HOKKAIDO UNIVERSITY

Title	2009年度 カオス・フラクタル 講義ノート
Author(s)	井上, 純一
Relation	2009年度前期に工学部情報工学コース3年生を対象に開講された講義「カオス・フラクタル」の講義資料
Issue Date	2009
Doc URL	https://hdl.handle.net/2115/38906
Rights(URL)	https://creativecommons.org/licenses/by-nc-sa/2.1/jp/
Type	learning object
File Information	CF2009_8.pdf, 第8回講義ノート

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

カオス・フラクタル 講義ノート #8

担当： 井上 純一 (情報科学研究科棟 8-13)

URL：http://chaosweb.complex.eng.hokudai.ac.jp/˜j _inoue/index.html

平成 21 年 6 月 16 日

目 次

9 アトラクタの埋め込み次元と相関次元 **57**

9.1 埋め込み次元とアトラクタの再構成 57

9.2 次元再考 57

9.3 相関次元 59

9.4 相関次元の一般化 62

先週の復習として, レポート課題を確認しておこう. $a = 10, b = 8/3, c = 30$ と選んだ場合のローレンツ方程式の $z = 20, 30, 40$ におけるポアンカレ断面を図 35 に載せる. また, このグラフを作成するには

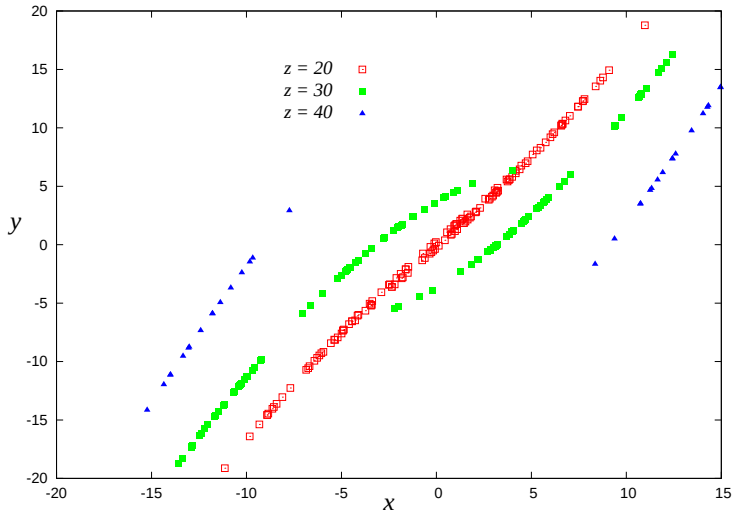


図 35: $a = 10, b = 8/3, c = 30$ と選んだ場合のローレンツ方程式の $z = 20, 30, 40$ におけるポアンカレ断面.

```
x = x + kx;
y = y + ky;
z = z + kz;
```

```
arg = fabs(z-20);
if((i>=iSkip) && (arg <5.0e-6)){
fprintf(fpr,"%lf %lf\n",x,y);
}
```

のように, $z = 20$ をある精度で満たす (x, y) のみをファイルに書き込むようにすればよい.

9 アトラクタの埋め込み次元と相関次元

前回までの講義ではレスラー方程式やローレンツ方程式などの非線形微分方程式の数値解が奇妙なアトラクタを持つことをみたが, レスラー方程式は x-y 平面内に強く引きつけられることで 3 次元というよりももしろ 2 次元的な構造を持っている. そこで, ここではこの奇妙なアトラクタの実質的な次元を評価するためのいくつかの方法をみていく.

9.1 埋め込み次元とアトラクタの再構成

レスラー方程式:

$$\frac{dx}{dt} = -y - z \tag{157}$$

$$\frac{dy}{dt} = x + ay \tag{158}$$

$$\frac{dz}{dt} = b + z(x - c) \tag{159}$$

を数値的に解いた後, 例えば $x(t)$ のみに着目し, 適当な時間遅れ τ を導入し, 次のような m 次元ベクトルを作る.

$$(x(t), x(t+\tau), \cdots, x(t+(m-1)\tau)) \tag{160}$$

そこで, この m 次元ベクトルを逐次プロットすることで, 前回みた奇妙なアトラクタが再構成できるかを調べてみよう. 図 36 に次元を $m = 2$ と選んだ場合の $(x(t), x(t+\tau)), (y(t), y(t+\tau)), (z(t), z(t+\tau))$ をぞれぞれプロットした. この図より, x のみ, y, z のみの軌道を適切な時間遅れ τ, 埋め込み次元 m で切り出すと, x, y, z を 3 次元にプロットした場合とほぼ同じアトラクタを再構成することができる. この例では $m = 2$, $\tau = 1, 2$ に選んでいるが, 与えられたデータが方程式からの時系列ではなく性質が不明なものである場合, これら 2 つのパラメータを適切に決定しなければならない. 時間遅れ τ は $x(t)$ と $x(t+\tau)$ の自己相関関数, あるいは, 相互情報量を評価することで見積もることができるが, ここではそれには触れない. 次元 m に関しては次に述べる「相関次元」がその指針を与える. そこで今回の講義では, 相関次元の計算を見ていく. その準備として, まずは「次元」について確認しておこう.

9.2 次元再考

空間の次元に関し, 我々は直線が 1 次元, 平面が 2 次元等を知っている. しかし, ここでは任意の対象の次元を少し変わった方法で定義する. この方法は後に「フラクタル」を学ぶ際にも重要である.

まず, 半径が ϵ の球 (円) で問題とする対象物を覆うことを考えよう. このとき, 対象を隙間無く覆うために必要な球の個数を $n(\epsilon)$ とおく. この $n(\epsilon)$ を用いて, この対象物の次元を次式で定義する.

$$D_0 = -\lim_{\epsilon \to 0} \frac{\log n(\epsilon)}{\log \epsilon} \tag{161}$$

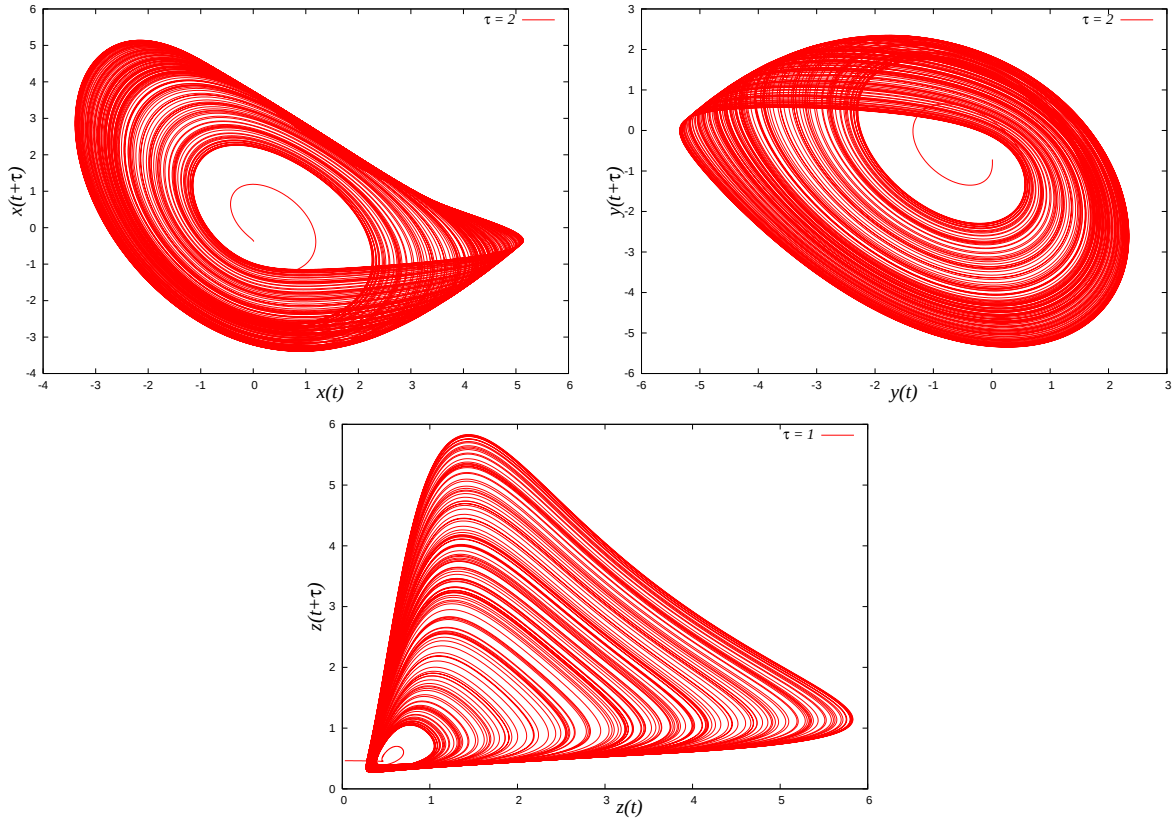


図 36: 再構成. 埋め込み次元を $m = 2$ に選んでいる.

この次元 — **容量次元**と呼ばれる— が我々が経験的に知っている次元に反しないかを確認するため, 長さ l の線分の次元 D_0 を計算してみよう.

まず, 長さ l の線分を隙間無く半径 ϵ の球で覆うためには $l/2\epsilon$ 個の球が必要であるから, $n(\epsilon) = l/2\epsilon$ である. 従って

$$D_c(\text{長さ } l \text{ の線分}) \quad = \quad -\lim_{\epsilon\to 0}\frac{\log(l/2\epsilon)}{\log\epsilon} = -\lim_{\epsilon\to 0}\left\{\frac{\log(l/2)}{\log\epsilon}\right\} + \frac{\log\epsilon}{\log\epsilon} = 1 \tag{162}$$

となり, 線分の次元は 1 であるという事実に合致する.

また, 1 辺の長さが l の正方形に対しては同様にしてこれを覆うための球の個数は $n(\epsilon) = l^2/\pi\epsilon^2$ であるから

$$D_c(\text{一辺の長さが } l \text{ の正方形}) \quad = \quad -\lim_{\epsilon\to 0}\frac{\log(l^2/\pi\epsilon^2)}{\log\epsilon} = -\lim_{\epsilon\to 0}\left\{\frac{\log(l^2/\pi)}{\log\epsilon}\right\} + 2\frac{\log\epsilon}{\log\epsilon} = 2 \tag{163}$$

となり, やはり我々の経験事実に合う. 従って, この方法を用いることで任意の図形の次元を計算することができる. 既にみた奇妙なアトラクタの次元もこの考え方を拡張することで算出することができる.

ここで述べた 2 つの例はほとんど自明な場合であったのだが, 海岸線など複雑に入り組んだ図形の次元を上記の方法で見積もると $1, 2, 3, \cdots$ のような整数次元ではなく, 非整数の次元を持つことがわかっている. これについては再来週からの [フラクタル編] で詳しくみていくことになる.

9.3 相関次元

前節で見たように, ある力学系から生成されたデータ列 $\{x(t_i)\}$: $i = 1, 2, \cdots, N$ を時間遅れ τ でサンプリングし, 次の m 次元のベクトル:

$$\boldsymbol{r}_i \equiv (x(t_i), x(t_i+\tau), \cdots, x(t_i+(m-1)\tau)), \quad i = 1, \cdots, N \tag{164}$$

を考える. このとき, $\boldsymbol{r}_i$ を中心とする半径 ϵ の球を考え, この球内にアトラクタ上の点 $\boldsymbol{r}_j$: $j = 1, \cdots, N$ が入る確率 p_i を考えると

$$p_i = \frac{1}{N}\sum_{j=1}^{N}\Theta(\epsilon - |\boldsymbol{r}_i - \boldsymbol{r}_j|) \tag{165}$$

となる. ここで, $\Theta(x)$ は階段関数 (ステップ関数) と呼ばれる関数であり, $x \geq 0$ で 1, $x < 0$ でゼロを返す. また, $|\boldsymbol{r}_i - \boldsymbol{r}_j|$ を陽な形で書き出しておくと

$$|\boldsymbol{r}_i - \boldsymbol{r}_j| = |x(t_i) - x(t_j)| + |x(t_i+\tau) - x(t_j+\tau)| + \cdots + |x(t_i+(m-1)\tau) - x(t_j+(m-1)\tau)| \tag{166}$$

であることに注意されたい. このとき**相関次元**を

$$D_2 = \lim_{\epsilon\to 0}\frac{\sum_{i=1}^{N} p_i^2}{\log\epsilon} \tag{167}$$

で定義する. ここで

$$\sum_{i=1}^{N} p_i^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\Theta(\epsilon - |\boldsymbol{r}_i - \boldsymbol{r}_j|) \equiv C(\epsilon) \tag{168}$$

に注意すると相関次元 (167) は $C(\epsilon)$ を用いて次のように書き直せる.

$$D_2 = \lim_{\epsilon\to 0}\frac{\log C(\epsilon)}{\log\epsilon} \tag{169}$$

よって, ϵ に関する極限をひとまず無視すれば, $D_2 = \log C(\epsilon)/\log\epsilon$, すなわち

$$C(\epsilon) = \epsilon^{D_2} \tag{170}$$

が成り立つ. 従って (168) 式に従って ϵ の変化に対する $C(\epsilon)$ を算出し, 得られる結果を両対数プロットした際のグラフの傾きが相関次元 D_2 を与える.

レスラー方程式において $x(t)$ から時間遅れ $\tau = 0.01 \times 200 = 2$ のもとで $m = 1, 2, 3, 4, 5$ 次元のベクトル $(x(t_i), x(t_i+\tau), \cdots, x(t_i+(m-1)\tau)) : i = 1, \cdots, N$ を作成し, それから上記レシピに従って相関次元を求めた結果を図 37 載せよう. この図より, m を増加するにつれ, グラフの傾きは実質的なアトラクタの次元である $D_2 = 2$ に近づくことがわかる. この方法はレスラー方程式からの軌道にみならず, 何らかの方法で実測されたデータ, もしくは背後の力学系や方程式が未知である経済/金融時系列データなどにも適用可能な方法である.

参考までに $m = 5$ の場合のプログラムのコード例を載せておこう.

```
/*********************************************************************/
/*                    相関次元の計算：レスラー方程式                 */
/*********************************************************************/
#include <stdio.h>
```

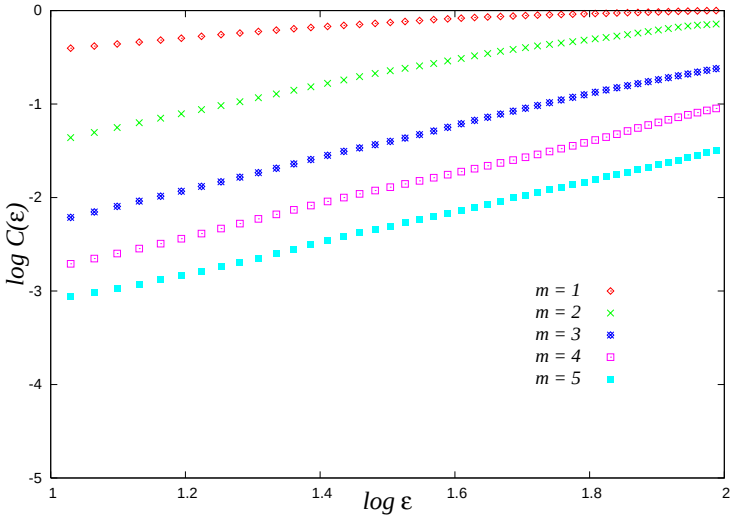


図 37: レスラー方程式において, $x(t)$ から時間遅れ $\tau = 2$ のもとで作成した $m = 1, 2, 3, 4, 5$ 次元のベクトル $(x(t_i), x(t_i + \tau), \cdots, x(t_i + (m-1)\tau)) : i = 1, \cdots, N$ から算出した相関次元. $N = 3000$ として計算してある. $m = 2$ とすると傾き $D_2 \sim 2$ の直線が得られ, この傾きは以降 m を増やしても変わらない.

```
#include <math.h>
#define h 0.01
#define a 0.398
#define b 2
#define c 4
#define imax 50000
#define N 3000
double g1[N];
double g2[N];
double g3[N];
double g4[N];
double g5[N];
double xx[imax];

double func1(x,y,z)
double x;
double y;
double z;
{return (-(y+z));}
double func2(x,y,z)
double x;
double y;
double z;
{return (x+a*y);}
double func3(x,y,z)
```

```
double x;
double y;
double z;
{return (b+z*(x-c));}
double STEP(x)
double x;
{
         if(x>=0.0){
         return (1.0);
         }else{
         return (0.0);
         }
}
double DISTANCE(i,j)
int i;
int j;
{
    return (fabs(g1[i]-g1[j])+fabs(g2[i]-g2[j])+fabs(g3[i]-g3[j])
    +fabs(g4[i]-g4[j])+fabs(g5[i]-g5[j]));
}
double COUNT(epsilon,n)
double epsilon;
int n;
{
         int i,j;
         double sum;
for(i=0,sum=0.0; i<n; i++){
              for(j=0; j<n; j++){
      sum = sum + STEP(epsilon-DISTANCE(i,j));
         }
}
        return ((double)sum/(n*n));
}
main()
{
    FILE *fpr;
    int i,k;
    double x,k1x,k2x,k3x,k4x,kx,y,k1y,k2y,k3y,k4y,ky,z,k1z,k2z,k3z,k4z,kz;
    double arg1,arg2,epsilon;
if((fpr = fopen("m5.dat", "wt")) !=NULL){
       for(i = 0,x=0.01,y=0.01,z=0.01; i < imax; i++){
   k1x = h*func1(x,y,z);
   k1y = h*func2(x,y,z);
   k1z = h*func3(x,y,z);
```

```
                    k2x = h*func1(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
                    k2y = h*func2(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
                    k2z = h*func3(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
          k3x = h*func1(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
          k3y = h*func2(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
          k3z = h*func3(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
                    k4x = h*func1(x+k3x,y+k3y,z+k3z);
                    k4y = h*func2(x+k3x,y+k3y,z+k3z);
                    k4z = h*func3(x+k3x,y+k3y,z+k3z);
          kx = (k1x+2.0*k2x+2.0*k3x+k4x)/6.0;
          ky = (k1y+2.0*k2y+2.0*k3y+k4y)/6.0;
          kz = (k1z+2.0*k2z+2.0*k3z+k4z)/6.0;
  x = x + kx;
  xx[i]=x;
  y = y + ky;
  z = z + kz;
   }
   for(k=0; k< N; k++){
   g1[k] = xx[k];
   g2[k] = xx[k+200];
   g3[k] = xx[k+400];
   g4[k] = xx[k+600];
   g5[k] = xx[k+800];
   }
   for(epsilon= 1.0; epsilon < 100; epsilon=epsilon+0.1){
   arg1 = COUNT(epsilon,N);
   arg2 = epsilon;
   fprintf(fpr,"%lf %lf\n", log(arg2), log(arg1));
   }
}
     fclose(fpr);
}
```

9.4　相関次元の一般化

前節で学んだ相関次元は単一のパラメータ q を導入することで次のように一般化することができる.

$$D_q = \frac{1}{q-1}\lim_{\epsilon\to 0}\frac{\log\sum_{i=1}^{n(\epsilon)} p_i^q}{\log\epsilon} \tag{171}$$

ここで, $n(\epsilon)$ は半径 ϵ の球で対象 (軌道) を覆うために必要な球の個数であったから, 前節で調べた相関時限の場合には $N = n(\epsilon)$ であることに注意しよう. そこで, 上式で $q = 2$ とおいてみると

$$D_2 = \lim_{\epsilon\to 0}\frac{\log\sum_{i=1}^{N} p_i^2}{\log\epsilon} \tag{172}$$

となり, 相関次元が復活する. また, $q = 0$ とおけば

$$D_0 = -\lim_{\epsilon\to 0}\frac{\log\sum_{i=1}^{n(\epsilon)} 1}{\log\epsilon} = -\lim_{\epsilon\to 0}\frac{\log n(\epsilon)}{\log\epsilon} \tag{173}$$

となり, 容量次元に一致する. また, $q = 1$ の場合も別な次元が定義される. ここで $q \to 1$ の極限は $0/0$ を与え, 厄介に見えるが, 分母/分子を q で微分した後に $q \to 1$ 極限をとっても良いことに気がつけば (いわゆる「ロピタルの定理」), 確率の規格化: $\sum_{i=1}^{n(\epsilon)} p_i = 1$ に注意して

$$\lim_{q\to 1}\frac{\partial}{\partial q}\log\left(\log\sum_{i=1}^{n(\epsilon)} p_i^q\right) = \lim_{q\to 1}\left\{\frac{\sum_{i=1}^{n(\epsilon)} p_i^q \log p_i}{\sum_{i=1}^{n(\epsilon)} p_i^q}\right\} = \sum_{i=1}^{n(\epsilon)} p_i \log p_i \tag{174}$$

$$\lim_{q\to 1}\frac{\partial(q-1)}{\partial q} = 1 \tag{175}$$

であるから

$$\lim_{q\to 1} D_q = D_1 = \lim_{\epsilon\to 0}\frac{\sum_{i=1}^{n(\epsilon)} p_i \log p_i}{\log\epsilon} \tag{176}$$

が得られる. これを情報理論の平均情報量 (エントロピー) との形式的類似性から**情報次元**と呼ぶ. 従って, D_q は相関次元 D_2, 容量次元 D_0, および情報次元 D_1 をその特別な場合として含む「一般化された次元」を与えている. 既に述べたように, これらの次元は場合によっては非整数次元となり得て, 再来週から学習していくことになる「フラクタル図形」を特徴つける際にも用いられる.

レポート課題 8

ローレンツ方程式 ($a = 10, b = 8/3, c = 30$) に対し, $m = 5, \tau = 2$ と選んだ場合の相関次元を計算せよ.

※ 次週 (6/23) は講義室にて演習を行います. ラップトップを持っている方は持参してください.

[カオス編] はこれでおしまいです. より進んだ内容や応用などは後期の情報工学演習 II, あるいは大学院講義「混沌系工学特論」などで扱います (予定). 大学院の講義ですが, やる気のある皆さんの聴講を歓迎します.