1/f Fluctuations
in
Conservative Systems and Computer Networks

by

深町 賢一

東京工業大学理工学研究科物理学専攻 (1996)

1/f Fluctuations

in

Conservative Systems and Computer Networks

by

深町 賢一

博士 (理学)

Abstract

$1/f$ゆらぎはデバイスをはじめ、DNA や心拍等の生命活動、銀河磁場やブラックホール等の巨大なスケールの現象、そして交通システムやコンピュータネットワークといった社会現象の分野まで、非常に広い範囲に渡って観測される現象である。$1/f$のある系では分散は発散し熱平衡状態への緩和が達成されない。つまり統計力学とは相容れない現象である。

$1/f$ゆらぎは一次元の差分方程式で物理量の時間変化が間欠的に急激な変化をおこすようなモデルで観測されることが知られている (Intermittency Chaos)。しかし差分方程式では現実とかけはなれているため本質的な部分を見えにくくしていると考えられる。そこで、簡単な偏微分方程式で $1/f$ゆらぎを考えてみたい。ここでは非線形項を持つ最も簡単な偏微分方程式である FPU(Fermi, Pasta, Ulam) 方程式を用いる。この式は水晶でのレーザー散乱実験で 「フォノン数が $1/f$ゆらぎを示す」という結果を説明するためのモデルである。

一方社会現象における $1/f$ゆらぎも興味深い。コンピュータネットワーク上を流れる情報量もまた $1/f$的振舞を示すことを我々は発見した。このネットワークにおけるゆらぎについて考えてみることで、広範囲の分野に渡る $1/f$性の共通点について理解が深まると期待される。

本論文では広範囲の分野に渡る $1/f$ゆらぎという現象に統一的な説明を与えることができるのか、ということを考えてみたい。そこで特に前述の FPU 方程式とコンピュータネットワークの二つの分野に絞り $1/f$ゆらぎについて考えてみる。

最初の例として、FPU 方程式を用いたフォノン数のゆらぎについての計算機実験を考える。エネルギー保存系で非常に有効な方法である Symplectic Integrator (一種の Trotter 分解) を用いることで非常に精度の良い計算をすることができた。フォノン数のゆらぎに注目しているので、物理量としてフォノン数に対応する線形モードのエネルギーに注目し、その時系列を FFT にかけパワースペクトル密度 (PSD) を求めた。モードエネルギーの時系列は静かな振動状態と非常に間欠的な挙動からなるのが特徴である。非線形項と線形項をほぼ同じオーダーにし、システムサイズが最大 32 個の系まで計算を行なったが、サイズが 8 個の系でも $1/f$ゆらぎを観測することができた。PSD は低周波数側で $1/f$、高周

波側で水平、つまりホワイトノイズ (ランダムな運動) という構造を示す。これは実験・観測等で見られる典型的な形である。$1/f$ 領域とホワイトノイズ領域の Crossover は非線形パラメータ (λ) とシステムサイズ (N) に依存する。我々の測定パラメータ域で Crossover $\sim \sqrt{\lambda}/N$ と決定した。全体としての定性的な変化はなく、$1/f$ ゆらぎは必ずしも大自由度を必要としないことが結論される。

$1/f$ を示す時系列はある状態への滞在時間分布を用いても特徴づけることができる。今の場合モードのエネルギーが間欠的にバーストしている。これをバーストしている状態とそうでないほぼ振動的に振舞っている静かな状態の 2 つに分け、それぞれの状態にどのように滞在するか (ある状態への滞在時間)、の分布として FPU 系の状態変化を記述することができる。我々はバーストとバーストの間の滞在時間 (T) の分布として T^{-2} を見い出した。系の状態変化と滞在時間 (T) 分布をセミ・マルコフ過程を用いて記述することで $1/f$ と分布 T^{-2} が結び付くことを示すことができる。このことは、位相空間内での運動と $1/f$ ゆらぎとの間に密接な関係があることを示している。

次に、位相空間内での運動をより詳細に調べるために位相空間内でのトラジェクトリの動きを追跡した。保存系では、トーラスとカオス領域の 2 つが系を特徴づけている。システムの不安定性を表すリアプノフ指数を用い、リアプノフ指数の収束速度を基準にして、位相空間の各場所がトーラスに属すかカオスに属すかを決定することができる。この方法を用い $1/f$ を示す軌道が、階層構造を持つ位相空間の中を渡り歩いている様子を見ることができた。つまり保存系における $1/f$ ゆらぎの基本構造は間欠性カオスと階層性 (フラクタル的構造) からなると考えられる。

二番目の例として、コンピュータネットワークの $1/f$ ゆらぎを考える。具体的にはメーリングリストのトラフィックの時系列の $1/f$ ゆらぎを測定したものである。メーリングリストとは電子版の文通のようなものであり、トラフィックとは単位時間に行きかう手紙の数である。トラフィックはある程度の定常的な量があり、時々バーストを起こす。つまりこの系も FPU 方程式同様間欠性を示すのである。PSD は一週間以下では人間の周期的活動を反映したピーク構造を持ち、一カ月を越える時間スケールでは $1/f$ ゆらぎを示した。

$1/f$ ゆらぎとは明らかに長時間の相関である。つまり長い時間系の初期情報を忘れないと言いかえても良い。しかし系の発展プロセスが初期情報から未来を完全に決定できる場合 $1/f$ ゆらぎは生じえない。そこで、ここでのネットワークのモデル化は「長い時間系の初期情報を記憶してはいるが、完全に記憶しているのではなく少しずつ系の情報を忘れる。」というメカニズムを明示的にモデルに組み込むことで $1/f$ ゆらぎを再現するモデルを作った。そして、PSD の $1/f$ 領域の大きさは「過去の情報をどれくらいずつ忘れていくか」という度合と相関していることを示すことができた。われわれはこの「長い時間系の初期情報を記憶してはいるが、完全に記憶しているのではなく少しずつ系の情報を忘れる。」という捉え方が $1/f$ ゆらぎを研究する上での分野を越えた共通の指針 (パラダイム) として重要だと考えている。

WHITE PAGE

Contents

I　保存系の $1/f$ ゆらぎ **9**

1　はじめに **11**

1.1　ゆらぎ 11

1.2　フラクタルとカオス 12

1.3　散逸系 :ローレンツ方程式の示すカオスの本質 14

1.4　標準写像と保存系 18

1.5　保存系の位相空間の階層構造 21

2　FPU 方程式 **25**

2.1　はじめに 25

2.2　FPU 方程式 26

2.3　パラメータ依存性 30

3　リアプノフスペクトル **33**

3.1　Cross Section 33

3.2　リアプノフスペクトル 36

3.3　ある構造へ捕獲されること 37

3.4　拡散 40

II　コンピュータネットワークの $1/f$ ゆらぎ **42**

4　コンピュータネットワークの $1/f$ **43**

5 メーリングリスト **45**

5.1 メールシステム 45
5.2 メーリングリストとそのモデル 46
5.3 観測例 49
5.4 メーリングリストのモデル化についての予備考察 49
5.5 メーリングリストのモデル 51

6 ネットニュースシステム **59**

6.1 ネットニュースシステム 59
6.2 ネットニュースシステムのモデル化 61
6.3 ネットニュースモデルのデータ構造 62
6.4 ネットニュースシステムの構造 62
6.5 ネットニュースの時系列の観測例 65
6.6 システムの形成と相関 65
6.7 拡大する系 69

III 結論と今後の課題 **74**

7 結論と今後の課題 **75**

A ネットワークの下位層におけるフラクタル **79**

A.1 はじめに 79
A.2 イーサネット LAN 80
A.3 FDDI とイーサネット LAN 83
A.4 まとめ 86

論文リスト

Ken'ichi Fukamachi,
"1/f Fluctuations in Internet Communication"
Interdisciplinary Information Sciences, vol.1, 157 (1995).

Ken'ichi Fukamachi,
"On the Phase Space in FPU lattices",
in print in Physica B, The proceeding of PHONON INTERNATIONAL CONFERENCE.

Ken'ichi Fukamachi,
"1/f Fluctuations in FPU Lattices",
Europhys. Lett, 26 (6), 449-454 (1994).

Ken'ichi Fukamachi and Hidetoshi Nishimori,
"Specific heat of the quantum Heisenberg antiferromagnet
on the kagome lattice",
Phys. Rev. B {\bf 49}, 651 (1994).

Part I

保存系の$1/f$ゆらぎ

,

Chapter 1

はじめに

1.1 ゆらぎ

$1/f$ゆらぎというものが１９２５年に発見されて以来７０年の月日が経っているが、今日でもゆらぎというものは重要ではないので無視してしまえばいいものという傾向はいまだあるように思われる。ゆらぎについて次のように書いた人もいる。

> It is probably fair comment to say that to many physicists the subject of fluctuations (or "noise" to put it bluntly) appears rather esoteric and perhaps even pointless; spontaneous fluctuations seem nothing but an unwanted evil which only an unwise experimenter would encounter!
>
> MacDonald, D. K. C., *Noise and Fluctuations* 1962 (Wiley, New York).

$1/f$スペクトルを再現する単純な発想は「$1/f$はローレンツスペクトル（$1/f^2$型）がたくさん重なり合った事象である」というものである。この考え方を式で表現するなら

$$1/f \sim \sum f_c/(f_c^2 + f^2) \tag{1.1}$$

となる。ここで f_c は f^{-2} と f^0 の境界を決める周波数（Corner Frequency）にあたる。この形の場合 $1/f$問題はいかにローレンツスペクトルを足し合わせるかという命題に還元される。この考え方は、$1/f$に対する有力な考え方として今日でも知られている。しか

しこの考え方ではデバイスだけでなく社会現象等まで含む非常に多くの分野の $1/f$ 現象を統一的に説明できないとおもわれる。

1.2 フラクタルとカオス

一般に f^{μ} 依存性 (μ は実数) を示すものをフラクタルと呼んでいる。μ は系の特徴的なスケールの次元である。たとえばある長さ r の直線にとって系の大きさとは r であるし、一辺 r の正方形の系の大きさは面積で r^2。同様に立方体は r^3 と考えるのが自然である。つまり r^{μ} の μ が次元と等しい。言いかえると、r という代表的な観測スケールを変えていく時のものの見え方の依存性が系の 「フラクタル次元」であるということになる。

図 1-1: 海岸線とその測定の例。海岸線の長さは長さ r の折れ線で近似して求める。

一般に自然界で見られる形を扱うとなるとそう単純ではないが、適当なスケールの間ではフラクタルの考え方でものごとを見ることが出来る。例えば、海岸線 (図. 1-1) の長さは観測のスケールを変えるとどんどん長くなる。これはものさしの長さ r を小さく

すると、細かい凹凸まで計るようになるので長くなるのである。この時、$r^{\mu}(\mu$の値はいろいろだが 1 と 2 の間$)$ という依存性を示すので海岸線はフラクタルである。

図 1-2: コッホ曲線。 線分の真ん中の 3 分の 1 を三角にする」という操作をひとつの線分から始めて、各直線部分について無限に繰り返すと得られる。

この海岸線のような図形を、単純なルールでまねることができる。例えば、コッホ曲線 (図. 1-2) は海岸線の荒い近似だと考えることができる。コッホ曲線の作り方は単純で、 線分の真ん中の 3 分の 1 を三角にする」という操作をひとつの線分から始めて、各直線部分について無限に繰り返すだけである。カントール集合 (図. 1-3) も同様に、 線分の真ん中の 3 分の 1 を切り取る」という操作を、ひとつの線分に対し無限に繰り返したものである。フラクタルは、あるスケールの自然の現象を見るときの物の見方に過ぎないが、単純なルールが非常に複雑な構造を作ることが出来ることとは驚きであろう。

単純なルールで複雑に見える構造を作る」という点ではカオスも同様である。この両者の共通性でわかるとおり、カオスとフラクタルは密接に結び付いている。例えば、ロ

図 1-3: カントール集合。フラクタル次元は 0.63 である。

ジェスティックマップ (差分方程式)

$$x_{n+1} = \alpha x_n(1 - x_n)(0 < \alpha < 4) \tag{1.2}$$

ここで、 x_n はステップ n での値である。この式は、都市の人口の増え方を記述や生物の集団の成長を記述する時によく使われる。これは一次元の単純な式だが、周期運動からカオスまで、多様な振舞を見せるのである。

1.3 散逸系 :ローレンツ方程式の示すカオスの本質

カオスの本質は引き延ばしと畳み込みである。E. N. Lorentz は大気循環の現象に興味を持つ研究者だったが、今日ローレンツ方程式と呼ばれる簡単な 3 次元方程式を用いカオスを発見した[1]。これは、偏微分方程式 (PDE) のような現実的な系がカオス的振舞をみ

[1]H. Poincare が最初に発見したので、厳密には 再発見」である

せることを示した最初の例としても知られている。このことは次の意味で重要である。すなわち、すでに知られてはいたが数学的な玩具の域を出ていなかった差分方程式でのカオスがPDEがカオスを示したことにより、単なる数学的な例外のようなものではないかもしれないと考えられるきっかけとなったのである。ローレンツ方程式は簡単でありながら、カオスの基本的性質、引き延ばしと畳み込みをその中に見ることが出来る。

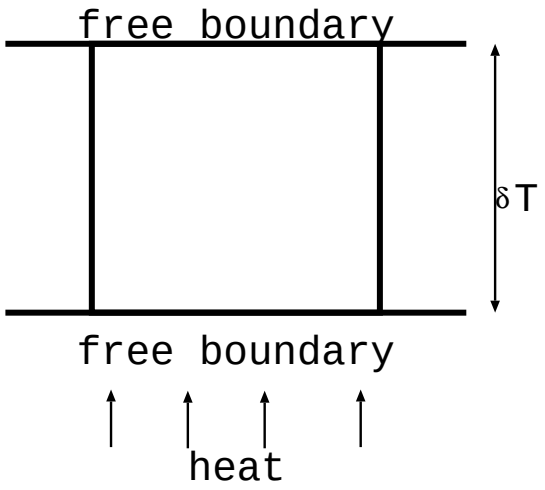

図 1-4: ローレンツ方程式は2次元の大気循環のモデルである。図の上と下は自由境界である。熱は下から加えられ、温度は図の垂直方向に沿って上がっている。

本来大気の運動は3次元だが、ローレンツ方程式は簡単のため2次元内に限定して簡単化している (図. 1-4)。運動は次のような3変数の方程式で記述される

$$\dot{X} = -\sigma(X - Y) \tag{1.3}$$

$$\dot{Y} = RX - Y - XZ \tag{1.4}$$

$$\dot{Z} = XY - bZ \tag{1.5}$$

ここで R は定数、σ, b も正の定数、X は対流運動を表し、Y は温度分布、そして Z は垂直方向の分布に関係した量である。ローレンツは、次のような条件$\sigma = 10, b = 8/3, R = 28$

のもとで初めてカオスを得ることができた。上の方程式から

$$\frac{\partial \dot{X}}{\partial X} + \frac{\partial \dot{Y}}{\partial Y} + \frac{\partial \dot{Z}}{\partial Z} = -(\sigma + b + 1) < 0 \tag{1.6}$$

が得られ、散逸的であることがわかる。つまり、位相空間の体積 (ルベーグ測度) は縮んでいく。しかし、このことは位相空間に何もなくなってしまうことを必ずしも意味しない。たとえルベーグ測度が 0 になっても、カントール集合のような奇妙な集合は存在しても構わない。実際長時間の運動の後、ローレンツ系のトラジェクトリが最終的に描く軌道は、ローレンツアトラクタ (図. 1-5) とよばれている奇妙な集合である[2]。

ローレンツ ([Lorentz 63]) は、トラジェクトリの引き延ばしと畳み込みがカオスを生み出すことを明らかにした。畳み込みは重要である。例えば、初期値のわずかの差が指数関数的に大きくなるという指数関数的不安定性は指数関数 (例 :$\dot{x} = x$) ですら持っている。しかし、非線形性から来る畳み込みが指数関数的に拡大している系を有限な領域に閉じ込めて、カオスを生成しているのである。ローレンツアトラクタ (図. 1-5) ではトラジェクトリは 2 枚の二次元平面 (実際には非常に薄い 3 次元的なもの) の中に閉じ込められ、その中でトラジェクトリは指数関数的不安定性を持つ。トラジェクトリが二つの平面の間で移り変わる時畳み込みが起こる。これが複雑な様相を生成しているのである。

ローレンツ方程式は大域的な対称性、つまり X, Y を同時に反転させても不変

$$X \rightarrow -X \tag{1.7}$$

$$Y \rightarrow -Y \tag{1.8}$$

という性質を持っているが、それはカオスとはあまり関係が無い。非線形項一つでもカオスになりうるのである。それを Rössler は次のような簡単な方程式により示した。

$$\dot{x} = -y - z \tag{1.9}$$

$$\dot{y} = x + ay \tag{1.10}$$

[2]例えば [Thompson and Stewart 86] を見よ

図 1-5: ローレンツアトラクタではトラジェクトリは 2 枚の二次元平面 (実際には非常に薄い 3 次元的なもの) の中に閉じ込められ、その中でトラジェクトリは指数関数的不安定性を持つ。トラジェクトリが二つの平面の間で移り変わる時畳み込みが起こる。これが複雑な様相を生成しているのである。

$$\dot{z} \quad = \quad b - z(x - c) \tag{1.11}$$

ここで a, b, c は定数である。 Rössler 方程式は物理学的考察とは関係なく導出された物であるので、ローレンツ系と違い x, y, z に物理的な意味づけは無い。この式は上述のローレンツ方程式のような対称性も持たず、 Rössler 方程式は ただひとつの非線形項 (zx) が 2 次元平面への引き戻しをすることで Rössler アトラクタ (図. 1-6) を生成している。

1.4 標準写像と保存系

系の全エネルギーが保存量になるため、保存系と呼ばれる系が存在する。多くの場合、系の運動量やその他の量も保存量になる。この系ではエネルギーが保存することから系の運動には制限が加わる。つまりトラジェクトリは有限領域に閉じ込められる。これは、保存系が指数関数的不安定性を持つなら、有限な領域の中で不安定であるからカオスを生み出し得るということである。保存量 I とは、元の系の変数を正準変換 [Goldstein] した後

$$\dot{I} \quad = \quad \frac{\partial S}{\partial \omega} = 0 \tag{1.12}$$

$$\dot{\omega} \quad = \quad \frac{\partial S}{\partial I} = \text{const.} \tag{1.13}$$

のように求められる保存量 I のことである。ここで、S は作用、$\omega = \int \partial S/\partial I = \text{const} \times t$ は I に付随した周波数という意味になる。つまり円でいえば、I が半径、 ω が角周波数ということである。例えば 1 次元の調和振動子 (ハミルトニアンが $H = p^2 + q^2$ なら作用 S は $S = H = I$) ならその形になる (楕円でも同様)。つまり保存系を考えるときはまず第一にトーラス (2 次元なら円、 3 次元ならドーナツ形) という構造を考えることが出発点である。また上述のようにカオスも容易に生成しうるのだから、トーラスとカオスの 2 つが保存系を考える上での手がかりとなる。

しかしながら、偏微分方程式の計算は大変なので次のように偏微分方程式から導出された差分方程式を代わりに用いることが多い。差分方程式はあるステップからあるステップへの変換だから、その分計算量が少ないため計算機実験が容易になるのである。簡単

図 1-6: レスラーアトラクタ。ここでは初期値をある点を中心にして小さい球の中に分布させてこれを時間発展させたもの。レスラーアトラクタは非常に薄い帯をメビウスの環のように半ひねりして張り合わせたような形をしている。球はすぐに 2 次元平面に押しつぶされ、その後一回転するごとに折り畳まれる。

な調和振動子の系 (2 次元) の正準変換後の式は次のような形で書かれる。

$$J = \mathbf{const.} \tag{1.14}$$

$$\theta = 2\pi\alpha t \tag{1.15}$$

ここで J,θ は正準変数である。また t は時間で α は回転数である。この式から導出された、マップと通常呼ばれる差分化された方程式は

$$J_{n+1} = J_n \tag{1.16}$$

$$\theta_{n+1} = \theta_n + 2\pi\alpha \tag{1.17}$$

ここで n はステップ数 (連続系では時間) である。

上の振動子を少し変形した次の式は、標準写像 [Chirikov 79] と呼ばれている。

$$J_{n+1} = J_n + Ksin\theta_n \tag{1.18}$$

$$\theta_{n+1} = \theta_n + J_{n+1} \tag{1.19}$$

ここで K は系のコントロールパラメータである。今の場合

$$(J_{n+1},\theta_{n+1}) \leftarrow (J_n,\theta_n) \tag{1.20}$$

の変換のヤコビアンが 1 であるから、(J_n,θ_n) というベクトルを対角線に持つ長方形の面積が保存しているとみなせる。多自由度の場合には、これらの面積を足し合わせたもの (厳密には two-form と呼ばれる) が保存されることになる。この系はこの幾何学的な保存量をもつという点で、保存系の性質を一部持っているといわれる。よって標準写像によって得られた結果も保存系の性質の一部を反映しているだろうと推定される。

1.5 保存系の位相空間の階層構造

保存系のトラジェクトリはどのように位相空間の中を動いているのか見てみよう。図. 1-7 は標準写像のトラジェクトリの動いたところがプロットされている。保存系なのでプロットは円状の部分と複雑に要り乱れたカオス領域から成る。$K = 0.9716$ より大きい K の値ではカオス領域は大域的につながっている。標準写像をより細かく見ると図. 1-8 のように島の周りに島があるというような階層構造が存在することがわかる [Meiss 92]。そのような階層構造は写像の結合系（標準写像と同じ意味で保存系）でも見られる。例えば「シンプレクティック写像」と呼ばれる写像 [Konishi and Kaneko 92, Konishi and Kaneko 94]

$$(x_i, p_i) \to (x'_i, p'_i) \quad i = 1, 2, ..., N \tag{1.21}$$

$$p'_i = p_i + \frac{K}{2\pi\sqrt{(N-1)}} \sum_{j=1}^{N} \sin 2\pi (x_j - x_i) \tag{1.22}$$

$$x'_i = x_i + p'_i \tag{1.23}$$

は「たまねぎ」様の階層構造を示す (図. 1-9)[Konishi and Kaneko 92]。この階層構造は保存系に特徴的だと推測されているが、一般性に関してはまだ明瞭ではない。

図 1-7: $K = 0.9716354$ の時の標準写像。

図 1-8: $K = 1.2014133$ の時の標準写像における階層構造。島のまわりに島があるように見える。

図 1-9: $N = 4, K = 0.3$ の場合の Symplectic maps。X 軸は momentum p_1 で Y 軸は p_2.
図中の ・はトーラスの領域であり、それ以外はカオスに属すると考えられる。

Chapter 2

FPU 方程式

2.1 はじめに

一次元マップで間欠性カオス [Pomeau and Manneville 80, Aizawa and Kohyama 84] を示すような系では、$1/f$ の PSD が観測されることが知られている。しかし、マップは簡単化されすぎた系であるため、得られた結論もどのくらい一般的なものであるのかはっきりしない。この観点から、より現実的な偏微分方程式で $1/f$ を研究する事が望ましいと考えられる。

統計力学は、調和振動子をベースとする近似を用いた場合、現実的な物質の比熱等についてかなり正確な記述や予言をする事ができる。しかし、その基本的な仮説「エルゴード仮説」の正当性は今でもはっきりしていない。電子計算機の利用が可能となった時、Fermi, Pasta そして Ulam は、計算機を用いてこの仮説の検証を試みた。具体的には、ひとつのモードにだけエネルギーを与えそのエネルギーの他のモードへのエネルギー輸送を調べた。実際の系は必ず多かれ少なかれ非線形性があるのだから非線形性が系をエルゴード的にすると彼らは期待していた。しかしモードエネルギーは分配されることなく、再び一つのモードに戻ってきてしまうという振舞が発見された。これは、系のエネルギーが小さくかつなめらかな初期条件の時には起こる。

ここでは、その Fermi, Pasta および Ulam (FPU) によって提唱された FPU 方程式と呼ばれる系について考えることにする。我々の FPU 方程式の非線形項は、４次の非線形

項だけである。この系はもっとも簡単な非線形系の一つであるから、出発点として理想的である。我々は、システムサイズが 8 の系でも $1/f$ を見出す事に成功した。得られた PSD は、十分低周波数側で $1/f$ 的振舞、高周波側ではホワイトノイズ ($\sim f^0$) のようになる。その 2 つの領域のクロスオーバー周波数は非線形パラメータ (λ) とシステムサイズ (N) に依存する。PSD 全体としての形はシステムサイズ (N) が $8 \leq N \leq 32$ の間で変化しないため、$1/f$ ゆらぎは現実的な系でも必ずしも大自由度性を必要としないことがわかる。

2.2 FPU 方程式

古典的な原子が一次元上に並んだリング (図 . 2-1) の方程式の一般形は、

$$\partial^2 u_i/\partial t^2 = \phi'(u_{i+1} - u_i) - \phi'(u_i - u_{i-1}) \tag{2.1}$$

である。ここで、ϕ' はポテンシャル ϕ の微分である。u_iの動きが十分小さいなら、ポテンシャル関数$\phi(x)$ は$\phi(x) \propto x^2/2 + \lambda x^4/4$ のように近似してよい。ここで、λ は非線形パラメータである。適当な正規化の後、次の形の式 (FPU 方程式) を得る。

$$\partial^2 u_i/\partial t^2 = u_{i+1} - 2u_i + u_{i-1} + \lambda\{(u_{i+1} - u_i)^Q - (u_i - u_{i-1})^Q\} \tag{2.2}$$

ここで Q は今の場合 3 であり、i は 1 から N まで動く。振幅 u_i の初期条件は-1 と 1 の間でランダムに決定し、初期速度はすべてのサイトで 0 にした。また周期境界条件を適用する。マップのハミルトニアンは、δ 関数を含むなどの異常性を持っているが、FPU 系は普通にハミルトニアンをもつという点だけでもマップより現実的である。[Aizawa and Kohyama 84, So and Mori 84]。

まず初めに、保存系の数値解法について考える。保存系では多くの保存量があるからそれぞれを保存しつつ、計算を進めることが重要である。Runge-Kutta のような計算法でもよいが、保存系専用に限るならより高精度の計算が出来る方法が考案されている。Symplectic Integrator (SI) として知られるこの方法は、保存系に対しては非常に有効で

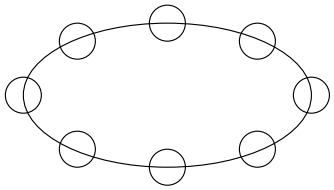

図 2-1: 我々のモデルは古典的な原子が一次元上に並んだリングである。

ある。基本的な概念は「保存系の時刻 t のハミルトニアン $H(t)$ は

$$H(t) = exp((A+B)t)H(t=0) \tag{2.3}$$

のように演算子 A, Bを用いて書けるから、演算子 A と Bに対してフラクタル分解を適用する」というものである。実際どの程度の精度がでるかは、ステップ (e.g., 0.01, 0.02, 0.04..) を変えながら計算してみて評価することができる。我々の場合、u_i の時系列について我々の計算は振幅について 10^{-9} 程度の精度で計算していると評価できた。またこの時、エネルギー保存則は 10^{-12} 程度の精度で成立している。

フォノン数のゆらぎを次のように定義する。モード a_n は u_i を $u_i = \sum_n a_n exp(2\pi in\sqrt{-1}/N)$ のように展開した係数である。$|a_n|^2$ が励起したフォノンの数ということになる。$\lambda = 0$ の時は、格子振動の n 番目のモードは正確に a_n である。我々は n を固定して $|a_n|^2$ の時系列を調べた。$|a_n|^2$ の時系列は規則的に振動しているが、時々バーストする (図. 2-2)。このような振舞は、間欠性カオス (Intermittency Chaos) として知られている。

PSD は時系列から 2^{13}個の点を取り、Fast Fourier Transform [Aho et. al. 74] を用いた。個々の PSD の初期条件への依存性を落す為に、PSD は時系列の分散が同じになるように正規化した。本来$\int S(f)df$ は、$|a_n(t)|^2$ の分散に等しいので、我々の PSD の正規化は、$S(f)$ を $S(f)/\int S(f)df$のように正規化したものである。異なる初期条件に対し $S(f)$ を求め、正規化した後、サンプル平均を取った。

十分低周波数での PSD は、$1/f$依存性を示す (図． 2-3)。我々の計算範囲では、十分低周波数でもホワイトノイズへ収束しなかった。これは現実の現象と一致している。図． 2-3 は $1/f$からホワイトノイズへ移り変わる様子を示している。ここでは図． 2-3 の $1/f$

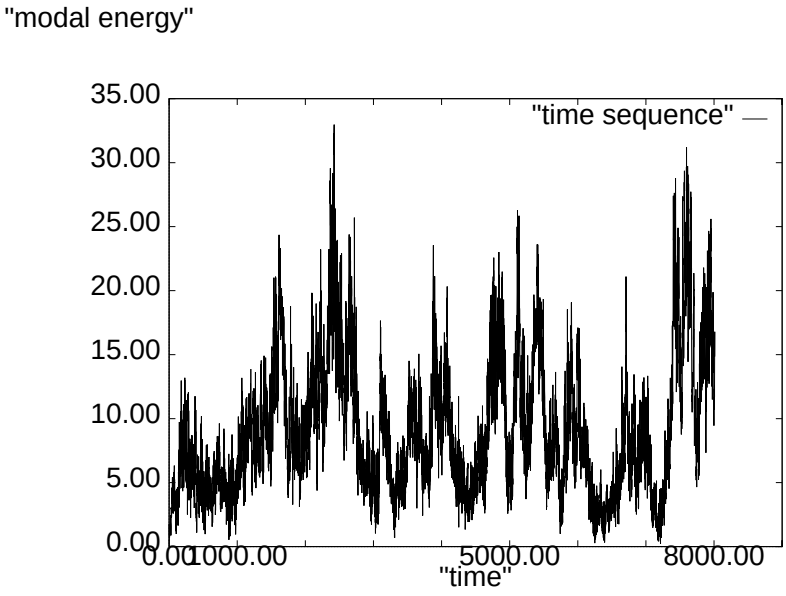

図 2-2: モードエネルギー (モード１) の時系列。水平軸が時間、垂直軸はモードエネルギーである。時間は全部で８０００秒分である。$1/f$ゆらぎのスケールはおおよそ１００秒より大きいスケールなのでこの図のスケールのバーストの間隔の分布は $1/f$を特徴づけている。

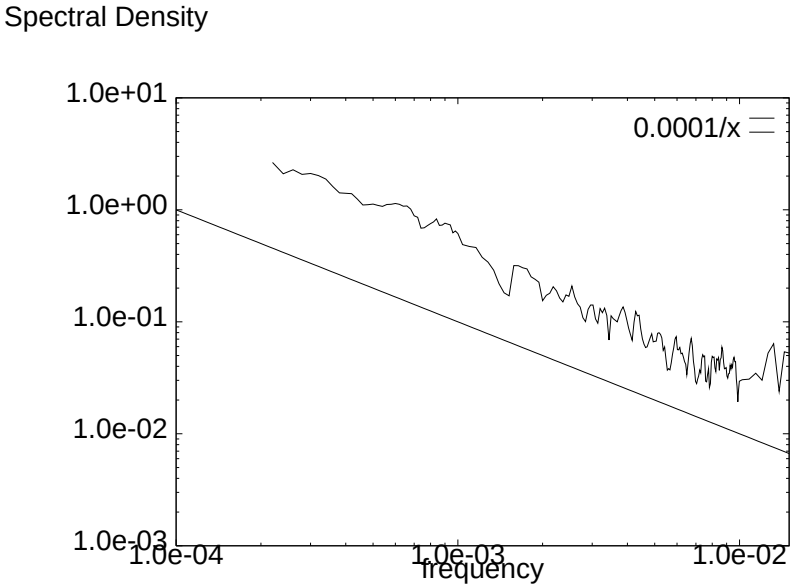

図 2-3: $N = 8, \lambda = 0.8$ に対する典型的な power spectrum density (PSD)。PSD は低周波で $1/f$ 形をとり、周波数があがるとホワイトノイズに近付いていく。

と f^0 PSD の間のクロスオーバーが $5 * 10^{-3}$ あたりで見られる。この形の PSD は現実によく観測される。

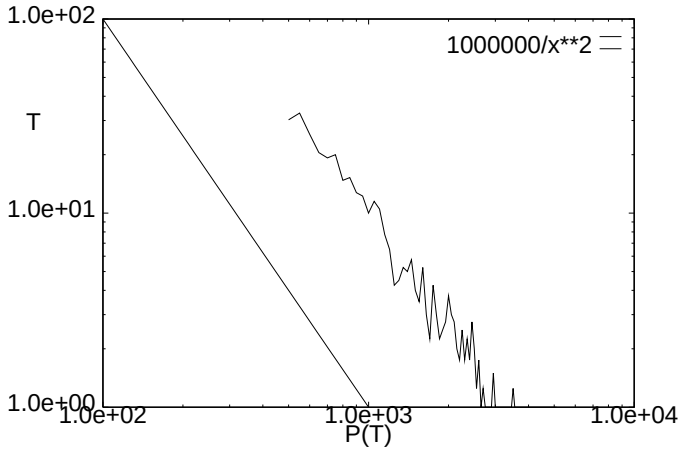

図 2-4: 時刻 T の分布関数 $P(T)$ は $|a_n|^2$ の時系列の 2 つのバーストの間の期間の分布として定義される。この図は $N = 8, \lambda = 0.8$ に対してのもので、図は１００個以上の異なる初期条件の PSD サンプルについて平均を取ったものである。ここでは $|a_n|^2$ が $|a_n|^2$ の時系列全体の最大値の $90\ \%$ を越えている部分をバーストと定義している。PSD の $1/f$ 領域 は $f \sim 10^{-2}$ あたりではじまる。この図では $T \geq 100$ に相当する。分布関数は T^{-2} 依存性を 100 sec. を越えた辺りから示し始める。

PSD の時系列の振舞はバースト状態とそうでない状態に分けられるが、バーストでない状態への滞在時間の分布関数を用いても特徴づけることが出来る (図 . 2-4)。ある物理量の時系列でバーストの間のほとんど周期的な振る舞いをする期間を T として分布関数 $P(T)$ を定義する。ここでは $|a_n|^2$ の時系列全体の最大値の 90% を越えている部分をバーストと定義している。このバーストの期間は大抵非常に瞬間的である。いいかえれば簡単のため時系列を 2 つの状態に分けシンボル化することで解析を容易にしている。つまり'1' をバースト状態'0' をそうでない状態として'0100000010000...' のような時系列を考えるのである。我々の場合物理量は $|a_n|^2$ もしくは 固有モードエネルギーである。これらの $P(T)$ を調べ、それぞれについて同じ $P(T) \sim T^{-2}$ 依存性を見い出した。$1/f$が観測される時 $P(T)$ が $P(T) \sim T^{-2}$ のように振る舞うことは低自由度系のカオス理論でも知られていた [Aizawa et. al. 89]。我々の結果は低自由度系のカオス理論と一致している。これは PSD の $1/f$的振舞を別の形で特徴づけている。

2.3 パラメータ依存性

クロスオーバー周波数の非線形パラメタおよびシステムサイズ依存性を調べる。クロスオーバー周波数は次のように定義する。PSD ($S(f)$) は $S(f) \sim a/f + b$ (a と b は定数) のように振る舞うから、外挿して a/f と b が $f_c = a/b$ でクロスオーバーするように f_cを合わせ、クロスオーバー周波数とした。$1/f$領域で$\partial ln S(f)/\partial ln f$ は -1 またホワイトノイズでは 0 となる。また、$f_c = a/b$ は$\partial ln S(f_c)/\partial ln f_c = -0.5$ for $f_c = a/b$, および $\partial ln(f_c S(f_c))/\partial ln f_c = 0.5$ も満たすから、 2 つの条件がうまく満たされる f_c がちょうどクロスオーバー周波数になる。f_c の自動処理による評価のため、上の二つの方法を併用した。

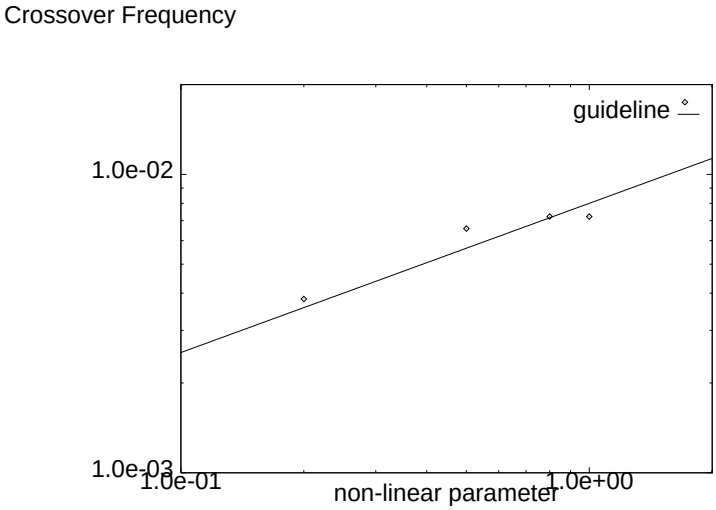

図 2-5: f_c の非線形パラメタ依存性 λ ($N = 12$)。ここでは固定された f_c についてサンプル平均を取ったものから計算された。我々の f_c のデータは様々な初期条件にわたって平均操作をしたものである。図の直線は $f_c \sim \sqrt{\lambda}$ である。この図の実験データは $\lambda = 0.2 \sim 1$ で取られた。ここでの結果は小さい非線形パラメータをもつであろう実際の物質では $1/f$ ゆらぎを観測するのは難しいであろうということを示している。

図 . 2-5は f_c がシステムサイズが固定されている時に非線形パラメータλとともにどう変わるかを示している。図 . 2-5 中の直線は見やすいようにガイドラインとして引かれた $f_c \sim \sqrt{\lambda}$ である[1]。この図は小さい非線形パラメータになると f_c が非常に小さくな

[1] クロスオーバー周波数のスケーリング則は自明ではないが、我々の結果が$\sim 1/\sqrt{\lambda}$というスケーリングの形にのるのはおもしろい。

るため $1/f$ ゆらぎを観測するのは非常に難しいであろうということを示している。

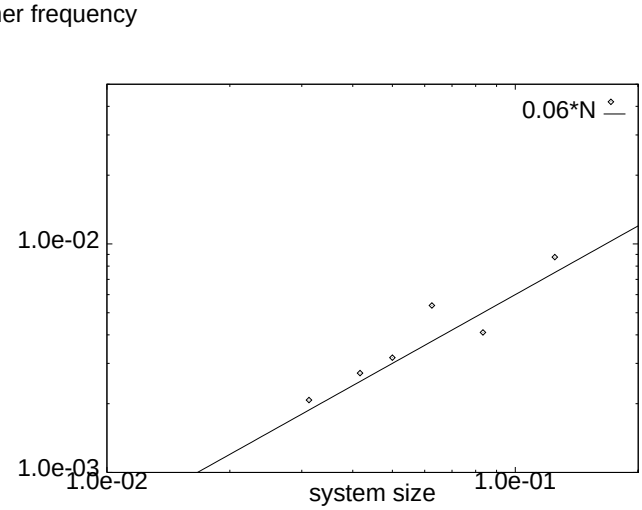

図 2-6: $\lambda = 1$ の時のクロスオーバー周波数 f_cのシステムサイズ N依存性。データは $N = 8, 12, 16, 24,$ および 32 の第 2 モードでとられた。図中の f_cは異なる初期条件についてのサンプル平均であり、エラーバーは f_cのサンプル平均から評価された。直線 $f_c \sim 1/N$ は見やすくするためのガイドである。熱力学極限 $f_c \sim 0$ ではになるであろうから、$1/f$ ゆらぎは非常に観測されにくくなるだろう。

同じように f_cのシステムサイズ N依存性を非線形パラメータを $\lambda = 1$ に固定した状態で調べる (図 . 2-6)。見やすくするために引かれた図中の直線はデータと合わせるため $f_c = 0.06/N$ とした。システムサイズが大きくかつ小さい非線形パラメータという現実的な極限では低励起モードが重要であろうから、低励起モード (第一モードから第四モード) の振舞を調べた。各システムサイズにおける f_c の評価は非線形パラメータとモードを固定し調べた。その場合節の数は固定されるので波数はシステムサイズとともに $1/N$ のように変化する。低波数励起ではフォノン数はフォノン周波数に比例する。そのため PSD の特徴的な構造、固有周波数のピークおよび $1/f$ とホワイトノイズ領域、は固有周波数とともに $1/N$ のようにシフトする。図 . 2-6がその $1/N$ シフトの傾向を示している。言いかえればシステムサイズが変化しても同じ波数なら f_c はほとんど変化しないといえる。低励起モードは多数存在しても PSD の低周波数領域にのみ寄与する。そのため低励起モード相関から来る $1/f$は PSD の低周波数領域にのみ現れると考えられる。この事実は現実の物質のように巨大な系で非線形性が弱い熱力学極限では $1/f\mathrm{PSD}$ を観測す

ることは難しいことを意味している[2]。また、自由度にかかわらず我々の定性的な結果は変わらないので $1/f$ ゆらぎは大自由度をかならずしも必要とはしないと結論された。

[2]巨大な系からなる大域結合系 [Kaneko 92] のように巨大なシステムサイズでは我々の結果もずれを生じるだろうと思われる。

Chapter 3

リアプノフスペクトル

ここでは少数自由度でも $1/f$ ゆらぎを研究できるという利点をふまえ、比較的小さい系 $N = 4, 6, ...$ の位相空間内の運動を調べる。

3.1 Cross Section

前述のように保存系の位相空間はトーラスとカオスからなる [Aizawa et. al. 89]。少数自由度系のマップの知識からトラジェクトリはトーラスとカオスの間をフラクタル的な時間分布で渡り歩く時に $f^{-\mu}$ 性が見出されると推測されているし、そのためにはフラクタル的構造が必要だろうとも推測されている。トーラスとカオス領域が位相空間の中でどう分布するのか、を知ることが出来れば $1/f$ ゆらぎと保存系での運動についての知見も深まる。位相空間の次元が上がると構造を低次元に投影して見るのが非常に困難になるので小さい系から調べ始める。ここではまず $N = 4$ の場合の位相空間を考えてみる。以下初期条件と計算法は前章と同じである。

$N = 4$ の場合は初期条件にかかわらずほとんどすべてのトラジェクトリは f^0 PSD (図. 3-2) を見せる。図. 3-2 と 図. 3-3 の点はトラジェクトリが $u_0 = 0 \pm 0.01$ で定義された 2 次元平面をよぎる点のプロットである。f^0 PSD のトラジェクトリは準周期的であると思われるが、この投影図でもトラジェクトリがかなり規則的に存在しているようにみえる。図. 3-3 のトラジェクトリはある構造の中に捕獲されているように見える。

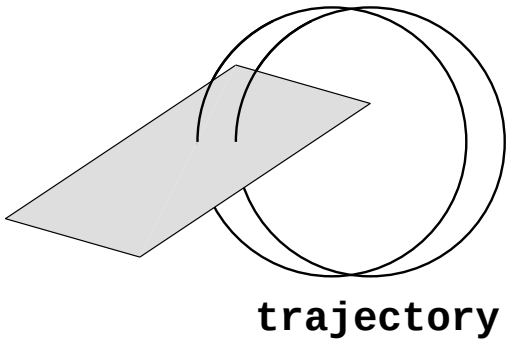

図 3-1: cross section の概念図。トラジェクトリ (線) が与えられた 2 次元平面をよぎる点をプロットしたものである。

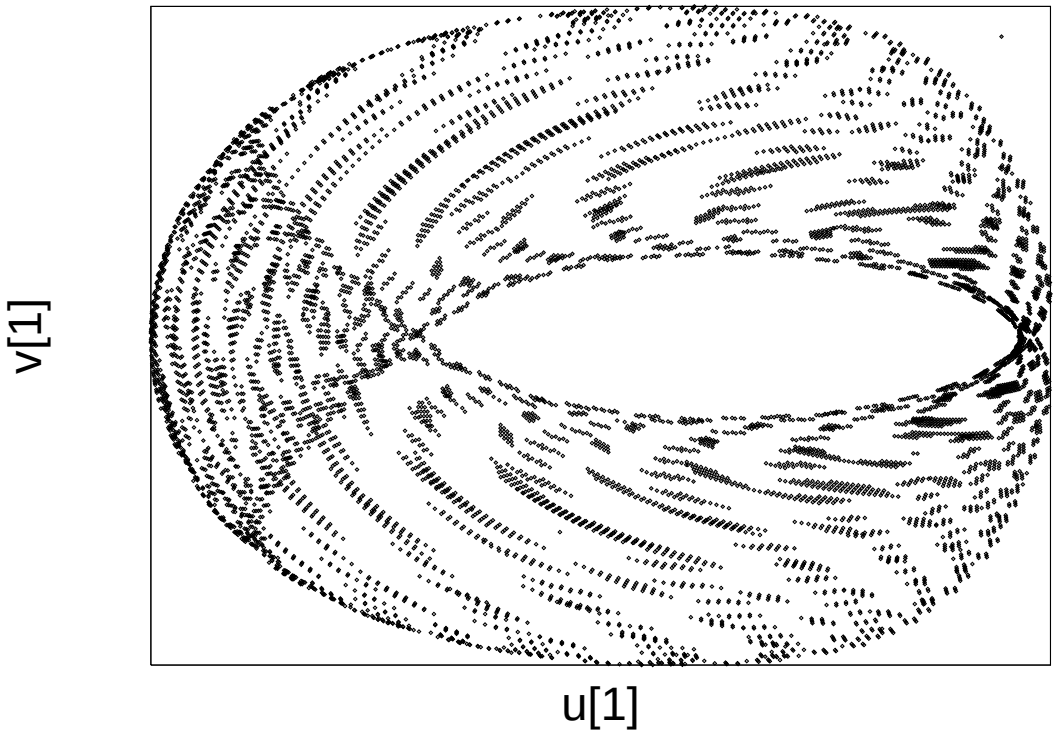

図 3-2: f^{-0} PSD の場合のトラジェクトリの cross section。ここで 2 次元平面は $u_4 = 0 \pm 0.01$ で定義された。f^{-0} トラジェクトリは規則正しくシフトし、準周期的に運動しているように見える。

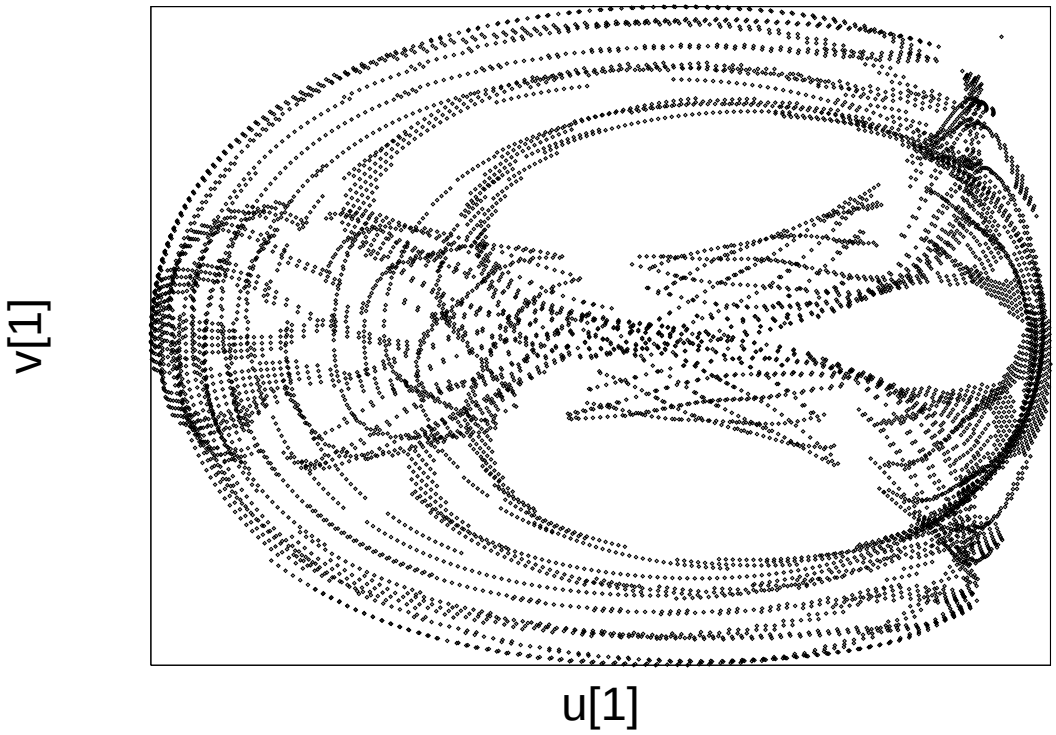

図 3-3: f^{-1} PSD の場合のトラジェクトリの cross section。ここで 2 次元平面は $u_4 = 0 \pm 0.01$ で定義された。f^{-1} トラジェクトリは規則正しい構造を取らず、ある構造にトラップされているように見える。

3.2 リアプノフスペクトル

$N = 6$ の場合を考えてみよう。$N = 4$ の場合は 2 次元に投影しても構造の区別がある程度ついたのだが、位相空間の次元が上がると単なる投影では構造が見えない。このためここではリアプノフスペクトルを用いて位相空間を調べる。

リアプノフスペクトルとは力学系の軌道不安定性の漸近的な振舞 (asymptotic orbital instability) を特徴づけるものである。スペクトルは $2N$ 個の実数の要素$\lambda_1, ..., \lambda_{2N}$ からなる集合である。要素は数学的にはヤコビマトリックス J の自乗の固有値で定義される ：

$$J(t) = \frac{\partial(p(t), q(t))}{\partial(p(0), q(0))} \tag{3.1}$$

$$^tJ(t)J(t) \sim \{\exp(2\lambda_1 t), ..., \exp(2\lambda_{2N} t)\} \tag{3.2}$$

ここで $p(t), q(t)$ はそれぞれ時刻 t での運動量と振幅であり、$t \to \infty$ での収束した値がリアプノフスペクトルといわれる。スペクトルの成分を並べ直し、降順に$\lambda_1, ..., \lambda_{2N} (\lambda_1 > \lambda_2 > ... > \lambda_{2N})$ と書くことにする。

実際のスペクトルの計算には Symplectic Integrator と標準的なグラムシュミット直交化法 [Shimada and Nagashima 79] を用いた。これは式 (3.2) を直接計算しようとすると桁溢れしてしまうので、適当な有限の大きさの t_{unit} ごとに評価されたローカルなリアプノフスペクトルを n 区間分足し合わせるという形で桁溢れを避け $t = t_{unit} \times n$ 時間後のリアプノフスペクトルを評価する方法である。つまり

$$\lambda(t) \sim \sum_n ln e^{\lambda t_{\rm unit}} / \sum n \tag{3.3}$$

のように評価する。一方、リウビルの定理は 「保存系は位相空間の体積を保存する」ことを意味する。リアプノフスペクトルの各要素はどの方向へ伸びどの方向へ縮むか、を記述しているので、$\lambda_i + \lambda_{2N+1-i} = 0$ でなければならないことになる。つまり、ある方向へ伸びたら必ずある方向へは縮み、総和として体積の変化が 0 でなければならない。

保存系のトラジェクトリはトーラスとカオスからなる空間の中を遊走している。トラジェクトリのもつ不安定性はトーラスのそばでは小さく、カオス領域では大きい。しか

しリアプノフ指数 (Lyapunov exponent) は系全体としての不安定性の特徴づけであってローカルな不安定性情報、つまり位相空間の中のどこが不安定でどこが安定か、等の情報は埋もれてしまっている。そのため、位相空間の各部分における不安定を求めるために 「位相空間のある点から出発したときのリアプノフ指数の有限時間内での収束の速度 (temporal decay rate)」を判定条件として用いる。ここでは指数 λ_1 の収束速度を調べた。また、この有限時間でのリアプノフ指数は普通ローカルリアプノフ指数 (local Lyapunov exponent) と呼ばれている。

有限時間 t でのリアプノフ指数の収束速度を次のように見積もることができる。また、t はトラジェクトリが系の特徴的な時間 (固有周波数の逆数の最大値) より長くなるように設定した。式 (3.3) からわかるとおり、我々の計算方法ではローカルな不安定さの度合を足し合わせることでリアプノフ指数を求めている。初期値がトーラス近傍からスタートする場合、リアプノフ指数の収束速度は各区間での収束速度が $\mathrm{e}^{\lambda t}$ のように振る舞わないため、[Konishi and Kaneko 92, Konishi and Kaneko 94]

$$\lambda(t) \sim \sum_{\mathrm{n}} \ln(\sim \mathrm{const.}) / \sum n \sim \frac{1}{t} \tag{3.4}$$

のようになる。逆に、カオス領域の場合は式 (3.3) が正しい場合なのでリアプノフ指数は定数に収束していくはずである。我々は軌道を t 時間測定してλ が $1/t^{\alpha}$ (例 ：α は $0.3 \sim 0.5$) より早く収束するか否か、でトーラス $(\leq 1/t^{\alpha})$ かカオスか $(\sim \mathrm{const})$ を分類する。この基準により位相空間の各点の不安定性の度合を定める。この方法は強力な方法だが位相空間の各点について実行するので膨大な計算時間が必要となる。そのため初期条件をエネルギー平面上に均一に取るのではなく、ひとつの f^{-1} を示すトラジェクトリに沿って初期値を取り、そのトラジェクトリに沿って各点の不安定さを調べた。

3.3 ある構造へ捕獲されること

非線形パラメータが小さい時の位相空間の様子を我々の方法が正しいことを確認するために調べてみよう。この場合はトーラス様の形になるはずだから、それをきちんと観測できなければいけない。実際我々の方法でトラジェクトリが位相空間の中をトーラス様

に走ることを見い出した (図. 3-4) ので、我々の方法が正しく動作することを確認した。

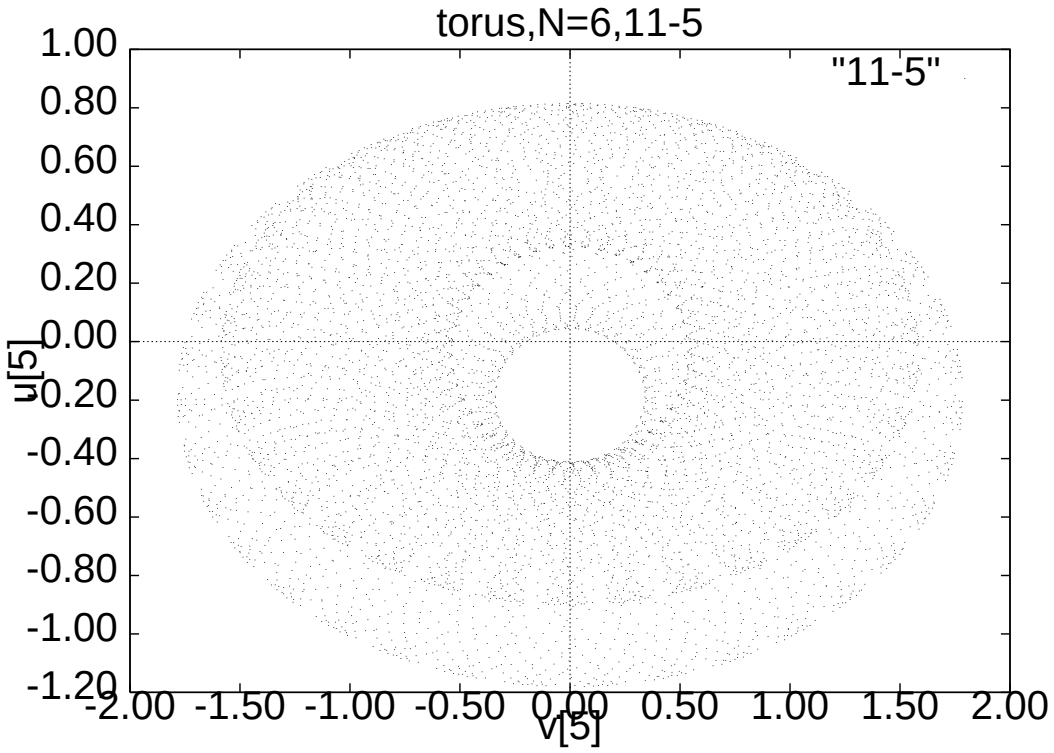

図 3-4: $N = 8, \lambda = 0.01$。点は$\lambda_{max}(t) \leq 1/t^{0.3}$。初期値はトラジェクトリに沿って取られた。

この方法を $N = 6$ の位相空間に適用した結果を位相空間構造を二次元へ投射したものが、図. 3-5 である。ここで $N = 6$、λ は 0.8 である。ここでの点は収束速度が $1/t^{0.3}$ よりも速かった初期値である。つまりトーラスに属する領域と推定される部分のプロットである。トラジェクトリ自体は図. 3-4 のように図全体をめぐっているが、トーラス部分 (円状) だけを見ると、円が円のまわり (図. 3-5中央) を取り囲んで、あたかもたまねぎのような構造に見える [Konishi and Kaneko 92, Konishi and Kaneko 94]。実際、図の空白部分もトラジェクトリ自体はめぐっているがここはトーラスに属さない領域である。また、ここの領域はちょうどモードエネルギーがバーストし、リアプノフ指数によって定められる不安定性が大きくなる場所に対応している。この運動は「トラジェクトリは階層化したトーラスの中を巡り、時々不安定領域に入りバーストを起こす」という描像と一致する。この一致によって $f^{-\mu}$ 性は直接位相空間内の運動を反映していることが推測され

る。また不安定領域を訪ねる事象間の時間 T の滞在時間分布 $P(T)$ もまた $P(T) \sim T^{-2}$ のように振舞う。

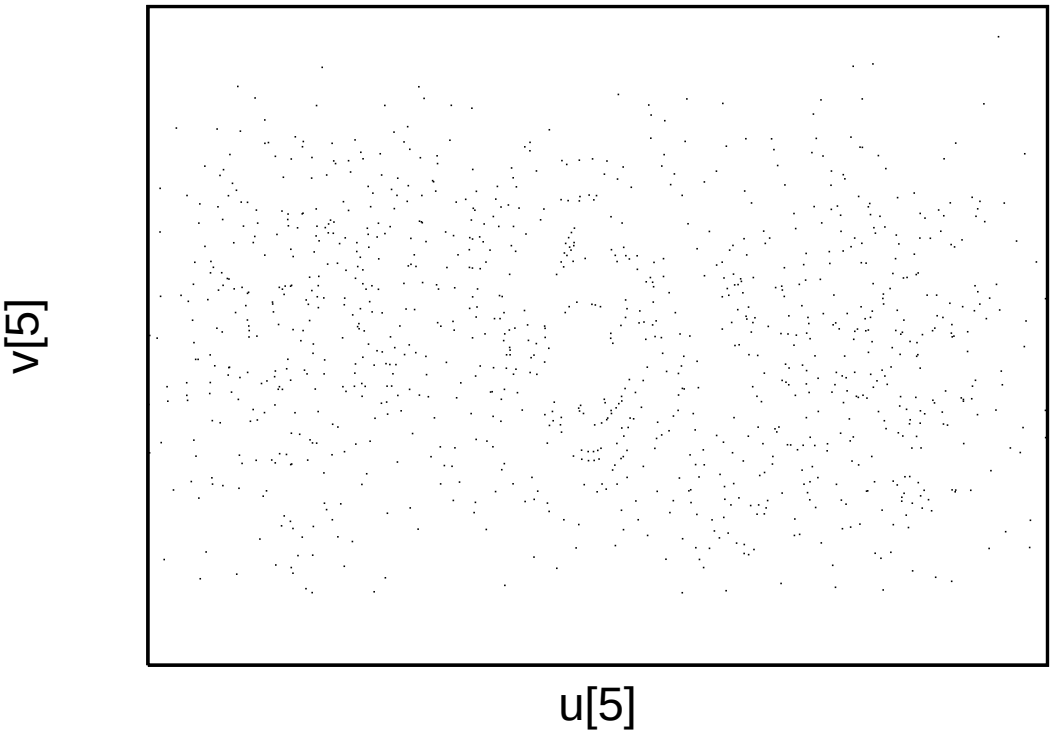

図 3-5: $N = 6, \lambda = 0.8$。ここでの点はローカルリアプノフ指数の収束速度が $1/t^{0.3}$ よりも速かった初期値である。つまりトーラスに属する領域と推定される部分を二次元へ投射したプロットである。トラジェクトリ自体は図全体をめぐっているが、トーラス部分 (円状) だけをみると、円が円のまわり中央を取り囲んであたかもたまねぎのような構造に見える。実際図の空白部分もトラジェクトリ自体はめぐっているが、ここはトーラスに属さない領域である。また、ここの領域はちょうどモードエネルギーがバーストし、リアプノフ指数によって定められる不安定性が大きくなる場所に対応している。この運動は 「トラジェクトリは階層化したトーラスの中を巡り時々不安定領域に入りバーストを起こす」という描像と一致する。

3.4 拡散

バースト性を特徴づける別の量も考えて見よう。ローカル拡散定数 (Local Diffusion Constant) は次のように定義される:

$$D(t) = \langle 1/(Nt) \sum_{i=1..N} (p_{n+t}[i] - p_n[i])^2 \rangle \tag{3.5}$$

ここで $\langle\rangle$ は長時間平均 (つまり n について平均)、N はシステムサイズで、$p_n[i]$ はステップ n での運動量 $\mathbf{p}$ の i 番目の成分である。拡散は t 時間 (ステップ) 後の運動量との差の長時間平均から評価されている。もし普通の拡散プロセスなら定数 $D = D(\infty)$ が存在する。もし拡散にフラクタル的な振舞が含まれるなら $D(t) \sim t^{-\delta}$ (δ は適当な定数) のようになるはずである。しかし f^{-1} と f^0 トラジェクトリの間で $D(t)$ のはっきりとした依存性の違いを見いだすことはなかった。それは瞬間的にバーストする時系列が特徴と考えられる系で、平均操作をしてしまっては特徴の抽出が難しくなるからだと考えられる。

代わりに短時間 t の拡散係数 $d_n(t)$ の分布 $P(d)$ を考えて見よう。$d_n(t)$ の定義は

$$d_n(t) = \sum_{i=1..N} (p_{n+t}[i] - p_n[i])^2 \tag{3.6}$$

である。ここで、n は時刻で、t はここでは 1 である。$P(d)$ はいくつもの $d_n(t=1)$ にわたって平均されたものなので n と $t(=1)$ は以下では省略する。$t = 1$ の時はワンステップでの運動量 $\mathbf{p_n}$ の差と同じになる。図. 3-6 はサンプル平均後の $P(d)$ である。f^{-1} トラジェクトリの $P(d)$ には d の大きいところに山がある。つまり、f^{-1} トラジェクトリは瞬間的に大きい運動量をもつ場所を持つ。これは f^0 トラジェクトリ ではみられない。これはバースト (つまり異常な拡散性をもつこと) につながる現象が $1/f$ につながることを別の形で表現しているのだと考えられる。

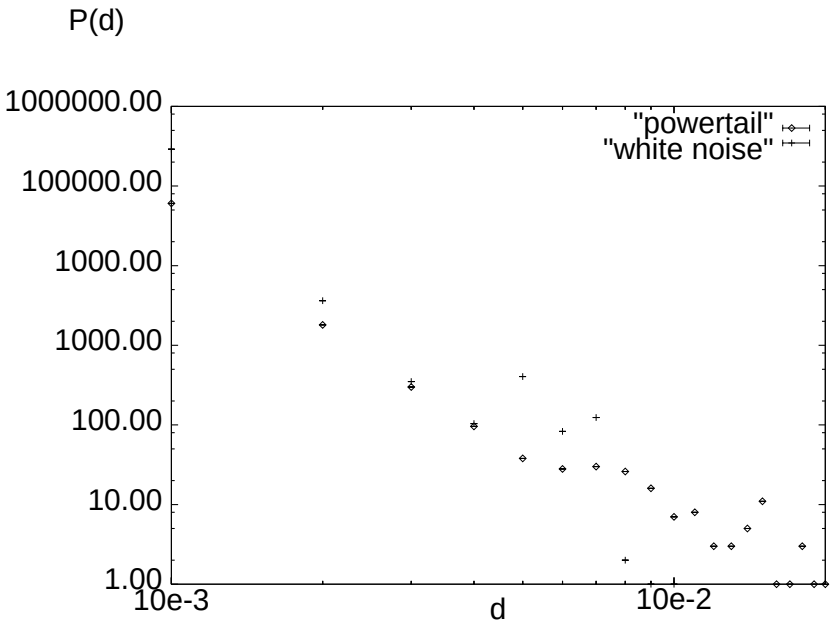

図 3-6: ローカル拡散の分布 $P(d)$。それぞれの点はサンプル平均後のものである。$+$ は f^0 トラジェクトリの $P(d)$ 、$\diamond$ は $1/f$ トラジェクトリの $P(d)$ である。$1/f$ トラジェクトリの $P(d)$ は大きい成分にピーク状の部分をもつが、この部分が $1/f$ をしめすか否か、が重要であると思われる。

Part II

コンピュータネットワークの$1/f$ゆらぎ

Chapter 4

コンピュータネットワークの$1/f$

物質での $1/f$現象に限らず、交通やコンピュータネットワークといった社会的システムでもフラクタル的振舞は多く知られている [Fukamachi 95a, Musha and Higuchi 76, Leland et. al. 94]。しかし、これら全体を統一して説明する$1/f$の理解の仕方というものは知られていない。ここでは間欠性をひとつの手がかりとして実際の系の観測とそのモデル化を考えてみることにする。

コンピュータネットワークは様々な層 (Layer) からなっている。つまり物理層 (イーサネットのようなハードウエア) からアプリケーション層 (実際に使われるプログラムおよびその間の通信。例えば Editor やメールや WWW のようなユーザが使うもの) 等の様々な階層である。コンピュータネットワークにおいても様々なレベル層 (Layer) でフラクタル性を見い出すことが出来る。下の層 (例えば データリンク層) におけるフラクタル性についても見い出すことが出来る (Appendix A) が、ここではユーザにもっとも近い部分つまりアプリケーション層での $1/f$について調べる。

具体的にはメーリングリストとネットニュースシステムの中で流れる情報の量の時系列を調べる。つまりメールの数といった量の時系列である。ここでもまた間欠的な時系列の振舞を見い出すことが出来る。我々は Internet における人間の活動のなかにおけるゆらぎが $1/f$を示すことを見い出したのである。

さらにコンピュータネットワーク全体の形成の仕方についても考察する。というのは Internet[Comver 91] は日々巨大化していく系である。近年のインターネットにおける情

報の流れの総量は定量的に年に 2 倍の割合で増加していると見積もられる[1]。このネットワークの中で人々は毎日メール [Comver 91, Costales et. al. 92] や ftp[Comver 91] そして様々な分散情報システム (distributed information services)[Korl 93] 例 ：wais, gopher, WWW ... を使い、ネットニュースやメーリングリスト [Fukamachi 92-96] に参加している。そういったシステム全体がどのように変化してきたのかを成長に特別な限界のない (open な) 系 [Li 89] として考えてみることにする。

[1]例えばアメリカの NSFNET のあるマシンを通るデータの総量から見積もる。このデータは一般に公開されているものである。

Chapter 5

メーリングリスト

5.1 メールシステム

メールシステムはインターネットにおける最も基本的なシステムの一つである。このシステムは通常の郵便配達のシステムの電子版と考えて差し支えない。基本的なメールの配送メカニズムは二点間接続である。コミュニケーションのより便利なツールとしてメーリングリストというものがある。定義としてはメーリングリストとはあるアドレスのことである。実際のメーリングリストの機能はあるアドレスに来たメールを複製し、メーリングリストの構成メンバーにそのコピーメールを配送することである (Fig. 5.1)。つまり複数相手の文通を電子的に処理していると考えれば良い。それを自動的に実行するプログラムはいろいろ存在する [Costales et. al. 92, Fukamachi 92-96]。普通メーリングリストは同じ興味を持つ人々のコミュニティでの便利な情報伝達手段として使われている。経験的にメーリングリストでのメールの流量 (トラフィック) の時系列は間欠的である。つまり、 「毎日平均してトラフィックは１０通くらい。でも昨日は０通で、今日は３０通を越えた。」というような具合である。この章ではある実際のメーリングリストのトラフィックデータの時系列のゆらぎを調べ、$1/f$ゆらぎを見い出したことを報告する。また、その観測例を説明するためにひとつのモデルを提唱し $1/f$性を説明する。

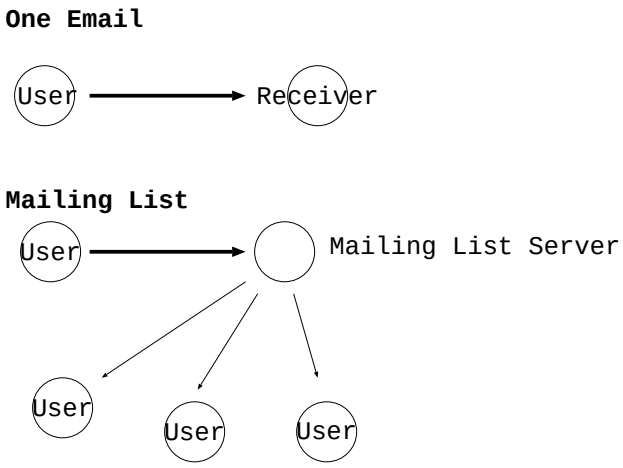

図 5-1: 一通のメールを配送する場合とメーリングリスト経由の配送の比較を示す図。一通のメールならば user から receiver へ配送されるだけである。メーリングリストの場合は user から来たメールはコピーされ、メーリングリストの他のメンバ (メールを送った以外の user) へ配送される。

5.2 メーリングリストとそのモデル

メーリングリストのモデルを考えるためにサイト (i でラベル付けをする) のセットを考えよう。ここで、i は 0 から N までを取る整数である。それぞれのサイトはメールを受け取りそれに対する返事を他のサイトへ送る。すべてのサイトはお互いに接続されているから返事はすべてのサイトへ送られる。現実との対応のためサイト 0 をメーリングリストサーバ [Fukamachi 92-96] と呼ぶことにしよう。

技術的な話をすればサイト i から来たメールはサイト 0 (メーリングリストサーバ) を経由して 0 を除くすべてのサイトへ配られている。実際の配送の時間スケールを考えれば、実質的にはすべてのサイトはサーバなしに接続されていると考えてよい。そのためサイト 0 は本質的に関係ないのでモデルから 0 を落とすことにする。また、我々は 0 以外のサイトはクライアントと呼ぶことにする。さらに今後 $\mathbf{ML}$, $\mathbf{server}$, $\mathbf{client}$ を上記の意味で用いることにする。こういったモデルはクライアント ・サーバモデルと呼ばれる [Leffler et. al. 89]。

サイト間で配送されているメールのデータ構造について考えてみよう。メールは様々

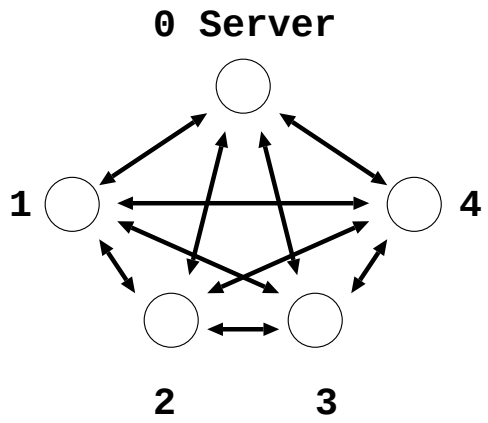

図 5-2: メーリングリストのモデル

な属性から成り立っている。メールは基本的にヘッダとボディからなる。例えばヘッダは送り先アドレス、Subject 等からなるし、ボディはメールの本文である (Fig. 5-3)。そのシンボル表現を mail, header, body のように表すことにしょう。0 個もしくはそれ以上の要素のシークエンスはリスト[1] 構造と呼ばれる。図式的には

```
mail
=> (header body)
header
=> (from to subject)
```

のようになる、ここで⇒ はシンボルの中身を表している。メールの属性 (例 length)、メールというシンボルの中身は

```
mail
=> (length header body).
```

のようになる。我々の観測時間スケールではメールシステムの細かい属性は落としてメールの総量のみに着目すれば十分である。後述するモデル化にあたっては mail の詳細、例えば header の中身の subject を考慮に入れる等の細部を考えることが必要になってくる。

[1]任意の Lisp Object でありうる [Steele Jr. 90]

```
Received: by axion.phys.titech.ac.jp (8.6.12+2.4W/3.1W-yokyun2.1a);
    Wed, 22 Nov 1995 05:12:08 +0900
Received: by beth.phys.titech.ac.jp (8.7.1/Elena-0.1) id FAA07636 for
    fukachan@axion.phys.titech.ac.jp; Wed, 22 Nov 1995 05:12:05 +0900 (JST)
Return-Path: <fml-request@phys.titech.ac.jp>
Date: Sat, 21 Oct 1995 21:27:32 JST
From: "No.6 Fukachan" <fukachan@phys.titech.ac.jp>
Subject: Summary of fix, change these days
To: fml-support@phys.titech.ac.jp (fml supporting ML)
Reply-To: fml-support@phys.titech.ac.jp
Posted: Sat, 21 Oct 95 21:27:35  JST
Message-Id: <199510212127.FML26450@axion.phys.titech.ac.jp>
in-reply-to: your message of "Sat, 21 Oct 1995 00:48:05 JST."
             <199510210049.FML5871@axion.phys.titech.ac.jp>
X-Stardate: [-31]6472.559548
X-ML-Name: fml-support
X-Mail-Count: 00594
X-MLServer: fml[1.6.0.5] (1.6gamma) #rmsc :smtp[1.6.0.2]
X-ML-INFO: If you have a question,
            %echo # help|Mail fml-ctl@phys.titech.ac.jp
Precedence: list
Lines: 53

...Body...
```

図 5-3: メールの例。最初の空行までがヘッダで、それ以降の部分がボディである。

5.3 観測例

観測の実際はあるメーリングリストサーバを通るトラフィックの時系列を測定することである。具体的にはメーリングリストのログを3時間単位に分け時系列を作った。トラフィックの時系列 (通/単位時間、単位は 3 時間) は $10, 3, 4, 0, 1, 2, 30, 10, 0, 3, 3, 0, 2, 0, \ldots$ のようになる。トラフィックはほとんどコンスタントでときどき $0, 3, 3, 0, 30, 10, 3, 0, \ldots$ のようにバーストする。一日あたりの平均的なトラフィックは３０通程度、つまり一年で約一万通のメーリングリストであった。そして最大のトラフィックは１００通／日に達した。ほとんどすべての単位時間辺りのトラフィックは0で3通に満たないものが全体の約９０％を占めた。時々急激にトラフィックは２０~３０通にふくれあがる。つまりここでも FPU の時と同様な間欠性を見い出した。

得られた PSD は典型的な形を示し、高周波数側はホワイトノイズであり低周波数に近付くと $1/f$ 性を示すようになる。ホワイトノイズの中に見えるピークは人間の周期的活動をあらわすものと考えられる。もっとも高周波数側のピークは一日、もっとも低いピークはちょうど一週間に相当している。人間が一日、一週間の単位での周期的な生活をしていることを反映しているからだと考えられる。$1/f$ は一週間を越えた辺りから見えはじめる。一週間以下のスケールでは明らかに周期性とランダム性とから成るであろうから、$1/f$ が一週間を越えた辺りから見え始めることは納得のいくところであろう。

5.4 メーリングリストのモデル化についての予備考察

過去の情報を忘れることは長時間相関がない、つまりホワイトノイズへの収束が起こるということである。逆に $1/f$ ゆらぎは長時間相関なのだから、ある種の系の情報を長い間保持する現象だと考えられる。よってモデル化にあたっての問題はいかにして系の情報を記述し、遠い未来へ伝えていくかというプロセスの設計である。さらにこの場合大事なことは、現在の情報を完全な形で未来へ伝えるのではなく少し不完全に未来へ伝えていくというプロセスでなければ成らないという点である。プロセスが可逆的で、過去と未来が一対一対応する場合には $1/f$ 現象は起こらない。一方ランダムなプロセスを組み合わせた場合には、所詮ランダムプロセスなのだから、PSD ではホワイトノイズしか

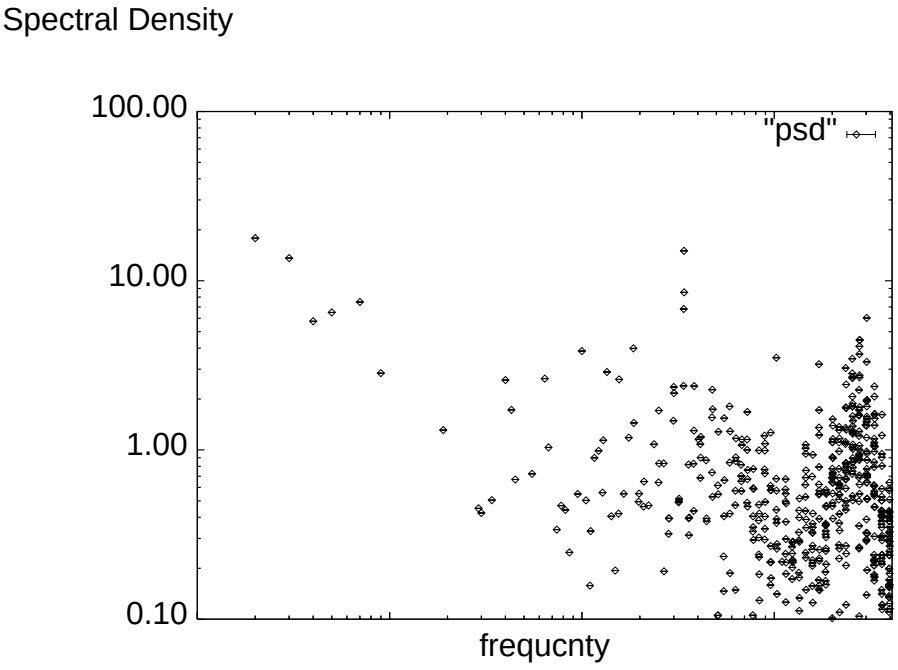

図 5-4: メーリングリストのログから計算された PSD。長時間相関が一週間を越えた辺りから見えはじめることは人間の周期的な活動を考えると納得のいくところであろう。

見い出されない。これらの間の中間的な振舞をモデル化するために遺伝的アルゴリズム (Genetic Algorithm, 略して GA) 的な考え方を導入する。

GA は自然界における進化プロセスの暗喩から導かれた学習理論だといえる。GA は我々の遺伝子に似たビット列で表現された集団をコンピュータ上で仮想的に考え、この集団の中での遺伝子の淘汰を行う。我々はビット表現と遺伝子操作の一部を用い、前述の 「少し不完全に未来へ情報を伝えていく」というプロセスを作り出すことを考えてみたい。

GA とその変形版は普通最適化問題の中で使われる。それは GA は単なるランダムな探索ではなくグローバル探索を行うと考えられているからである。もう少し詳しく述べるとここで言う最適化問題は 「環境にもっとも適応する遺伝子を探す」ということである。我々は分子レベルでの GA のアナロジーとして遺伝子 (DNA) の進化を次のような CROSSOVER、SELECTION、MUTATION という 3 つの操作を組み合わせて行う。CROSSOVER は環境に適応した複数の遺伝子の再結合、SELECTION は環境にもっとも適応した個体はどれか、を選び出す操作、そして MUTATION は進化におけるランダ

ム性を生み出す別の操作である。この３つ操作の組合せが GA である。MUTATION だけではランダム探索にすぎないが、CROSSOVER というグローバルに DNA データを操作する方法が組み合わされることで GA はグローバル探索ができると考えられている。

GA のアルゴリズムは疑似的に次のようにあらわされる ：

```
PSEUDO CODE
   start with an initial time
     initialize the population of genes randomly
     evaluate fitness of all initial genes

     while ( until the given condition is satisfied )
     do {
           increase the time counter
           select a sub-population for offspring production
           recombine the "genes" of selected parents
           perturb genes stochastically
           evaluate the new fitness of genes
           select the survivors from actual fitness
     }
```

まず、初期化しその各 DNA の適応度を決める。その後は次のようなステップ　子孫の中から適応度の高いものを選び　(SELECTION)、それらを組合せ　(CROSSOVER) 再結合し、さらに DNA のデータをランダムに変化させる　(MUTATION)。その後、最も適応した DNA を選びだし、それらだけが生き残る。」ということを　与えられた条件」を満足するまで繰り返す。

5.5　メーリングリストのモデル

ここで我々は経験と観察に基づくメーリングリストの抽象的なモデルを提唱する。スレッド　(thread) というものの進化を考えよう。スレッドとは要するにひとつの話題の連なり

のことである。スレッドは様々な属性を持つだろうが、余分な要素は落とし、次のように２つの変数のみで記述することにする。一つは「人気があるのかないのか」it is active or not now そして、もうひとつは「人気の度合」how active it is である。それぞれをシンボルとして次のように書こう

```
(active-p how-active)
```

ここで p は predicate の略である。スレッドに番号づけをしておくとより便利である。番号 number を i 番目のスレッドに加えて

```
thread
=> (number active-p how-active)
=> (i      active-p how-active)
```

ここで active-p は１ビットで 1 なら「人気がある」、0 ならそうでないとしている。また、how-active は実数である。経験的には、同時に複数のスレッドがメーリングリストの中では走っているものであるから、スレッドの集合を考えそれがメーリングリストの状態を記述していると考えるのが妥当である。このスレッドの集合は superthread と呼ぼう。それはリストとして次のように書ける。

```
superthread
=> ( thread-1  thread-2  thread-3   ... )
=> ( (1 0 1.0) (2 1 2.3) (3 0 0.2)  ... ).
```

一つの superthread が現在のメーリングリストの情報を記述している。便宜上、これは **ML superthread** と呼ぼう。上で述べたようなサイト $i(i = 0, \ldots N)$ の集合を考える時、その集合に対して ML superthread を定義し、それが系全体の様子を記述していると考える。

便宜上、スレッドとスレッドの距離（話題の隔たりとでも言うべきもの）を **D** (**thread**, **thread**$'$) と定義しよう[2]。もし、**D** (**thread-1**, **thread-2**) ~ 0 なら二つのスレッド thread-1 と thread-2 の話題は近いし、そうでないなら違う話題だということである。

[2]もっとも数学的に厳密な定義は不明瞭だが。

本来メーリングリストとは「ある共通の興味を持ったコミュニティ」から成るため、似たような考え方の集団からなる。そのため話題の数も種類も有限個と考えることが出来るし、経験的にそれは正しい。そのためメーリングリストの中のそれぞれのスレッドはお互いに近いと考えることが出来る。しかし時には非常にかけ離れた話題に話が進むこともある。つまり話題そのものは有限だが普段とかなり異なるスレッドも存在しうるということである。これを表現するために予備の superthread とでもいうべきものを ML superthread とは別に用意する。その予備の superthead の集合を情報溜め (Information Reservoir) と呼ぼう。情報溜めを用いることで話題の突然の転換等について記述することが出来るようになる。これらの定義を用いて、スレッドのデータ構造と振る舞いについては以下のようにまとめられる：

1. 人間の興味というものは流動的である。人気は移ろいやすく、すぐに変わっていく。それを active 変数の変化であらわす。例えば一番目のスレッドの人気がなくなるのは

   ```
   (1 0 1.0) -> (1 1 1.0)
   ```

   のように表現される。

2. 確かに話題は移り変わるものだが、短い時間スケールではそんなに大きな変化は起きないものである。つまり ML superthread の集合要素は変わらないが、その中のスレッド一つ一つの人気は移り変わると言える。この話題の移り変わりはスレッドの集合の人気の移り変わりとして記述できる。たとえば一番目のスレッドの人気が落ちて、2 番目の話題が盛り上がるなら

   ```
   (1 1 1.0) -> (1 0 1.0)
   (2 0 1.4) -> (2 1 1.4)
   ```

   もちろん同じことが複数のスレッドにわたって一斉に起きても良い。

3. 非常に異なる話題にいきなり人気が出ることもある。これは ML superthread の中の集合要素も変化していくと考えることで表現される。実際ある程度長い時間スケー

ルで見る時にはML superthreadの中の話題は近いものが多いから、この場合は情報溜めのsuperthead と系の情報交換を行い話題の傾向そのものを変える必要がある。また、このために情報溜めを設定したのである。実際の操作はCROSSOVER で、ML superthreadの一部を情報溜めのsuperthreadの一部と再結合し新しいML superthrad を作る[Kitano ed. 92]。このプロセスの過程では初期値のML superthradの集合が情報溜めの中で切り剥ぎされながら不完全に保存されていることで不完全な情報伝達ということが実装されているのである。

実際のトラフィックの計算の仕方はスレッドごとに各サイトが出すメール (返事) の総量である。サイト i (client server model) の集合を再び考えよう。それぞれのサイトはメールの返事を何通返すのかを次のような関数に従って行う。

```
if active thread (active-p is 1)
        f (random-number, how-active)
else
        0
```

簡単のためこの関数 $f()$ は単なる積としている。つまり、人気のあるスレッドのみがトラフィックへ寄与するということである。

シミュレーションはまず、ML および情報溜め中のすべてのスレッドをランダムに初期化する。簡単のためhow-active 変数はここで一回決定した後は変化しないが、active-p変数はMUTATION の変化を受け変化していく。how-active が変化しないということの意味は話題の盛り上がり具合はいつの時代も同様であると仮定しているからである。これは我々の観測例のような数年のスケール程度でなら充分正しいだろうと考えられる。次にトラフィックをML superthead に従って決定する。その後MUTATION およびML superthread と情報溜め間の CROSSOVER を行う。それぞれMUTATION RATE とCROSSOVER RATE という数値により何回に一度程度操作が行われるかを定義する。これらのステップを 2^{13} から 2^{14} ステップ行った。

MUTATION は新しいスレッド (active-p how-active) を現在のスレッドからランダムに作る。MUTATION は本質的にランダムにスレッドの変数を変えるだけなのでランダムサーチであり、ローカルサーチの役目しか果たさない。この場合システムはすぐに過去

の情報を忘れていく。そのため MUTATION のみによる更新アルゴリズムでは f^0 PSD だけが見い出される。

系の情報の適度な MIXING のため CROSSOVER を適用する。この CROSSOVER 操作が情報溜めを通じて過去と未来をつなぐ。CROSSOVER 操作は 2 つの親 superthread から一つの子 superthread を作る操作である。二親の片方は ML superthread であり、もう一方は情報溜めからランダムにえらばれた superthread である。CROSSOVER の疑似的なコードはつぎのようになる ：

```
PSEUDO CODE for CROSSOVER

Superthread of ML                Superthread in information reservoir
(                                (
(1 active-p how-active)           (11 active-p how-active)
(2 active-p how-active)           (12 active-p how-active)
(3 active-p how-active)           (13 active-p how-active)
   ...                                ...
)                                )

                      Apply CROSSOVER
                      (exchange after 3rd thread).

Superthread of ML                Superthread in information reservoir
(                                (
(1 active-p how-active)           (11 active-p how-active)
(2 active-p how-active)           (12 active-p how-active)
(13 active-p how-active)          (3 active-p how-active)
   ...                                ...
)                                )
```

結果としてシミュレーション全体は次のようにまとめられる [GA FAQ 95] ：

```
PSEUDO CODE

     Initialize superthreads of ML and information reservoir

     while ( until the time end ) {
        for each site {
            calculate the traffic
            sum up the total traffic
        }
        log the total traffic
        MUTATION  at a given  MUTATION RATE to superthread
        CROSSOVER at a given CROSSOVER RATE to superthread
     }
```

小さい CROSSOVER RATE で、CROSSOVER と MUTATION 両方を適用した時 $1/f$ ゆらぎを PSD の低周波数側に見い出すことが出来る (Fig. 5-5)。小さい CROSSOVER RATE (1 %) の時 PSD の低周波数側ではほとんど $1/f$ であるが、すこし大きくなると (16 %) PSD は $1/f$ 領域の後ホワイトノイズに収束してしまう。

それは次のように考えると理解出来る。CROSSOVER RATE が大きい時は、ML superthread と情報溜めの間の情報交換がひんぱんに起こるためすぐに昔の superthread の形跡は分解されていく。そのためすぐにランダムになってしまうのと同じになる。一方 CROSSOVER RATE が小さい時 ML superthread と情報溜めとの間の交換がほとんど起きないため、ML superthread の集合は MUTATION による修正を除けば変化していかない。そのため系全体 (ML superthread と情報溜め) はほとんど変わらないため、かなり長い時間初期条件を覚えていることができる。つまり、なかなか系の情報は MIXING されてはいかない。逆に、系の情報が充分 MIXING されたらその時間スケールより大きいところでは PSD はホワイトノイズに収束してしまうだろう。今の場合どこでホワイトノイズへの変化が起こるかは CROSSOVER RATE が決定している。つまり、この

図 5-5: 我々のモデルから計算した PSD。見やすくするため、$1/f$ の直線を引いてある。実線は CROSSOVER RATE = 1 % であり、点線は CROSSOVER RATE = 16 % の場合である。なお、どちらの場合も MUTATION RATE は 0.03 で固定されている。CROSSOVER RATE が小さい時 $1/f$ ゆらぎが見い出される。小さい CROSSOVER RATE (1 %) の時 PSD の低周波数側ではほとんど $1/f$であるが、すこし大きくなると (16 %) PSD は $1/f$領域の後ホワイトノイズに収束してしまう。つまり、CROSSOVER による系の情報が MIXING される時間スケールより大きいところでは PSD はホワイトノイズに見える。

CROSSOVER 操作 〝いかに少しだけ不完全に情報を未来へ伝達するか」というメカニズムが $1/f$性を制御しているのである。

ここでも $1/f$ゆらぎは滞在時間分布 $P(T)$ で特徴づけることが出来る [Fukamachi 94a]。前と同様にあるトラフィックの時系列でバーストの間のほとんど周期的な振る舞いをする期間を T として分布関数 $P(T)$ を定義する。ここではトラフィックが トラフィックの時系列全体の最大値の 75 % を越えている部分をバーストと定義している。(Fig. 5-6)。Fig. 5-6 は見易くするため累積滞在時間分布

$$\int_{t>T} P(t)dt \sim T^{-1} \tag{5.1}$$

であるので、T^{-1} が $P(T) \sim T^{-2}$ に相当する。ここでも時系列を大雑把に捉える場合なら間欠性という共通の性質がかいま見えるのである。

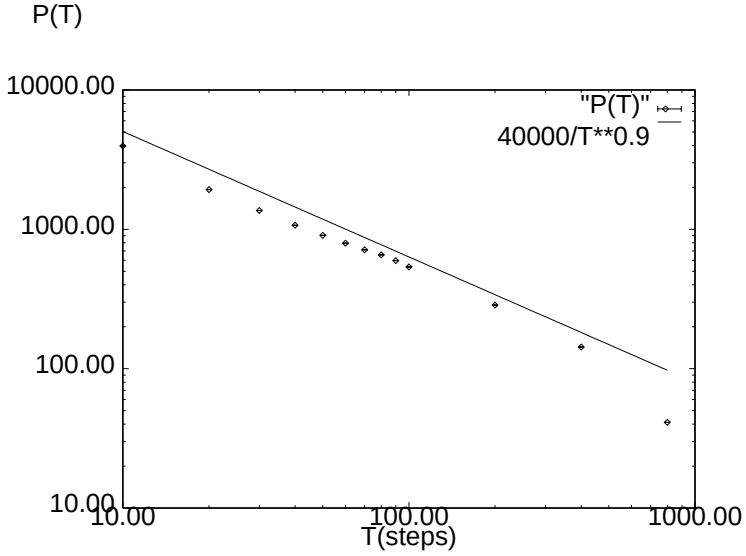

図 5-6: 時系列のバーストしていない状態への累積滞在時間分布。見易くするために直線 $T^{-0.9}$ を引いている。累積分布のためここでの T^{-1} が $P(T) \sim T^{-2}$に相当している。

Chapter 6

ネットニュースシステム

> "A newsgroup is a bulletin board which readers interested in that newsgroup's particular topic can read and respond to messages posted by other readers" [RFC1594 94].

6.1 ネットニュースシステム

前章で述べた仮説を再考するために、メールとは少し異なるネットニュースのシステムというものを考えてみよう。ネットニュースシステムはメールとは違う実装のコミュニケーションツールであるといえる。(図. 6.1)。メーリングリストとネットニュースはコミュニティのサイズと公共性の程度において異なるが実際の挙動は似ているものである。ここでは一日あたりに流れるネットニュースの記事の総量をトラフィックと定義する。我々の経験ではこのトラフィックもまた平均値の回りでゆらぎ、そして突然バーストするものである。つまりネットニュースシステムもまた間欠的のように見えるのである。それは実装の違いと公共性の度合を除けば、見かけ上ネットニュースシステムの「部分」の振る舞いはある共通の興味を抱いたコミュニティと考えられるので、メーリングリストと類似していると考えられるからだと思われる。

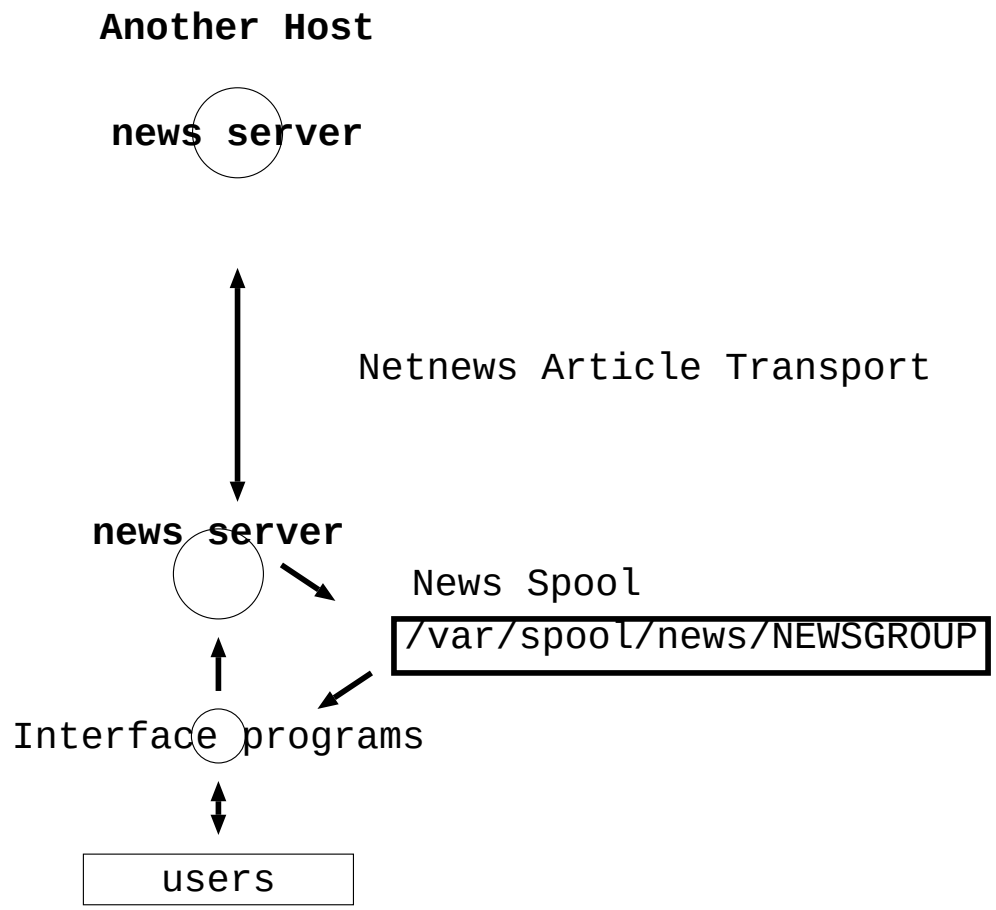

図 6-1: ネットニュースの配送システム例。ニュースサーバは自分の持っている記事を他のニュースサーバへ送る。送られてきた記事はローカルディスクの上に保存されている。ユーザが記事を読み書きするには専用のプログラムを用いてニューススプールへアクセスする。

6.2 ネットニュースシステムのモデル化

前章にならって [Fukamachi 95a]、ネットニュース配送のモデルとして、サイト i の集合を考えよう。それぞれのサイトはローカルにポストされた記事を他のサイトへ配送する。すべてのサイトは互いに接続されている。サイト i から他のサイト j への配送は直接的な 2 点間配送ではない (図. 6.2 の左図)。実際にはサイト i とサイト j の間の配送には多くの中継サイトがある。その仕組みはよくバケツリレーに例えられる。しかしながら、我々の興味のある時間スケール、つまり日 ~ 年のスケールと現在のテクノロジーでの配送は非常に早いことを考えるあわせると細かい実装は無視しても構わない。よってすべてのサイト i はお互いに接続されているとみなしてよい。つまり、ネットニュースとメールシステムは我々の観測条件下では同一と見なして良いことになる。

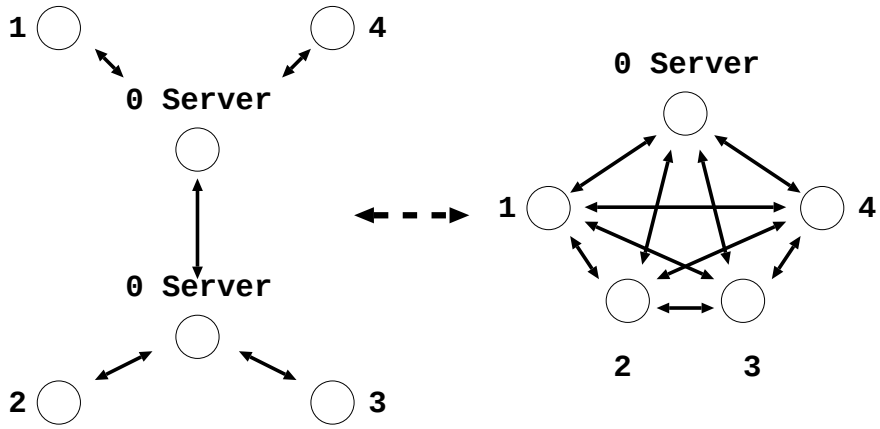

図 6-2: ニュースフィードの例。左の図はサーバとユーザーの間の接続をしめしている。'1', ... '4' はユーザであり、 2 つの '0' はニュースサーバを意味している。 2 つのサーバは同じサイトのマシンでもいいし、全く別の場所にあるマシンでもよい。ユーザ 1 と 4 は記事の読み書きを上の 0 のサーバで行う。同様にユーザ 2 と 3 は下側のサーバで行う。我々の観測時間スケールでは細かい構造は無視できて結局接続の関係は右図と同じになる。

6.3　ネットニュースモデルのデータ構造

前章にならって記事のデータ構造を考えてみよう。記事もまたメール同様ヘッダとボディからなる [RFC1034 87, RFC1035 87]。その要素は前章と同様にリスト構造にまとめることができる。

```
article
=> (header body)
header
=> (path from subject message-id ... )
```

ここで⇒ はシンボルの中身を表している。ヘッダ構造はメールと同様のものだが path という「どう配送されてきたか」の履歴や記事の一意性を保証するための Message-ID 等が付加されている点が異なる。

6.4　ネットニュースシステムの構造

ネットニュースは一連のニュースグループと呼ばれる階層構造からなっている。各ニュースグループはメーリングリストのようなもので特定の話題の記事からなる。階層構造は木構造 (図. 6-4) を成している。ニュースグループは複数の下位ニュースグループからなる。その階層構造はリストにすれば次のようになるだろう。

```
newsgroup
=> (sub-newsgroup-1 sub-newsgroup-2 sub-newsgroup-3 ... )
sub-newsgroup-1
=> (article-1 article-2 article-3 ... sub-sub-newsgroup-1 ... )
```

sub-newsgroup-1 は下位のサブニュースグループを表し、他も同様である。実質的には一つのグループが一つの興味のコミュニティを代表している、つまりスレッドに対応すると考えてよい。ここでもラベルをつけた新しいリスト

```
(label (list))
```

```
Path:phys.titech!axion!fukachan
From: fukachan@beth.phys.titech.ac.jp
Newsgroups: fj.sources
Subject: fml 1.5 ML-Server (0/6)
Supersedes: <FUKACHAN.95Jun12224315@beth.phys.titech.ac.jp>
Followup-To: fj.sources
Date: 12 Jun 1995 15:07:36 GMT
Organization: Department of Physics, Tokyo Institute of Technology
Lines: 568
Distribution: world
Message-ID: <FUKACHAN.95Jun13000736@beth.phys.titech.ac.jp>
NNTP-Posting-Host: beth.phys.titech.ac.jp

深町@東工大物理です

fml の 1.5 snapshot をおおくりします
＃なんとなく仕様も安定したみたいなので

……以下本文省略……
```

図 6-3: ネットニュースの記事の例。これは実際に **fj.sources** へ投稿された記事の最初の部分である。メールと似ているのはネットニュースがメールシステムを参考にしているためである。ヘッダは最初の空行までボディはそれ以降の部分である。

を考えると後々便利である。

```
newsgroup
=> (newsgroup (sub-newsgroup-1 sub-newsgroup-2 sub-newsgroup-3 ... ))
```

ここで最初の要素が newsgroup のラベルである。これを用いて最終的にニュースシステムの階層構造は次のように記述される。

```
newsgroup
=> (newsgroup
    (sub-newsgroup-1 sub-newsgroup-2 sub-newsgroup-3 ...))
=> (newsgroup
     (sub-newsgroup-1
      (sub-newsgroup-2
        (article-1 article-2 sub-sub-groups-1 ...))
      sub-newsgroup-3 ...
    ))
```

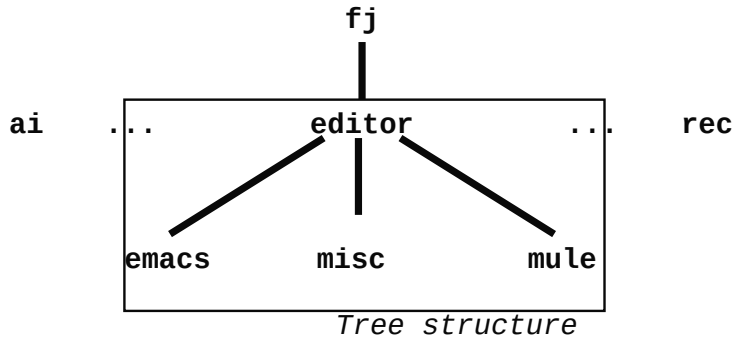

図 6-4: 木構造。UNIX におけるファイルシステム構造と似ている。

6.5 ネットニュースの時系列の観測例

我々はニュースサーバ (ss.titech.ac.jp ドメインのニュースサーバ) のログを用いて、トラフィックのデータとして調べた。各ニュースグループのトラフィックは記事の数で定義する。前章と同様に時系列を Fast Fourier Transform[Aho et. al. 74] で処理した。図. 6-5 と図. 6-6 はトップニュースグループ (ニュースグループの最上位構造) **alt, fj, comp ...** ごとの PSD である。alt(alternative) や rec(recreation) といったチャット的性格の強いニュースグループの記事は $1/f$ 的である。反対に専門的知識を必要とするようなニュースグループ、例えば comp (computer)、では長時間相関は見られずホワイトノイズである。これは専門的知識を必要とするようなニュースグループでは決定的な記事がひとつあればすぐにスレッドを終わらせることが出来る等の理由であまり長期にわたって続いていく話題が少ないのだと考えられる。

前述のようにネットニュースの部分だけ見れば振る舞いはメーリングリストに似ている。そのためニュースシステムをメーリングリストの巨大な集合のように考えることが出来る。よってニュースグループのモデルとメーリングリストは同じであるから、モデル化も全く同様に考えることが出来る。これにより時系列に関しては同様の理解が可能である。

6.6 システムの形成と相関

ネットニュースシステムの進化について考えてみよう。前述のように Internet は年に 2 倍の率で大きくなっていく開放力学系 (Open Dynamical System) と考えることが出来る。その進化のプロセスは簡単なルール

1. ある話題のトラフィックが多すぎる場合、下部 (木) 構造を作り分散させる
2. 新しい種類の話題が現れた場合も、下部 (木) 構造を作り分散させる

という表現にまとめられる。現在のシステムは１５年間それを続けてきた結果のものであると考えられる。

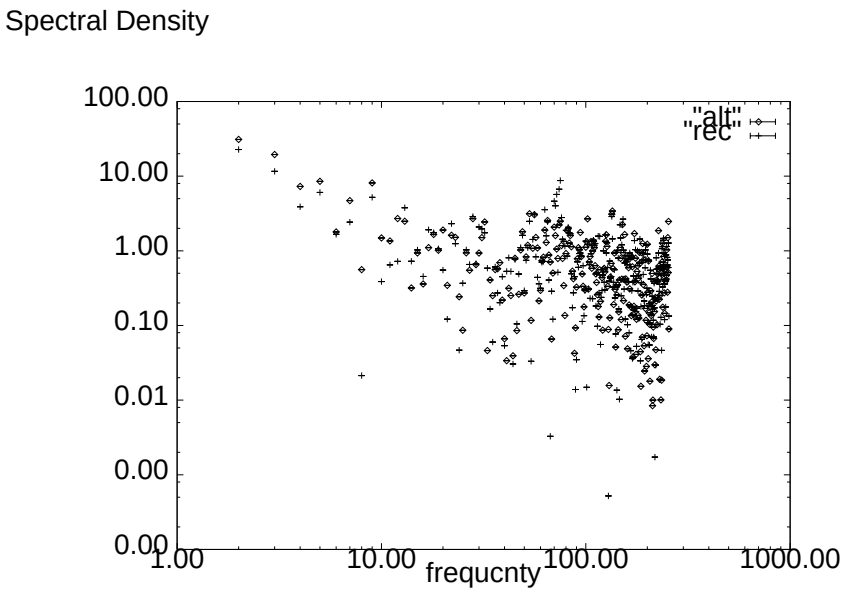

図 6-5: **alt** および **rec** のニュースグループの PSD。PSD はニューススプールの記事の数の時系列から FFT で計算された。alt(alternative) や rec(recreation) といったチャット的性格の強いニュースグループの記事はだらだらと話題が続いていくため、長時間相関が現れるということだとおもわれる。

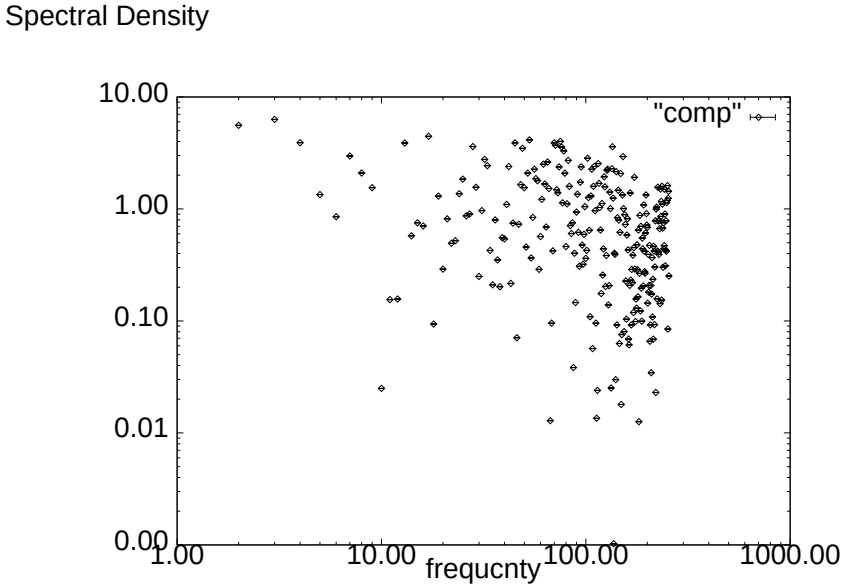

図 6-6: ニュースグループ **comp** の時系列から得られた PSD。PSD はニューススプールの記事の数の時系列から FFT で計算された。低周波数で $f^{-\mu}$ PSD は見られない。これは専門的知識を必要とするようなニュースグループでは決定的な記事がひとつあればすぐにスレッドを終わらせることが出来る等の理由であまり長期にわたって続いていく話題が少ないのだと考えられる。

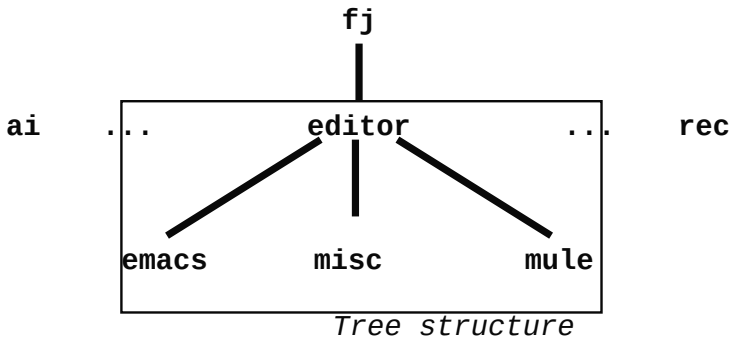

図 6-7: 箱の中が木の様な構造であり、枝が (editor (emacs mule misc)) の各要素である。実際ファイルシステムの上では “/var/spool/news” 以下の構造に対応している。今の場合具体的には、“/var/spool/news/fj/editor/emacs”, “/var/spool/news/fj/editor/misc”, “/var/spool/news/fj/editor/mule” 等に記事は格納されている。

再びリストで木構造 (図. 6-7) を考えてみよう。ニュースグループのトラフィックがある限界を越えると、新しい木構造が作られる (図. 6-4) ということは、次のようになることである。

```
(editor)
=>
(editor (emacs mule misc)).
```

少なくとも 2 つの下部構造の木を持つ新しい木が現在の木構造のどこにでも付け加えられうる。結果として木全体は$\sim 2^N$ のように大きくなる。例えば次のような木である。

```
newsgroup
=> (fj ... (editor
             (emacs
             (mule
               (mule1 ... )
               (mule2 ... ))
              misc))
     ...)
```

実際の木構造の生成プロセスの歴史は “/var/spool/news” の下に刻み込まれている[1]。

実際のニュースシステムの木構造だが、ユーザーにとっては次のような一次元の配列のように見える。

```
(... fj.editor fj.editor.emacs fj.editor.mule fj.editor.mule1 ...).
```

要素それぞれがひとつのある話題に関するニュースグループ、つまりスレッドである。言うなればユーザには一次元的に少しずつ違ったコミュニティが並んでいるように見える。

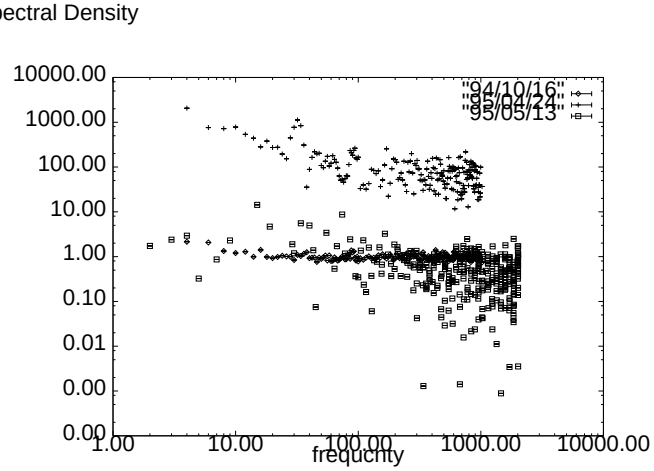

図 6-8: “/var/spool/news” のトラフィックパターンから計算された PSD。ニュースグループ間のトラフィックの相関を示している。PSD は週末のトラフィックの少ない時は f^0 のようになる傾向が見られた。

図. 6-8 は news.phys.titech.ac.jp 上でのニュースグループ間のトラフィックの相関を測定した例である。ニューススプールの木構造は２８００のニュースグループからなっている。一次元へ変換する順番は木構造の生成の仕方を考慮にいれ、UNIX の関数 readdir () system call[2] を繰り返し適用した場合の順である ：

```
(... fj.editor fj.editor.emacs fj.editor.mule fj.editor.mule1 ...).
```

得られた PSD の形は測定した日により異なる。傾向としては $1/f$ PSD はトラフィックが多い日に観測され、逆にトラフィックが少ない (例 :週末) 日の PSD はホワイトノイズだけが見られた。

[1]厳密にはそのサイトがニュースサーバを立ち上げた時点以降の歴史

[2]システムコール readdir() は指定された directory 中のファイルを返す。返される値はファイルがどういう順番で作られたかに依存している。

図. 6-9 は別のサイト (ss.titech.ac.jp) のニュースサーバ上の 6700 のグループについて取られた PSD である。非常に大きいスケールでは相関が見られないが、それはトップニュースグループ (例: fj.editor.mule 等の先頭にある fj のこと。木構造の根 (root) といっても同じ) ごとの相関がほとんどないことを反映しているためと考えられる。つまりトップニュースグループはそれぞれ非常に異なるカテゴリごとに作られるから、トップニュースグループ内での相関はあってもトップニュースグループにまたがるような相関はあまり存在しないということを反映して、大きいスケールでは相関が見られないと考えられる。

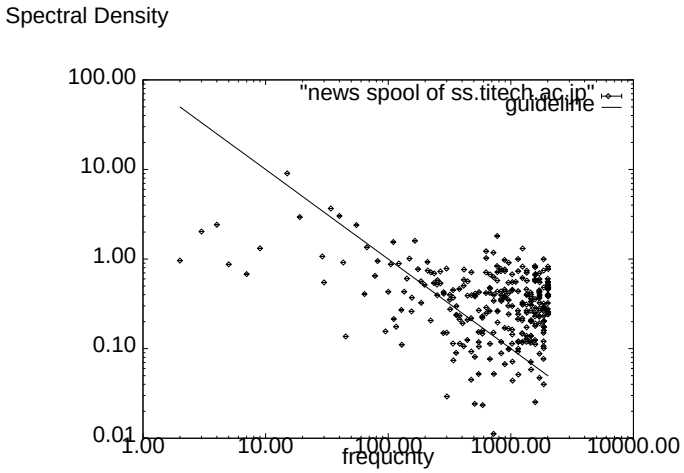

図 6-9: 別のサイト (ss.titech.ac.jp) のニュースサーバ上の 6700 のグループについてとられたニュースグループのトラフィック相関の PSD である。見やすくするためのガイドの直線は $\sim 1/f$ である。非常に大きいスケールでは相関が見られないが、それはトップニュースグループ (例: fj.editor.mule 等の先頭にある fj のこと。木構造の根 (root)) 間の相関がほとんどないことを反映しているためと考えられる。つまりトップニュースグループはそれぞれ非常に異なるカテゴリごとに作られるから、トップニュースグループ内での相関はあってもトップニュースグループにまたがるような相関はあまり存在しないということを反映して、大きいスケールでは相関が見られないと考えられる。

6.7 拡大する系

再び一次元配列に戻って、このような振舞を簡単なモデルから生成することができないか考えてみる。

```
(... fj.editor fj.editor.emacs fj.editor.mule fj.editor.mule1 ...).
```

ニュースグループのトラフィックがあるかないか、はその話題の人気の度合である。簡単のため人気があるか否か、という点のみに注目する。上のリストに対応して 1 ビット、0 か 1、の要素をもつリストを考えよう。

```
(... fj.editor fj.editor.emacs fj.editor.mule fj.editor.mule1 ...)
=> (... 1        0                1               1                ...)
```

ここで 1 はトラフィック (人気) がある、 0 はトラフィックがないということである。この系は拡大系 (expansion system) と呼ぶことにしよう。

このビット列を生成するルールを考える。あるステップ n から次のステップ $n+1$ へのビット列を次のようなビット列が倍化する基本ルールで生成する。

$$(1) \to (11) \tag{6.1}$$

$$(0) \to (00) \tag{6.2}$$

つまり $(1) \to (11)$ は 「トラフィックのあるニュースグループは 2 つの子供ニュースグループを作ることができる」という意味である。その一方ルール $(0) \to (00)$ は奇妙な状況 「はやってないグループにも子供グループが生まれる ?」ということになってしまう。しかしながらこれはある程度まで適用するべきルールである。例えば、「comp.sources のトラフィックがありすぎるので分割して新たに UNIX のソース専用のニュースグループ comp.sources.unix を作ることにする。しかし下位構造を作るならトラフィックがほとんどなくても UNIX 以外の機種、例えば Macintosh のソース用の comp.sources.mac も作るべきである。」といった作り方もありうる。この効果を考慮に入れるためルール $(0) \to (00)$ も必要である。ただしこのルールは常に適用されるわけではなくて、続けて “有限回ステップだけ” 実行されるという補正ルールを適用することにする。その回数をここでは TTL (Time To Live) と呼ぶ。ルール $(1) \to (11)$ は各ステップで 1 というビットに対して常に実行される。ルール $(0) \to (00)$ は新たにランダム反転で (0) が生成されるとその後 TTL 回ステップ適用される。

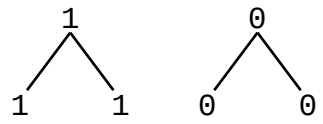

図 6-10: 基本的な expansion のルールは $(1) \to (11)$ と $(0) \to (00)$ である。

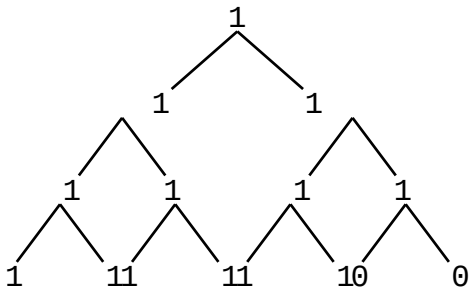

図 6-11: 基本ルール $(1) \to (11)$ および $(0) \to (00)$ は recursive に適用され、結果としてこのような木構造が生成される。

こういった拡大する系に関しては W. Li [Li 89] による例がある。その系は上述のように次のステップのビット列をルール $(1) \to (11)$ と $(0) \to (00)$ から作る。ただしこのモデルでは TTL は無限大である点が異なる。さらに各ステップでランダムな反転 $(1) \leftrightarrow (0)$ を加える。つまり 1 ステップあたりのシステムの発展は 「$(1) \to (11)$ と $(0) \to (00)$ を適用し、その後各要素につき反転するか否か、をランダムにある割合 α に従って決める」という簡単なものである。簡単ではあるが、α が小さい極限で美しい $1/f$ パターンを作り出すことが知られている [Li 89] 。

我々も同様のプロセスを行うが、Li のモデルでは現実に合わないため、TTL というパラメータを導入したのである。TTL は次のように非対称に導入されている ：「ルール $(1) \to (11)$ の TTL は無限大、つまり常に適用。ルール $(0) \to (00)$ は有限。」。このルールは新たにランダム反転で (0) が生成されるとその後 TTL 回ステップ 適用される。全体としてはルール $(1) \to (11)$ が優勢なため系の拡大は $\sim 2^N$ のようになる。

α が小さい時、(0) 要素はあまり生成されない。そのためルール $(1) \to (11)$ だけが系全体の振る舞いを支配している。よって長時間相関は失われ、PSD はホワイトノイズ

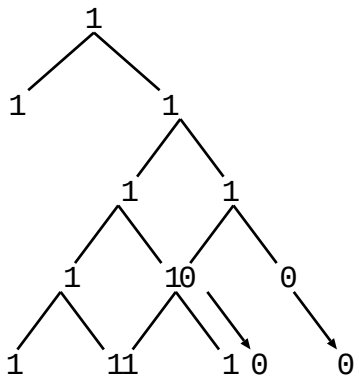

図 6-12: $(1) \to (11)$ ルールは常に適用される。系全体の大きくなりかたとしては $\sim 2^N$ である。一方ルール $(0) \to (00)$ は新たにランダム反転で (0) が生成されるとその後 TTL 回ステップ 適用される。

のようになる。α の大きさが適当で (0) 要素生成がルール $(1) \to (11)$ に圧倒されず (0) と (1) の要素が競合している場合は複雑なパターンが形成され相関が残る。系の TTL への依存性はそれが $1/f$PSD の形を決定するという点で重要である。TTL が小さい時は $(1) \to (11)$ プロセスが系を支配してしまうので、$1/f$領域は急速に f^0 PSD へ収束する。TTL が大きい時は $(1) \to (11)$ と $(0) \to (00)$ は競合する。このような場合は系全体にわたる $1/f$パターンが形成される。図. 6-13 はα を 0.03 に固定し、TTL を変えた時のシミュレーションの例である。

実際のニュースシステムは１５年程度の歴史しか持たない。しかも指数関数的に拡大したのはここ数年のことであるから、現実の系と合わせるためには TTL は $1 <$ TTL < 10 程度になるだろう。このパラメータではある程度の範囲内では $1/f$的な相関が見られるが、大きいスケールでの相関は見られない。これは我々の実際の観測と一致する。

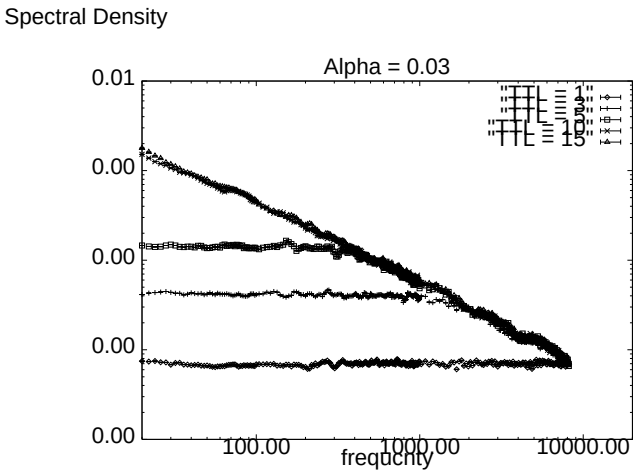

図 6-13: $\alpha = 3$ および $\mathrm{TTL} = 1 \sim 15$ の範囲でのシミュレーションの PSD。PSD はビット列に対して FFT をかけたものである。PSD の形は TTL に依存する。

Part III

結論と今後の課題

Chapter 7

結論と今後の課題

我々は FPU 方程式を調べ、十分低周波数では $1/f$ ゆらぎはクロスオーバー周波数 f_c が $f_c \sim \sqrt{\lambda}/N$ のように振る舞うことを示した。調べたパラメータ範囲では定性的な振舞に特に変化は見られなかった。それは $1/f$ ゆらぎが必ずしも大自由度からくるものではないことを意味していると考えられる。そのことをふまえて $1/f$ の基本構造は低自由度で調べても共通であろうと考え、位相空間を調べた。$1/f$ ゆらぎの基本構造は間欠性カオスと階層性 (フラクタル的構造) からなるとまとめられる。つまりフラクタル的な構造の中で間欠的な振る舞いをもつ系が $1/f$ を示すということである。マップおよびマップの連結した系ではフラクタル的な階層構造は観測できる。本論文では FPU のような偏微分方程式もフラクタル的な構造を持つことを示した。保存系に関する限り、この考え方は妥当であろうと考えられる。しかしながら、この説明では 『$1/f$ の”1”はなぜ特別安定な magic number なのだろうか ?』という疑問には答えてくれない点が将来の課題として残っている。さらに、ここでは一次元での FPU 方程式の例を調べたわけだがより高次元での FPU 方程式や非線形項の入れ方の違いと $1/f$ との関係を調べることも必要だろう。その場合、今よりも位相空間の構造が見極め難くなるであろうから、それにともなう測定法の開発も今後の課題である。

一方、コンピュータネットワークのモデルに関しては、『情報散逸メカニズムを人工的に組み込んだモデル』を考えた。ここでいう『情報散逸』はかなり拡張された意味で用いられている。保存系は散逸をしないわけだが、方程式の変数を時間反転させても不

変であるから、その意味合いで時間反転可能である。それは未来から過去へ遡ることが可能であるという意味でもある。時間反転可能という概念を拡張し、『時間反転のプロセスが 1 対多になる場合でも、過去の状態が再現できるプロセスなら時間反転可能』と呼ぶことにすれば、『時間反転可能性からのずれ』という言い方をしても良い。

ネットニュースのモデルである拡大系は次のような意味において時間反転可能である：『ランダムなビット反転がない時ルール $(11) \to (1)$ およびルール $(00) \to (0)$ により、1 ステップ前の状態を再現することが出来る』。つまりランダムフリップが無いなら完全に過去へ遡ることが出来る。もしランダムフリップの割合α が 0 でないなら、ランダム性がビット列をくずしてしまうため、過去と未来をつなぐ情報の鎖が少しずつ壊されていく。言い変えれば、系の構造は過去を少し不完全になら再現できるものになっている。$1/f$領域の大きさは TTL の大きさ、つまり過去の情報を未来に繋げるメカニズムと相関している。α が小さく TTL が大きいなら、時間軸に沿ってかなり長い相関が生まれるから、$1/f$領域が発生しうる。メーリングリストのモデルでも同様に情報の時間方向の相関の大きさをコントロールするメカニズム（CROSSOVER）と $1/f$領域の大きさは相関してした。結局これらの系の場合『時間反転可能な系からほんの少しずれた時、$1/f$は発現する』とまとめられる。また、そのずれ方の度合が $1/f$領域の存在の仕方を決定している。われわれはこの概念が $1/f$を考える上で重要だと考えている。コンピュータネットワークの 2 つのモデルはこの視点において共に同じであるし、その関係を明らかにしたといえると考えている。その一方、実際の現象に合わせるべく設計された我々のモデルは複雑過ぎる傾向がある。初めからシンボリックに記述することにするのなら Cellular Automata(CA) のような単純なモデルを用いて $1/f$現象のモデルを考えることも可能だろう。その場合は CA の複数のルールをいかに組合せ情報散逸の構造を構築するか、が重要となるだろう。

我々は保存系とコンピュータネットワークを双方を通じて $1/f$現象を調べてきたが、そこで出会った共通の性質は間欠性であった。ここでもシンボリックな表現を用いることでモードエネルギーやローカルリアプノフ指数の時系列でその性質を見い出すことができた。間欠性という考え方は $1/f$を考えるうえでよい手がかりである。系をシンボリックに二状態で考えるという大枠の捉え方が可能な系ならば更新理論 [Aizawa and Kohyama 84]

で $1/f$ を説明することが出来る。これは少し抽象的だが有力な考え方のひとつである。

本論文でのコンピュータネットワークのモデルは 「情報散逸、もしくはわずかに壊れた反転可能性」という考え方が先にあって、そこから誘導されたモデルである点を重視したい。$1/f$ の分野にははっきりとしたパラダイム (ガイドライン)[Kuhn 62,70] があるとはいいがたい。「研究」というものは 「パラダイムに納得がいかないなら、別のパラダイム (ガイドライン) を探す」という要素を含んでいる。間欠性はかなり強力な手がかりであったが、この考え方は大まかな枠であり、結果の解析の指針とはなるが $1/f$ の研究における問題設定等を誘導してくれるパラダイムとしては使っていなかった。求めるパラダイムは $1/f$ の一般的な記述と研究を誘導するものでなければならない。「情報散逸、もしくはわずかに壊れた反転可能性」という考え方はひとつのパラダイムを提供していると考えている。このパラダイムの確認と応用もまた今後の重要な課題である。

謝辞

本論文の研究において田口善弘氏には実り多いディスカッションをしていただき深く感謝いたします。また、小西哲郎助教授と柳田達雄氏には保存系におけるリアプノフスペクトルに関して、佐野光貞氏には Symplectic Integrator に関して有益な助言を与えていただいた。さらに FPU 方程式における $1/f$ ゆらぎという興味深いテーマへ筆者を誘ってくれた川合敏雄教授 (慶応大学) には深い感謝の意を表したいと思います。

コンピュータネットワークにおける研究においてはインターネットを通じて知りえた人々との議論と経験が活かされている。特に、ニュースシステムに関する予備的データ収集について椿井政成氏、針谷喜久雄氏、清水頭武信氏、そして清水敬司氏に協力していただいた。ここに感謝の意を捧げ謝辞としたい。

最後に研究を通じ、多大なる援助をいただいた西森秀稔助教授に深く感謝いたします。

Appendix A

ネットワークの下位層におけるフラクタル

A.1 はじめに

コンピュータネットワークの仕組みは多くの独立したプロトコルを持つ層が重なり合って通信しあうことで実現されている [Tanenbaum 88]。それはちょうど荷物の輸送のようなものである。例えるなら「荷物 (メール等) を送りたい人 (層) がいるとする。その人は宛先を書く係の人 (下の層) に宛先を教え、荷物を預ける。宛先を書く人 (層) は書いた後、発送は他の人 (下の層) にまかせる。発送を受け持つ人は、実際の輸送過程は運送会社にまかせる。そして荷物の配達先では逆の順で相手に届けられる。」といった具合である。メールおよびニュースシステムはユーザに最も近い層に属し、そこはアプリケーション層と呼ばれている。ユーザに見える実際の多くの仕事、例えばファイル転送、電子メール、画面制御等のアプリケーションはこの層に分類される。つまり、この層におけるトラフィックは直接人間の活動を反映しているのだと考えることが出来るだろう。いままではこの層での現象のフラクタル性について考えて来たが、フラクタル的振舞はアプリケーション層以外でも見られる。ここではネットワークの下位層において観測されるフラクタルの例について紹介することにする。

	TCP/IP	OSI
	user process	application presentation session
	TCP, UDP	transport
ICMP, ARP	IP	network
		datalink physical

表 A.1: TCP/IP と OSI 7階層モデルとの対応関係を示す表。

A.2 イーサネットLAN

イーサネット[1] による LAN では転送速度の理論上の限界値は 10M bps (Bit Per Second) であるが、実際には 3M も出れば高性能である[2]。ここで bps は転送速度の単位である。今日では、100Mbps のテクノロジー (例 FDDI, CDDI, 100BaseT, ...) が使われ始めているが、それでも遅い[3]。

イーサネットの中の実際の挙動は 「鬼ごっこ」に例えるのがわかりやすいだろう。つまり 「今通信していいかい？」 「ま～だだよ」もしくは 「いいよ」という具合に、それぞれのマシンが同時に主張しあい通信を制御しているのである[4]。当然このような方式ではある程度トラフィックが多くなるとお互いの要求が衝突し合うことになる。イーサネット LAN (Fig. A.2) の中でほとんどトラフィック (パケット) がない時はトラフィックの時系列はマルコフ過程のランダムプロセスとして記述してよい。一方トラフィックが大きい時は、単純なマルコフ過程では記述し得ない (Fig. A.2 [Leland et. al. 94]) 。Leland 達は非マルコフ過程 [Leland et. al. 94] でコンピュータ・ネットワークを記述すべきだと主張

[1]イーサネットベースで作られたネットワークは今日コンピュータネットワークにおける基本的なインフラストラクチァアになっている。イーサネットは実のところ XEROX で 1970's に開発された古いテクノロジーである。

[2]普通の構成では 3M だが、ネットワークの組み方次第で、１対１の通信ならばほぼ限界の性能を引き出すことも出来る。

[3]速くなれば速くなっただけいろいろなことがネットワーク上で行われるので、回線速度を上げたいという欲求には限りがない、イタチごっこなのである。

[4]Carrier Sense Multiple Access with Collision Detection (CSMA/CD) が特徴とよく言われる [Stevens 90]

図 A-1: イーサネット LAN の例。この場合トラフィックはイーサネット上を流れるパケットの量のことである。

図 A-2: 実際のイーサネット LAN 上で観測されたフラクタル性を示す例。上図 (c) PSD に対応する periodogram と呼ばれるプロット下図 (d) は Hurst parameter と呼ばれる量で、Hurst parameter の 0.5 がちょうど PSD では $1/f$ に相当する。

した。しかし、ネットワーク設計における性能評価の問題に関してはこれでよいのだが、『なぜそうなるのか」という疑問点に関しては研究はあまり進んでいないようである。

A.3 FDDIとイーサネットLAN

別の例として、FDDI Token Ring とイーサネット LAN とからなるネットワークを考えてみよう。FDDI は Fiber Distributed Data Interface の略で、光ファイバーを物理的媒体に用いた 100Mbps の転送速度が可能な LAN の構成要素である。FDDI ネットワークは普通リング状に作るので Token Ring と呼ばれる。当然リング形状に適応したプロトコルで動くので、イーサネットと動作原理はかなり異なる。

動作原理の異なるネットワーク間では相互にプロトコルの変換をしなければならないが、以下で紹介する現象は、FDDI とイーサネットの変換を行う層が不調の原因だと考えられる例である[5]。ある程度以上のトラフィックがある時に、正常であれば常に接続されているはずのネットワークがつながったり切れたりするようになってしまう。この接続の状態はどう変化しているかを時間追跡をした例を示す。コネクションが可能 (1) かだめ (0) かの二つの状態を考えた時、その時系列は間欠的になるのである。

接続しているか否か、は次のように定義し時系列を測定した。本来は ICMP[6] (Fig. A.1) でやるべきだが、実装を簡単にするため TCP/IP ベースで接続可能かどうかを調べる。相手のマシンと TCP レベルで接続可能かどうかを調べ、『実際に接続できたら 1、だめなら 0」といういうことにする (Fig. A.3)。プログラムは『Perl」を用いて、BSD[7] が提供する基本関数 gethostbyname(), getservbyname(), socket() および connect() で相手

[5]データリンク層 (Table. A.1) の現象のように思われる。

[6]ICMP はネットワークの状態を調べるために定義されているプロトコルである。ユーザは ping や traceroute といったコマンドが行うことだと思っていれば十分である。

[7]BSD は『Berkeley Software Distribution 」の省略形である。慣習的に BSD はバークレイ版 UNIX のことをさす。通常のコンテキストではリリース 4.x(特に 4.3 と 4.4) のことが暗黙のうちに念頭に置かれていると考えて良い。4.2BSD はもともとアメリカ国防省の依頼により TCP/IP によるネットワーク機能を標準装備し、ARPANET という今日のインターネットの原型となったネットワークで使われた。4.2 のバグ修正版である 4.3BSD 以降は可能な部分から順次ライセンスフリーのソースコードとして USENET 上に配布され、BSD は USENET 上で議論され、バグを取られ、改良され続けてきた。そして 4.4BSD (ベースの OS) は今ライセンスフリーでありながら、最も完成された優れた Operaging System の一つとして世界中で使われている。そのネットワーク上に公開され人類の共有財産として認識されてきたその歴史は、インターネットの思想基盤を考える点においても重要である。

のマシンの daytime ポート (通常 13/tcp) へ接続できるか否か、で判定した。

疑似的なコードを書けば、次のようになる ：

```
PSEUDO CODE

while
{
    # Get IP Address for $host (the host to connect)
    gethostbyname ($host);

    # which port to connect
    getservbyname ('daytime', 'tcp');

    # make a terminal where we get answers
    socket (S, &PF_INET, &SOCK_STREAM, $proto)

    # try to connect $host.
    if connect (S, $host) is OK.
    else
        # the connection is established.
        state is 1
    then
        # fail of the connection
        state is 0
    fi
}
```

Fig. A.3 はその時系列のアラン分散であり、水平になる時が $1/f$ を示す時間スケールである。

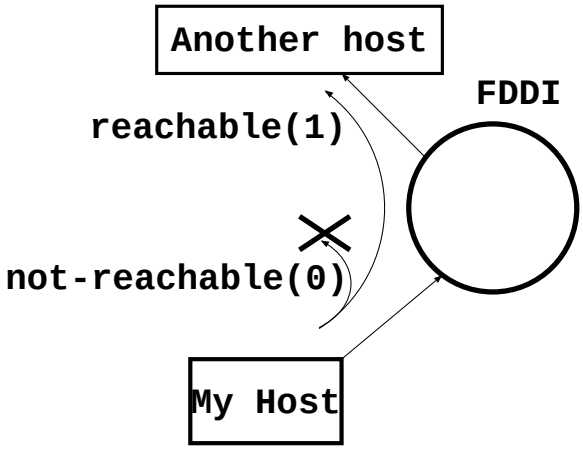

図 A-3: 我々の興味は 『My Host』と 『Another Host』の間の接続性である。ネットワークが正常に接続されているなら、相手のマシンに接続可能であり、その状態は 1、接続できなかったときは 0 である。このビットの時間列を測定すし、やはり間欠的な振舞を見い出した。

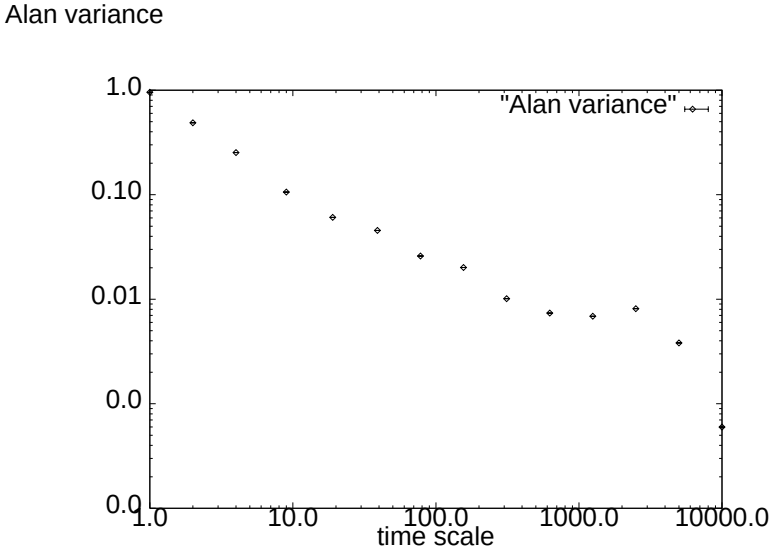

図 A-4: 接続しているか否かの時系列のアラン分散。水平になる時が $1/f$ を示すタイムスケールである。

A.4　まとめ

ここでのふたつのフラクタル性を示す例はメールやネットニュースより時間スケールが違うことで明らかだが、ネットワークの下位層の現象である。ここでも間欠的な振舞が観測される。層がメール等の場合とは異なるため、同様な簡単なモデルは作り難いが、基本的に通信時に渋滞しているというような考え方は同じであるから、直感的には理解しやすい。実際にこれらの現象のモデルを構築をする場合、メーリングリストでは時間スケールの違いにより落とすことが可能であった細部の情報を落とすわけにはいかないだろう。ハードウエアとネットワークの中を流れているパケットの詳細について知っていなければならない。この点がモデル化を困難にしているといえる。

Bibliography

[Goldstein] "Classical Mechanics" ()

[Aho et. al. 74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, 1974) p. 252.

[Aizawa and Kohyama 84] Y. Aizawa and T. Kohyama, Prog. Theor. Phys. **71**, 847 (1984).

[Aizawa et. al. 89] Y. Aizawa, Y. Kikuchi, T. Harayama, K. Yamamoto, M. Ota and K. Tanaka, Prog. Theor. Phys. Supple. **98**, 36 (1989).

[Alan 87] D. W. Alan, *Proc. Symposium on 1/f Fluctuation* (July, Tokyo, 1987).

[Allman 83-96] メールシステムの最も有名なシステム例は Sendmail であろう。現在でも Release 8 が E. P. Allman によりメインテナンスされている。そのソースは例えば 4.4BSD とともに配布されている **4.4BSD-source/usr/src/usr.sbin/sendmail**。Sendmail は各地の anonymous ftp から手に入れることができる。

[Bach 86] M. J. Bach, *THE DESIGN OF THE UNIX OPERATING SYSTEM* (Prentice-Hall, New Jersey, 1986).

[Chirikov 79] B. V. Chirikov, Phys. Rep., **52**, 263 (1979).

[Comver 91] D.E.Comver *Internetworking With TCP/IP, Principles, Protocols, and Architecture*, (Prentice Hall, Englewood Cliffs, NJ, 1991).

[Costales et. al. 92] Costales, Allman, and Rickert, *Sendmail* (O'Reilly & Associates, 1992).

[Dutta and Horn 81] P. Dutta and P. M. Horn, Rev. Mod. Phys. **53**, 497 (1981).

[Dutta et. al. 79] P. Dutta, P. Dimon, and P. M. Horn, Phys. Rev. Lett. **43**, 646 (1979).

[FYI 0018 94] A. Marine, J. Reynolds and G. Malkin, *"FYI on Questions and Answers Answers to Commonly asked "New Internet User" Questions"* (1994).

[Feller 57] W.Feller, *Introduction to Probability Theory* (John Willey & Sons, New York, 1957).

[Fremi et. al. 65] E. Fermi, J. Pasta, and S. Ulam, *Collected Papers of Enrico Fermi* (Univ. of Chicago Press, 1965), Vol. II, p. 978.

[Fukamachi 92-96] メーリングリストサーバの実装例としては私のフリーソフトウエア (これは GNU Public License の下で配布許可を与えている) がある。この「フリー」は「フリーダム」(freedom) であることに注意されたい。公開先として、ネットニュースでは **fj.sources** で、投稿した最後の版の **Message-ID** は

<FUKACHAN.95Jun13000736@beth.phys.titech.ac.jp>
<FUKACHAN.95Jun13001452@beth.phys.titech.ac.jp>
<FUKACHAN.95Jun13001132@beth.phys.titech.ac.jp>
<FUKACHAN.95Jun13001217@beth.phys.titech.ac.jp>
<FUKACHAN.95Jun13001255@beth.phys.titech.ac.jp>
<FUKACHAN.95Jun13001321@beth.phys.titech.ac.jp>
<FUKACHAN.95Jun13001352@beth.phys.titech.ac.jp>

である。また anonymous ftp
ftp://ftp.phys.titech.ac.jp/pub/net/fml-current/
および、World Wide Web

URL: http://www.phys.titech.ac.jp/uja/fml/. では常に最新版を配布している。

[Fukamachi 94a] K. Fukamachi, Europhys. Lett, **26**, 449 (1994).

[Fukamachi 95b] K. Fukamachi, to appear in Physica B (1995).

[Fukamachi 95a] K. Fukamachi, Interdisciplinary Information Sciences, **1**, 157 (1995).

[GA FAQ 95] **comp.ai.genetic (A Guide to Frequently Asked Questions)**

Message-ID: <part1_795637026@cm.cf.ac.uk>

Message-ID: <part2_795637026@cm.cf.ac.uk>

Message-ID: <part3_795637026@cm.cf.ac.uk>

Message-ID: <part4_795637026@cm.cf.ac.uk>

Message-ID: <part5_795637026@cm.cf.ac.uk>

Message-ID: <part6_795637026@cm.cf.ac.uk>

[Hooge 76] F. N. Hooge, Physica **B83**, 14 (1976).

[Jargon 95] Jargon は「comprehensive compendium of hacker slang illuminating many aspects of hackish tradition, folklore, and humor」つまり（正しい意味での）ハッカー文化の用語を伝えるファイルのことである。マスコミが使うハッカーという単語は多くの場合クラッカーもしくはイントルーダーという単語の間違いである。

version 2.9.6 が"The New Hacker's Dictionary" (Eric Raymond (ed.), MIT Press 1991, ISBN 0-262-68069-6) そして、version 3.0.0 が 1993 年秋に第 2 版の"The New Hacker's Dictionary" (MIT Press, ISBN 0-262-18154-1) として出版された。

[Kaneko 92] K. Kaneko, Physica **55D**, 368 (1992).

[Kitano ed. 92] *Genetic Algorithm*, edited by H. Kitano (Sangyou Tosho, Tokyo, 1992)[in Japanese].

[Konishi 89] T. Konishi, Prog. Thoer. Phys. Suppl. **98** 19 (1989).

[Konishi and Kaneko 92] T. Konishi and K. Kaneko, J. Phys. A **25**, 6283 (1992).

[Konishi and Kaneko 94] K. Kaneko and T. Konishi, Physica D **71** 146 (1994).

[Korl 93] E. Korl, *The Whole Internet: Guide for the beginners* (Addison-Wesley, , 1993).

[Kuhn 62,70] Thomas S. Kuhn, *The Structure of Scientific Revolution* (The Universty of Chicago Press, London, 1962, 1970).

[Leffler et. al. 89] S. J. Leffler, M. K. McKusick, M. J. Karels and J. S. Quarterman, *The design and implementation of the 4.3BSD UNIX Operating System*, (Addison-Wesley, , 1989).

[Leland et. al. 94] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, IEEE/ACM Trans. on Networking, Vol.2, No. 1, 1-15 (1994).

[Li 89] W. Li, Europhys. Lett. **10**, 395 (1989).

[Li 91] W. Li, Phys. Rev. **B**, **43** , 5240 (1991).

[Li 92] W. Li and K. Kaneko, Europhys. Lett. **17**, 655 (1992).

[Li 94] W. Li, G. M. Thomas and K. Kaneko, Physica **D75**, 392 (1994).

[Lorentz 63] E. N. Lorentz, J. Atmos. Sci., **20**, 130 (1963).

[Meiss 92] J. D. Meiss, Rev. Mod. Phys., **64**, 795 (1992).

[Minsky 75] M. Minsky, *The Psychology of Computer Vision* edited by P. H. Winston, P. 237 - , "A FRAMEWORK FOR REPRESENTING KNOWLEDGE" (Mcgraw-Hill, 1975), (「コンピュータービジョンの心理」P.H. ウインストン編白井良明杉原厚吉訳 (産業図書,1979)).

[Musha and Higuchi 76] T. Musha and H. Higuchi, Jpn. J. Appl. Phys. **15**, 1271 (1976).

[Musha et. al. 88] T. Musha, A. Nakajima, and H. Akabane, Jpn. J. Appl. Phys. **27**, L311 (1988).

[Musha et. al. 90] T. Musha, B. Gabor, and M. Shoji, Phys. Rev. Lett. **64**, 2394 (1990).

[Pomeau and Manneville 80] Y. Pomeau and P. Manneville, Comm. Math. Phys. **74**, 189 (1980).

[RFC821 82] David H. Crocker, *"SIMPLE MAIL TRANSFER PROTOCOL"* (1982).

[RFC822 82] David H. Crocker, *"STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES"* (1982).

[RFC1034 87] P. Mockapetris, *"DOMAIN NAMES - CONCEPTS AND FACILITIES"* (1987).

[RFC1035 87] P. Mockapetris, *"DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION"* (1987).

[RFC1392 93] G. Malkin, *"Internet Users' Glossary"* (1993).

[RFC1425 93] J. Klensin, N. Freed, M. Rose, E. Stefferud and D. Crocker, *"SMTP Service Extensions"* (1993).

[RFC1426 93] J. Klensin, N. Freed, M. Rose, E. Stefferud and D. Crocker, *"SMTP Service Extension for 8bit-MIMEtransport"* (1993).

[RFC1427 93] J. Klensin, N. Freed, and K. Moore, *'SMTP Service Extension for Message Size Declaration"* (1993).

[RFC1428 93] G. Vaudreuil, *"Transition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME"* (1993).

[RFC1594 94] A. Marine, J. Reynolds and G. Malkin, *"FYI on Questions and Answers Answers to Commonly asked "New Internet User" Questions"* (1994).

[Schutz 68] B. F. Schutz, *Geometrical Methods of Mathematical Physics* (Cambridge Univ., 1968), p. 167.

[Shimada and Nagashima 79] I. Shimada and T. Nagashima Prog. Theor. Phys. **61** 1605 (1979).

[So and Mori 84] B. C. So and H. Mori, Prog. Theor. Phys. **72**, 1258 (1984).

[Steele Jr. 90] Guy L. Steele Jr., *Common Lisp* (Digital Equipment Corporation, Bedford, 1990).

[Stevens 90] W. R. Stevens, *UNIX Network Programing*, (A Simon & Schuster Company, NJ, 1990) ("UNIX ネットワークプログラミング W. リチャード ・スティーヴンス 著 / 篠田 陽一 訳プリンティスホール / トッパン 定価 7,900 円 ISBN4-8101-8509-5").

[Takayasu 86] 高安秀樹 「フラクタル」(浅倉書店,1986)

[Takayasu ed. 87] 高安秀樹編 「フラクタル科学」(浅倉書店,1987)

[Tanenbaum 88] A. S. Tanenbaum, *Computer Networks* (Prentice-Hall , New Jersey, 1988).

[Thompson and Stewart 86] J.M.T. Thompson and H.B. Stewart, *Nonlinear DYnamics and Chaos* (John Willey & Sons, New York, 1986).

[Tsujii and Anzai 88] 機械の知　人間の知 認知科学選書２０辻井潤一 ・安西祐一朗 (東京大学出版会,1988).

[Umeno 94] K. Umeno, Phys. Let. A, **190**, 85 (1994).

[Voss 92] R. F. Voss, Phys. Rev. Lett. **68**, 3805 (1992).

[Wall 89-96] Perl は優れたフリーのプログラミング言語 (のインタープリタ) である。作者は Larry Wall で、現在でもメインテナンスされている。最新版は version 5.001 である。各地の anonymous ftp から手に入れることが出来る。

[Weissman 88] M. B. Weissman, Rev. Mod. Phys. **60**, 537 (1988).

[Weissman 88] M. B. Weissman, Rev. Mod. Phys. **60**, 537 (1988).

[Yoshida 90] H. Yoshida, Phys. Lett. **150A**, 262 (1990).