



# JAVAアニメーションによる 歩道の渋滞について

中央大学 理工学部 物理学科 香取研究室 古沢佑樹

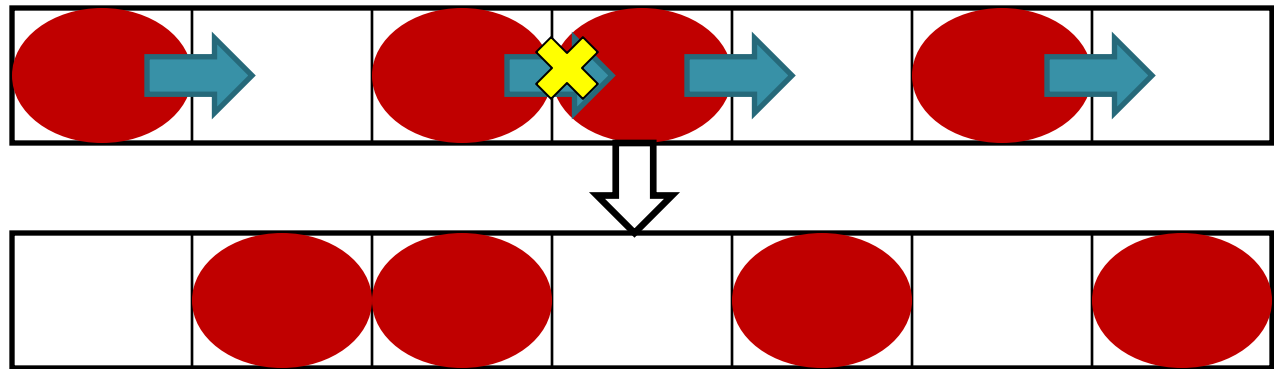
2012年1月30日(月)

# 目次

1. ASEP
2. 歩道のモデル
3. JAVAについて
4. まとめ

# I. ASEP

- ASEPとは交通渋滞などを考える上で、簡単で理解し易く、とても性質の良いモデルである。
- ルールはまず、たくさんの箱を道路、玉を車だと考える。
- 1つの箱には1つの玉しか入れないとする。
- 玉を一斉に右隣の箱に移動させるが、ただしその時右隣の箱に玉が既にいる場合は動けない。
- この操作を繰り返す。



- ASEP(Asymmetric Simple Exclusion Process)  
非対称単純排除過程
- 非対称：バックせずに右にしか進まない。  
(左右非対称)
- 排除：1つの箱に1つの玉しか入れない。

また以下のような条件を加えたりできて工夫がしやすい。

- 必ず前に進むのではなく、確率 $p$ で進める。
- 端まで来たら初めの所に戻る(環状) など

## 2. 歩道のモデル

- ASEPを使って車ではなく歩道について研究した。

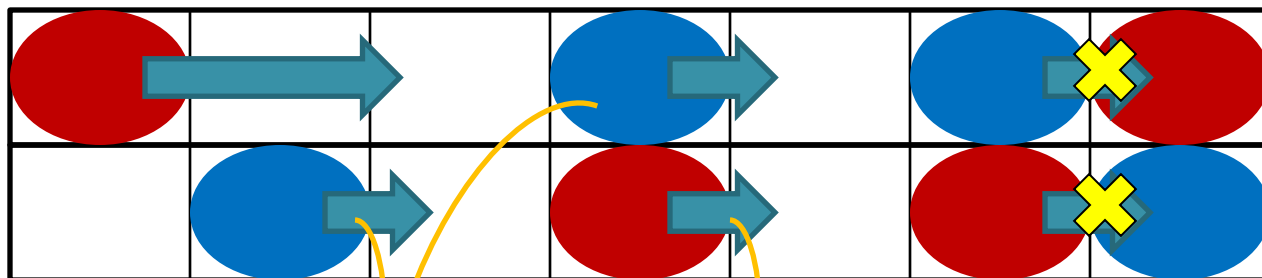
### 研究の目的

- 私は普段歩くスピードが比較的速い方だが、前に遅い人がいるとイライラしてしまう事がある。
- どうすればスムーズに進めるか、つまりどうすれば歩く人の渋滞をなくす事が出来るかを調べたい。
- また、JAVA言語の勉強も兼ねて歩道のシミュレーションを作り、アニメーションを用いて視覚的に直感的にモデルを考察する。

## 歩道のモデルのルール

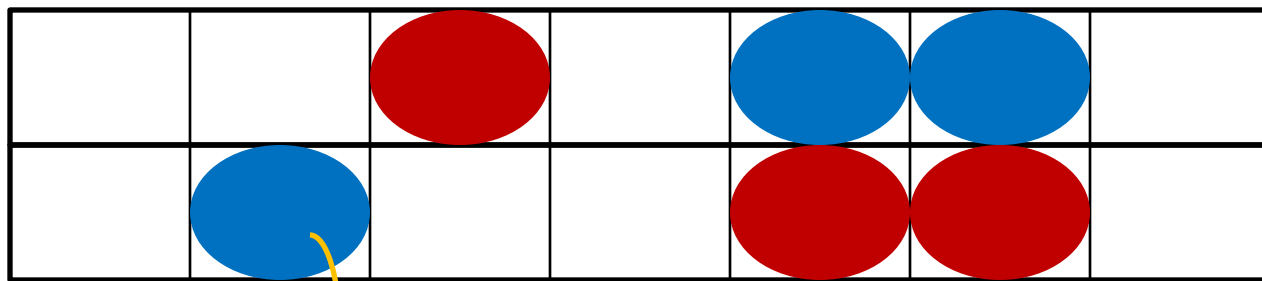
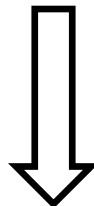
- 歩行者は右へ一方向に進むとする。
- 歩道の幅は2車線とする。
- 速く歩く人と、遅く歩く人を考えるために2マス進む人（赤玉）と1マス進む人（青玉）を準備する。
- 1マス進む人は確率 $p$ で前に進めるとした。  
（ $p$ はシミュレーション時に指定できる。）  
ただし前に人がいる場合は車線変更できる。
- 2マス進む人は100%の確率で進むとした。  
前に2マス以上空いていれば2マス進む。2マス空いていない場合は車線変更する。車線変更もできなくて、前に1マスのみ空いているときは1マス進む。（車線変更のルールは後述）

例



確率 $p$

1マスだけ



確率 $(1 - p)$ で  
進めなかった

# 車線変更のルール

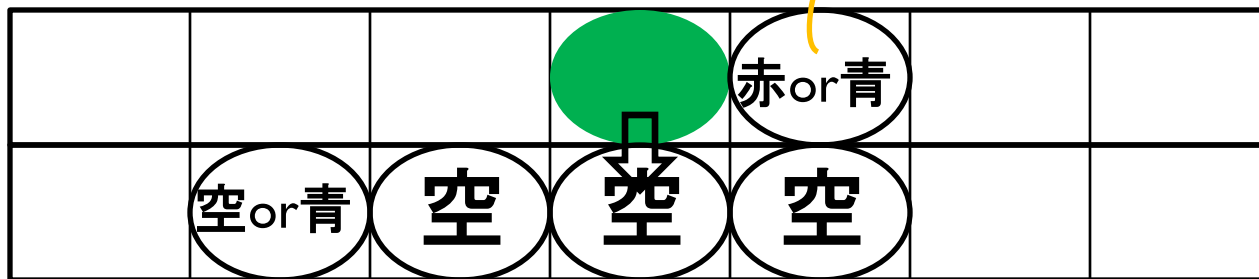
- 遅い人（青）、速い人（赤）はどちらも以下の状況で車線変更ができる。  
(ただし、青は確率 $p$ で車線変更できる。)

以下の状況で緑は車線変更できる。

空：人がいない

空欄：人がいてもよい

ここは赤が車線変更する場合のみ「空」でも良い。  
(1マス進むより車線変更が優先される)





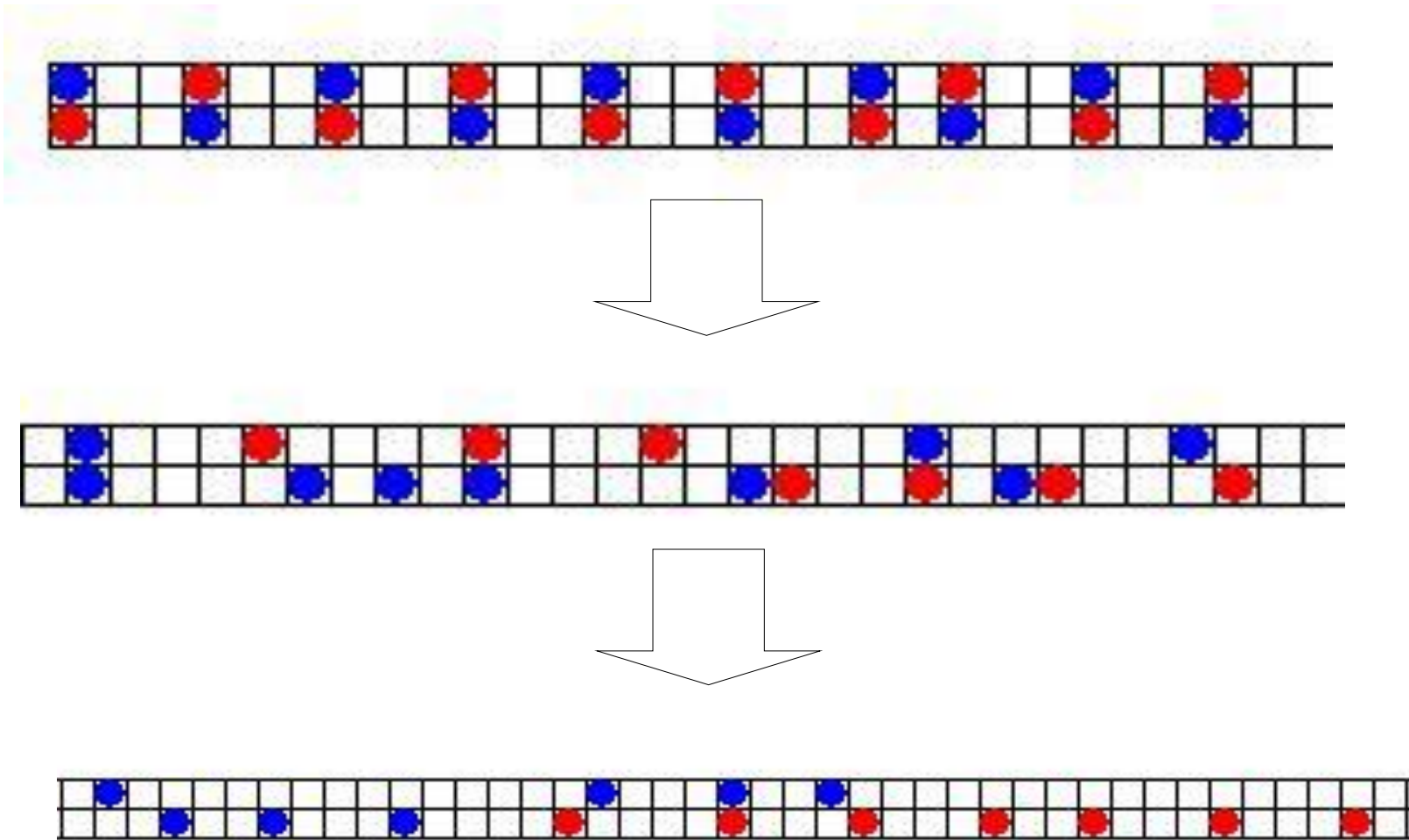
## 細かい話

- 一部きつい条件もあるが、現実の人間の動きにより近いものを再現するために設定している。
- 初期配置は密集させずに赤青均等にバラけさせて置いた。
- $p = 100\%$ にすると青がカベになってしまい、車線変更せずにほぼ初期配置の形のままに進む。
- 遅い人は時々立ち止まりながら、速い人は1マスでもとにかく急いで歩くという事を想定した。
- 青も車線変更しないと極端な渋滞を起こし、赤が進めない事があった。
- 赤を1マスでも動くようにしないと、青の方が速い場合があった。
- アニメーションでご覧下さい。

## 結果

- やはり全体的に赤の方が速く進む。
- $p = 70\%$ 程度にすると初期配置で赤青バラバラにしても、時間がたつとおおよそ**赤の列、青の列に分離**する。
- しかし、人の数が多いと青同士、赤同士で渋滞を起こしてしまうのは変わらない。
- 初期配置を密集にしても、同様に赤青分離する。
- また初期配置で初めから赤青分離させても、分離状態を保つ。

# 結果の一例

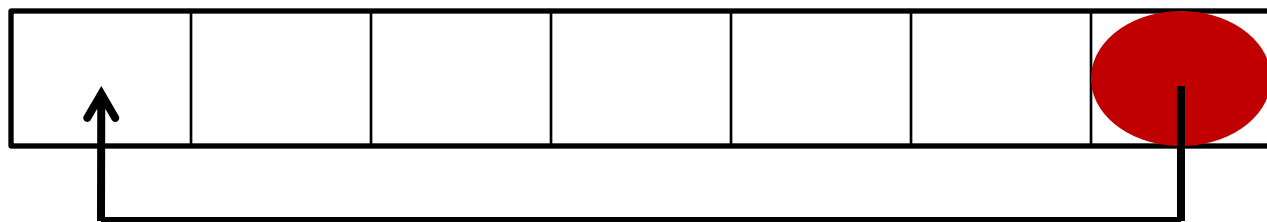


### 3. JAVAについて

- JAVAとはC言語をもとにして作られた、インターネット向けのアニメーションもできる汎用プログラミング言語である。
- 近年は企業でのシステム開発でよく利用されている。
- Web上で動くのでパソコンの機種や環境に関係なく、どこでも動かすことができるのが強み。
- FORTRANは科学技術計算に特に強い、言語である。
- アニメーションの部分以外のプログラムはJAVAもFORTRANもほぼ共通である。

## 苦勞した点

- プログラミングの難しい文法を勉強するよりかは、アニメーションで見せたい状況を作りだすのに工夫が必要だった。
- 例えば、環状に先端まで行ったら一番初めに戻るというのを実現するために、代数学（合同式）の知識が役に立った。



- 繰り返し文の条件に、マス数で割った余りを利用した。
- これを思い付けたのは、物理・数学を勉強してきたおかげだと感じた。

# 4. まとめ

## 歩道のモデルについて

- 時間がたつと速い人の列と、遅い人の列に分離する。つまり急いでいるのであれば、**速く歩いている人の後ろにくっついて歩く**のが良い。
- この分離するという現象は他にもある。例えばエスカレーターでは左側が歩かない人の列、右側が歩く人の列という暗黙のルールがあり、列が分離していると言える。また二車線以上ある道路でも、左車線は初心者の遅い人で右車線は追い越し車線である。
- まとめると速いものと遅いものが混ざり合うと時間がたつにつれて両者は、人間心理ではなく自然現象として分離すると言える。
- 今後の課題としては、歩道をもっと現実に近いモデルにするために、反対方向からも歩いてきたり、3車線以上にして二人組で並んで歩く人を用意したりというシミュレーションを作ってみたいと思う。

## JAVAについて

- 私の今回の研究発表では数式を一切使っていません。データをとってグラフにしたり、解析的に研究したりという事をほとんどせず、アニメーションを動かして視覚的にどうなっているかを調べました。
- 数値計算をしたり複雑な関数を解いたりして研究する事ももちろん大切なことですが、「渋滞学」などのような理科系でない一般の人にとっても身近な問題を考える時は、数値計算の結果や難しい数式で説明するのではなく、JAVAアニメーションなどのように見れば誰にでも分かるような説明をすることも重要だと思います。
- 今後身近なテーマで研究をするようであれば、誰にでも伝わるような説明をしてもらえればなと思います。その道具の一つとして、JAVAアニメーションがあります。

以上