討したものが最初であると言われる。

1 車線 (2 次元) のモデルを考える。ルールは非常に単純である。

1. できること
  - (a) 前が空いていれば進む
  - (b) 前が詰まっていれば進めない
2. できないこと
  - (a) 追い越し
  - (b) 突き飛ばしたり恫喝して前をあげさせる

この単純なルールで行進 (草食恐竜、野牛、亀、蟻、車) のシュミレーションができる。

0/1 は確率で打ち出す。これを  $\rightarrow$  の方向に進める。

$(0.5 * 24) > 24 ? 24$  NB.  $p=0.5, n=24$

0 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0

ほんの少し手を加えればランダムウォークになる。

$+\backslash(12 > 24 ? 24)\{ 1 \_1$

1 2 1 2 3 4 3 2 3 2 1 0 1 2 3 2 1 0 1 2 1 0 1 0

ASEP の行進モデル  $\rightarrow$  に進む。上が  $t$ 、下が  $t+1$  である。

TMP, : 1 forward\_sub\_p TMP

```
1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 1
0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1
```

行進モデルとして作成する場合、前方に押さえがないと快調に行進して早い時期に整流になる。従って自己の行列をカウンターで制御して 0/1 を同確率で与えて前方の指標とするか、または、自己の前方に同じ確率の複製を何個か置き、同時に駆動することで行進を制御する必要がある。

## 2.2 ZRP

ZRP(Zero Range Process) は箱に複数の玉を収納し、隣の箱に玉を移す確率はその箱に入っている玉の数に依存する確率過程であり、分岐ルールを定めれば、車間変更のモデルに応用できる。

慎重しやかな 慎重しやかなレーン変更

1. 集団の 4 台目以降が変更を考える
2. 空いている場合のみ行動する
3.  $t+1$  で横に (のみ) 動く

些か行動的な 些か行動的なレーン変更

1. 1, 2 は同じ
2.  $t+1$  で斜め前に (のみ) 動く

もっと行動的なモデルも考えられるが、このような単純なモデルでも、行進は少しの時間の経過で団子状になっていくのが観察できる。

## 3 タートルグラフィックスのデモの解読

Fraser Jackson が J にタートルグラフィックスを移植してくれた。デモのなかに、星が浮かんでいる絵があり、これを車の挙動の表示に応用できないかと考えた。

次の図と SCRIPT を見比べながら解読することから始めた。  
(この章は記録であるので、読み飛ばしても差し支えない)

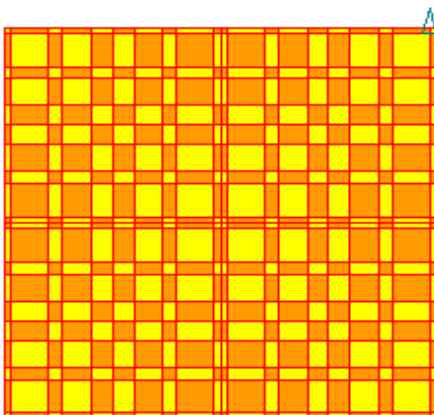


図 1 aya

This script assumes more knowledge of J but shows how you can build complex patterns and use multiple colors.

```

k=: >:i.5 NB. Use 5 increasing width columns and rows
kc =:}: 0,+/\ k0 =: ,k,.,k NB. Interleave them

cONE =: 255 255 0 NB. Set colors Orange and yellow
cTWO =: 255 155 0

kcol =: 0 1 NB. Identifiers to select colors

j =: rect each |. each k1 =:{k0;k0 NB. construct program for rectangles
j =: 100 24$,>j NB. Convert to a simple array
colseq0 =: 100$ (10$0 1),10$1 0 NB. Color sequence starting with cONE
colseq1 =: 100$ (10$1 0),10$0 1 NB. Color sequence starting with cTWO
NB. add color commands to rectangles
jn0=: (colseq0 { (color cONE fill rt 0),: color cTWO fill rt 0),.j
jn1=: (colseq1 { (color cONE fill rt 0),: color cTWO fill rt 0),.j

NB. Generate the starting positions of a single set of rectangles
js =: ,kc j. / kc
NB. generate four blocks of rectangles using changing color pattern
show ( js,(0j30+js),(30j0+js),30j30+js) start <"1 jn0,jn1,jn1,jn0

```

図は大小のタイルを埋め込んだ壁面と考えると次の k0 の組み合わせが行と列の幅と位置を定義しているようだ。each は L:0 のことである。

```

k
1 2 3 4 5
k0
1 5 2 4 3 3 4 2 5 1
kc
0 1 6 8 12 15 18 22 24 29

|. L:0 |: L:0 k0;k0
+-----+-----+
|1 5 2 4 3 3 4 2 5 1|1 5 2 4 3 3 4 2 5 1|
+-----+-----+

```

ここに形の定義 rect を組み入れると 32 桁の形の定義ベクトルができる。これを基礎にして色と位置の 45 桁のベクトルを描くタイルの回数分作成して、show で一度に見せてしまうことになる。

```

,.rect L:0 |. L:0 |: L:0 k0;k0
+-----+
|1 1 2 0 0j_1 1 5 1 2 0 0j_1 1 1 1 2 0 0j_1 1 5 1 2 0 0j_1 1 2 4 3 3 4 2 5 |
+-----+
|1 1 2 0 0j_1 1 5 1 2 0 0j_1 1 1 1 2 0 0j_1 1 5 1 2 0 0j_1 1 2 4 3 3 4 2 5 |
+-----+

```

```
# L:0,.rect L:0 |. L:0 |: L:0 k0;k0
```

```

+---+
|32|
+---+
|32|
+---+

```

色の定義は先ず RGB を定義して、カラーテーブルを並べ、乱数で取り出している。

```

cONE =: 255 255 0 NB. Set colors Orange and yellow
cTWO =: 255 155 0

```

```

jn0=: (colseq0 { (color cONE fill rt 0),: color cTWO fill rt 0),.j
jn1=: (colseq1 { (color cONE fill rt 0),: color cTWO fill rt 0),.j

```

start で位置情報 (js) と色と形のベクトルを結合して 45 桁のベクトルをタイルの枚数分作る。位置情報があるので一覧表のようにべったりと { で box にして、重ねておけばよい。

最後に show で描画する。

```
show ( js, (0j30+js), (30j0+js), 30j30+js) start <"1 jn0, jn1, jn1, jn0
```

このように色と形状はタートル側で作成し、位置はシュミレーション側で作成して最後に組み合わせる。リアルタイムの動画にもできる。

タートルの数は 90 段 × 30 列程度は可能なようだ。

## 4 タートルグラフィックスの作成

データを先に計算 (120 ステップ程度なら即時) しておいて動画で表示する。↑ に進む。次の図と SCRIPT を見ながら解説する。(亀(車)の120ステップの動きは、1ステップ1秒とすれば2分に相当する。)

レーン 2レーン

RAW=: 1 5 とし少しあげた。

縦のサイズ 縦 (長さ) のサイズはデータから自動指定 (60 程度) としている。

```
'SIZE TIMES'=: ({: $ ; {. y) ; # y
```

色 三色重 A は白 (0) にライム (1)、B は白 (0) に青 (1)。車線変更はピンク (2) で他と区別する。



図2 ASEP/ZRP

```
COL0=(color CL1 fill rt 0),(color CL2 fill rt 0),: color CL4 fill rt 0 NB. Lane A
COL1=(color CL1 fill rt 0),(color CL3 fill rt 0),: color CL4 fill rt 0 NB. Lane B
```

データも 0,1,2 で打ち出す。

データ 次のような形をしている。

```
+-----+
|1 0 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 0 1 2 1 1 0 1 1 0 0 0 1 1
|1 0 0 0 1 0 1 2 1 1 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0 1 1 0 1
+-----+
```

亀 亀の形は正六角形のタートルグラフィックスを用いる

```
J0=. repeat 6 fd 1 rt 60 NB. Hexagon
```

データと色の結合 CLR0=. { . TMP { COL0 NB. 30 column fixed

```
CLR1=. { : TMP { COL1 NB. 30 column fixed
```

動き 120 ステップとした。

```
while. COUNTER < 120 do. NB. TIMES do.
```

六角形とデータ、色の結合 CLR2=. CLR0, . >(# CLR0) #<J0

```
CLR3=. CLR1, . >(# CLR1) #<J0
```

ここまでが描画パーツで形と色の組み合わせの情報である。

描画位置 横は実数、縦は複素数で表示。(show で立ち上がる)

```
1 5 +/ j. (2* >: i. 10)
1j2 1j4 1j6 1j8 1j10 1j12 1j14 1j16 1j18 1j20
5j2 5j4 5j6 5j8 5j10 5j12 5j14 5j16 5j18 5j20
```

描画位置と描画パーツの組み合わせ 6 角形一つに 45 項の情報が入る。6 角形一つずつボックスの形をしている。

```
((SIZE{. JS) start {CLR2},((- SIZE){. JS) start {CLR3
```

表示 最後にボックスをほどいて、show

#### 4.0.1 Script

```
turtle_zrp=: 3 : 0
NB. y is anal_each_zrp0 0.5
'SIZE TIMES'=:({: $ ;{: y) ; # y
CL1 =. 255 255 255 NB. empty white
CL2 =. 0 255 0 NB. A Lime
CL3 =. 0 0 255 NB. B Blue
CL4 =. 255 0 255 NB. change Lane Fuchsia
NB. -----
COL0 =. (color CL1 fill rt 0), (color CL2 fill rt 0), : color CL4 fill rt 0 NB. Lane A
COL1 =. (color CL1 fill rt 0), (color CL3 fill rt 0), : color CL4 fill rt 0 NB. Lane B
NB. -----
COUNTER =. 0
while. COUNTER < 120 do. NB. TIMES do.
TMP =. ; COUNTER{ y
CLR0 =. {: TMP { COL0 NB. 30 column fixed
CLR1 =. {: TMP { COL1 NB. 30 column fixed
NB. construct program for rectangles
RAW =. 1 5
J0 =. repeat 6 fd 1 rt 60 NB. Hexagon
CLR2 =. CLR0, . > (# CLR0) # < J0
CLR3 =. CLR1, . > (# CLR1) # < J0
NB. -----
JS =. ; RAW +/ j. (2* >: i. SIZE) NB. i. 60 turtle block (2/3) is important
ht'' NB. hide turtle
show ; ("1), ((SIZE{. JS) start {CLR2}, ((- SIZE){. JS) start {CLR3
COUNTER =. >: COUNTER
end.
)
```

## 5 Reference

西成 活祐 渋滞学 新潮選書 新潮社 2006

金井政宏・西成活祐・時弘哲治 交通量の確率モデルについて 九州大学応用力学研究所研究集会報告 2005