複雑系科学演習　資料

スライド後半は、参考資料

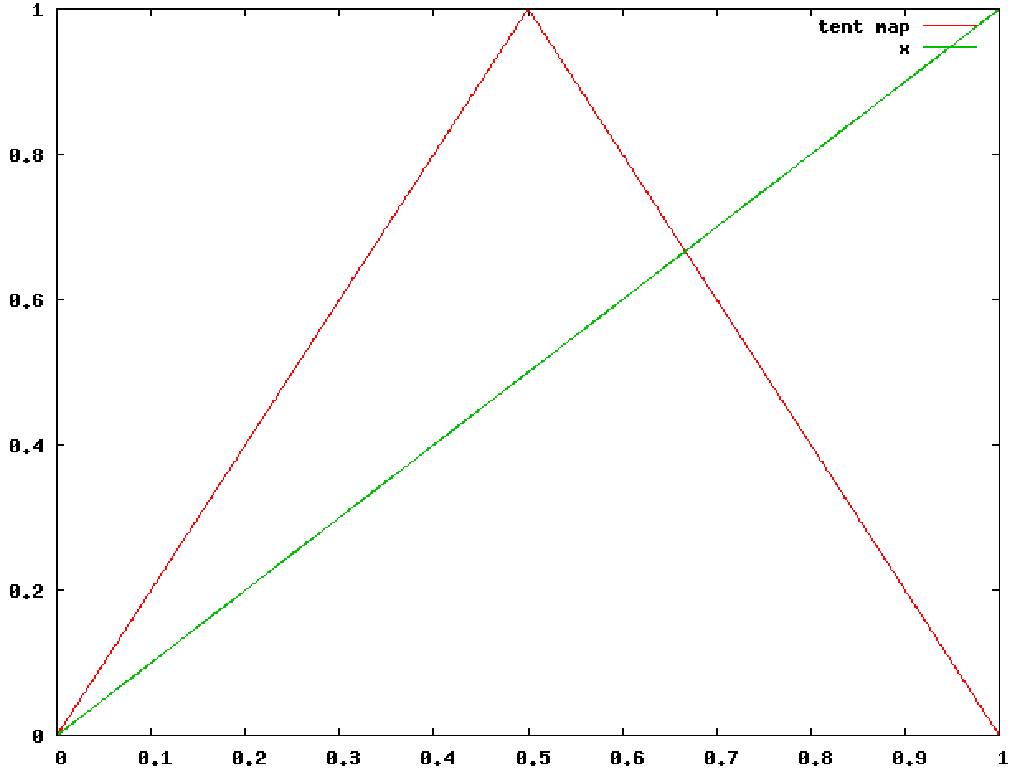
これまでに学んだこと

Logistic写像

- 時系列変化，リターンマップの描画
- 初期値鋭敏性
- 不動点，周期解
  - 安定性，不安定性
- 周期倍分岐
- 3周期窓
- リアプノフ指数

- 今日は復習をかねてテント写像を使います

テント写像：$x_{n+1} = \begin{cases} 2x_n & (0 \le x_n \le 1/2 \text{ のとき}) \\ 2(1 - x_n) & (1/2 < x_n \le 1 \text{ のとき}) \end{cases}$



課題2 解答例を関数を使って書き換えておく

```
#include<stdio.h>
#define N 100

int main(void){
  int n;
  double r;
  double xn = 0.7;
  double xn1;

  scanf("%lf", &r);
  for(n=0; n<=N; n++){
    xn1 = r * (1-xn) * xn;
    printf("%lf %lf\n", xn,
   xn1);
    printf("%lf %lf\n", xn1,
   xn1);
    xn = xn1;
  }
return 0;
}
```

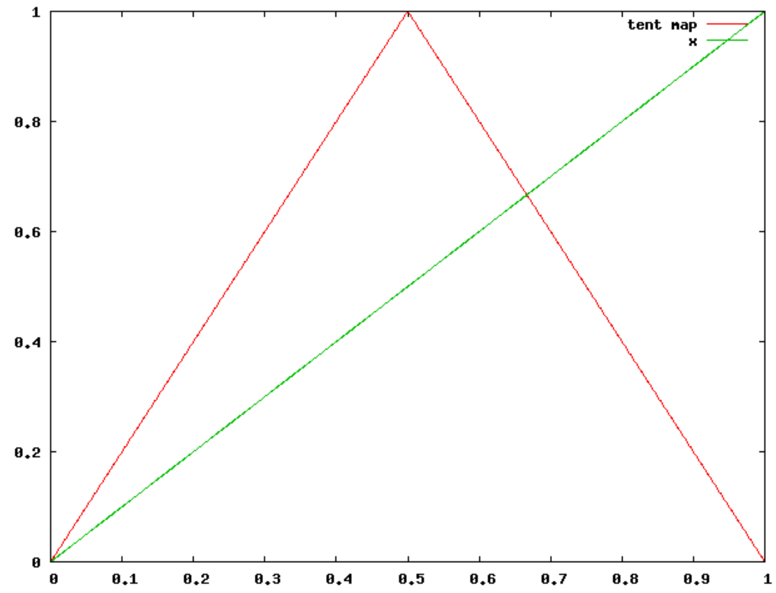
⇒

```
#include<stdio.h>
#define N 100

double next(double x, double r) {
  return r*(1-x)*x;
}

int main(void){
  int n;
  double r;
  double xn = 0.7;
  double xn1;

  scanf("%lf", &r);
  for(n=0; n<=N; n++){
    xn1 = next(xn, r);
    printf("%lf %lf\n", xn, xn1);
    printf("%lf %lf\n", xn1, xn1);
    xn = xn1;
  }
return 0;
}
```



- 提出課題

1. 前頁の関数nextをテント写像に書き換え，リターンマップを描け

テント写像：$x_{n+1}=\begin{cases}2x_n & (0\le x_n\le 1/2\ \text{のとき})\\ 2(1-x_n) & (1/2<x_n\le 1\ \text{のとき})\end{cases}$

• 周期解の例を作ってみる

2. $x_0 = 1/10$として，どのような周期解になるか厳密に手計算せよ．

3. $x_0 = 0.1$として，前のプログラムを実行せよ．ただし$N = 100$としx_{100}まで数値計算せよ．どのような結果となったか．理由も述べよ．

必要であればこのPDFファイルの後半を参考にして
参考プログラム１を実行してみよ

・テント写像のカオス性について

4. テント写像のリアプノフ指数を計算せよ.
プログラミングが必要なければ手計算でよい.

・テント写像とロジスティック写像について

タイトルがHintです

5. テント写像は x_n から x_{n+1} を求める写像である。
$y_n = \sin^2\left(\frac{\pi}{2}x_n\right)$ と定義して, y_{n+1} を y_n の式として表わせ.

レポート課題

- 上記1～5について，
- *月*日*時までに
  234号室に紙の形で提出

【参考資料】

以下のページは丸め誤差に関しての資料です・

誤差について

- 誤差とは？
  - 値に対する「ずれ」の量
  - 「本当の値」が存在するなら、その値との差

48g？

48．1g？

48．10009g？

これは測定誤差と呼ばれる

計算機の限界？

- 計算機は計算を「正確」に実行できる

Bugがなければ…

- しかし完璧ではない

例）符号なし整数（32bits）

$0 \sim 2^{32} - 1 (= 4294967295)$しか表せない

大きな数が表せない？

⇒ もっとビット数を用意すればよい

⇒ ただしメモリは有限

- 有理数の計算

例）

$\frac{5}{2}$ なら"5"と"2"の2つの整数を保持すればよい

- 例）

a/bの有理数を(a, b)と表すことにする。(a, b)−(c, d)=(x, y)としたとき、xとyを求めよ。

xとyは必ず整数に取れる。整数2つを用いて表せる

四則演算では誤差を生じない

- 実数はどうか？

例）　有理数を実数として表すと…

$$\frac{5}{2}=2.5$$

いつまでも終わらない
(循環するから明示はできる)

$$\frac{5}{3}=1.66666666666666666666666666666666\cdots$$

例）　無理数は分数の形で表せない

いつまでも終わらないし
(おそらく)規則性もない

$$\pi=3.14159265358979323\cdots$$

有限のメモリしかないコンピュータでは
円周率という数値を表わせない

計算誤差　とは？

- どのような誤差があるのか知っておくことが必要
  - 丸め誤差
  - 打切り誤差
  - 桁落ち誤差
- 誤差があっても十分信頼できる値であればよい

（参考）プログラム　１

```
#include <stdio.h>
int main() {
  double x;
  x = 1.0;
  printf("x = %.30lf\n", x);
  x = 1.3;
  printf("x = %.30lf\n", x);
}
```

丸め誤差

- 数値を有限桁で打ち切ることによる誤差

- 先ほどのプログラムでは…
  1. 10進数を与える
  2. その数を2進数に変換して保存(変数xの中身)
  3. printf でその2進数を10進数に変換して表示

２）の2進数に変換するところで誤差が入る

2進数の復習

- 整数なら

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- 10進数

  <u>１００の位</u>　<u>10の位</u>　<u>1の位</u>

- 2進数

  <u>4の位</u>　<u>2の位</u>　<u>1の位</u>

- 10進数

　１００の位　　10の位　　1の位　　0.1の位

- 2進数

　　　　4の位　　　2の位　　1の位　　0.5の位

例）10進数の0.5は、2進数の0.1
　　10進数の1.5は、2進数の1.1
　　10進数の1.25は、2進数の1.01

では、10進数の1.3は2進数で?

- $1.3_{(10)} = 1.0100110\cdots_{(2)}$
- double型では、小数点以下52ビット目で丸める
- 丸め方は省略
  - 切り上げ、切捨て、四捨五入、五捨六入、偶数丸め、など

- $1.3_{(10)} = 1.0100110\cdots_{(2)}$
- この2進数を小数点以下4ビット目を切り捨てると　1.010
- 10進数に戻して表示させると 1.25 になってしまう ⇒ 丸め誤差

浮動小数点 （IEEE754形式）

例） -1.234×2^{56}

$(\text{符号})(\text{仮数})x(\text{基数})^{\text{指数}}$

の形

double型の場合、

符号 1bit
仮数 52bits
指数 11bits
(基数は2で固定)

あわせて64ビットで表す取り決め。

double型の内部表現

(符号)(仮数)x(基数)^{指数}