ライフゲーム
～MATLAB実験～

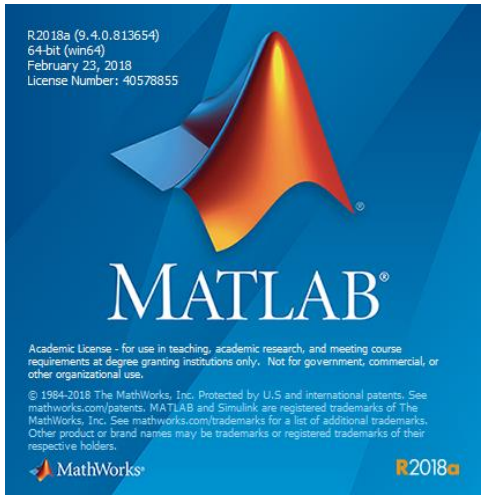
2019年8月24日（土）12:50 ～ 14:30

静岡理工科大学　情報学部　コンピュータシステム学科
幸谷　智紀 (こうや　とものり)
https://cs-tklab.na-inet.jp/
TA:　小杉亮太，VU TRUNG TIN，水野雄太，LIU CHEN

本日のメニュー

1. 「MATLAB」と「生物系」
2. 離散系シミュレーション
3. MATLABの操作方法
4. ライフゲーム体験

1.「MATLAB」と「生物系」



- 「MATLAB」（マトラボ，マトラブ）とは？

・・・40年以上の歴史ある統合型数値計算ソフトウェア。様々な方程式を解いたり，結果をグラフ化したりできる。

- 「生物系」(Biological system)とは？

・・・一人では生きていけない生物の相互作用を表現したもの。ここでは数学的に表現できるものだけを扱う。

２．離散系シミュレーション

- シミュレーション(simulation)とは？
- 離散(discrete) vs. 連続(continuous)
- 離散系シミュレーションの例・・・ライフゲーム

シミュレーション(simulation)とは？

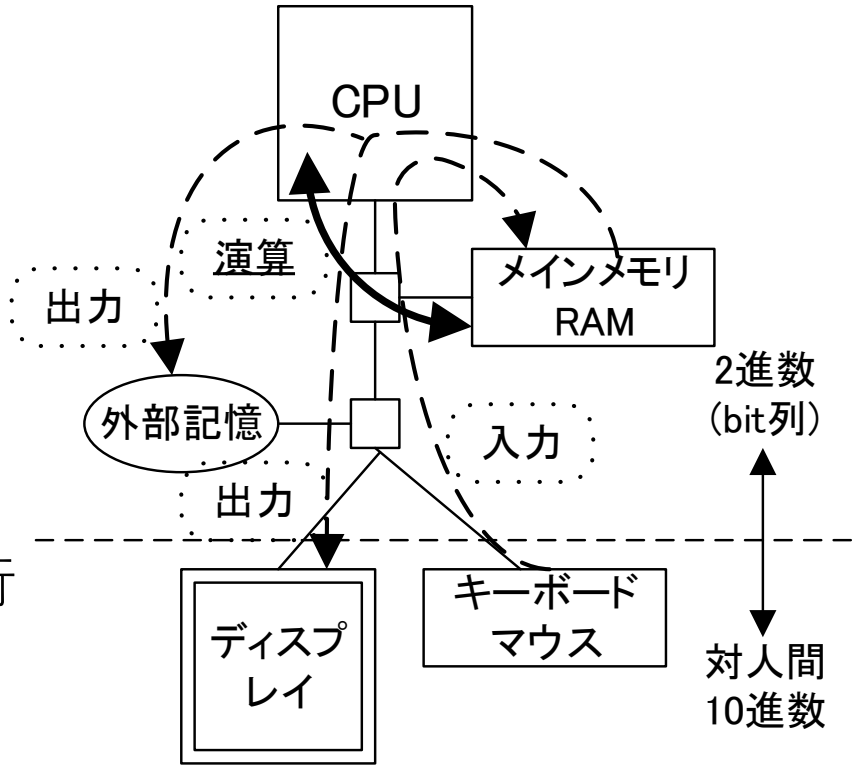
- さまざまな現象をコンピュータ上で模擬的に再現すること。
- 全ての現象をコンピュータが実行できる「計算」として表現する必要がある。

↓なぜ？↓

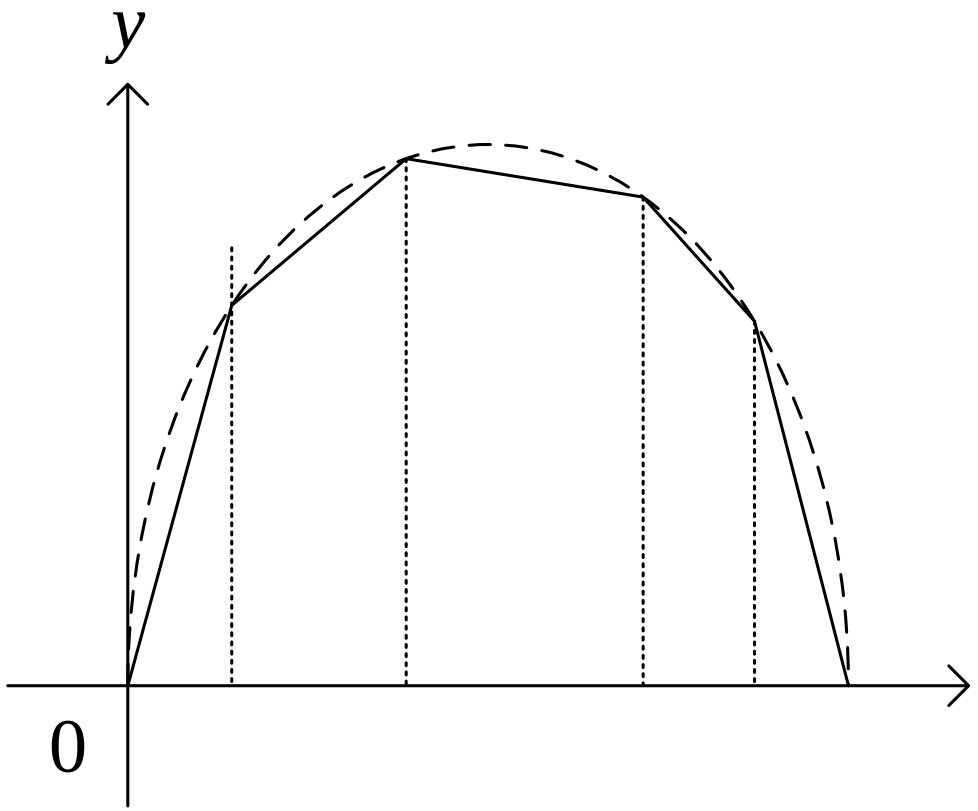
- コンピュータ(ハードウェア)ができること
  - [プログラム] メモリに置かれた手順書。「ソフトウェア」とも言う。

↓プログラムに従って処理を実行

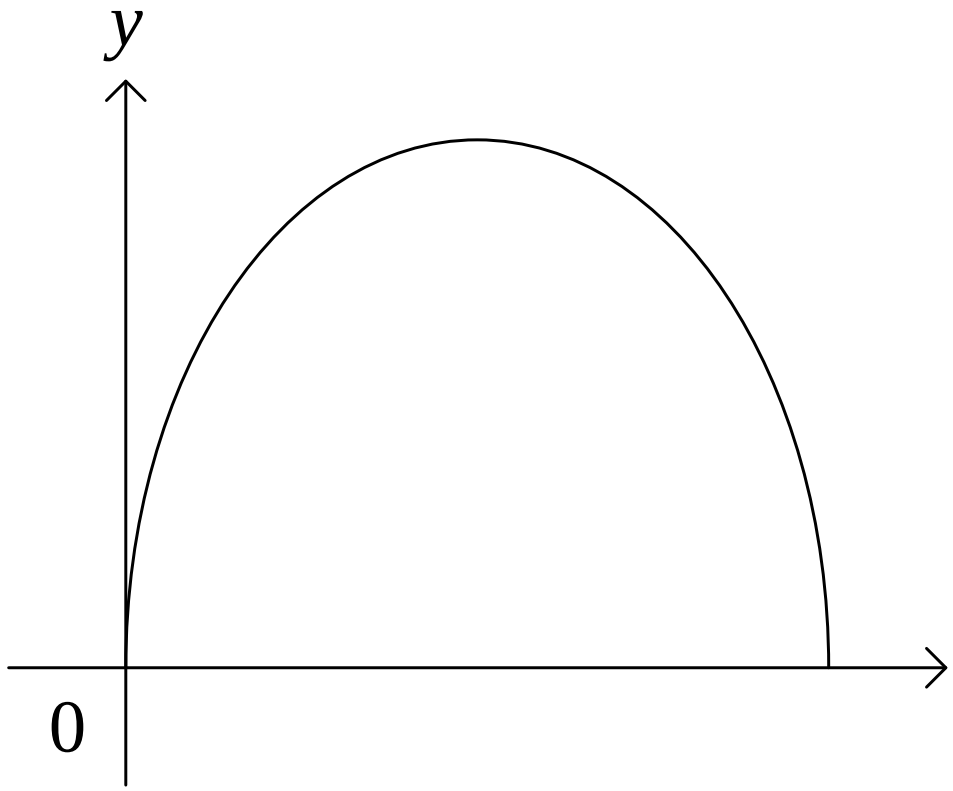
  - [入力] データを読み込み，メモリに蓄える
  - [演算] メモリからデータを取得して計算（演算）を実行し，結果をメモリに書き戻す
  - [出力] メモリにあるデータを人間が理解できる形でディスプレイなどに表示



離散(discrete) vs. 連続(continuous)



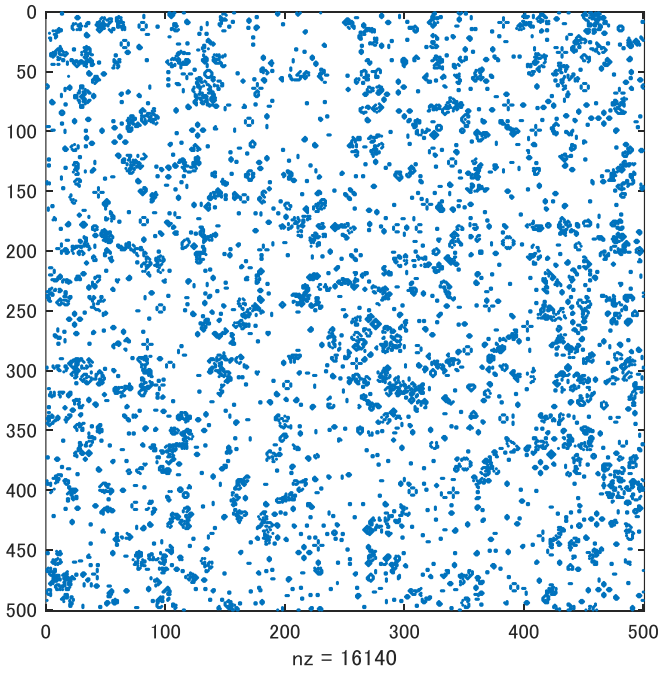
- 離散・・・バラバラなイメージ
- コンピュータが扱えるのは離散的なデータのみ

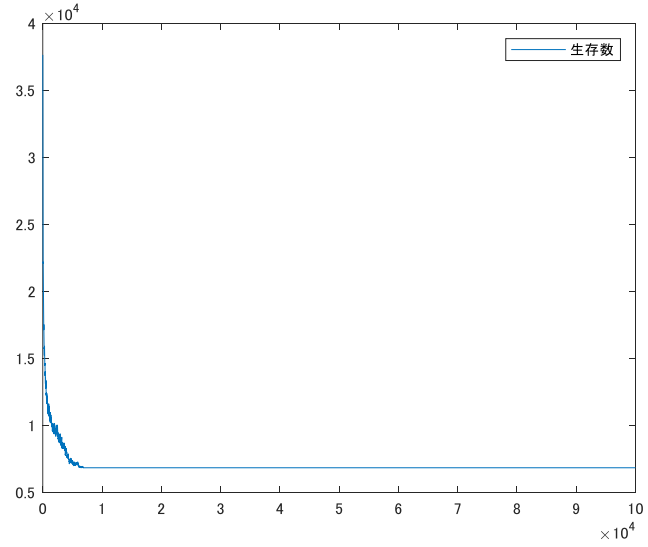


- 連続・・・滑らかに繋がっているイメージ
- 理論的，かつ，理想的なもの。コンピュータでは扱いづらい。

離散系シミュレーションの例・・・ライフゲーム

- 升目状に生物を配置
  - 周りに適切な数の生物が存在していれば生存 or 誕生
  - 過密すぎると死亡
  - 過疎過ぎても死亡
- １ステージ毎に，周囲の生物数をカウントし，生存，誕生 or 死亡を判定し，次のステージに進む
- 時間間隔が一定（離散的）なので，離散系の生物シミュレーションの一種
- セル・オートマトンの原型
- 生物数をプロットしていくと，連続的に変化しているように見えてくる





3. MATLABの操作方法

- 諸注意
  - 実習終了まで，PCは起動したまま！（再起動するとファイル全消去）
  - トラブル時はTAに救助を求めて下さい。

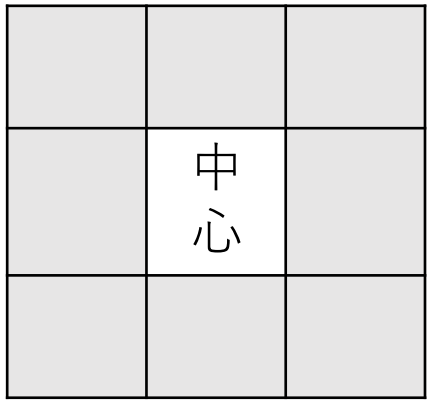
以下は口頭で解説しながら実演します。

- コマンドウィンドウの使い方
- エディタの使い方
- 作成物の保存方法

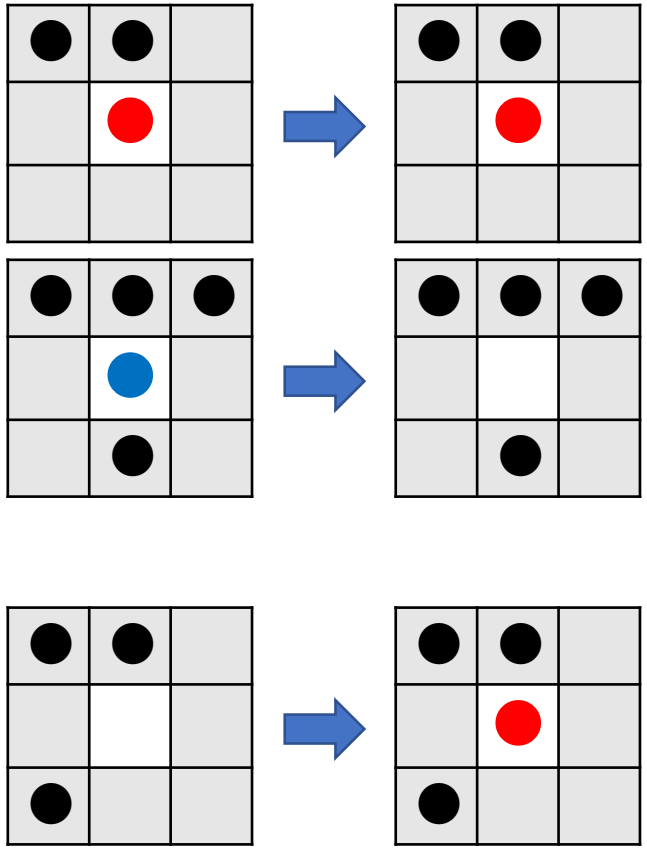
4. ライフゲーム体験

- ライフゲーム(life game)の規則
- MATLABによるライフゲーム
- 有名なパターン
- 大規模シミュレーション

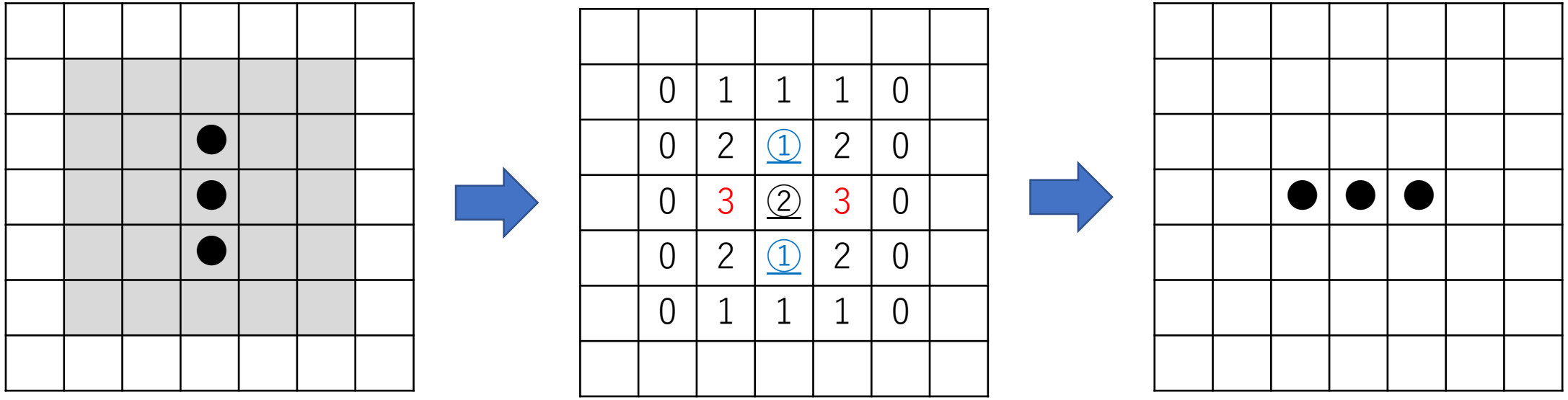
ライフゲーム(life game)の規則



- 平面に広がった升目に生物が存在しているとする
- 中心の升目を取り囲む８つの升目に存在している生物数(n)によって，次のステージにおける中心の升目の状態が変わる
- 中心に生物が存在している場合
  - ２ ≦ ｎ ≦ ３　→　生物は生存を続ける
  - ｎ ≦ １ →　過疎のため，生物は死亡（升目は空に）
  - ｎ ≧ ４ →　過密のため，生物は死亡
- 中心が生物が存在していない場合
  - ｎ＝３ → 生物誕生（升目に生物が生成される）



手動計算によるライフゲーム

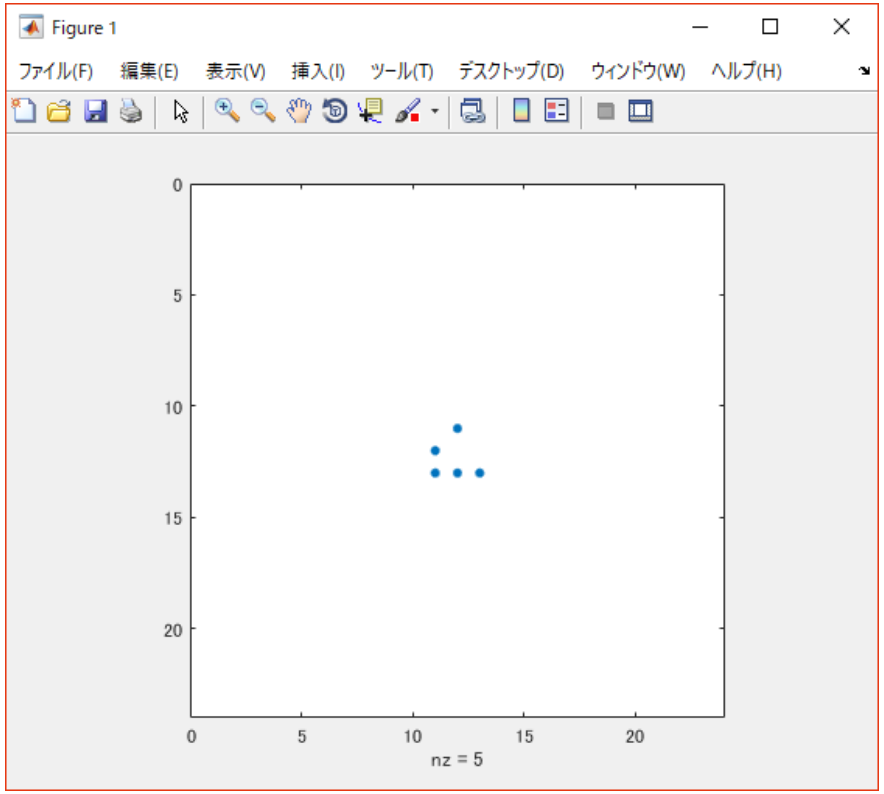


①盤面の端は無視（生物のいない壁とする）し，塗りつぶし部分を中心としたときの生物数をカウントする。

②カウントした生物数に基づいて，次のステージの状態を作る。

MATLABによるライフゲーム: lifegame1.m

- ssh_lifegame1.mとしてスクリプトを生成（右参照）
- 下記のようなウィンドウが開けば成功



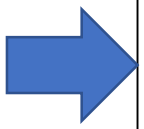

```
% 初期設定
G = [
     0, 1, 0;
     1, 0, 0;
     1, 1, 1
];

% 升目のサイズ
n = 23;

% 升目
X = sparse(n, n);

% XにGをセット
X(11:13, 11:13) = G;

% 盤面描画
spy(X);
```

lifegame1.mの続き

```
pause; % 一時停止

% Xの横サイズ( = n)を取得
sizeX = size(X, 1);

% 壁面をつなげる
p = [sizeX, 1:sizeX - 1];
q = [2:sizeX, 1];

% 周囲8マス分をカウント
Y = X(:, p)+ X(:, q)+ X(p, :)+ X(q, :)+ X(p, p)+ X(q, q)+ X(p, q)+ X(q, p);

% 周囲に2ないし3生物存在しているか？
X = (X & (Y == 2)) | (Y == 3);

% 状態プロット
spy(X)
```

ステージ数を増やす：lifegame2.m

```
% 周囲8マス分をカウント
Y = X(:, p) + X(:, q) +
X(p, :) + X(q, :) + X(p, p) +
X(q, q) + X(p, q) + X(q, p);

% 周囲に2ないし3生物存在しているか？
% → X(i, j)に生物を生成 or 生存継続
X = (X & (Y == 2)) | (Y == 3);

% 状態プロット
spy(X)
```

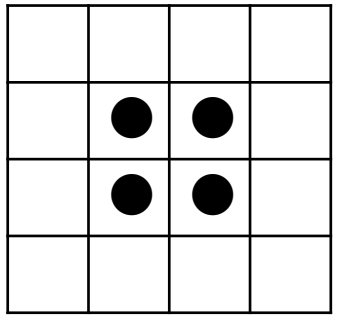


```
% メインループ
for year = 1:100
    % 周囲8マス分をカウント
      （同じ）

    % 周囲に2ないし3生物存在しているか？
      （同じ）

    % 状態プロット
      （同じ）
    drawnow;
    %pause;
end
```

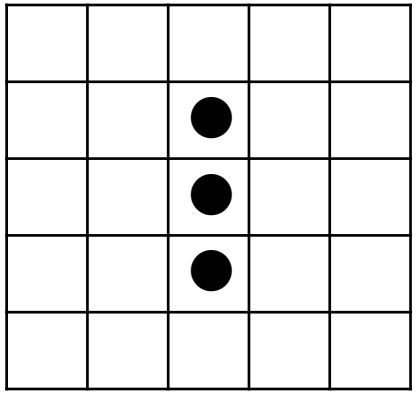
- ssh_lifegame1.mをssh_lifegame2.mとして別名保存
- 上記のように下線部分のみ変更
- 1世代ごとに状態を見たい時にはpauseの前の%を外す

有名なパターン

- 永久にそのまま：ブロック(block)

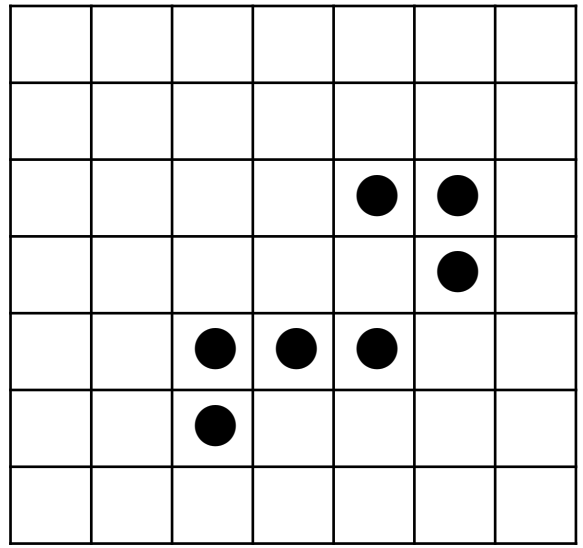


- 振動する：ブリンカー(blinker)



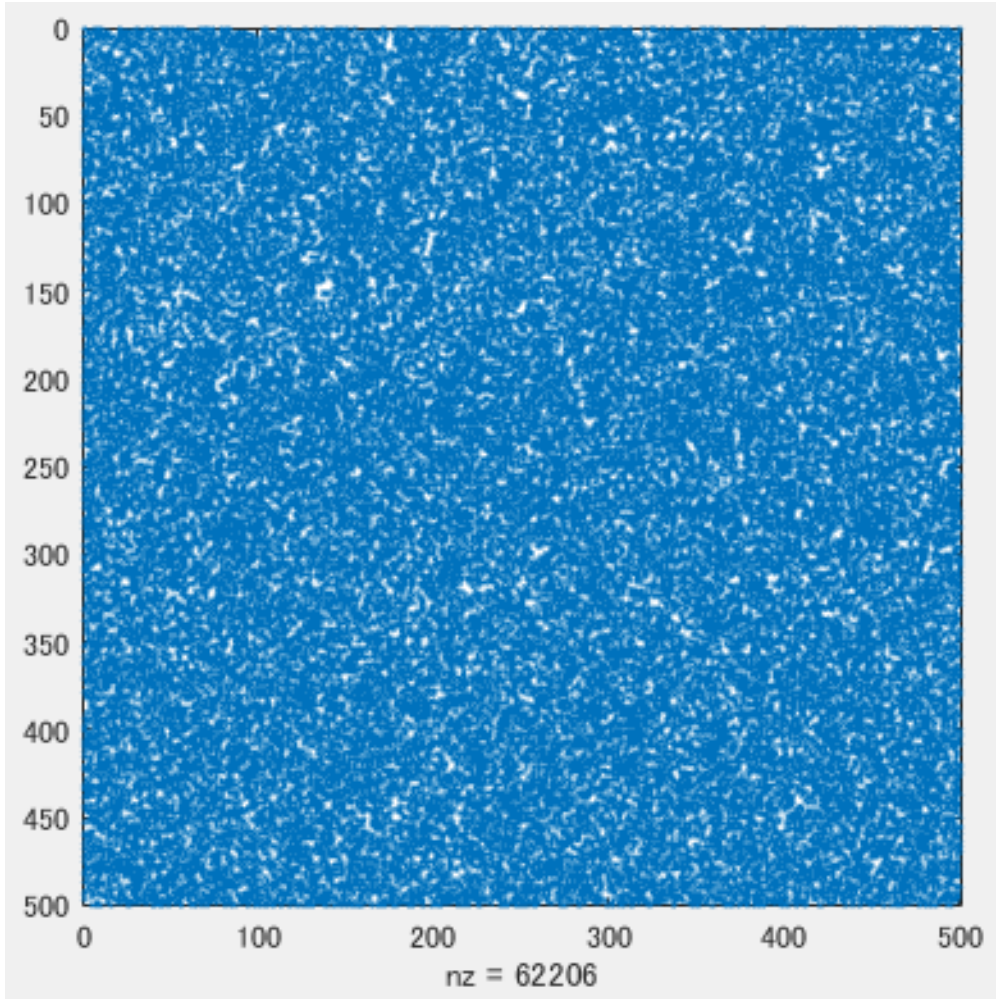
- 他のパターンを食べる：イーター(eater)

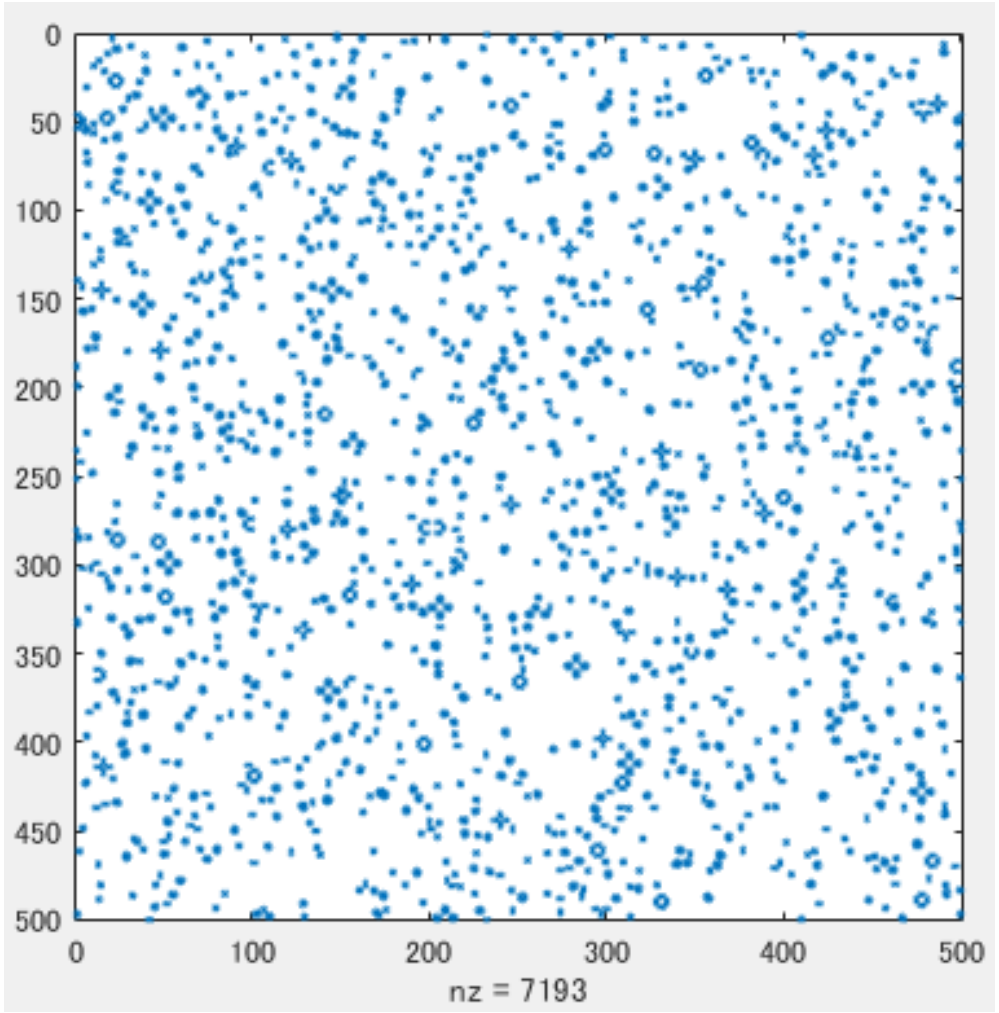
→ イーターの近くにブリンカーを配置すると？



大規模シミュレーション: ssh_lifegame3.m

- 生物数が一定数にとどまるまで実行してみる





lifegame3.m(1/2)

```
% 升目のサイズ
n = 500;

% 初期設定
initial_pattern = (rand(n, n) > 0.5);

% 升目
X = sparse(n, n);

% 初期状態を設定
X = initial_pattern;
```

```
% 初期状態を表示
spy(X);
pause; % 一時停止

% Xの横サイズ( = n)を取得
sizeX = size(X, 1);

% 壁面をつなげる
% p = sizeX, 1, 2, .., sizeX - 1
% q = 2, 3, ..., sizeX, 1
p = [sizeX, 1:sizeX - 1];
q = [2:sizeX, 1];

% 生物数を数える
num = [];
```

lifegame3.m (2/2)

```
% メインループ
for year = 1:1000
    num(year) = nnz(X);

    % 周囲8マス分をカウント
    Y = X(:, p)+X(:, q)+X(p, :)+X(q, :)+X(p, p)+X(q, q)+X(p, q)+X(q, p);

    % 周囲に2ないし3生物存在しているか？
    % → X(i, j)に生物を生成 or 生存継続
    X = (X & (Y == 2)) | (Y == 3);

    % 状態プロット
    spy(X);
    drawnow;
    %pause;
end
plot(num);
```

プログラムを変えてみましょう！

1. 初期設定を変えるとどうなるか？

   % 初期設定

   initial_pattern = (rand(n, n) > ここを0～1の間の数に設定);

2. ライフゲームの生存条件，死亡条件，誕生条件を変えるとどうなるか？

   X = (X & (Y == ここ)) | (Y == ここ);

3. 生物数をなるべく多く残すためにはどうすればいいか？

参考文献

- C.B.Moler, MATLABドキュメント
  - https://jp.mathworks.com/help/distcomp/examples/stencil-operations-on-a-gpu_ja_JP.html
  - https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/moler/exm/chapters/life.pdf

- W.Poundstone, 有澤誠・訳「ライフゲイムの宇宙」日本評論社