

研究会資料

ガスタービンで発生するサージ・燃焼不安定の非線形ダイナミックスの解明と短期予測手法の開発



立命館大学 理工学部 機械工学科 後藤田研究室

4 回生 奥野 佑太 (OKUNO YUTA)

September 14, 2012

研究計画表(2012年度第3期)

燃焼器関連

| 期日 | 内容 | 備考 |
|---------|-------------------------------|------------------|
| 11/30 | 第9回アジア太平洋国際燃焼会議 講演原稿 提出 | 1/15(2013), 結果通達 |
| 12/05 | 第50回燃焼シンポジウム(ポスター講演) | |
| 2/28 | 第9回アジア太平洋国際燃焼会議 最終公演原稿 提出 | |
| 3/15-17 | 日本機械学会 関西支部 定時総会講演会 at 大阪工業大学 | |

回転翼関連

| 期日 | 内容 | 備考 |
|--------|----------------------|---------------|
| 9/26 | 順列エントロピーの検知可能最小時間の調査 | 立花さんと打ち合わせ |
| 10月下旬? | 回転要素実験 見学 | 同時に立花さんと打ち合わせ |

非線形時系列解析関連

| 期日 | 内容 | 備考 |
|----|--------------------------------|----|
| / | マルチフラクタル解析の検証 | |
| / | Cycle surrogate (small) | |
| / | ガウシアンカーネル・アルゴリズム (相関次元) | |
| / | マルチフラクタル解析とハースト指数の関連づけ | |
| / | 相互情報量(C言語版)の作成 | |
| / | 多層パーセプトロンの学習 | |
| / | KSエントロピー・リアプノフ指数・順列エントロピーの関連づけ | |
| / | 最大リアプノフ指数の時間変化(検知) | |

管理ソースコード一覧

2012/9/13現在

非線形時系列解析 ※下線：主に使用するコード

| No(仮) | 名称 | C (gcc 4) + OpenMp | MATLAB | 備考 |
|---------|----------------------------|--|-------------------------------------|-----------|
| 1001-01 | 3次元位相空間(アトラクタ) | - | attractor3D.m | - |
| 1002-01 | 相互情報量 | - | mutual.m | - |
| 1003-01 | 並進誤差 | 書換予定 | Etrans.m | 変数一致作業中 |
| 1003-02 | 並進誤差の時間変化 | - | EtransDetect.m | 変数一致作業中 |
| 1004-01 | 最大リアプノフ指数(Kantz x Wayland) | lyap_w_omp.c lyap_w.c | lyapunov.m | - |
| 1005-01 | 局所非線形予測法(Sugihara&May) | sugihara.c | - | - |
| 1006-01 | マルチフラクタル解析(3次元) | multi_f_3D.c | - | - |
| 1007-01 | サイクルサロゲートデータ | cy_surrogate.c | rs.m (RS対応) | - |
| 1007-02 | FTサロゲートデータ | - | ft.m | - |
| 1007-03 | AAFTサロゲートデータ | - | aaft.m | - |
| 1007-04 | IAAFTサロゲートデータ | - | iaaft.m | - |
| 1008-01 | 動底基底関数ネットワーク | rbfn.c | - | - |
| 1009-01 | 多層パーセプトロン | 書換中 | - | - |
| 1010-01 | 順列エントロピー | p_entropy.c | permentropy.m | 変数一致作業中 |
| 1010-02 | 順列エントロピーの時間変化 | - | PermEntropyDetect.m | 変数一致作業中 |
| 1011-01 | 相関次元(GP法) × nonsubjective | gp_harikrishnan.c | | ドキュメント作成中 |

管理ソースコード一覧

2012/9/13現在

力学モデルの時系列生成 ※下線：主に使用するコード

| No(仮) | 名称 | MATLAB | 備考 |
|---------|-----------------|---------------------|----|
| 2001-01 | ロジスティック写像 | <u>logistic.m</u> | - |
| 2002-01 | テント写像 | <u>tent.m</u> | - |
| 2003-01 | (磁場?)ローレンツ方程式 | <u>horizontal.m</u> | - |
| 2004-01 | van der pol 方程式 | vanderpol_main.m | - |

確率過程の時系列生成 ※下線：主に使用するコード

| No(仮) | 名称 | MATLAB | 備考 |
|---------|-----------------|-------------------------|-------------|
| 3001-01 | 非整数ブラウン運動 | <u>fbm.m</u> | 名古屋大コード |
| 3001-02 | 非整数ガウスノイズ | <u>fgn.m</u> | fbm.mの増分を取得 |
| 3002-01 | ホワイト・ピンク・レッドノイズ | <u>SpatialPattern.m</u> | Webコード |

線形時系列解析 ※下線：主に使用するコード

| No(仮) | 名称 | MATLAB | 備考 |
|---------|-------------------|-----------------------------|---------------------|
| 4001-01 | 自己相関 - [1]巡回推定 | <u>CircularEstimation.m</u> | |
| 4001-02 | 自己相関 - [2]不偏推定 | <u>UnbiasedEstimation.m</u> | |
| 4002-01 | パワースペクトル解析(welch) | <u>psd_welch.m</u> | 立花さんのコードとは別 |
| 4003-01 | 1次・2次統計量の計算 | <u>CheckStatistics.m</u> | 平均・分散・度数分布・自己相関・PSD |

① コーディングルール・テクニック

コーディングルール(2012/4/13)

関数・変数全般について

MATLABでは、組み込み関数はすべて小文字である。可読性の向上およびコード最適化のため、関数や変数の命名はPascal形式を用いることにする。

<参考>Pascal 形式

識別子の最初の文字と、後に続いて連結されている各単語の最初の文字を大文字にする。Pascal形式は、3文字以上から構成される識別子に対して使用できる。

Ex. SampleProgram, Attractor3D

※注意事項

識別子が複数の単語で構成される場合、アンダースコア(“_”)やハイフン(“-”)などの区切り記号を単語間に使用しない。代わりに、大文字小文字の区別を使用して各単語の先頭を示す。

プログラムカウンタ・作業用変数について

・変数に i と j は使用しない

MATLAB は、i と j の文字を使用して虚数単位を表す。複素数の計算に使用する場合は、変数名に i および j を使うのは避ける。

(出典:MathWorks社Webサイト)

作業用変数は変数名+WorkまたはTempとする。

Ex. IWork, dTemp, VariableNameWork

高速化のためのテクニック

ループのベクトル化

MATLAB は行列言語を使用し、ベクトルや行列の演算用に設計されている。この設計を利用して、ベクトル化アルゴリズムを使用することで、コードを高速化できる。“ベクトル化”とは、for や while ループを等価なベクトル、または行列演算に変換することを指す。

Ex. ベクトル化の簡単な例

```
l = 0;  
for T = 0:.01:10  
    l = l + 1;  
    y(l) = sin(T);  
end
```



ベクトル化

```
T = 0:.01:10;  
y = sin(T);
```

2重forループにも適用できるなど、応用例が多々ある。詳しくは MathWorks社 Webサイト パフォーマンスを向上させる方法を参照。

※ xIndexなど、変数名+Indexはベクトル計算するための変数。

配列の事前割り当て ※ベクトル化ができない場合

ループを繰り返すたびにデータ構造のサイズが次第に“増大”する for ループと while ループは、パフォーマンスと使用メモリにマイナスに影響する可能性がある。配列に必要な最大スペースを事前に割り当てれば、コードの実行時間を改善できることが多い。

```
x = 0;  
for k = 2:1000  
    x(k) = x(k-1) + 5;  
end
```



事前割り当て
(zeros, onesなど)

```
x = zeros(1, 1000);  
for k = 2:1000  
    x(k) = x(k-1) + 5;  
end
```

1 位相空間 (アトラクタ) - attractor3D(x,Tau)

概要

位相空間(アトラクタ)を3次元で描く。

使用変数

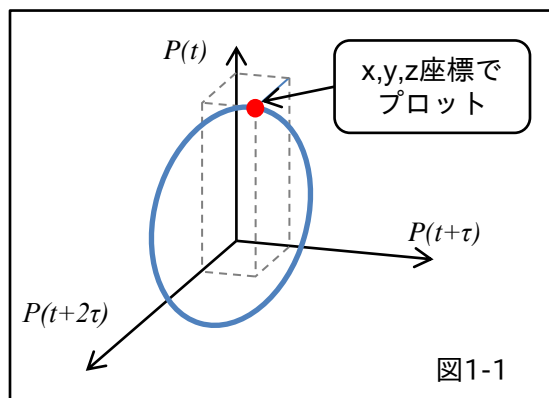
D: 埋め込み次元 (Embedding dimension)
 d: 埋め込み次元まで変化させる作業用変数 (=1-3)
 N: 時系列データの点数
 Tau: 遅れ時間 (Delay time of phase space)
 x: 時系列データ
 X: 計算結果(3次元)

計算過程・図示方法

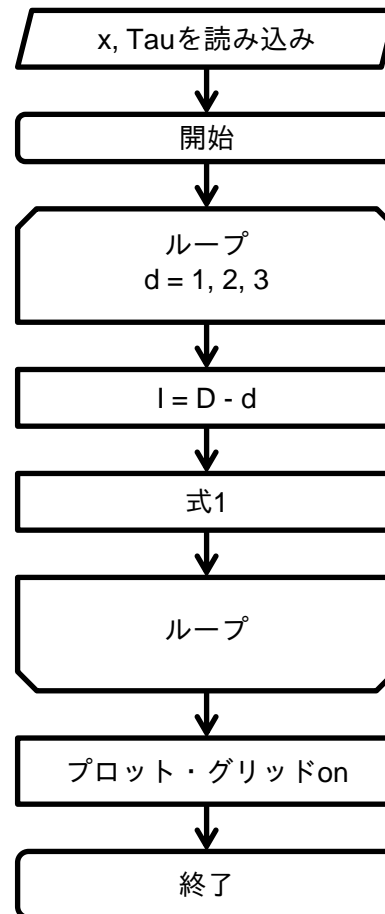
式1: $X(:, d) = x(1+l \times \tau : N-(d-1) \times \tau)$

- ・ $d = 1$ のとき, $P(t+2\tau)$ (ただし, $1+2\tau \sim N$)
 - ・ $d = 2$ のとき, $P(t+\tau)$ (ただし, $1+\tau \sim N-\tau$)
 - ・ $d = 3$ のとき, $P(t)$ (ただし, $1 \sim N-2\tau$)
- を表し, それぞれをXに代入する。

この過程により, x座標 $P(t+2\tau)$, y座標 $P(t+\tau)$, z座標 $P(t)$ として位相空間を表すことができる。 (図1-1)



フローチャート



式1: $X(:, d) = x(1+l \times \tau : N-(d-1) \times \tau)$

2 相互情報量 - Mutual(x,xLength)

概要

時系列データの相互情報量を遅れ時間別に出力,
横軸を遅れ時間, 縦軸を相互情報量として2次元で描く.

使用変数

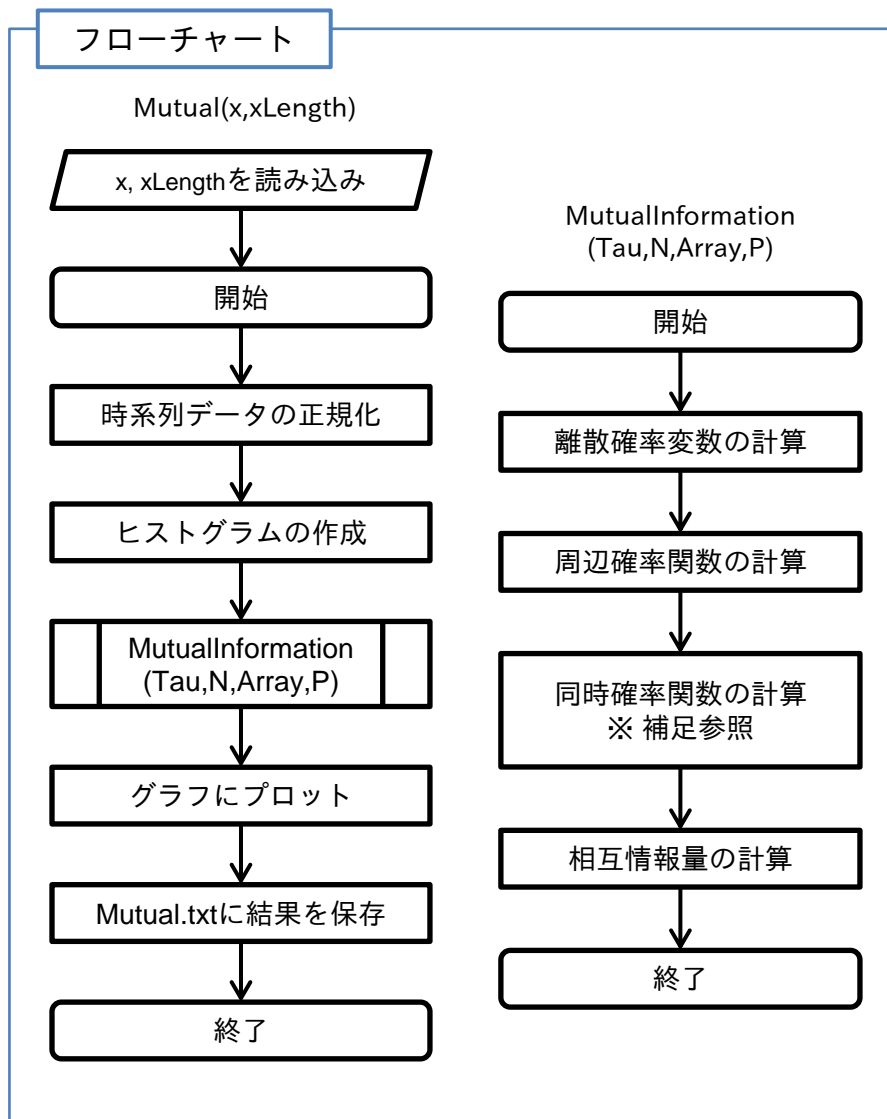
x: 時系列データ
xLength: 計算する遅れ時間の最大値(x軸の最大値)
N: 時系列データの点数
P: ヒストグラムを作成する分割数
xNormalized: 正規化した時系列データ
Tau: 遅れ時間 (Delay time of phase space)

計算過程

- ・時系列データの正規化
時系列データを0～1の範囲に正規化する(= xNormalized)
- ・ヒストグラムの作成(P分割)
(xNormalizedの各行列要素)×Pを切り上げ(=Array),
Arrayのある要素=0 (xNormalizedのある要素=0)のとき 1 とする.
- ・離散確率変数の計算 = Hi, Hj, Hij
1～Pまでの各データの個数を計算
- ・同時確率関数・周辺確率関数の計算 = Pi, Pj, Pij
同時確率関数 = 離散確率変数 ÷ データの個数
※ 周辺確率関数についても同様
※ データの個数 = 時系列データ数(N) - 遅れ時間(Tau) + 1
- ・相互情報量の計算

$$MI = \sum_{I=1}^{N-\tau+1} \sum_{J=1}^{N-\tau+1} P_{ij}(I, J) \log_2 \frac{P_{ij}(I, J)}{P_i(I)P_j(J)}$$

フローチャート



2 [補足1]ヒストグラム・同時確率関数について

ヒストグラム

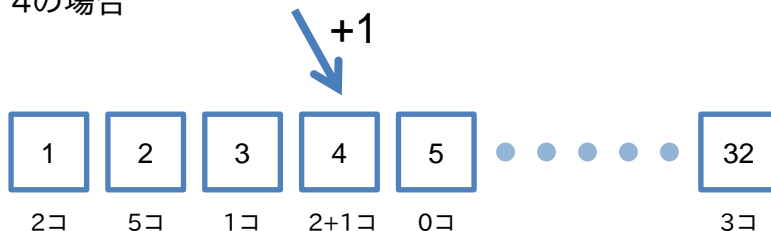
正規化した時系列データ×ヒストグラムを作成する分割数を切り上げ、各数の個数を数える。 ※ $\text{ceil}(x_{\text{Normalized}} \times P)$

例. $x_{\text{Normalized}} = 0.100$, $P = 32$ のとき

$$0.100 \times 32 = 3.2$$

3.2を切り上げ → 4

4の場合



※ 1～32までの総数 = 正規化した時系列データの総数

注意事項 $x_{\text{Normalized}} = 0.000$, $P = 32$ のとき

$$0.000 \times 32 = 0.0$$

0.0を切り上げ → 0

if文 0→1

※ 0のときは1として算入

同時確率関数 (joint probability function)

例. 黒いサイコロと白いサイコロがあるとして...

試行X: A=1, B=4 / 試行Y: A=5, B=2 / 試行Z: A=B=3

同時確率関数は、条件によって結果が異なる。サイコロA,Bの結果(例1)、サイコロ2つの結果の大小(例2)などがある。

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | A | | | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | | | | | | | |
| 2 | | | | | Y | | |
| 3 | | | | Z | | | |
| 4 | X | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |

例1: サイコロA,Bの結果

A,Bが互いに独立



全体に分布

※サイコロを区別する

相互情報量はこの条件

| | | | | | | | |
|---|--|---|---|---|---|---|---|
| | | 大 | | | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | | | | | X | | |
| 2 | | | | | | Y | |
| 3 | | | | Z | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |

例2: 結果の大小

大きい・小さいで区別する
(結果が出たあとに判断)



右上半分に分布

※サイコロを区別しない

③ 並進誤差page1- Etrans(x)

概要

時系列データの並進誤差を各条件に基づき計算する。

使用変数

| | |
|------|----------------------------------|
| x: | 時系列データ |
| N: | 時系列データの点数 |
| NN: | 埋め込み次元後のデータ数 |
| Tau: | 遅れ時間 (Delay time of phase space) |
| D: | 埋め込み次元 |
| T: | 予測ステップ |
| K: | 近傍点数 |
| M: | 既定関数の個数 |
| Q: | 計算の繰り返し数 |

計算過程

- ・データを位相空間へ変換 (=Xi) ※ソースではベクトル化
 $X_i(\text{データ番号}, d) = x(\text{データ番号} + (d-1) \times \tau)$
- ・ランダムに基準点を設定 (=X0) ※ソースではベクトル化
 x のデータ番号は $0 \sim 1$ までのランダム数 $\times (NN-T)$ を切り上げたものとする。これを $1 \sim D$ までそれぞれ決定する。

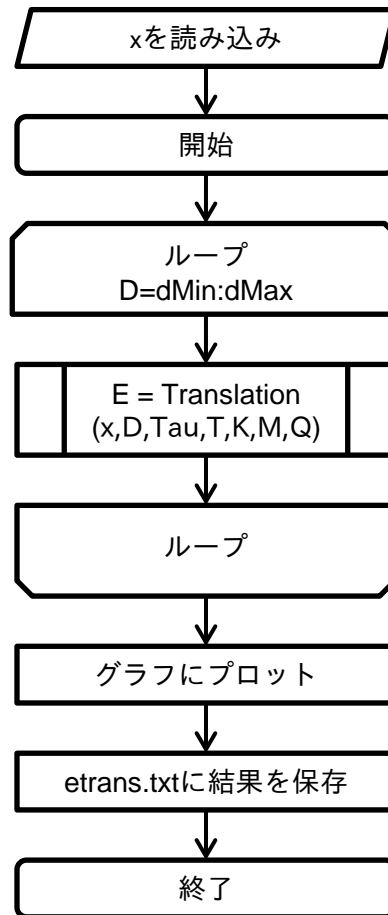
- ・ユークリッド距離の計算

$$Distance = \sum_{I=1}^{NN-T} \sum_{d=1}^D (X_{iI,d} - X_{0d})^2$$

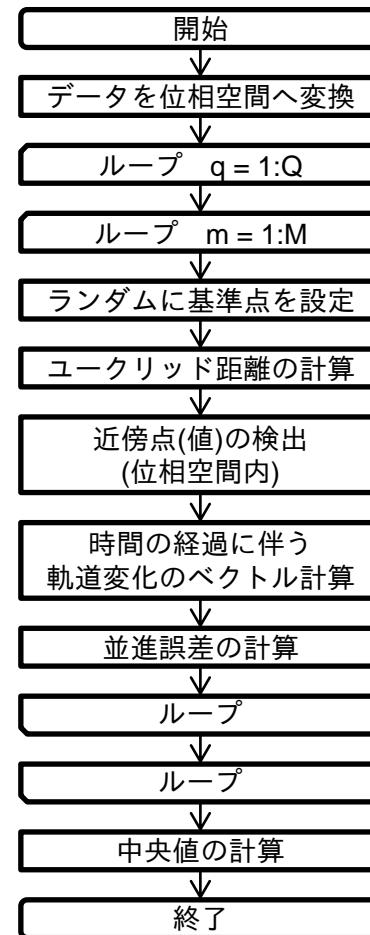
- ・近傍点(値)の検出 ※位相空間内 / ※近傍点の時系列データ順番号=X0NPI
 時系列データ順番号を保ち、ユークリッド距離を昇順に並び替え
- ・時間の経過に伴う軌道変化のベクトル(=V)計算
 $V(k,d) = X_i(X0NPI+T,d) - X_i(X0NPI,d)$ (ただし $X0NPI = 1, 2, \dots, K$)
 ※近傍点K個分それぞれのベクトルを計算
- ・並進誤差の計算・中間値の計算 …[追加1]並進誤差を参照

フローチャート

E=Etrans(x)



E=Translation
(x,D,Tau,T,K,M,Q)



③ 並進誤差page2- Etrans(x)

並進誤差の計算 ※ソースではベクトル化

時間の経過に伴う各軌道の方向の分散を並進誤差(=Etrans)とする。

$$V_{Average} = \frac{1}{K+1} \sum_{k=1}^{K+1} V$$

※ただし各次元ごとに計算(d=1,2,...,D)

$$V_{Norm} = \|V - V_{Average}\|^2$$

$$V_{AverageNorm} = |V_{Average}|^2$$

$$E_{trans} = \frac{1}{K+1} \sum_{k=1}^{K+1} \frac{V_{Norm}}{V_{AverageNorm}}$$

ここで位相空間内にある位置ベクトルX0を無作為にM個選び、Q回繰り返し計算し、各並進誤差(Etrans)を保存する。

中間値の計算

求めた各並進誤差を、各回(q=1,2,...,Q)ごとに(M個を)昇順で並べる。その後、M/2を四捨五入したもの(各回の中央値)を合計し、Qで割る。これをD次元の並進誤差とする。

並進誤差の結果について

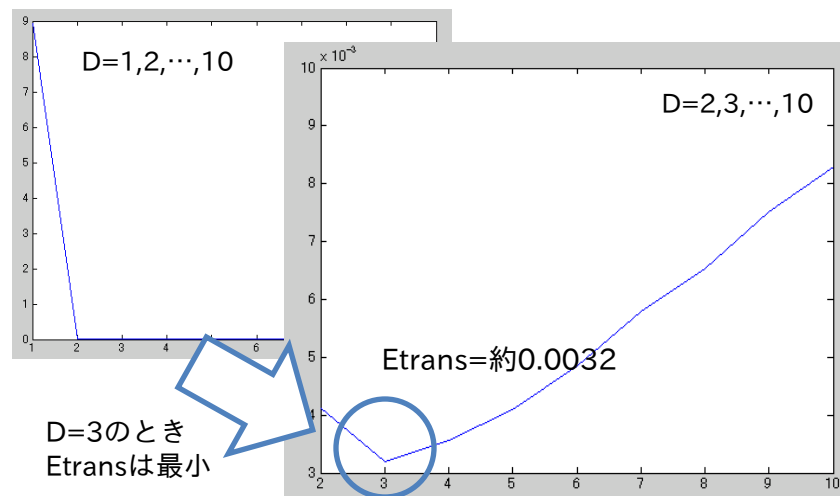
並進誤差Etransは、0<=Etrans<=1の範囲を示す。対象となる時系列データは、Etrans=0のとき周期振動、Etrans=1のときランダム振動であると判断できる。※0.01以下の場合、カオスと判定する。

計算結果の考察

テスト時系列データlorenzを、遅れ時間 $\tau=16$ 、近傍点数 $K=4$ 、予測ステップ $T=10$ 、既定関数の個数 $M=5$ 、計算回数 $Q=10000$ の条件下にて、今回更新したEtrans.mを用いて並進誤差の計算を行った。その結果を下図に示す。

並進誤差はD=3のときに最小値、約0.0032をとる。すなわち、テスト時系列データlorenzはD=3のとき、もっとも周期振動に近いと考えられる。

また、Q=1000のとき、D=2~10の計算に必要な時間は従来は約150秒であったが、アルゴリズムの改善により現在は約60秒である。これは従来比約60パーセントの計算時間短縮である。(井川さん作成のwayland.cでは約97秒)



4 最大リアプノフ指数(Kantzアルゴリズム) page1- lyapunov(x)

概要

時系列データの最大リアプノフ指数を計算する。

使用変数

| | |
|----------|----------------------------------|
| x: | 時系列データ |
| D: | 埋め込み次元 |
| Tau: | 遅れ時間 (Delay time of phase space) |
| M: | 無作為に基準点を選択する回数 |
| K: | 近傍点数 |
| T: | 最大経過時間 |
| xStride: | 入力データの刻み幅 |
| dt: | 最大リアプノフ指数の刻み幅 |
| N: | 時系列データの点数 |

計算過程

・ S(t)の計算

$$S(t) = \frac{1}{M} \sum_{m=1}^M \ln \left(\sum_{k=0}^K \sum_{d=1}^D |X_{0+t} - X_{k+t}| \right)$$

経過時間tにおけるS(t)を上式で計算する。本来はすべての点(N点)について計算しなければならないが、これは膨大な計算時間を要する。そこで不作為にM回基準点を選び計算することで近似する。

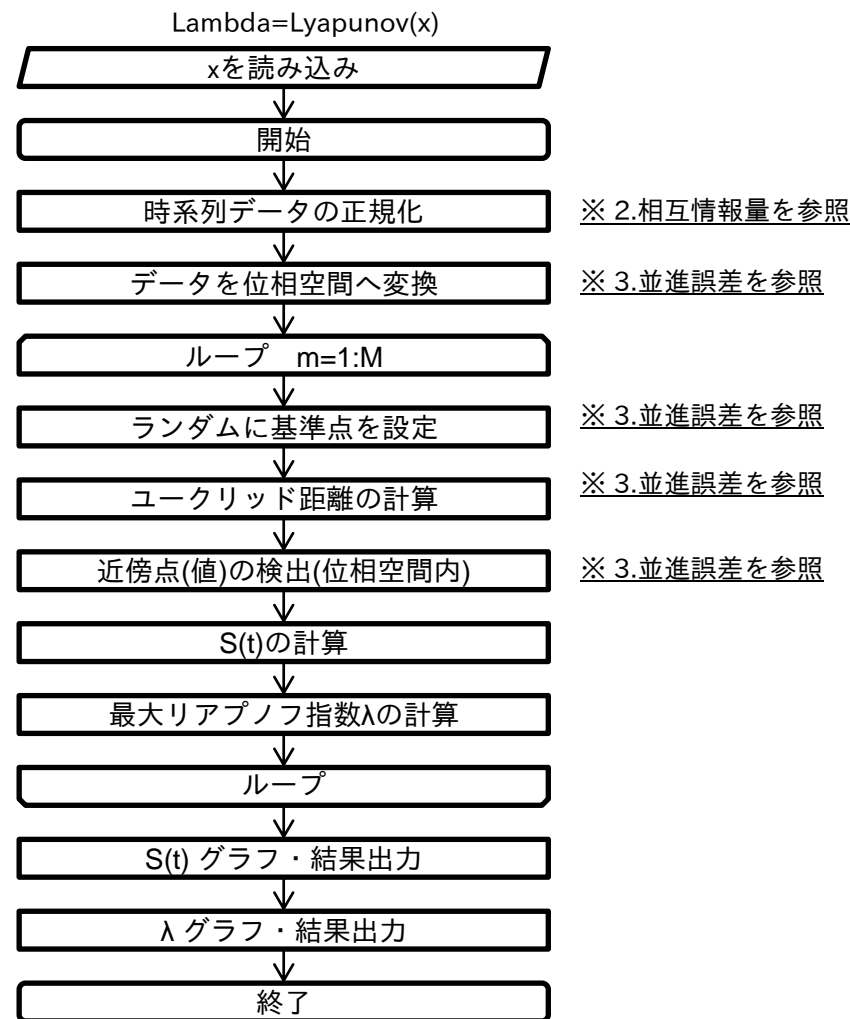
・ 最大リアプノフ指数λの計算

$$\lambda = \frac{\partial S(t)}{\partial t}$$

kantzアルゴリズムでは、スケーリング域におけるS(t)の傾きが最大リアプノフ指数である。

Our numerical value for the maximal Lyapunov exponent is the slope of the curve S(τ) in the scaling region. [1]

フローチャート



4 最大リアプノフ指数(Kantzアルゴリズム) page2- lyapunov(x)

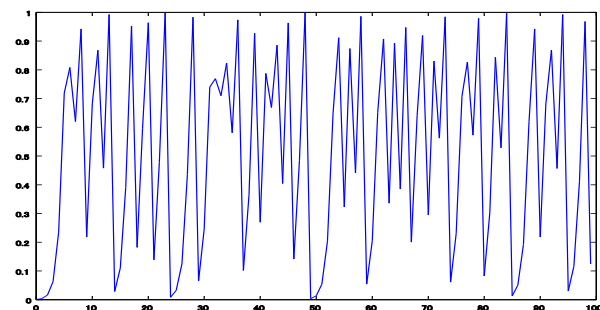
プログラムの信頼性

ローレンツ方程式, ロジスティック写像($a=4$)をテストデータとして用いる. 井川さんは過去に論文などを調べ, ローレンツ方程式のリアプノフ指数は約1.5と報告(昨年度研究会(2011年8月24日))している. これは $dt=100$ (1秒間あたり)における計算結果(図4.4)がその値を示している. また $a=4$ のとき, ロジスティック写像(式4.1)の時系列データを用意し計算を行った. 本プログラムで数値計算をした結果(図4.1)は約0.7を示し, 解析的に求められる $\ln 2$ とほぼ等しい. 以上より, 本プログラムの出力結果は信頼できる.

ロジスティック写像について

$$x_n = ax_{n-1}(1 - x_{n-1}) \quad (4.1)$$

ロジスティック写像は $a=4$ のとき, 典型的なカオスで, ノイズがないのにも関わらず不規則な変動を生じる^[1]. また, 解析的にリアプノフ指数が $\ln 2$ と求められる^[2].



ロジスティック写像
($a=4$, $n=1,2,\dots,100$)

計算結果の考察

リアプノフ指数の計算が目的ではなく, カオスの判定や短期予測可能時間の調査が目的であれば, dt を小さくしたことで図4.2や図4.3に示すように, 最大リアプノフ指数の値が上下しても影響はほとんどない. むしろ, 短期予測可能時間やその精度を予測したい場合, よりリアルタイムに近い情報が得られ, メリットの方が大きいと思われる.

さらに検討すれば, Kantzアルゴリズムは $S(T)$ の傾きが正であればカオスと判定でき, 最大リアプノフ指数の計算はプログラムの信頼性を確かめる場合を除き必要ないことが多いと思われる.

リアプノフ指数が負を示す場合, ランダム時系列と判断できる.

※各テストデータの遅れ時間はmutual.m, 埋め込み次元はEtrans.mより求め, Lyapunov.mによって各図を作成している.

ロジスティック写像 作成プログラム

function Logistic

```
N=5000;
a=4;
x=zeros(N,1); x(1,1)=0.001;
for I=2:N
    x(I,1)=a*x(I-1)*(1-x(I-1,1));
end
```

```
plot(0:N-1,x)
save logistic.txt x -ascii
```

※注意事項

Rosensteinにより提案されている手法は

$$d_j = C_j e^{\lambda_{MAX} T \Delta t}$$

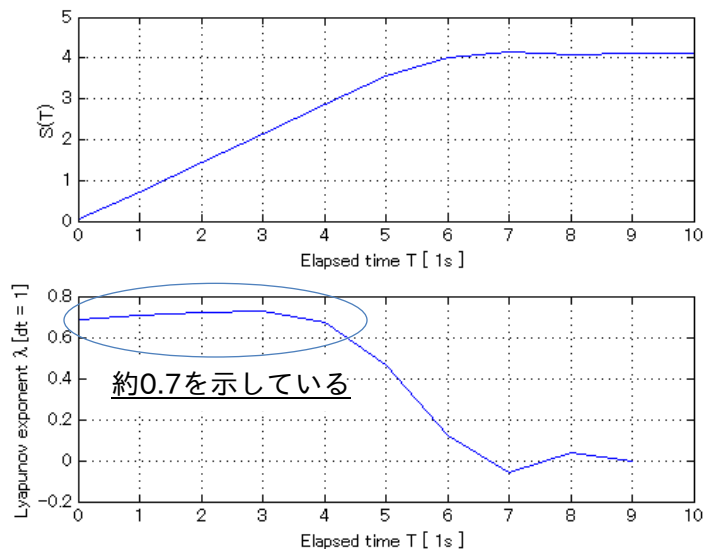
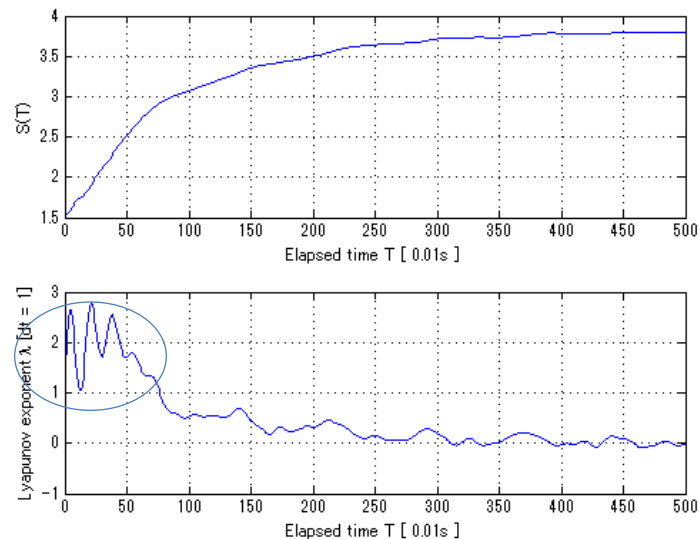
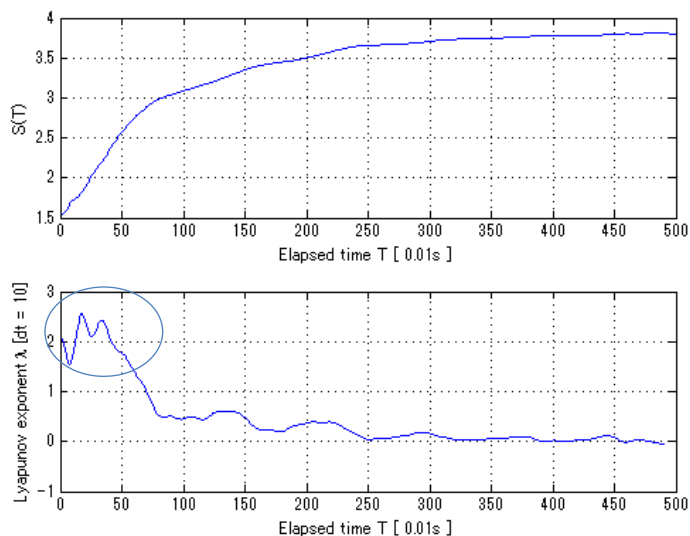
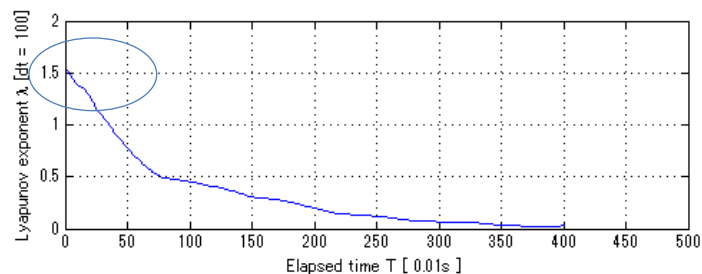
から

$$\ln d_j = \ln C_j + \lambda_{MAX} (T \cdot \Delta t)$$

と変形し, リアプノフ指数 λ_{MAX} は $\langle \ln d_j \rangle$ 対 $i \Delta t$ の変化の傾きから推定する.

今回のKantzが提案している手法とは異なる.

4 最大リアプノフ指数(Kantzアルゴリズム) page3- lyapunov(x)

図4.1 ロジスティック写像($a=4$)図4.2 ローレンツ方程式($dt=1$)図4.3 ローレンツ方程式($dt=10$)図4.4 ローレンツ方程式($dt=100$)

5 局所非線形予測法(Sugihara&Mayアルゴリズム)

概要

データベース更新型非線形予測プログラム。
動的メモリの確保, ファイル入出力エラー処理実装済。

使用変数

NMAX: 入力データ点数
N: データベースのデータ点数
P: 予測ステップ数
K: 近傍点数
D: 埋め込み次元
Tau: 遅れ時間
M: 無作為に基準点を選択する回数
Part: データベース更新間隔
dx: 時間スケール

計算過程

・非線形予測

$$\mathbf{X}(t_p + T) = \frac{\sum_{k=1}^K \exp(-d_k) \mathbf{X}(t_k + T)}{\sum_{k=1}^K \exp(-d_k)} \quad (5.1)$$

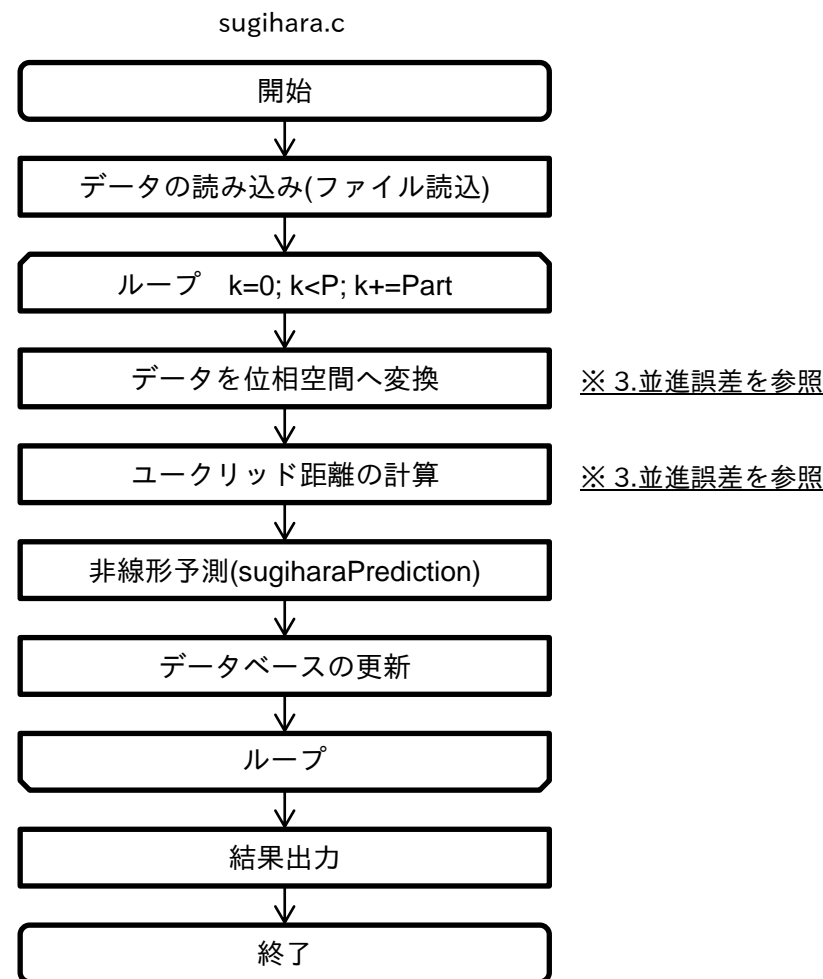
$$d_k = \|\mathbf{X}(t_p) - \mathbf{X}(t_k)\| \quad (5.2)$$

時系列データの前半をデータベース, 後半を予測結果との比較をするための参照データとして用いる。データベースから構築される位相空間内の軌道の予測される点を $\mathbf{X}(t_p)$ とし, Tステップ後に $\mathbf{X}(t_p+T)$ に移るものとする。近傍点も同様に考え, 式(5.1)で重みづけることで求まる。

・データベースの更新

本プログラムで予測を行うデータは決定論的特徴「短期予測可能・長期予測不可能」を有する。そのため, データベースを更新すれば常に短期予測を行える状態となる。

フローチャート



5 局所非線形予測法(Sugihara&Mayアルゴリズム)page2

予測例(ローレンツカオス)

ローレンツカオスを用いて予測した結果を図5.1,図5.2に示す.

※設定条件

NMAX : 5000, N : 2500, P : 2500, K : 4, D : 3, Tau : 16

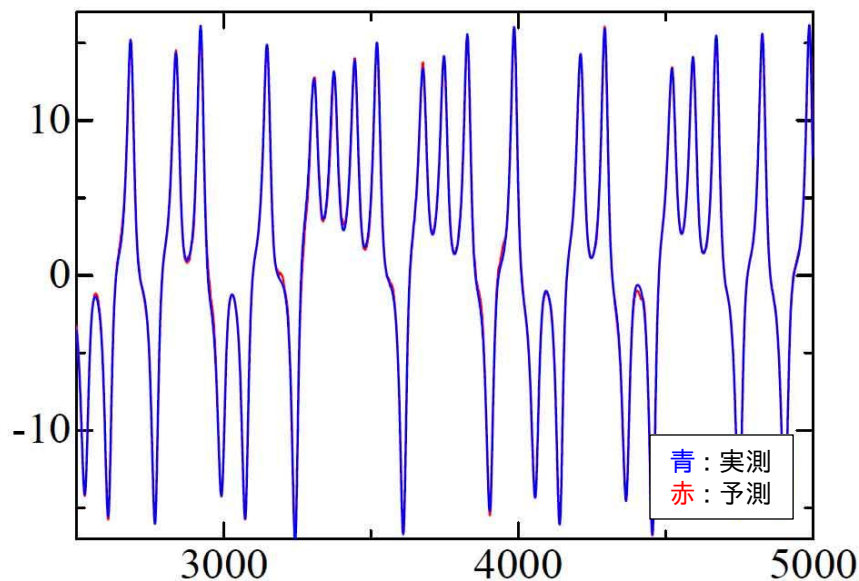


図5.1 ローレンツカオス(Part=1)

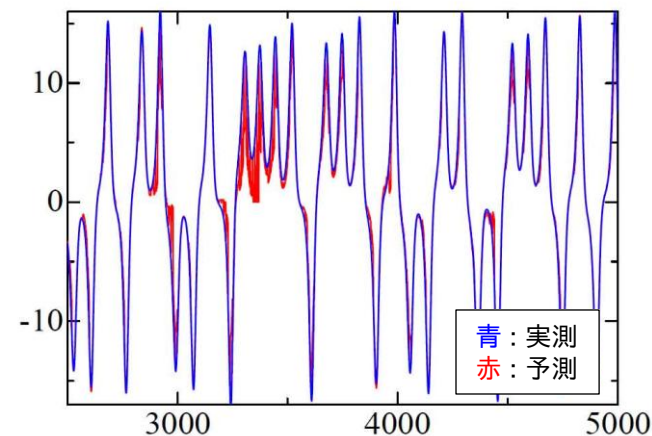


図5.2 ローレンツカオス(Part=10)

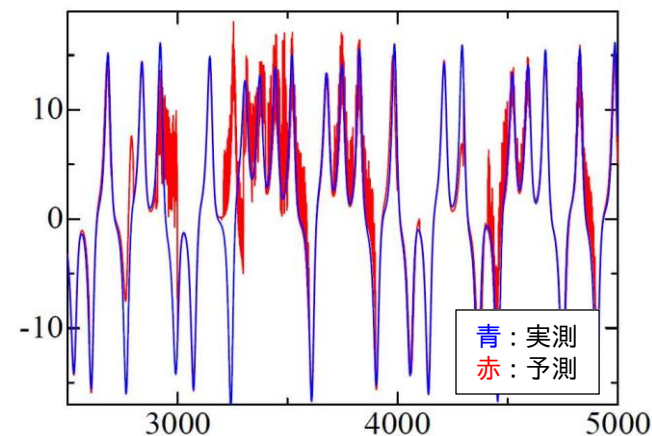


図5.3 ローレンツカオス(Part=100)

⑥ マルチフラクタル解析(ボックスカウンティング法)

概要

ボックスカウンティング法を用いたマルチフラクタル解析。

使用変数

| | |
|----------|------------------|
| NMAX: | 入力データ点数 |
| D: | 埋め込み次元 |
| Tau: | 遅れ時間 |
| dq: | 分割数(Qの刻み幅) |
| epsilon: | ボックスカウンティング法の分割数 |

計算過程

・ボックスカウンティング法

位相空間に埋め込まれた時系列データにボックスカウンティング法^[1]を用いる。

・確率測度の分配関数 Z_q の計算・出力

$$p_i = \frac{N_i}{N_N} \quad (6.1) \quad Z_q = \sum_{i=1}^{N_N} p_i(\varepsilon)^q \quad (6.2)$$

N_i は i 番目のセルにおけるデータ点数とし、確率測度 $p_i(\varepsilon)$ は式(6.1)、分配関数 $Z_q(\varepsilon)$ は式(6.2)である。

・最小二乗法(累乗近似)による $\tau(q)$ を計算

$$Z_q(\varepsilon) \propto \varepsilon^{\tau(q)} \quad (6.3)$$

※累乗近似の最小二乗法は計測工学レジメ3(2011年度)参照

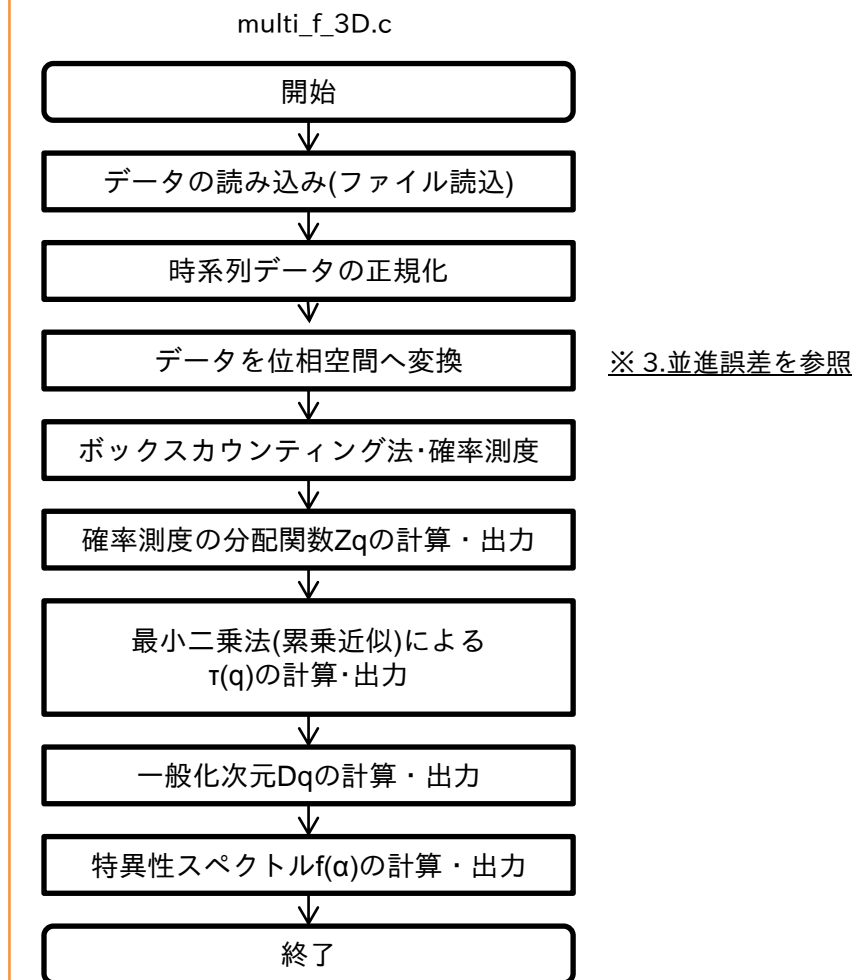
・一般化次元 D_q の計算

$$D_q = \frac{\tau(q)}{q-1} = \lim_{\varepsilon \rightarrow 0} \frac{\ln Z_q(\varepsilon)}{\ln \varepsilon} \quad (6.4)$$

・一特異性スペクトル $f(\alpha)$ の計算

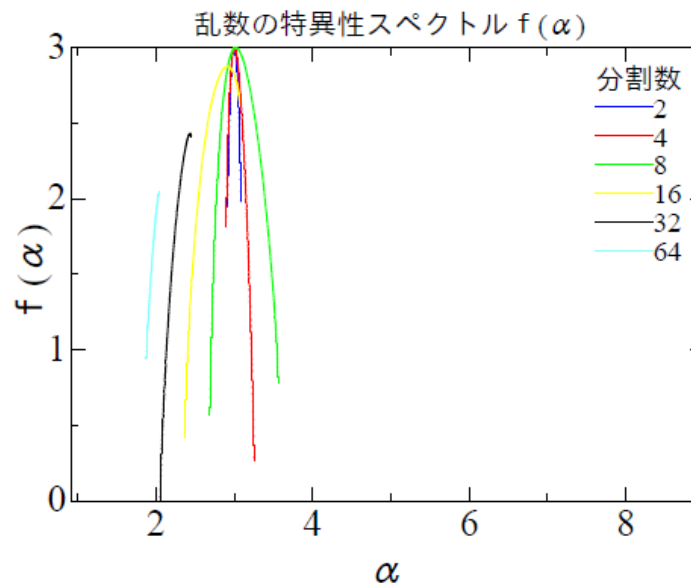
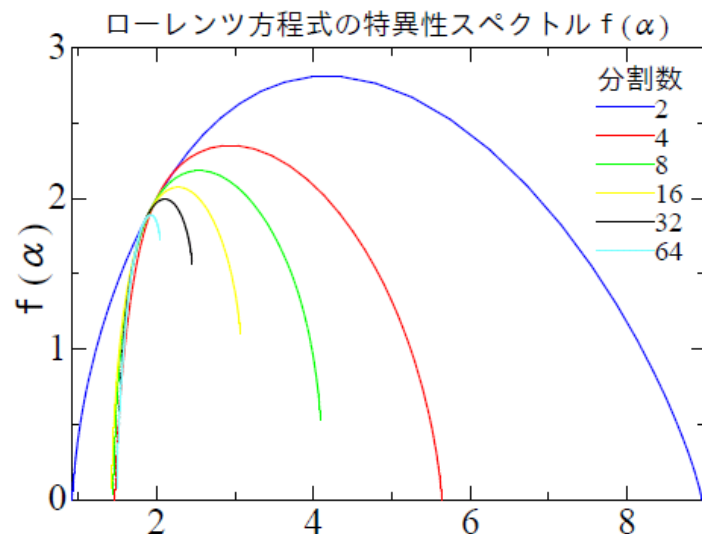
$$\alpha = \frac{d\tau(q)}{dq} \quad (6.5) \quad f(\alpha) = q\alpha - \tau(q) \quad (6.6)$$

フローチャート



⑥ マルチフラクタル解析(ボックスカウンティング法)

考察①: 分割数 ε の増加による影響(特異性スペクトル)



結果

乱数の特異性スペクトル $f(\alpha)$ より、飽和する分割数 ε を推定できる。

分割数 $\varepsilon = 16$ のとき、文献[1]に示されたローレンツ方程式の $f(\alpha) = 2.07^{[1]}$ に最も近い。

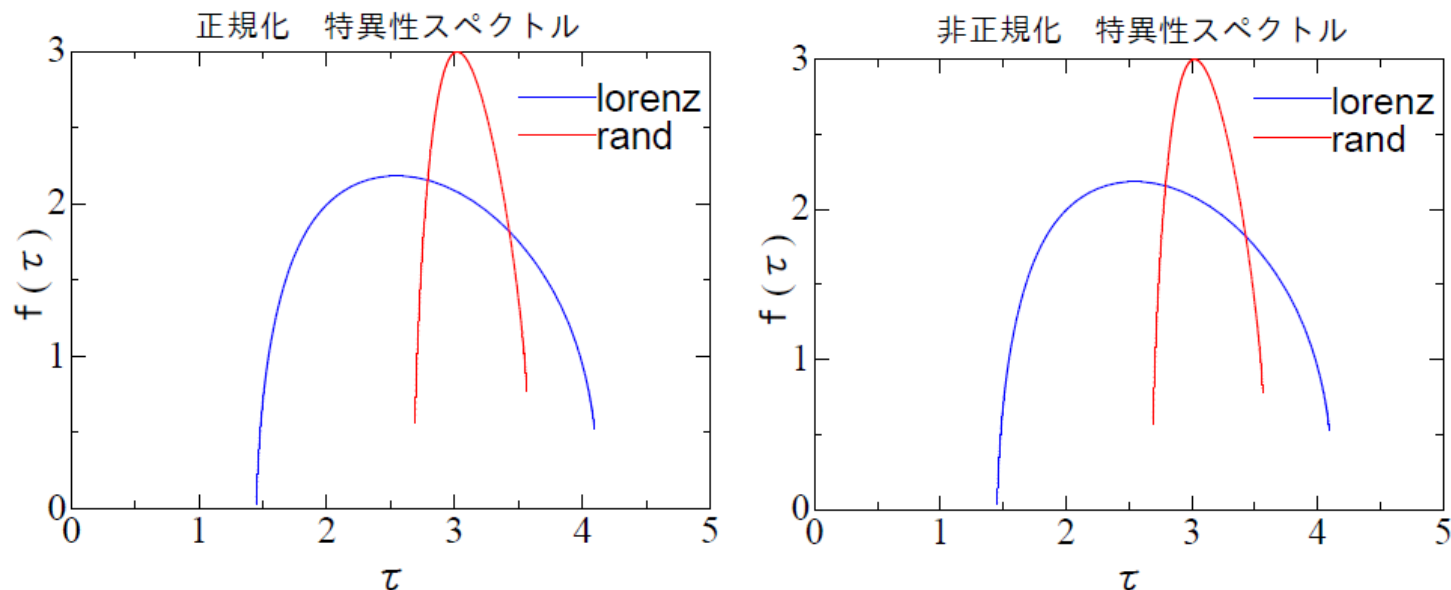
※randのデータ点数Nを増やせば、飽和しない領域が増加することを確認済 → 調査①参照

結論

乱数の飽和がしない分割数の最大値が適切である。($\varepsilon = 8-16$ に最適値ありと推定)

⑥ マルチフラクタル解析(ボックスカウンティング法)

考察②: 正規化による影響(特異性スペクトル)



結果

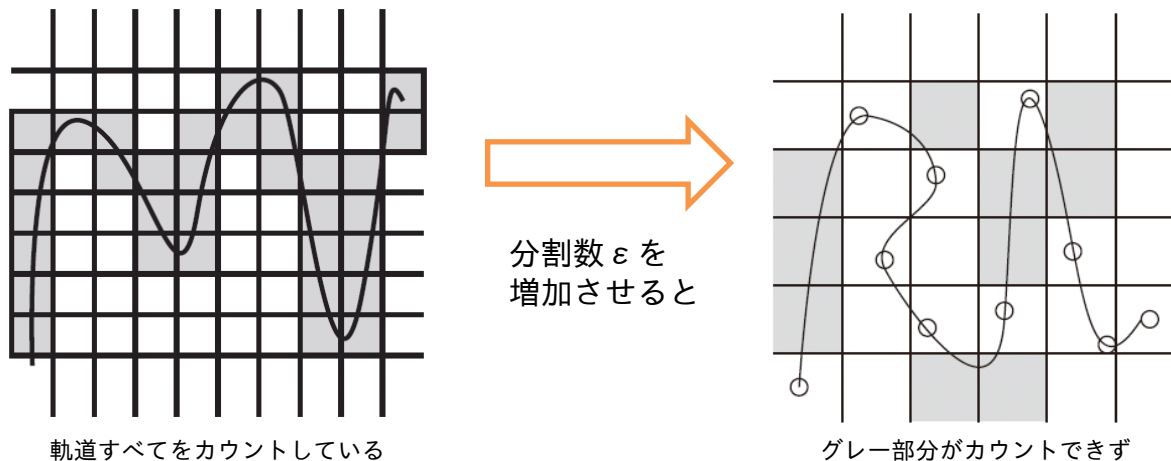
同じ結果が出力されており，正規化は結果に影響しないと確認できる。
正規化による浮動小数点の計算誤差は増加しない。

結論

他のプログラム(Waylandなど)を考慮し，正規化するプログラムが妥当と考える。

⑥ マルチフラクタル解析(ボックスカウンティング法)

調査①: 分割数 ε が飽和する原因



原因と対策

- ・ 離散データ間のカウントできない点が存在^[1]
 - 区分線形補間処理で解決する可能性
 - 2次元の場合、画像化して処理している事例^[2]あり
- ・ edge effect(境界の影響)の可能性
 - 本プログラムは正方形でカウントのため球状乱数では $D < 3$

結論

ボックスカウンティング法では問題点が多く、他の方法も試す必要がある
→ K.P.Harikrishnanらによる相関積分を用いた事例

[1] 山際謙太, フラクタル解析を用いた金属破断面の特性化と金属破壊機構解明への応用, 1999年度東京大学修士論文

[2] 塩谷昌典, ボックスカウント次元の計算, JAPLA 2003 シンポジウム資料

⑥ マルチフラクタル解析(ボックスカウンティング法)

参考: 相関積分を用いたマルチフラクタル解析

Chaos 19, 043129 (2009)

Computing the multifractal spectrum from time series: An algorithmic approach

K. P. Harikrishnan,^{1,a)} R. Misra,^{2,b)} G. Ambika,³ and R. E. Amritkar⁴

¹Department of Physics, The Cochin College, Cochin 682 002, India

²Inter University Centre for Astronomy and Astrophysics, Pune 411 007, India

³Indian Institute of Science Education and Research, Pune 411 021, India

⁴Physical Research Laboratory, Navarangapura, Ahmedabad 380 009, India

(Received 16 August 2009; accepted 19 November 2009; published online 18 December 2009)

We show that the existing methods for computing the $f(\alpha)$ spectrum from a time series can be improved by using a new algorithmic scheme. The scheme relies on the basic idea that the smooth convex profile of a typical $f(\alpha)$ spectrum can be fitted with an analytic function involving a set of four independent parameters. While the standard existing schemes [P. Grassberger *et al.*, J. Stat. Phys. 51, 125 (1989); A. Chhabra and R. V. Jensen, Phys. Rev. Lett. 62, 1327 (1989)] generally

概要

- ・ ボックスカウンティング法ではなく, 相関積分(generalized correlation sum)
- ・ edge effect(境界の影響)等を指摘
- ・ 特異性スペクトル $f(\alpha)$ を求めるとき, D_q を用いない様子

提案

- 過去に試していなければ, 6月後半に期限付きで試す.
- 2009年に本論文は報告されている.
- 現在用いている解析手法は2003年の論文^[1]を基にしている.

※全文理解をしておらず, 一部誤った解釈をしている可能性あり

[1]大西立顕, 合原一幸, 株価変動のマルチフラクタル解析, 電子情報通信学会技術研究報告. NLP, 非線形問題 101(723), 93-100

7 サイクルサロゲートデータ

概要

サイクルサロゲートデータを生成する。

※ N=MでRSサロゲートデータ

使用変数

N: 入力データ点数

M: データ分割数

計算過程

・データのブロック分割(M分割)

入力された時系列データxをM分割する。

(例) N=100, M=4のとき

x[0] - x[99]

x[0] - x[24]

x[25] - x[49]

x[50] - x[74]

x[75] - x[99]

・データをブロック単位でシャッフル

(例)

x[50] - x[74]

x[25] - x[49]

x[75] - x[99]

x[0] - x[24]

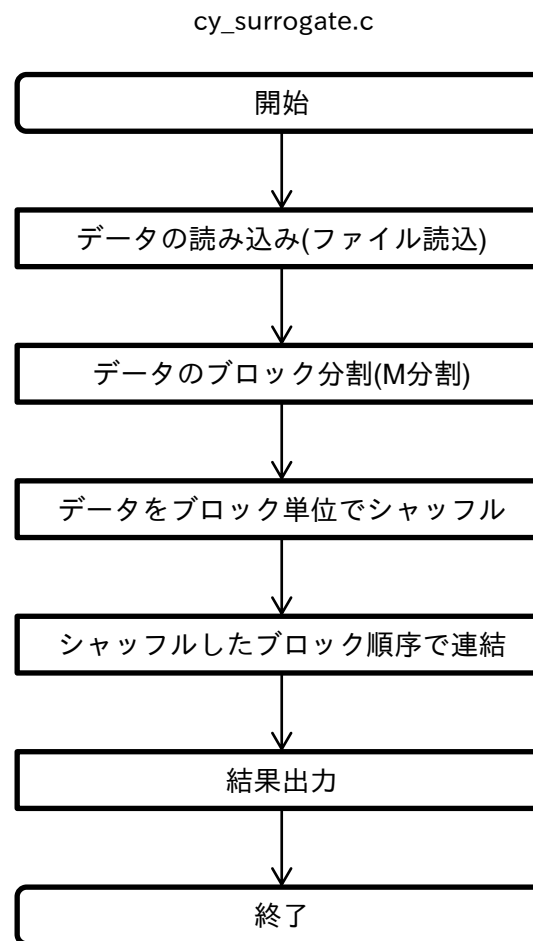
※本プログラムではvoid shuffle(int *array, int N)を用意

・シャッフルしたブロック順序で連結

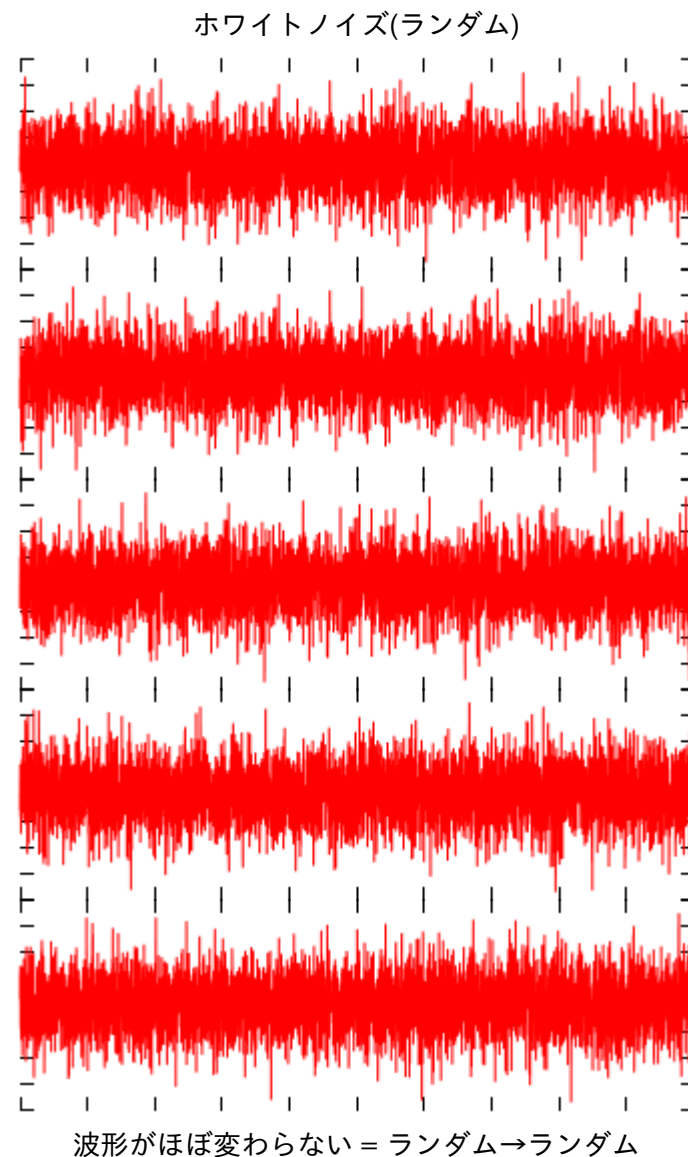
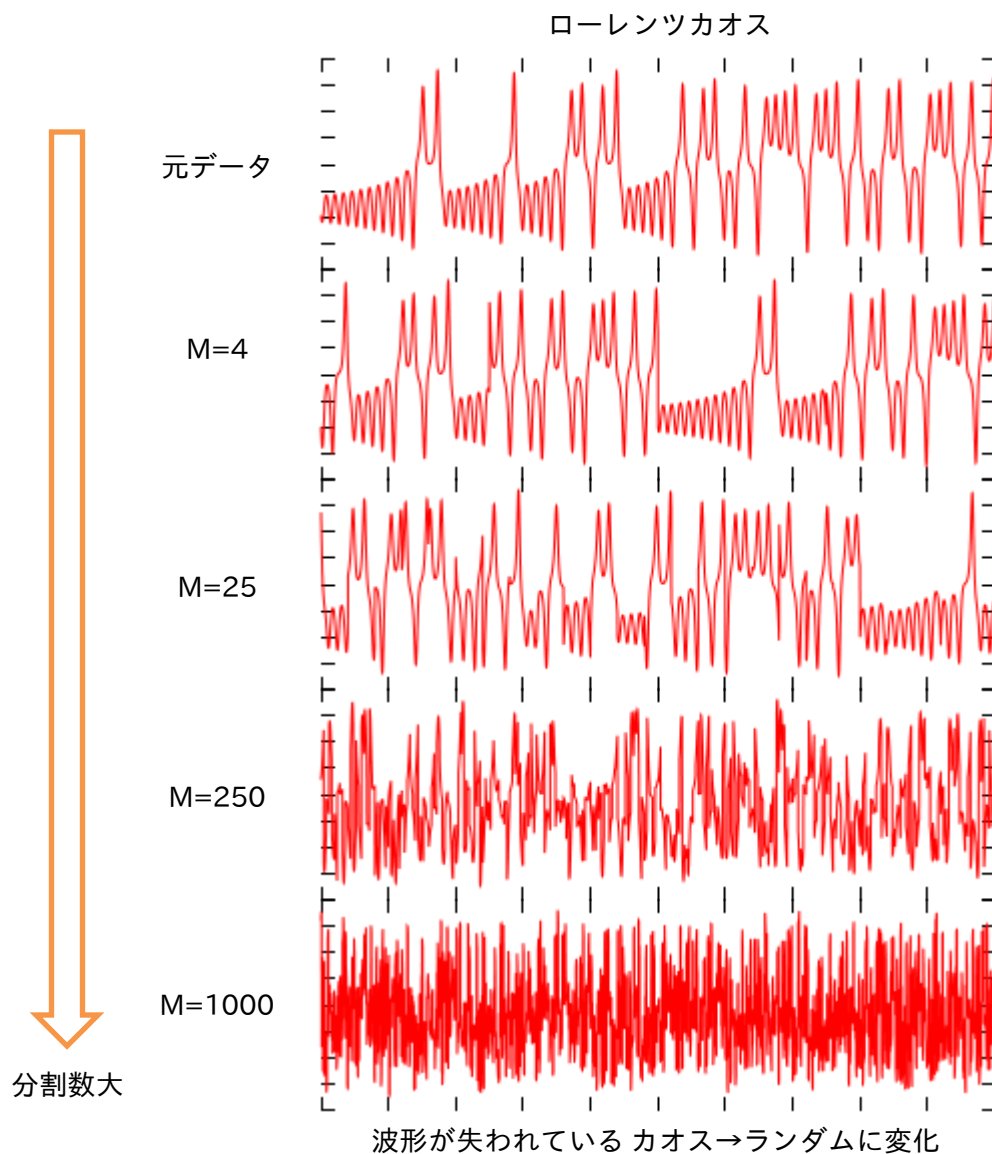
(例)

x[50]-x[74], x[25]-x[49], x[75]-x[99], x[0]-x[24]

フローチャート



7 サイクルサロゲートデータpage2



7 FTサロゲートデータ

概要

「観測された時系列信号は、時間的に線形相関を持つ確率的データである。」という帰無仮説では、時系列データは相関関数によってのみ特徴づけられる。ウィナー-ヒンチンの定理により、「相関関数とパワースペクトル密度は等価」である。これらに従い、FTサロゲートデータを作成する。

使用変数

x: 時系列データ(オリジナルデータ)
N: 時系列データの点数

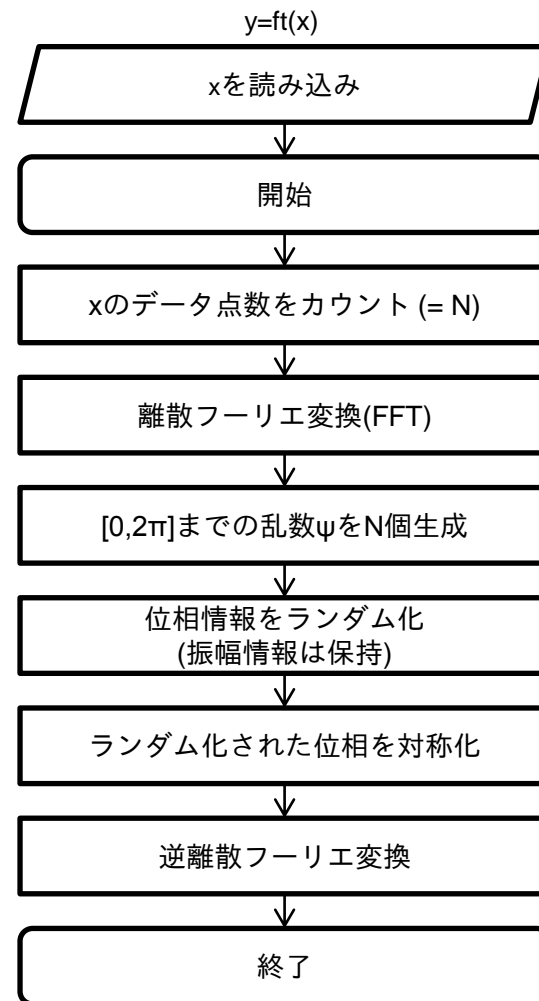
計算過程 (jは複素数とする)

- ・ 離散フーリエ変換
時系列データ $x(n)$ のフーリエ変換 $X(n)=A(n)e^{j\phi(n)}$ を求め、パワースペクトルを推定する。
- ・ 位相情報をランダム化 (振幅情報は保持)
1. $[0, 2\pi]$ に分布する乱数 $\Psi(n)$ をN個生成
2. $X(n)$ に $e^{j\Psi(n)}$ を掛け、位相情報をランダム化($=X'(n)$)
- ・ ランダム化された位相を対称化
逆フーリエ変換した値を実数とするため、 $X'(n)$ を対称化^[1]

備考

- ・ [用語説明] 帰無仮説
最終的に間違いだと思いたい仮説
→自分の証明したい事実とは反する仮説
- ・ 使用しているMATLAB関数
i, j: 虚数, fix(): 切り捨て, flipplr(): 列を反転, conj(): 複素共役,
real(): 複素数の実部を出力, fft(): 高速フーリエ変換,
ifft(): 逆高速フーリエ変換, [A, B, C]: A, B, C行列を結合

フローチャート



[1]平井慎一, 機械システム学のための数値計算法, (2008), p121, コロナ社, p121

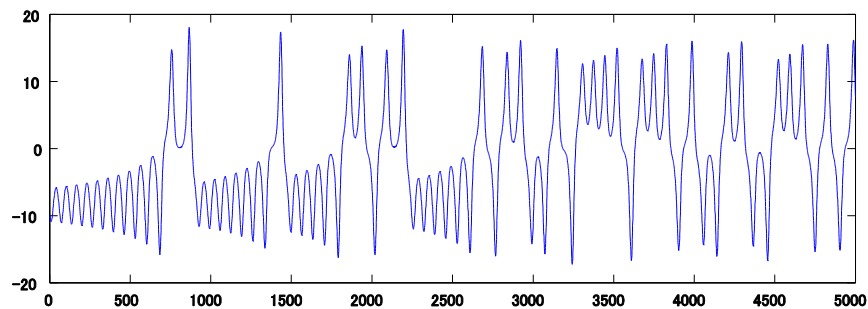
[2]合原一幸, カオス時系列解析の基礎と応用, (2000), p257-259, 産業図書

7 FTサロゲートデータ page2

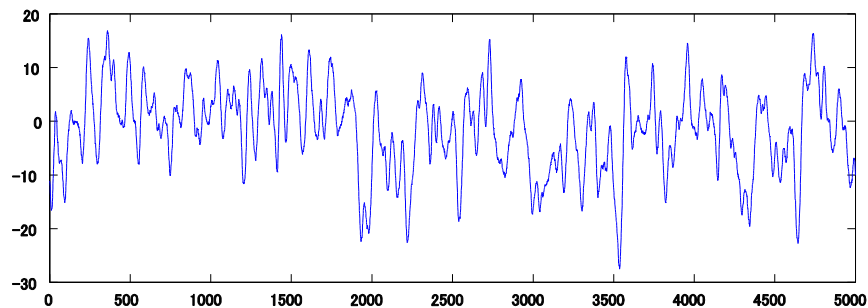
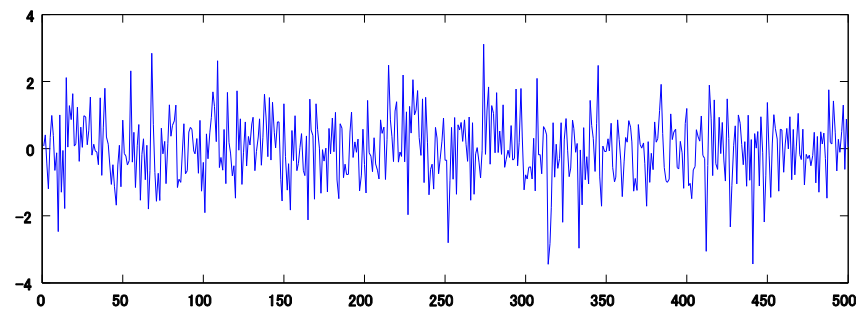
FTサロゲートの例

上段にオリジナルデータ, 下段にFTサロゲートデータを示す。

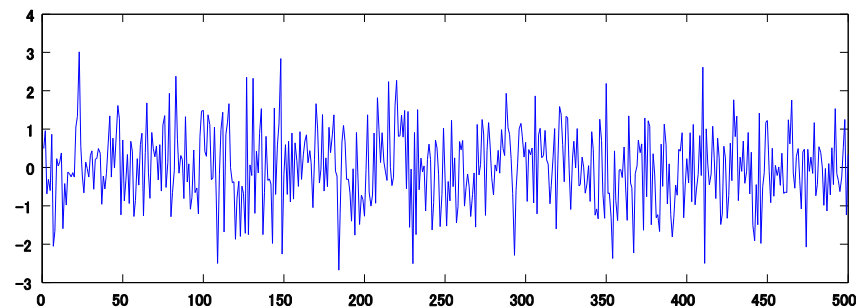
ローレンツカオス



ホワイトノイズ



波形が失われている カオス→ランダムに変化



波形がほぼ変わらない = ランダム→ランダム

7 AFTサロゲートデータ

概要

「観測された時系列信号は、線形確率過程から作り出されたが、観測する際に静的な単調非線形変換を施されたことにより得られたデータである。」という帰無仮説で考える。

使用変数

x: 時系列データ(オリジナルデータ)
N: 時系列データの点数

計算過程

・正規乱数(ガウシアン乱数)を生成

時系列データ $x(n)$ のフーリエ変換 $X(n)=A(n)e^{i\Phi(n)}$ を求め、パワースペクトルを推定する。

・ランクオーダーによる並び替え = `samerank(target, ref)`

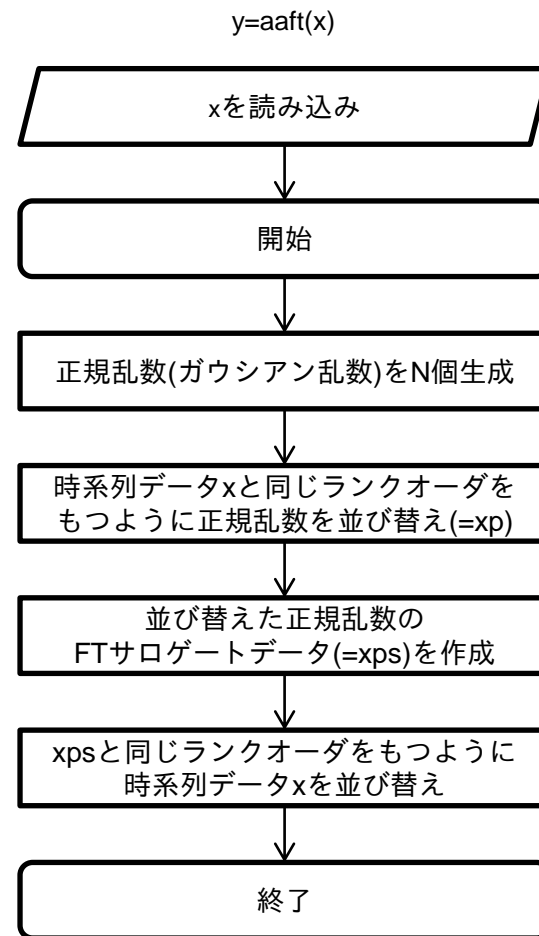
あるデータ(=参照データ)と同じランクオーダー順に対象データを並び替える。以下にアルゴリズムを示す。

1. 対象データ(並び替えるデータ)を昇順ソート
2. 参照データをインデックス(=sri)付きで昇順ソート
3. sriをインデックス(=srii)付きでさらに昇順ソート
4. 対象データをsrii順にソート

(例)参照データの並び替え(N=5のとき)

| sri | value | | srii | sri | value | | srii | sri | value |
|-----|-------|--|------|-----|-------|--|------|-----|-------|
| 1 | 100 | | 1 | 3 | 1 | | 5 | 1 | 100 |
| 2 | 10 | | 2 | 5 | 4 | | 3 | 2 | 10 |
| 3 | 1 | | 3 | 2 | 10 | | 1 | 3 | 1 |
| 4 | 20 | | 4 | 4 | 20 | | 4 | 4 | 20 |
| 5 | 4 | | 5 | 1 | 100 | | 2 | 5 | 4 |

フローチャート

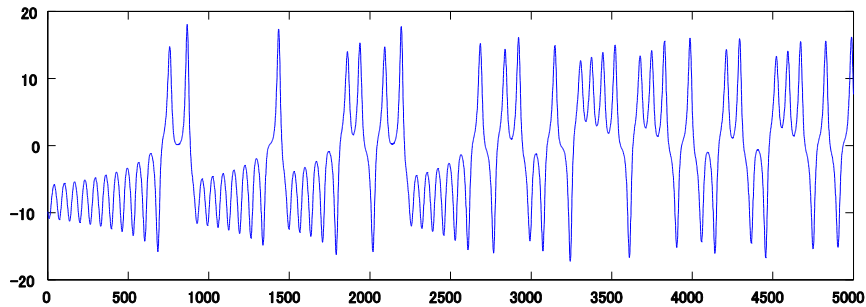


7 AAFTSロゲートデータpage2

AAFTサロゲートの例

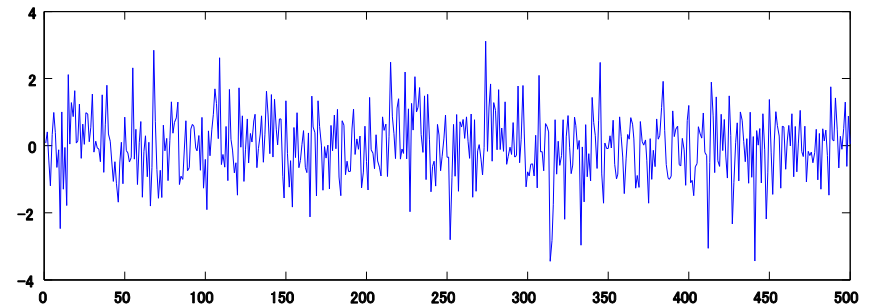
上段にオリジナルデータ, 下段にAAFTサロゲートデータを示す。

ローレンツカオス



波形が失われている カオス→ランダムに変化

ホワイトノイズ



波形がほぼ変わらない = ランダム→ランダム

7 IAAFTサロゲートデータ

概要

IAAFT(Iterative Amplitude Adjusted Fourier Transform) アルゴリズムは平均・分散・頻度分布を保存する。またAAFTと比べ、自己相関・パワースペクトルの保存が優れている。約7回、(4)-(6)を繰り返せば十分であるとされる。^[1]

使用変数

x: 時系列データ(オリジナルデータ)
N: 時系列データの点数
M: 繰り返し数(The number of iterations)

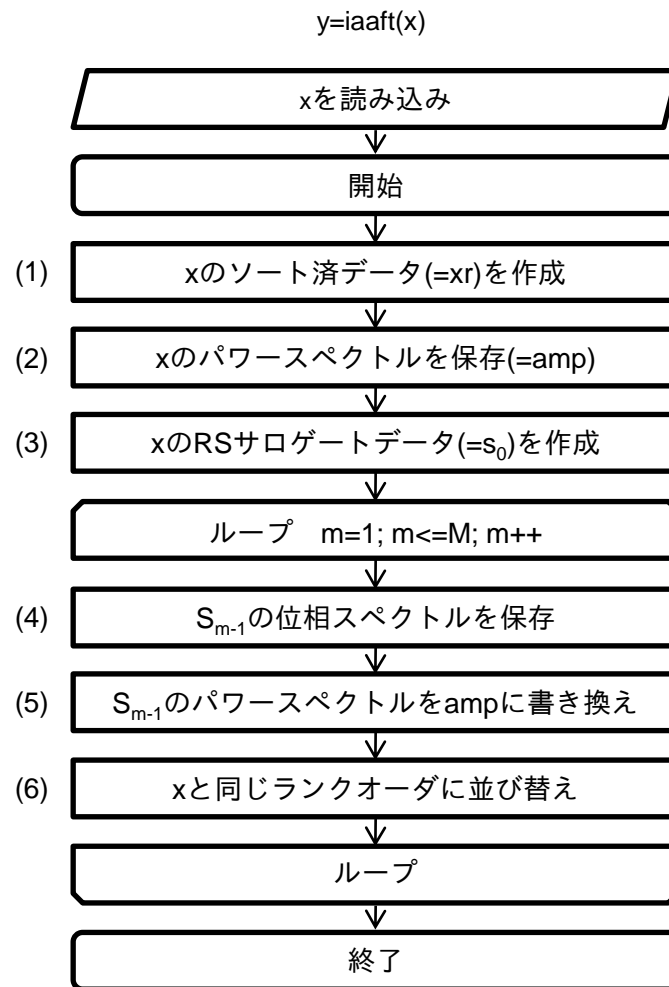
計算過程

- ・RSサロゲートデータ
※RSサロゲートデータ参照
- ・[STEP1] S_{m-1} の位相を保存、パワースペクトルをampに書き換え
標本分布は変化するが、パワースペクトルと自己相関をオリジナルに近づける。
- ・[STEP2] xと同じランクオーダーに並び替え
オリジナルと同じランクオーダーに並び替えれば、標本分布を一致させることができる。アルゴリズムはsamerank(target, ref)関数との違いは、オリジナルxのソート済データxrを事前に用意し効率化していることである。

備考

- ・STEP1のときに繰り返しを終えた場合について
STEP1後のデータは帰無仮説を棄却することが難しく、標本分布を一致させたSTEP2後のデータが適しているとされる。^[2]
- ・池口徹氏のアルゴリズムとの相違
文献[1]と日本語文献[3]の手法は異なる。しかし、文献[3]の著者と同じである英語文献[4]の手法は文献[1]と同じである。以上より、本研究では文献[1][2][4]の手法を用いる。

フローチャート



[1]Thomas Schireiber and Andreas Schmitz. "Improved surrogate data for nonlinearity tests". Physical Review Letters, Vol. 77, No. 4, pp. 635-638, 1996.

[2]D. Kugiumtzis, "Test your surrogate data before you test for nonlinearity", Physical Review E 60 (3) (1999) 2806-2816

[3]合原一幸, カオス時系列解析の基礎と応用, (2000), p268-271, 産業図書; [4] M. Hasegawa and T. Ikeguchi, Proc. IEEE. Int. Symp. Circ. Syst. (ISCAS) 3, 599 (2002).

7 [参考] IAAFTサロゲートデータ (池口日本語文献アルゴリズム)

概要

IAAFTのアルゴリズムはAAFTのアルゴリズムとデータ処理内容はほぼ同じである。相違点は「正規乱数」ではなく「オリジナルデータのFTサロゲートデータ」を用い、AAFTアルゴリズムを複数回繰り返す。これは、オリジナルデータのパワースペクトルに作成データを近づけるためである。約7回の繰り返しで十分であるとされる。^[1]

使用変数

x: 時系列データ(オリジナルデータ)
 N: 時系列データの点数
 M: 繰り返し数(The number of iterations)
 xft[0]: オリジナルデータのFTサロゲートデータ

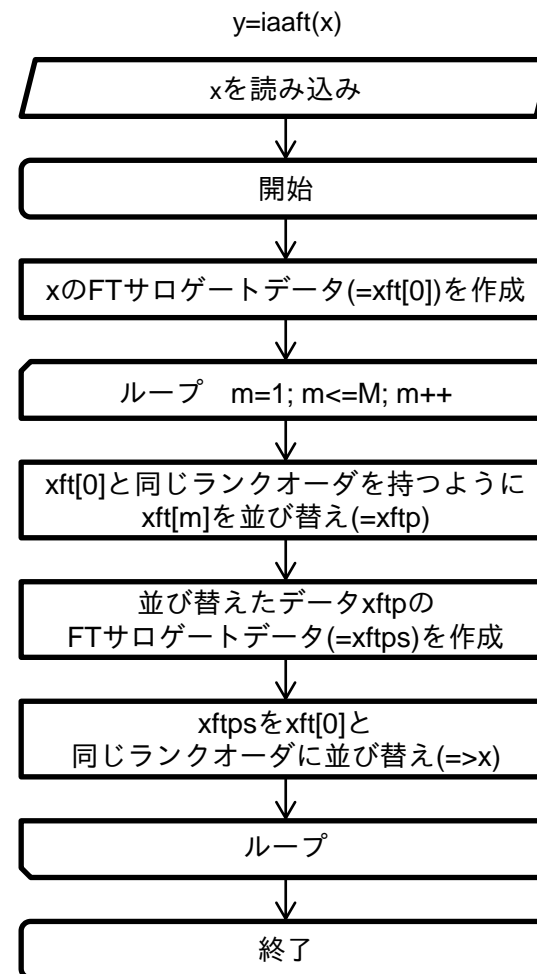
計算過程

- ・ FTサロゲートデータ
 ※FTサロゲートデータ参照
- ・ ランクオーダーによる並び替え
 samerank(target, ref)関数を作成済。 (※AAFTサロゲート参照)

AAFTサロゲートとの相違点

- ・ 正規乱数(AAFT)と時系列データ(IAAFT)
 AAFTでは正規乱数を用いてランクオーダーを並び替えていたが、IAAFTでは時系列データ(オリジナルデータ)のパワースペクトルを参照して並び替えている。
 ※FTアルゴリズムはパワースペクトルの振幅情報を保存
- ・ 繰り返しの有無
 IAAFTではi番目に作成したサロゲートデータをxft[0]を参照し、AAFTアルゴリズムで作成したデータをi+1番目とする。

フローチャート



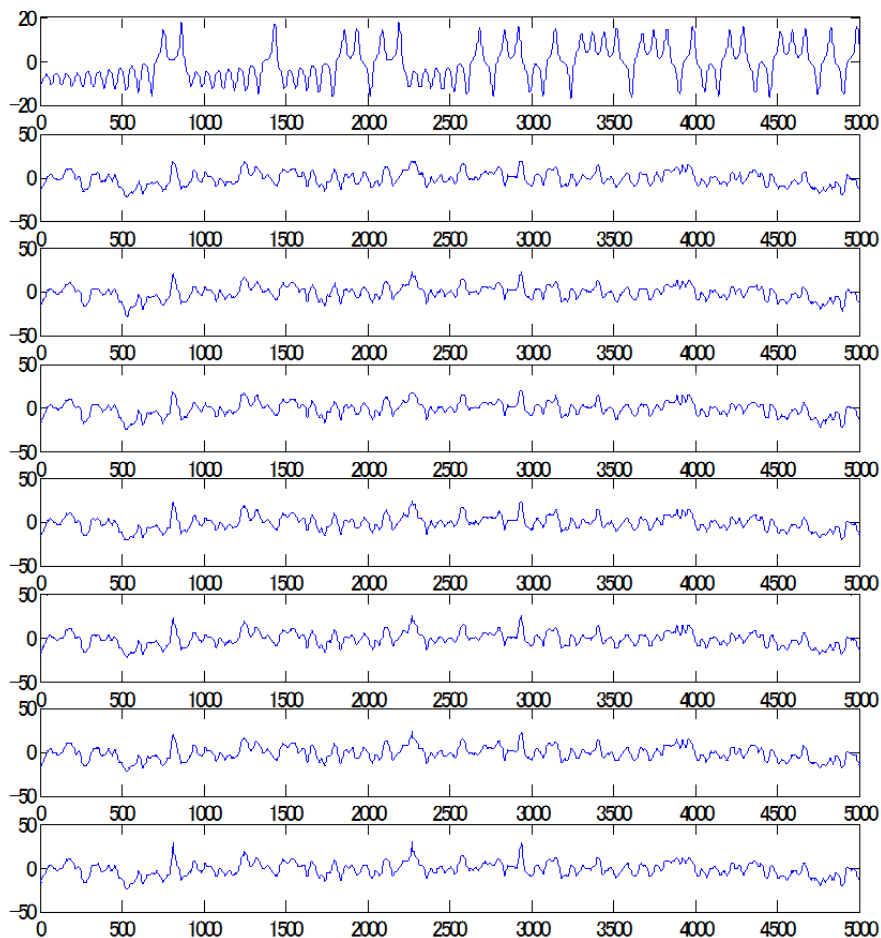
[1]合原一幸, カオス時系列解析の基礎と応用, (2000), p268-271, 産業図書

7 [参考] IAAFTサロゲートデータ (池口日本語文献アルゴリズム)

iaaft.m(ver 0.0.1)の簡易テスト結果 (※より正確なテストの検討要)

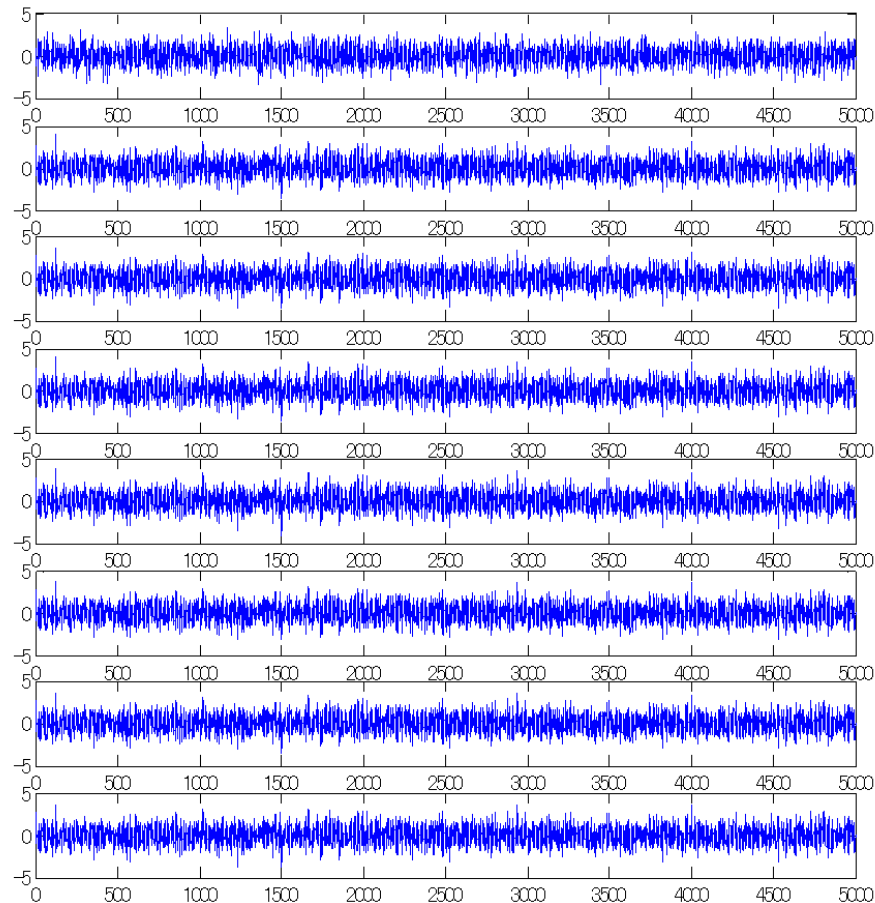
最上段にオリジナルデータ，その下段より順に1回，2回，・・・，7回の繰り返しによって生成されたサロゲートデータを示す。

ローレンツカオス



波形が失われている カオス→ランダムに変化

ホワイトノイズ



波形がほぼ変わらない = ランダム→ランダム

8 動径基底関数ネットワーク

概要

動径基底関数ネットワークを用いて学習し予測を行う。
 予測ステップ変更可, 学習位置変更可,
 ガウス関数を基底関数として使用。

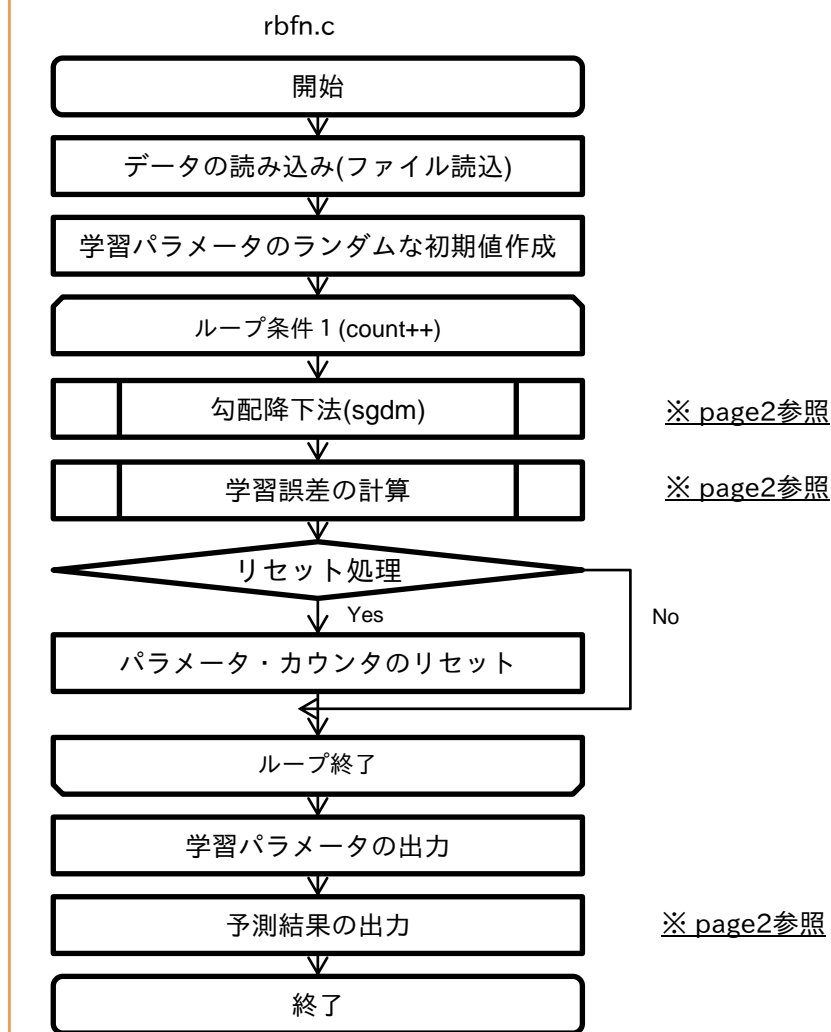
使用変数

NMAX: 入力データ点数
 N: 学習データ点数
 M: 学習パラメータ点数
 D: 埋め込み次元
 Tau: 遅れ時間
 Start: 学習開始データ番号(最小値=1)

計算過程

- ・ **学習パラメータ初期値作成**
 不作為に学習パラメータ C , α , θ の初期値を生成する。
 ※ $D=3$ のとき, $C \cdot \alpha$ は10個, θ は30個生成
- ・ **ループ条件1**
 $\text{count} < \text{LOOP}$ (繰り返し処理) または 最大繰り返し数達成
 または 目標学習誤差に到達 または 終了設定時間経過
 ※ 4つのループ終了条件
- ・ **リセット処理**
 $E > E_{\text{max}}$ かつ $\text{count} > \text{OVERLIMIT}$
 (最大学習誤差以下にならず, 限度回数を越えたとき)
 または
 $E_{\text{min}} > \text{ErrAgree}$ かつ $\text{count} > \text{AGREE_LIMIT}$
 (許容学習誤差以下にならず, 限度回数を越えたとき)
- ・ 勾配降下法・学習誤差の計算・予測結果の出力
 page2参照

フローチャート



8 動径基底関数ネットワークpage2

勾配降下法(実際の学習則の計算)

$$\begin{cases} \frac{\partial E}{\partial C_j} = -\frac{2}{N} \sum_{t=1}^N \exp(-\alpha_j |\mathbf{p} - \boldsymbol{\theta}_j|^2) \left\{ p(t+T) - \sum_{m=1}^M C_m \exp(-\alpha_m |\mathbf{p} - \boldsymbol{\theta}_m|^2) \right\} \\ C(s+1) = C(s) - \eta_C \frac{\partial E}{\partial C} \end{cases}$$

$$\begin{cases} \frac{\partial E}{\partial \theta_j} = -\frac{4C_j\alpha_j}{N} \sum_{t=1}^N |\mathbf{p} - \boldsymbol{\theta}_j|^2 \exp(-\alpha_j |\mathbf{p} - \boldsymbol{\theta}_j|^2) \left\{ p(t+T) - \sum_{m=1}^M C_m \exp(-\alpha_m |\mathbf{p} - \boldsymbol{\theta}_m|^2) \right\} \\ \theta(s+1) = \theta(s) - \eta_\theta \frac{\partial E}{\partial \theta} + \delta(s)\xi(s) \end{cases}$$

$$\begin{cases} \frac{\partial E}{\partial \alpha_j} = -\frac{2C_j}{N} \sum_{t=1}^N |\mathbf{p} - \boldsymbol{\theta}_j|^2 \exp(-\alpha_j |\mathbf{p} - \boldsymbol{\theta}_j|^2) \left\{ p(t+T) - \sum_{m=1}^M C_m \exp(-\alpha_m |\mathbf{p} - \boldsymbol{\theta}_m|^2) \right\} \\ \alpha(s+1) = \alpha(s) - \eta_\alpha \frac{\partial E}{\partial \alpha} + \delta(s)\xi(s) \end{cases}$$

の順に計算する。s:時間ステップ, \mathbf{p} :入力層, M:基底関数の個数, $\alpha \cdot \boldsymbol{\theta} \cdot C$:学習パラメータ, E:誤差(残差), η :学習率, N:データ数である。また, $\boldsymbol{\theta} \cdot \alpha$ 局所最適解を持ち, 確率的勾配降下法によって学習誤差の改善が停滞するとき, ノイズを加えることで脱出を促す, ξ :乱数, δ :ノイズ発生スイッチである。

学習率 η は経験的に0.1, 基底関数の個数Mは10, 遅れ時間 τ を1としたほうがよい。

学習誤差の計算

$$E = \frac{1}{N} \sum_{t=1}^N \left\{ p(t+T) - \sum_{m=1}^M C_m \exp(-\alpha_m |\mathbf{p} - \boldsymbol{\theta}_m|^2) \right\}^2$$

を計算する。

予測結果の出力

$$p'(t+1) = C_m \exp \left(-\alpha_m \sum_{d=1}^D |\mathbf{p} - \boldsymbol{\theta}_{m,d}|^2 \right)$$

を順に計算し予測を得る。 p' :出力値である。

ロジスティック写像(a=4)の予測結果

予測に用いる条件をM=10, D=3, $\tau=1$, T=1, P=1とし, 前半1000点を学習し, 後半1000点を予測した結果を図8.1に示す。おおよそ予測できていると考えられる。

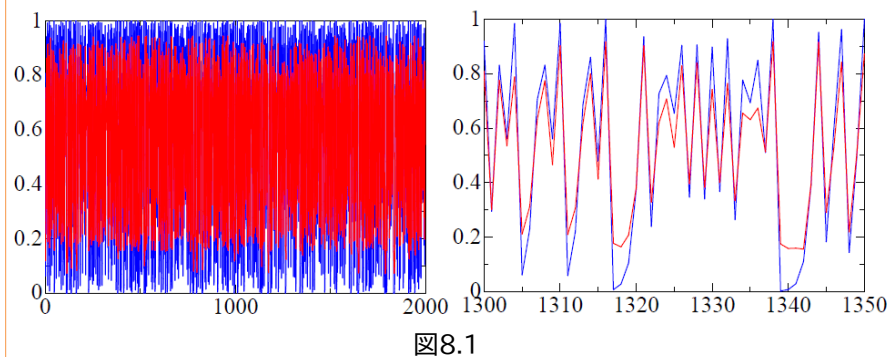


図8.1

A 順列エントロピー (Permutation Entropy)

概要

規格化された順列エントロピーの計算を行う。

使用変数

x: 時系列データ(オリジナルデータ)
 Tau: 遅れ時間
 D: 埋め込み次元(Embedding Dimension)
 N: 時系列データの点数
 freq: 実現度数
 p: 相対実現度数

計算過程

・ 順列の実現度数

次ページ参照

・ 相対実現度数

度数を相対度数にする計算に同じ。

$$p(i) = \frac{freq(i)}{N_N} \quad (A.1)$$

・ 順列エントロピー

順列エントロピーは式(A.2)で求められる。BandtとPompeは、 $H(D)$ がDに対して線形に増加することを指摘し、 $\log_2 D!$ で割り規格化(正規化)^[1]した。本研究にでも規格化する。

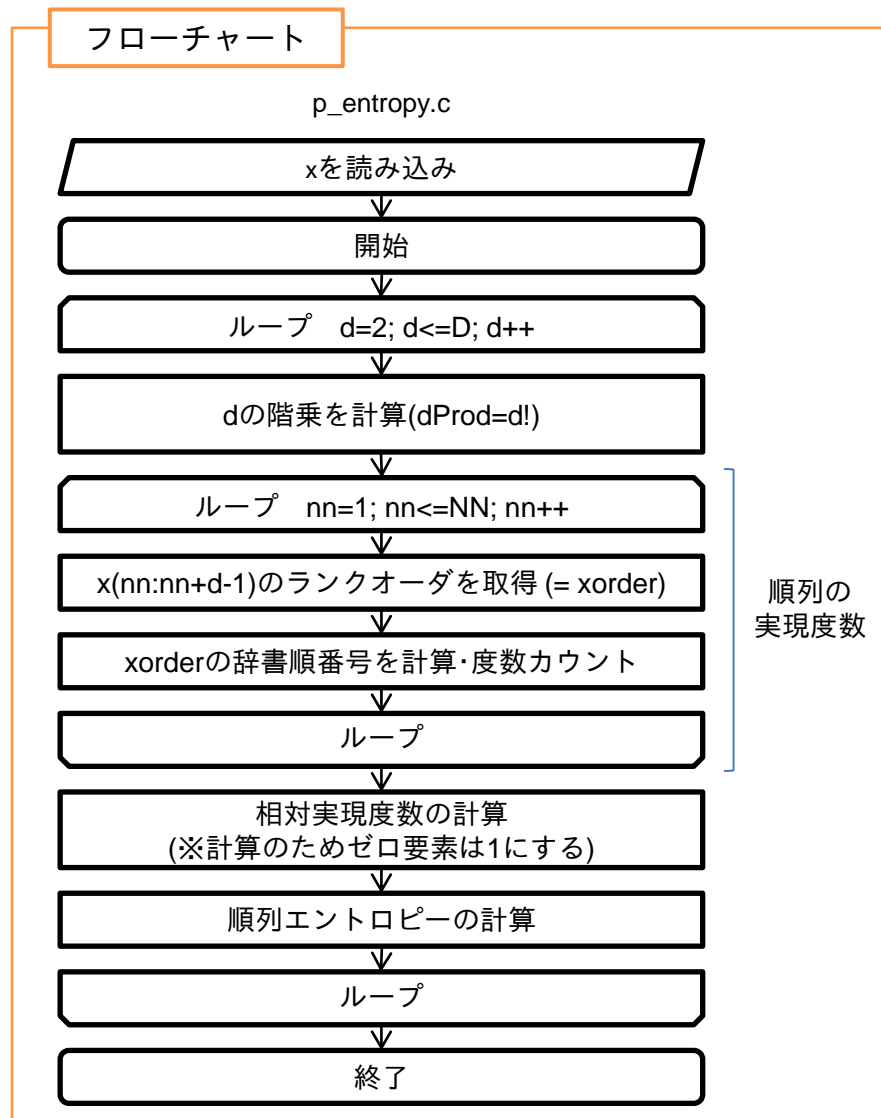
※文献[2]では式(A.4)のpermutation entropy symbolを規格化しているが、意図が不明確であり、本研究では採用しない。

$$H(D) = - \sum_{i=1}^{N_N} p(i) \log_2 p(i) \quad (A.2)$$

$$H_{normal}(D) = H(D) / \log(D!) \quad (A.3)$$

$$h(D) = \frac{H(D)}{D - 1} \quad (A.4)$$

フローチャート



[1] Christoph Bandt, Permutation Entropy: A Natural Complexity Measure for Time Series, PhysRevLett.88.174102

[2] 宮野尚哉, 藤ランゲルハンス島β細胞集団のカオス同期における順列エントロピー, 信学技報 NLP2002-103(2003-02)

A 順列エントロピー (Permutation Entropy) page2

順列の実現度数

計算例) D=4, xorder = [3 4 1 2] のとき

1. 一番左の数字(=A)が3であるから、2の最後までの個数を計算

$$na = (4-1)! * 2 = 12$$

2. 左から2番目の数字(=B)が4であるから、2の最後までの個数を計算

→ 現時点で残っている数字は [1 2 4] の3つ

すなわち、1は1つ目、2は2つ目、4は3つ目と考えられる

$$nb = na + (4-2)! * 2 = na + 4 = 16$$

3. 左から3番目の数字(=C)が1であるから、nbから考える

→ 現時点で残っている数字は [1 2] の1つ

すなわち、1は1つ目、2は2つ目と考えられる

$$nc = na + nb + (4-3)! * 1 = na + nb + 1 = 17$$

Dがさらに大きくなった場合も同様に考えることができる。
 ポイントは、毎回残っている数字の順番を用いることである。
 ※ 高校数学(数A)の考え方。下記サイトでも紹介されている。

- ・ 特定の順列を見つける方法 (サイト名: 私的数学塾)

http://www004.upp.so-net.ne.jp/s_honma/urawaza/permutation.htm

| n | A | B | C | D |
|-----|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 1 | 2 | 4 | 3 |
| 3 | 1 | 3 | 2 | 4 |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | 2 | 1 | 3 | 4 |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | 3 | 1 | 2 | 4 |
| 14 | 3 | 1 | 4 | 2 |
| 15 | 3 | 2 | 1 | 4 |
| 16 | 3 | 2 | 4 | 1 |
| 17 | 3 | 4 | 1 | 2 |
| ... | | | | |
| 24 | 4 | 3 | 2 | 1 |

(D-1)!個

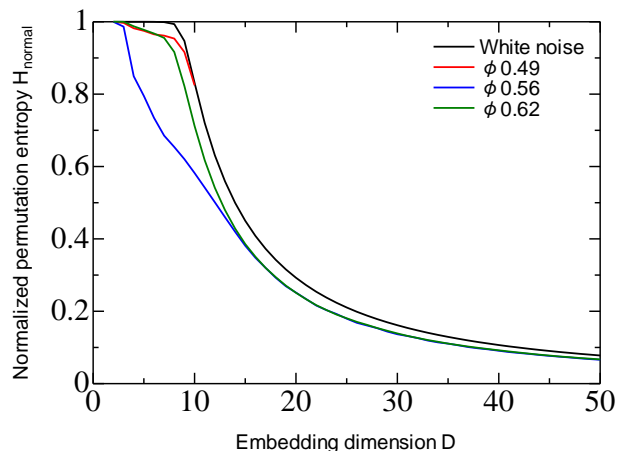
(D-1)!個

na

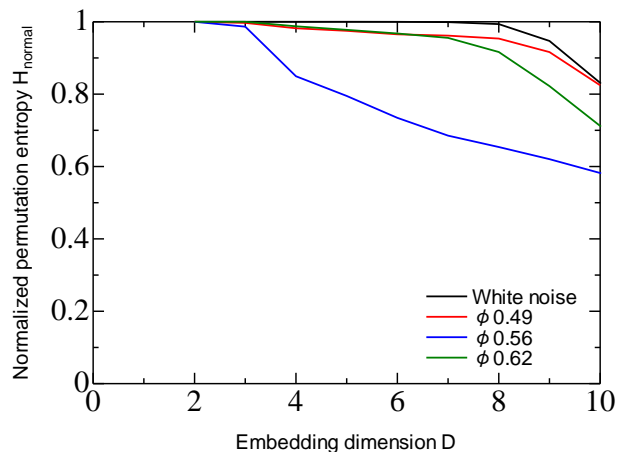
nb

A 順列エントロピー (Permutation Entropy) page2

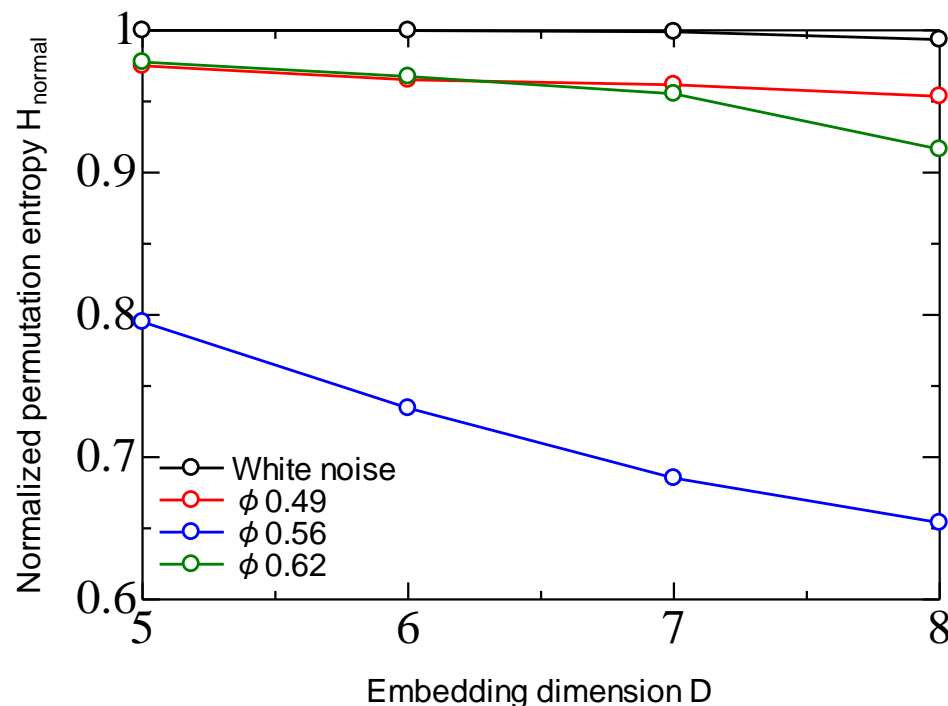
次元Dの設定について



図A.1 PEの最大次元D=50



図A.2 PEの最大次元D=10



図A.3 PEの最大次元D=8

結論

論文^{[1][2]}で指摘のinteresting theoretical problem※を確認し、
D=5,6,7程度が妥当. ※[1]が示した文献を確認できず

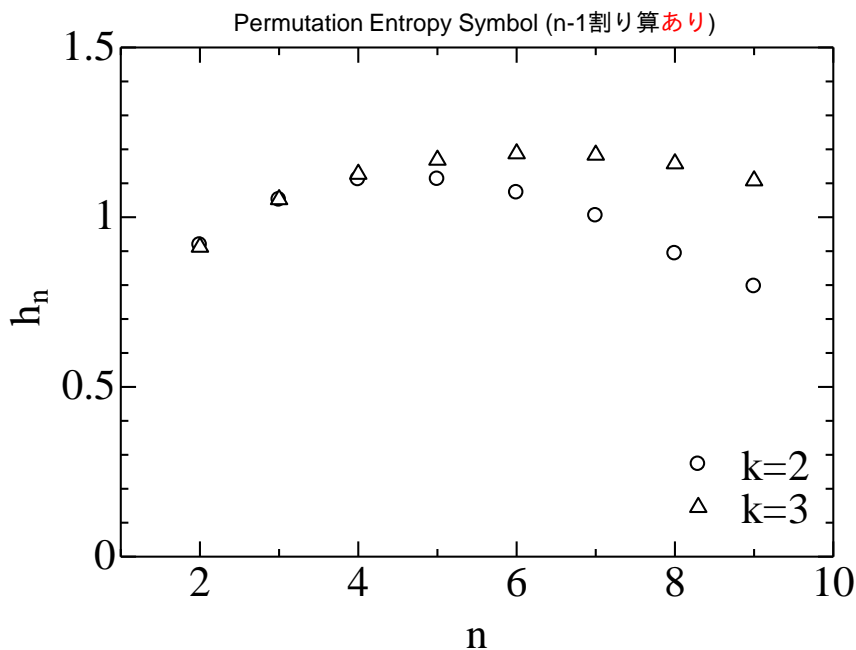
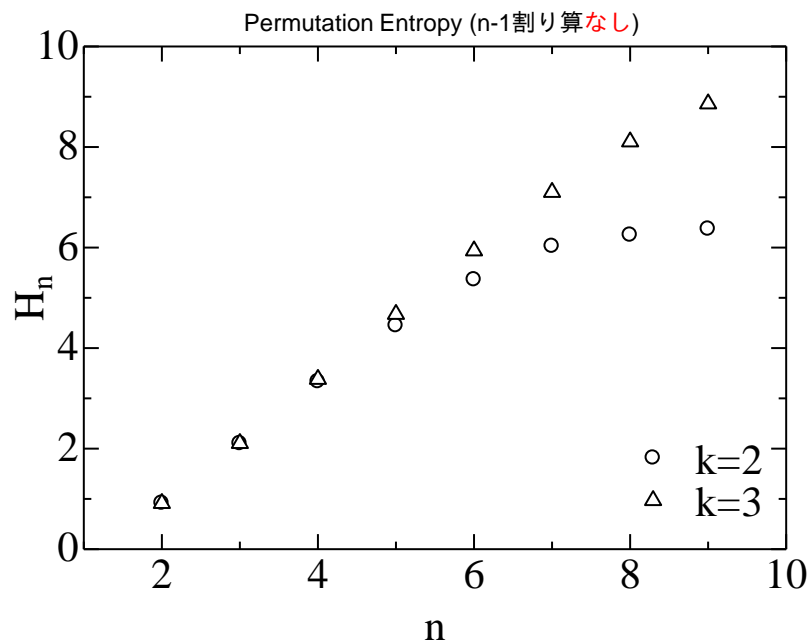
使用データ: whitenoise.txt; 0.49.txt; 0.56.txt; 0.62.txt (p-kPa/2012/U=3.5/1)
計算条件: N=300000, D=50, Tau=1; 2; 8; 8

[1] Christoph Bandt, Permutation Entropy: A Natural Complexity Measure for Time Series, PhysRevLett.88.174102

[2] Yinhe Cao, Detecting dynamical changes in time series using the permutation entropy, PHYSICAL REVIEW E 70, 046217 (2004)

A 順列エントロピー (Permutation Entropy) inspection

プログラム検証結果



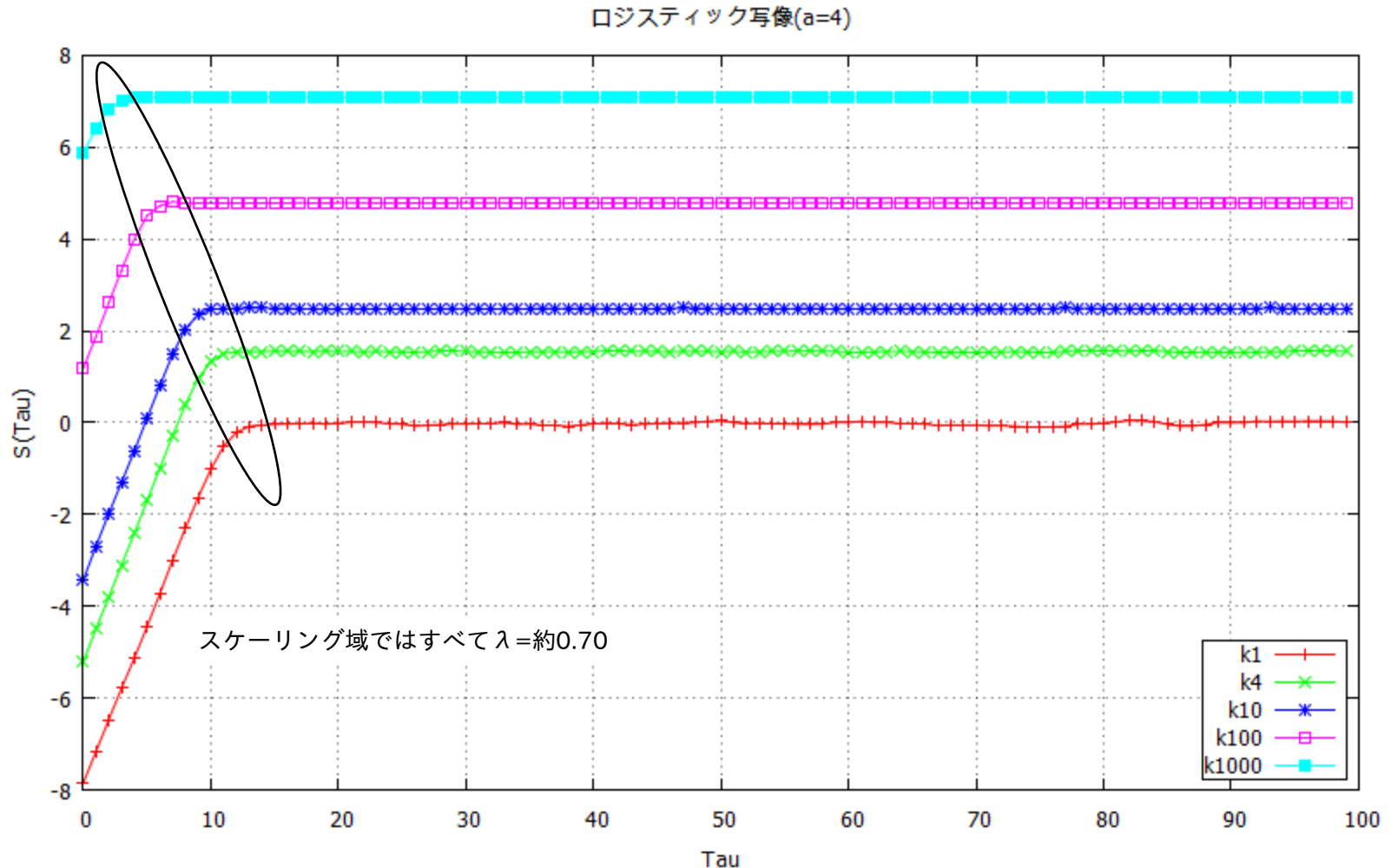
結果

Permutation Entropy Symbol(n-1割り算あり)が文献[1]FIG.3.と一致

結論

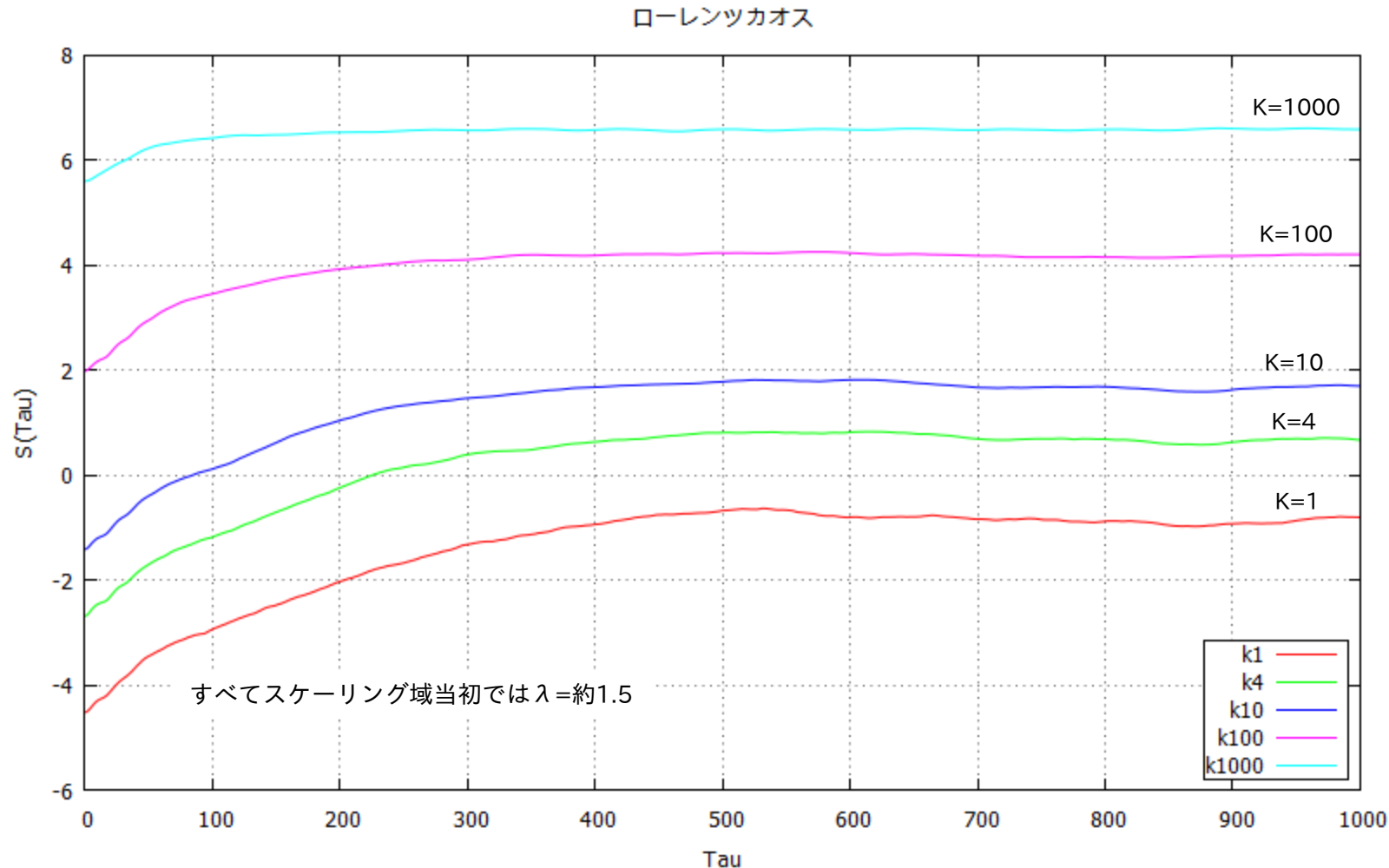
プログラムは正常に計算できている。
※実際の計算ではn-1で割ることはない。

4 近傍点数Kの影響について(テストデータによる検証)

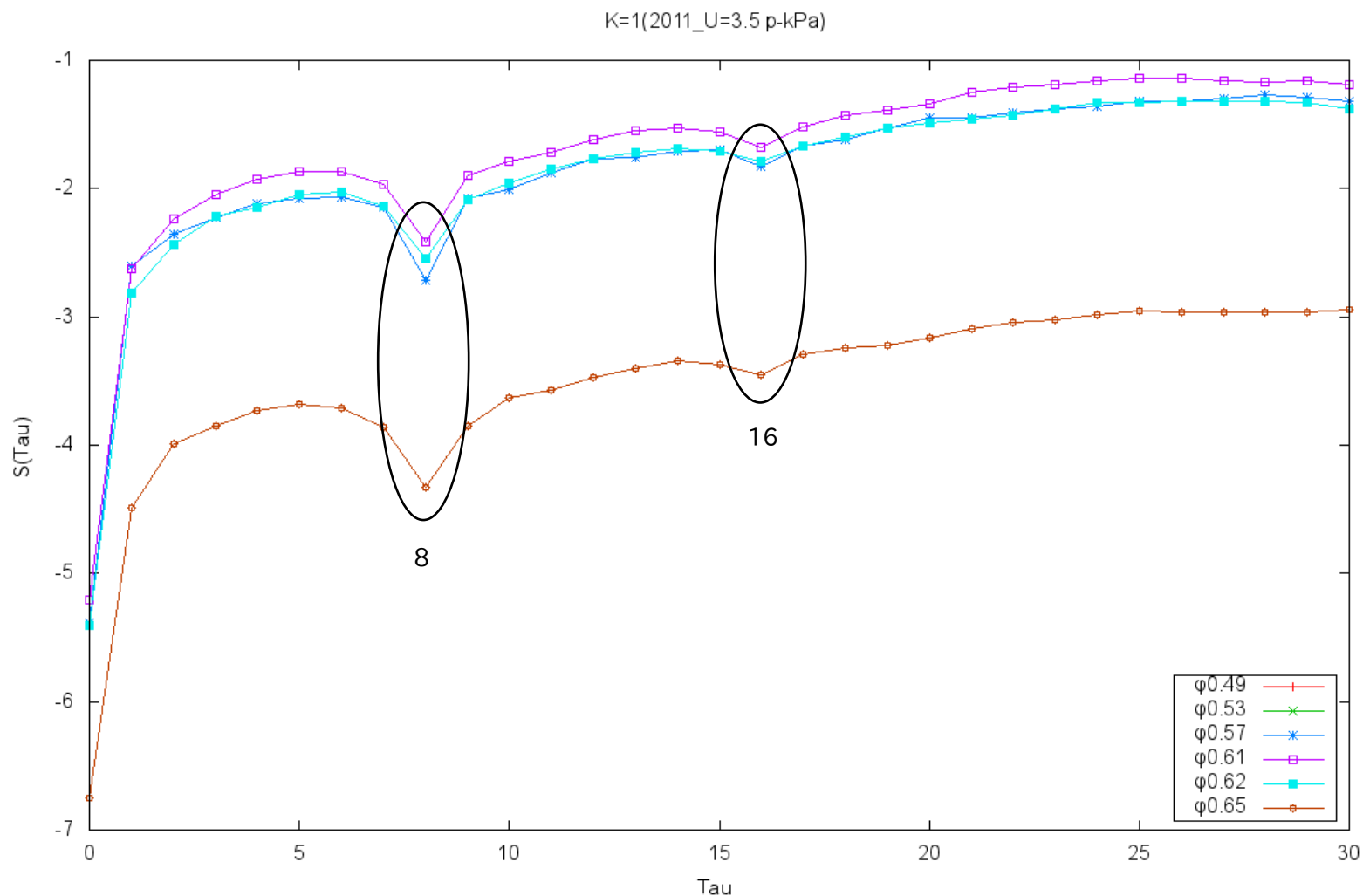


- ・ロジスティック写像では、結果に一種の線形的な関係性が見られる。
- ・近傍点数を増加させるとスケーリング域が減少するが、 $S(\text{Tau})$ の傾きは変わらない。
- ・ロジスティック写像($a=4$)では、近傍点数は $K=1$ が妥当(短期予測可能範囲も増えると考えられる)

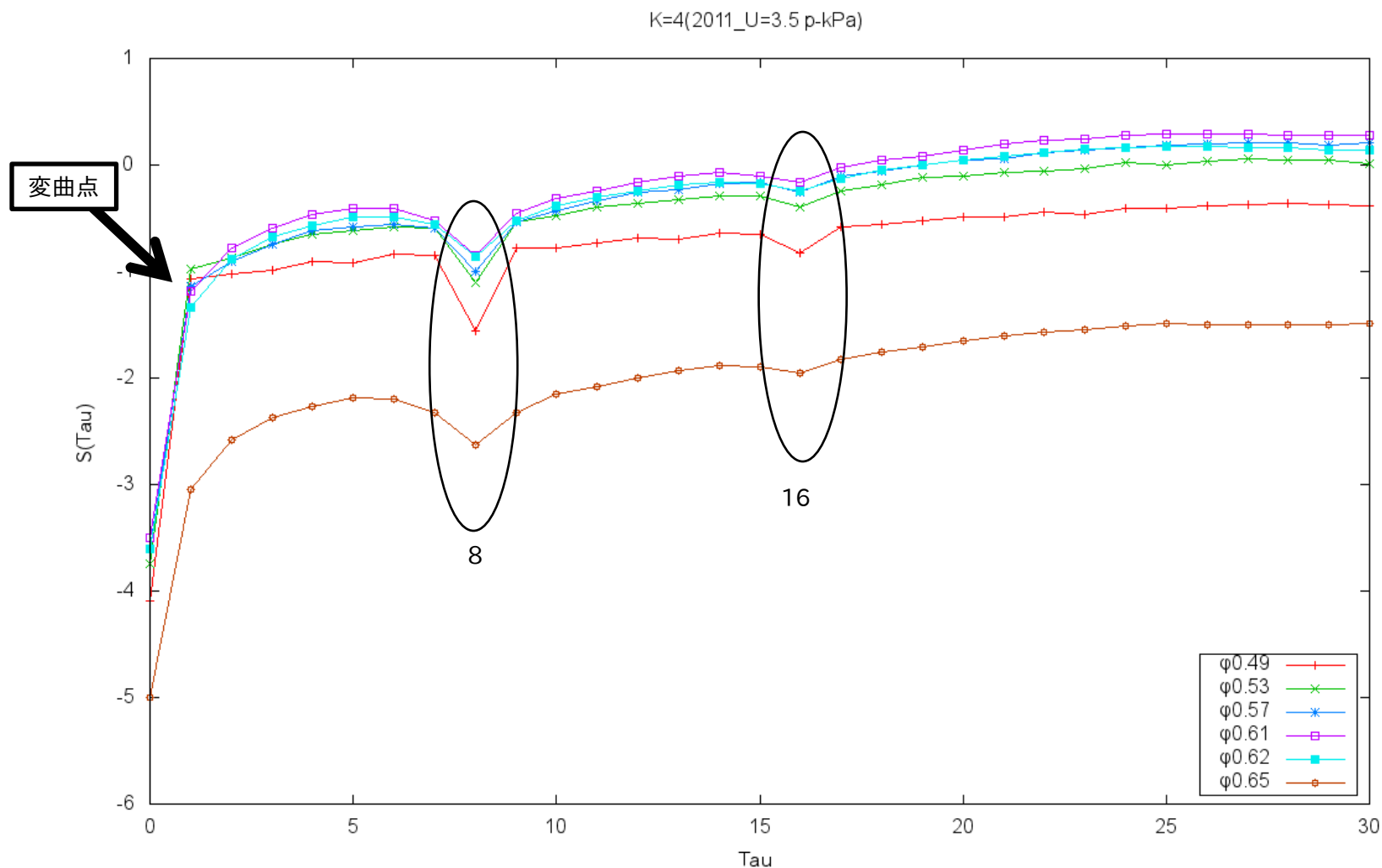
4 近傍点数Kの影響について(テストデータによる検証)



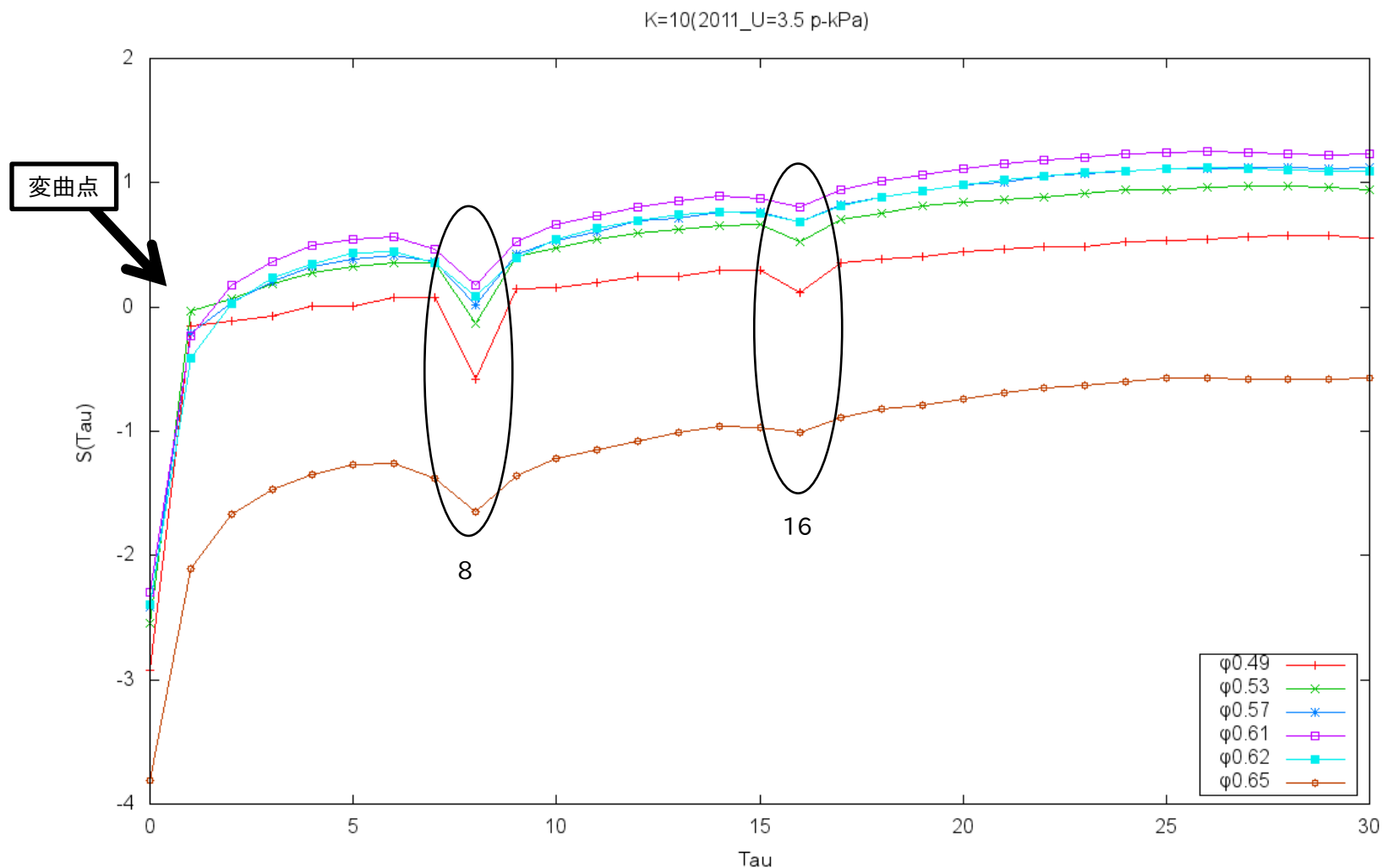
- ・ 近傍点数を増加させるとスケーリング域は減少し、 $S(\text{Tau})$ の傾きは近傍点数によって減少傾向が変わる。
- ・ 今回は $M=1000$ で計算を行った。より M を大きくすれば減少傾向が同一化する可能性がある。

4 流量 $U=3.5\text{m/sec}$ 圧力変動データ(2011年度)の最大リアプノフ指数

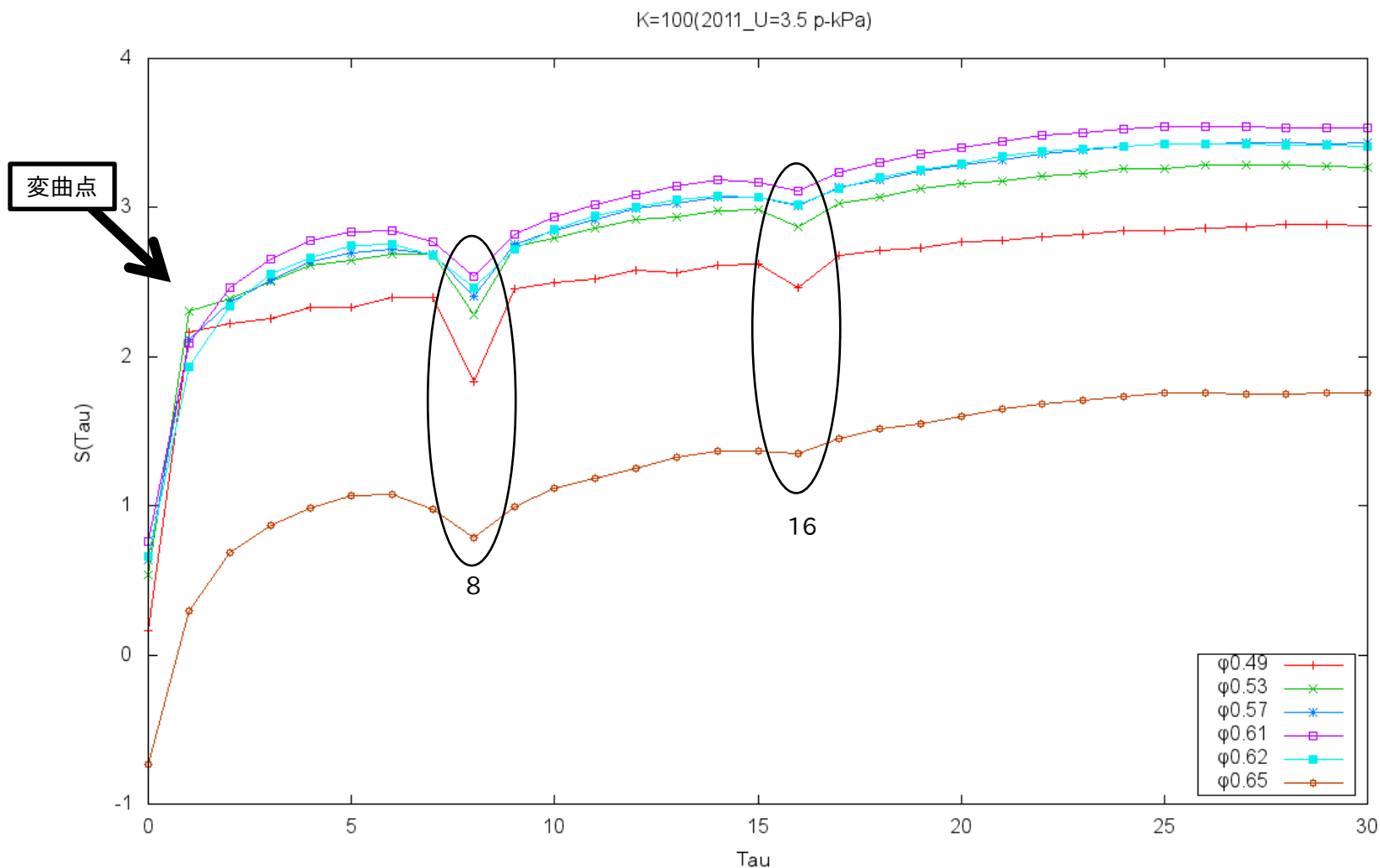
- ・ 当量比0.49, 0.53の全点は負の無限大となった。
- ・ 決定論的カオスでないことを示しているか、浮動小数点計算による桁落ち・情報落ちの可能性がある。

4 流量 $U=3.5\text{m/sec}$ 圧力変動データ(2011年度)の最大リアプノフ指数

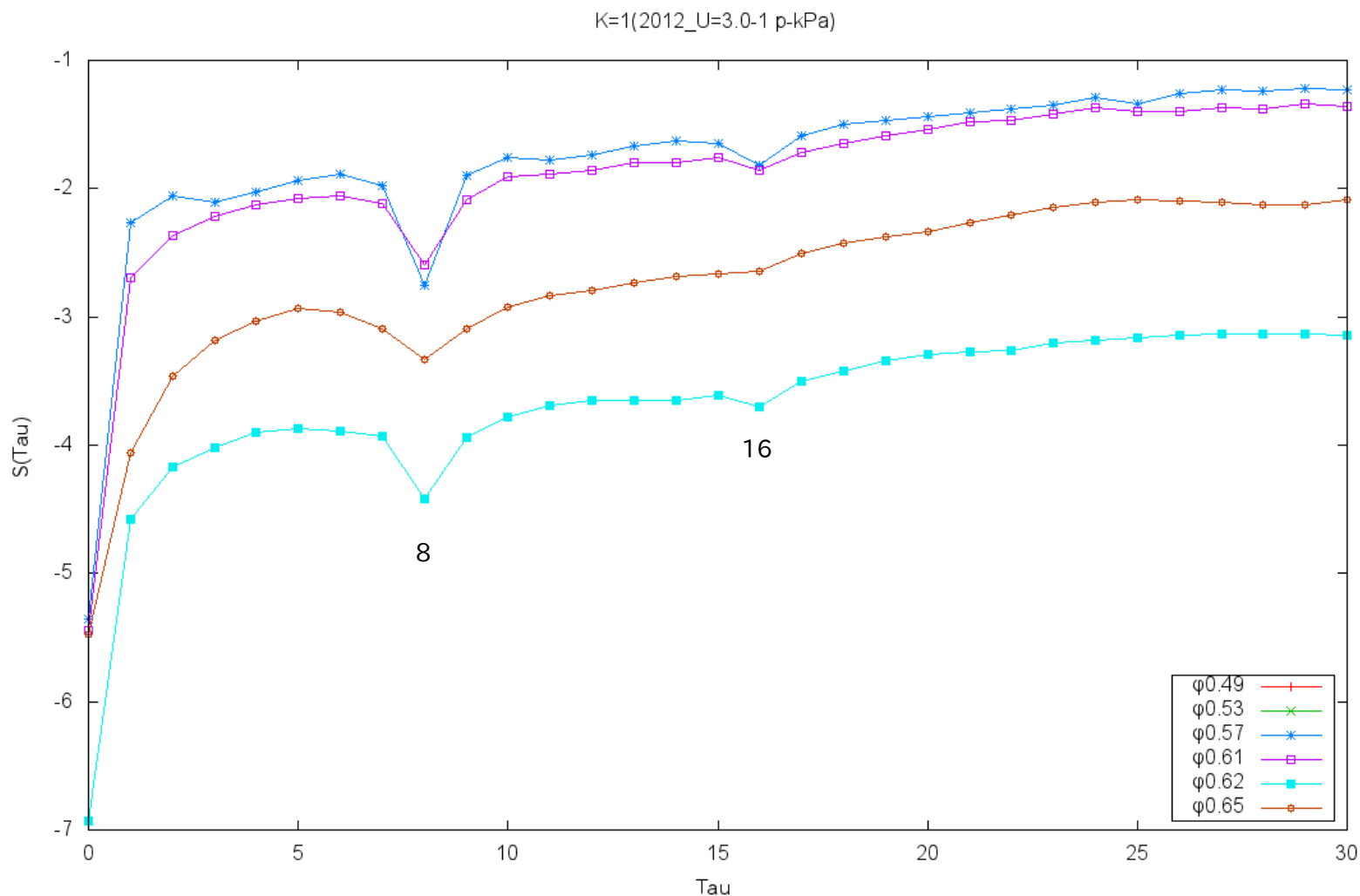
- ・ 全当量比で遅れ時間と同じ経過時間8で特徴的な挙動が見られる(2倍の16も同様)
- ・ 当量比0.49(吹き消え付近)でも最大リアプノフ指数は正を示したが、変曲点が他と比べ角張っている。

4 流量 $U=3.5\text{m/sec}$ 圧力変動データ(2011年度)の最大リアプノフ指数

- ・ K=4と比べ、 $S(\text{Tau})$ の値は変わっているものの、全体としての形は相似的である。
- ・ 最大リアプノフ指数はK=4とほぼ等しいといえる。

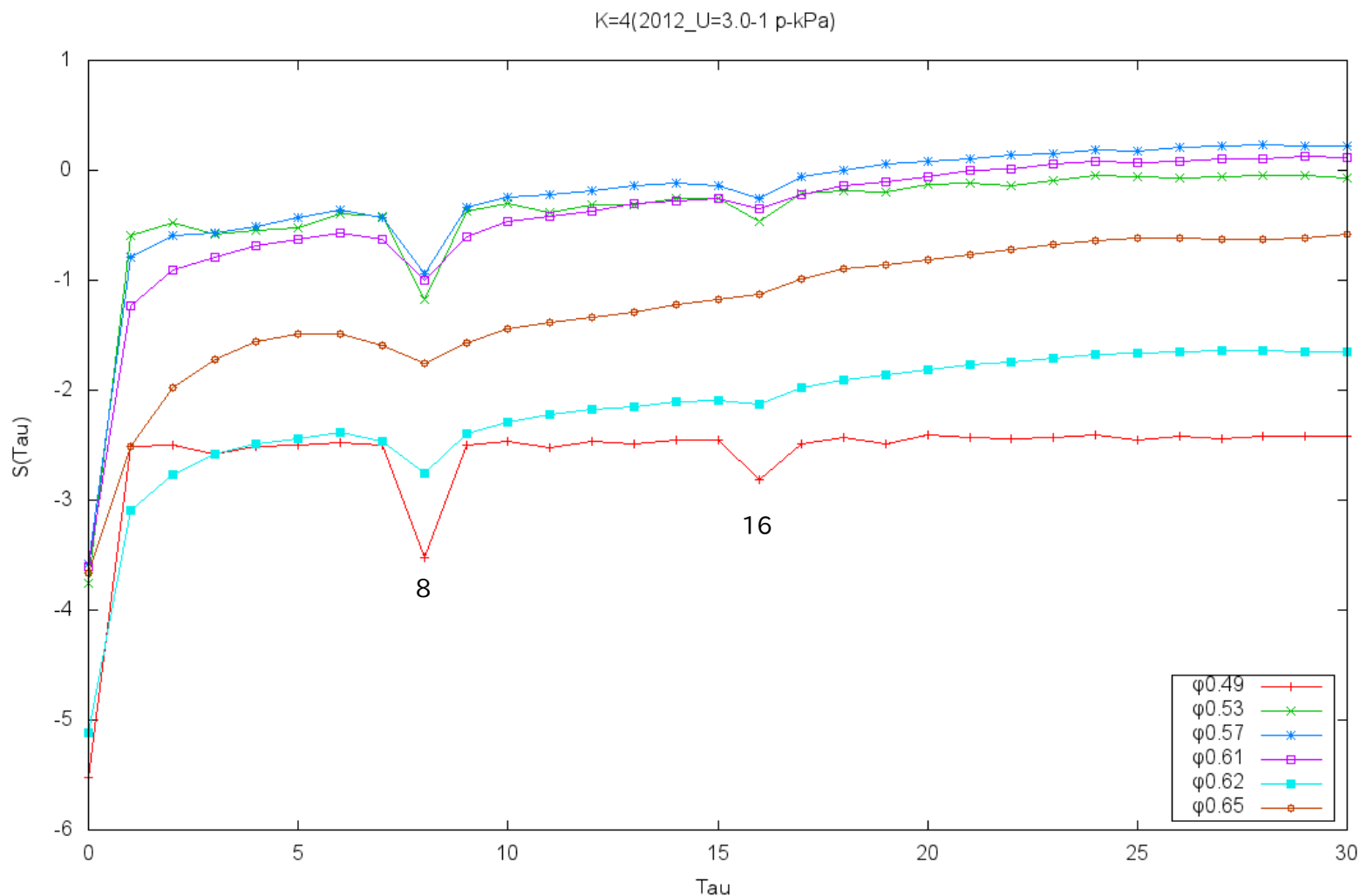
4 流量 $U=3.5\text{m/sec}$ 圧力変動データ(2011年度)の最大リアプノフ指数

- ・ $K=4, 10$ と比べ、 $S(\text{Tau})$ の値は変わっているものの、全体としての形は相似的である。
- ・ 0からの立ち上がりの伸びが K が増すごとに減少している。(=近傍点数が増えれば予測可能範囲が減ると予想できる)

4 流量 $U=3.0\text{m/sec}$ 圧力変動データ(2012年度第1回)の最大リアプノフ指数

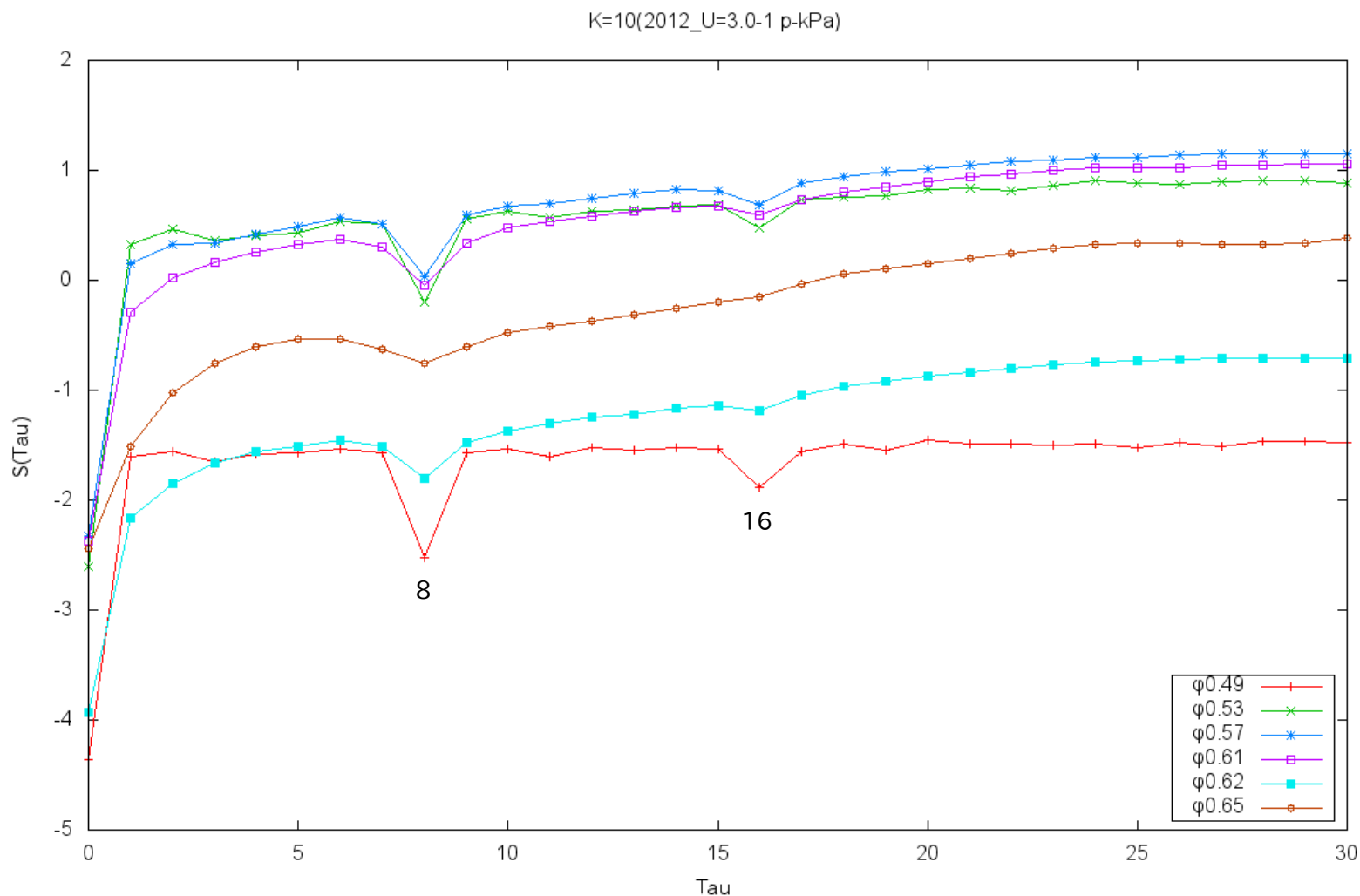
当量比0.49, 0.53, の全点は負の無限大となった。(プロットできず)

※ y軸目盛りのスケールが各近傍点ごとに異なるので注意

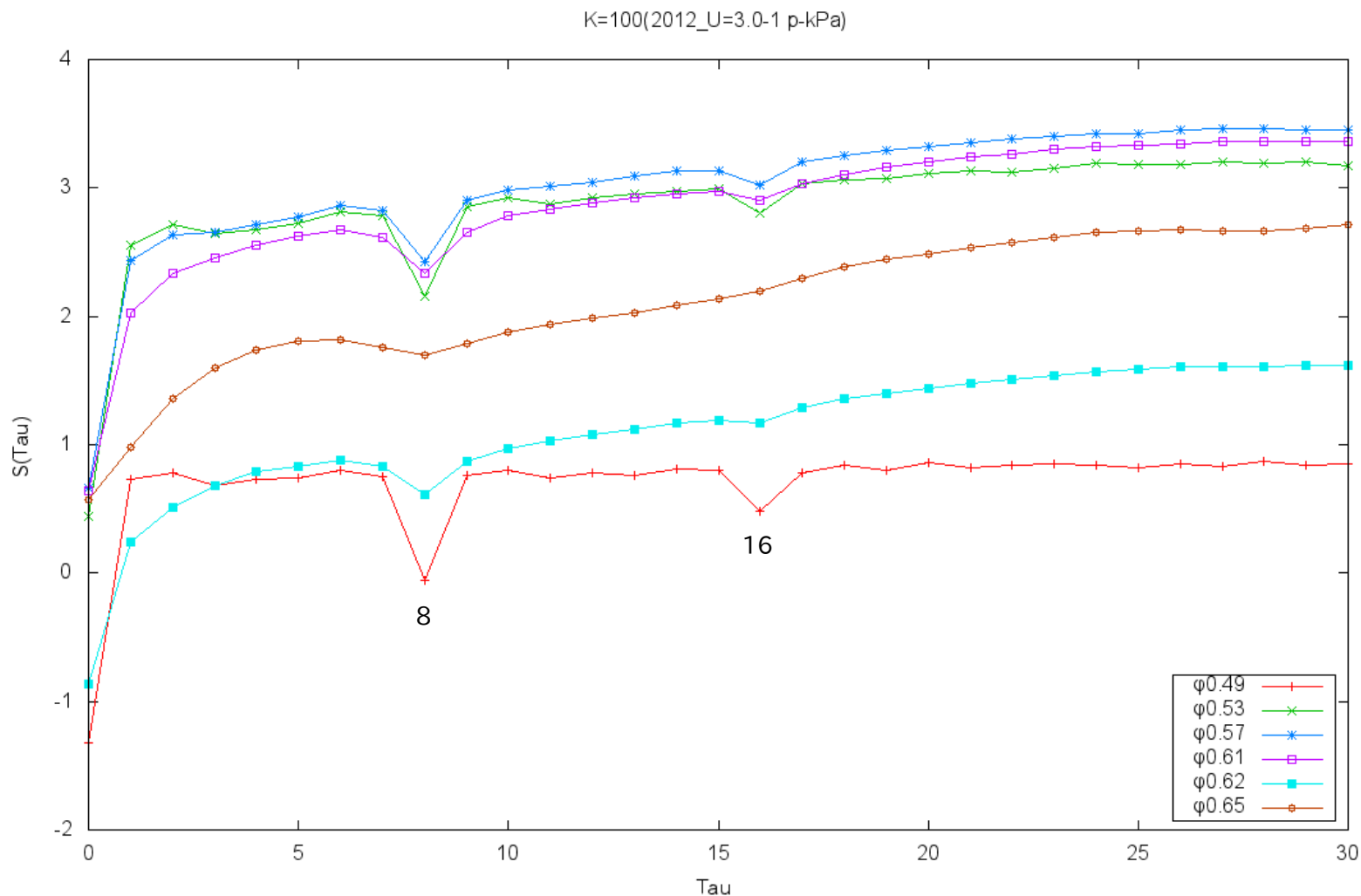
4 流量 $U=3.0\text{m/sec}$ 圧力変動データ(2012年度第1回)の最大リアプノフ指数

$K=4$ のとき, λ_{MAX} の傾きが他の近傍点数に比べて大きい($K=1$ を除く)

※ y軸目盛りのスケールが各近傍点ごとに異なるので注意

4 流量 $U=3.0\text{m/sec}$ 圧力変動データ(2012年度第1回)の最大リアプノフ指数

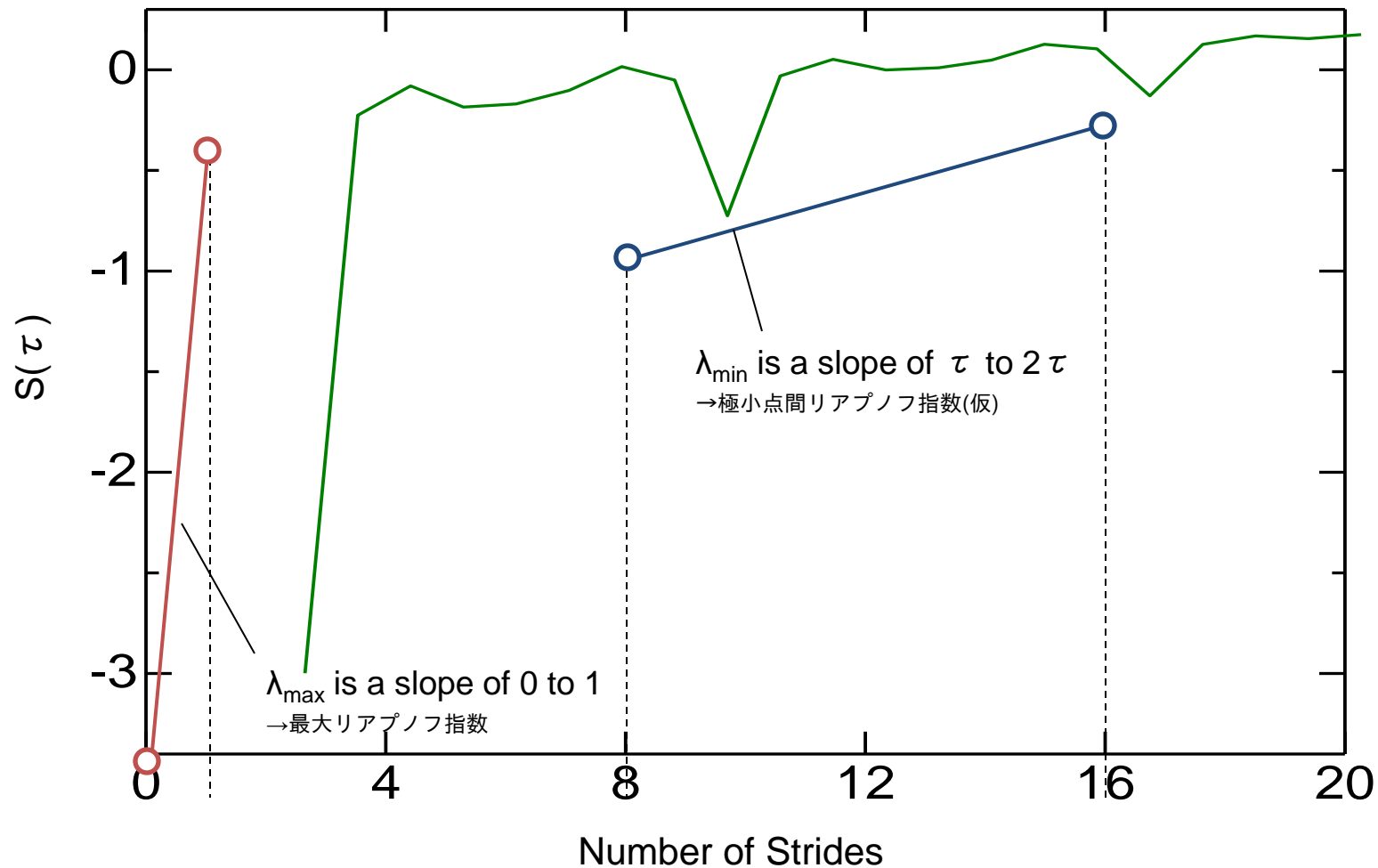
※ y軸目盛りのスケールが各近傍点ごとに異なるので注意

4 流量 $U=3.0\text{m/sec}$ 圧力変動データ(2012年度第1回)の最大リアプノフ指数

※ y軸目盛りのスケール・値が各近傍点ごとに異なるので注意

4 各当量比における圧力変動データのリアプノフ指数 λ_{\max} , λ_{\min}

本研究におけるリアプノフ指数の定義(案)

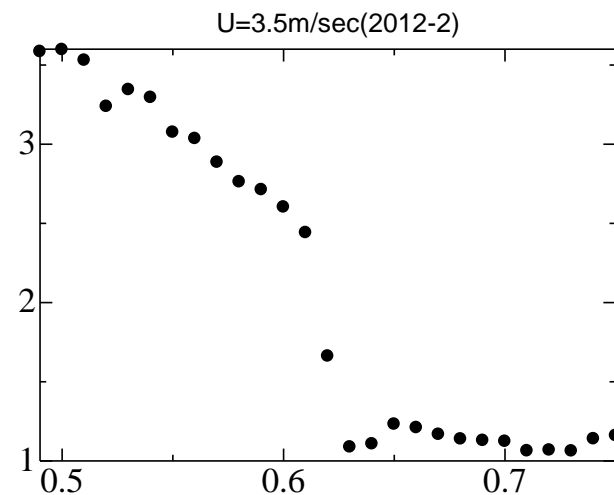
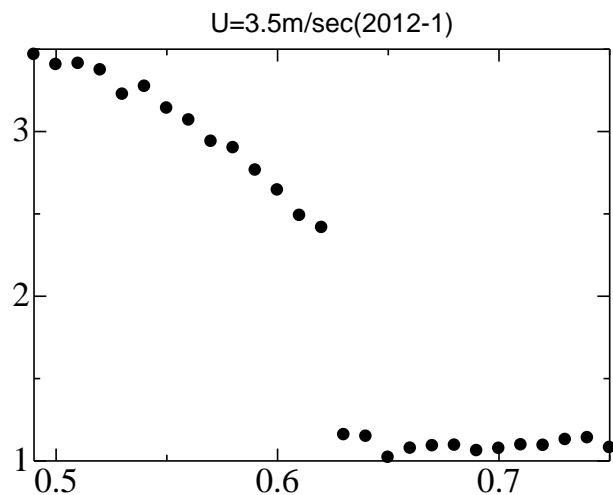
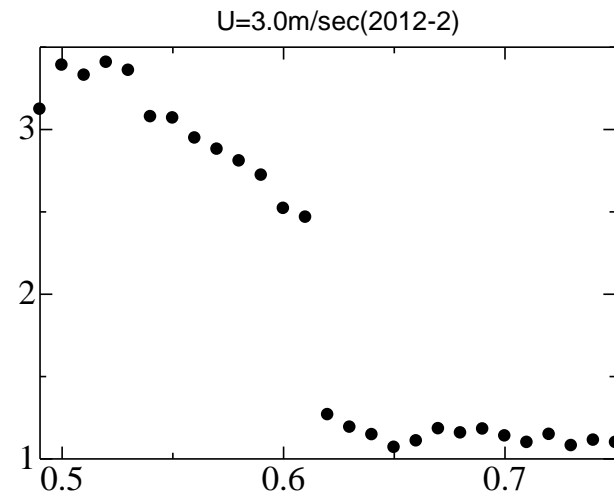
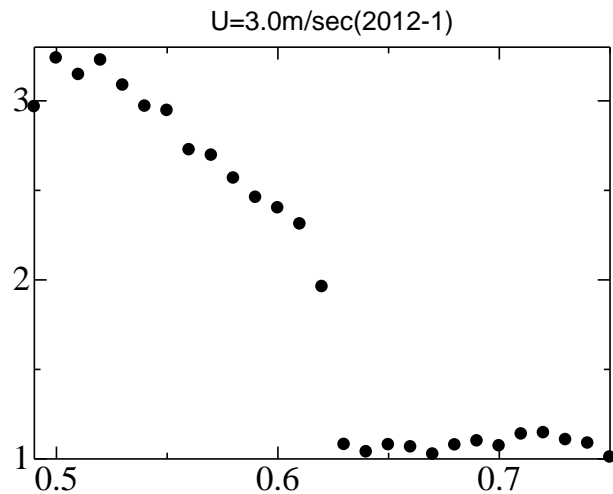


使用データ: 2012/U3.5/1/p-kPa0.55.txt

計算条件: $N=300000$, $D=3$, $\text{Tau}=8$, $M=100000$, $K=5$, $\lambda_{\max}=T_1-T_0$, $\lambda_{\min}=T_{16}-T_8$

4 各当量比における圧力変動データの最大リアプノフ指数 λ_{\max}

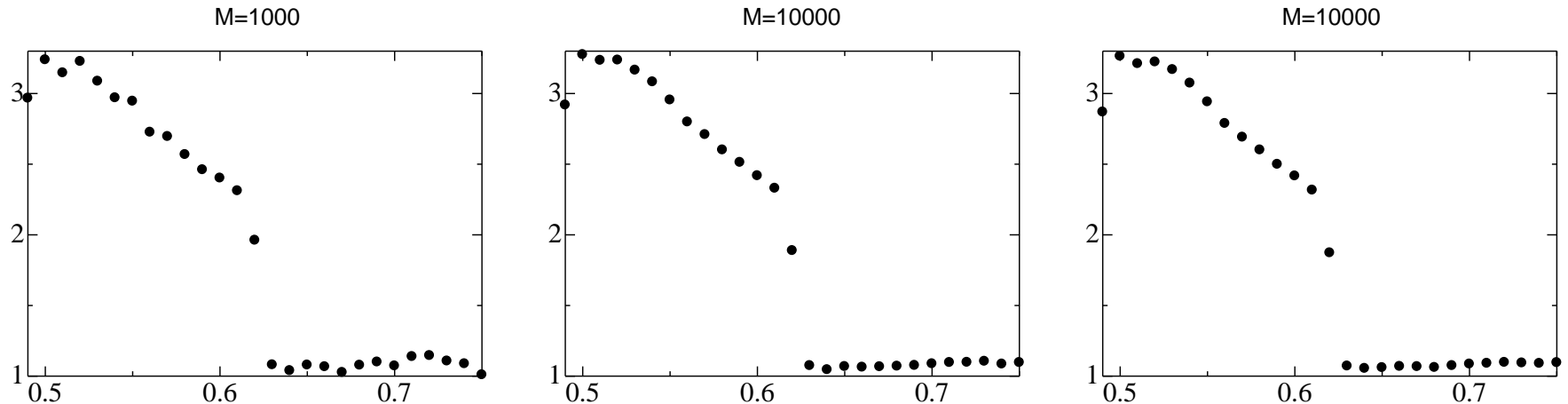
既定関数の個数 $M=1000$ ※ 横軸: 当量比 ϕ , 縦軸: λ_{\max}



4 各当量比における圧力変動データの最大リアプノフ指数 λ_{\max}

既定関数の個数(M)の影響

(例) $U=3.0\text{m/sec}(2012-1)$



※ 横軸: 当量比 ϕ , 縦軸: λ_{\max}

結果

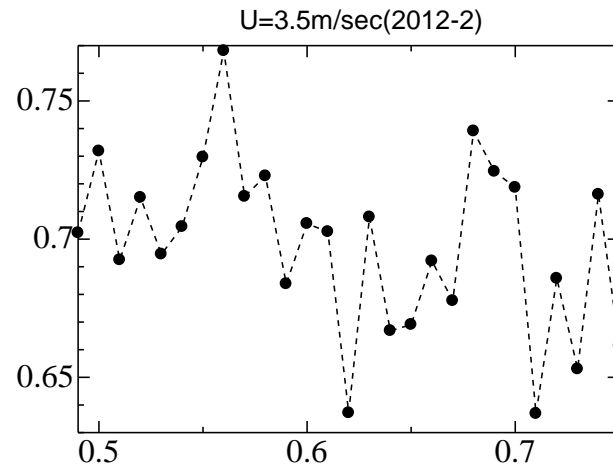
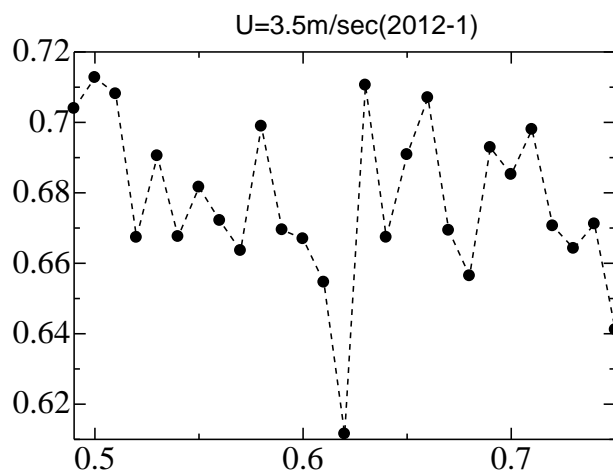
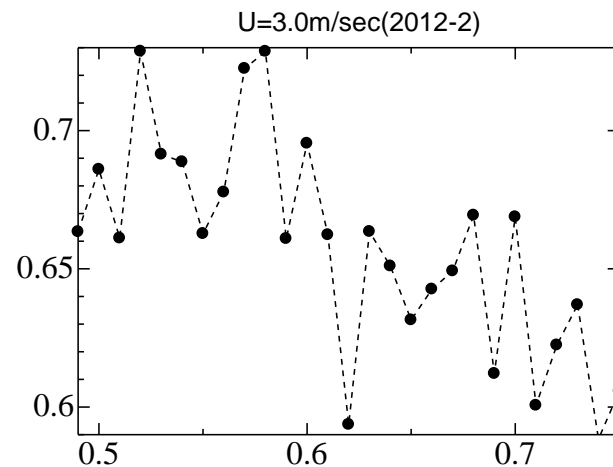
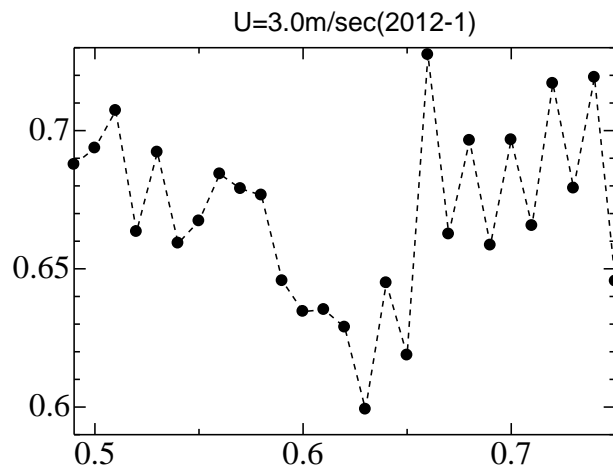
最大リアプノフ指数での推定で吹き消えの検知は有効
 $M=1000$ の結果は収束し切れていない。

提案

既定関数の個数Mの値を1000から変更する必要あり
 → 極小点間リアプノフ指数スライドにて提案

4 各当量比における圧力変動データの極小点間リアプノフ指数 λ_{\min}

既定関数の個数 $M=1000$ ※ 横軸: 当量比 ϕ , 縦軸: λ_{\min}

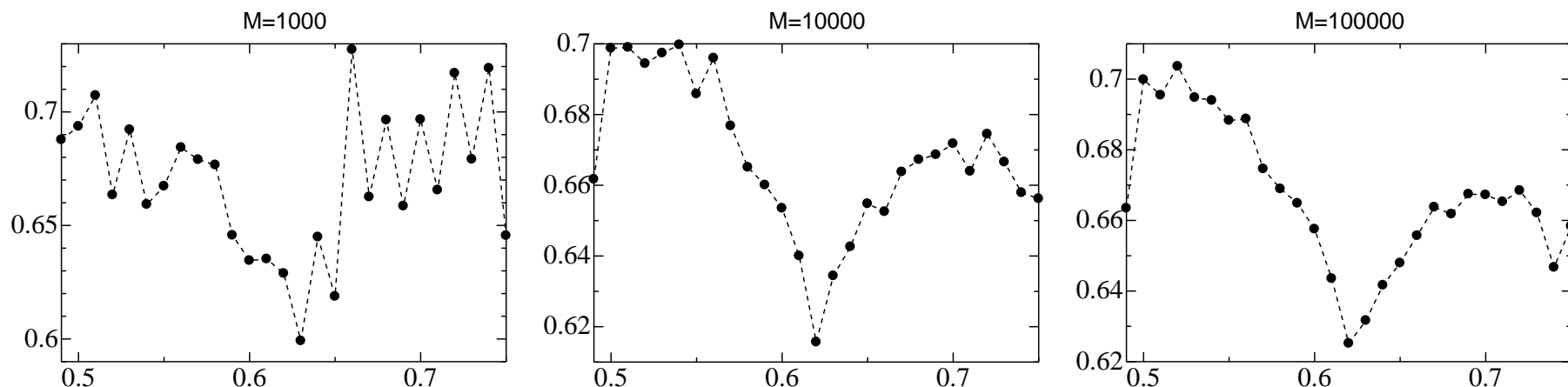


$M=1000$ のとき, 極小点間リアプノフ指数 λ_{\min} で特徴は見られない

4 各当量比における圧力変動データの極小点間リアプノフ指数 λ_{\min}

既定関数の個数(M)の影響

(例) $U=3.0\text{m/sec}(2012-1)$



※ 横軸: 当量比 ϕ , 縦軸: λ_{\min}

結果

$M=10000$ は吹き消え付近～間欠性までに負の傾きが見られる

$M=100000$ も同様の傾向がある

間欠性カオスである当量比0.62が最小である

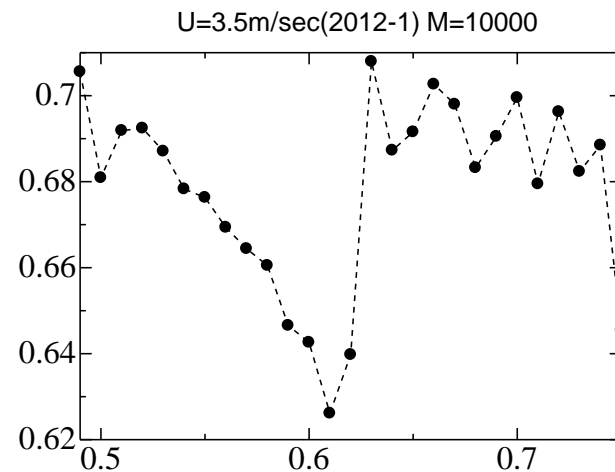
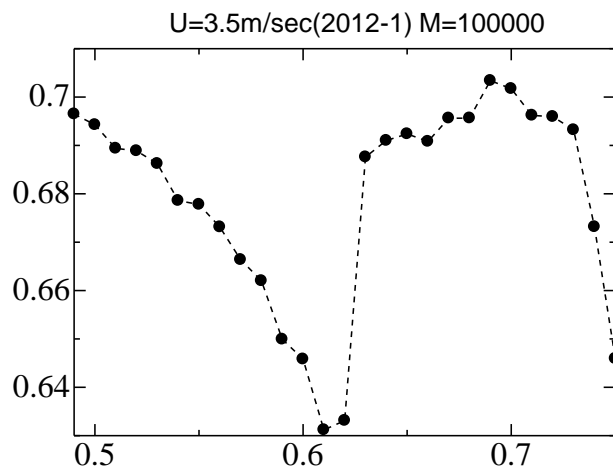
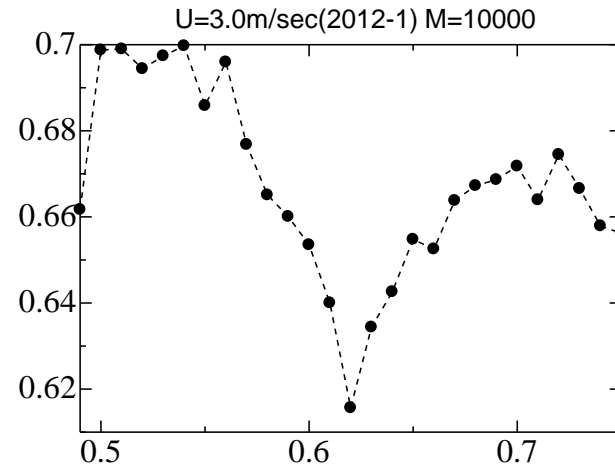
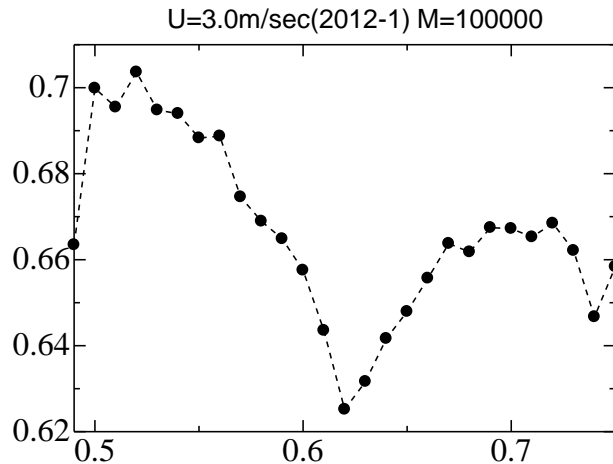
提案

既定関数の個数 $M=100000$ にする

※ $M=10000$ のとき, 収束していない結果があり得る

4 各当量比における圧力変動データの極小点間リアプノフ指数 λ_{\min}

既定関数の個数 $M=100000, 10000$ ※ 横軸: 当量比 ϕ , 縦軸: λ_{\min}



$M=100000$ のとき, 極小点間リアプノフ指数 λ_{\min} で特徴があらわれる