カオス現象のシミュレーション

講師：高安 亮紀

第7回

目的

- ローレンツ方程式を数値計算で解く。
- 4次のルンゲ･クッタ法をローレンツ方程式に適用する。MATLABでプログラムを作成し、その解の軌道を確認する。
- 一般的なm元連立常微分方程式を解けるように拡張する。

ローレンツ方程式

- カオス的なふるまいを示す非線形方程式
- 大気変動を簡略化・数理モデル化したもの
- システムのふるまいは3つの定数p,r,bによって決まる

$$\begin{cases} \dfrac{dx}{dt} = -px + py \\ \dfrac{dy}{dt} = -xz + rx - y \\ \dfrac{dz}{dt} = xy - bz \end{cases}$$

ここでは

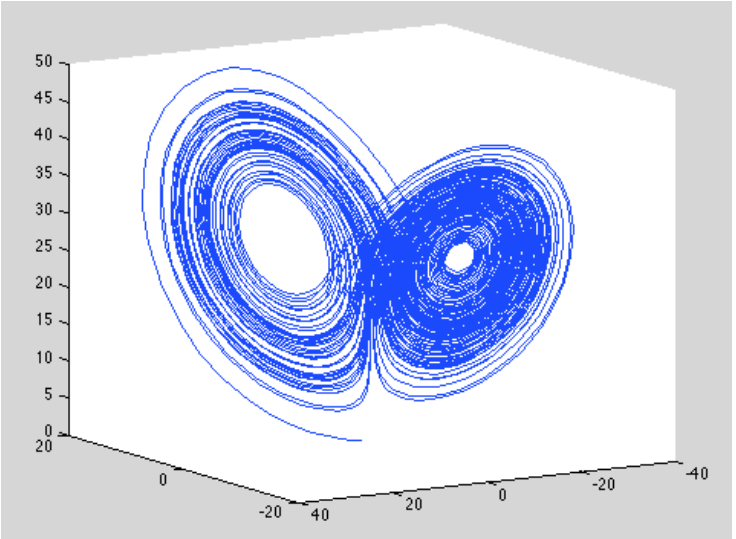
$$\begin{cases} p = 10 \\ r = 28 \\ b = 8/3 \end{cases}$$

とする。

まずは，ODEソルバでローレンツ方程式を解く

lorenz1.m

```
function dy = lorenz1(t,y)
p = 10; %初期値
r = 28;
b = 8/3;
dy = zeros(3,1);
dy(1) = -p*y(1) + p*y(2);
dy(2) = -y(1)*y(3) + r*y(1) - y(2);
dy(3) = y(1)*y(2) - b*y(3);
```




```
>> [t,y]=ode45('lorenz1', [0 100], [1;1;1]);
>> plot3(y(:,1),y(:,2),y(:,3))
```

常微分方程式の数値解法

- コンピュータを用いて数値的に微分方程式

$$\frac{dy}{dx} = f(x, y)$$

の解を求める。

- 本講義では、MATLABのODEソルバなどの標準的な手法になっているルンゲ・クッタ法を用いる。

ルンゲ・クッタ法（１変数）

- 微分方程式の数値解法として、4次のルンゲ・クッタ法を使用。
- 基本的な考え方はオイラー法と同じ。

$$k_1 = f(x_i, y_i)$$

h: 刻み幅

$$k_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1)$$

$$k_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2)$$

$$k_4 = f(x_i + h, y_i + hk_3)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$x_{i+1} = x_i + h$$

ルンゲ・クッタ法でローレンツ方程式を解く（1）

下記のように，関数 f_1, f_2, f_3 を定める.

$$\begin{cases} f_1(y_1, y_2) = -10y_1 + 10y_2 \\ f_2(y_1, y_2, y_3) = -y_1y_3 + 28y_1 - y_2 \\ f_3(y_1, y_2, y_3) = y_1y_2 - 8/3\, y_3 \end{cases}$$

多変数版のルンゲ・クッタ法を適用して解y_1, y_2, y_3を求める。

ルンゲ・クッタ法でローレンツ方程式を解く（２）

時間tの増分hに対するy_1の増分をk、同様にy_2の増分をl、y_3の増分をmとすると次の関係式が成り立つ。

$$\begin{cases} t_{i+1} = t_i + h \\ y_1^{(i+1)} = y_1^{(i)} + k \\ y_2^{(i+1)} = y_2^{(i)} + l \\ y_3^{(i+1)} = y_3^{(i)} + m \end{cases}$$

$$\begin{cases} k = \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ l = \dfrac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \\ m = \dfrac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \end{cases}$$

ルンゲ・クッタ法でローレンツ方程式を解く（３）

また、$k_{1\sim4}$、$l_{1\sim4}$、$m_{1\sim4}$は次のように与えられる。
これらの式からy_1、y_2、y_3を求めることができる。

$$\begin{cases} k_1 = hf_1(y_1^{(i)}, y_2^{(i)}) \\ l_1 = hf_2(y_1^{(i)}, y_2^{(i)}, y_3^{(i)}) \\ m_1 = hf_3(y_1^{(i)}, y_2^{(i)}, y_3^{(i)}) \end{cases}$$

$$\begin{cases} k_2 = hf_1(y_1^{(i)} + \frac{k_1}{2}, y_2^{(i)} + \frac{l_1}{2}) \\ l_2 = hf_2(y_1^{(i)} + \frac{k_1}{2}, y_2^{(i)} + \frac{l_1}{2}, y_3^{(i)} + \frac{m_1}{2}) \\ m_2 = hf_3(y_1^{(i)} + \frac{k_1}{2}, y_2^{(i)} + \frac{l_1}{2}, y_3^{(i)} + \frac{m_1}{2}) \end{cases}$$

$$\begin{cases} k_3 = hf_1(y_1^{(i)} + \frac{k_2}{2}, y_2^{(i)} + \frac{l_2}{2}) \\ l_3 = hf_2(y_1^{(i)} + \frac{k_2}{2}, y_2^{(i)} + \frac{l_2}{2}, y_3^{(i)} + \frac{m_2}{2}) \\ m_3 = hf_3(y_1^{(i)} + \frac{k_2}{2}, y_2^{(i)} + \frac{l_2}{2}, y_3^{(i)} + \frac{m_2}{2}) \end{cases}$$

$$\begin{cases} k_4 = hf_1(y_1^{(i)} + k_3, y_2^{(i)} + l_3) \\ l_4 = hf_2(y_1^{(i)} + k_3, y_2^{(i)} + l_3, y_3^{(i)} + m_3) \\ m_4 = hf_3(y_1^{(i)} + k_3, y_2^{(i)} + l_3, y_3^{(i)} + m_3) \end{cases}$$

手順のまとめ

- 初期条件 y_0 を与える。
- 時間 t の刻み幅 h を決める。（hが小さいほど解の精度は高くなる）
- h に対する y_1 の増分を k、y_2 の増分を l、y_3 の増分を m とする。
- i ステップ目の y と h から $k_{1\sim4}$、$l_{1\sim4}$、$m_{1\sim4}$ をそれぞれ求め、(i+1)ステップ目の y を求める。

実習

- 自分でルンゲクッタ法をMATLABで実装し、解の軌道を描いてみよう（3次元の図の描画には、plot3を使う）。
- わからない場合は、次ページのスライドを参考にしよう。

lorenz2.m

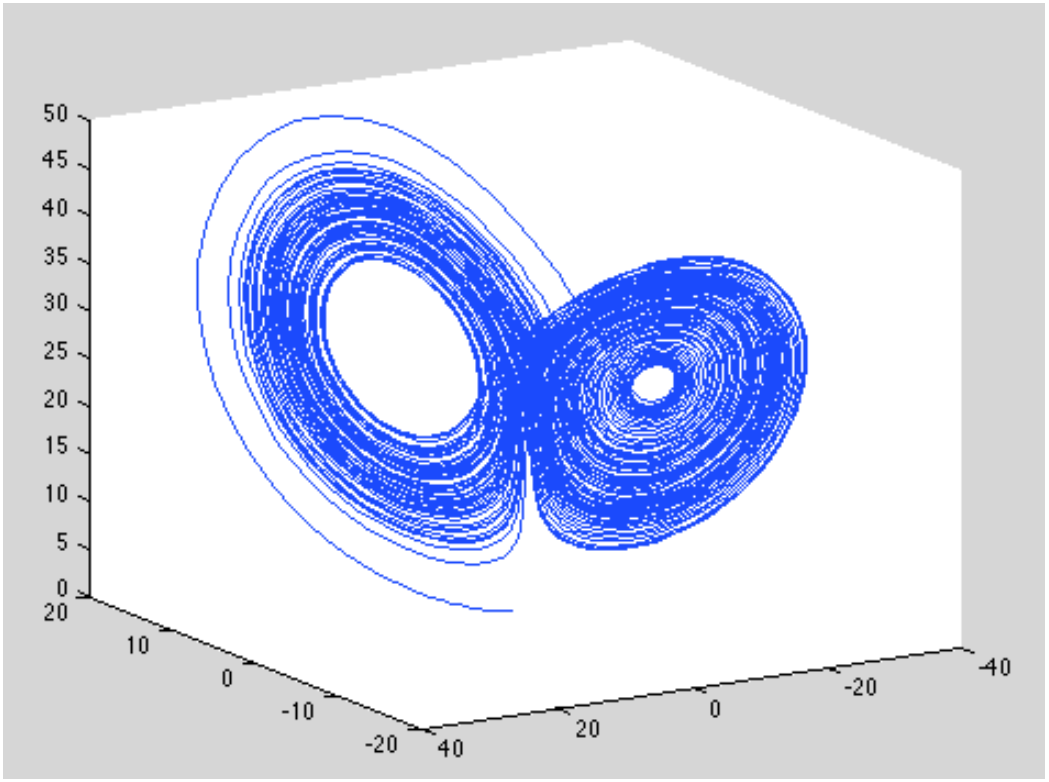
```
function [t,y1,y2,y3] = lorenz2(f1,f2,f3,y0,ts,n)
% f1, f2, f3: 関数
% n: 総ステップ数
% y0: y1, y2, y3の初期値ベクトル
% ts = [tstart,tend]

h = (ts(2) - ts(1))/(n - 1); % 時間の刻み幅
t = zeros(n,1); % 時間
t(1) = ts(1); % 時間の初期値
y1 = zeros(n,1);
y2 = zeros(n,1);
y3 = zeros(n,1);
y1(1) = y0(1); % y1の初期値
y2(1) = y0(2); % y2の初期値
y3(1) = y0(3); % y3の初期値
```

（右につづく）

```
% ルンゲクッタ法
for i = 1:n-1
   k1 = h*f1(y1(i),y2(i));
   l1 = h*f2(y1(i),y2(i),y3(i));
   m1 = h*f3(y1(i),y2(i),y3(i));
   s1 = y1(i) + k1/2; s2 = y2(i) + l1/2; s3 = y3(i) + m1/2;
   k2 = h*f1(s1,s2);
   l2 = h*f2(s1,s2,s3);
   m2 = h*f3(s1,s2,s3);
   s1 = y1(i) + k2/2; s2 = y2(i) + l2/2; s3 = y3(i) + m2/2;
   k3 = h*f1(s1,s2);
   l3 = h*f2(s1,s2,s3);
   m3 = h*f3(s1,s2,s3);
   s1 = y1(i) + k3; s2 = y2(i) + l3; s3 = y3(i) + m3;
   k4 = h*f1(s1,s2);
   l4 = h*f2(s1,s2,s3);
   m4 = h*f3(s1,s2,s3);
   y1(i+1) = y1(i) + (k1 + 2*k2 + 2*k3 + k4)/6;
   y2(i+1) = y2(i) + (l1 + 2*l2 + 2*l3 + l4)/6;
   y3(i+1) = y3(i) + (m1 + 2*m2 + 2*m3 + m4)/6;
   t(i+1) = t(i) + h;
end
```

```
>> f1 = inline('-10*y1 + 10*y2');
>> f2 = inline('-y1*y3 + 28*y1 - y2');
>> f3 = inline('y1*y2 - (8/3)*y3');
>> n = 10000; [t,Y1,Y2,Y3] = lorenz2(f1,f2,f3,[1 1 1],[0 100],n);
>> plot3(Y1,Y2,Y3)
```



(1) nや初期値y0の値を変えたらどうなるかを確認してみよう。
(2) 解の軌跡をアニメーションで描くように変更してみよう。

応用問題

- ルンゲ･クッタ法で一般的なm元連立常微分方程式を解けるように拡張しよう。
- ヒント
  - 関数fは、下記のように定義することを想定する。

```
f = inline('[-10*y(1) + 10*y(2), -y(1)*y(3) + 28*y(1) - y(2), y(1)*y(2) - (8/3)*y(3)]','y')
```

このようにすると、ベクトル y に対して f(y) としたとき、結果もベクトルとなる。

  - 他の演算もベクトル化する。