数理生物学演習

第10回　パターン形成

第10回：パターン形成

本日の目標

- 2次元配列
- 有限差分法による離散化
- 分子の拡散
- 濃度勾配モデル
- 反応拡散モデル

拡散方程式

熱伝導方程式

$$\frac{\partial u}{\partial t} = D \nabla^2 u$$

拡散が生じる分子などのダイナミクスを記述する
集団遺伝学で出てくることもある

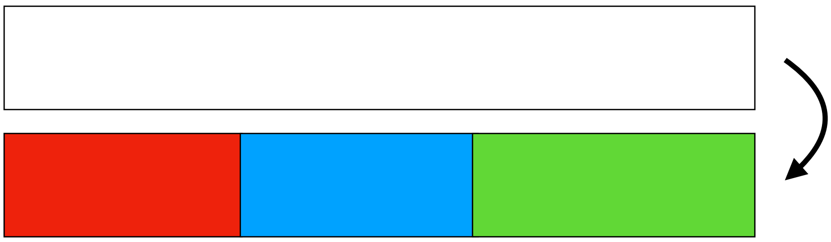
空間微分演算子

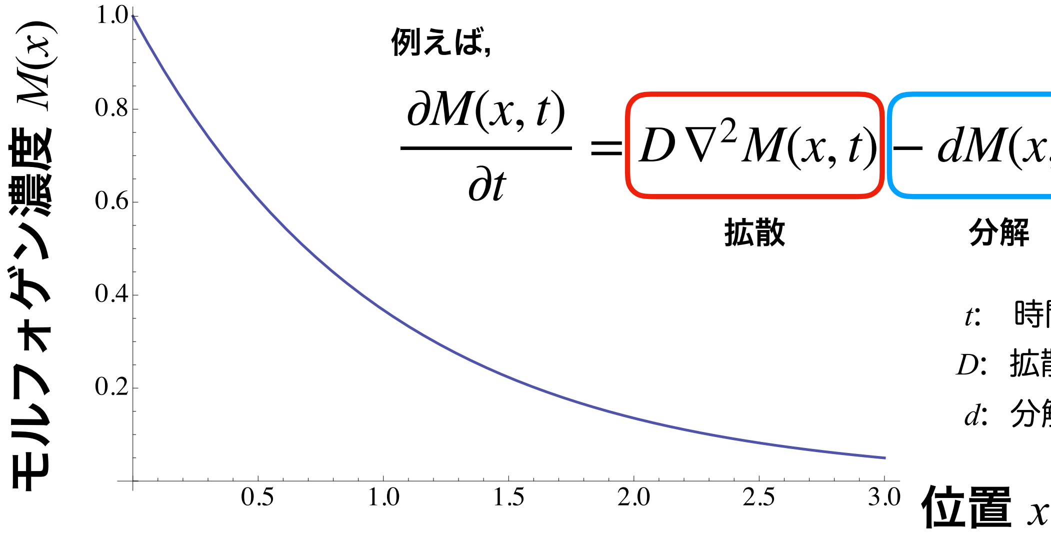
$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \cdots, \frac{\partial}{\partial x_n} \right)$$

スカラ量（例えば，拡散性分子の濃度）の勾配

$$\mathrm{grad}\, u = \nabla u = \frac{\partial u}{\partial x_1}\mathbf{e}_1 + \frac{\partial u}{\partial x_2}\mathbf{e}_2 + \cdots + \frac{\partial u}{\partial x_n}\mathbf{e}_n$$

モルフォゲンによるパターン形成



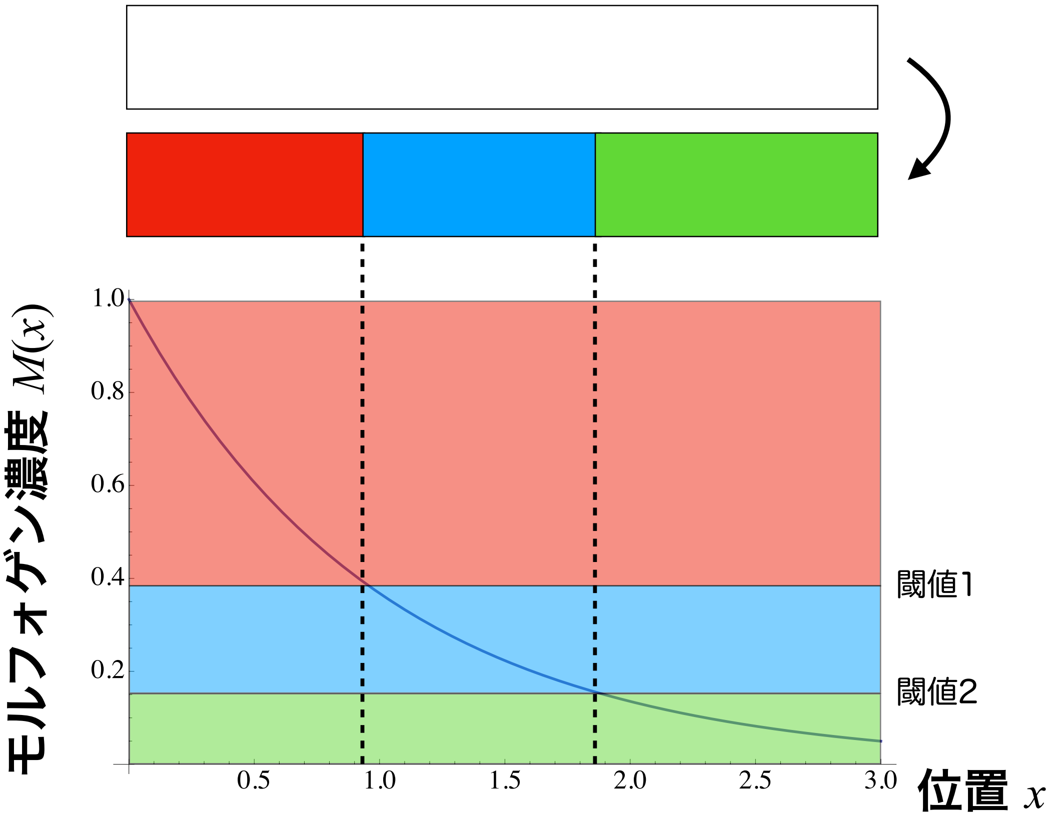


例えば,

$$\frac{\partial M(x,t)}{\partial t} = \underbrace{\boxed{D \nabla^2 M(x,t)}}_{\text{拡散}} \underbrace{\boxed{- dM(x,t)}}_{\text{分解}}$$

t: 時間
D: 拡散係数
d: 分解速度

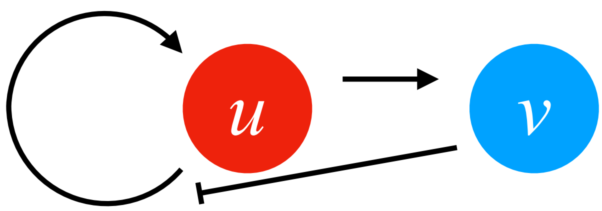
モルフォゲンによるパターン形成



反応拡散モデル ギーラー-マインハルト系

仮定

- uとvは共に拡散する
- uは自己活性化する
- uはvの合成を促進する
- vはuの合成を抑制する



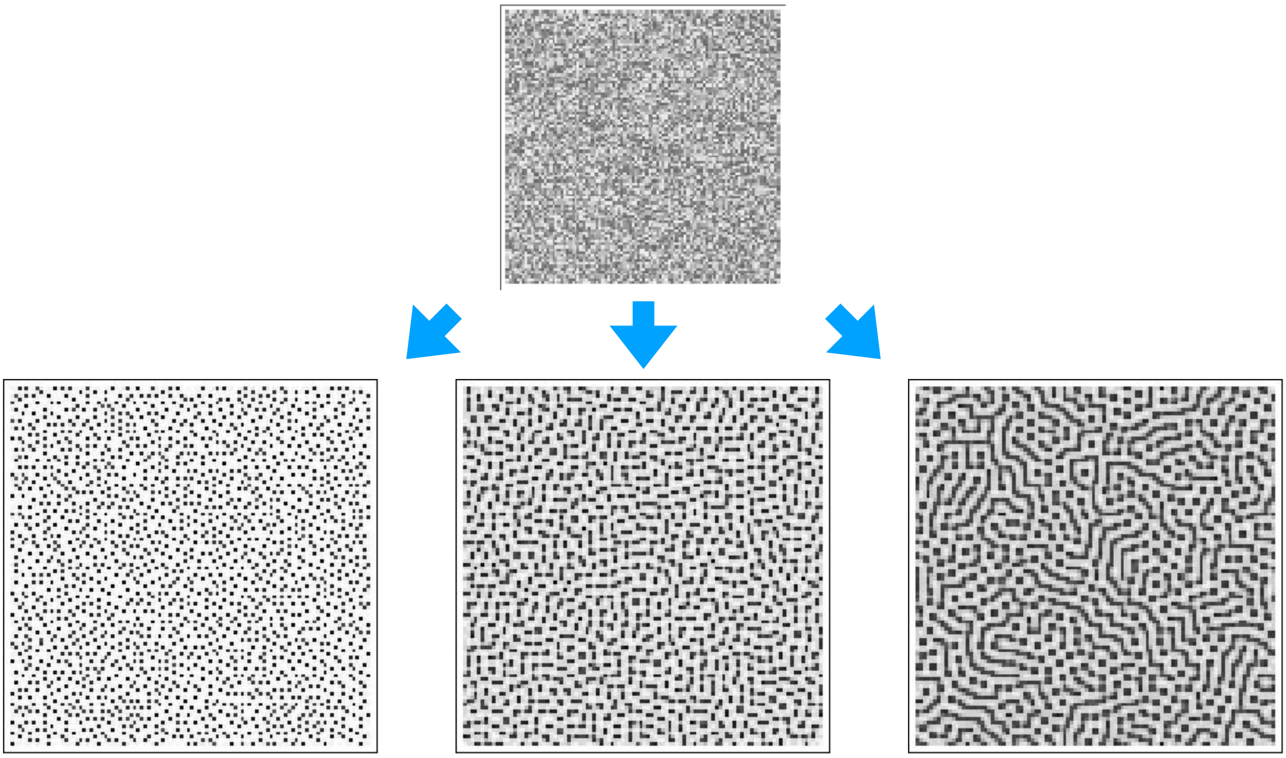
活性化因子
アクチベーター

抑制因子
インヒビター

$$
\begin{cases}
\dfrac{\partial u}{\partial t} = D_u \nabla^2 u + \dfrac{u^2}{v} - u \\
\dfrac{\partial v}{\partial t} = D_v \nabla^2 u + u^2 - v
\end{cases}
$$

チューリングパタン

ほぼ一様な構造のない状態から，自発的に空間的パタンが生じる



有限差分法による離散化

差分商により微分を近似することで，微分方程式を離散化する

差分を刻み幅で割ったもの

本演習では空間方向の離散化に用いる
時間方向へはこれまで通りオイラー法を使う

ある関数$u(x, y)$を2次元空間上で離散化する．
$u_{i,j}=u(x_i, y_j)$, (x_i, y_j)でのuのx方向への偏微分を$\left(\frac{\partial u}{\partial x}\right)_i$，$x$方向への刻み幅を$\Delta x$とすれば，

前進差分による近似

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}$$

オイラー法と同じ

中心差分による近似

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}$$

後退差分による近似

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

2階の中心差分による2階偏微分の近似

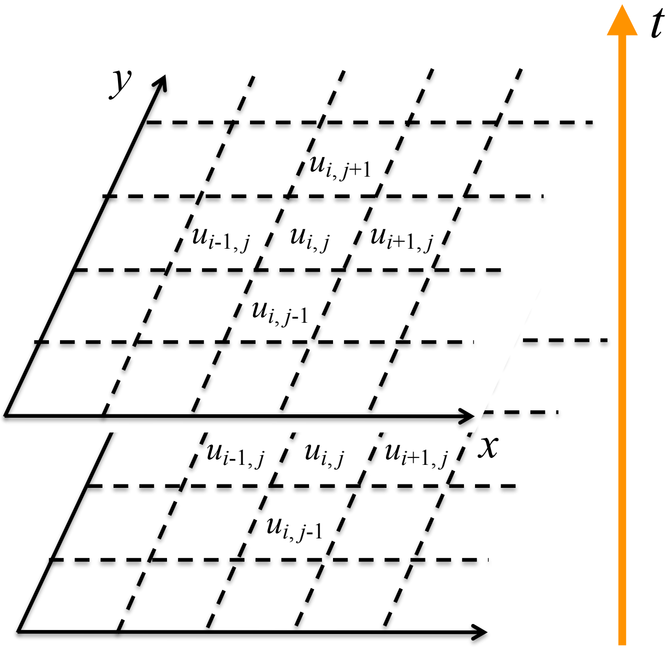
$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

導出については補足資料参照

y方向へも同様に考える

拡散方程式の離散化

- **時間方向の離散化**
  前進差分により近似
  いつものオイラー法
- **空間方向の離散化**
  2階の中心差分により近似



$$\frac{\partial u}{\partial t} = D\nabla^2 u$$

離散化

$$u_{i,j,t+\Delta t} = u_{i,j,t} + \frac{\Delta t}{h^2}\left[D(u_{i-1,j,t} + u_{i+1,j,t} + u_{i,j-1,t} + u_{i,j+1,t} - 4u_{i,j,t})\right]$$

導出してみてください

実際にプログラムを組んでみよう！

復習

配列

同じ型を持つ変数の集まり

型　配列名[配列サイズ];

型　配列1[サイズ], 配列2[サイズ],…, 配列n[サイズ];

たくさんの変数を個別に宣言するのは面倒！

```
4-1. 配列
#include <stdio.h>

int main(void){
  int i;
  int a[10];

  for(i=0;i<10;i++){
    a[i]=i;
  }

  for(i=0;i<10;i++){
    printf("%d\n",a[i]);
  }

  return 0;
}
```

i番目の要素にiを代入

- 配列のなかのそれぞれの変数を配列要素という

a[0], a[1], …, a[9]　―　配列

配列要素

- 各要素へは添字によってアクセスする

特に注意！！

添字は0から始まり，(サイズ-1)で終わる

int a[10];　で定義したならば,

a[0]〜a[9]までの要素が存在する

復習

２次元配列

型　配列名[配列サイズ][配列サイズ];

型　配列1[サイズ][サイズ], 配列2[サイズ][サイズ],
　　…, 配列n[サイズ][サイズ];

a[0][0], a[0][1], …, a[0][n]

a[1][0], a[1][1], …, a[1][n]

a[2][0], a[2][1], …, a[2][n]

…

a[m][0], a[m][1], …, a[m][n]

添字2, 1の
配列要素

2次元配列

基本は1次元の場合と同じ.

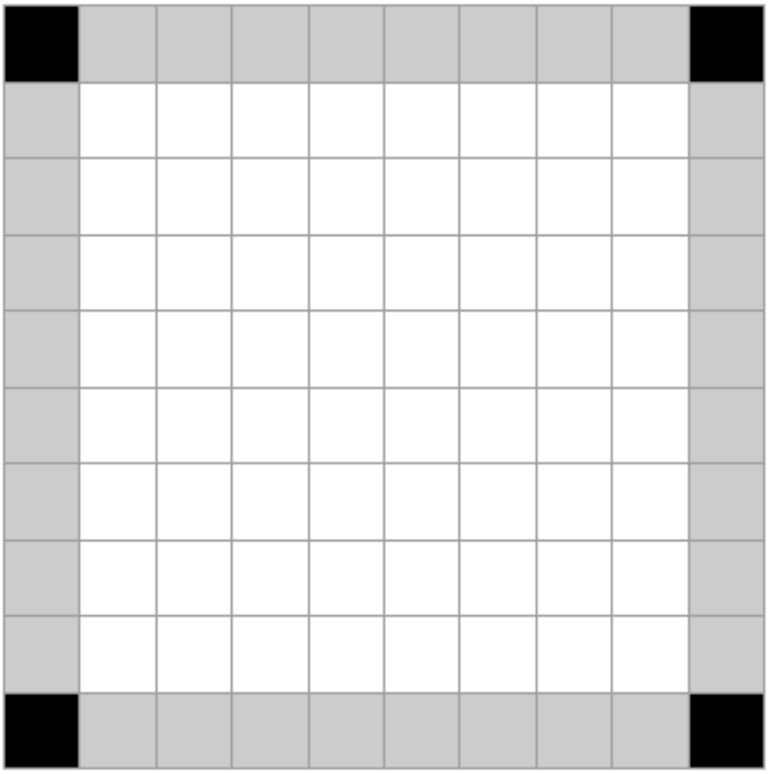
より多次元の配列も定義できます！

```
10-1. 2次元配列
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void){
  int i,j;
  int a[10][10];

  srand(time(NULL));

  for(i=0;i<10;i++){
    for(j=0;j<10;j++){
      a[i][j]=rand()%10;
    }
  }

  for(i=0;i<10;i++){
    for(j=0;j<10;j++){
      printf("%d",a[i][j]);
      if(j!=9){
        printf(", ");
      }
    }
    printf("¥n");
  }

  return 0;
}
```

11-1. 2次元の拡散方程式

```
#include <stdio.h>

int main(){
  int i,j;
  int t;
  double dt=0.01;
  double u[100][100];
  double utemp[100][100];

  double D=0.4;

  FILE *fp;

  fp=fopen(“11-1.csv","w");

  //初期化
  for(i=0;i<100;i++){
    for(j=0;j<100;j++){
      u[i][j]=0;
    }
  }

  u[49][49]=1;
  u[49][50]=1;
  u[50][49]=1;
  u[50][50]=1;


  //初期値出力
  for(i=0;i<100;i++){
    for(j=0;j<100;j++){
      fprintf(fp,"%f",u[i][j]);
      if(j!=99){
  fprintf(fp," ,");
      }
    }
    fprintf(fp,"\n");
  }

  for(t=1;t<5000;t++){

  //境界条件
   //i=0,j=0
    i=0;
    j=0;
    utemp[i][j]=u[i][j]+dt*(D*(u[99][j]
+u[i+1][j]+u[i][99]+u[i][j+1]-4*u[i]
[j]));
    //
i=0,j=99


    i=0;
    j=99;
    utemp[i][j]=u[i][j]+dt*(D*(u[99][j]
+u[i+1][j]+u[i][j-1]+u[i][0]-4*u[i]
[j]));
    //
i=99,j=0


    i=99;
    j=0;
    utemp[i][j]=u[i][j]+dt*(D*(u[i-1][j]
+u[0][j]+u[i][99]+u[i][j+1]-4*u[i][j]));
    //i=99,j=99
    i=99;
    j=99;
    utemp[i][j]=u[i][j]+dt*(D*(u[i-1][j]
+u[0][j]+u[i][j-1]+u[i][0]-4*u[i][j]));

    //i=0
    i=0;
    for(j=1;j<99;j++){
      utemp[i][j]=u[i][j]+dt*(D*(u[99]
[j]+u[i+1][j]+u[i][j-1]+u[i][j+1]-4*u[i]
[j]));
    }
    //i=99
    i=99;
    for(j=1;j<99;j++){
      utemp[i][j]=u[i][j]+dt*(D*(u[i-1]
[j]+u[0][j]+u[i][j-1]+u[i][j+1]-4*u[i]
[j]));
    }
    //j=0
    j=0;
    for(i=1;i<99;i++){
      utemp[i][j]=u[i][j]+dt*(D*(u[i-1]
[j]+u[i+1][j]+u[i][99]+u[i][j+1]-4*u[i]
[j]));
    }
    //j=99
    j=99;
    for(i=1;i<99;i++){
      utemp[i][j]=u[i][j]+dt*(D*(u[i-1]
[j]+u[i+1][j]+u[i][j-1]+u[i][0]-4*u[i]
[j]));
    }
    //その他
    for(i=1;i<99;i++){
      for(j=1;j<99;j++){
  utemp[i][j]=u[i][j]+dt*(D*(u[i-1][j]
+u[i+1][j]+u[i][j-1]+u[i][j+1]-4*u[i]
[j]));
      }
    }
    //更新
    for(i=0;i<100;i++){
      for(j=0;j<100;j++){
  u[i][j]=utemp[i][j];
      }
    }

    //出力
    if(t%500==0){
      for(i=0;i<100;i++){
  for(j=0;j<100;j++){
    fprintf(fp,"%f",u[i][j]);
    if(j!=99){
      fprintf(fp," ,");
    }
  }
  fprintf(fp,"\n");
      }
    }

  }
  fclose(fp);
  return 0;
}
```

2次元空間での境界条件



反応拡散モデル

11-2. ギーラー-マインハルト系の反応拡散モデルについてプログラムを組み，様々なパタンを描く

方針

1. モデルの離散化

アクチベーター
インヒビター

$$\begin{cases} \dfrac{\partial u}{\partial t} = D_u \nabla^2 u + \dfrac{u^2}{v} - u \\ \dfrac{\partial v}{\partial t} = D_v \nabla^2 u + u^2 - v \end{cases} \xrightarrow{\text{離散化}} \ ?$$

2. 2次元拡散方程式を参考にプログラムを組む
基本的には，拡散方程式を反応拡散方程式系に変えるだけ．
ただし，拡散性分子が2種類あることに注意．

3. 初期値の設定
u=1.0, v=1.0にわずかなノイズ
(0.0〜0.01程度)を加える．

4. 拡散係数（Du, Dv）を変化させて
どのようなパタンが生じるか調べる

注意：出力を増やすとファイルが重くなる．
ifを使い，ある程度間隔を空けて出力すること．

本日の課題

ノーマル：1つ選ぶ

ハード：両方

1. 反応拡散系のパラメータや初期値を変化させた様々なパタンを観察せよ．また，どういった傾向があるかを考察せよ．
2. 反応拡散系のパラメータや初期値を変化させて生物の体表面に観察される模様を幾つか再現せよ．また，それはどういった生物にみられるか例を挙げよ．
3. 質問，意見，要望等をどうぞ．

課題をPDFファイルにまとめて，Google フォームにて提出すること

次回予告

第11回：アレルギー治療法の数理モデル

7月2日

復習推奨

- 微分方程式の数値計算（第3回の内容等）