



Title	2011年度 カオス・フラクタル 講義ノート
Author(s)	井上, 純一
Description	http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/index.html http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/ChaosFractal2011/ChaosFractal2011.html
Issue Date	2011
Doc URL	https://hdl.handle.net/2115/46977
Rights(URL)	https://creativecommons.org/licenses/by-nc-sa/2.1/jp/
Type	learning object
File Information	CF2011_LectureNote.pdf, 講義ノート一括ダウンロード版



カオス・フラクタル 講義ノート

2011年度 ver.1.2

北海道大学 大学院情報科学研究科

井上 純一

この講義ノートについて

この講義ノートは 2011 年度に北海道大学工学部 情報エレクトロニクス学科 情報工学コースにおいて 3 年生を対象にして開講された「カオス・フラクタル」で使用したものである。表紙の版番号「ver.1.2」は、担当者がこの講義を急逝した前任者の「代打」として急遽、担当し始めたのが 2009 年、そのとき作成した版を「ver.1.0」とし、そこから 2010 年、2011 年と 2 年 2 回のマイナーチェンジを経たもの、という意味あいを含めて振られている。カリキュラム改訂等により、この講義が消滅せずいつまで存続するのか、不確定な部分もありはするが、とにかく、今後、大幅な改訂があれば、その版からは「ver.2.x」となる。

過去 2 年分の旧版や関連する講義情報に関しては北大 OCW、北大図書館 HUSCAP および著者のホームページ:

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/

から参照できる。

この講義ノート作成においては数多くの文献を参考にした。それらを巻末にリストアップしてある。担当者は今まで、必ずしも「カオス」「フラクタル」を専門(中心)に研究してきたわけではないため、取り上げた内容(トピック)の取捨選択が適切でないところもあるかもしれないが、その場合には、これら文献を参照することで必要な知識を補って頂きたい。また、工学部での開講ということもあり、多少は「(工学的)応用」にも踏み込むべきだったのかもしれない。しかし、今回は時間的に(そして担当者個人の能力の問題も多分にあり)無理であった。そのなかで「非線形性」「スケールフリー性」「非線形データ解析」など、やや応用寄り、若干、当講義に関連すると思われる話題については、担当者が大学院で開講するいくつかの講義(「混沌系工学特論」「知性創発発達特論」「脳科学入門 V: 神経回路の情報処理」等。これらは上記担当者の HP から内容を閲覧できます)で取り上げているので、そちらを参考にしてもらえれば幸いである(一応「大学院講義」ですが、学部生を含め、どなたであっても時間と興味があれば聴講を歓迎します)。

なお、参考までに、本講義ノート最後にガイダンス配布資料(シラバス)を付した。授業ペースはともかくとして、内容に関してはほぼ、当初のシラバス通りに消化できたと思われる。

内容の記述に関し、単純な誤植はもちろん、担当者の間違った理解で書かれた部分がある可能性も大いにあるが、それらに関してお気づきの点は随時ご指摘/ご教示頂ければ幸いである。

平成 23 年 8 月 札幌の研究室にて
井上 純一

目次

1 「線形」「非線形」とは何か?	6
1.1 簡単な, しかし, 身近で興味深い事例での説明	6
1.2 アカデミックな事例: 脳科学における非線形性	9
2 運動方程式とその線形化: 高校物理の復習	11
3 生態系における個体数の従う微分方程式	12
3.1 ロジスティック方程式の線形化	13
3.2 ロジスティック方程式の差分方程式	14
3.2.1 線形化されたロジスティック方程式の差分	14
3.2.2 非線形なロジスティック方程式の差分	15
4 非線形写像	18
4.1 ロジスティック写像から得られる数列のグラフを用いた求め方	18
4.2 初期値依存性	19
4.3 数値計算上の丸め誤差/精度 (桁) 落ちによる影響	20
4.4 非線形写像の周期について	22
4.5 $x_{n+1} = 4x_n(1 - x_n)$ の周期解とその計算	22
5 軌道の稠密性	27
5.1 ベルヌーイ写像	28
5.1.1 2進小数の復習	28
5.1.2 2進小数のビットシフト演算とベルヌーイ写像	29
5.2 稠密性再考	30
5.3 テント写像とベルヌーイ写像の関係	31
5.4 テント写像とロジスティック写像の関係	31
5.5 軌道の密度関数	32
6 写像の折りたたみ度と軌道のエントロピー	38
6.1 写像の折りたたみ度	38
6.2 軌道のエントロピー	39
7 軌道の安定性とリアプノフ指数	39
7.1 いくつかの計算例	41
7.1.1 テント写像のリアプノフ指数	41
7.1.2 ベルヌーイ写像のリアプノフ指数	41
7.1.3 ロジスティック写像 $L_4(x)$ のリアプノフ指数	41
7.1.4 リアプノフ指数の数値計算例: $L_{a \neq 4}(x)$ のリアプノフ指数	42
7.1.5 参考: 固定点, 周期解に対するリアプノフ指数	43

8	分岐現象とカオス	47
8.1	ロジスティック写像 $L_a(x)$ 再考	47
8.1.1	固定点とその安定性	48
8.1.2	2周期解とその安定性	49
8.1.3	2^n 周期解とその安定性	50
8.1.4	カオスが出現する分岐パラメータ a 値の粗い見積もり	51
8.1.5	窓, 自己相似構造	51
8.2	参考: ニュートン法の復習	52
9	労働市場のカオスモデル	55
9.1	背景: 労働市場におけるフィリップス曲線	56
9.2	Neugart モデル	57
9.2.1	非線形写像の導出	57
9.2.2	非線形写像の分岐構造とカオス	61
9.2.3	カオス・アトラクタとしてのフィリップス曲線	62
9.2.4	失業率と物価上昇率の稠密性と不変測度	63
10	数値計算の準備: 常微分方程式の数値解法	64
10.1	テーラー展開による漸化式の導出	64
10.2	ルンゲ・クッタ法: 基本的アイデア	65
10.2.1	適用例 1: ロジスティック方程式	66
10.2.2	適用例 2: レスラー方程式	67
11	非線形力学系とカオス	73
11.1	レスラー方程式とその振る舞い	74
11.2	ポアンカレ断面	76
12	アトラクタの埋め込み次元と相関次元	79
12.1	埋め込み次元とアトラクタの再構成	80
12.2	次元再考	81
12.3	相関次元	81
12.4	相関次元の一般化	85
13	フラクタル	87
13.1	自己相似性	87
13.2	計算機を使って描かれるフラクタル図形	88
13.2.1	線形写像の確率的切り換えによるフラクタル図形	88
13.2.2	パスカルの三角形とフラクタル図形	90
13.2.3	フラクタル図形を「なか抜き」の繰り返しにより描く	91
13.2.4	粒子のランダムウォークと吸着/凝集によるフラクタル図形	92
13.3	カントール集合	92
13.3.1	作り方	93
13.3.2	カントール集合の3進小数表現と自己相似性	93

14 複素力学系	96
14.1 ロジスティック写像の複素数への拡張	96
14.2 マンデルブロ集合	97
15 確率的フラクタル	98
15.1 シェルピンスキー・ガスケット	98
15.2 菌糸成長の計算機シミュレーション	100
15.2.1 準備: 2次元ランダムウォーク	100
15.2.2 菌糸成長のアルゴリズム	101
15.3 フラクタル次元とその計算方法	105
16 カオス編の演習	108
17 マルチフラクタル	114
17.1 一般化次元の復習	114
17.2 パラメータ q で特徴つけられる「フィルタ」を介して図形を見る	115
17.3 二項分岐過程とそのマルチフラクタル次元	115
17.3.1 描画アルゴリズム	115
17.3.2 マルチフラクタル次元の計算	117
17.4 マルチフラクタル次元の一般論	118
17.4.1 特異性の強さと大域スペクトル	119
17.4.2 二項分岐過程再考	121

この講義の前半では、物理学、数学、生物学や経済学などの社会科学、工学など非常に広範な分野で研究されているカオスについて、その性質や基礎的概念を数値実験等を用いることで理解する。初回である今回はその「イントロ」である。

1 「線形」「非線形」とは何か?

この講義に限らず、カオスやフラクタルの分野では「線形」「非線形」という言葉が頻出する。この講義でも、これらの言葉を多用し、これらが本講義での重要なキーワードとなっているので、これからの講義内容の理解に支障がでない程度に、まずははじめに線形、非線形とは何か、それらがこの講義ではどのような意味合いで使われることになるのかを改めて確認しておきたい。

1.1 簡単な、しかし、身近で興味深い事例での説明

まずは身近な、そして皆さんにとって比較的差し迫った問題に似た例を取り上げて考えてみたい。今年の 10-11 月頃、北海道大学工学部 情報エレクトロニクス学科 情報工学コース 3 年生は希望調査の後、各研究室に配属されることになる。そのとき、人気のある研究室には学生が集まり、そうでない研究室には集まらない。結果として「大所帯」の研究室と「コジンマリ」した研究室ができるとしよう。このとき、横軸に研究室の学生数 x 、縦軸 y にその研究室のアウトプット — 例えば「論文数」 — をとってプロットする。すると、図 1 のように、学生数 x が増えれば増えるほど、アウ

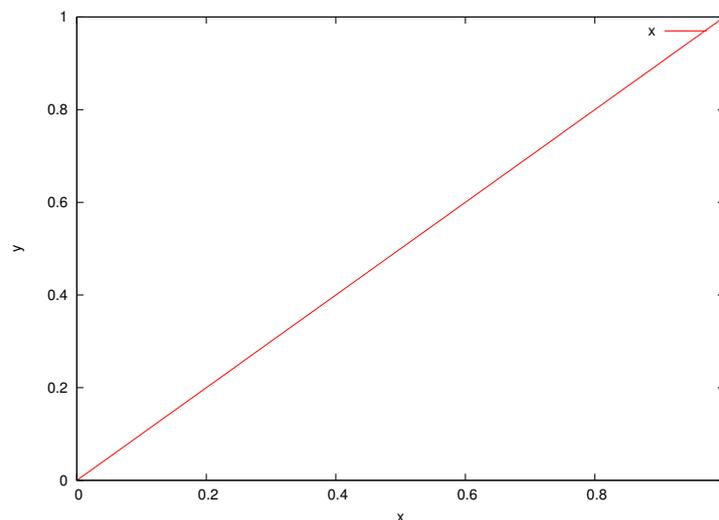


図 1: 学生数 x と研究室のアウトプット y の関係。学生数が増えれば線形に (1 次関数で) アウトプットも増える例。

トプット y は増える¹。それでは「学生数を可能な限り増やす」という戦略が研究室のアウトプット、ひいては、研究室のアクティビティを高めるための最善策なのであろうか?

ここで、改めて考えなければならぬのは、学生数が増えるほど、教員個々への負担も増大し、その結果として、学生一人ひとりとディスカッションする時間は減少し、指導時間が短くなる結

¹ この図の横軸のスケールは適当に決めてます。0.2 目盛りで「一人」と見なしても良いし、0.4 目盛りで一人と考えても良いであろう。実際はどうなのかに関しては… ここではコメントを控えさせていただきます。

果として、ある学生数を越えると「いくら学生を増やしてもアウトプットは増えないで一定値に落ち着く」という状況になりうるかもしれないという事実である (図 2 参照).

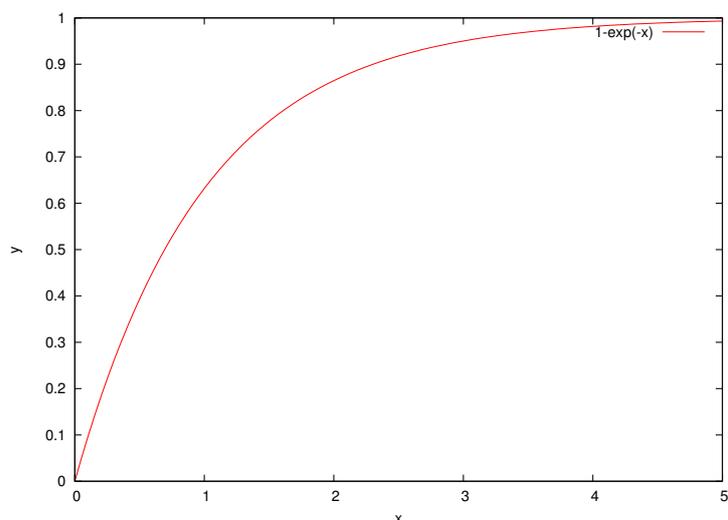


図 2: 学生数 x と研究室のアウトプット y の関係. 学生数が増えれるとはじめはアウトプットも増えるが、ほどなく「頭打ち」になり、平衡値に近づく場合.

そればかりか、もっと最悪なシナリオとして — 人生では常に最悪のケースを想定して行動しなければならないが — 学生数を欲張って増やしたばかりに、本来ならば増えるはずのアウトプットが期待に反して学生数の増加とともに減少して行く場合もありうる (図 3 参照).

図 1 に示したように、学生数に比例してアクティビティも高くなるような状況を学生数の増加 (変化) とアクティビティの増加 (変化) は**線形な関係**にあると言う. 式で書くとすると、学生数を x , アクティビティを y とおけば

$$y = ax \quad (1)$$

と「直線」で書ける場合である (今の例の場合 $a > 0$). 一方、この直線の比例関係から外れる図 2 や図 3 の場合を「線形で無い関係」ということで、**非線形な関係**と言う.

例えば、図 2 のような関係を式で書けば

$$y = 1 - \exp(-ax) \quad (2)$$

と書けるであろうし、図 3 の場合ならば

$$y = ax \exp(-x) \quad (3)$$

と書ける. また、これらを $x = 0$ の近傍でテーラー展開するとそれぞれ

$$y = 1 - \exp(-ax) \simeq ax + \frac{a^2}{2}x^2 + \dots \quad (4)$$

$$y = ax \exp(-x) \simeq ax - ax^2 + \dots \quad (5)$$

となり、 x が十分小さいのであれば、線形な関係 (1) で近似 — **線形近似** — できることがわかる. つまり、この事例で言うならば「学生数を増やせば増やすほど研究室のアクティビティが上がる」と

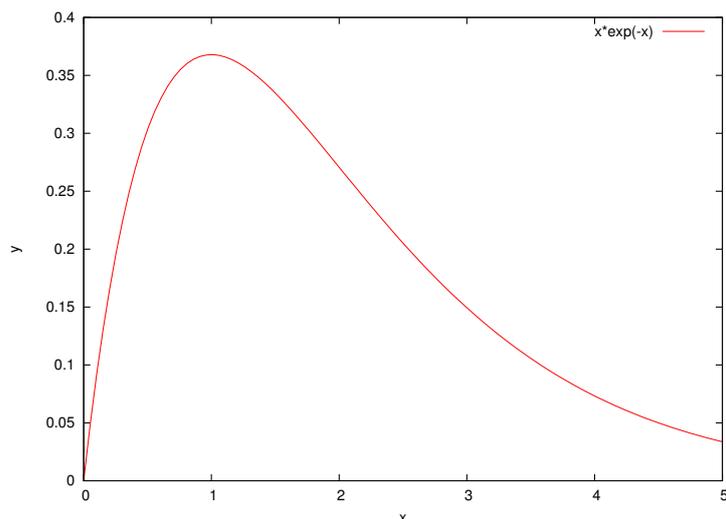


図 3: 学生数 x と研究室のアウトプット y の関係. 学生数が増えれるとはじめはアウトプットも増えるが, ほどなく「頭打ち」の状態になり, さらに学生数が増えるとアウトプットは減少に転じる例.

いう主張は, $x \simeq 1.0$ 程度までの「線形近似」が成り立つ範囲内では正しい². しかし, それ以上に学生数が多くなると非線形性 ((4)(5) の右辺第 2 項以降) が無視することができなく利いてきて一般にこれは成立しなくなる.

ところで, 研究室 A(学生数 x_A) と研究室 B(学生数 x_B) が合併し, 一つの新研究室ができたとしてしよう. すると, この研究室のアクティビティ y_{A+B} は, (1) 式の線形関係が成り立つのであれば

$$y_{A+B} = a(x_A + x_B) = ax_A + ax_B = y_A + y_B \quad (6)$$

となり, 単純に研究室 A と研究室 B のアクティビティの和となる. しかし, (2) や (3) の場合にはそうならない. 例えば, (2) の場合には

$$y_{A+B} = 1 - \exp[-a(x_A + x_B)] \neq 1 - \exp(-ax_A) + 1 - \exp(-ax_B) = y_A + y_B \quad (7)$$

となり, どのように (7) 式を変形しようが, **単純に両研究室のアクティビティの和にはならない.**

さらに, 別な状況として, 研究室へ配属された学生が発奮して ($|c| > 1$)/怠けて ($|c| < 1$), x 人あたりの処理能力 (研究遂行能力) が実質的に cx 人分になったとしてしよう. このとき, 学生数とアクティビティの関係が線形 (1) ならば

$$y' = a(cx) = cax = cy \quad (8)$$

となり, 研究室のアクティビティは単純に c 倍される. しかし, 非線形な関係 (2) の場合には

$$y' = 1 - \exp[-a(cx)] \neq c\{1 - \exp(-ax)\} = cy \quad (9)$$

となり, (9) 式をどのように変形しても, **研究室のアクティビティは単純に c 倍されるわけではない.** よって, 変数 x, y の間の関係が線形ならば, 対象を (6) や (8) のような「好ましい性質」を使って,

² 何人 (x) くらいまで「線形近似」が効くのかは教員や研究課題 (実験系か理論系か等) に依存する. 従って, 研究室配属をシステム論的に考えると, 全体のアウトプットを最大化するには, (全学生数一定のもとに) 研究室ごとに「適正サイズ (比)」が存在するので, その適切サイズのもとに配属された学生を丁寧に指導教育することであろうと思われる. 逆に学生の立場でこれをみれば, 自分が学生の間で人気の無い研究室に配属されたとしても「少人数なので手厚い指導をしてくれるかもしれない」と気持ちを切り換えて, その後の勉学研究に臨むことが重要となろう.

単純な問題として議論することができる。しかし、不幸にも非線形ならば、問題はそんな単純にはならない。

このように、問題を単純にして考えるため、非線形な関係 ((2) や (3) 等) をある限定された条件下で(テーラー展開などを用いて) 線形な関係 (1) として扱うことを、関係式 (もしくは方程式) の**線形化**と呼ぶ。また、 x が大きくなると、上記の線形関係は「近似」でしかなくなり、正確にモノゴトを調べようとする、非線形な関係式をそのまま扱わなければならない。これは技術的には格段に取り扱いが難しくなる場合が多い。よって、それを扱うための道具立てや諸概念を今のうちにしっかりと系統だてて学んでおく必要がある。この講義ではそのための学習を行う。

また一般的に言って、面白く興味深い現象の背後にはこの非線形な関係がある場合が多い³。そのような複雑現象の代表がこの講義で学ぶ**カオス**と**フラクタル**である。

1.2 アカデミックな事例: 脳科学における非線形性

以上はあまりにも身近な日常例を用いた説明だったので、ここからはややアカデミックな話題を用いて「線形」「非線形」を考えてみたい。そこで、脳とその情報処理を取り上げ、それを数理モデル化する際に現れる非線形性について見て行こう。

まずは脳における最小構成要素と考えられている**神経細胞 (ニューロン)** の実物写真を図 4(左) に示す。これに対して、このニューロンの模式図と各部位の名称を図 4(右) に載せる。

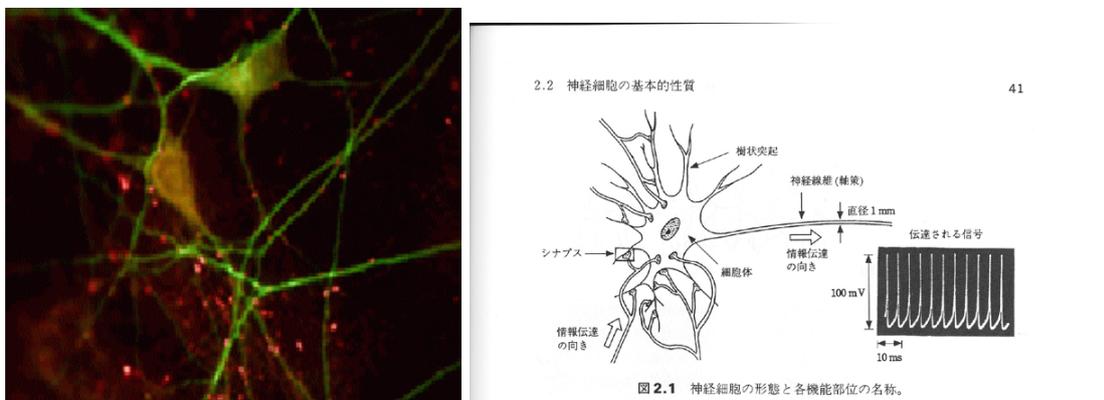


図 4: 実際の神経細胞 (左). 右はニューロンの構成要素とその電圧特性。「神経細胞が行なう情報処理とそのメカニズム」松本元, 大津展之共編, 培風館 (1991) p. 41 の図 2.1 から抜粋。

これらからわかることを要約すると、各ニューロンは**シナプス**と呼ばれる神経機構を介して他のニューロンに結合しており、各ニューロンは信号を電気的パルスとして伝える。電気的なパルスの生成過程は次の様式に従う。すなわち、各シナプスはその結合部分に K^+ , Na^+ 等の正のイオンや Cl^- 等の負のイオンを透過させる**イオンチャンネル**を有し、このチャンネルを調節することにより、シナプス内部と外部の電位差 (通常は -70 mV 程) を調節することができる。一般的には、図 4(右) のようにこの電位差がある値 (約 -55 mV) を越えると高さ 100 V 幅 1 ms の大きさのパルスが生成され、このパルスは軸索上を電気的な信号 (イオンの流れ) として伝わっていく。その後、パルスを生成したニューロンは約 10 ms の「不応期」を経て、また、電位差がある閾値を越えるとパルスを生成する。

これを素朴に数理モデル化すると図 5 のようになる。

³ ここでみた例のような皮肉な結果を「興味深い」と言っているわけではない。念のため。

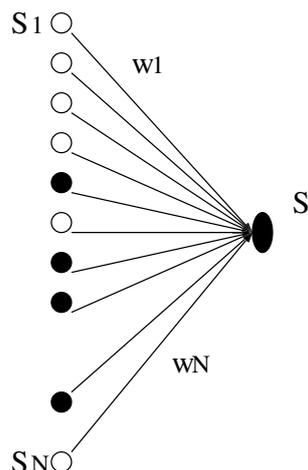


図 5: 神経素子 (ニューロン) の数理モデル. 各素子 $S_i (i = 1, \dots, N)$ は $S_i = 1$ (発火) あるいは $S_i = 0$ (静止) のどちらかの状態をとる. 注目するニューロン S の状態はこれらのニューロンからの重みつき和 $\sum_i w_i S_i \equiv \mathbf{w} \cdot \mathbf{S}$ がある閾値 θ を越えれば $S = 1$ 越えなければ $S = 0$ として決定される.

ここで, 各ニューロンは $S_i = 1$ (発火) か $S_i = 0$ (静止) の 2 つの状態のいずれかを取り, それぞれの値は注目するニューロンに重みつき和

$$h \equiv w_1 S_1 + w_2 S_2 + \dots + w_N S_N \equiv \mathbf{w} \cdot \mathbf{S} \quad (10)$$

$$\mathbf{w} \equiv (w_1, w_2, \dots, w_N) \quad (11)$$

$$\mathbf{S} \equiv (S_1, S_2, \dots, S_N) \quad (12)$$

として入力される. S はこの入力 h に対して

$$h > \theta : S \rightarrow 1 \quad (13)$$

$$h < \theta : S \rightarrow 0 \quad (14)$$

で決定される.

そこで, $\Theta(x)$ を階段関数として

$$\Theta(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (15)$$

で定義し, 簡単のため閾値を $\theta = 0$ とすれば (結合 \mathbf{w} の各成分は負の値も取れるものとする) ニューロンの出力 S は

$$S = \Theta(h) = \Theta(\mathbf{w} \cdot \mathbf{S}) \quad (16)$$

と書くことででき, このニューロンの入力 h と出力 S の関係は研究室配属の例で説明した「非線形な関係」そのものである.

例えば, 注目するニューロン S へ i 番目の他ニューロンからの入力を $h_i = w_i S_i$ で定義し, ニューロン S の出力がこれらの入力に対して「線形」: $S^{(i)} = a h_i$ ならば, N 個のニューロンからの入力の総和に対するニューロン S の出力は

$$S = a(h_1 + h_2 + \dots + h_N) = a h_1 + a h_2 + \dots + a h_N = S^{(1)} + S^{(2)} + \dots + S^{(N)} \quad (17)$$

となり, S の出力はそれぞれのニューロンの入力 of 単純な和で書けることになる. しかし, 出力が (15)(16) 式のような階段関数であれば (17) のような単純な関係式は得られない.

このように非線形素子の数学的な取り扱いが難しくなることが予想されるが, ニューロンをある種の「情報処理機械」として見た場合, この非線形性が思いのほか役立つ側面もある.

例えば, N 個のニューロンからの入力 $h \equiv \sum_{i=1}^N h_i$ に小さな「ノイズ」 ϵ が加わったものがニューロン S に入力されるとしよう. このとき, 線形入出力関係を持つ素子であれば, 出力 S' は

$$S' = a(h + \epsilon) = ah + a\epsilon = S + a\epsilon \neq S \quad (18)$$

となり, 入力誤差 ϵ が a 倍された $a\epsilon$ だけの誤差が出力側からも観測されてしまう. 一方, 非線形素子の場合には

$$S' = \Theta(h + \epsilon) \quad (19)$$

となり, h と ϵ の符号が同じであれば, 出力は $S' = \Theta(h + \epsilon) = \Theta(h) = S$ となって不変であるし, h と ϵ の符号が逆であっても, $|h| > |\epsilon|$ が成り立つ範囲内では $S' = \Theta(h + \epsilon) = \Theta(h) = S$ となり, **ニューロン S の出力は加法的なノイズの影響を受けにくい**ことになる.

このように, 非線形素子はノイズの影響を受けにくい — 入力誤差に対する動作が**ロバスト**である — という利点を持っており, 従って, この「**非線形性**」が**雑音が多い生体内での信号処理を脳がうまく実行している理由の一つ**になっていると考えることができる.

この基本的事実を踏まえて, このようなニューロンの持つ非線形な入出力関係が脳のもつ「記憶」や「学習」などの高次機能にどのように関わっているのかを調べることは興味深く, 実際, 様々な学問分野で研究が進められている. この講義ではその詳細を述べる余裕はないが, 興味のある者は担当者達が大学院で担当している講義「脳科学入門 V: 神経回路の情報処理」:

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/BASIC_MATH/basic_math2011.html

を参考程度に覗いてみると良いであろう.

多くの皆さんにとって, この「脳」に関する話題は初めて聞く内容であったに違いない. しかし, 気がつかなかったかもしれないが, 実は我々が高校生のときに学んだ物理 (力学) でも, 上記の意味での「非線形な方程式」の「線形化」を行い, 問題を単純化して扱っていた. 別な言い方をすれば, 高等学校のときには触れることができなかつた現象を数値計算などを使って調べてみるのがこの講義の目的の一つである. それを次節で取り上げ, ここで述べた「線形」「非線形」の観点から再検討してみよう.

2 運動方程式とその線形化: 高校物理の復習

力学の問題の典型例として, 質量の無視できる長さ l 紐の先に質量 m の質点を取り付け, これを重力加速度 g のもとに振る「単振り子」を思い出そう. この振り子の運動方程式は鉛直方向と振り子のなす角 (ふれ角) を θ として

$$m \frac{d(l\dot{\theta})}{dt} = -mg \sin \theta \quad (20)$$

すなわち

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin \theta \quad (21)$$

となる. この方程式はこのままでは解けないが, θ が十分小さいものとし, $\theta = 0$ のまわりで線形化する

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\theta \quad (22)$$

となり, これは定数係数の線形微分方程式であるから, $\theta = e^{i\lambda t}$ を上式に代入し, λ に関する特性方程式: $\lambda^2 + (g/l) = 0$ を満たす $\lambda = \pm i\sqrt{g/l}$ に対し, 微分方程式の解は α, β を定数として

$$\begin{aligned} \theta &= \alpha e^{i\sqrt{g/l}t} + \beta e^{-i\sqrt{g/l}t} \\ &= (\alpha + \beta) \cos \sqrt{\frac{g}{l}}t + i(\alpha - \beta) \sin \sqrt{\frac{g}{l}}t = A \sin \sqrt{\frac{g}{l}}t + B \cos \sqrt{\frac{g}{l}}t \end{aligned} \quad (23)$$

で与えられる ($\alpha = (A - iB)/2, \beta = (A + iB)/2$). しかし, ふれ角 θ が大きくなってくると, $\sin \theta$ の $\theta = 0$ のまわりの展開での高次の項が無視できなくなり, 運動方程式には

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots \right) \quad (24)$$

のように, θ に関する非線形項 ($-\theta^3/3! + \theta^5/5! - \dots$) が現れる. 前節で非線形項がある場合には取り扱いが難しくなる, と述べたが, この場合もそれが当てはまる. 線形の場合と同じように, $\theta = e^{i\lambda t}$ を (24) に代入して, λ に関する特性方程式を作ろうとすると

$$\lambda^2 = \frac{g}{l} \left(1 - \frac{\theta^2}{3!} + \frac{\theta^4}{5!} - \dots \right) \quad (25)$$

となり, θ に依らない (t に依らない) λ として特性方程式の解が求まるのは, 上式右辺の非線形項 ($-\theta^2/3! + \theta^4/5! - \dots$) が無視できる場合だけであることがわかる. 従って, 非線形振動の方程式は $\theta = e^{i\lambda t}$ のような「調和振動」の解を持たない.

この例での方程式 (21) は楕円積分と呼ばれる特殊関数を用いることで解くことができるが, 多くの場合, 非線形項を含む方程式を陽な形で解くことは難しく, たいていの場合に計算機での数値計算による解法を用いることになる. 本講義で扱うカオスも非線形微分方程式, あるいはそれを差分化した方程式を数値計算することで直接的に確認できる. 従って, この講義ではその手の数値計算の技法のいくつかを学習することも目標の一つとなる. そのような数値計算の技法は次週以降詳しく見ていくことになる.

3 生態系における個体数の従う微分方程式

ここでは力学の問題を離れて, 生態系におけるある個体数 N の従う方程式を考える. N は時間とともに変化する量であるから, $N(t)$ のように時間の関数であることに注意しておこう. このとき, 個体数 N の時間的な増加率が現在の個体数に比例するとすれば, 個体数 N の従う方程式は直ちに

$$\frac{dN}{dt} = \lambda N \quad (26)$$

と書くことができる. ここに λ は正の定数である. これはやはり線形の微分方程式であり, 変数分離操作により解くことができる. 両辺を N で割り, 辺々積分すると

$$\int_{N_0}^N \frac{dN}{N} = \lambda \int_0^t dt \quad (27)$$

つまり, $\log(N/N_0) = \lambda t$ であるから, 個体数は

$$N(t) = N_0 e^{\lambda t} \quad (28)$$

のように時間とともに指数関数的に増加する。

しかし, 実際の生態系ではある特定の個体が増加し続けるということではなく, 周りの環境 (餌の欠乏, 天敵の存在など) により, 個体数が増えすぎると, それを抑制する効果が現れる。そこで, 方程式 (26) を次のように修正する。

$$\frac{dN}{dt} = \lambda N - \mu' N^2 = \lambda N(1 - \mu N) \quad (29)$$

ここに $\lambda, \mu > 0$ である。これを**ロジスティック方程式**と呼ぶ。この方程式からわかることは

- $\mu \neq 0$ であれば, N に関する方程式は線形でない。
- N が大きくなると, N の変化の割合は非線形項: $-\lambda\mu N^2$ の効果により減少する。

この方程式は生態系の方程式ではあるが, 見方を変えれば, 様々な社会科学上の問題にも適用できる。例えば, 最近, 私の自宅の周りにいくつかのコンビニエンス・ストアができたが, 各社がどのような計画 (戦略) のもとに新たな店舗を出しているのか, ということに思いを馳せるととても興味深い。あまり地理的に近い場所に競合店舗を出してしまうと, 顧客の奪い合いになり, 収益が下がることは明らかである⁴。上記生態系での個体を「コンビニエンス・ストア」に, 餌を「顧客」にみなせば, 上記の方程式が当てはまりそうなことはすぐにわかる。実際には, それ以外のファクタも取り込んで行かなければ現実を反映したモデルにならないこともまた明らかなので, 逐次「立地条件」や「人口密度 (空間的な構造)」などの要因を取り込んでモデル (方程式) を複雑にしていく。一般的に方程式が複雑になればなるほど, その非線形性は強くなるので, やはり非線形についての基礎的な知識を養っておくことは重要なのである⁵。

情報科学/工学コースでは数学や物理コースと違って, カオスや非線形方程式そのものを研究することは少ないが, 研究対象を数学的にモデル化し, それを数値的に調べる際, 非線形性/カオスが必然的に現れる (例えば, 2 足歩行の計算機シミュレーション等でもカオスの「歩行パターン」が現れる)。今はピンと来ないかもしれないが, 近い将来, それらの知識を持っている上で問題に取り組むのと, そうでないのとでは, 研究を進めて行く上で明らかに違いがあるのである。

以下ではこのロジスティック方程式をいくつかの側面から詳しく見ていくことにしよう。

3.1 ロジスティック方程式の線形化

ロジスティック方程式は N に関して非線形であるから, 一見すると解くことが難しいように感じるが, 実は適切に変数を選び直すことで線形化でき, それを容易に解くことができる。 $N \neq 0$ に対し, ロジスティック方程式:

$$\frac{dN}{dt} = -\mu\lambda N^2 \left(1 - \frac{1}{\mu N}\right) \quad (30)$$

⁴ もちろん, 実際には, その店舗にしかない特別な商品で差別化をはかることで, 集客力を高める努力もされるであろう。

⁵ この講義で後ほど紹介する予定だが, 我々の研究室で最近はじめた「労働市場」の数学モデルにおいても, この講義で扱う「カオス」が現れる。興味のある者は, 今春に卒業された日野光さんの修士論文が HUSCAP からダウンロードできるので, 軽く読んでみると良いであろう。 <http://eprints.lib.hokudai.ac.jp/dspace/handle/2115/44985> 質問があれば, 著者の日野さんは卒業されてしまったので井上まで遠慮なくどうぞ。

において, $X = 1 - (1/\mu N)$ と変数を変換すると, $dX/dN = 1/\mu N^2$ であるから

$$\frac{dN}{dt} = \frac{dX}{dt} \cdot \frac{1}{\left(\frac{dX}{dN}\right)} = \mu N^2 \frac{dX}{dt} \quad (31)$$

に注意すれば, 新しい変数 X に関する方程式は

$$\frac{dX}{dt} = -\lambda X \quad (32)$$

のような変数分離型となり, 直ちに $X = X_0 e^{-\lambda t}$ がその解となる. 従って, 変数 N に戻すと

$$N(t) = \frac{1}{\mu(1 - X_0 e^{-\lambda t})} \quad (33)$$

となる. $t = 0$ のとき, $N = N_0$ とし, X_0 を N_0 で表すと, $X_0 = 1 - (1/\mu N_0)$ であるから結局

$$N(t) = \frac{1}{\mu - \left(\mu - \frac{1}{N_0}\right) e^{-\lambda t}} \quad (34)$$

が解となる⁶.

3.2 ロジスティック方程式の差分方程式

微分方程式の振る舞いを数値的に調べるための最も簡単な方策は, その方程式を差分方程式に直すことである. ここではロジスティック方程式を差分化することを考えよう. しかし, 線形化したロジスティック方程式:

$$\frac{d}{dt} \left(1 - \frac{1}{\mu N}\right) = -\lambda \left(1 - \frac{1}{\mu N}\right) \quad (35)$$

と非線形のままのロジスティック方程式:

$$\frac{dN}{dt} = \lambda N(1 - \mu N) \quad (36)$$

のどちらを差分化するかによって, 得られる結果が異なるかもしれない. そこで, 以下では上記2つのケースに対し, それぞれ差分化の結果を調べてみよう. (* **これを調べてもらうことが今週のレポート課題**)

3.2.1 線形化されたロジスティック方程式の差分化

(35) 式を差分化すると

$$\left(1 - \frac{1}{\mu N_t}\right) - \left(1 - \frac{1}{\mu N_{t-1}}\right) = -\lambda \left(1 - \frac{1}{\mu N_{t-1}}\right) \quad (37)$$

すなわち, $(1 - 1/\mu N_t)$ は初項が³ $(1 - 1/\mu N_0)$, 公比が³ $(1 - \lambda)$ の等比級数なので

$$\left(1 - \frac{1}{\mu N_t}\right) = (1 - \lambda) \left(1 - \frac{1}{\mu N_{t-1}}\right) = (1 - \lambda)^2 \left(1 - \frac{1}{\mu N_{t-2}}\right) = \dots (1 - \lambda)^t \left(1 - \frac{1}{\mu N_0}\right) \quad (38)$$

⁶ この線形化は単振子の線形化のように注目する変数 (θ) が十分小さい ($\theta \ll 1$) などの近似を使っていないことに注意. $N \neq 0$ の全ての N に対して正確になりつつ.

であるから, これを N_t について解いて

$$N_t = \frac{1}{\mu \left\{ 1 - (1 - \lambda)^t \left(1 - \frac{1}{\mu N_0} \right) \right\}} \quad (39)$$

が得られる. そこで, ロジスティック方程式の解析解 (34) と差分化した方程式の解 (39) 双方を図 6

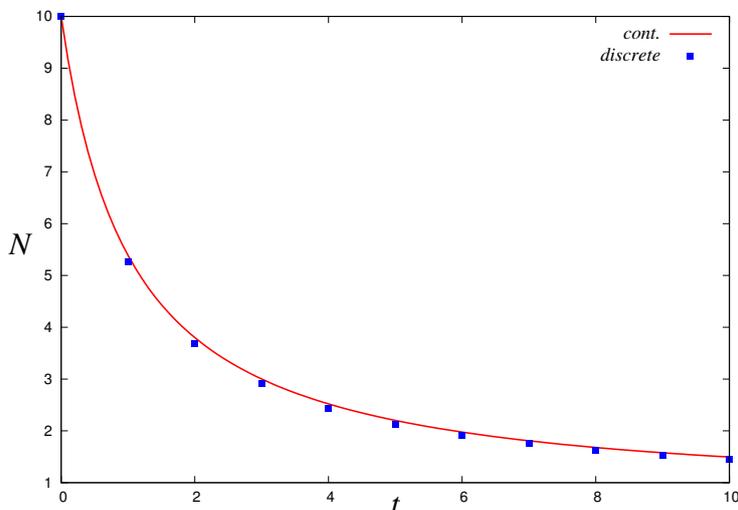


図 6: 線形化されたロジスティック方程式の解析解 ((34) 式, 実線) と差分化された方程式の解 ((39) 式, 四角点). $\mu = 1, N_0 = 10, \lambda = 0.1$ に選んである.

にプロットした. ここに, $\mu = 1, N_0 = 10, \lambda = 0.1$ に選んである. この図より, 線形化したロジスティック方程式を差分化した方程式の解はロジスティック方程式の解析解と極めて近い値を与えることがわかる.

3.2.2 非線形なロジスティック方程式の差分化

次に非線形なロジスティック方程式 (36) を差分化することを考えよう. これも直ちに

$$N_t - N_{t-1} = \lambda N_{t-1} (1 - \mu N_{t-1}) \quad (40)$$

すなわち

$$N_t = (1 + \lambda) N_{t-1} - \mu \lambda N_{t-1}^2 \quad (41)$$

が得られる⁷. 特に, $\lambda = a - 1, \mu = a / (a - 1), a > 1$ と選ぶと, この差分方程式は

$$N_t = a N_{t-1} (1 - N_{t-1}) \quad (42)$$

のように書き直すことができるが, この差分方程式を特に**ロジスティック写像**と呼ぶ.

⁷ 差分化がわかりづらい場合には, $dN/dt \simeq (N_t - N_{t-\Delta t})/\Delta t$ で $\Delta t = 1$ とおき, $dN/dt = N_t - N_{t-1}$ としたと考えるといよい.

この方程式 (42) は何の変哲もない, 一本の単純な 2 項間漸化式に見えるが, その見かけ上のシンプルさに反し, それの示す振る舞いは面白く, 持っている内容は深く, 我々はカオスに関する基本的知識の多くをこの方程式から学ぶことができる.

そこで, この講義の前半 (カオス編) ではこの方程式の振る舞いを詳しく調べることでカオスについての理解を深めていくことになる.

レポート課題 1

(41) 式に対し, N_t を $N_0 = 10, \lambda = 0.1, \mu = 1$ に選ぶことで $t = 0$ から $t = 10$ までプロットし, それが (34)(39) 式からの結果 (図 6) と一致するか否かを調べよ. これは計算機を使わずに電卓でもできる.

レポートは次回の講義開始時に回収します.

注意事項:

この講義では, 毎回のレポート解答例などを前年度までに全員が履修した (計算機プログラミング I, II・同演習) プログラミング言語である C 言語を用いて提示します. しかし, 皆さんの**レポート作成においては特にプログラミング言語を指定しません**. しかし, 万が一, 多くの人々の知らない, マニアックで特殊な言語でソースコードを記述し, それを添付する際には, その旨を明記してください (参考文献や適切な URL など).

課題 1 の解答例

前回のこの講義ではロジスティック方程式の解析解, 2つのタイプの差分方程式を学んだ. 線形化

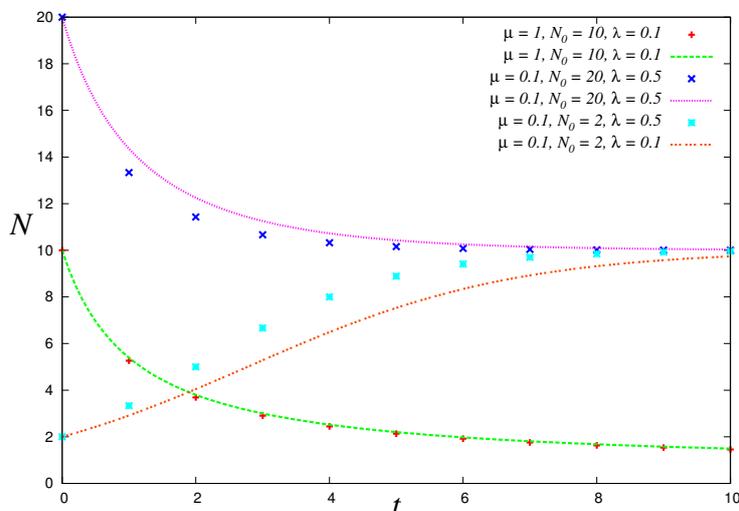


図 7: ロジスティック方程式の解析解と線形化されたロジスティック方程式の差分方程式の比較.

されたロジスティック方程式の差分方程式とロジスティック方程式の解析解との間には各パラメータ (λ, μ, N_0) の選び方によっては図 7 に示したような差異が見られるが, この誤差は評価することができ, 必要とあらば改善することができる. それに関しては後日, この講義の「常微分方程式の数値計算」のところで系統的に学習することになる.

一方, 非線形のロジスティック方程式を差分化したものはこれらの結果と異なるのか否かを見しておくことは必要であろう. 非線形のロジスティック方程式を差分化したものにおいて, 例えば $\lambda = 0.1, \mu = 1, N_0 = 10$ とおくと

$$N_t = (1 + 0.1)N_{t-1} - 0.1N_{t-1}^2 \tag{43}$$

であるから, はじめの数項を計算することにより

$$N_1 = (1 + 0.1)N_0 - 0.1N_0^2 = (1 + 0.1) \times 10 - 0.1 \times 100 = 1 \tag{44}$$

$$N_2 = (1 + 0.1) \times 1 - 0.1 \times 1 = 1 \tag{45}$$

であるから, 以降では常に 1 である. これは, ロジスティック方程式の解析解, 線形化したロジスティック方程式の解析解とも異なる. また, 当講義ノート 15 ページの最後で述べたようにそれぞれの方程式で $\lambda = a - 1, \mu = a / (a - 1), a > 1$ とおき, $a = 3.9$ と選んでプロットすると図 8 のようになる. この図より, 解析解は滑らかにある一定値に近づくが, 線形化されたロジスティック方程式の差分方程式の振る舞いはこれと異なり, 初期段階では振動しながら最終的にはある一定値に近づいている (これらは (33)(39) 式の形からもある程度予想できる). この一定値はロジスティック方程式の解析解のそれとは異なっている. 一方, 非線形ロジスティック方程式の差分方程式の振る舞いはこれら 2つの方程式の振る舞いのどちらとも全く異なり, 極めて不規則な挙動を示すことがわかる. また, この振る舞いを実際に計算する前に第 1 回講義ノート (42) 式から予想することは難しい.

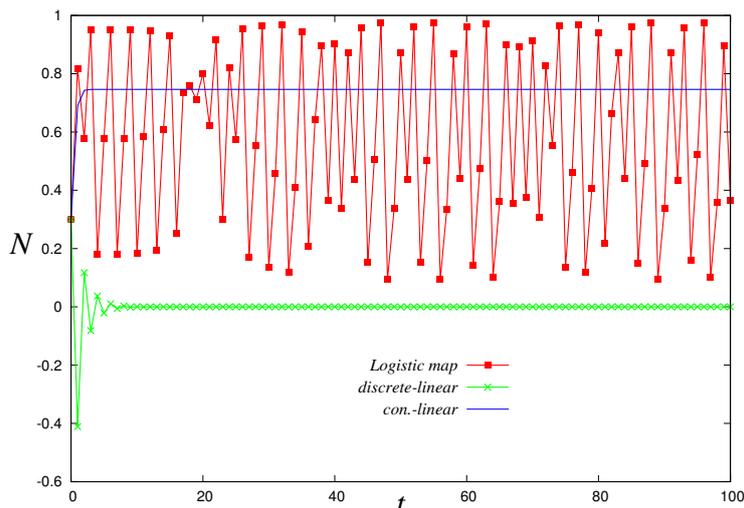


図 8: $\lambda = a - 1, \mu = a/(a - 1), a = 3.9$ と選んだ場合のロジスティック方程式の解析解 (*con.-linear*) と線形化されたロジスティック方程式 (*discrete-linear*), 非線形ロジスティック方程式の差分方程式 (*Logistic map*) の比較.

今回から数回に渡る講義では, 不規則な振る舞いをする非線形ロジスティック写像の差分方程式を扱い, その振る舞いを詳細に見て行くことにする.

4 非線形写像

ここまでで述べてきたことから, 非線形なロジスティック方程式を差分化した方程式:

$$N_t = (1 + \lambda)N_{t-1} - \mu\lambda N_{t-1}^2 \tag{46}$$

の振る舞いは, それ自体がとても興味深いので, 研究や学習の対象となる. そこで, ここからはこの場合 — ロジスティック方程式の非線形版の差分方程式 — に焦点を絞って話しを進めることにしよう. その際, (46) 式をそのまま扱うのではなく, $N_t = (a/\mu(1 - a))x_{t+1}$ なる変数変換を施すことで, 方程式を簡略化しておく. つまり, ここからは x_n についての更新式:

$$x_{n+1} = ax_n(1 - x_n) \tag{47}$$

の振る舞いについて調べていくことにしよう. この非線形な方程式を**ロジスティック写像**と呼ぶ⁸

4.1 ロジスティック写像から得られる数列のグラフを用いた求め方

方程式 (47) を初期値 $x = x_0$ から逐次求めることは計算機を用いれば難しくなく. しかし, $y = f(x) = ax(1 - x)$ と $y = x$ のグラフを描き, $(x, y) = (x_0, 0)$ からスタートし, (x_0, x_0) ($y = x$ と $x = x_0$ の交点), $(x_0, f(x_0))$ ($y = x_0$ と $y = f(x)$ の交点), $(f(x_0), f(x_0))$ ($y = x$ と $x = f(x_0)$ の交点), ... のように順次これらの点を辿って行くプロセスを繰り返す, その順番に x 軸の値を読みとったものが, (47) 式から生成される数列の値 x_0, x_1, x_2, \dots となっている事実を用いれば, 視覚的にこの数列を求めていくことができる. この図より, $a = 2.5, 2.9$ の場合には最終的に一つの点に収

⁸ 一般には $a = 4$ の場合の方程式をロジスティック写像と呼ぶが, この講義では $a > 1$ の任意の a での方程式をこのように呼ぶことにする.

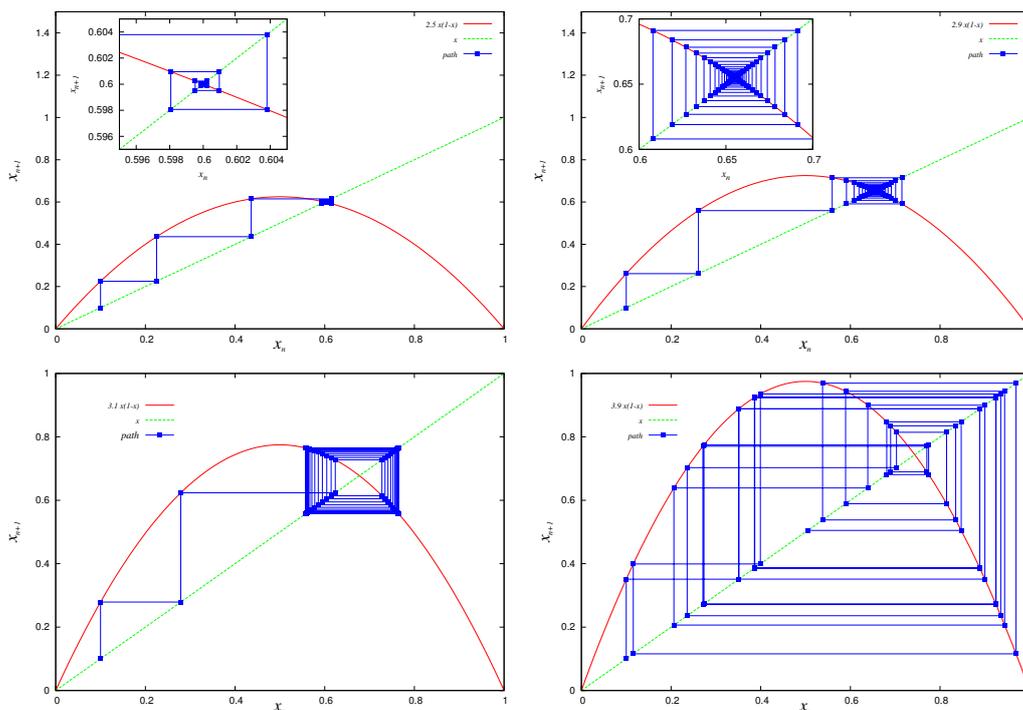


図 9: ロジスティック写像からの数列のグラフを用いた求め方. $y = f(x) = ax(1 - x)$ と $y = x$ のグラフを描き, $(x, y) = (x_0, 0)$ からスタートし, (x_0, x_0) ($y = x$ と $x = x_0$ の交点), $(x_0, f(x_0))$ ($y = x_0$ と $y = f(x)$ の交点), $(f(x_0), f(x_0))$ ($y = x$ と $x = f(x_0)$ の交点), ... のように順次このプロセスを繰り返し, 点を辿って行く. 左上から右下にかけて $a = 2.5, 2.9, 3.1, 3.9$ の場合. いずれの場合も初期値は $x_0 = 0.1$ に選んである.

束していく様子が見て取れる. また, $a = 3.1$ の場合には x の 2 つの値の間を交互に行き来きする周期運動を示すこともわかる. しかし, a の値を増やし, $a = 3.9$ にした場合, x の値の変化から明確な規則を読み取ることは難しい. この $a = 3.9$ の場合の写像の振る舞いは既に図 8 でも見たように非常に複雑である. よって, ここからはいくつかの観点からこの複雑な振る舞いについての詳細を見ていくことにする.

4.2 初期値依存性

例えば $a = 3.9$ の場合, 初期値を $x_0 = 0.1$ 選んだが, この際にみられる複雑な挙動が x_0 の値を $x_0 = 0.1$ から $x_0 = 0.10001$ に変えた場合にどうなるのかを調べてみよう. このような微小量の違いからくる振る舞いの差異を前節で説明したグラフによる作図から見るのは難しいので, 計算機を用いて写像 (漸化式) を逐次解くことでこれを数値的に調べてみる. 結果を図 10 に載せよう. この図より, $n = 7, 8$ ステップまで両者は完全に重なっているが, それ以降, これら両者の振る舞いには食い違いが生じる. 従って, この結果より, $a = 3.9$ の場合のロジスティック写像から生成される数列は初期値の選び方に対して非常に鋭敏であることがわかる. また, 図 9 で $a = 2.2, 2.9$ の場合のロジスティック写像に対しては, これらの初期値鋭敏性はみられない (各自, 調べてみること).

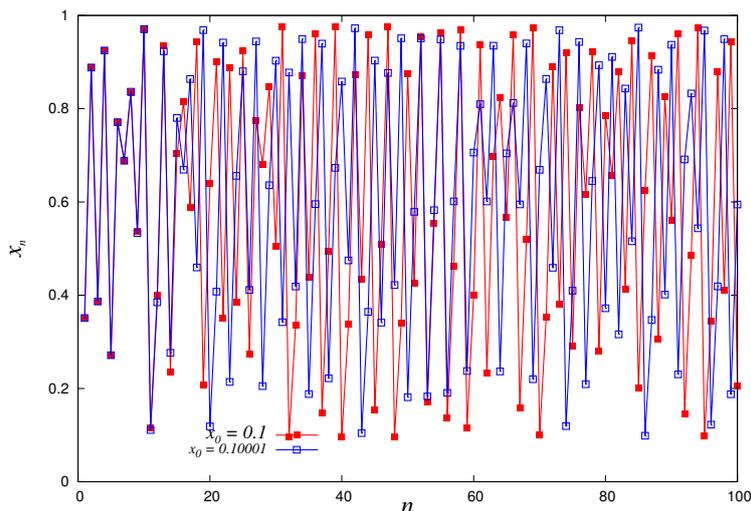


図 10: $x_{n+1} = 3.9x_n(1-x_n)$ において, 初期値を $x_0 = 0.1$ に選んだ場合と $x_0 = 0.10001$ に選んだ場合の数値の振る舞い.

4.3 数値計算上の丸め誤差/精度 (桁) 落ちによる影響

ここまでの考察で $a = 3.9$ の場合のロジスティック写像は初期値依存性を有することがわかった. このような非線形写像が初期値の選び方に鋭敏である場合には, 数学上の同値変形から得られる 2 つの数式であっても, それらは数値計算においては異なる結果を与える可能性があることに注意しなければならない.

例えば, 数学上は全く同じである次の 2 つの方程式を考えよう.

$$x_{n+1} = ax_n(1-x_n) \quad (48)$$

$$x_{n+1} = ax_n - ax_n^2 \quad (49)$$

2 年生のときに受講したであろう「計算機プログラミング I・同演習」では, 計算量の観点から上記 2 つの方程式の表現法に対する計算結果の差について学んだクラスもあったかもしれない. 例えば, 金曜クラス (A グループ) では

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/PROG2011/ProgI2011_5.pdf

のようなことを学習した.

すなわち, (48) 式において, 掛け算の回数は 1 回である一方, (49) 式においては 3 回であるから (足し算/引き算はそれぞれ 1 回ずつ), 可能な限りこの回数の少ない表現を用いてプログラムを書いた方が計算量の観点からは得であるというわけである.

しかし, 方程式が非線形の場合, これには注意が必要である. そこで, その問題点を具体的にみるために, $a = 3.9 = 39/10, x_0 = 0.5 = 1/2$ として, (48) から得られる数列のはじめの数項を「分数」を用いて正確に書き出してみると

$$x_1 = \frac{39}{40}, x_2 = \frac{1521}{16000}, x_3 = \frac{858879801}{2560000000}, \dots \quad (50)$$

であるが, 計算機上は分母を分子で割ることで得られる少数を小数点以下どこで打ち切るかによる誤差の影響が無視できなくなる. また, 上に示したはじめの数項から推測できるように, 分母分子

を正確に表現するための数字の桁数は繰り返しの増加とともに指数的に増加するので、計算機の変数の表現範囲によっては、その制限から来る精度落ちの効果も無視できなくなる⁹。

ここで注目すべきは、これらの影響が始める回数 n は表現 (49) 式を用いた場合と一般的に異なるという点である。実際、(49) 式に対してははじめの数項を書き出してみると

$$x_1 = \frac{39}{40}, x_2 = \frac{39}{10} \times \frac{39}{40} - \frac{39}{10} \times \frac{39}{40} \times \frac{39}{40}, \dots \quad (51)$$

となっていくので、例えば、上の x_2 の第 1 項において分子・分母の大きくなるスピードは第 2 項と比べて遅く、従って、打ち切りの効果が顕著に出始める繰り返し回数 n は (48) 式に基づく繰り返しと違う。よって、この打ち切り効果が異なり始めてその後の数列の振る舞いは、ちょうど、その時刻で初期条件を異なるものと置き換えて漸化式を作り直した状況として考えることができるので、前に見たように初期値の違う非線形写像に対する初期値鋭敏性がここでも効いてきて両者の差はどんどん大きくなっていくことになる。この「差」が生まれるのはどのくらい繰り返しを行った後なのか、は使用する計算機に依存する。参考までに担当者の所有するノート型パソコンで計算した結果を図 11 に載せる。この図からわかるように、初期条件が同じであっても、両者の間際立った食い

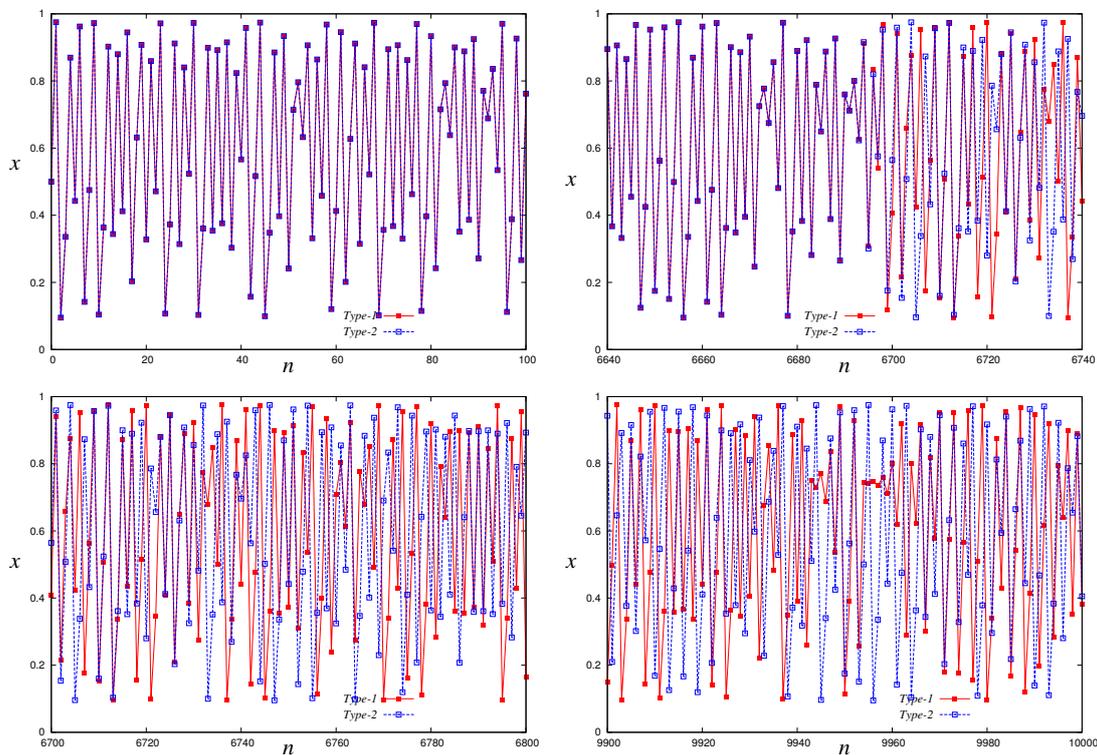


図 11: 式 (48)(これが Type-1), 式 (49)(これが Type-2) の振る舞い。初期条件は同じ $x_0 = 0.5$ 。だいたい $n \simeq 6700$ あたりで食い違いが始める。

違いはだいたい $n \simeq 6700$ あたりで出始める。

⁹ 例えば、32 ビット単精度 (C 言語では `int` で宣言された整数) で任意整数を表現する際、計算機内部では任意の十進整数 z を 2 進整数 $(b_{31}, b_{30}, \dots, b_0)$ として $z = \sum_{i=0}^{31} b_i 2^i$ と表現するので、これで表現しうる最大の整数は $(b_{31}, b_{30}, \dots, b_0) = (1, 1, \dots, 1)$ 、すなわち、 $z = 1 + 2 + 2^2 + 2^3 + \dots + 2^{31} = 2^{32} - 1 = 4294967295$ である。従って、この値を越える整数値が分子あるいは分母に現れる場合、その整数値は何らかの規則 (四捨五入、切り捨て等) でそれ以下の整数に変換されてしまう。

4.4 非線形写像の周期について

$a = 3.9$ あるいは、 $a = 4.0$ に選んだロジスティック写像は非常に複雑な不規則運動を見せるが、繰り返し回数を十分取ったときに得られる数列の中には、いくつかの周期解が含まれるかもしれない。そこで、ここからはその周期について詳しくみていくことにする。

一般に非線形写像: $x_{n+1} = f(x_n)$ に対し、ある x に写像を k 回繰り返し施したものを $f^k(x)$ で表すとすれば¹⁰、 $f^k(x) = x$ が成り立つとき、**写像 f は周期 k をもつ** と言う。例えば、 $f^6(x) = x$ ならば写像 f は周期 6 をもち、関係式

$$f^6(x) = f^3(f^3(x)) = f^3(x) = x \quad (52)$$

から、 f が 6 周期をもつならば、3 周期をもつことになる。従って、一般に p を n の約数とすると、 $f^n(x) = x$ を満たす x の中には $f^p(x) = x$ を満たす x も含まれている。これを具体的にみるために、 $f(x) = 4x(1-x)$ として、 $f^k(x) = x$ を満たす x の値を調べてみよう。これは前節までに見てきたロジスティック写像の $a = 4$ の場合に相当する。

多少面倒ではあるが、具体的に計算してみると

$$f(x) = 4x(1-x) \quad (53)$$

$$f^2(x) = 4\{4x(1-x)\}(1-\{4x(1-x)\}) = 16x - 80x^2 + 128x^3 - 64x^4 \quad (54)$$

$$f^3(x) = \dots \quad (55)$$

であるから、図 12 に示すように、非線形写像 $x_{n+1} = f(x_n) = 4x_n(1-x_n)$ の 1-4 周期解はそれぞれ $y = f, \dots, f^4$ と $y = x$ の交点の x 座標の値である。この図からわかるように、例えば、 $y = f^2$ と $y = x$ の交点は $y = f$ と $y = x$ の交点である 1 周期点 2 個を除いた 2 つの点からなり (2 は 1 を約数として持つ)。

ところで、 $f^2(x) = x$ を満たす 2 周期解 x を具体的に求めようとするのであれば、 x に関する代数方程式:

$$15x - 80x^2 + 128x^3 - 64x^4 = 0 \quad (56)$$

を解かなければならない。この中には $x = 0, 3/4$ の 1 周期解も含まれているから、この左辺は $x(x-3/4)$ で割り切るはずなので、その商である x に関する 2 次方程式を解いて残りの 2 つの解を求めればよい。しかし、一般的に n 周期解で n が大きくなると、この手の代数方程式の解を求めることは極めて厄介になる。ただし、 $a = 4$ のロジスティック写像: $x_{n+1} = f(x_n) = 4x_n(1-x_n)$ に限っては、三角関数の倍角公式を使うことで、周期解を首尾よく求めることができる。以下にそれを見ておこう。

4.5 $x_{n+1} = 4x_n(1-x_n)$ の周期解とその計算

三角関数の倍角公式を用いるために、 $x_{n+1} = 4x_n(1-x_n)$ において $x_n = \sin^2 \theta_n$ なる変数変換¹¹を行うと

$$\sin^2 \theta_{n+1} = 4 \sin^2 \theta_n (1 - \sin^2 \theta_n) = 4 \sin^2 \theta_n \cos^2 \theta_n = \sin^2 2\theta_n \quad (57)$$

¹⁰ 「 f の k 乗」ではないことに注意。

¹¹ この変数変換を用いた議論は次回以降も使うので覚えておくと良い。

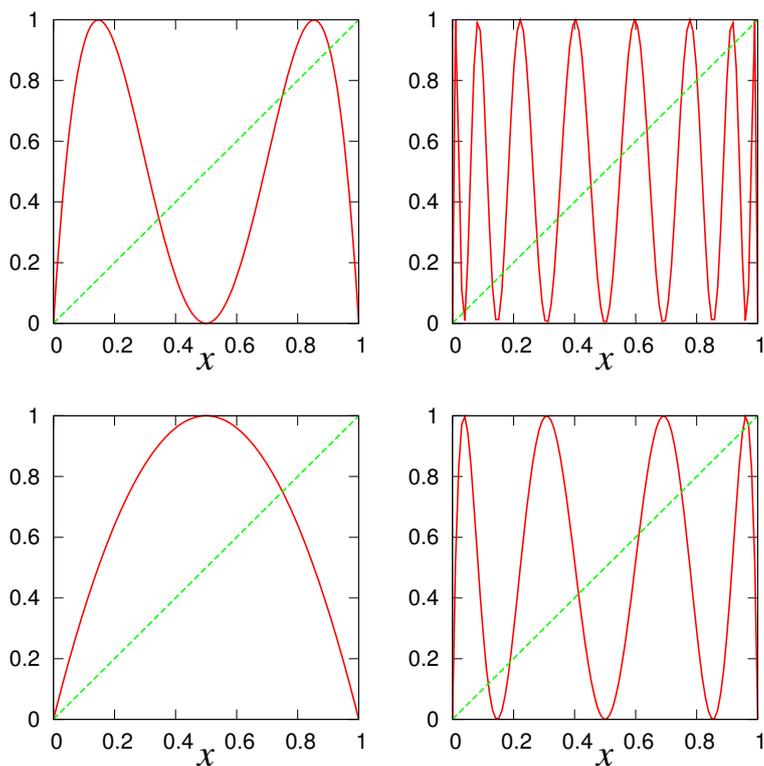


図 12: $y = f^1, \dots, f^4$ と $y = x$. $f(x) = 4x(1-x)$ である. 左上から右下へ f^2, f^4, f^1, f^3 .

であるから, 結局, $x_{n+1} = 4x_n(1-x_n)$ はこの変換によって

$$\sin^2 \theta_{n+1} = \sin^2 2\theta_n \tag{58}$$

と書き換えることができる. 従って, $\sin^2 \theta_{n+2}$ が求めたいのであれば $\sin^2 \theta_{n+2} = \sin^2 2\theta_{n+1}$ に対して上記を繰り返し使うことで

$$\sin^2 2\theta_{n+1} = 4 \sin^2 \theta_{n+1} \cos^2 \theta_{n+1} = 4 \sin^2 \theta_{n+1} (1 - \sin^2 \theta_{n+1}) = 4 \sin^2 2\theta_n \cos^2 2\theta_n = \sin^2 4\theta_n \tag{59}$$

であるから

$$\sin^2 \theta_{n+2} = \sin^2 4\theta_n \tag{60}$$

が得られる. 従って, 今欲しいのは p 周期解なので, これを θ_{n+p} に対して一般化すると

$$\sin^2 \theta_{n+p} = \sin^2 2^p \theta_n \tag{61}$$

となる¹². 従って, これを満たすためには, m を自然数として

$$\theta_{n+p} = \pm 2^p \theta_n + m\pi \tag{62}$$

¹² 必要であれば帰納法で証明せよ.

が成り立つべきである. 一方, p 周期である条件 ($x_n = \sin^2 \theta_n = x_{n+p} = \sin^2 \theta_{n+p}$) から, やはり l を自然数として

$$\theta_{n+p} = \pm \theta_n + l\pi \quad (63)$$

が成り立つ. (62)(63) 式より, $L \equiv l - m$ として

$$\theta_n = \frac{L\pi}{2^p \pm 1} \quad (64)$$

が得られる. 自然数 L には任意性があるが, ここでは簡単のため $L = 1$ と置くことにする.

まずは, 1 周期解 ($x = 0, 3/4$) が再現されるか確認してみると, $p = 1$ とおいて, $\theta_n = \pi, \pi/3$ であるから, $x_{n+p} = \sin^2 \theta_{n+p} = \sin^2 2^p \theta_n$ に注意すれば, それぞれから $x_n = \sin^2 \pi = 0, x_n = \sin^2 \pi/3 = (\sqrt{3}/2)^2 = 3/4$ となり, OK である. また, 2 周期解については, $p = 2$ として $\theta_n = \pi/3, \pi/5$ が得られるが, $\theta_n = \pi/3$ のときには, $x_n = \sin^2 \pi/3 = (\sqrt{3}/2)^2 = 3/4$ であり, これは 2 周期の中にその約数である 1 周期が含まれていることを表している. $\theta_n = \pi/5$ のときには $x_n = \sin^2 \pi/5 = 0.345$, また, (61) 式から $x_{n+p} = \sin^2 \theta_{n+p} = \sin^2 2^p \theta_n$ であるから, $x_{n+1} = \sin^2 2\pi/5 = 0.904, x_{n+3} = \sin^2 4\pi/5 = \sin^2(\pi - \pi/5) = x_n$. よって, 2 周期解は 0.345 と 0.904 であり, これは図 12 からも見えてとれる.

ここで見た周期解は安定であるわけではなく, 不安定化する. その安定性条件に関する詳細は次回以降に見ていくことにする.

レポート課題 2

複雑な振る舞いを見せるのは何もロジスティック写像だけではない. 例えば次のような**テント写像**:

$$x_{n+1} = 1 - |1 - 2x_n| \quad (65)$$

もロジスティック写像同様に複雑な挙動を示す. この写像のはじめの 100 ステップの値を初期値 $x_0 = 0.1$ に対して計算し, 横軸に繰り返し回数 n , 縦軸に x_n の値をプロットせよ¹³. この結果は次回の講義内容に関係するので, 各自が必ず計算, グラフ化し, 結果を目で見てみることにする.

連絡: 次週 4/26 は休講とします.

¹³ どうもうまく行かない場合には $2 = 1.9999$ として, $x_{n+1} = 1 - |1 - 1.9999x_n|$ を用いよ.

課題 2 の解答例

次のようなテント写像¹⁴ :

$$x_{n+1} = 1 - |1 - 2x_n| = \begin{cases} 2x_n & (0 \leq x_n \leq 1/2) \\ 2 - 2x_n & (1/2 < x_n \leq 1) \end{cases} \quad (66)$$

はロジスティック写像同様に複雑な挙動を示す. 簡単なプログラムなので, 「計算機プログラミング I・同演習」の復習を兼ねて下記にプログラム例を書く

```
#include <stdio.h>
#include <math.h>
#define x0 0.10 /* 写像の初期値 */
#define N 100
main()
{
    FILE *pt; /* ファイルポインタ */
    double x;
    int i;
    if((pt = fopen("tent_map.dat","wt")) !=NULL){ /* ファイルオープン*/
        for(i = 0, x=x0; i <= N; i++){
            x = 1.0-fabs(1.0-1.99999*x); /* テント写像 */
            fprintf(pt,"%d %lf\n",i,x); /* ファイル書き込み */
        }
    }
    fclose(pt); /* ファイルクローズ*/
}
```

となる. ここで得られる結果を横軸に繰り返し回数 n , 縦軸にその n での数列の値をプロットすると図 13 のようになる.

※ この課題に関する Remark

$T(x) = 1 - |1 - 2x|$ に対し, $x_0 = 0.1 = 1/10$ として, $x_{n+1} = T(x_n)$ より具体的にはじめの数項を書き出してみると

$$\begin{aligned} x_1 &= 1 - \left| 1 - \frac{2}{10} \right| = 1 - \left| \frac{4}{5} \right| = \frac{1}{5} \\ x_2 &= 1 - \left| 1 - \frac{2}{5} \right| = 1 - \left| \frac{3}{5} \right| = \frac{2}{5} \\ x_3 &= 1 - \left| 1 - \frac{4}{5} \right| = 1 - \left| \frac{1}{5} \right| = \frac{4}{5} \\ x_4 &= 1 - \left| 1 - \frac{8}{5} \right| = 1 - \left| -\frac{3}{5} \right| = \frac{2}{5} \\ x_5 &= 1 - \left| 1 - \frac{4}{5} \right| = 1 - \left| \frac{4}{5} \right| = \frac{4}{5} \\ \dots &= \dots \end{aligned}$$

¹⁴ なぜ「テント」写像と呼ばれるのか, は $y = 1 - |1 - 2x|$ を $x \in [0, 1]$ の関数として図示してみるとわかる.

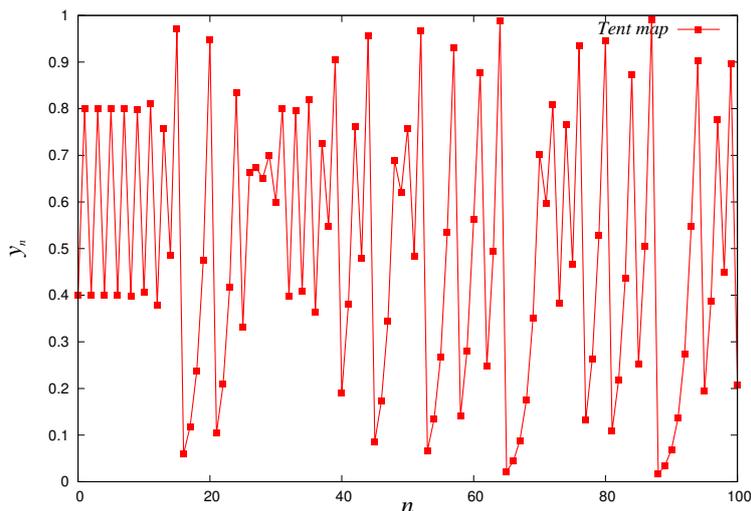


図 13: テント写像から生成される数列 x_n の様子.

となり, 以降は $2/5 = 0.4, 4/5 = 0.8$ を繰り返す 2 周期を示す.

また, $x_0 = 0.3 = 3/10$ とすると, 同様に, $x_1 = 3/5, x_2 = 4/5, x_3 = 2/5, x_4 = 4/5, x_5 = 2/5, \dots$ のように, やはり $2/5, 4/5$ を繰り返す 2 周期を示す. さらに詳しくみていくと, 初期値 x_0 の値が $0.\gamma_1$ ($\gamma_1 \neq 0, 5$) の形をした全ての小数点以下 1 桁目まで残した数に対し, $2/5, 4/5$ を繰り返す 2 周期を示すことがわかり, $x_0 = 0.\gamma_1\gamma_2$ ($\gamma_2 \neq 0, 5$) に対しては 10 周期を与え, 一般に $x_0 = 0.\gamma_1\gamma_2 \dots \gamma_N$ ($\gamma_N \neq 0, 5$) に対しては $2 \cdot 5^{N-1}$ 周期を持つことがわかる. 図 14 に $x_0 =$

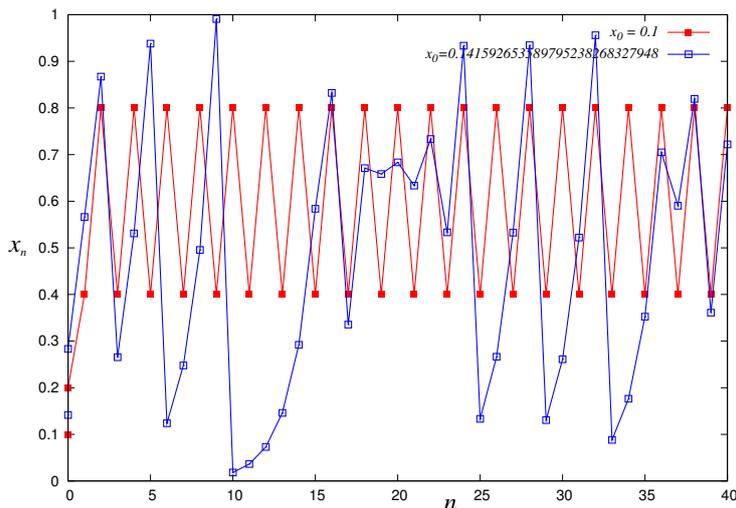


図 14: $x_0 = 0.1, x_0 = 0.141592653589795238268327948$ から開始したテント写像からの数列の様子.

$0.1, x_0 = 0.141592653589795238268327948$ を初期値に選んだ場合のテント写像からの数列を計算機によって生成した結果を示す. 上で述べたように, $x_0 = 0.1$ の場合には 2 周期が得られ, $N = 27$ である $x_0 = 0.141592653589795238268327948$ に選んだ場合には $2 \cdot 5^{26} = 2980232238769531250$

周期が得られる。これは実質的に観測することが難しいほど長い周期である。

また, $x_0 = 0.1$ の場合, 計算機内で 0.799999 と 0.8 を区別できないことにより, それ以降の数列がこの微妙な差により変化して行き, 最終的に 0.5 に到達した時点で, x_n の値は 0 に収束してしまう。実際に, データファイルを見てみると

```
35 0.400000
36 0.799999
37 0.400002
38 0.800003
.....
.....
52 0.750000
53 0.500000
54 1.000000
55 0.000000
```

このような症状は $x_0 = 0.141592653589795238268327948$ の場合も同様である。従って, 数値計算の際にはテント写像の式で $2 = 1.999$ とするなどの工夫が必要となる。

5 軌道の稠密性

前回学んだ $a = 4.0$, あるいは $a = 3.9$ の場合のロジスティック写像: $x_{n+1} = ax_n(1 - x_n)$, 上記に見たテント写像から得られる数列: x_0, x_1, x_2, \dots , は一見すると全くランダムな点列のように見える。この数列は 0 から 1 の間の実数値をとるので, これを $[0, 1]$ の一様乱数と比較してみることは興味深い。そこで, x_n が取り得る範囲 $[0, 1]$ を 1000 等分し, 写像の各ステップで x_n の値がこの 1000 等分された「ビン」の中にどれくらい入るかを数えてみることにする。つまり, ここでのビンのサイズ (幅) は 0.001 である。もし, 十分な繰り返し回数 (例えば $n = 100000$ 程度) とり, 1000 個のビンの中にほぼ均等に x_n の値が入れば, この写像は一様乱数に近いことが言える。これをプログラミングするためには, 先に例示したプログラムのメイン関数を次のように書き換えればよい。

```
for(k=0; k<=999;k++){
    count[k]=0; /* ビンの初期化 */
}
for(i = 0, x=x0; i <= N; i++){
    x = 1.0-fabs(1.0-1.99999*x);
    for(k=0; k<=999; k++){
        if((x>=0.001*k) && (x<0.001*(k+1))){
            count[k]++;
        }else{
            count[k]=count[k];
        }
    }
}
for(k=0; k<=999; k++){
```

```

    fprintf(pt, "%lf %lf\n", 0.001*k, (double)count [k]*1000/N);
}
}

```

そこで, まずは上記のテント写像についての結果を図 15 に載せる. この結果を見ると, テント写像

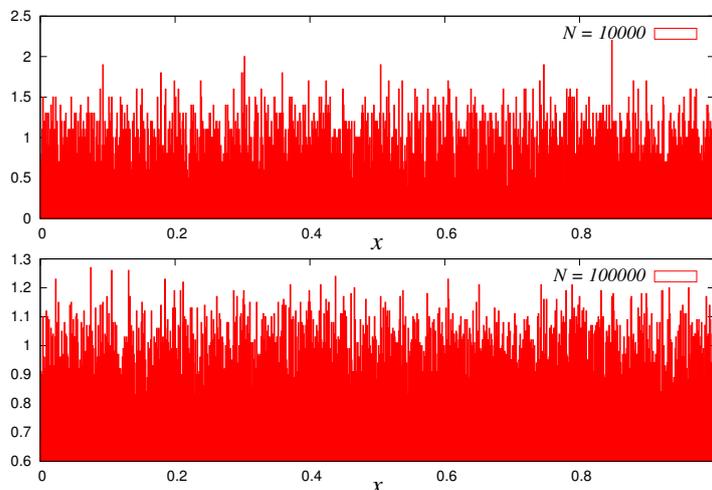


図 15: テント写像の各ビンに落ちる x_n の割合.

から生成される数列は一様分布すると考えられる. 以上は計算機に基づく数値計算からの実験結果であるが, これをもう少し手計算により別な角度から直接的に理解したい. そこで, ここではベルヌーイ写像と呼ばれる第三の写像を導入し, それについてみていくことにする.

5.1 ベルヌーイ写像

次のような写像 $B(x)$ を考える.

$$x_{n+1} = B(x_n) = \begin{cases} 2x_n & (0 \leq x_n < 1/2) \\ 2x_n - 1 & (1/2 \leq x_n \leq 1) \end{cases} \quad (67)$$

この写像は既に学んだテント写像との間に関係を持ち, $0 \leq x < 1/2$ で $B(x) = T(x)$ となり, $1/2 \leq x \leq 1$ で $B(x) = 1 - T(x)$ を満たす.

5.1.1 2進小数の復習

ここではベルヌーイ写像に関する性質を調べる前に, 簡単に 2 進数の復習をしておこう. ここでは $0 \leq x \leq 1$ の実数のみを扱っているので, 2 進小数のみを考える. このとき, 10 進小数 x が

$$x = \frac{\alpha_1}{2} + \frac{\alpha_2}{2^2} + \frac{\alpha_3}{2^3} + \cdots + \frac{\alpha_n}{2^n} + \cdots \quad (68)$$

のように表現できたとすると, この x の 2 進小数は

$$x = (0.\alpha_1\alpha_2\cdots\alpha_n\cdots)_2 \quad (69)$$

となる¹⁵. 従って, 例えば, $x = 0.5$ であれば

$$0.5 = \frac{1}{2} + \frac{0}{2^2} + \frac{0}{2^3} + \dots + \frac{0}{2^n} + \dots \quad (70)$$

であるから, その 2 進小数は $x = (0.1)_2$ であり, $x = 1$ ならば

$$1 = \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^n} + \dots = \frac{\frac{1}{2}}{1 - \frac{1}{2}} = 1 \quad (71)$$

であるから, その 2 進数表示は $x = (0.111\dots)_2$ となる.

それでは, 10 進少数 x の値からは簡単に (68) 式の形が推測できない場合はどうするのであろうか. これを見るために, $x = 0.2009$ を考える. この値の 2 進小数を得るための「レシピ」は, まず, この値を 2 倍し, 整数部 1 桁目 (α_0) に「0」が来るか, 「1」が来るかを調べ, 「0」が来れば「0.」, 「1」が来れば「1.」から開始し, 逐次 2 をかけて, 整数部 1 桁目に「0」が来るか, 「1」が来るかを調べていき, 「0」が来れば「 $\alpha_0.0$ 」, 「1」が来れば「 $\alpha_0.1$ 」のように繰り返していく. その過程で整数部に 1 が現れた場合には, 次のステップで 2 をかける直前に整数部の 1 を消すために 1 を引いておく.

具体的に $x = 0.2009$ の場合には $0.2009 \times 2 = 0.4018$ なので, まず $x = (0.0)_2$ であり, $0.4018 \times 2 = 0.8036$ なので $x = (0.00)_2$ である. 次いで, $0.8036 \times 2 = 1.6072$ なので $x = (0.001)_2$ であり, 整数部に 1 がきたので, 次に 2 をかける前に 1 を引いておき, $1.6072 - 1 = 0.6072$ に 2 をかける. すると, $0.6072 \times 2 = 1.2144$ なので $x = (0.0011)_2$, 次は $0.2144 \times 2 = 0.4288$ なので $x = (0.00110)_2 \dots$ のように逐次的に 2 進小数が求まっていくことになる. 筆算でやると面倒極まりないが, 電卓などを使うか, 簡単な計算プログラムを作成することで, 例えば

$$\frac{1}{\sqrt{2}} = 0.707106811865472440084436210485 = (0.10110101000001001111001010011100\dots)_2 \quad (72)$$

なども手軽に得られる.

5.1.2 2 進小数のビットシフト演算とベルヌーイ写像

前節の (68) 式より, $x < 1/2$ であれば, $\alpha_1 = 0$ であり, このとき x は

$$x = \frac{\alpha_2}{2^2} + \frac{\alpha_3}{2^3} + \dots + \frac{\alpha_n}{2^n} + \dots = (0.0\alpha_2\alpha_3\dots\alpha_n\dots)_2 \quad (73)$$

と書ける. 従って, この x を $x < 1/2$ の場合のベルヌーイ写像 $B(x) = 2x$ に作用させると

$$\begin{aligned} B((0.0\alpha_2\alpha_3\dots\alpha_n\dots)_2) &= B\left(\frac{\alpha_2}{2^2} + \frac{\alpha_3}{2^3} + \dots + \frac{\alpha_n}{2^n} + \dots\right) \\ &= 2\left(\frac{\alpha_2}{2^2} + \frac{\alpha_3}{2^3} + \dots + \frac{\alpha_n}{2^n} + \dots\right) \\ &= \frac{\alpha_2}{2} + \frac{\alpha_3}{2^2} + \dots + \frac{\alpha_n}{2^{n-1}} + \frac{\alpha_{n+1}}{2^n} + \dots = (0.\alpha_2\alpha_3\dots\alpha_n)_2 \quad (74) \end{aligned}$$

となる. ここで得られた $x = (0.\alpha_2\alpha_3\dots\alpha_n)_2$ に対し, さらに $B(x)$ を作用させて $B^2(x)$ を計算してみると

$$\begin{aligned} B^2((0.0\alpha_2\alpha_3\dots\alpha_n\dots)_2) &= B((0.\alpha_2\alpha_3\dots\alpha_n\dots)_2) \\ &= \left(\frac{\alpha_2}{2} + \frac{\alpha_3}{2^2} + \dots + \frac{\alpha_n}{2^{n-1}} + \frac{\alpha_{n+1}}{2^n} + \dots\right) \\ &= \frac{\alpha_2}{2^0} + \frac{\alpha_3}{2} + \dots + \frac{\alpha_n}{2^{n-2}} + \frac{\alpha_{n+1}}{2^{n-1}} + \dots = (\alpha_2.\alpha_3\alpha_4\dots\alpha_n\dots)_2 \quad (75) \end{aligned}$$

¹⁵ $(\dots)_2$ の添え字「2」はカッコ内の小数が 2 進小数であることを明示するためのものである. そうすると 10 進小数は $(\dots)_{10}$ ということになるが, この場合添え字「10」は省略する.

となる. しかし, 最後の式で $\alpha_2 = 1 \neq 0$ であれば, $x > 1$ となってしまうので, $\alpha_2 = 0$ でなければいけないから, 結局, $B((0.\alpha_2\alpha_3 \cdots \alpha_n \cdots)_2) = (0.\alpha_3\alpha_4 \cdots \alpha_n \cdots)_2$ であることがわかる. この結果より, **ベルヌーイ写像は 2 進小数の小数点以下のビット (0,1 の並び) を 1 ずつ左へずらす作用を持つことがわかる.**

一方, $1/2 \leq x \leq 1$ であれば, ベルヌーイ写像としては $B(x) = 2x - 1$ で与えられたので, x として $x = (0.\alpha_1\alpha_2 \cdots \alpha_n \cdots)_2$ ($\alpha_1 = 1 \neq 0$) を作用してみて, 上記の「ビットずらし」の効果があるのかを確かめなければならない. 実際に計算してみると

$$\begin{aligned} B((0.\alpha_1\alpha_2 \cdots \alpha_n \cdots)_2) &= B\left(\frac{\alpha_1}{2} + \frac{\alpha_2}{2^2} + \cdots + \frac{\alpha_n}{2^n} + \cdots\right) \\ &= 2\left(\frac{\alpha_1}{2} + \frac{\alpha_2}{2^2} + \cdots + \frac{\alpha_n}{2^n} + \cdots\right) - 1 \\ &= \alpha_1 + \frac{\alpha_2}{2} + \frac{\alpha_3}{2^2} + \cdots + \frac{\alpha_n}{2^{n-1}} + \cdots - 1 \\ &= (\alpha_1 - 1) + \frac{\alpha_2}{2} + \frac{\alpha_3}{2^2} + \cdots + \frac{\alpha_n}{2^{n-1}} + \cdots = (0.\alpha_2\alpha_3 \cdots \alpha_n \cdots)_2 \end{aligned}$$

となる. ここで, $\alpha_1 = 1$ という事実をつかった. 従って, やはり, 2 進小数の小数点以下のビットを左へずらす作用があることがわかる.

5.2 稠密性再考

ここまで学んだ知識を用いて先に言及した「稠密性」を再度考えてみよう. このとき, 写像 $x_{n+1} = f(x_n)$ から得られる軌道 $x_0, x_1, \dots, x_n, \dots$ が「稠密」であることをもう少し正確に言うと

ここで考える稠密性

区間 $[0, 1]$ 上のどんな無限小区間 $[\epsilon_1, \epsilon_2]$ の中にも軌道 $x_0, x_1, \dots, x_n, \dots$ の点が必ず入る.

となる.

これを具体的にみるために, $[0, 1]$ を 4 等分し, $I_1 = [0, 1/4], I_2 = [1/4, 1/2], I_3 = [1/2, 3/4], I_4 = [3/4, 1]$ とすると, $1/4 = (0.01)_2, 1/2 = (0.10)_2, 3/4 = (0.11)_2$ であるから, この区間はそれぞれ 2 進小数の小数点以下 2 桁でもって $I_1 \rightarrow (0.00)_2, I_2 \rightarrow (0.01)_2, I_3 \rightarrow (0.10)_2, I_4 \rightarrow (0.11)_2$ と特徴付けることができる. このとき, ベルヌーイ写像から得られる軌道が上記 4 区間のどれに入るのかを調べてみたい. そのために, ベルヌーイ写像の初期値を $x_0 = 1/\sqrt{2}$ とする. この値は先にみたように, 2 進数小数表示のもとでは

$$x_0 = \frac{1}{\sqrt{2}} = 0.707106811865472440084436210485 = (0.10110101000001001111001010011100\dots)_2 \quad (77)$$

であったから, これを「1 ビットずらし」の効果を持つベルヌーイ写像に作用させると, そのたびに小数点以下 1 桁ずつが左にずれていく. そのときの小数点以下 2 桁目までが, 上記, I_1, I_2, I_3, I_4 のどれに該当するのかを調べていけばよい. 具体的にやってみると

$$\begin{aligned} x_0 &= (0.\underline{1}0110101000001001111001010011100\dots)_2 \rightarrow I_3 \text{ に入る} \\ B(x_0) &= (0.0\underline{1}0101000001001111001010011100\dots)_2 \rightarrow I_2 \text{ に入る} \\ B^2(x_0) &= (0.0\underline{1}0101000001001111001010011100\dots)_2 \rightarrow I_4 \text{ に入る} \\ B^3(x_0) &= (0.1\underline{0}101000001001111001010011100\dots)_2 \rightarrow I_3 \text{ に入る} \end{aligned}$$

$$\begin{array}{c} \dots \dots \dots \\ B^8(x_0) = (0.\underline{000001}001111001010011100\dots)_2 \rightarrow I_1 \text{ に入る} \\ \dots \dots \dots \end{array}$$

となる。これより、4つの区間のいずれにも軌道が入り、かつ、この場合、それらはほぼ同数(この例の場合、それぞれの個数は $|I_1| = 7, |I_2| = 9, |I_3| = 10, |I_4| = 6$) となっているから4区間に一様に分布することがわかる。もちろん、区間 $[0, 1]$ をより細かく等分割し、小数点以下数桁の判定で軌道がどこに入るのかを調べることもできるが、その場合でも、その各々の区間にはほぼ同数の軌道上の点が入ることを示すことができる(今週のレポート課題)。

5.3 テント写像とベルヌーイ写像の関係

既に述べたように、テント写像 $T(x)$ とベルヌーイ写像 $B(x)$ の間には $T(x) = 1 - B(x)$ の関係がある。従って、例えば、 $1/2 \leq x \leq 1$ において $x = (0.\alpha_1\alpha_2\alpha_3 \dots \alpha_n \dots)_2$ を $T(x)$ に作用してみると、1の2進小数表示が $(0.111\dots)_2$ であったことを思い出して

$$\begin{aligned} T((0.\alpha_1\alpha_2\alpha_3 \dots \alpha_n \dots)_2) &= 1 - B((0.\alpha_1\alpha_2\alpha_3 \dots \alpha_n \dots)_2) \\ &= (0.111\dots)_2 - (0.\alpha_2\alpha_3 \dots \alpha_n \dots)_2 \\ &= (0.(1 - \alpha_2)(1 - \alpha_3) \dots (1 - \alpha_n) \dots)_2 \equiv (0.\beta_2\beta_3 \dots \beta_n \dots)_2 \quad (78) \end{aligned}$$

が得られる。ここに、 $\beta_k \equiv 1 - \alpha_k$ ($k = 2, 3, \dots$) であり、 α_k が「1」ならば β_k は「0」、 α_k が「0」ならば β_k は「1」となる。(ちなみに、 $x < 1/2$ の場合には、 $x = (0.0\alpha_2\alpha_3 \dots \alpha_n \dots)_2$ であり、 $T(x) = B(x)$ だったので、 $T((0.0\alpha_2\alpha_3 \dots \alpha_n \dots)_2) = (0.\alpha_2\alpha_3 \dots \alpha_n \dots)_2$ である。) 従って、 $1/2 \leq x \leq 1$ の場合には**テント写像 $T(x)$ は2進小数の小数点以下のビットを左に一つずつずらし、かつ、得られる小数点以下のビット列を反転させる**ことがわかる。この事実と前節での稠密性の定義を思い出すと、ベルヌーイ写像が $[0, 1]$ 上の任意の小区間に必ずその軌道が入り、かつ、 $[0, 1]$ 上に一様分布するのであれば、テント写像も同様の性質を持つことが直ちに言えることになる。なぜならば、例えば、前節でみた4つの各区間の定義において、小数点以下で「0」と「1」を交換し、 $I_1 \rightarrow (0.11)_2, I_2 \rightarrow (0.10)_2, I_3 \rightarrow (0.01)_2, I_4 \rightarrow (0.00)_2$ としても結果は変わらないことは明らかだからである。

以上の議論より、テント写像もまた稠密性を満たすことがわかった。

5.4 テント写像とロジスティック写像の関係

それでは前回学んだロジスティック写像: $x_{n+1} = 4x_n(1 - x_n)$ の稠密性はどうか? ここでもまずは感じをつかむために、計算機を使った数値実験をしてみると図16のようになる。この図を見ると、繰り返し回数の増加にともなって、「分布」の形は滑らかになっていくが、一様分布とみなすためには、 $x_n = 0, 1$ 近くの両端に x_n の値が来る頻度が高くなっているように見える。

この場合をテント写像に対する前節での結果を用いて理解するために、ロジスティック写像 $x_{n+1} = L_4(x_n)$ において¹⁶ 前回行った変数変換: $x_n = \sin^2 \theta_n$ を行うと、 $\sin^2 \theta_{n+1} = \sin^2 2\theta_n$ であるから、 $a = 4$ のロジスティック写像は

$$\theta_{n+1} = \begin{cases} 2\theta_n & (0 \leq \theta_n < \pi/4) \\ \pi - 2\theta_n & (\pi/4 \leq \theta_n \leq \pi/2) \end{cases} \quad (79)$$

¹⁶ ここから簡単のため、 $L_a(x) = ax(1 - x)$ と定義する。

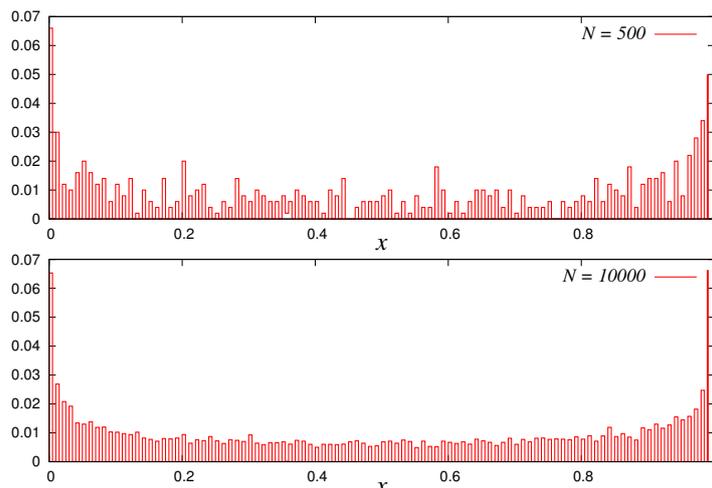


図 16: ロジスティック写像: $x_{n+1} = 4x_n(1 - x_n)$ の各ビンに落ちる x_n の割合. $[0, 1]$ を 100 等分したビンを考えている.

と書き直すことができる. 従って, 例えば, $y_n = (2/\pi)\theta_n$ で y_n を定義すれば, y_n はテント写像となる (各自が確認すること). 従って, テント写像が稠密であれば, ロジスティック写像 ($a = 4$) も稠密であることがわかる. 以上の議論より, $B(x) \rightarrow T(x) \rightarrow L_4(x)$ の順に稠密性を示すことができることを確認することができた. しかし, 図より, ロジスティック写像 $L_4(x)$ の軌道がある微小区間 x と $x = x + \delta x$ に入る頻度 — **密度関数** — はベルヌーイ写像 $B(x)$, テント写像 $T(x)$ のそれと比べるとかなり異なる形をしている. この $L_4(x)$ の密度関数はどのような関数形で与えられるのかを次節で見よう.

5.5 軌道の密度関数

前節までの結果からみてテント写像, ベルヌーイ写像は $[0, 1]$ の間に一様に分布するわけであるから, 軌道の密度関数 $P(x)$ は $P(x) = 1$, つまり

$$\int_0^1 P(x)dx = 1 \tag{80}$$

を満たす. また, 写像からの軌道: $x_1, x_2, \dots, x_n, \dots$ が与えられたとき, その密度関数は一般に

$$P(x) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \delta(x - x_i) \tag{81}$$

で与えられる¹⁷. さて, テント写像とベルヌーイ写像の場合の密度関数はともに $P(x) = 1$ であったが, ロジスティック写像: $L_4(x)$ の場合のその形状は自明ではない. そこで, 前節での結果を使ってこれを求めよう.

¹⁷ これら密度関数 $P(x)$ を「不変測度」と呼んだりもする. ここで, (81) が規格化条件 (80) を満たすことは次のように簡単に示すことができる.

$$\int_0^1 P(x)dx = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \int_0^1 \delta(x - x_i)dx = \lim_{n \rightarrow \infty} \frac{1}{n} \times n = 1$$

$a = 4$ のロジスティック写像において, $x_n = \sin^2 \theta_n$ と変数変換し, $\theta_n = (\pi/2)y_n$ とすると, y_n はテント写像になったので, その密度関数は $P(y) = 1$ であることに注意しよう. そこで, $x_n = \sin^2 \theta_n$ の両辺を θ_n で微分すると

$$\frac{dx_n}{d\theta_n} = 2 \sin \theta_n \cos \theta_n = 2\sqrt{x_n(1-x_n)} \tag{82}$$

となる. 一方, $\theta_n = (\pi/2)y_n$ の両辺を y_n で微分すると, $d\theta_n/dy_n = (\pi/2)$ であるから, これらより

$$dx_n = 2\sqrt{x_n(1-x_n)} d\theta_n = 2\sqrt{x_n(1-x_n)} \frac{\pi}{2} dy_n \tag{83}$$

すなわち

$$dy_n = \frac{dx_n}{\pi\sqrt{x_n(1-x_n)}} \tag{84}$$

が得られる. 従って, 上式を用いることで積分 $\int_0^1 P(y)dy$ を x についての積分に変換することができて

$$\int_0^1 P(y)dy = \int_0^1 1dy = 1 = \int_0^1 \frac{dx}{\pi\sqrt{x(1-x)}} = \int_0^1 P(x)dx \tag{85}$$

となるから密度関数は

$$P(x) = \frac{1}{\pi\sqrt{x(1-x)}} \tag{86}$$

であることがわかる. これは図 17 より, 計算機実験による結果とも良く一致する. ところで, ロジ

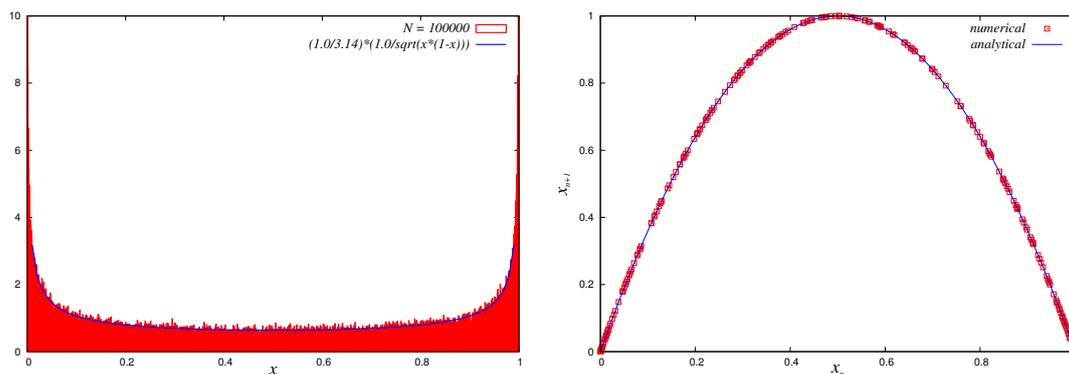


図 17: $x_{n+1} = 4x_n(1-x_n)$ の各ビンに落ちる x_n の頻度から求めた x_n の確率密度関数 (左). 実線は (86) 式. 右図は x_n, x_{n+1} の関係. この図から「稠密」であれば, 曲線: $y = 4x(1-x)$ の上を点が隙間無く埋める (図では説明のため, 繰り返し回数 500).

スティック写像において密度関数を解析的に求めることのできるのは $a = 4$ の場合のみである. このような解析的な密度関数は, 例えば関数 $g(x)$ の軌道にわたる平均値を密度関数 $P(x)$ の期待値として

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n g(x_i) = \int_0^1 g(x)P(x)dx \tag{87}$$

が計算できるので有益である¹⁸. しかし, $a \neq 4$ の場合には密度関数の定義式 (81) に戻って計算機による数値計算を用いて算出することになる. 例として図 18 にいくつかの $a (\neq 4)$ の場合を載せる. これらの図からわかるように, 一般に $P(x)$ の形状は非常に複雑となる.

以上で我々はいくつかの写像から得られる複雑な軌道 (数列) を稠密性という観点からとらえることができることを学んだ. しかし, この「稠密性」は軌道が複雑であるために必要な条件ではあるが, それで十分ではないことを忘れてはいけない. 例えば, 先の 4 つの区間に入るかどうか, また, 一様に分布するのかを確認した実験において, 軌道が規則的に $I_1 \rightarrow I_2 \rightarrow I_3 \rightarrow I_4 \rightarrow I_1 \rightarrow \dots$ のように周期 4 であったとしても, 全ての区間に入り, かつ, 一様に分布する. 従って, 稠密性とは別の尺度で軌道の複雑さを測る必要があるであろう. そこで, 今回はそのような尺度である「軌道のエントロピー」「リアプノフ指数」とその計算方法について学び, ある特殊な場合に対し, これらの指標を解析的に求めることができることをみていく. これらを学習することで, 我々がここまで「複雑な軌道」と呼んでいたものが, これらの指標から「カオス」と呼ばれる運動であることが明らかとなる.

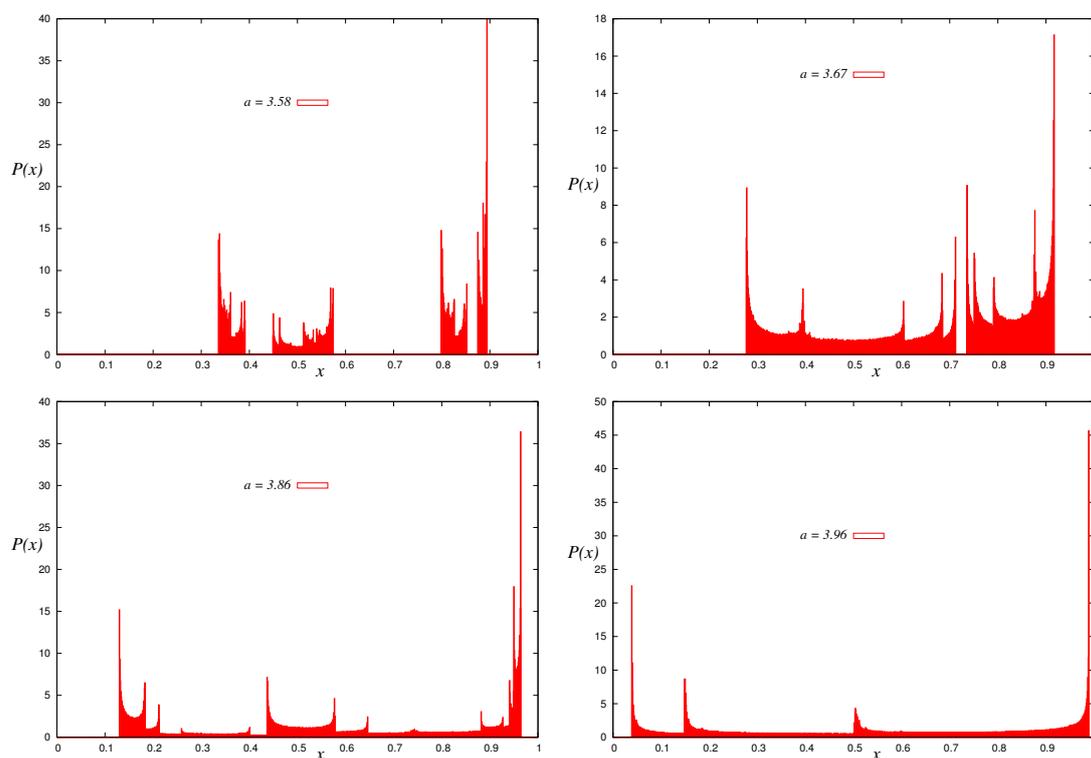


図 18: ロジスティック写像: $x_{n+1} = L_a(x) = ax(1-x)$ の密度関数. 定義 (81) を用いて数値計算した. 繰り返し回数を 1000000, 区間 $[0, 1]$ を千等分したビンを用意した.

レポート課題 3

¹⁸ 実際, 次回の講義では「軌道のエントロピー」「リアプノフ指数」と呼ばれる物理量がこの方法によって解析的に計算できることを学ぶ.

$[0, 1]$ を 8 区間に等分割し, $1/\sqrt{3}$ を初期値とするベルヌーイ写像の軌道が各々の区間にいくつ入るかを調べよ. ただし, $1/\sqrt{3}$ の 2 進小数表示は小数点以下 80 桁程度までとること.

※ 計算機を用いて算出したものは, 計算プログラム・コードを添付すること.

課題 3 の解答例

まず前回の復習として $[0, 1]$ を 8 区間に等分割し, $1/\sqrt{3}$ を初期値とするベルヌーイ写像の軌道が各々の区間にいくつ入るかを調べてみる. 手で逐次計算していくことは非常に骨が折れるので, 計算機で実行してみる. そのサンプル・プログラムを下記に載せる.

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#define x0 (1.0/sqrt(3.0))
#define N 80
int count[20000];
main()
{
    FILE *pt;
    double x;
        int i,bin1,bin2,bin3,bin4,bin5,bin6,bin7,bin8;
        bin1=bin2=bin3=bin4=bin5=bin6=bin7=bin8=0;
        printf("x0=%lf\n",x0);
    if((pt = fopen("test3.dat","wt")) !=NULL){
        fprintf(pt,"%d.",0);
        for(i = 0,x=x0; i < N; i++){
            x = 2.0*x+0.0000001;
            if(x<1.0){
                x = x;
                printf("%lf\n",x);
                count[i]=0;
                fprintf(pt,"%d",0);
            }else if(x>=1){
                printf("%lf\n",x);
                x = x-1.0+0.0000001;
                count[i]=1;
                fprintf(pt,"%d",1);
            }
        }
    }
    fclose(pt);
    for(i=0; i< N; i++){
        if((count[i]==0) && (count[i+1]==0) && (count[i+2]==0))
            bin1++;
        }else if((count[i]==0) && (count[i+1]==0) && (count[i+2]==1)){
            bin2++;
```

```

}else if((count[i]==0) && (count[i+1]==1) && (count[i+2]==0)){
bin3++;
}else if((count[i]==0) && (count[i+1]==1) && (count[i+2]==1)){
bin4++;
}else if((count[i]==1) && (count[i+1]==0) && (count[i+2]==0)){
bin5++;
}else if((count[i]==1) && (count[i+1]==0) && (count[i+2]==1)){
bin6++;
}else if((count[i]==1) && (count[i+1]==1) && (count[i+2]==0)){
bin7++;
}else if((count[i]==1) && (count[i+1]==1) && (count[i+2]==1)){
bin8++;}
}
printf("%d %d %d %d %d %d %d %d\n",bin1,bin2,bin3,bin4,bin5,bin6,bin7,bin8);
}

```

上記のプログラムで 2 進小数点以下 80 桁までの結果は、2 進小数が

```
0.1001001111001101001111001101000111111001000111000110000111101000000011
0010101111
```

であり、それぞれのビンへ入った個数の分布が

```
10 12 7 10 13 5 10 13
```

また、1000 桁まで使うと 2 進小数が

```
0.10010011110011010011110011010001111110010001110001100001111010000000110
0101011111101011101110101101011110110000110011001100101110010111111001110
0001111101000110110000110010001110001010011110110110010000111001011000110
0111010101010111000011111001000100000100001101100100111010101111101000001
0001001101001001000111101100101000100010000010001001101001011111001110011
1100001000011111101111100011010011011010000001011000110001100111110111101
0111110101001001010111000101101110000010110001001001010100100011011110001
0010001011100011001101010001001000110010001110110111110000011010110011110
101100001110111111110010101101110110011100001100110010111101100110000011
0011100011010001010011101101100111101100101100110001000000011110110000010
0100100110001001000110000000011111010011011100110110111101000000000010110
101101001110010110001000101111001110111111000001110001011101101001111101
1011011010000001001010111101100100001001100001010100100111101010001110101
00111101111000111000111101111100111010100011111100110
```

であり、それぞれのビンへ入った個数の分布は

```
120 132 109 132 133 109 132 133
```

となる。

6 写像の折りたたみ度と軌道のエントロピー

前回ではロジスティック写像などから生成される軌道の複雑さを「稠密性」という観点から見てきた。今回はこの複雑さを別の指標を用いて考えてみたい。

6.1 写像の折りたたみ度

既に学んだように、ロジスティック写像からの軌道をグラフを用いて求める方法においては、 $L_a(x), L_a^2(x) = L_a(L_a(x)), \dots, L_a^n(x)$ のグラフと $y = x$ の交点の関係によって、[固定点] [周期解] [複雑な軌道] のいずれになるのかが決まった。そこで、例えば、軌道が固定点に収束する $a = 2$ のロジスティック写像: $L_2(x) = 2x(1-x)$ に対し、 $L_2^2(x), L_3^2(x), L_4^3(x)$ を x の関数としてプロットしてみると図 19 のようになる。この図と、 $a = 4$ のロジスティック写像で同様な振る舞いをプロッ

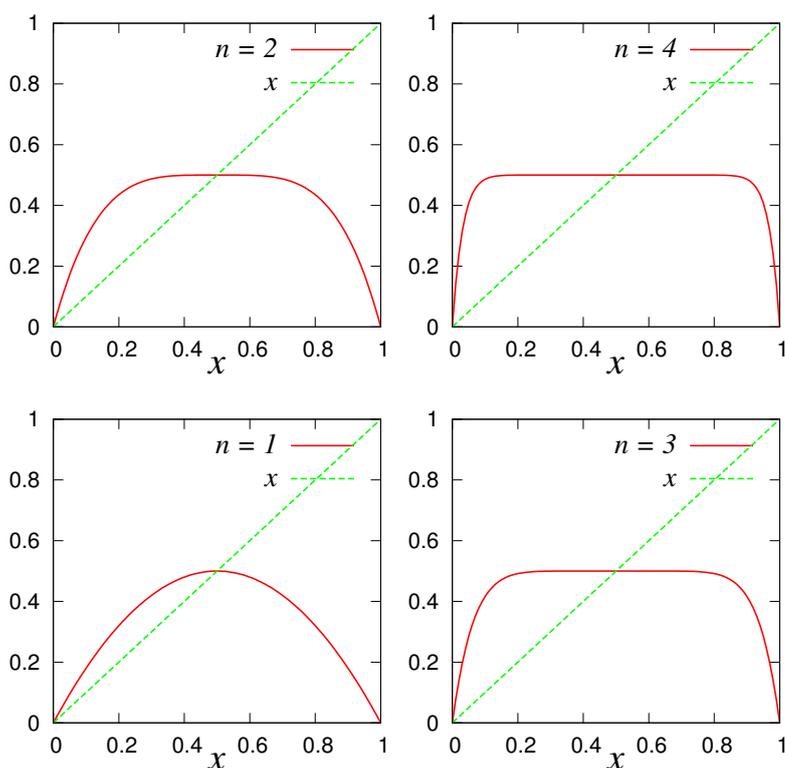


図 19: $y = L_2^1, \dots, L_2^4$ と $y = x$. $L_2(x) = 2x(1-x)$ である.

トした第 2 回講義ノート p.14 の図 7 を比べてみると、遥かに $a = 4$ のロジスティック写像の $L_4^n(x)$ の形状の方がより複雑である。

そこで、写像の n 解繰り返し $L_a^n(x)$ において単調に増加 (あるいは減少) する区間がいくつあるのかを数え、それを「写像の折りたたみ度」として定義し、 $w_a(n)$ と表記することにしよう。p.14 の図 7 より、 $a = 4$ の場合にこの $w_4(n)$ は明らかに n の増加とともに倍々に増えていくから、 $w_4(n) = 2^n$ である。一方、 $a = 2$ の場合には図 19 より、 n に依らずに $w_2(n) = 2$ であるように思われる。実際、

これはきちんと確かめることができる. そのためには $dL_2^n(x)/dx = 0$ が $x = 1/2$ のみを解に持つことを示せばよい. 陰関数の微分:

$$\frac{d}{dx}\{f(g(x))\} = \frac{df(x)}{dg(x)} \cdot \frac{dg(x)}{dx} \tag{88}$$

を思い出すと, これを繰り返し用いることにより

$$\begin{aligned} \frac{dL_2^n(x)}{dx} &= \frac{dL_2^n(x)}{dL_2^{n-1}(x)} \cdot \frac{dL_2^{n-1}(x)}{dL_2^{n-2}(x)} \cdot \frac{dL_2^{n-2}(x)}{dL_2^{n-3}(x)} \cdots \frac{dL_2^2(x)}{dL_2(x)} \cdot \frac{dL_2(x)}{dx} \\ &= 2(1 - 2L_2^{n-1}(x)) \cdot 2(1 - 2L_2^{n-2}(x)) \cdots 2(1 - 2L_2(x)) \cdot 2(1 - 2x) = 0 \end{aligned} \tag{89}$$

が得られる. 従って, $L_2(x) = L_2^2 = \cdots = L_2^{n-1}(x) = 1/2$ を満たす x は $x = 1/2$ であり, $L_2^n(x) = 0$ の解は n に依らずに $x = 1/2$ の一つだけである¹⁹ 従って, L_2^n の単調な区間数は n に依らずに $w_2(n) = 2$ である.

6.2 軌道のエントロピー

そこで, この写像の折りたたみ度をもとに**軌道のエントロピー**: h_a を次式で定義する.

$$h_a = \lim_{n \rightarrow \infty} \frac{\log w_a(n)}{n} \tag{90}$$

よって, この式に従えば $a = 2$ の場合には

$$h_2 = \lim_{n \rightarrow \infty} \frac{\log 2}{n} = 0 \tag{91}$$

であり, $a = 4$ の場合には

$$h_4 = \lim_{n \rightarrow \infty} \frac{\log 2^n}{n} = \lim_{n \rightarrow \infty} \frac{n \log 2}{n} = \log 2 \tag{92}$$

となり, 有限の値を持つ. そこで, この軌道のエントロピーが有限値をとるような軌道を複雑 — カオス — であると定義することにする. 簡単にわかるが, 先に学んだテント写像, ベルヌーイ写像に対する軌道のエントロピーは全て有限値をとるから, この意味でカオスであると言える. このことは写像の重なり度に立ち戻って考えると, 写像の反復数 n とともに単調部分が指数関数的に増加していく写像から生成される軌道はカオスであるということになる.

7 軌道の安定性とリアプノフ指数

図 20 のように, 写像 $x_{n+1} = f(x_n)$ において x_0 からスタートした写像の 1 反復で初期値のずれ Δx_0 が Δx_1 に拡大した (あるいは縮小) とすると

$$\frac{\Delta x_1}{\Delta x_0} = \tan \theta = |f'(x_0)| \tag{93}$$

¹⁹ 念のため証明しておく. まず, $L_2^n(1/2) = 1/2$ が任意の n で成り立つことを帰納法で示す. $L_2(x) = 1/2$ は $4x^2 - 4x + 1 = (2x - 1)^2 = 0$ なので $n = 1$ のとき成立. $L_2^n(1/2) = 1/2$ の成立を仮定すれば, $L_2^{n+1}(1/2) = L_2(L_2^n(1/2)) = L_2(1/2) = 1/2$ なので $n + 1$ で成立. よって, $L_2^n(1/2) = 1/2$ が任意の n で成り立つ. 実解 $x = 1/2$ の唯一性は $L_2(x) = 1/2$ のときに $4x^2 - 4x + 1 = (2x - 1)^2 = 0$ なので実解は $x = 1/2$ のみ. 従って, $L_2^2 = L_2(L_2(x)) = 1/2$ を満たす $L_2(x)$ は $L_2(x) = 1/2$ のみであるが, $L_2(x) = 1/2$ を満たす x は $1/2$ のみだったので, 結局, $L_2^2(x) = 1/2$ を満たす x も $1/2$ のみ. この議論を繰り返していけばよい.

つまり

$$\Delta x_1 = |f'(x_0)|\Delta x_0 \tag{94}$$

が成り立つ。ここに、 $f'(x_0) = (df(x)/dx)|_{x=x_0}$ のような略記を用いたことに注意されたい。そこ

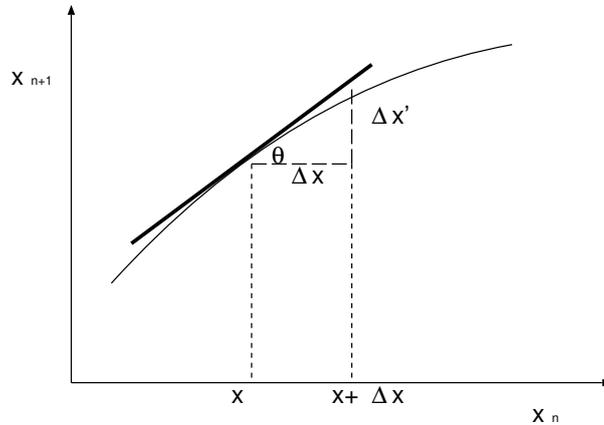


図 20: 写像の 1 反復で初期値のずれ $\Delta x = \Delta x_0$ が $\Delta x' = \Delta x_1$ に拡大する。

で、さらに、 Δx_1 が次の反復で Δx_2 に拡大された (あるいは縮小) とすれば、上記の議論を繰り返すことで

$$\Delta x_2 = |f'(x_1)|\Delta x_1 = |f'(x_1)||f'(x_0)|\Delta x_0 \tag{95}$$

が得られる。従って、これを n 回繰り返せば

$$\Delta x_n = |f'(x_{n-1})||f'(x_{n-2})|\cdots|f'(x_1)||f'(x_0)|\Delta x_0 \tag{96}$$

となる。従って、写像の n 回反復により、初期値のずれ Δx_0 は $|f'(x_{n-1})||f'(x_{n-2})|\cdots|f'(x_1)||f'(x_0)|\Delta x_0$ に拡大 (あるいは縮小) される。

感じをつかむために、最も簡単な場合に対し、上記の拡大 (あるいは縮小) についてみてみよう。まず、全ての反復に対し、 $|f'(x_i)| \geq \delta$ ($\delta > 1$ の定数) とすると

$$\Delta x_n \geq \delta^n \Delta x_0 = e^{n \log \delta} \Delta x_0 \rightarrow \infty \quad (n \rightarrow \infty) \tag{97}$$

となり、初期値のずれは繰り返し回数の増加とともに無限に拡大されることがわかる。従って、軌道のずれが拡大されるかどうかは上記の指数の肩に乗っている $\log \delta$ の符号をみればよいことがわかる。これはつまり

$$\begin{aligned} \lambda &= \underbrace{\frac{1}{n} \log \left(\frac{\Delta x_n}{\Delta x_0} \right)}_{\text{対数スケールで測った単位ステップあたりの拡大率}} \\ &= \frac{1}{n} \log |f'(x_{n-1})||f'(x_{n-2})|\cdots|f'(x_1)||f'(x_0)| \\ &= \frac{1}{n} \left\{ \log |f'(x_{n-1})| + \cdots + \log |f'(x_0)| \right\} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log |f'(x_i)| \end{aligned} \tag{98}$$

で λ を定義すると、これが正の場合には初期値のずれは無限に拡大されることを意味する。つまり、上式の第 1 行目を $\Delta x_n / \Delta x_0$ に関して解くと

$$\frac{\Delta x_n}{\Delta x_0} = e^{n\lambda} \quad (99)$$

となり、 $\lambda > 0$ ならば、軌道の拡大率 $\Delta x_n / \Delta x_0$ は $n \rightarrow \infty$ で $e^{n\lambda} \rightarrow \infty$ のように指数関数的に発散する。逆に $\lambda < 0$ ならば、初期値の誤差 Δx_0 は写像の繰り返しと共に指数関数的に小さくなっていき、しまいには、初期値の異なる 2 つの軌道は一致してしまう。

この λ は **リアプノフ指数** と呼ばれる。従って、初期値のずれが反復回数とともに指数的に増加していくような複雑な軌道 — カオス — を判定するためには、このリアプノフ指数を計算し、その符号の正負を調べればよい。すなわち、**カオスは正のリアプノフ指数で特徴づけられる**。

7.1 いくつかの計算例

以下で直ぐに計算のできるいくつかの例を見ておこう。

7.1.1 テント写像のリアプノフ指数

テント写像:

$$x_{n+1} = 1 - |1 - 2x_n| = \begin{cases} 2x_n & (0 \leq x_n \leq 1/2) \\ 2 - 2x_n & (1/2 < x_n \leq 1) \end{cases} \quad (100)$$

の場合、 x の値によらずに $|f'(x)| = |T'(x)| = 2$ であるから、このリアプノフ指数は

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log |T'(x_i)| = \lim_{n \rightarrow \infty} \frac{n \log 2}{n} = \log 2 > 0 \quad (101)$$

となり、瞬時に求めることができる。従って、テント写像から生成される軌道はカオスである。

7.1.2 ベルヌーイ写像のリアプノフ指数

ベルヌーイ写像:

$$x_{n+1} = B(x_n) = \begin{cases} 2x_n & (0 \leq x_n < 1/2) \\ 2x_n - 1 & (1/2 \leq x_n \leq 1) \end{cases} \quad (102)$$

の場合、テント写像と同様に $|f'(x)| = |T'(x)| = 2$ であるから、このリアプノフ指数は $\lambda = \log 2$ であり、ベルヌーイ写像から生成される軌道はカオスであると結論づけられる。

7.1.3 ロジスティック写像 $L_4(x)$ のリアプノフ指数

$a = 4$ のロジスティック写像 $L_4(x) = 4x(1-x)$ の場合、 $|L_4'(x)| = |4(1-2x)|$ であるから、 x の値によって 1 階微分の係数の値は異なる。従って、定義式に立ち返って (98) 式を数値的に評価する

必要があるように思われるが, 前回みたように密度関数 $P(x)$ が解析的に与えられているので, これを用いて計算を進めることができる. 具体的には

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log |f'(x_i)| = \int_0^1 dx P(x) \log |L_4'(x)| = \int_0^1 dx \frac{\log |4(1-2x)|}{\pi \sqrt{x(1-x)}} \quad (103)$$

のように和を積分に置き換えることができる. あとはこの x に関する積分が実行できればよい. 被積分関数が $x = 1/2$ を中心に対称であることに注意し, 前回まで度々出てきた $x = \sin^2 \theta$ なる変換を施すと

$$\begin{aligned} 2 \int_0^{1/2} dx \frac{\log |4(1-2x)|}{\pi \sqrt{x(1-x)}} &= 2 \int_0^{\pi/4} \frac{\log(4 \cos 2\theta)}{\pi \sin \theta \cos \theta} \cdot 2 \sin \theta \cos \theta \\ &= \log 4 + \frac{4}{\pi} \int_0^{\pi/4} \log(\cos 2\theta) d\theta \equiv \log 4 + I \end{aligned} \quad (104)$$

となる. この最終式第 2 項の積分は次のようにいくつかの変数変換, および, 積分区間の対称性から次のように書き換えることができる.

$$I = \frac{2}{\pi} \int_0^{\pi/2} \log(\cos \theta) d\theta = \frac{2}{\pi} \int_0^{\pi/2} \log(\sin \theta) d\theta \quad (105)$$

ここで, 最後の変形では変数変換: $\theta \rightarrow \pi/2 - \theta$ を用いたことに注意されたい. この式の 2 番目と 3 番目の表現を用いると

$$\begin{aligned} I &= \frac{1}{2} \cdot \frac{2}{\pi} \left\{ \int_0^{\pi/2} \log(\cos \theta) d\theta + \int_0^{\pi/2} \log(\sin \theta) d\theta \right\} \\ &= \frac{1}{2} \cdot \frac{2}{\pi} \int_0^{\pi/2} \log(\cos \theta \sin \theta) d\theta \\ &= \frac{1}{2} \cdot \frac{2}{\pi} \int_0^{\pi/2} \log\left(\frac{\sin 2\theta}{2}\right) \\ &= \frac{1}{2} \cdot \frac{2}{\pi} \int_0^{\pi/2} \{\log(\sin 2\theta) - \log 2\} \\ &= -\frac{1}{2} \log 2 + \frac{1}{2} \cdot \frac{1}{\pi} \int_0^{\pi} \log(\sin \theta) d\theta = -\frac{1}{2} \log 2 + \frac{I}{2} \end{aligned} \quad (106)$$

となる. ここで最後の等式では $2\theta \rightarrow \theta$ と変数変換したことに注意されたい. 従って, これを I について解いて, $I = -\log 2$ が得られる. 従って, ロジスティック写像 $L_4(x)$ のリアプノフ指数は

$$\lambda = \log 4 - \log 2 = \log 2 \quad (107)$$

となり, これはテント写像, バルヌーイ写像と同じ値となる.

7.1.4 リアプノフ指数の数値計算例: $L_{a \neq 4}(x)$ のリアプノフ指数

前節までの結果はリアプノフ指数が比較的簡単な解析計算で求まる場合であった. しかし, そのような場合はむしろ特殊であり, 通常は計算機を用いて数値計算を行わなければリアプノフ指数を求めることはできない. そのような数値計算のなかで軌道の生成メカニズムとして写像が与えられ

ている場合は比較的簡単であり, 定義 (98) に従ってリアプノフ指数を数値計算すればよい²⁰. 例えば, ロジスティック写像 $L_{a \neq 4}(x)$ の場合には $|L'_a(x)| = a|1 - 2x|$ であったから, そのリアプノフ指数は

$$\lambda_a = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log(a|1 - 2x_i|) \tag{108}$$

で与えられる. 従って, いつものように反復で生成される x_0, x_1, \dots を用いて上記の和を直接計算すればよい. しかし, 計算機上では上記の n の無限大極限は取れないので, 大きな有限の n で置き換える. 結果を図 21 に載せる. この図を得るために, 写像の繰り返し回数を $n = 1000000$, a の刻み

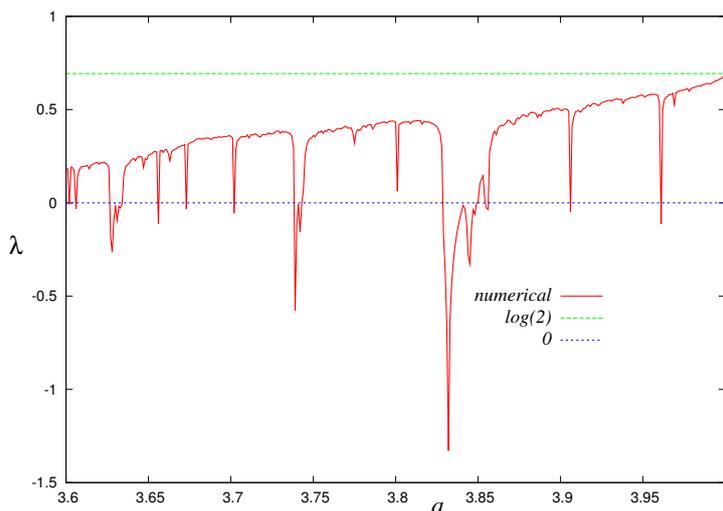


図 21: ロジスティック写像 $L_a(x)$ のリアプノフ指数. 写像の繰り返し回数を $n = 1000000$, a の刻み幅を 0.001 に選んだ.

幅を 0.001 に選んだ (担当者のノート型パソコンで 5, 6 分の計算時間). この図より, a の変化とともにリアプノフ指数の形状は複雑に変化し, いくつかの a の値に対し, リアプノフ指数は負になっている. また, 図 21 の区間 $[0.82 : 0.85]$ を拡大したものを図 22(左) に, また, この図をさらに拡大したものを図 22(右) に載せる. この図から, リアプノフ指数の形状には「自己相似的」(各部分が全体の縮小版となっている) な性質をみることができ. このあたりの詳細は後日に見ていくことになる.

7.1.5 参考: 固定点, 周期解に対するリアプノフ指数

固定点に引き込まれる場合, 周期解を持つ場合のリアプノフ指数は一部解析的に求めることができる. 例えば, 固定点 x_* ($0 \leq x_* \leq 1$) を持つ場合のリアプノフ指数は, 密度関数がデルタ関数を用いて $P(x) = \delta(x - x_*)$ で与えられるので, デルタ関数に関する恒等式: $\int_0^1 \delta(x - x_*) f(x) dx = f(x_*)$ に注意して

$$\lambda = \int_0^1 \delta(x - x_*) \log a|1 - 2x| dx = \log a|1 - 2x_*| = \log a \left| 1 - \frac{2(a-1)}{a} \right| \tag{109}$$

²⁰ 単にデータ x_0, x_1, \dots のみが与えられている状況では工夫が必要となる. 例えば, このデータから「実質的」な写像の形を推定し, その数値的に推定された写像 (関数) を補間しつつ数値微分を用いて和 (98) を計算したりする. そのような計算例は時間が許せば後の講義で紹介したい.

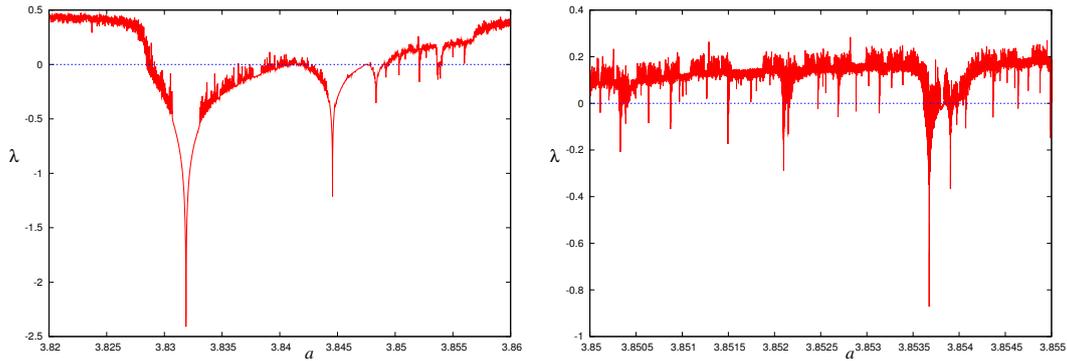


図 22: 図 21 の拡大図.

となる. ここで, 最後の等式は $ax_*(1 - x_*) = x_*$ 非零の解, $x_* = (a - 1)/a$ を用いた. これを図 23(左) にプロットする. この図より, 数値的に求めたリアプノフ指数と上記 (109) 式はロジスティック写像が固定点を持つ範囲内で一致することがわかる. また, ロジスティック写像が周期解を持つ場合, その周期が $T: (x_1^*, x_2^*, \dots, x_T^*)$ であるとする, 密度関数が

$$P(x) = \frac{1}{T} \sum_{i=1}^T \delta(x - x_i^*) \tag{110}$$

で与えられることに注意して, リアプノフ指数は

$$\lambda = \frac{1}{T} \int_0^1 \sum_{i=1}^T \delta(x - x_i^*) \log a |1 - 2x| dx = \frac{1}{T} \sum_{i=1}^T \log a |1 - 2x_i^*| \tag{111}$$

と形式的に書くことができる. この式から明らかなように, これを使うためには, 周期解を構成する x の値 $(x_1^*, x_2^*, \dots, x_T^*)$ を全て求めておく必要があり, T が増えるにつれてこの方法もやはり厄介になってくる. 図 23(右) に $[1, 4]$ の範囲内の a に対するリアプノフ指数を載せる.

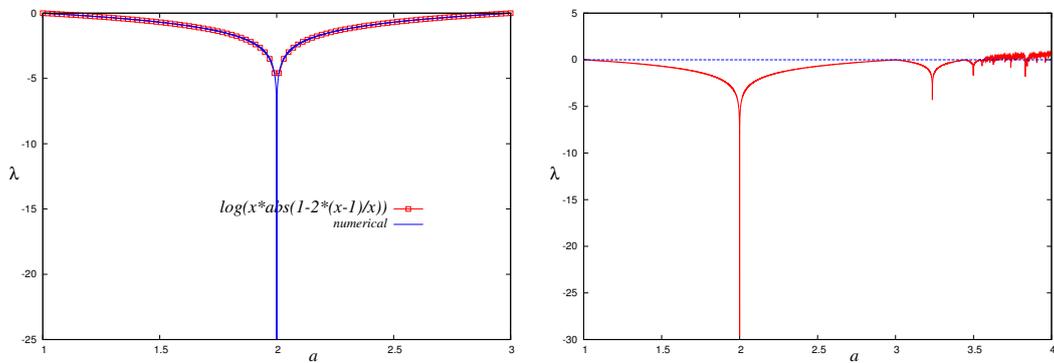


図 23: 固定点を持つ場合のロジスティック写像のリアプノフ指数 (左). 右図は $[1, 4]$ の範囲内の a に対するリアプノフ指数.

以上の結果より, ロジスティック写像のパラメータ a を変化させると, 軌道がある軌道から別の軌道へ移り変わることが示唆される. そこで, 次回はこのような写像における分岐現象について詳

しく見ていくことにする.

レポート課題 4

既に学んだテント写像はパラメータ a を導入することで, 次のように一般化することができる.

$$T_a(x) = a(1 - |1 - 2x|), \quad x_{n+1} = T_a(x_n) \quad (112)$$

a を $1/2 \leq a \leq 1$ の範囲で変化させ, 各 a でリアプノフ指数を計算し, 図 21 に倣って結果を図示せよ.

課題 4 の解答例

まずは前回学んだリアプノフ指数の計算例を見ておこう. テント写像はパラメータ a を導入することで次のように一般化することができる.

$$T_a(x) = a(1 - |1 - 2x|), \quad x_{n+1} = T_a(x_n) \quad (113)$$

リアプノフ指数の定義から, (113) 式に基づいて生成される軌道の各点での $|T'_a(x)|$ が求まればリアプノフ指数が計算できる. これは明らかに x の値に依らず $2a$ であるから数値計算するまでもなく

$$\lambda_a = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log |T'_a(x_i)| = \lim_{n \rightarrow \infty} \frac{n \log 2a}{n} = \log 2a \quad (114)$$

である. ということは, a を $1/2 \leq a \leq 1$ の範囲で変化させると必ずリアプノフ指数は正の値になり, パラメータ a で一般化されたテント写像から生成される軌道はカオスとなる. また, a を変化しつつ, 写像からの軌道の取りうる値を重ね打ちすると興味深い現象が見られる. その結果をリアプノフ指数の a 依存性ととも図 24 に載せる.

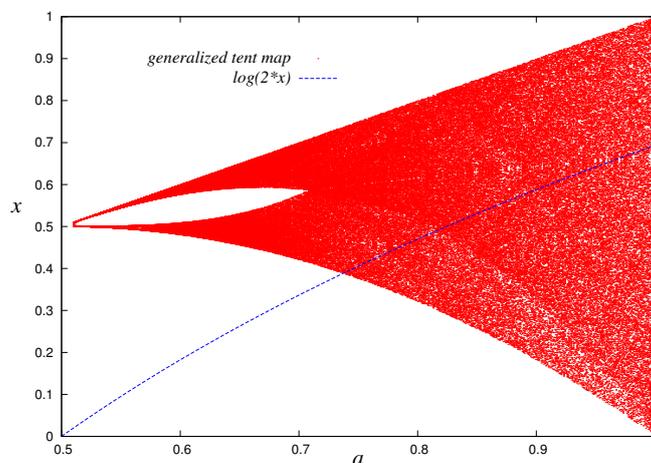


図 24: 一般化されたテント写像の分岐図と対応するリアプノフ指数.

参考までにロジスティック写像 $L_{a \neq 4}(x)$ のリアプノフ指数を計算するためのプログラムの一部を以下に載せておこう²¹.

```

/* 写像が陽に与えられた場合のリアプノフ指数の数値計算 */
for(a=1.00; a <= 4.00; a = a + 0.0001){
    for(i = 0,sum=0.0,x=x0; i <= N; i++){
        // x = a*(1.0-fabs(1-1.999*x)); /* テント写像ではこちらを使う */
        x = a*x*(1-x);
        sum = sum + log(a*fabs(1.0-2.0*x)); /* 写像の出力結果 x に対して和を計算 */
    }
}

```

²¹ ここではファイルのオープン/クローズを使わないが, ファイル名を指定してリダイレクションすればよい.

```
printf("%lf %lf\n",a,(double)sum/N);
}
```

8 分岐現象とカオス

今回の講義では今まで学んできたロジスティック写像 $L_a(x)$ に対し、パラメータ a の値を変えていくことで、写像からの軌道が周期運動からカオスへと段階的に変化する様子を「分岐現象」として捉え、その分岐が生じるたびに今まで安定であった周期解が不安定化し、さらに周期の長い解が安定解として現れる様子を見ていく。 a の変化による周期解の不安定化と、さらなる周期解の出現を繰り返し、ある臨界パラメータ a 以上でカオスが現れることを学習する。また分岐図には「窓」や「自己相似構造」などの特異な性質が現れることも同時に見ていく。

8.1 ロジスティック写像 $L_a(x)$ 再考

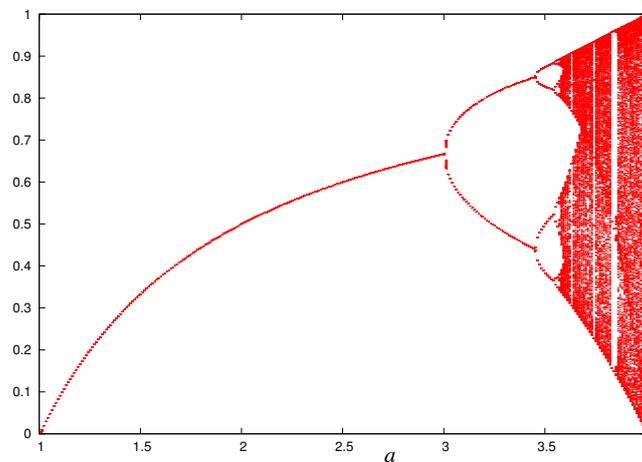


図 25: ロジスティック写像のパラメータ a に対する軌道の分岐図。

まずは図 25 にロジスティック写像のパラメータ a に対する軌道の分岐図を載せよう。この図は横軸にパラメータ a を縦軸に写像: $x_{n+1} = ax_n(1 - x_n)$ に対し、 x_n の取りうる値を重ねうちしたものである²²。先に載せたりアプノフ指数の計算とほぼ同じループ処理を行えば良いのだが、念のため下記に図 25 を計算する際に用いたプログラムの一部を載せておく。

```
/* 分岐図計算のプログラム */
for(a=1.00; a <= 4.00; a = a + 0.0001){
    for(i = 0,sum=0.0,x=x0; i <= N; i++){
        // x = a*(1.0-fabs(1-1.999*x)); /* テント写像の場合はこちらを使う */
```

²² 細かな計算技術上の話になるが、固定点に収束する場合でさえも、その固定点へ 1 ステップで収束するとは限らない。多くの場合、このような固定点 (あるいは周期解) へ収束するまでには有限回のステップを必要とする。このような収束までに経過する x の点までプロットするとうまくないので、はじめの数ステップはかさ値打ちせずに捨てるなどの工夫をすると良い。

```

x= a*x*(1-x);
  if(i >= Nskip){
    printf("%lf %lf",a,x);
  }
}
}

```

8.1.1 固定点とその安定性

この図より、パラメータ a を $a = 1$ から徐々に大きくしていくと $a = 3$ 付近で固定点は 2 周期軌道に分岐する。そこで、まずは固定点の位置 x_* を a の関数として求めるために、 $L_a(x) = x$ を x について解いてみると、 $x_* = 0$ 、および、 $x_* = (a-1)/a$ が得られる。これを図 25 に重ねうちしたものを図 26 に載せる。この図より、 $a = 3$ 近傍まで $x_* = (a-1)/a$ と写像 $x_{n+1} = L_a(x_n)$ から得られ

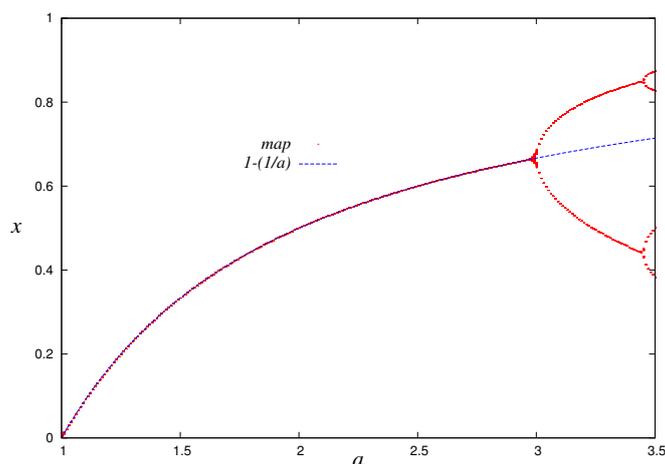


図 26: 写像から得られる分岐の様子と固定点を a の関数として求めたもの $x_* = (a-1)/a$ を重ねうちしてプロットした。

る軌道のプロットは完全に一致するが、 $a > 3$ では実際に写像から得られる軌道は $x_* = (a-1)/a$ ではなく、その両側に現れる「2 周期軌道」となっている。従って、固定点 $x_* = (a-1)/a$ は $a > 3$ においても解としては存在するが、それは安定ではない。

そこで、リアプノフ指数の導出の議論で見たように、 $|\{dL_a(x)/dx|_{x=x_*}\}|$ が 1 をこえる条件を調べてみると、直ちに

$$\left| \left\{ \frac{dL_a(x)}{dx} \Big|_{x=x_*} \right\} \right| = \left| a \left\{ 1 - \frac{2(a-1)}{a} \right\} \right| = |2-a| > 1 \quad (115)$$

が得られ、従って、これは $(1-a)(3-a) > 1$ すなわち、 $a > 3$ である。よって、固定点 $x_* = (a-1)/a$ は $a > 3$ で不安定化する ($1 < a < 3$ で安定である)。

ところで、 $L_a(x) = x$ のもう一つの解 $x_* = 0$ は実現されないのか、気になるかもしれない。実際、この解は常に存在するのであるが、これもやはり安定に実現されるための条件がある。そこで、上記のように $L_a(x)$ の一階微分の $x_* = 0$ での絶対値が 1 をこえる条件 (不安定化条件) を出してみると $|\{dL_a(x)/dx|_{x=0}\}| = a > 1$ であるから、 $a > 1$ の領域で $x_* = 0$ は決して実現されない。逆に、 $a \leq 1$ であれば $x_* = 0$ は安定な固定点である (各自が写像の反復数値計算で確かめてみよ)。

8.1.2 2 周期解とその安定性

図 25 の数値計算結果によると $a > 3$ では固定点の代わりに 2 周期解が実現されるわけであるから、この 2 周期解を求めてみる。これは $L_a^2(x) = L_a(L_a(x)) = x$ を解けばよい。 $x, (-ax + a - 1)$ を因数に持つことに注意して $(x_* = 0, (a - 1)/a$ は常に解として存在する)

$$L_a^2(x) - x = x(-ax + a - 1)\{a^2x^2 - a(a + 1)x + a + 1\} = 0 \tag{116}$$

すなわち、 $a^2x^2 - a(a + 1)x + a + 1 = 0$ の実解である

$$x_{*2}^{(+)} = \frac{a + 1 + \sqrt{(a + 1)(a - 3)}}{2a}, \quad x_{*2}^{(-)} = \frac{a + 1 - \sqrt{(a + 1)(a - 3)}}{2a} \tag{117}$$

の 2 つが 2 周期解の「枝」であり、それぞれ上記のような a の関数である。従って、 $x_{*2}^{(\pm)}$ が安定か

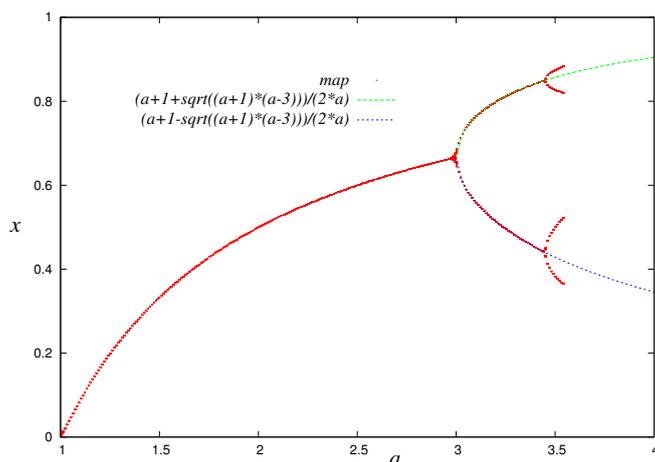


図 27: 写像から得られる分岐の様子と 2 周期点を a の関数として求めたもの $x_{*2}^{(+)}, x_{*2}^{(-)}$ を重ねうちしてプロットした。

どうかは $L_a^2(x)$ の $x_{*2}^{(\pm)}$ における微分係数が 1 よりも大きいかどうかを調べればよい²³。前回見た陰関数の微分を思い出すと

$$\frac{dL_a^2(x)}{dx} = \frac{dL_a^2(x)}{dL_a(x)} \cdot \frac{dL_a(x)}{dx} = a(1 - 2L_a(x)) \cdot a(1 - 2x) \tag{118}$$

であるから、この右辺を $x_{*2}^{(\pm)}$ で評価すればよい。

これは面倒で骨の折れる作業であるが、以下で $x_{*2}^{(+)}$ について地道に計算してみると、まず

$$\begin{aligned} L_a(x_{*2}^{(+)}) &= ax_{*2}^{(+)}(1 - x_{*2}^{(+)}) \\ &= \frac{a + 1 + \sqrt{(a + 1)(a - 3)}}{2} \cdot \frac{a - 1 - \sqrt{(a + 1)(a - 3)}}{2a} \\ &= \frac{a + 1 - \sqrt{(a + 1)(a - 3)}}{2a} = x_{*2}^{(-)} \end{aligned} \tag{119}$$

が得られ、従って、 $a(1 - 2L_a^{(+)}(x_{*2})) = \sqrt{(a + 1)(a - 3)} - 1$ である。一方

$$a(1 - 2x_{*2}^{(+)}) = -1 - \sqrt{(a + 1)(a - 3)} \tag{120}$$

²³ 2 周期解の安定性を調べるのであるから、 $L_a(x)$ の微分係数ではなく $L_a^2(x)$ の微分係数を見る必要があることに注意。

であるから, 結局

$$\left. \frac{dL_a^2(x)}{dx} \right|_{x=x_{*2}^{(+)}} = -(\sqrt{(a+1)(a-3)} - 1)(\sqrt{(a+1)(a-3)} + 1) = -(a^2 - 2a - 4) \quad (121)$$

となる. 従って, $|a^2 - 2a - 4| < 1$ が安定条件であるが, 両辺を自乗して, 右辺の 1 を左辺に移してまとめると $(a+1)(a+\sqrt{6}-1)(a-3)(a-1-\sqrt{6}) < 0$ が得られる. よって, $a-3 > 0$ かつ $a-1-\sqrt{6} < 0$, すなわち, $3 < a < 1+\sqrt{6}$ で $x_{*2}^{(+)}$ が安定に存在する²⁴.

次に $x_{*2}^{(-)}$ の安定性であるが, $L_a^2(x_{*2}^{(-)}) = x_{*2}^{(+)}$ であるから

$$a(1 - 2L_a^2(x_{*2}^{(-)})) = -1 - \sqrt{(a+1)(a-3)} \quad (122)$$

であり, また

$$a(1 - 2x_{*2}^{(-)}) = -1 + \sqrt{(a+1)(a-3)} \quad (123)$$

であるから, (118) 式より

$$\left. \frac{dL_a^2(x)}{dx} \right|_{x_{*2}^{(-)}} = -(a^2 - 2a - 4) = \left. \frac{dL_a^2(x)}{dx} \right|_{x_{*2}^{(+)}} \quad (124)$$

が得られる. 従って, $x_{*2}^{(-)}$ の安定領域は $x_{*2}^{(+)}$ の安定領域と同じ $3 < a < 1+\sqrt{6}$ であることがわかる. これは図 25 からも見て取れる.

8.1.3 2^n 周期解とその安定性

図 25 を見る限りにおいて, $x_{*2}^{(\pm)}$ も $a = 1 + \sqrt{6}$ を越えると不安定化し, $a > 1 + \sqrt{6}$ での安定解は 2 周期解から 4 周期解に移る. 従って, 前節までの議論をここで繰り返すためには $L_a^4(x) = x$ を満たす $x = 0, (a-1)/a, x_{*2}^{(\pm)}$ 以外の解を求め, その解に対する微分係数 $dL_a^4(x)/dx$ の絶対値を計算し, それと 1 と比較すればよい.

図 25 の分岐図から, その後, a の増加とともに周期は倍々に増えていくことがわかるので, その解の安定性の議論も上記と同じ手続きをふめばよい. しかし, 系統的な手続きがわかっていることと, それが実際に実行可能であるか否かは別問題である. 例えば, $L_a^4(x) = x$ を具体的に書き下してみると

$$\begin{aligned} & a(a(a(ax(1-x))(1-(ax(1-x))))(1-(a(ax(1-x))(1-(ax(1-x)))))) \\ \times & (1-a(a(ax(1-x))(1-(ax(1-x))))(1-(a(ax(1-x))(1-(ax(1-x)))))) - x = 0 \end{aligned} \quad (125)$$

となり, x に関する 16 次方程式となる. 上記方程式は $x = 0, (a-1)/a, x_{*2}^{(\pm)}$ を解に持つので, $x(x - (a-1)/a)(x - x_{*2}^{(+)}) (x - x_{*2}^{(-)})$ を因数に持つことがわかるから, この多項式で (125) 式左辺を割れば, その商である方程式の次数は 12 次まで減少する. しかし, これでもその実解を解析的に求めることは不可能である.

このような場合には, 2 年生のときに履修した「計算機プログラミング I」の講義で学んだであろう「ニュートン法」などの数値計算技法を用いて多項式のゼロ点を探すことになる (⇒ **今週のレポート課題**).

²⁴ はじめの 2 項の積: $(a+1)(a+\sqrt{6}-1)$ はこの領域で常に正なので問題にはならないことに注意.

8.1.4 カオスが出現する分岐パラメータ a 値の粗い見積もり

前節で述べたように、周期はパラメータ a の増加とともに倍々で増加してゆき、ある極限でカオスが現れることが予想される。この分岐の詳細を正確に議論することは前述の理由で難しいのであるが、かなり粗い見積もりならばカオスが出現する a の近似値を評価することができる。以下でそれを述べよう。

今、我々が正確にわかっているのは $x = 0$ から $x = (a-1)/a$ への分岐点: $a_0 = 1$, $x = (a-1)/a$ から、 $x = x_{*2}^{(\pm)}$ への分岐点: $a_1 = 3$, $x = x_{*2}^{(\pm)}$ から 4 周期解 (具体的には求まっていないが) への分岐点: $a_2 = 1 + \sqrt{6}$ である。そこで、 $a_2 - a_1$, $a_1 - a_0$ の比、つまり、「階差の比」を計算してみると

$$\frac{a_2 - a_1}{a_1 - a_0} = \frac{1 + \sqrt{6} - 3}{3 - 1} = \frac{\sqrt{6} - 2}{2} = 0.225 \equiv r \quad (126)$$

となる。次の階差比: $(a_3 - a_2)/(a_2 - a_1)$ を計算するためには a_3 の具体的な値が必要であるが、我々は(さしあたり)それを求めることができない。そこで、これ以降、**任意の階差比**: $(a_{n+2} - a_{n+1})/(a_{n+1} - a_n)$ についても (126) 式の公比 r が有効であると仮定する。すなわち、階差は次の等比級数に従うとする。

$$a_{n+2} - a_{n+1} = r(a_{n+1} - a_n) \quad (127)$$

これを具体的に書き下してみると

$$\begin{aligned} a_2 - a_1 &= r(a_1 - a_0) \\ a_3 - a_2 &= r(a_2 - a_1) \\ &\dots \dots \dots \\ a_\infty - a_{\infty-1} &= r(a_{\infty-1} - a_{\infty-2}) \end{aligned}$$

となるから、これらを辺々足していくと

$$a_\infty - a_1 = r(a_{\infty-1} - a_0) \quad (128)$$

が得られる。そこで $a_\infty = a_{\infty-1}$ と置き、 a_∞ について解くと

$$a_\infty = \frac{a_1 - ra_0}{1 - r} = \frac{3 - 0.225 \times 1}{1 - 0.225} = 3.58 \quad (129)$$

となる。数値計算による正確な値は $a_\infty = 3.569$ であることがわかっているので²⁵、これと比べてもかなり良い近似値になっていることがわかる。この結果と第 4 回講義ノート図 18(右) のリアプノフ指数が正となる領域の対応を考えてみるとよいであろう。

8.1.5 窓、自己相似構造

$a > a_\infty$ のパラメータ領域に対する分岐図を一部拡大してみると図 28 が得られる。この図からわかるように、ところどころに軌道が密でない周期解を持つ「窓」(4 周期窓や 3 周期窓)が見られる。また、この中で 3 周期窓の領域を越えたところを注目すると、図 25 の縮小版のような分岐図が見られ、自己相似的な「入れ子構造」を見ることが出来る。このような微細な構造の詳細については、時間があれば、この講義の後半(フラクタル編が一通り終わった後)で見えていくか、時間が無ければ、後期「情報工学演習 II」でとりあげたい。

²⁵ この正確な値を求めるための、やや込み入った方法は後期開講の「情報工学演習 II」で扱う(多分)。

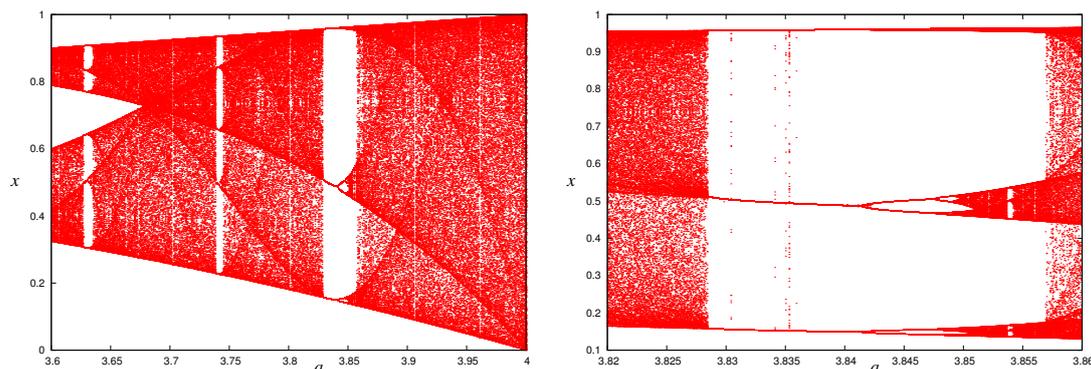


図 28: ロジスティック写像のパラメータ a に対する軌道の分岐図. 図 25 の一部を拡大したもの.

8.2 参考: ニュートン法の復習

$f(x) = a^2x^2 - a(a+1)x + a + 1 = 0$ の手の方程式 (x の多項式 = 0) の数値解を求めたい場合, ニュートン法と呼ばれる手法を用いるとよい.

この方法は方程式 $f(x) = 0$ の解の候補 $x = x_0$ で曲線 $y = f(x)$ に接する接線の方程式:

$$y = f(x_0) + f'(x_0)(x - x_0) \quad (130)$$

の x 軸との交点:

$$x_1 \equiv x_0 - \frac{f(x_0)}{f'(x_0)} \quad (131)$$

を解候補 x_0 の改良版とするもので, これを推し進めて n -番目の解候補と $n+1$ -番目の解候補の間に成り立つ漸化式

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (132)$$

を反復的に解き, この収束点を $f(x) = 0$ の解とするものである.

※注: 「2分法」というやり方でも解を求めることができる. 詳しくは

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/PROG2010/ProgI2010_5.pdf

を参照してみるとよい.

レポート課題 5

2 次方程式:

$$f(x) \equiv a^2x^2 - a(a+1)x + a + 1 = 0$$

をニュートン法により求めるプログラムを作成せよ. このとき

- x の初期値 x_0 を $[0, 1]$ の一様乱数により与え, 初期値をいくつか変えることで全ての実解が自動的に求まるようなプログラムを組むとなお良い.

- a の値を $3 < a < 1 + \sqrt{6}$ の範囲内で変化させると, 解の値も変化するが, この解を a の関数として求めるようなプログラムにするとよい。
- 求めた解 x に対し, $|dL_a^2(x)/dx| - 1$ の値を計算し, これがゼロ以上か否かを判定する機能を持たせるとよい。

上記 3 つの機能は「オプション」. 初期値を手で与えて解を求めるプログラムが書ければよい。

課題 5 の解答例

ニュートン法の素朴なソースコードを以下に載せる.

```
/* ニュートン法 */
#include <stdio.h>
#include <math.h>
double func(x,a) /* f(x)=0 を求める. f(x) の定義 */
double x;
double a;
{
    return (pow(a*x,2)-a*(a+1)*x+a+1);
}
double Dfunc(x,a) /* f(x) の 1 階微分の定義. 微分が困難な場合には数値微分値を返すよう
にする. */
double x;
double a;
{
    return (2*a*a*x-a*(a+1));
}
main()
{
    FILE *fpr;
    int i,imax=100;
    double x,y,a;
    if((fpr = fopen("lower.dat", "wt")) !=NULL){
        for(a=3.0; a<=1.0+sqrt(6.0); a=a+0.01){ /* a の値を振る */
            for(i = 0, x=0.10; i <= imax; i++){
                x = x - func(x,a)/Dfunc(x,a);
                if(fabs(y-x)<1.0e-5){
                    fprintf(fpr,"%lf %lf\n",a,x);
                    break;
                }else{
                    y = x;
                }
            }
        }
    }
    fclose(fpr);
}
```

このプログラムでは, a の値に対して解は 2 個しかないとわかっているため, その 2 つのうち大きい方の解を求めたい場合には x の取りうる上限値に近いところから反復をスタートさせ, 逆に, 小さい方の解を求めたい場合には x の取りうる下限値に近いところから反復をスタートさせれば

よい. ちなみに, a を少しずつ変化させてその解を求めているが, a を少し変化させても x の値が急激に変化しないようであれば, 一つ手前の a での解 x の値を a に関するループで次の a に対する反復の初期値 x として利用し, この間, 方程式が「断熱的」に振舞うと仮定してプログラムを書いてもよい. 具体的には次のように a のループの部分で x の値を一回だけ初期化すればよい.

```
for(a=3.0,x=0.10; a<=1.0+sqrt(6.0); a=a+0.01){ /* a の値を振る */
  for(i = 0; i <= imax; i++){
```

このプログラムからの出力を解析解とともにプロットしたものを図 29 に載せよう.

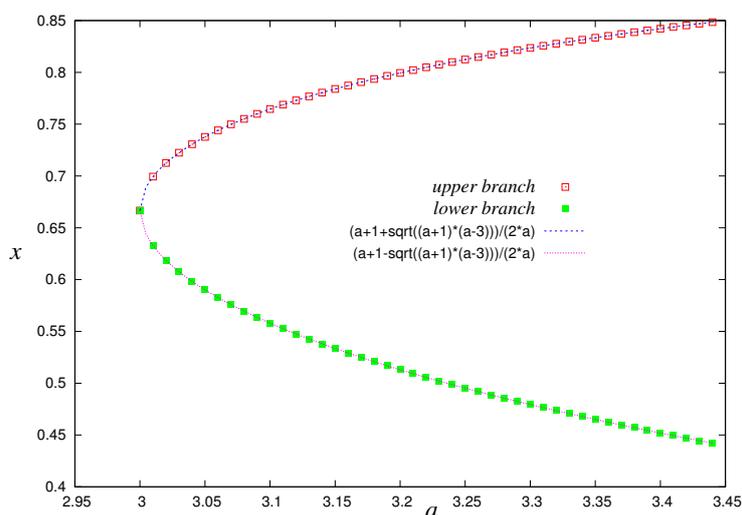


図 29: ニュートン法に基づく数値解と 2 次方程式の解公式による解析解.

9 労働市場のカオスモデル

前回までの講義では, 主にロジスティック写像:

$$x_{n+1} = ax_n(1 - x_n) \quad (133)$$

を題材にとりあげ, その振る舞いを学んできた. 既に講義でみたように, ロジスティック写像は生態系を記述する微分方程式を差分化したものであるから (第 1 回講義ノートを見よ), (それは無味乾燥な単なる漸化式にみえるが) もともと自然現象の説明を目指して提案された数理モデルである. 前回 (第 5 回) の講義では, スライドを用いて「受動歩行」と呼ばれる 2 足歩行のパターンに関し, 膝の間接角を変数 (ロジスティック写像における「変数 x 」に相当) に選んで得られる運動方程式を差分化して解くことで, 床の傾斜角 (ロジスティック写像における「パラメータ a 」に相当) の増加とともに間接角の軌道が分岐を繰り返す, やがてその力学系はカオスとなることを簡単に紹介した.

このように, 自然現象や工学的対象を何らかの近似 (仮定) のもとで「非線形方程式」を用いて表現すると, その方程式 (の振る舞い) はカオスの性質を持つことが少なくない. また, 工学部に所属する者には意外に思えるかもしれないが, 自然現象や工学的応用のみならず, 経済活動に関する社会科学的研究対象の数理モデルにおいてさえもやはりカオスが現れ, それが考える対象の重要な

一側面を説明する場合も決して少なくない。この講義「カオス・フラクタル」は情報工学コースのカリキュラムのなかにおいてはやや「異質」に思えるかもしれないが(そして学生の間では小難しく評判が悪いと伝え聞いているが)、上に述べたように、現代的問題を精密な方法で調べようとするれば非線形、とりわけカオスに関する基礎知識が必要になる場面は案外多いのである²⁶。

そこで、第6回目となる今回の講義では、ロジスティック写像を通じてカオスの基本的な性質を説明してきた通常の講義の流れから少し離れて、そうしたカオスの概念が使われる実例として「労働市場」をとりあげ、それを説明する数理モデル(方程式)が確かにカオスの性質を示すことをみていく。今まで学んだ非線形性やカオスに関するいくつかの知見がどの部分でどのように用いられるのかに着目しながら、式の一つひとつを追って行くといいであろう。

9.1 背景: 労働市場におけるフィリップス曲線

労働は人間の基本的活動であり、労働による対価として人々は貨幣や物品(賃金)を受け取り、それらの相互交換によって市場は形成される。一方、生産/サービス活動を行う企業は、自社の商品やサービス等を効率的に生み出すために労働者を必要とし、その労働の対価として労働者に賃金を支払う。従って、収益が低迷している企業は労働者の雇用を抑え(ときにリストラを行使し)、逆に、成長企業は自社グループ拡大のため、労働者雇用を促進する。社会全体の景気が悪化し、十分な雇用が確保できない状況では、失業者が増加し、また、その結果、国や地方自治体の税収も悪化する。そうした状況下にある国家(例えば、現在の日本)は金利を大幅に下げ、大量の国債を発行し、それを(多くの国内/海外投資家に)売却することで、不足した税収を国家の「借金」として補うことになり、これが国としての信用を落とすばかりか、場合によっては「国家的経営破綻」につながりかねないリスクを生んでいる。よって、明らかに「失業」と「その対策」はミクロな労働者個人の文化的生活の維持のためだけでなく、よりマクロに「地方自治体レベル」「国家レベル」においても重大な(おそらくは最重要な)問題である。従って、労働者がある種の「商品」として捉え、雇用側(企業)、非雇用側(労働者)の需要と供給のバランスを考え、その「市場」がどのように振る舞うか、政府の施すマクロな政策が、どの程度、雇用の問題に対して有効なのかを系統的に調べることは社会科学における重要な研究テーマである。

我々は人工労働市場を考えるのだが、ここで注目するマクロ変数は、**失業率と物価上昇率**であり、それら変数は**フィリップス曲線**と呼ばれる関係を満たすことが知られている。

フィリップス曲線とは1958年に英国の経済学者 Alban William Housego Phillips が実データに基づいて発見した法則に基づく曲線である。具体的には物価上昇率を縦軸に、失業率を横軸に選んだとき、これらの変量間に成立する「負の相関」を示したものである。ここで言う物価上昇率とは、消費者物価指数を表しており、消費者物価指数は、全国の世帯が購入する家計に係る財及びサービスの価格等を総合した物価の変動を時系列的に測定したものであり、家計の消費構造を一定のものに固定し、これに要する費用が物価の変動によって、どのように変化するかを指数値で示したものである。

図30は日本の1970年代から2000年代までのフィリップス曲線を10年おきのデータからプロットしたものである。この図より、全体として負の相関が見てとれる。

労働市場の振る舞いを調べるための数理モデルはこれまでも数多く提案されているが、ここでは、Neugart (2004) によって提案された、2変数 — 失業率と物価上昇率 — に関する決定論的な非線

²⁶ 実際にカリキュラムを作った人間は、このような理由で(思想のもとで)「カオス・フラクタル」を必須科目として導入したと思いたい。

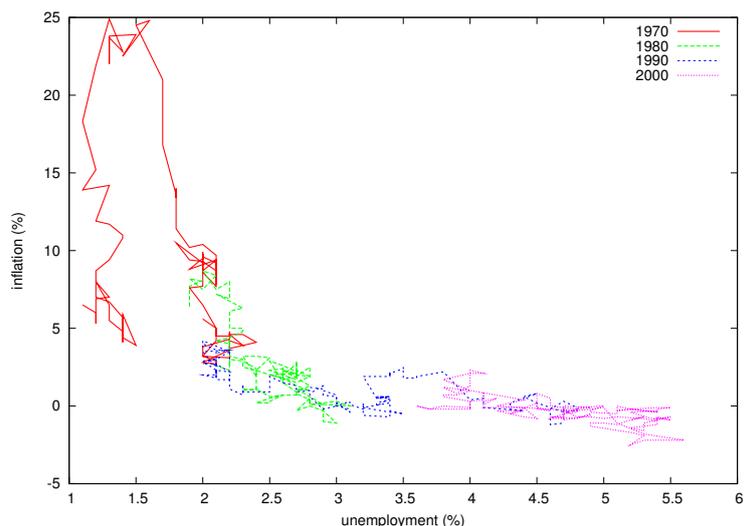


図 30: 我が国の年代別フィリップス曲線.

形写像で与えられるマクロなカオス労働市場モデルを紹介する. このモデルでは, フィリップス曲線が非線形写像のカオス・アトラクタとして説明されることになる.

9.2 Neugart モデル

Neugart モデルはマクロ変数に関する決定論的非線形写像で与えられる. 具体的には, 失業率と物価上昇率の関係をマクロに与え, その年次ごとの状態更新が非線形写像を与える. その導出はやや込み入っているが, 得られる「連立写像」はパラメータの選び方次第ではカオスの性質を持つ.

9.2.1 非線形写像の導出

まず, ステップ t での失業率²⁷ を U_t , 外的な社会要因のためそれまで定職にあった者が職を失う「離職率」を i とする. また, それまで失業中だった者が職にありつき, 「就労者組」へと流れ込む確率を就職率として o_t とする. ここに, この就職率はステップ t に依存することに注意されたい. このとき, ステップ $t+1$ での失業率 $U_{t+1} (\leq 1)$ は前ステップ t での失業率 U_t を用いて

$$U_{t+1} = U_t + i(1 - U_t) - o_t U_t \quad (134)$$

で与えられるものとする. この式の意味するところは, ステップ $t+1$ での失業率 U_{t+1} は, 前ステップの失業率 U_t にステップ t で職にありつけた労働者のために「失業組」が減る寄与: $-o_t U_t$ と, このステップで職を失った労働者: $i(1 - U_t)$ の和で与えられるという事実である.

Neugart モデルでは離職率 i は時間的に変化せず, 「外的コントロールパラメータ」として扱う. 具体的にはカオス力学系として広く知られるロジスティック写像:

$$x_{t+1} = ax_t(1 - x_t) \quad (135)$$

²⁷ 「ステップ t 」をここでは「年次」と考えるとイメージしやすいと思われる.

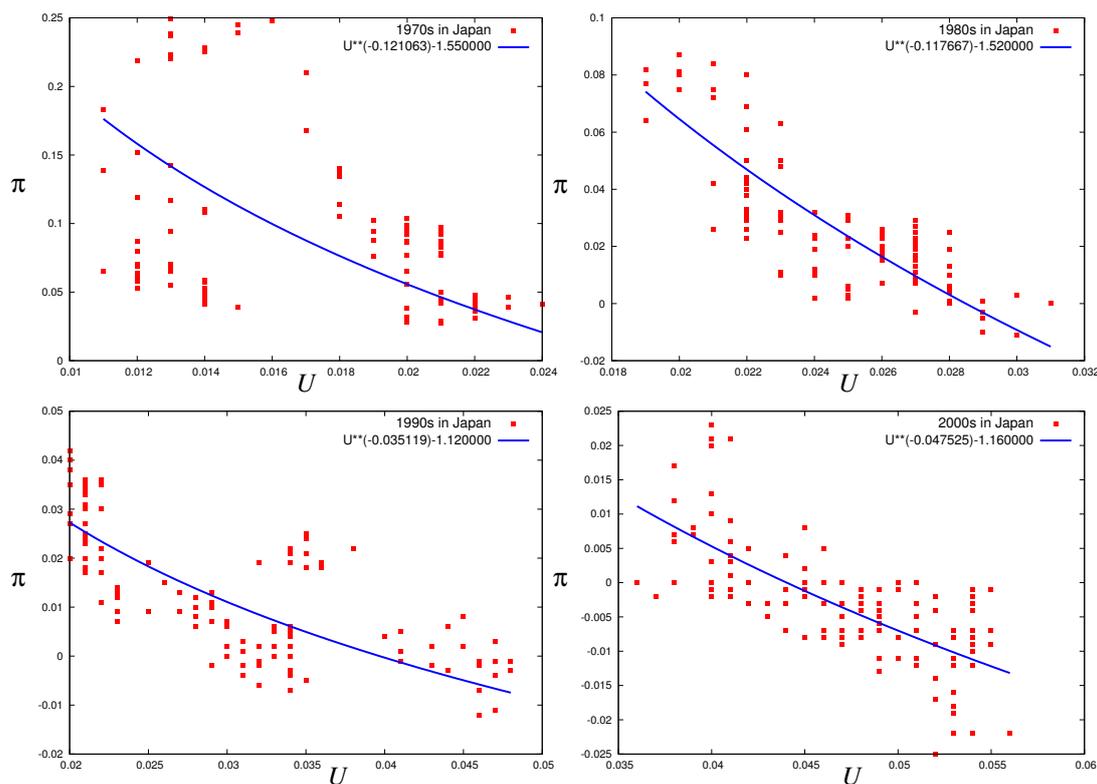


図 31: 我が国の年代別フィリップス曲線.

における分岐パラメータ a と同様の扱いとする. 後に見ることになるが, この離職率 i を変化させていくことで, システムは周期軌道からカオス軌道への分岐を見せることになる.

一方, ステップ t で職を見つける労働者の割合 o_t はステップに依存して変化するものとして扱う. 具体的にはステップ t での値が次式によって与えられるものとする.

$$o_t = \frac{J_s + J_{c,t}}{U_t + d(1 - U_t)} \tag{136}$$

$$J_{c,t} = \gamma(m - \pi_t) \tag{137}$$

ここに, (136) 式分子に現れる $J_s + J_{c,t}$ は考える社会における全求人数を与える. この 2 項のうちの第 1 項 J_s は定数であり, 時間が経過しようが不変に保たれる社会が供給する一定数の求人数を意味する. 一方, 第 2 項 $J_{c,t}$ は社会情勢等によって変化をうける一定求人数 J_s からの「偏差」であり, この時間的に変化する求人数は, Neugart モデルにおいては (137) 式のように貨幣価値の成長率 m と物価上昇率 π_t の差によって決まるものと仮定される. γ は $0 < \gamma < 1$ を満たす定数である. 従って, 貨幣価値が上昇すればするほど, 実質求人数 $J_s + J_{c,t}$ は増加し, 逆に, 物価上昇率が上がればあがるほど減少する.

分母は現在 (ステップ t) に職にありつけていない求職者数であり, 分母第 2 項 $d(1 - U_t)$ は職にありつきながら, より良い職を探している就労者の数を表し, この項と第 1 項である離職者数 U_t を合算したものが $U_t + d(1 - U_t)$ が社会全体の求職者数を与えることになる. 従って, d は所謂 *On the job searching* の割合を表すパラメータと考えることもできる.

以上を総合すると, (136) 式はステップ t で職にありつける人の割合は求人数に比例し, 求職者数

に反比例するという事実を表すことになる。

次に、期待物価上昇率 $\pi_{e,t}$ 、交渉賃金 $w_{b,t}$ 、(実質)賃金 w_p を定義する。期待物価上昇率とは、将来に予想される物価上昇率のことである。Neugart モデルでは、ステップ $t+1$ における期待物価上昇率が、ステップ t における期待物価上昇率と物価上昇率 π_t との間の、パラメータ a ($0 \leq a \leq 1$) を用いた加重平均として、次式で定義される。

$$\pi_{e,t+1} = a\pi_t + (1-a)\pi_{e,t} \quad (138)$$

また、物価上昇率 π_t を定義するために、以下で、労働者の賃金 w_p 、ステップ t での交渉賃金 $w_{b,t}$ を以下に説明するように定義する。

まずは賃金を次で定義する。

$$w_p = (1-\mu)y \quad (139)$$

ここに、 y は労働者一人当たりの生産量であり、 μ は「需要弾力度」の逆数を表す。需要弾力度とは、需要の増加にともなって価格が下落する割合を表し、この価格の下落率が小さいほど需要弾力度が大きいものとして定義される (図 32 参照)。従って、労働者の実質賃金は、需要の弾力性が大きけれ

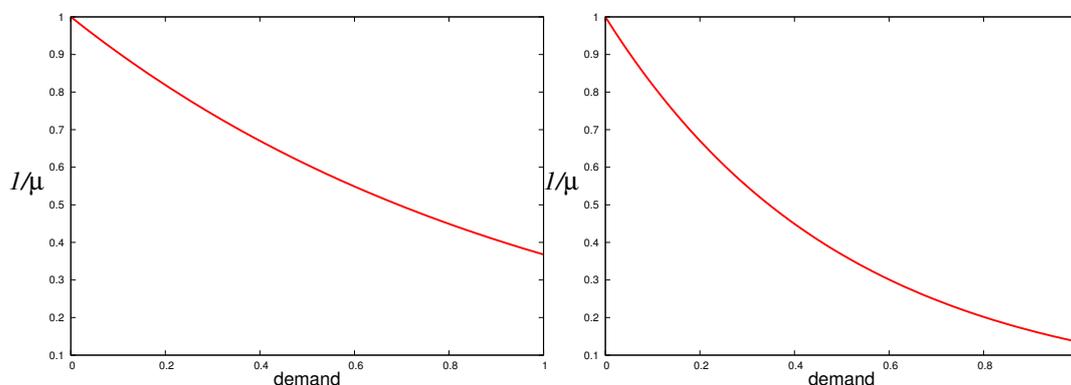


図 32: 重要弾力度 $1/\mu$ の需要 (demand) 依存性. 左図が弾力性が高く、右図が低い。

ば大きいほど高くなる。これは例えば、いわゆる「贅沢品」の需要弾力性は大きい、日用品の弾力性は低いということからわかるように、一人あたりの生産量 (y) が同じであれば、より高価なものを生産すればするほど、その労働者の実質賃金は上昇することを意味している。これは理にかなった設定である。簡単のため、以下では $y = 1$ として議論を進める。

多くの就労者は労働組合に加入しており、労働組合は次年度の賃金²⁸、を引き上げるために交渉を行う。この交渉が成功すれば賃金は上昇する。従って、次にステップ t での交渉賃金を

$$w_{b,t} = 1 - (1-b)U_t \quad (140)$$

で定義する。ここに、パラメータ b は $0 < b < 1$ の値をとることから、交渉賃金は離職率が高ければ高いほど減少し²⁹、逆に離職率ゼロの極限、つまり、全ての労働者が就労している状況下においては、一人あたりの生産量、すなわち、 $y = 1$ に一致する³⁰。これらを用いて、ステップ t での物価上昇

²⁸ 多くの場合には、給与の基本的部分(「ベース」)の引き上げ、つまり、ベースアップ(所謂「ベア」)。

²⁹ 労働組合の構成員が減少するため、賃金交渉が十分適切に行われないと考える。

³⁰ 労働組合の構成員による賃金交渉の結果、就労者の意見が取り入れられ、生産量に見合った賃金が得られると考える。

率を次式で定義する.

$$\pi_t = \frac{1}{\delta} \left(\pi_{e,t} + \frac{w_{b,t} - w_p}{w_p} \right) = \frac{1}{\delta} \left(\pi_{e,t} + \frac{\mu - (1-b)U_t}{1-\mu} \right) \quad (141)$$

ここで, 上式の説明のため, ステップ t での賃金上昇率を

$$\Delta w_t \equiv \frac{w_{b,t} - w_p}{w_p} \quad (142)$$

で定義しよう. つまり, Δw_t はある時刻における賃金 w_p に対し, 時刻 t における交渉賃金 $w_{b,t}$ がどの程度まで上昇したかの割合を w_p を基準に測ったものである.

従って, (141) 式の意味は, ステップ t での賃金上昇率が正 ($\Delta w_t > 0$) ならば, ステップ t での物価上昇率は, 期待される物価上昇率 $\pi_{e,t}$ より大きくなり, 逆に, 賃金上昇率が負 ($\Delta w_t < 0$) ならば, 物価上昇率は, 期待物価上昇率よりも小さく押さえられる. このことは, 賃金が労使交渉によって上昇し, 消費者でもある就労者にお金がまわれば回るほど, 人々の購買意欲が上がり, 多くの消費材の需要が増加することで, それらの価格も高騰していくことを考えると, 合理的な設定であると思われる.

また, (141) 式における δ は, (141) 式において $w_p = w_{b,t}$ とおけば明らかなように, 物価上昇率と期待上昇率の間のスケール変換 (大きさの調整) を表すパラメータである.

以上から, 物価上昇率 π_t と離職率 U_t の間の連立非線形写像を得ることができる. 具体的には, (134) 式に (136),(137) 式を (141) 式に (138) 式を代入すると

$$U_{t+1} = U_t + i(1 - U_t) - U_t \frac{J_s + \gamma(m - \pi_t)}{U_t + d(1 - U_t)} \quad (143)$$

$$\begin{aligned} \pi_{t+1} &= \frac{1}{\delta} \left(\frac{\mu}{1-\mu} + a\pi_t + (1-a) \left(\delta\pi_t - \frac{\mu - (1-b)U_t}{1-\mu} \right) \right) \\ &\quad - \frac{1}{\delta} \left(\frac{1-b}{1-\mu} \left(U_t + i(1 - U_t) - U_t \frac{J_s + \gamma(m - \pi_t)}{U_t + d(1 - U_t)} \right) \right) \end{aligned} \quad (144)$$

となる. つまり, U_t と π_t は上記の非線形写像に従って, ともに決定論的に状態更新していくことになる.

ところで, 上記の非線形写像 (143)(144) の固定点 $U^* = U_{t+1} = U_t$, $\pi^* = \pi_{t+1} = \pi_t$ は直ちに

$$U^* = \frac{\mu - m(\delta - 1)(1 - \mu)}{1 - b} \quad (145)$$

$$\pi^* = m \quad (146)$$

のように求めることができる. ここでは, 物価上昇率の固定点 π^* が, 貨幣価値の上昇率 m に等しい, すなわち, $\pi^* = m$ と仮定すると (つまり, 固定点の条件のうちで (146) のみを仮定すると), (143) 式で, $U^* = U_{t+1} = U_t$, $\pi^* = \pi_{t+1} = \pi_t = m$ とおくことで

$$U^* + i(1 - U^*) - U^* \frac{J_s}{U^* + d(1 - U^*)} = U^* \quad (147)$$

すなわち

$$J_s = \frac{i(1 - U^*)(U^* + d(1 - U^*))}{U^*} \quad (148)$$

として社会が供給する一定数の求人数 J_s が定まる. 以下の数値計算では, 上記 (148) の条件下で非線形写像 (143)(144) の振る舞いを調べて行くことにする.

9.2.2 非線形写像の分岐構造とカオス

前出の非線形写像 (143)(144) は分岐パラメータ i を変化させていくことで、やがてカオスの振る舞いを見せる. 図 33 に分岐パラメータを $i = 0.18$ と選んだ場合の失業率 U_t と物価上昇率 π_t の時

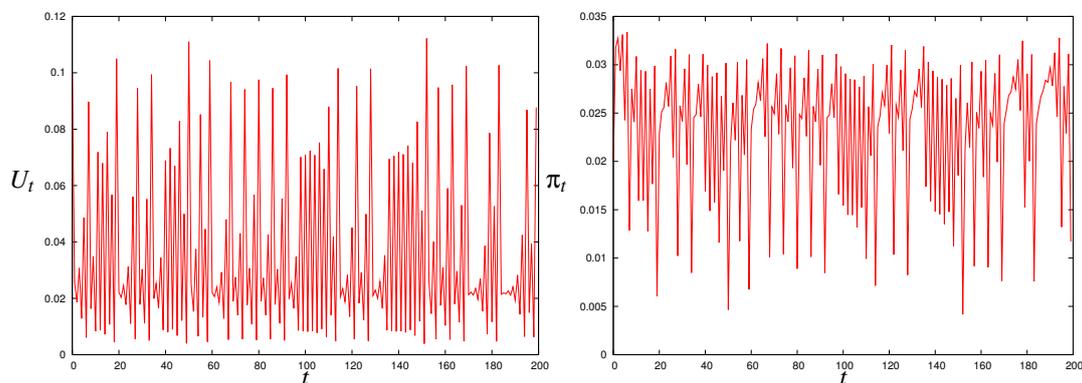


図 33: 分岐パラメータを $i = 0.18$ と選んだ場合の失業率 U (左) と物価上昇率 (右) の時間変化. 各種パラメータは, $d = 0.01, b = 0.5, \mu = 0.04, \gamma = 0.5, \delta = 2, a = 0.5, m = 0.03$ と選んだ.

間変化を $t = 200$ 程度までプロットした. この図から明らかにそれぞれの軌道は固定点へ収束することなく, また, 周期軌道をとることなく, 「複雑な振る舞い」をみせることがわかる. そこで, 分岐パラメータを $i = 0.1$ から $i = 0.185$ 程度まで増加させた場合の U_t, π_t のとりうる値をプロットし, この非線形写像の分岐構造を調べてみる. 結果を 図 34 に載せる. ここに, i 以外の各種パラメータは, $d = 0.01, b = 0.5, \mu = 0.04, \gamma = 0.5, \delta = 2, a = 0.5, m = 0.03$ と設定した. この図から, $i \simeq 0.13199$ を境に分岐し, パラメータ i が増加していくと分岐が増え, カオス的な振舞いを見せることがわかる. このカオスを発生させる分岐パラメータとしては, m や d でも同様であり, これらの

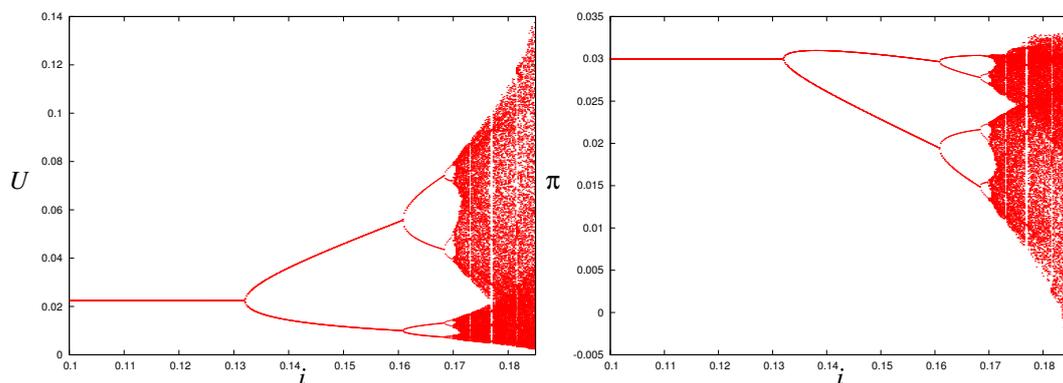


図 34: パラメータ i を変化させた場合の失業率 U (左) と物価上昇率 (右) の分岐構造. $i = 0.18$ 程度でカオスが生じる. 各種パラメータは, $d = 0.01, b = 0.5, \mu = 0.04, \gamma = 0.5, \delta = 2, a = 0.5, m = 0.03$ と選んだ.

分岐パラメータが, ある一定の値を超えた時, パラメータと失業率 U や物価上昇率 π の関係はカオスになる.

9.2.3 カオス・アトラクタとしてのフィリップス曲線

分岐パラメータ i (離職率) をカオスが出現するような値を取った時の失業率 U と物価上昇率 π の関係は図 35 のようになる. 図 35 から, 失業率 U と物価上昇率 π が負の関係を表すフィリップス曲線

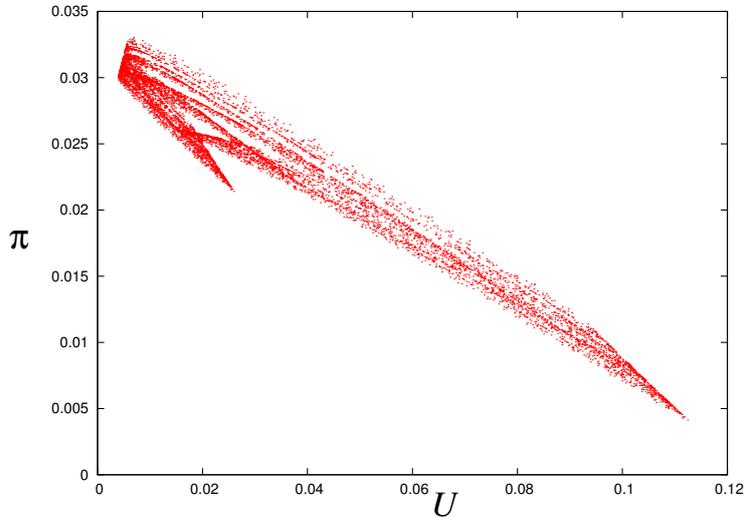


図 35: 失業率 U と物価上昇率 π の関係 (フィリップス曲線). 各種パラメータは, $d = 0.01, b = 0.5, \mu = 0.04, \gamma = 0.5, \delta = 2, a = 0.5, i = 0.18, m = 0.03$ と選んだ.

線が現れていることがわかる.

また, 図 36 には, 失業率, および, 物価上昇率の時間発展を横軸を時刻 t , 縦軸を $t+1$ での値にそれぞれプロットした. スケールを変えてプロットを眺めて行くと, 自己相似的な構造が見える. 従って, Neugart モデルでは, このような「カオス・アトラクタ」がフィリップス曲線を与えることがわかる.

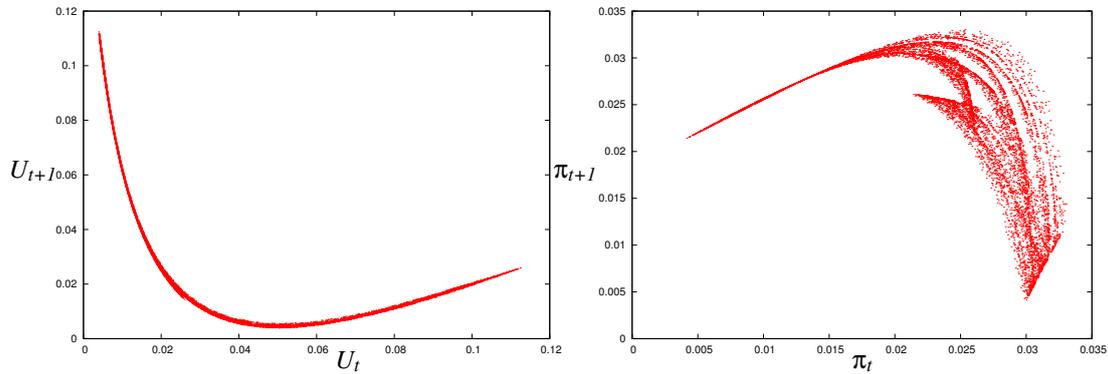


図 36: 分岐パラメータを $i = 0.18$ と選んだ場合の失業率 U (左) と物価上昇率 (右) の時間変化を横軸を時刻 t , 縦軸を $t+1$ での値にそれぞれプロットしたもの. 各種パラメータは, $d = 0.01, b = 0.5, \mu = 0.04, \gamma = 0.5, \delta = 2, a = 0.5, m = 0.03$ と選んだ.

9.2.4 失業率と物価上昇率の稠密性と不変測度

U_t, π_t で記述される力学系はカオスであるから、有界区間の全ての微小領域をそれらの軌道が埋め尽くし、結果として、これらの変数に関する不変測度 (密度関数) $P(U), P(\pi)$ が得られる。その結果を図 37 に示す。従って、Neugart モデルは決定論的な状態更新の非線形写像によって失業率と物

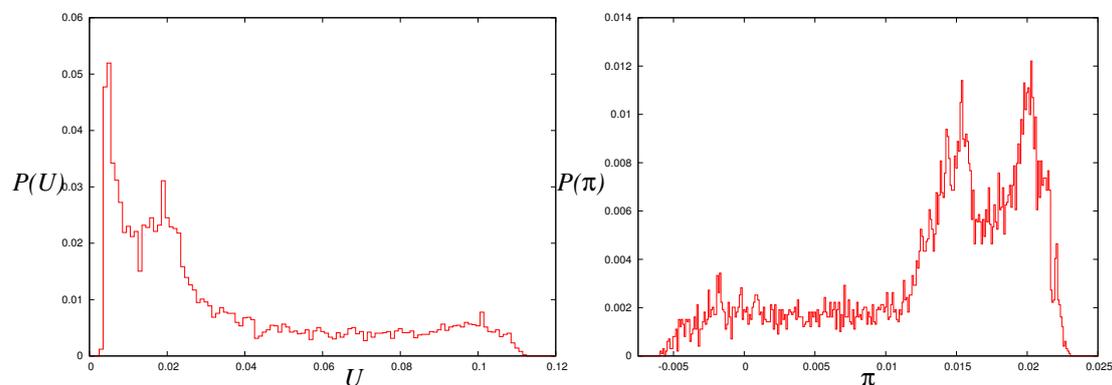


図 37: 失業率と物価上昇率の不変測度, $P(U)$ (左), および $P(\pi)$ (右). $d = 0.01, b = 0.5, \mu = 0.04, \gamma = 0.5, \delta = 2, a = 0.5, m = 0.03$ と選んだ.

価上昇率が決まるが、その写像からのサンプリングを行うと、それは上記のような不変測度を持ち、ある種のバイアスのかかった「乱数生成装置」のような性質も兼ね備えていることがわかる。しかし、重要なのは、各ステップでは決定論的な状態更新式に従って、失業率と物価上昇率が変化する点である。

(参考): 以上の議論および、その背景は混沌系工学研究室を昨年度終了した日野光さんの修士論文に詳しい。また、この Neugart モデルを踏まえて、我々の研究グループがどのような課題に取り組んだのかも書いてあります。日野さんの修士論文は北大の図書館が運営している HUSCAP から誰でもダウンロードできるようになっています。

<http://eprints.lib.hokudai.ac.jp/dspace/handle/2115/43998>

内容に関してももちろんであるが、修士論文 (あるいは卒業論文) はどのような形で書かれるものなのか (つまり、どんな「感じ」なのか)、今のうちから参考程度に見ておくと良いと思います。

また、この研究課題の進捗状況は陳鶴さんの HP:

<http://chaosweb.complex.eng.hokudai.ac.jp/~chen/>

から参照できます。

(連絡): 次週 (6/7) は担当者 (井上) が学会で出張のため休講とします。ちなみに、その学会では、今回この講義で紹介した労働市場に関する話をさせていただきます。

今回のレポート課題はありません。

10 数値計算の準備: 常微分方程式の数値解法

今までこの講義では1次元写像の振る舞いを詳しくみてきた。例えば、ロジスティック写像は次のロジスティック方程式:

$$\frac{dN}{dt} = \lambda N(1 - \mu N) \quad (149)$$

を差分化した方程式であり、本来解くべき方程式の「近似」である。しかし、現実の自然現象は運動方程式など微分方程式で記述されるものが多く、従って、これらの現象を正確に調べようとすれば、差分化された近似式ではなく、微分方程式そのものを扱わなければならない。いくつかの常微分方程式は解析的に解くことができるが、非線形項が入り、しかも、変数が複数絡み合った連立微分方程式は多くの場合に手で解くことはできない。このような手で解くことのできない方程式の多くはカオスなどの複雑で多彩な挙動を示し、従って、そのような現象の理解を深めることを目的の一つとする本講義では、そのような方程式を解くことを避けて通ることはできない(逆に言えば、手できれいに解ける方程式は本講義の対象とならない)。

そこで、今回はやや本題を外れることになるが、複雑な挙動を示す常微分方程式を解析するためのほぼ唯一の手段になる数値計算技法の典型例についてみていく(一部の人間たちにとっては復習になるかもしれないが)。現在、微分方程式を数値的に解くための様々なアプリケーションソフトやパッケージが利用できるが、はじめからこれらを「ブラックボックス」として用いるのではなく、一度はその原理を学んでおくことは重要である。

具体的に我々が解こうとする方程式の典型例は次のようなものである。

$$\frac{dx}{dt} = -y - z \quad (150)$$

$$\frac{dy}{dt} = x + ay \quad (151)$$

$$\frac{dz}{dt} = b + z(x - c) \quad (152)$$

この連立微分方程式は**レスラー方程式**と呼ばれる。ここに現れるパラメータ a, b, c を適切に選んだ際、我々が1次元写像でみたようなカオスが生じるかどうか、どのような力学的性質を持つかなどを調べることがここからの目標である。今回の講義ではそのための準備を行う。

10.1 テーラー展開による漸化式の導出

話を簡単にするために次の1変数の微分方程式:

$$\frac{dy}{dt} = f(t, y) \quad (153)$$

を考えよう。ただし、関数 f は微分可能な関数であるとする。ここで h を十分小さい量とし

$$t_n = t_0 + nh, \quad n = 1, 2, 3, \dots \quad (154)$$

で「時間」 t を離散化すると (t_0 は時間原点), $t_{n+1} = t_0 + (n+1)h$ であるから, $t_{n+1} - t_n = h \ll 1$ となることに注意しよう (あるいは $t_{n+1} = t_n + h$)。このとき、小さな量 h に関するテーラー展開を思い出すと

$$\begin{aligned} y(t_{n+1}) &= y(t_n + h) = y(t_n) + \left. \frac{dy}{dt} \right|_{t_n, y_n} h + \mathcal{O}(h^2) \\ &= y(t_n) + f(t_n, y_n)h + \mathcal{O}(h^2) \end{aligned} \quad (155)$$

となる. 従って, $y(t_n) = y_n, y(t_{n+1}) = y_{n+1}$ と略記することを約束すれば

$$y_{n+1} = y_n + f(t_n, y_n)h + \mathcal{O}(h^2) \quad (156)$$

$$t_n = t_0 + nh \quad (157)$$

が得られる. この方程式 (156) を漸化式として解き, 各 n で (157) 式で与えられる t_n を y_n に対する時刻とみなすことで微分方程式を数値的に解く方法を **オイラー法** と呼ぶ. 上記から明らかなように, 各ステップでの所謂「局所誤差」は h^2 のオーダーである ($\mathcal{O}(h^2)$ と表記した部分).

10.2 ルンゲ・クッタ法: 基本的アイデア

上記のテーラー展開で微分係数 $dy/dt = f(t, y)$ を評価する点は (t_n, y_n) であったが, これを (t_n, y_n) と (t_{n+1}, y_{n+1}) の中点である $(t_{n+1/2}, y_{n+1/2})$ に選んだ場合, 上記のテーラー展開のプロセスと結果として得られる漸化式はそれぞれの区間でどうなるかを考えてみよう.

まず, (154) 式から $t_{n+1/2} = t_0 + (n + 1/2)h$ であるから

$$t_{n+1} = t_{n+1/2} + h/2, \quad t_n = t_{n+1/2} - h/2 \quad (158)$$

であることに注意して

$$\begin{aligned} y_{n+1} = y(t_{n+1}) &= y(t_{n+1/2} + h/2) \\ &= y_{n+1/2} + f(t_{n+1/2}, y_{n+1/2}) \left(\frac{h}{2}\right) + \frac{1}{2} f'(t_{n+1/2}, y_{n+1/2}) \left(\frac{h}{2}\right)^2 + \mathcal{O}(h^3) \end{aligned} \quad (159)$$

および

$$\begin{aligned} y_n = y(t_n) &= y(t_{n+1/2} - h/2) \\ &= y_{n+1/2} - f(t_{n+1/2}, y_{n+1/2}) \left(\frac{h}{2}\right) + \frac{1}{2} f'(t_{n+1/2}, y_{n+1/2}) \left(\frac{h}{2}\right)^2 + \mathcal{O}(h^3) \end{aligned} \quad (160)$$

が得られる. ここで, $f' \equiv df/dt$ と略記したことに注意されたい. 従って, これらの方程式 (159)(160) を辺々引くと

$$y_{n+1} - y_n = f(t_{n+1/2}, y_{n+1/2})h + \mathcal{O}(h^3) \quad (161)$$

となり

$$y_{n+1/2} = y(t_{n+1/2}) = y(t_n + h/2) = y(t_n) + \frac{h}{2} f(t_n, y_n) = y_n + \frac{k_1}{2} \quad (162)$$

であるから, 問題の微分方程式 (153) を

$$y_{n+1} = y_n + k_2 + \mathcal{O}(h^3) \quad (163)$$

$$k_1 = hf(t_n, y_n) \quad (164)$$

$$k_2 = hf(t_n + h/2, y_n + k_1/2) \quad (165)$$

$$t_n = t_0 + nh \quad (166)$$

の漸化式で書き直すことで, ステップごとの局所誤差をオイラー法での $\mathcal{O}(h^2)$ から $\mathcal{O}(h^3)$ へまで低減させることができることがわかる³¹. これら (163)(164)(165)(166) 式を 2 次のルンゲ・クッタ法と呼ぶ.

このような方法でうまく中継点を選び, 上記の手続きを繰り返すことにより, 局所誤差を系統的に減らしていくことができることは容易に推測できるであろう. 導出が煩雑なので詳細は省略するが, 局所誤差が $\mathcal{O}(h^5)$ まで低減できるものとして, 次の 4 次のルンゲ・クッタ法が知られており, 多くの場面で用いられている (本講義でも数値計算では専らこの 4 次のルンゲ・クッタ法を用いる).

$$y_{n+1} = y_n + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} + \mathcal{O}(h^5) \quad (167)$$

$$k_1 = hf(t_n, y_n) \quad (168)$$

$$k_2 = hf(t_n + h/2, y_n + k_1/2) \quad (169)$$

$$k_3 = hf(t_n + h/2, y_n + k_2/2) \quad (170)$$

$$k_4 = hf(t_n + h, y_n + k_3) \quad (171)$$

$$t_n = t_0 + nh \quad (172)$$

次で具体的なコーディングについてみていこう.

10.2.1 適用例 1: ロジスティック方程式

以上が一般論であるが, 感じをつかむためにいくつかの具体的な方程式への適用例を見ておこう.

まずは最も簡単な例の一つとして, 上記の 4 次のルンゲ・クッタ法 (167)-(172) でロジスティック方程式:

$$\frac{dN}{dt} = \lambda N(1 - \mu N) \quad (173)$$

を数値的に解いてみる. その結果を図 38 に載せよう. この図より, 解析解と極めて良好な一致を見せていることがわかる³². また, そのプログラムを載せる.

/* ルンゲ・クッタ法のロジスティック方程式への適用例 */

```
#include <stdio.h>
#include <math.h>
#define h 0.01
#define mu 0.1
#define lambda 0.5
double func(y)
double y;
{return (lambda*y*(1.0-mu*y));} /* 関数 f の定義 */
main()
{
    FILE *fpr;
    int i,imax=1000;
    double y,k1,k2,k3,k4,k;
```

³¹ h は十分に小さな量であったことを思い出す. 例えば $h = 0.01$ などとして, $\mathcal{O}(h^2)$ と $\mathcal{O}(h^3)$ を比較してみるとよい.

³² 大域誤差 (真の解 $y(t)$ と y_n の差) についての評価は後の演習などで触れたいと思う.

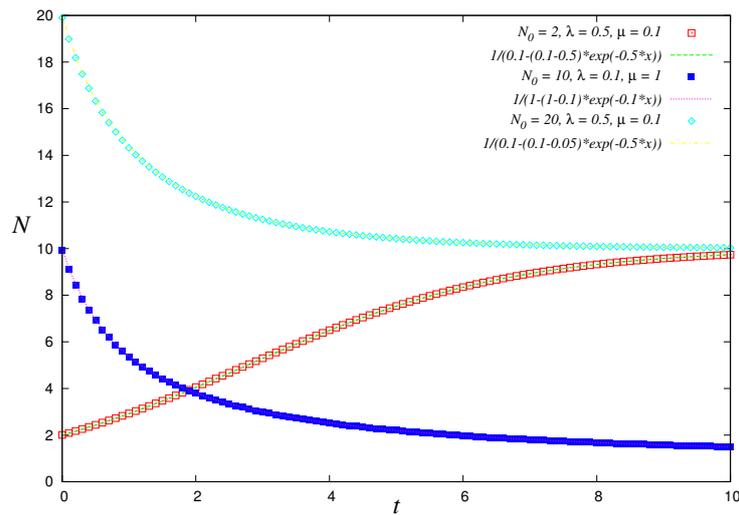


図 38: ロジスティック方程式の 4 次のルンゲ・クッタ法に基づく数値解と解析解.

```

if((fpr = fopen("rk4.dat", "wt")) !=NULL){
    for(i = 0,y=20.0; i <= imax; i++){
        k1 = h*func(y);
        k2 = h*func(y+0.5*k1);
        k3 = h*func(y+0.5*k2);
        k4 = h*func(y+k3);
        k = (k1+2.0*k2+2.0*k3+k4)/6.0;
        y = y + k;
        fprintf(fpr,"%lf %lf\n",i*h,y);
    }
}
fclose(fpr);
}

```

10.2.2 適用例 2: レスラー方程式

変数が 3 つに増えたレスラー方程式 (150)(151)(152) も上記 1 変数の場合のプログラムを拡張することで容易に解くことができる. 参考までにプログラミングのソースコードを載せておく.

```

/* ルンゲ・クッタ法のレスラー方程式への適用例 */
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define h 0.01
#define a 0.5
#define b 0.5

```

```
#define c 5.7
double func1(x,y,z) /* 関数その 1 */
double x;
double y;
double z;
{return -(y+z);}
double func2(x,y,z) /* 関数その 2 */
double x;
double y;
double z;
{return (x+a*y);}
double func3(x,y,z) /* 関数その 3 */
double x;
double y;
double z;
{return (b+z*(x-c));}

main()
{
    FILE *fpr;
    int i,imax=50000;
    double x,k1x,k2x,k3x,k4x,kx,y,k1y,k2y,k3y,k4y,ky,z,k1z,k2z,k3z,k4z,kz;
    if((fpr = fopen("rossler.dat", "wt")) !=NULL){
        for(i = 0,x=-0.01,y=-0.01,z=0.01; i <= imax; i++){
            k1x = h*func1(x,y,z);
            k1y = h*func2(x,y,z);
            k1z = h*func3(x,y,z);
            k2x = h*func1(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
            k2y = h*func2(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
            k2z = h*func2(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
            k3x = h*func1(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
            k3y = h*func2(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
            k3z = h*func3(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
            k4x = h*func1(x+k3x,y+k3y,z+k3z);
            k4y = h*func2(x+k3x,y+k3y,z+k3z);
            k4z = h*func3(x+k3x,y+k3y,z+k3z);
            kx = (k1x+2.0*k2x+2.0*k3x+k4x)/6.0;
            ky = (k1y+2.0*k2y+2.0*k3y+k4y)/6.0;
            kz = (k1z+2.0*k2z+2.0*k3z+k4z)/6.0;
            x = x + kx;
            y = y + ky;
            z = z + kz;
        }
        fprintf(fpr,"%lf %lf %lf %lf\n",i*h,x,y,z);
    }
}
```

```

}
}
fclose(fpr);
}

```

上記のプログラムを用いて、まずは、3つの変数 x, y, z の時間変化をプロットすると図 39 のようになる。この図 39 より、1 次元写像に見たような複雑な挙動を x, y, z のそれぞれが示すことがわかる。

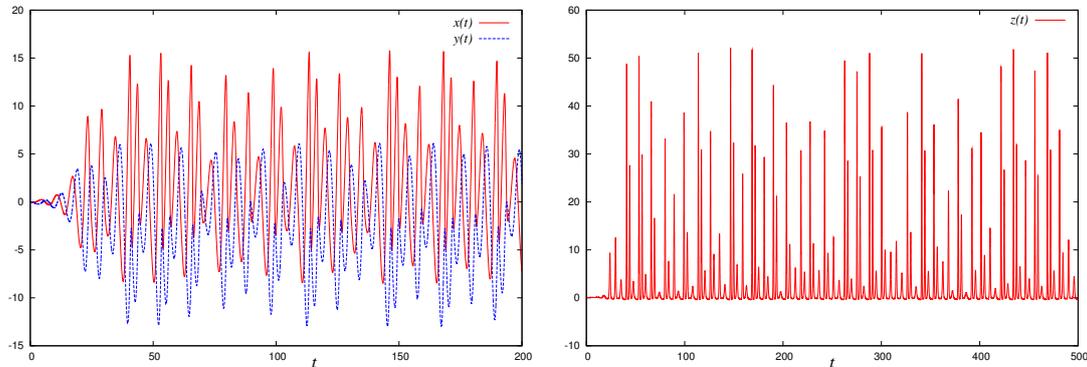


図 39: レスラー方程式の 4 次のルンゲ・クッタ法に基づく数値解. $x(t), y(t)$ および $z(t)$ の振る舞い.

この図 39 を見方を変えて、 xyz の 3 次元空間内にその軌道をプロットすると興味深い。これは上記プログラムでファイルに書き出されたデータの第 2 軸を x 軸、第 3 軸を y 軸、第 4 軸を z 軸に選んだプロットに相当する。

ちなみに、「計算機プログラミング I・同演習」や実験などで `gnuplot` を学んだものは `gnuplot` を起動したのちに

```

gnuplot> set parametric
gnuplot> splot 'lossler.dat' u 2:3:4 w line

```

とすればよい (先頭の `gnuplot>` は `gnuplot` のカーソルであることに注意)。このようにして 3 次元プロットした結果を図 40 に載せる³³。軌道の詳細を調べるため、図 40 の 3 次元プロットを x - y 平面、 x - z 平面に射影して再プロットしたものが図 41 である。

以上で本講義の [カオス編] 後半で必要な数値計算上の道具立ては全て揃ったことになる。これらの図 40,41 の意味するところは次回以降に詳しくみていくことにしよう。

レポート課題 6

³³ ここで出てきた `gnuplot` の使用法などは演習の時間に取り上げたいと思う。

$$a = b = 0.5, c = 5.7 \text{ ———}$$

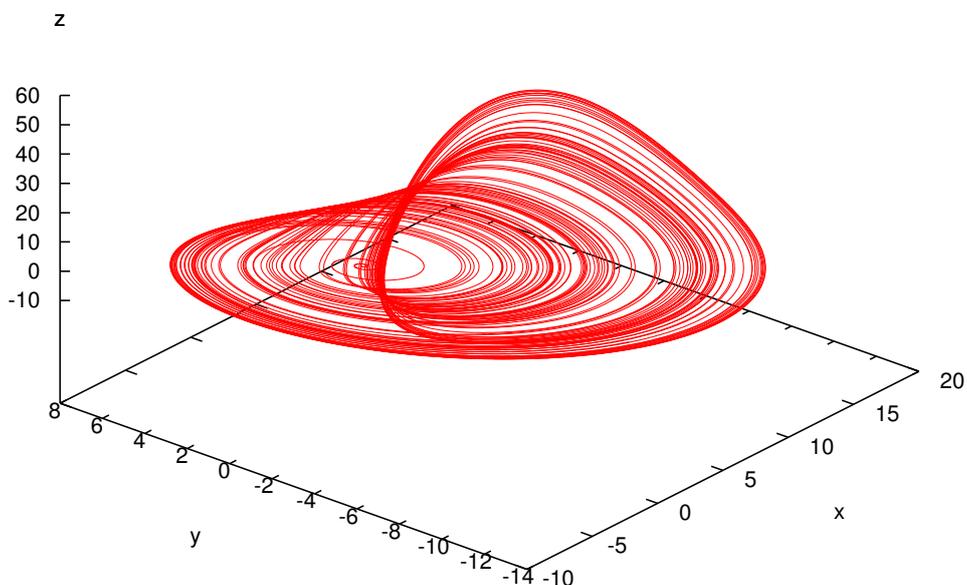


図 40: レスラー方程式の数値解から得られる軌道の様子. パラメータは $a = b = 0.5, c = 5.7$ に選んである.

次の x に関する 2 階の常微分方程式:

$$\frac{d^2x}{dt^2} - \epsilon(1-x^2)\frac{dx}{dt} + x = 0 \quad (174)$$

は速度を $dx/dt = v$ で定義すれば

$$\frac{dv}{dt} = \epsilon(1-x^2)v - x \quad (175)$$

$$\frac{dx}{dt} = v \quad (176)$$

と x, v に関する 1 階の連立微分方程式に書き直すことができ³⁴, これは今回学んだルンゲ・クッタ法を用いて解くことができる. そのプログラムを書き (ソースコードの提出), いくつかの ϵ の値, 初期条件に対する x, v の振る舞いをプロットせよ (gnuplot などの作図ツールを知らないものはソースコードのみでよい).

(連絡): 次週 (6/21) は担当者 (井上) が午後 3 時より札幌北会場 WING 進学相談会 (ダイヤ書房) シャトレーゼガトーキングダムサッポロブースにて, 「北大工学部進学相談会」を実施するため, 通

³⁴ 多くの運動方程式はこの手の書き換えができることに注意しよう.

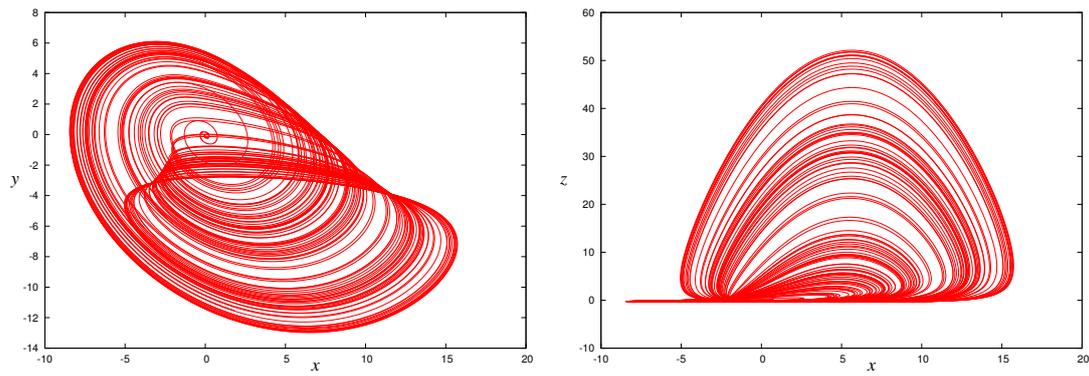


図 41: 図 40 の 3 次元プロットを x - y 平面, x - z 平面に射影してプロットしたもの

常より早めに終わる可能性があります。この講義で習得すべき内容が終わらない可能性が出てきた場合, 1 回程度の補講を行う場合があることを了解ください。

課題 6 の解答例

$$\frac{dv}{dt} = \epsilon(1-x^2)v - x \quad (177)$$

$$\frac{dx}{dt} = v \quad (178)$$

の振る舞いを数値的に調べてみる. まず, パラメータ ϵ がゼロの場合, 方程式は単振動 (調和振動子) を表す. 横軸を x , 縦軸を v とした場合の (位相空間内の) 軌道は (177) 式の両辺に $v = dx/dt$ をかけるとエネルギーの時間変化についての関係式がでるので, 実行すると

$$v \frac{dv}{dt} = -x \frac{dx}{dt} \quad (179)$$

すなわち

$$\frac{d}{dt} \left(\frac{v^2}{2} + \frac{x^2}{2} \right) = 0 \quad (180)$$

となり, E を初期条件から決まる一定値とすると

$$v^2 + x^2 = (\sqrt{2E})^2 \quad (181)$$

となり, 半径が $\sqrt{2E} = \sqrt{v_0^2 + x_0^2}$ の円となる (x_0, v_0 は $t = 0$ のときの位置と速度). 実際に $\epsilon = 0$ の場合に (177)(178) 式を 4 次のルンゲ・クッタ法で数値的に解き, 軌道 (x, v) をプロットすると図 42 のようになる. これを計算するためのソースコードを次に載せる.

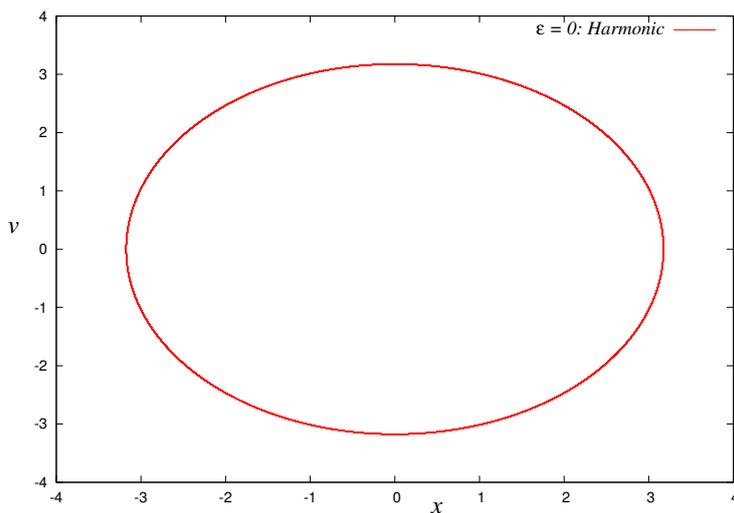


図 42: $\epsilon = 0$ の場合の軌道. 半径を $\sqrt{2E} = \sqrt{v_0^2 + x_0^2} = \sqrt{1^2 + 3^2} = \sqrt{10} = 3.16$ とした円になる. gnuplot によってプロットした図は縦軸と横軸のスケールが違っていていることに注意.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define h 0.01
```

```
#define epsilon 0.0
double func1(x,y)
double x;
double y;
{return (y);}
double func2(x,y)
double x;
double y;
{return (epsilon*(1.0-x*x)*y-x);}
main()
{
    FILE *fpr;
    int i,imax=50000;
    double x,k1x,k2x,k3x,k4x,kx,y,k1y,k2y,k3y,k4y,ky,z,k1z,k2z,k3z,k4z,kz;
    if((fpr = fopen("test0.dat", "wt")) !=NULL){
        for(i = 0,x=1.01,y=3.01; i <= imax; i++){
            k1x = h*func1(x,y);
            k1y = h*func2(x,y);
            k2x = h*func1(x+0.5*k1x,y+0.5*k1y);
            k2y = h*func2(x+0.5*k1x,y+0.5*k1y);
            k3x = h*func1(x+0.5*k2x,y+0.5*k2y);
            k3y = h*func2(x+0.5*k2x,y+0.5*k2y);
            k4x = h*func1(x+k3x,y+k3y);
            k4y = h*func2(x+k3x,y+k3y);
            kx = (k1x+2.0*k2x+2.0*k3x+k4x)/6.0;
            ky = (k1y+2.0*k2y+2.0*k3y+k4y)/6.0;
            x = x + kx;
            y = y + ky;
            fprintf(fpr,"%lf %lf %lf\n",i*h,x,y);
        }
    }
    fclose(fpr);
}
```

いくつかの有限の ϵ に対する位相空間内の軌道を求めてみると図のようになる。図 43 (上, および左下) は $\epsilon = 0.01$, および $\epsilon = 0.5, 1.5$ の場合の軌道をそれぞれプロットした。また, 右下は $\epsilon = 0.5$ の場合の位置 x , 速度 v の時間変化である。これらの図より, 時間の経過とともに, 軌道はある閉曲線に収束することがわかる。このような軌道の収束する点, もしくは曲線を**アトラクタ**と呼ぶ。

11 非線形力学系とカオス

今回の講義では, 前回の講義で学んだレスラー方程式の振る舞いを方程式の形から定性的に説明するとともに, アトラクタを可視化するためのポアンカレ断面について詳しくみていく。

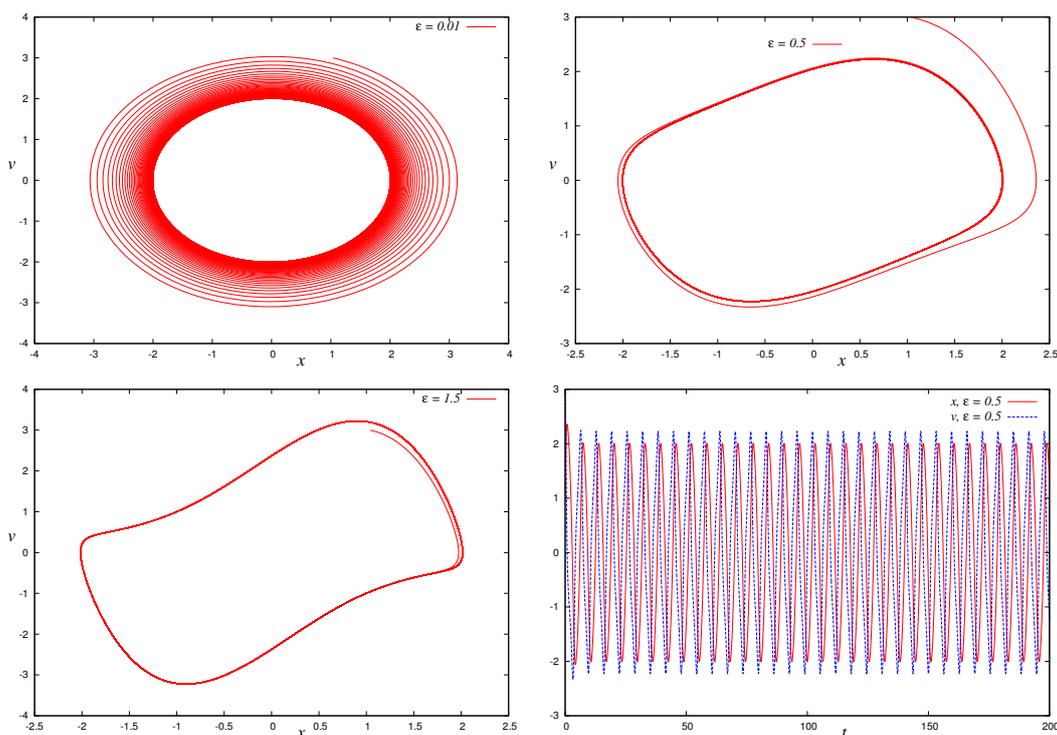


図 43: 有限の ϵ の場合のアトラクタの様子と $\epsilon = 0.5$ の場合の時間変化 (右).

11.1 レスラー方程式とその振る舞い

前回みたように, レスラー方程式:

$$\frac{dx}{dt} = -y - z \tag{182}$$

$$\frac{dy}{dt} = x + ay \tag{183}$$

$$\frac{dz}{dt} = b + z(x - c) \tag{184}$$

をルンゲ・クッタ法を用いて数値的に解き, その軌道を 3 次元に描画すると図 44 のようになる. この図を描画するのに, 方程式の初期値は原点近傍 $(x, y, z) = (0.01, 0.01, 0.01)$ に選んである. この図をみるからに複雑なアトラクタであることがわかるが, この振る舞いを方程式の形からある程度説明することができる.

まず, この図 44 から, 軌道がスタートしてからしばらくは x - y 平面内に停滞していることがわかる. そこで, 方程式 (182)(183)(184) において, x - y 平面内近傍での初期の振舞いがどうであるかを調べるために, $z \simeq 0$ とおいてみると, (182)(183) 式から

$$\frac{dx}{dt} = -y \tag{185}$$

$$\frac{dy}{dt} = x + ay \tag{186}$$

が得られる. (185) 式の両辺をを時間で微分し, それに (186) 式を代入すると, 次の x に関する 2 階

$$a=0.398, b=2, c=4$$

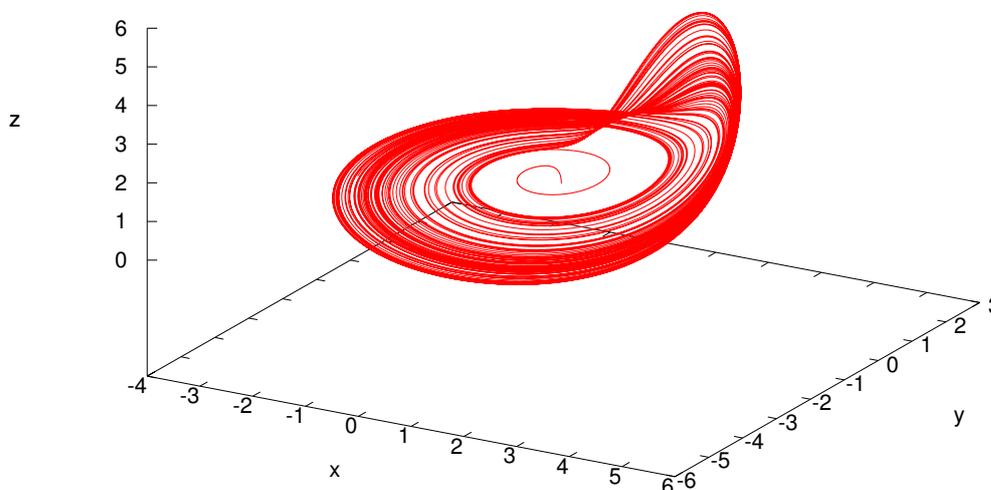


図 44: レスラー方程式の数値解から得られる軌道の様子. パラメータは $a = 0.398, b = 2, c = 4$ に選んである.

の線形微分方程式が得られる.

$$\frac{d^2x}{dt^2} - a \frac{dx}{dt} + x = 0 \quad (187)$$

そこで, これに $x = e^{pt}$ を代入すると, 特性方程式は $p^2 - ap + 1 = 0$ となるので, この 2 次方程式の解は

$$p = \frac{a \pm \sqrt{a^2 - 4}}{2} \quad (188)$$

であり, $0 < a < 2$ のときに $\sqrt{4 - a^2}/2 \equiv \omega$ とおけば, この微分方程式 (187) の解は

$$x = e^{at/2} \{ \alpha e^{i\omega t} + \beta e^{-i\omega t} \} = e^{at/2} \{ A \cos \omega t + B \sin \omega t \} \quad (189)$$

となる. 従って, y は (186) 式から

$$y = -\frac{dx}{dt} = -e^{at/2} \left\{ \left(\frac{aA}{2} + B\omega \right) \cos \omega t + \left(\frac{aB}{2} - A\omega \right) \sin \omega t \right\} \quad (190)$$

である (A, B は初期条件により決まる積分定数). 従って, $x-y$ 平面内での軌道はその「半径」が時間とともに $e^{at/2}$ で大きくなって行くような円運動 (らせん運動) を行う.

ここで, z の寄与が何も無ければ, らせん運動の半径は時間の経過とともに指数関数的な速さで無限大に達するはずであるが, 図 44 からわかるように, そうはなっていない. 従って, 時間の経過

とともに z からの寄与が効いてきているはずなので、それを調べるため、(184) 式を見てみる。この式より、 x の値が c 以下であれば、 $dz/dt = 0$ となる点: $z = -b/(x-c) > 0$ へ収束する。しかし、既にみたように、 x の絶対値は時間とともに大きくなるのであるから、らせん軌道上で $x > c$ となる点で $dz/dt = b + z(x-c) > 0$ となり ($b > 0$ と選んでいるので (ちなみに図 44 では $b = 2$)), z はこの点で急激な増加に転じる (この増加の「急激さ」は b が大きいほど大きい)。実際に図 45 を見

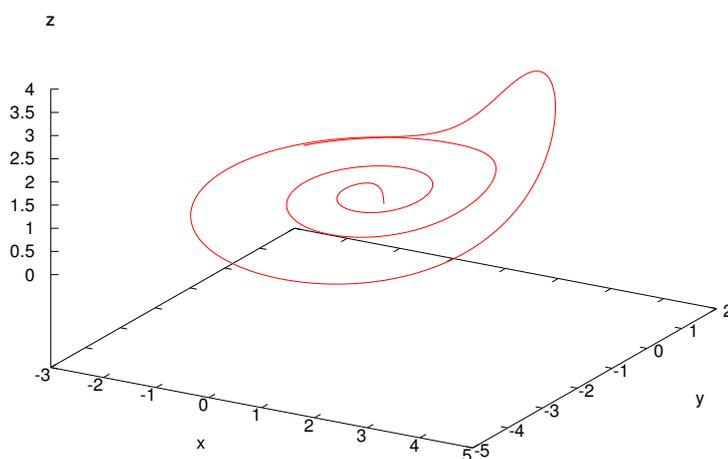


図 45: 図 44 の軌道のうち、ルンゲ・クッタ法でのステップを途中で打ち切って z が急激に増加する部分だけを抜き出したもの。 $x > c$ となる時点で z 方向への急激な増加が始まる。

ると、確かに x の大きさがある値を超えた瞬間に z 方向への軌道の急激な増加が見られる。

しかし、同時にこの図 45 より、この z の方向の増加は長続きせず、軌道は x - y 平面に引き戻されている。これは方程式からも見て取れる。つまり、 z が大きくなると (182) 式から dx/dt の符号が負に傾き始め、従って、 x が減少し始める。すると、 z が増加するための条件であった $x > c$ が崩れ始め、やがて $x < c$ となり、 $dz/dt = b + z(x-c) < 0$ へと転じることから z が減少しはじめ、やがて、 x - y 平面に引きつけられることになる。上記の一連の挙動を繰り返すことで図 44 が得られることになる。

11.2 ポアンカレ断面

前節では数値実験によって得られた「奇妙な」アトラクタ (実際に**奇妙なアトラクタ**, あるいは**ストレンジ・アトラクタ**と呼ばれる) が、その方程式の振る舞いからある程度までは定性的に説明できることを見た。図 44 に示した奇妙なアトラクタは 3 次元空間内の軌道であるので、角度を変えてみることを繰り返すことにより、その構造を視覚的に観察することが重要である。例えば、gnuplot では

```
gnuplot> set view 60,15
```

で見る角度を指定することができる (60,15 などの数値を適時変える. これは後の演習の時間に詳しく説明する).

しかし, ここでは, 軌道がある平面で切った切り口を考えてみる. 3次元軌道を平面で切ったの断面なので, 得られるものは2次元像である. そこで, $x-y$ 面内における x 軸からのなす角度 θ を用いて $y \cos \theta - x \sin \theta = 0$ とし, この平面と z 軸で囲まれた「半平面」でこの軌道を切ってみる. その結果を図 46 に載せよう. この図 46 (図 44 と同じ図) に示した軌道を $y \cos \theta - x \sin \theta = 0$ と z 軸で

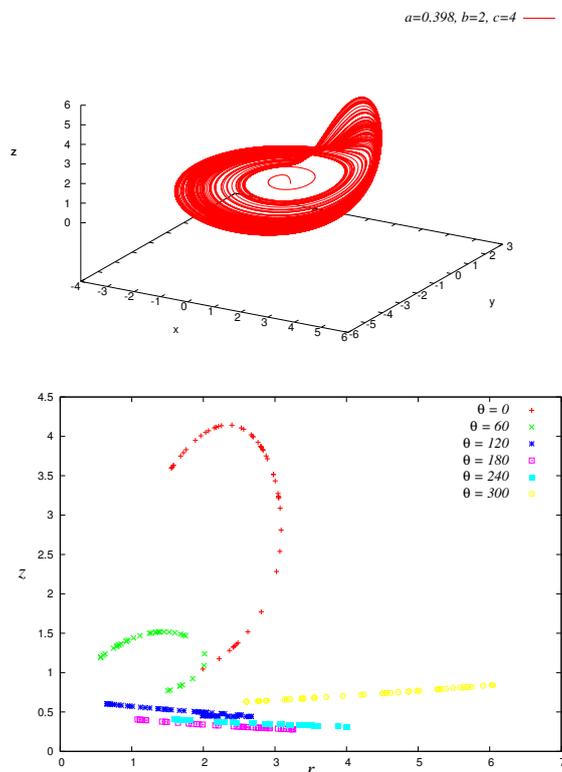


図 46: 左図の軌道を $y \cos \theta - x \sin \theta = 0$ と z 軸で囲まれる半平面で切った断面が右図.

囲まれる半平面で切った断面が右図である. この図より, 角度 θ を $\theta = 0$ から $\theta = 300^\circ$ へ増やすにつれて, 軌道の断面を表す「線分」が「折りたたまれ」, 次いで「引き伸ばされる」様子が見取れる.

このように, 奇妙なアトラクタをある半平面で切り取った断面をポアンカレ断面と呼ぶ. 後の演習の時間で実際に求めてもらうが, このポアンカレ断面を通過する点列を (x_t, x_{t+1}) としてプロットすることで, この力学系を表す実質的な「写像」を取り出すことができる³⁵.

さて, 上に述べた奇妙なアトラクタを示す力学系 (微分方程式) はレスラー方程式だけではない. 例えば, 次のようなローレンツ方程式:

$$\frac{dx}{dt} = -ax + ay \tag{191}$$

$$\frac{dy}{dt} = cx - y - xz \tag{192}$$

³⁵ 「リターンマップ」と呼ばれる.

$$\frac{dz}{dt} = -bz + xy \quad (193)$$

もパラメータ a, b, c の選び方によっては奇妙なアトラクタを持つ。その結果の一例を図 47 に載せる。講義ではこのローレンツアトラクタの動画デモを見せる予定である。

$$a=10, b=8/3, c=28 \quad \text{——}$$

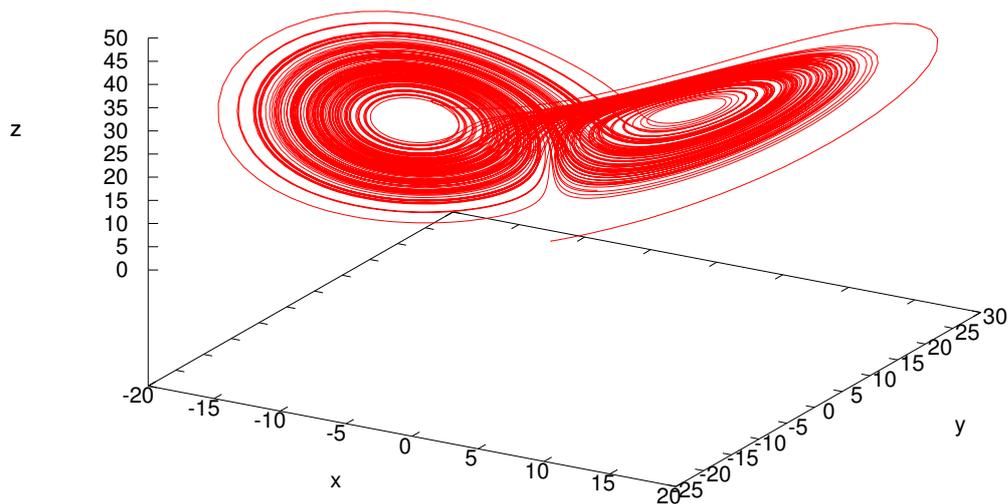


図 47: ローレンツ方程式 (191)(192)(193) から得られる奇妙なアトラクタ. $a = 10, b = 8/3, c = 28$ に選んである.

レポート課題 8

ローレンツ方程式を $a = 10, b = 8/3, c = 30$ に対して数値的に解き, 平面 $z = 20, 30, 40$ で切り取ったポアンカレ断面を求めるプログラムを作成し, 描画ソフトが使えるものはそれをプロットせよ (その手のソフトが使えない者は後の演習の時間に説明するので, 現段階ではソースコードのみで良い).

課題 8 の解答例

先週の復習として, レポート課題を確認しておこう. $a = 10, b = 8/3, c = 30$ と選んだ場合のローレンツ方程式の $z = 20, 30, 40$ におけるポアンカレ断面を図 48 に載せる. また, このグラフを作成

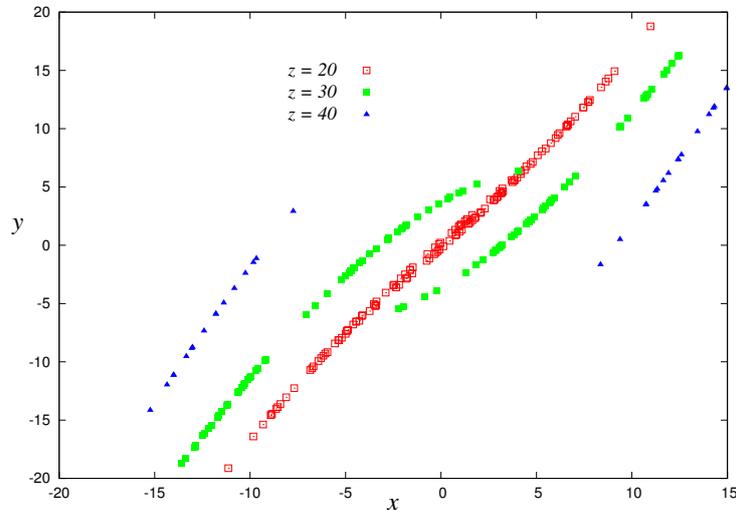


図 48: $a = 10, b = 8/3, c = 30$ と選んだ場合のローレンツ方程式の $z = 20, 30, 40$ におけるポアンカレ断面.

するには

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

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

12 アトラクタの埋め込み次元と相関次元

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

12.1 埋め込み次元とアトラクタの再構成

レスラー方程式:

$$\frac{dx}{dt} = -y - z \tag{194}$$

$$\frac{dy}{dt} = x + ay \tag{195}$$

$$\frac{dz}{dt} = b + z(x - c) \tag{196}$$

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

$$(x(t), x(t + \tau), \dots, x(t + (m - 1)\tau)) \tag{197}$$

そこで, この m 次元ベクトルを逐次プロットすることで, 前回みた奇妙なアトラクタが再構成できるかを調べてみよう. 図 49 に次元を $m = 2$ と選んだ場合の $(x(t), x(t + \tau)), (y(t), y(t + \tau)), (z(t), z(t + \tau))$ をそれぞれプロットした. この図より, x のみ, y, z のみの軌道を適切な時間遅れ τ , 埋め込み次元

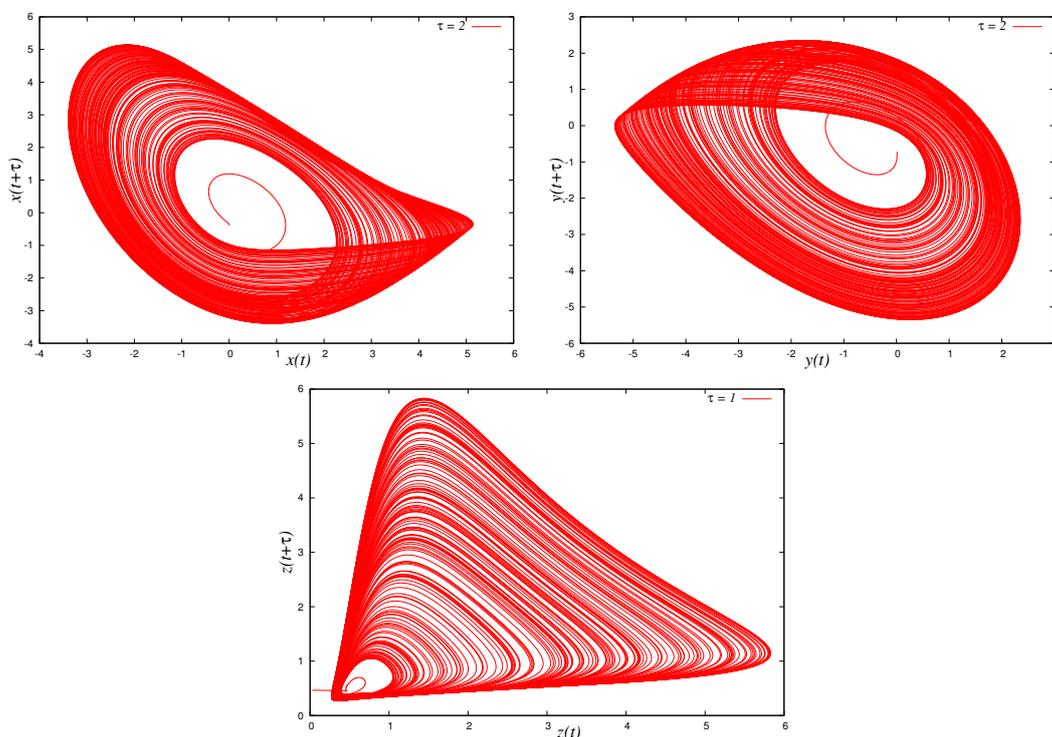


図 49: 再構成. 埋め込み次元を $m = 2$ に選んでいる.

m で切り出すと, x, y, z を 3 次元にプロットした場合とほぼ同じアトラクタを再構成することができる. この例では $m = 2, \tau = 1, 2$ に選んでいるが, 与えられたデータが方程式からの時系列ではなく性質が不明なものである場合, これら 2 つのパラメータを適切に決定しなければならない. 時間遅れ τ は $x(t)$ と $x(t + \tau)$ の自己相関関数, あるいは, 相互情報量を評価することで見積もることができるが, ここではそれには触れない. 次元 m に関しては次に述べる「相関次元」がその指針を与

える. そこで今回の講義では, 相関次元の計算を見ていく. その準備として, まずは「次元」について確認しておこう.

12.2 次元再考

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

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

$$D_0 = -\lim_{\epsilon \rightarrow 0} \frac{\log n(\epsilon)}{\log \epsilon} \quad (198)$$

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

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

$$D_2(\text{長さ } l \text{ の線分}) = -\lim_{\epsilon \rightarrow 0} \frac{\log(l/2\epsilon)}{\log \epsilon} = -\lim_{\epsilon \rightarrow 0} \left\{ \frac{\log(l/2)}{\log \epsilon} \right\} + \frac{\log \epsilon}{\log \epsilon} = 1 \quad (199)$$

となり, 線分の次元は 1 であるという事実に合致する.

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

$$D_2(\text{一辺の長さが } l \text{ の正方形}) = -\lim_{\epsilon \rightarrow 0} \frac{\log(l^2/\pi\epsilon^2)}{\log \epsilon} = -\lim_{\epsilon \rightarrow 0} \left\{ \frac{\log(l^2/\pi)}{\log \epsilon} \right\} + 2 \frac{\log \epsilon}{\log \epsilon} = 2 \quad (200)$$

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

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

12.3 相関次元

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

$$\mathbf{r}_i \equiv (x(t_i), x(t_i + \tau), \dots, x(t_i + (m-1)\tau)), \quad i = 1, \dots, N \quad (201)$$

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

$$p_i = \frac{1}{N} \sum_{j=1}^N \Theta(\epsilon - |\mathbf{r}_i - \mathbf{r}_j|) \quad (202)$$

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

$$|\mathbf{r}_i - \mathbf{r}_j| = |x(t_i) - x(t_j)| + |x(t_i + \tau) - x(t_j + \tau)| + \dots + |x(t_i + (m-1)\tau) - x(t_j + (m-1)\tau)| \tag{203}$$

であることに注意されたい. このとき**相関次元**を

$$D_2 = \lim_{\epsilon \rightarrow 0} \frac{\sum_{i=1}^N p_i^2}{\log \epsilon} \tag{204}$$

で定義する. ここで

$$\sum_{i=1}^N p_i^2 = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \Theta(\epsilon - |\mathbf{r}_i - \mathbf{r}_j|) \equiv C(\epsilon) \tag{205}$$

に注意すると相関次元 (204) は $C(\epsilon)$ を用いて次のように書き直せる.

$$D_2 = \lim_{\epsilon \rightarrow 0} \frac{\log C(\epsilon)}{\log \epsilon} \tag{206}$$

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

$$C(\epsilon) = \epsilon^{D_2} \tag{207}$$

が成り立つ. 従って (205) 式に従って ϵ の変化に対する $C(\epsilon)$ を算出し, 得られる結果を両対数プロットした際のグラフの傾きが相関次元 D_2 を与える.

レスラー方程式において $x(t)$ から時間遅れ $\tau = 0.01 \times 200 = 2$ のもとで $m = 1, 2, 3, 4, 5$ 次元のベクトル $(x(t_i), x(t_i + \tau), \dots, x(t_i + (m-1)\tau)) : i = 1, \dots, N$ を作成し, それから上記レシピに従って相関次元を求めた結果を図 50 載せよう. この図より, m を増加するにつれ, グラフの傾きは

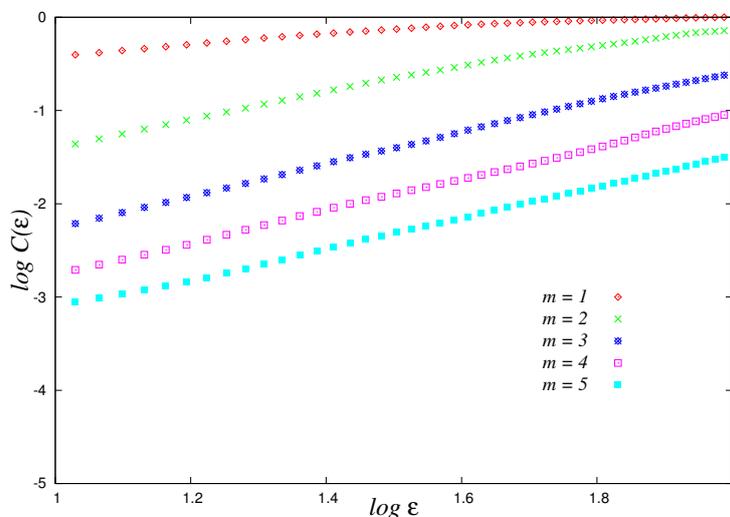


図 50: レスラー方程式において, $x(t)$ から時間遅れ $\tau = 2$ のもとで作成した $m = 1, 2, 3, 4, 5$ 次元のベクトル $(x(t_i), x(t_i + \tau), \dots, x(t_i + (m-1)\tau)) : i = 1, \dots, N$ から算出した相関次元. $N = 3000$ として計算してある. $m = 2$ とすると傾き $D_2 \sim 2$ の直線が得られ, この傾きは以降 m を増やしても変わらない.


```
        }else{
        return (0.0);
        }
}
double DISTANCE(i,j)
int i;
int j;
{
    return (fabs(g1[i]-g1[j])+fabs(g2[i]-g2[j])+fabs(g3[i]-g3[j])
    +fabs(g4[i]-g4[j])+fabs(g5[i]-g5[j]));
}
double COUNT(epsilon,n)
double epsilon;
int n;
{
    int i,j;
    double sum;
for(i=0,sum=0.0; i<n; i++){
    for(j=0; j<n; j++){
        sum = sum + STEP(epsilon-DISTANCE(i,j));
    }
}
    return ((double)sum/(n*n));
}
main()
{
    FILE *fpr;
    int i,k;
    double x,k1x,k2x,k3x,k4x,kx,y,k1y,k2y,k3y,k4y,ky,z,k1z,k2z,k3z,k4z,kz;
    double arg1,arg2,epsilon;
if((fpr = fopen("m5.dat", "wt")) !=NULL){
    for(i = 0,x=0.01,y=0.01,z=0.01; i < imax; i++){
k1x = h*func1(x,y,z);
k1y = h*func2(x,y,z);
k1z = h*func3(x,y,z);
        k2x = h*func1(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
        k2y = h*func2(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
        k2z = h*func3(x+0.5*k1x,y+0.5*k1y,z+0.5*k1z);
k3x = h*func1(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
k3y = h*func2(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
k3z = h*func3(x+0.5*k2x,y+0.5*k2y,z+0.5*k2z);
        k4x = h*func1(x+k3x,y+k3y,z+k3z);
        k4y = h*func2(x+k3x,y+k3y,z+k3z);
```

```

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

```

12.4 相関次元の一般化

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

$$D_q = \frac{1}{q-1} \lim_{\epsilon \rightarrow 0} \frac{\log \sum_{i=1}^{n(\epsilon)} p_i^q}{\log \epsilon} \quad (208)$$

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

$$D_2 = \lim_{\epsilon \rightarrow 0} \frac{\log \sum_{i=1}^N p_i^2}{\log \epsilon} \quad (209)$$

となり, 相関次元が復活する. また, $q = 0$ とおけば

$$D_0 = - \lim_{\epsilon \rightarrow 0} \frac{\log \sum_{i=1}^{n(\epsilon)} 1}{\log \epsilon} = - \lim_{\epsilon \rightarrow 0} \frac{\log n(\epsilon)}{\log \epsilon} \quad (210)$$

となり, 容量次元に一致する. また, $q = 1$ の場合も別な次元が定義される. ここで $q \rightarrow 1$ の極限は $0/0$ を与え, 厄介に見えるが, 分母/分子を q で微分した後に $q \rightarrow 1$ 極限をとっても良いことに気が

つけば (いわゆる「ロピタルの定理」), 確率の規格化: $\sum_{i=1}^{n(\epsilon)} p_i = 1$ に注意して

$$\lim_{q \rightarrow 1} \frac{\partial}{\partial q} \log \left(\log \sum_{i=1}^{n(\epsilon)} p_i^q \right) = \lim_{q \rightarrow 1} \left\{ \frac{\sum_{i=1}^{n(\epsilon)} p_i^q \log p_i}{\sum_{i=1}^{n(\epsilon)} p_i^q} \right\} = \sum_{i=1}^{n(\epsilon)} p_i \log p_i \quad (211)$$

$$\lim_{q \rightarrow 1} \frac{\partial(q-1)}{\partial q} = 1 \quad (212)$$

であるから

$$\lim_{q \rightarrow 1} D_q = D_1 = \lim_{\epsilon \rightarrow 0} \frac{\sum_{i=1}^{n(\epsilon)} p_i \log p_i}{\log \epsilon} \quad (213)$$

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

レポート課題 9

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

[カオス編] はこれでおしまいです. 次回 (7/5) から [フラクタル編] に移ります.

13 フラクタル

[カオス編] は前回で終了したので、ここからはフラクタルについて学んでいく。

13.1 自己相似性

自然界には縮尺 (スケール) を変えても次々と同じ構造・形が現れるようなものが多い。例えば、図 51 の上 2 枚の写真はいずれもブロッコリを撮影したものであるが、それら 2 つを並べた図 51(下) の 1 枚を見るまで、同じ大きさのブロッコリに見える。この例からわかるように、ものによっては



図 51: 自己相似性を持つブロッコリ。

対象への視点をズームイン/アウトしても次々と同じ形状が現れ、我々が日頃その典型的 (特徴的) な大きさを知っている「鉛筆」や「タバコ」などの第三の物体をそれらの傍らに置かないかぎり、その大きさについて知ることはできない (図では背景にあるノートの罫線が薄く見えるが、これによってかろうじて 2 つの大小を比較することができる)。このような性質を**自己相似性**と呼ぶ。この性質は、ここから学習していく「フラクタル」と呼ばれる複雑図形の特徴の一つとなっている。

このブロッコリの例以外にも雲や木、海岸線など実に様々な対象がフラクタルの性質を持っている。また、このようなフラクタル図形の持つ性質を有効に使ったアルゴリズムを構築したり、あるいはコンピュータ・グラフィクス (CG) に用いたりなど工学分野への「応用」も数多くなされている。ここからは、そのようなフラクタルについて基本的な事柄を学習していくが、[カオス編] 同様、この講義ではフラクタルの応用面/より進んだ内容について紹介/学習する余裕はないので、それらについて興味のある者はこの講義を受講後に大学院講義「混沌系工学特論」などを別途聴講するとよいであろう。

13.2 計算機を使って描かれるフラクタル図形

我々の身のまわりには挙げればきりがなほどのフラクタルが存在するが, [カオス編] でみたのと同様, ここでも計算機を有効に用いて人工的なフラクタル図形を描くことができる. ここではフラクタルに関する詳細を見る前にそれらについて概観しておこう.

13.2.1 線形写像の確率的切り換えによるフラクタル図形

変数 r を $[0, 1]$ の一様乱数とし, この各ステップで更新される r の値によって, 線形 2 次元写像の形が切り替わるような簡単な数理模型を考えてみる.

具体的には $0 \leq r < 0.1$ で

$$x_{n+1} = 0.05 x_n \quad (214)$$

$$y_{n+1} = 0.6 y_n \quad (215)$$

$0.1 \leq r < 0.2$ では

$$x_{n+1} = 0.05 x_n \quad (216)$$

$$y_{n+1} = -0.5 y_n + 1.0 \quad (217)$$

$0.2 \leq r < 0.4$ では

$$x_{n+1} = 0.46 x_n - 0.32 y_n \quad (218)$$

$$y_{n+1} = 0.39 x_n + 0.38 y_n + 0.6 \quad (219)$$

$0.4 \leq r < 0.6$ では

$$x_{n+1} = 0.47 x_n - 0.15 y_n \quad (220)$$

$$y_{n+1} = 0.17 x_n + 0.42 y_n + 1.1 \quad (221)$$

$0.6 \leq r < 0.8$ では

$$x_{n+1} = 0.43 x_n + 0.28 y_n \quad (222)$$

$$y_{n+1} = -0.25 x_n + 0.45 y_n + 1.0 \quad (223)$$

$0.8 \leq r \leq 1$ では

$$x_{n+1} = 0.42 x_n + 0.26 y_n \quad (224)$$

$$y_{n+1} = -0.35 x_n + 0.31 y_n + 0.7 \quad (225)$$

とする. つまり, 写像の各ステップにおいて確率 $1/10$ で (214)(215) あるいは (216)(217), 確率 $1/5$ で (218)(219), または (220)(221), または (222)(223), または (224)(225) を選ぶようにする.

実際に上記の写像を 200, 1000, 5000, および 100000 回反復し, その結果を gnuplot で描画した結果を図 52 に載せよう³⁶. 図からわかるように得られる図形は「木」のように見え, ブロccoli と同種の自己相似性を有することがわかる. 参考までにプログラムの主要部分のみを載せておくので, 各自が写像の係数や条件分岐における r の範囲を変更し, 得られる結果を考察してみると良いであろう.

³⁶ Step=100000 の図形は微細な構造をも描くために, gnuplot において with dots オプションをつけてプロットした.

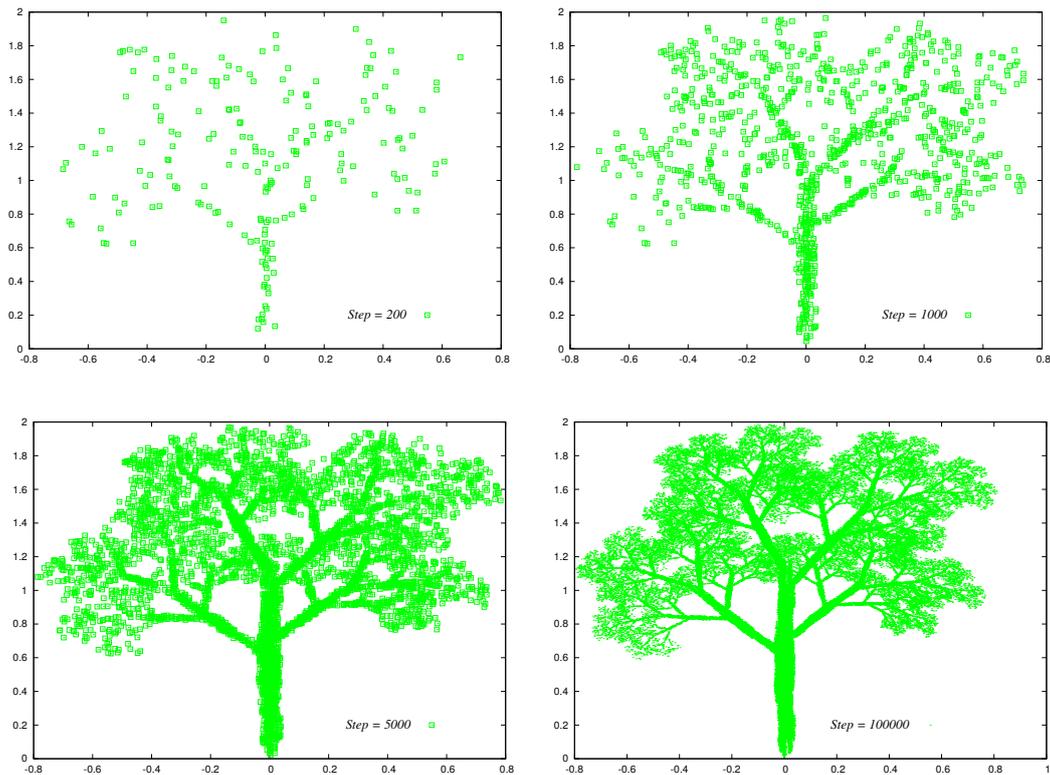


図 52: 計算機を用いて描かれた木。ブロッコリに見えなくもない。

```
#define max 100000
#define SEED 23 /* 乱数の種 */
main()
{
    int i;
    double x,y,xn,yn,r;
    double ran3(long*);
    long idum=(SEED);
    FILE *pt;
    xn = 0.5; /* 初期値 */
    yn = 0.0; /* 初期値 */
    if((pt=fopen("tree.dat","wt")) != NULL){
        for(i=1; i <= max; i++){
            r = ran3(&idum); /* 乱数を入れる */
            if(r < 0.1){
                xn = 0.05*x;
                yn = 0.6*y;
            }else if((r > 0.1) && (r < 0.2)){
                xn = 0.05*x;
```

```
    yn = -0.5*y + 1.0;
}else if((r > 0.2) && (r < 0.4)){
    xn = 0.46*x - 0.32*y;
    yn = 0.39*x + 0.38*y + 0.6;
}else if((r > 0.4) && (r < 0.6)){
    xn = 0.47*x - 0.15*y;
    yn = 0.17*x + 0.42*y + 1.1;
}else if((r > 0.6) && (r < 0.8)){
    xn = 0.43*x + 0.28*y;
    yn = -0.25*x + 0.45*y + 1.0;
}else{
    xn = 0.42*x + 0.26*y;
    yn = -0.35*x + 0.31*y + 0.7;
}
    fprintf(pt, "%f %f\n", xn, yn);
    x = xn;
    y = yn;
}
}
fclose(pt);
}
```

13.2.2 パスカルの三角形とフラクタル図形

この他にも、代表的なフラクタル図形として図 53 のようなシェルピンスキー・ガスケットが知られている。この図形の描画方法は様々あるのだが、例えば、よく知られた「パスカルの三角形」に

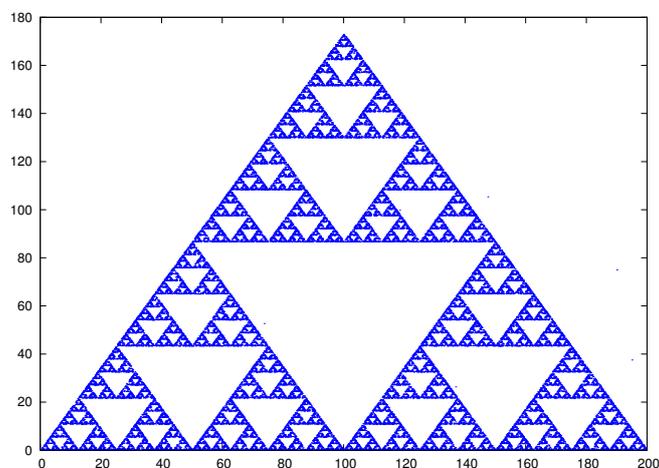


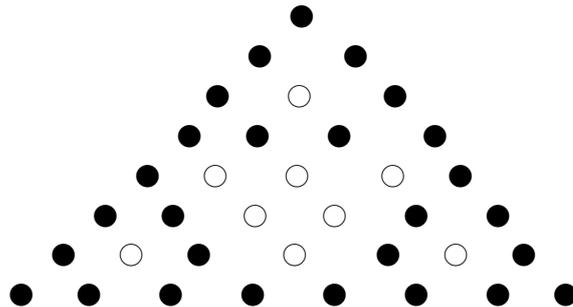
図 53: シェルピンスキー・ガスケット.

において奇数を●, 偶数を○で塗り分けるとシェルピンスキー・ガスケットが得られる³⁷.

パスカルの三角形

$$\begin{array}{cccccccc}
 & & & & & & & 1 \\
 & & & & & & 1 & 1 \\
 & & & & & 1 & 2 & 1 \\
 & & & & 1 & 3 & 3 & 1 \\
 & & & 1 & 4 & 6 & 4 & 1 \\
 & & 1 & 5 & 10 & 10 & 5 & 1 \\
 & 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 7 & 21 & 35 & 35 & 21 & 7 & 1
 \end{array}$$

シェルピンスキー・ガスケット (の一部)



13.2.3 フラクタル図形を「なか抜き」の繰り返しにより描く

シェルピンスキー・ガスケットは上記以外にも, 図 54 のように正三角形の中央部分から逆正三角形を「なか抜き」していくプロセスを無限回繰り返すことでも描画することができる. この図

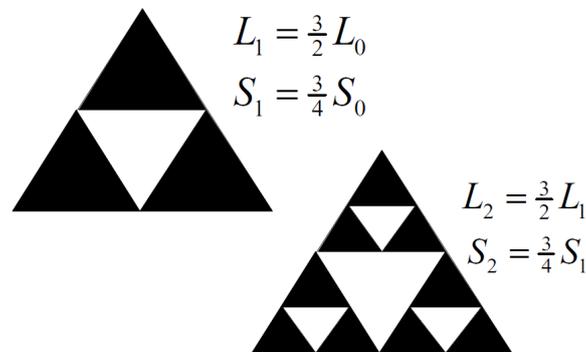


図 54: 正三角形の中央部分から逆正三角形を「なか抜き」していくプロセスでもガスケットは描ける. S_0, S_1, S_2 は各ステップでの図形の総面積. L_0, L_1, L_2 は各ステップでの図形の全周囲長.

³⁷ かなり大きなサイズのパスカルの三角形を作らないと見えない. 計算機で描画するにはどうすればよいか考えてみるとよい.

で S_0, S_1, S_2 は各ステップでの図形の総面積. L_0, L_1, L_2 は各ステップでの図形の全周囲長を表すが, はじめの正三角形の一辺の長さを l とすると, その全周囲長は $L_0 = 3l$ である. 一方, 上記のプロセスを 1 回経た後の図形の全周囲長は $L_1 = 3l + (l/2) \times 3 = (1 + 1/2)3l = (3/2)L_0$ である. 従って, このプロセスを 2 回経ると全周囲長は $L_2 = (3/2)L_1 = (3/2)^2 L_0$ となり, n 回繰り返せば $L_n = (3/2)^n L_0$ である. 従って, この操作を無限回繰り返せば

$$\frac{L_n}{L_0} = \left(\frac{3}{2}\right)^n \rightarrow \infty \quad (226)$$

となり, 全周囲長は発散することになる.

一方, 面積の方は, もとの正三角形の面積が $S_0 = (\sqrt{3}/4)l^2$ で与えられるが, 上記のプロセスを 1 回経た後には $S_1 = (\sqrt{3}/4)(l/2)^3 \times 3 = (\sqrt{3}/4)l^2 \times (3/4) = (3/4)S_0$ であるから, n ステップ後には $S_n = (3/4)^n S_0$ となる. 従って, この操作を無限回繰り返せば

$$\frac{S_n}{S_0} = \left(\frac{3}{4}\right)^n \rightarrow 0 \quad (227)$$

となって総面積はゼロになる.

このように, 無限回の操作後に描かれるこの図形の総面積はゼロに収束するが, 一方の全長が無限大に発散する. これはこの図形に特徴的な (典型的な) 長さスケールが存在しないことを意味しており, この点がフラクタル図形の持つ顕著な性質の一つとなっている.

13.2.4 粒子のランダムウォークと吸着/凝集によるフラクタル図形

例えば, 菌糸の成長過程などは「核 (種)」の周りがある確率に従って動き回る微粒子が次々と核に吸着していくことにより複雑な図形が出来上がるプロセスとして計算機上で再現できる. できあがる図形はフラクタルである.

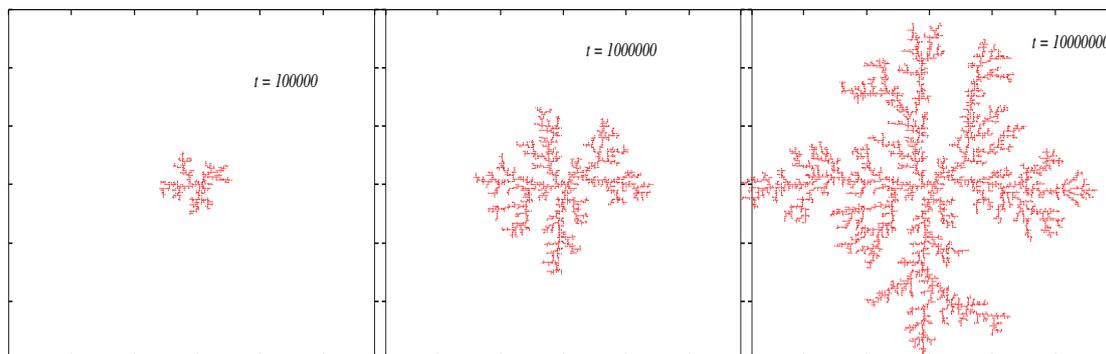


図 55: 菌糸の成長過程. 時間は左から右に流れている.

図 55 は中央に置かれた菌糸の「種」に周りからランダムウォーク (酔歩) しながらやってくる菌が次々と付着してゆくことによって, この菌糸が成長していく様子をあらわしている.

13.3 カントール集合

ここまでいくつかのフラクタル図形の作り方についてざっと観てきたが, ここではもう少しシンプルなフラクタル図形を考えることで「フラクタル」という概念についての理解を深めていこう.

13.3.1 作り方

その図形の作り方は簡単であり、図のように長さ 1 の線分を 3 等分し、その真ん中の部分を削除する。次いで、両端の残された長さ 1/3 の線分を同様に 3 等分し、その真ん中の部分をそれぞれ削除する。この一連の操作を無限に繰り返す。この手続きで得られる図形はフラクタルであり、**カン**

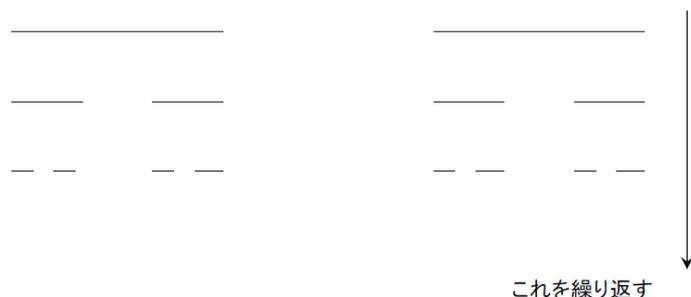


図 56: カントール集合の作り方.

トール集合と呼ばれている。

13.3.2 カントール集合の 3 進小数表現と自己相似性

カントール集合の自己相似性をみるために、上記のような作り方で「なかぬぎ」されて残った部分の「端点」を考えてみよう。その際、作り方から明らかなように、十分な繰り返しの後においても、これら端点はカントール集合の一部となることに注意しよう。ここで、線分を 1/3 ずつ区切り、その中央線分を取り除いていくわけであるから、端点は 1/3, 2/3, 1/9 のように、 $1/3^n$ を共通因子に持つ。そこで、これら端点を 3 進小数で表現してみることにする。[カオス編]において、ベルヌーイ写像が 2 進小数を用いて表現できることは既に学んでいるが、3 進小数表現はそれとほぼ同じ操作を行えばよい。x を 10 進小数とすると、これを 3 進小数で書き直すためには x を

$$x = a_1 \cdot 3^{-1} + a_2 \cdot 3^{-2} + a_3 \cdot 3^{-3} + \dots + a_n \cdot 3^{-n} + \dots \tag{228}$$

と表して $(0.a_1a_2 \dots a_n \dots)_3$, $a_i \in \{0, 1, 2\}$ がその 3 進小数表示となる。

このとき、例えば 1/3 は

$$\begin{aligned} 1/3 &= 0 \cdot 3^{-1} + 2 \cdot 3^{-2} + 2 \cdot 3^{-3} + \dots + 2 \cdot 3^{-n} + \dots = \frac{2}{3} \left\{ \frac{1}{3} + \frac{1}{3^2} + \dots \right\} \\ &= \frac{2}{3} \frac{\frac{1}{3}}{1 - \frac{1}{3}} = 1/3 \end{aligned} \tag{229}$$

であるから、 $1/3 = (0.02 \dots)_3$ であり、1/9 は 1/3 と同様にして

$$\begin{aligned} 1/9 &= 0 \cdot 3^{-1} + 0 \cdot 3^{-2} + 2 \cdot 3^{-3} + \dots + 2 \cdot 3^{-n} + \dots = \frac{2}{9} \left\{ \frac{1}{3} + \frac{1}{3^2} + \dots \right\} \\ &= \frac{2}{9} \frac{\frac{1}{3}}{1 - \frac{1}{3}} = 1/9 \end{aligned} \tag{230}$$

となるので, $1/9 = (0.002\cdots)_3$ である.

一方, 端点ではなく, なかぬきされた方の線分上の点, 例えば $x = 1/3 + 1/9 = 4/9$ を 3 進小数表現してみると $4/9 = (0.102\cdots)_3$ が得られる. ここで注目すべきは, カントール集合の要素, すなわち, **各ステップで残される線分の端点の 3 進小数表現には 1 が含まれず, 逆に, なかぬきされて除かれた線分上の点の 3 進小数表現には 1 が含まれる**という事実である. 言い方を換えれば, カントール集合は $[0, 1]$ の実数を 3 進小数表現した集合から, 1 を含む部分集合を取り除いた集合である.

このことから, また, カントール集合の自己相似性を 3 進小数表現の観点から直接みることができ. つまり, 任意のカントール集合の要素は

$$x = a_1 \cdot 3^{-1} + a_2 \cdot 3^{-2} + a_3 \cdot 3^{-3} + \cdots + a_n \cdot 3^{-n} + \cdots, \quad a_i \neq 1 \quad (231)$$

で与えられるので, その縮尺を $1/3$ 倍してみるには, 上記 x を $1/3$ 倍すればよいので

$$\frac{x}{3} = a_1 \cdot 3^{-2} + a_2 \cdot 3^{-3} + a_3 \cdot 3^{-4} + \cdots + a_n \cdot 3^{-n-1} + \cdots \quad (232)$$

が得られるが, これは x の 3 進小数を右側に 1 つシフトすることを意味するので, x の 3^{-i} についての展開の全ての係数 a_i が 1 を含まなければ $x/3$ の展開係数も 1 を含まず, 逆に, x の展開係数に 1 が含まれれば, $x/3$ の展開係数にも必ず 1 が含まれることになる. よって, カントール集合は自己相似であることがわかる.

レポート課題 10

次の 1 次元写像を考える.

$$x_{n+1} = \begin{cases} 3x_n & (x \leq 0.5) \\ -3x_n + 3 & (x > 0.5) \end{cases} \quad (233)$$

このとき, 写像の初期値 x_0 がカントール集合の要素である場合とそれ以外の場合とで写像の振る舞いに違いがでるか, もし, 違いがでるとしたらどのような振る舞いか, を計算機を用いた数値計算で調べよ.

課題 10 の解答例

まずは復習を兼ねて前回 (7/5) のレポート課題についてみていく.

我々がここで考える写像は

$$x_{n+1} = \begin{cases} 3x_n & (x \leq 0.5) \\ -3x_n + 3 & (x > 0.5) \end{cases} \quad (234)$$

である. この写像の形を見て直ぐにわかるのは, $x = 0$ は固定点であり, x はずっと $x = x_0 = 0$ に居座り続けることである. また, $x_0 = 1/3$ も 1 回目の反復で $x_1 = 1$ となり, 2 回目の反復で $x_2 = -3 \cdot 1 - 3 = 0$ となり, 以降ずっと $x = 0$ である. しかし, 例えば $x_0 = 1/2$ に選ぶと, 反復を繰り返すごとに x の値はマイナス方向に大きくなっていき, $-\infty$ に至ることになる. 従って, この写像の初期値の選び方によっては x の行き先が負の無限大に至るものと, ゼロに至るものの 2 通りが存在するのではないかと示唆を与える. 実際に計算機を使っていくつかの初期値に対し, 写像のはじめの数ステップをプロットしたものを図 57(左) に載せよう. この図からわかるように, $x_0 = 1/3, 1/9$ の場合, 数ステップの後に x の値はゼロとなる. 一方, $x_0 = 1/2, 1/5$ の場合にはステップ数の増加とともに, x の値は負の無限大へ行くように見える. そこで, x_0 の値を 0 から細か

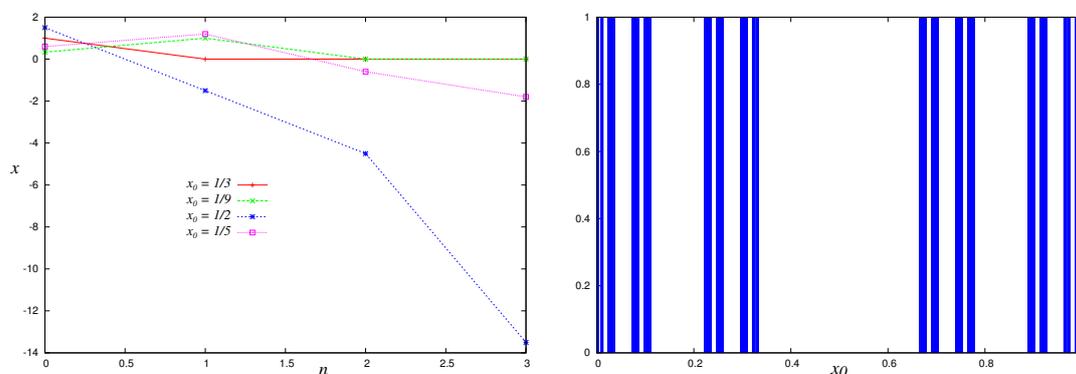


図 57: 写像力学系のゼロへ収束する初期値の集合がコントロール集合を与える例. 左図はいくつかの初期値に対する写像 (234) の数ステップの様子. 右図はゼロに収束する初期値の集合 (図では見やすいように「バーコード」で描いている). これはコントロール集合となる.

く刻んで行って, x の絶対値がゼロに収束するもののみをプロットすると図 57(右) のようになる. これは前回の講義で学んだコントロール集合に他ならない.

この課題でみたように, 非線形写像の振る舞いを決める初期値条件の集合が自己相似な図形を与える場合がある. 今回はそのような写像を複素数に拡張した場合, 同様な自己相似図形が得られるかをみていく. 講義での解説を聞いているだけではなかなか感じがつかめないと思うので, 各自が計算機を用いて実験されることを強く勧める.

今回の講義はこの講義ノート中に現れる課題に関して説明していく.

14 複素力学系

既に [カオス編] で学んだ一次元写像を複素数に拡張した場合を考えてみよう.

14.1 ロジスティック写像の複素数への拡張

既に [カオス編] で学んだロジスティック写像を思い出してみよう.

$$x_{n+1} = ax_n(1 - x_n) \quad (235)$$

写像 (漸化式)(235) に含まれるパラメータ a を変えていくと $x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n \dots$ という時系列は「固定点」「周期軌道」「カオス」というような様々な振る舞いを見せた. また, a の変化に対してその周期軌道の分岐構造がどうであるかは図 58 のようになったことも確かめた. そこで

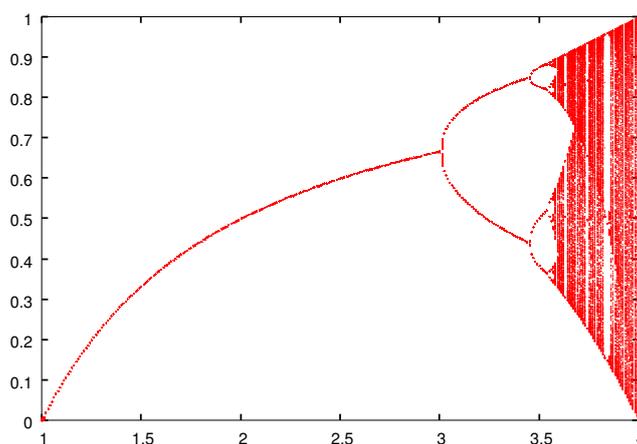


図 58: ロジスティック写像の分岐図

は写像の変数 x を実数であるとしたが, これを複素数 z に拡張した場合に得られる複素数版のロジスティック写像からは何がわかるのかを調べてみよう. 複素数 z の実部を x , 虚部を y で表せば, z は $z = x + iy$ と書けるので, これをロジスティック写像の式: $z_{n+1} = az_n(1 - z_n)$ に代入すれば

$$\begin{aligned} x_{n+1} + iy_{n+1} &= a(x_n + iy_n)(1 - x_n - iy_n) \\ &= a(x_n - x_n^2 + y_n^2) + iay_n(1 - 2x_n) \end{aligned} \quad (236)$$

が得られる. ここでパラメータ a は $a = a_R + ia_I$ として複素数に選ぶこともできるが, まずは簡単のため実数であるとして話を進めよう.

さて, (236) 式の両辺の実部, 虚部をそれぞれ等しいと置くことにより

$$x_{n+1} = a(x_n - x_n^2 + y_n^2) \quad (237)$$

$$y_{n+1} = ay_n(1 - 2x_n) \quad (238)$$

が得られる. この連立漸化式 (237)(238) をある初期条件 $x_0 \equiv (x_0, y_0)$ からスタートさせると, この初期条件の選び方が良ければ連立漸化式は複素平面 x - y 内のある点へと収束するか, 有限範囲内の周期軌道へと収束していく. しかし, 一方で初期条件の選び方がうまくなければこの連立漸化

式は収束せず, $|x_\infty| \equiv \sqrt{x_\infty^2 + y_\infty^2} \rightarrow \infty$ のように発散してしまう. 従って, 全ての可能な初期条件はその初期条件から出発した軌道が [発散してしまうもの] と [有限に留まるもの] との 2 種類に分類されることがわかる. つまり, 初期条件の作る集合: $\{x_0\}$ は

$$\begin{aligned} \{x_0\}_\infty &\equiv \{x_0 \mid |x_\infty| \rightarrow \infty\} \\ \{x_0\}_{\text{finite}} &\equiv \{x_0 \mid |x_\infty| < \infty\} \end{aligned}$$

でそれぞれが定義される部分集合 $\{x_0\}_\infty$ と $\{x_0\}_{\text{finite}}$ に分類される.

連立漸化式 (237)(238) を数値的に解き, 複素平面 x - y 内にこの $\{x_0\}_{\text{finite}}$ をプロットしてできる図は, その作り方が上述のように極めて単純なのにも関わらず非常に複雑な形状を持つ. それを図 59 に示そう. この集合 $\{x_0\}_{\text{finite}}$ のことを**ジュリア集合**と呼ぶ. 図 59 では $a = 3.3$ に選んだが, こ

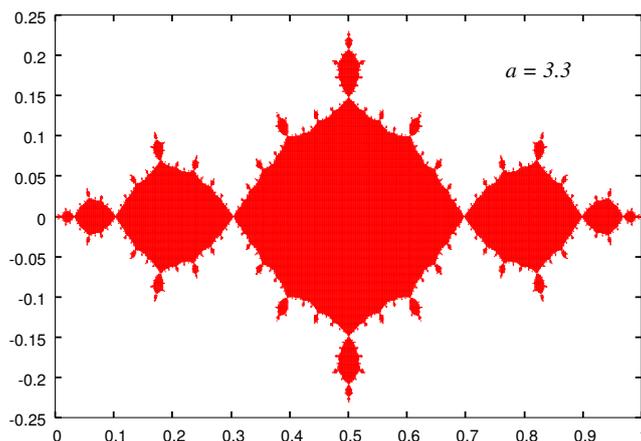


図 59: ジュリア集合. $a = 3.3$

の場合に得られるジュリア集合は自己相似性を有する. もちろん, a の値をいろいろ変えれば様々なジュリア集合が得られる (各自がいろいろ試してみると楽しいと思う). 図 60 にその中から数例を載せる.

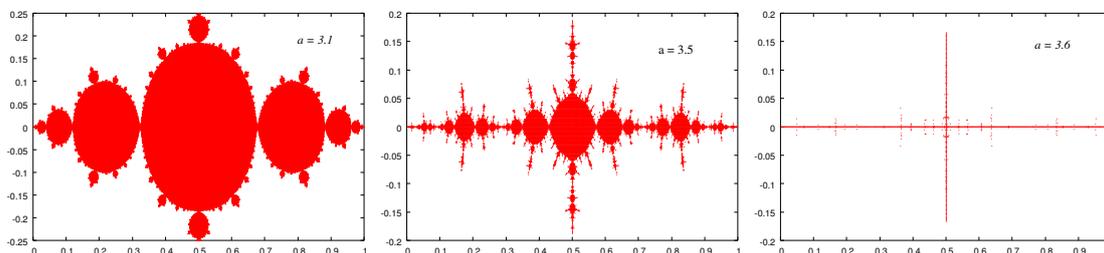


図 60: 左から $a = 3.1, 3.5$, 及び, $a = 3.6$ に対するジュリア集合.

14.2 マンデルブロ集合

前節ではロジスティック写像をコントロールするパラメータ a を実数として扱ったが, 先に述べたようにこれをも複素数に拡張することができる. $a = a_R + ia_I$ として (236) 式に代入し, 実部と

虚部にわたった反復式を書き出してみると

$$x_{n+1} = a_R(x_n - x_n^2 + y_n^2) - a_I y_n(1 - 2x_n) \quad (239)$$

$$y_{n+1} = a_R y_n(1 - 2x_n) + a_I(x_n - x_n^2 + y_n^2) \quad (240)$$

が得られる. ジュリア集合を求めたときには, a の値を固定し, 反復式 (239)(240) が有限の値に収まるような初期条件の集合を複素平面内にプロットした. ここでは逆に, 初期条件 $\{x_0, y_0\}$ が一つ与えられた場合, a_R, a_I の値を様々変えたときに, 反復式 (239)(240) の解が有限の値を持つような集合 $\{a_R, a_I\}$ を x - y 平面にプロットしたならばどのような図形が得られるのかを考えてみたい. 図

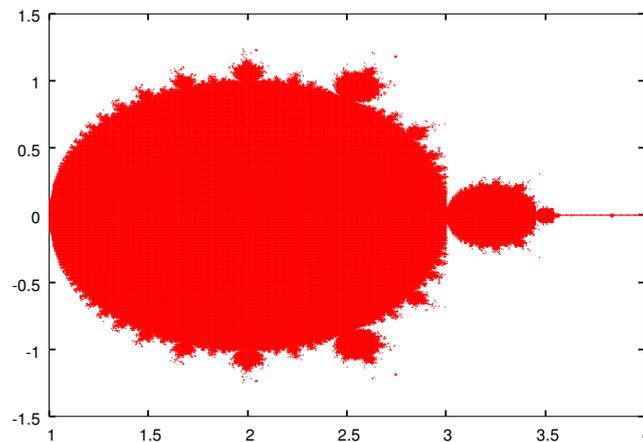


図 61: マンデルブロ集合. $x_0 = 0.5, y_0 = 0$ と初期条件を選んである.

61 に $x_0 = 0.5, y_0 = 0$ を初期条件に選んだ場合の結果を載せる. このような図形のことを**マンデルブロ集合**と呼ぶ.

15 確率的フラクタル

ここからは確率モデルを用いたフラクタルについて見ていく.

15.1 シェルピンスキー・ガasket

図 62 は前回学んだシェルピンスキー・ガasketであり, これは典型的なフラクタル図形の一つである³⁸. この図形は次に示すような非常に簡単なルールを用いて生成させることができる.

- (1) 図 63 のように頂点を $(X_1, Y_1) = (0, 0)$ (点 O), $(X_2, Y_2) = (2l, 0)$ (点 A), $(X_3, Y_3) = (l, \sqrt{3}l)$ (点 B) とする正三角形を考える.
- (2) この正三角形内部の任意の点 $P^{(0)} = (x_0, y_0)$ を選ぶ.
- (3) 数 0, 1, 2 を**ランダム**に選び
 - $0 \Rightarrow$ O と $P^{(0)}$ を結ぶ線分の midpoint に点 $P^{(1)} = (x_1, y_1)$ を置く.

³⁸ ガasket (gasket) とは「詰め物」の意味である.

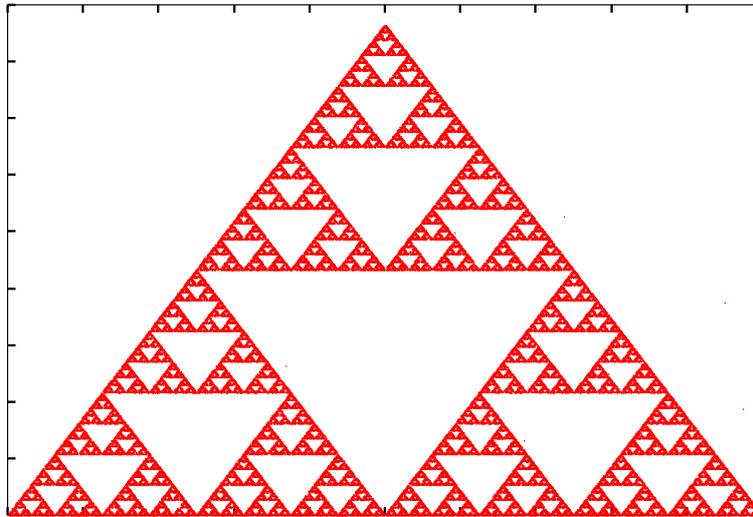


図 62: シェルピンスキー・ガスケット.

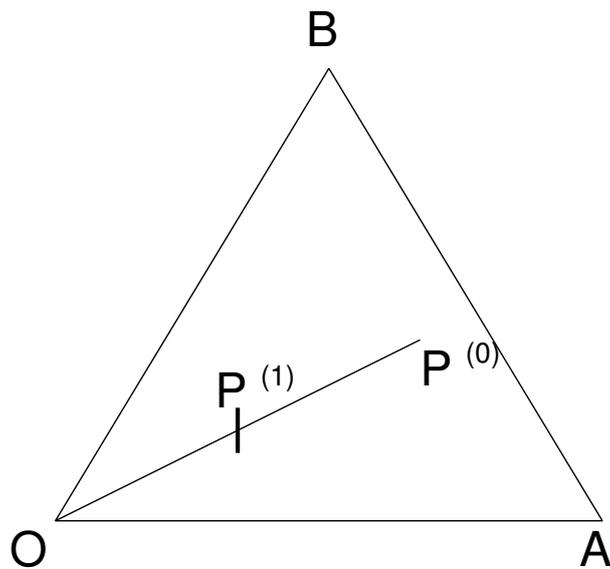


図 63: O が選ばれた場合には $P^{(0)}$ と O の中点に $P^{(1)}$ をおく.

- 1 \Rightarrow A と $P^{(0)}$ を結ぶ線分の midpoint に点 $P^{(1)} = (x_1, y_1)$ を置く.
- 2 \Rightarrow B と $P^{(0)}$ を結ぶ線分の midpoint に点 $P^{(1)} = (x_1, y_1)$ を置く.

(4) プロセス (3) を十分多数回繰り返す.

上記のアルゴリズムにより, 点列

$$P^{(0)}(x_0, y_0) \rightarrow P^{(1)}(x_1, y_1) \rightarrow \dots \rightarrow P^{(n)}(x_n, y_n) \rightarrow \dots$$

を 2 次元平面内にプロットする. 具体的には

x_0 y_0

x_1 y_1

x_2 y_2

x_3 y_3

.....

という形式でデータファイル (例えば `gasket.dat`) に格納する.

課題 1

上記 (1) から (4) のプロセスをコーディングし, シェルピンスキー・ガasket を描け.

(注): 実際の作図には `gnuplot` を用いよ.

15.2 菌糸成長の計算機シミュレーション

前節で見たシェルピンスキーガasket は確率的なメカニズムがその生成過程に含まれているとはいえ, コンピュータ上に人工的に作られたフラクタル図形であった. しかし, この他にも自然界に目を向ければ, 木, 海岸線など, 数多くのフラクタル構造を持つものが実在する. また, はじめに我々が見たジュリア集合のフラクタル図形はその作り方から明らかなように, 点列: $P^{(0)}(x_0, y_0) \rightarrow P^{(1)}(x_1, y_1) \rightarrow \dots \rightarrow P^{(n)}(x_n, y_n) \rightarrow \dots$ は決定論的な規則³⁹ に従ってグラフ上に点が打ち込まれることにより, 全体がフラクタルとなった.

一方で, 自然界に見られる多くのフラクタル図形の形成メカニズムには確率的要素が重要になってくるものも少なくない. ここでは, その中でも菌糸の成長過程, つまり, 核の周りがある確率に従って動き回る粒子が次々と核に吸着していくことにより複雑な図形が出来上がるプロセスをコンピュータ上にシミュレートし, 出来上がる図形がフラクタルであることを確認する.

図 64 は中央に置かれた菌糸の「種」に周りから **ランダムウォーク (酔歩)** しながらやってくる菌が次々と付着してゆくことによって, この菌糸が成長していく様子をあらわしている. この過程の作図を始める前に, まずは 2 次元格子上的ランダムウォークを考えてみよう.

15.2.1 準備: 2 次元ランダムウォーク

我々の最終目標である「菌糸の成長」をシミュレートするためには 2 次元格子上のランダムウォークのプログラムを用意する必要がある. そこで, ここでは準備としてそれを見てゆくことにする. ラ

³⁹ これを作図プログラムには乱数の入る余地がない. この意味で確率を用いない決定論的規則である.

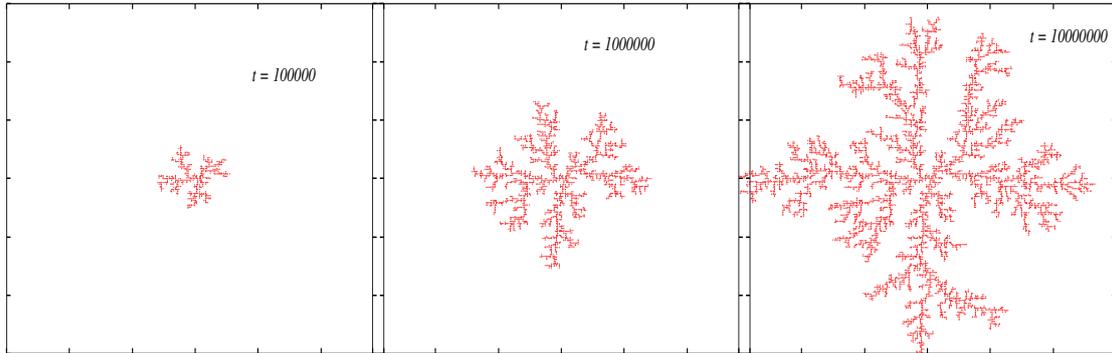


図 64: 菌糸の成長過程. 時間は左から右に流れている.

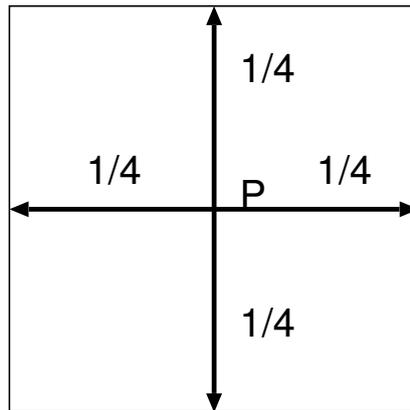


図 65: 点 P から菌は隣接する 4 つの格子点に等確率でジャンプする.

ランダムウォークは任意の点 (例えば原点 $(0, 0)$) から出発して, 2次元正方格子 (格子の形は何でもよいが, ここでは正方格子とする) で, 等確率で 4 つの方向の中から 1 つをランダムに選択し (図 65 参照), 移動してゆくものである.

課題 2

2次元正方格子ランダムウォークの軌跡を描け. 例を図 66 載せる.

15.2.2 菌糸成長のアルゴリズム

以上をふまえて具体的な菌糸成長のアルゴリズムを以下で説明する.

- (1) 菌 (粒子) の可動範囲は縦 \times 横 $= l_{\max} \times l_{\max}$ の正方格子とする. (l_{\max} は偶数). この中央 $S(l_{\max}/2, l_{\max}/2)$ に「種」を置き, 菌糸を取り囲む最小半径 r_{\max} を $r_{\max} = 1$ にセットする (図 67 参照).
- (2) S を中心とする半径 $r_s (= r_{\max} + 2.0)$ の円周上の任意の一点 $P(rx, ry)$ をスタート地点に選ぶ. ただし

$$rx = r_s \cos \phi$$

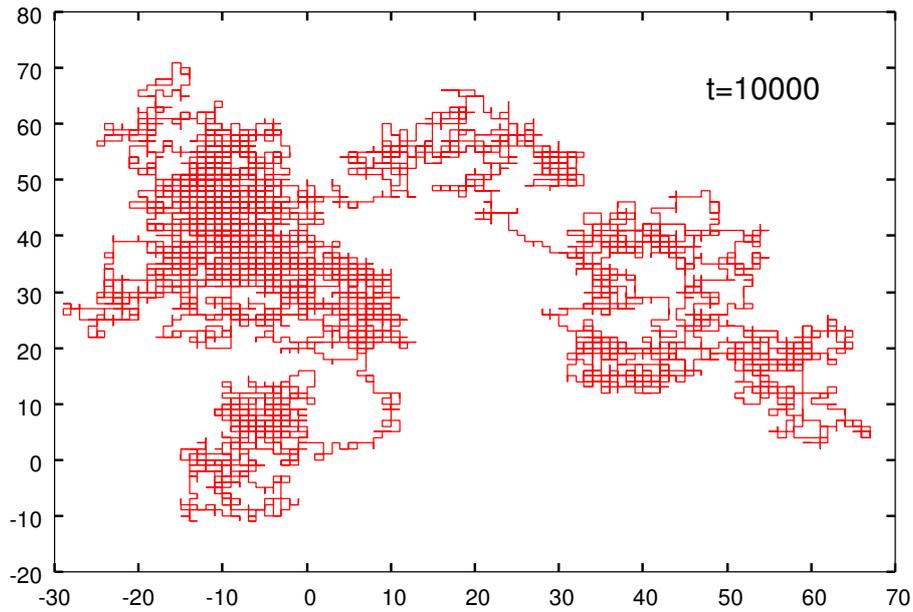


図 66: 2次元正方格子上のランダムウォークの例.

$$ry = r_s \sin\phi$$

とし, ϕ は $\phi \in [0, 2\pi]$ の乱数として選ぶ (図 68 参照).

- (3) この点 P をスタート地点に選び, **課題 2** で行った 2次元正方格子上のランダムウォークをさせる.
- (4) $r \equiv \sqrt{(rx)^2 + (ry)^2}$ の大小関係により, 次のように処理を分岐させる.

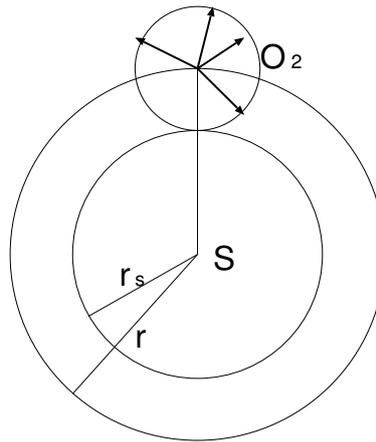
- (k) r が r_k (この値は自分で適当に決める.) を超えたら, P を置き直す. そして (2) へ戻る.
- (c) もし, $r_k > r_s$ なる $r_d (= r_{\max} + 5.0)$ に対し, $r \geq r_d$ ならば次のようなジャンプをさせる. 図 69 のように, 中心が半径 r の円周上, 半径が $(r - r_s)$ の円を考え, この円 O_2 上の任意の点にランダムにジャンプさせる. つまり,

$$\begin{aligned} rx &= rx + (r - r_s)\cos\theta \\ ry &= ry + (r - r_s)\sin\theta \end{aligned}$$

とし, θ を $\theta \in [0, 2\pi]$ の乱数として選ぶ.

- (a) もし, 粒子の現在地 $(rx + l_{\max}/2, ry + l_{\max}/2)$ に隣接する 4 つの格子点 $(rx + l_{\max}/2 + 1, ry + l_{\max}/2)$, $(rx + l_{\max}/2, ry + l_{\max}/2 + 1)$, $(rx + l_{\max}/2 - 1, ry + l_{\max}/2)$, $(rx + l_{\max}/2, ry + l_{\max}/2 - 1)$ に粒子が存在すれば $(rx + l_{\max}/2, ry + l_{\max}/2)$ に粒子を置き, r_{\max} を

$$r_{\max} = \max \left[r_{\max}, \sqrt{(rx)^2 + (ry)^2} \right]$$

図 69: 半径 $(r - r_s)$ の円 O_2 上にジャンプする.

```

/*      菌糸成長のシミュレーション (main 関数のみ)      */
/*****
#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#define lmax 620
#define rs (rmax+2.0)
#define rd (rmax+5.0)
#define rkill (5*rmax)
#define PI 3.14159265
/*****
/*      乱数の種      */
/*****
#define SEED1 131
#define SEED2 233
#define SEED3 151

/* (rx,ry) に粒子がいるかどうかを決める配列. いれば xf[rx][ry]=1, いなけりやxf[rx][ry]=0. */
int xf[lmax][lmax];

int rx, ry;
double rmax;

main()
{
    FILE *ptr;
    long i,k;
    rmax=10;

```

```
init();      /* 中央に種を置く関数 */
occupy();    /* スタート地点 P を選ぶ関数 */
jump();      /* 任意の点で 4 つの隣接格子点のいずれかにジャンプさせる関数 */

for(i = 0; i <= 1000000; i++){
  switch(check()) { /* check() は (4) での分岐条件の k,a,j,c のいずれかを返す関数 */
  case 'k':
    occupy();
    jump();
    break;
  case 'a':
    aggregate();      /* r_max を算出する関数 */
    if((ptr=fopen("test.dat","at")) != NULL){
      fprintf(ptr, "%d %d \n", rx, ry); /* 粒子を位置 (rx,ry) に置く */
    }
  }
  fclose(ptr);

  occupy();
  jump();
  break;
  case 'j':
    jump();
    break;
  case 'c':
    longjump(); /* 半径 (r-r_s) の円周上へジャンプ. (c) の作業に相当する関数 */
    break;
  }
}
}
```

15.3 フラクタル次元とその計算方法

前節まででいくつかのフラクタル図形に関して学び、実際にそれらを作図してきた。ところで、フラクタル図形をスケールを変えてみても同じ図形に見えるものとして定義したが、もう少し定量的にフラクタル図形とその他の図形を区別するための指標はないであろうか？

ところで「次元」というものに関し、我々は経験上、直線は 1 次元、正方形は 2 次元、立方体は 3 次元であると認識している。ここでは各図形に関する次元の決め方を我々の経験則とも一致するような形で定義することにより、フラクタル図形の次元 — **フラクタル次元** — を考えてみることにする。

一辺が l の図形の中に含まれる格子点の数を $m(l)$ とする. もし, 格子点が全て同じ重さを持っていたとすればこれは一辺が l の長さの図形の質量を表している (図 70 参). さて, l を 1 から徐々に増加させてみると, 正方形の場合, $m(1) = 4, m(2) = 9, \dots, m(l) = (l+1)^2$ となる. 従って, l が十分大きいときには

$$m(l) = Al^2$$

となることはわかるであろう. ここで, A は比例係数である. 同様にして考えると立方体の場合は

$$m(l) = Al^3$$

となっていることがわかるであろう.

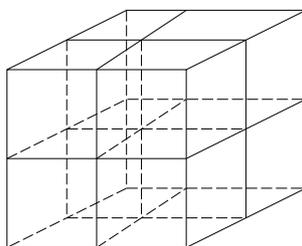
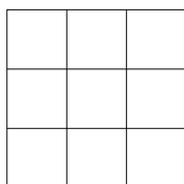


図 70: 正方形と立方体 (書くまでもないが, 念のため).

従って, これを拡張し, 対象とする図形を基本ユニットに分け, その中に含まれる点の数を対象図形 1 辺の長さ l の関数 $m(l)$ として表したとき, l と m の間に

$$m(l) = Al^{d_f}$$

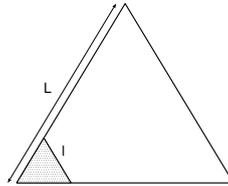
なる関係があったとき, この d_f を次元と定義することにしよう. ここで, 既に見たように, このような次元の定義は, 我々の経験則, つまり, 正方形は 2 次元, 立方体は 3 次元, に反しないことに注意されたい.

さて, 我々は常識的に次元というものは整数であると思い込んでいる. しかし, 我々が既に作図した 2 つの図形, シェルピンスキーガスケット, 菌糸の成長図は**非整数次元**をもつ. このような d_f を**フラクタル次元 (ボックスカウント次元)**と呼んでいる. フラクタル次元はフラクタル図形を特徴付ける最も基本的な量である. 逆にフラクタル次元が非整数値を持つような図形をフラクタル図形と呼ぶ.

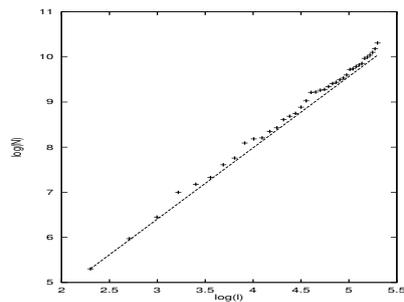
以下の課題で, それぞれの図形のフラクタル次元を具体的に求めてみよう.

課題 4

課題 1 で作図したシェルピンスキー・ガスケットにおいて



長さ l の正三角形に含まれる点の数を N とし, l を 1 から $L = l_{\max}$ までいくつか変化させ (上図), l と $N(l)$ を求める. 次いで, 縦軸に $\log N$, 横軸に $\log l$ をプロットし, その傾きからフラクタル次元 d_f を求めよ (例えば下図のように).



16 カオス編の演習

上記の課題 1 ~ 課題 4 とは別に, 既に終了している [カオス編] の演習課題 5 課題 6 をやってみよう。

課題 5

カオスの基本モデルとして次のエノン写像:

$$x_{n+1} = 1 - ax_n^2 + y_n \quad (241)$$

$$y_{n+1} = bx_n \quad (242)$$

が知られている. このエノン写像に対し

- (1) $a = 1.6, b = 0.2$ と選んだ場合に対し, x_n, y_n をプロットし, 初期値鋭敏性を調べよ. 時間があれば, パラメータ (a, b) の値を変え, 結果を考察せよ.
- (2) x - y 曲線をプロットし, アトラクタを描画せよ (描画例として図 71 を参照). また, gnuplot などの描画ソフトの描画領域を変化させることで, アトラクタの自己相似性を確認せよ.
- (3) 相関次元を求めよ. その際, 時間遅れ τ , 標本点の次元 m の組み (τ, m) をいくつか選び, 相関次元の計算結果を比較せよ.

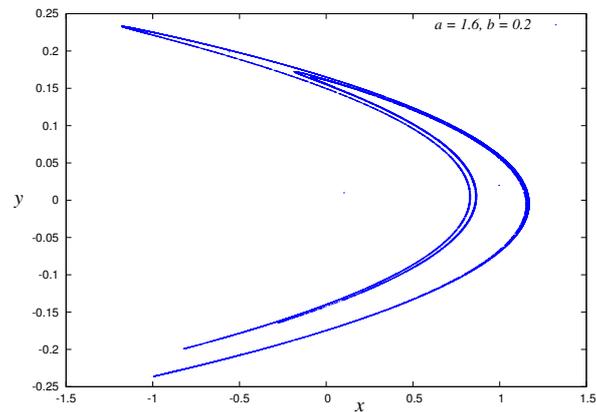


図 71: エノン写像のアトラクタの描画例.

課題 6

ロジスティック写像:

$$x_{n+1} = 4x_n(1 - x_n) \quad (243)$$

を適当な初期値からスタートさせ、次の時間平均値を計算せよ.

$$\bar{x} = \frac{1}{T} \sum_{n=0}^{T-1} x_n \quad (244)$$

ただし、 T が十分大きな数であり、各自が設定せよ.

一方、写像を十分繰り返した後、ヒストグラムを計算することで、ロジスティック写像は次の不変測度を持つことをみた.

$$P(x) = \frac{1}{\pi} \frac{1}{\sqrt{x(1-x)}} \quad (245)$$

この「密度関数」に対し、次の「平均値」を計算することもできる.

$$\langle x \rangle = \int_0^1 xP(x) = \int_0^1 \frac{1}{\pi} \frac{xdx}{\sqrt{x(1-x)}} \quad (246)$$

この積分を解析的、もしくは数値的に計算することで、 $\bar{x} = \langle x \rangle$ が成立するか否かを調べよ.

課題 6 に関する Remark:

平均値: $\langle x \rangle$ は次のようにして解析計算できる. この講義で度々用いてきた $x = \sin^2(\theta/2)$ なる変数変換を行うと、 $dx = \cos(\theta/2) \sin(\theta/2) d\theta$ であるから、積分区間に注意して

$$\langle x \rangle = \frac{1}{\pi} \int_0^\pi \frac{\sin^2(\theta/2) \cdot \sin(\theta/2) \cos(\theta/2)}{\sin(\theta/2) \cos(\theta/2)} d\theta = \frac{1}{2\pi} \int_0^\pi (1 - \cos \theta) d\theta = \frac{1}{2} \quad (247)$$

である. 従って、ここでは \bar{x} が $1/2$ に等しいか否かを調べればよい.

また、もし、 $\langle x \rangle = \bar{x}$ が成立するのであれば、被積分関数に区間 $[0, 1]$ で定義された $1/\sqrt{x(1-x)}$ を含む定積分:

$$I = \int_0^1 \frac{f(x)}{\sqrt{x(1-x)}} \quad (248)$$

を数値的に求めるための「ツール」として、ロジスティック写像を用いることができる. ここに、 $f(x)$ は任意の関数である. つまり、ロジスティック写像からの時系列 x_0, x_1, \dots, x_T に対し、

$$I = \lim_{T \rightarrow \infty} \frac{\pi}{T} \sum_{i=0}^{T-1} f(x_i) \quad (249)$$

がその数値積分値となる. これはロジスティック写像からの系列をある種の「乱数」と見なした場合の「モンテカルロ積分」と考えることができる (ここで、実用上は T を大きな整数値で置き換えることに注意).

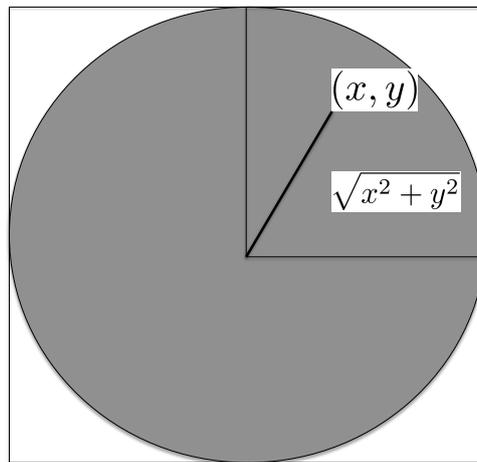
この数値積分の有効性や精度を調べるため、通常の「短冊近似」の数値積分と比較してみればよい。短冊近似の数値積分に関しては、計算機プログラミング I・同演習 (金曜クラス) で中間試験問題として出題した。その問題と解答例を参考にすると良いであろう。

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/PROG2009/ProgI2009_exam1.pdf

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/PROG2009/ProgI2009_exam1_ans.pdf

既に見たように、ロジスティック写像からの系列は一様乱数に似てはいるが、相対的に $x = 0, 1$ の近傍数値が出やすいため、一般にはモンテカルロ法 (積分) に適さない。例えば、これもまた計算機プログラミング I・同演習 (金曜クラス) で行った中間試験問題を思い出してみよう。そこでは次のような問題を出した (下記の問題文は当初のものから若干書き換えてあることに注意)。

円周率 π の計算法としては、古代から様々な方法が提案されてきたが、現代では計算機を用いることで、簡易にその値を見積もることができる。例えば、図に示したような半径 1 の 1/4 円とそれを取り囲む 1 辺が長さ 1 の正方形を考え、**この正方形の中にランダムに点を打ち込む**ことを考える。



このとき、1/4 円の内部に落ちる点の個数はその面積に比例するから

$$\frac{\text{半径 1 の 1/4 円の内部に落ちる点の個数 } n}{\text{1 辺の長さが 1 の正方形内部に落ちる点の個数 (= ランダム点を打ち込む総数 } N)} = \frac{\text{半径 1 の 1/4 円の面積}}{\text{1 辺の長さ 1 の正方形の面積}} = \frac{\frac{\pi}{4}}{1} = \frac{\pi}{4}$$

となり、ランダム点を打ち込む回数 N を増やすことで、円周率の 1/4 値が得られることが予想できる。そこで、上記のアルゴリズム (手続き) を具体的に C 言語のプログラムに書いて求めてみよう。

- (1) ここでは一様乱数に似た点 (x, y) , $x, y \in [0, 1]$ を生成するプログラムとして、次の初期値の異なる 2 つのロジスティック写像を一様乱数生成器の「代用」とする。

$$\begin{aligned}x_{n+1} &= 4x_n(1 - x_n), & x_0 &= 0.1 \\y_{n+1} &= 4y_n(1 - y_n), & y_0 &= 0.1001\end{aligned}$$

この写像から生成される点列 $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k), \dots$ がここでの「ランダム点」となる。ここでは、ランダム点の総数 N を $N = 100000$ 程度に選ぶ。原点 $(0, 0)$ からランダム点 (x_k, y_k) までの距離が 1 以内であれば、「半径 1 の $1/4$ 円内に落ちた」と見なすことにし、その総数を n としよう。上記の説明から n/N が $\pi/4$ の「近似値」を与えるはずである。結果を

pi/4=.....

の形式で表示するプログラムを作成せよ。

- (2) x_n の値を生成する関数を再帰的関数定義を用いて作りたい。初期値 $x_0 = 0.1$ からスタートさせた x_n の値をメイン関数の中に作成した for ループで

```
for(i=1, i<=N; i++){
  x = Logmap(i, x0);
}
```

のように使うための関数 Logmap(n, x0) を作成せよ。

この問題の解答例は以下の通りであった。

```
/****** 小問 (1) *****/
#include<stdio.h>
// #include<stdlib.h> /* rand() を使うときはコメント外す */
#include<math.h>
#define N 100000

main()
{
  int i, sum;
  double x, y, p;
  x=0.1;          /* 一様乱数使う場合にはコメントつける */
  y=0.0001;
  for(i=1, sum=0; i<=N; i++){
    x=4.0*x*(1.0-x);
    y=4.0*y*(1.0-y);
  }
  // x=(double)rand()/RAND_MAX; /* 一様乱数使う場合はコメント外す */
  // y=(double)rand()/RAND_MAX;
  if(sqrt((x*x)+(y*y))<1.0){
    sum++;
  }
}
```

```

    }else{
    sum= sum;
    }
    }
    p=(double)sum/N;
    printf("pi/4=%lf\n",p);
}

```

```

/***** 小問 (2) *****/
double Logmap(int n, double x0)
{
    if(n==1){
    return (x0);
    }else{
    return (4.0*Logmap(n-1,x0)*(1.0-Logmap(n-1,x0)));
    }
}

```

実際に上記 (1) のプログラムを実行し、点 (x_n, y_n) を 2 次元平面内にプロットしてみよう。この図 72 の上段は問題に与えられた漸化式を用いた場合の分布図であり、下段は C 言語で定義されている `rand()` という疑似乱数生成関数を用いたときの分布図である。右側が $N = 10^7$ 、左側が $N = 3 \times 10^7$ と選んである。この図より、疑似乱数を用いた場合は $1/4$ 円内に一様に偏りが無く点が分布している一方、ロジスティック写像を用いた場合には、相対的に $(x_n, y_n) = (0, 0), (1, 0), (0, 1)$ 周辺の密度が濃くなっている⁴⁰。つまり、 $1/4$ 円内に一様に点が分布するのではなく、偏りが生じてしまっている。この性質は既に見たようにここで用いたロジスティック写像特有のものである。円周率 π が精度良く求まるためには、この点の打ち込まれ方に偏りがあってはいけなから明らかに、漸化式を用いた π の近似値よりも、疑似乱数を用いた結果の方が良好であるべきであるが、実際、 $N = 3 \times 10^7$ の場合の結果を比較してみると、漸化式による π が 2.8082 程度である一方、一様乱数による結果は 3.1440 となる。このように、「一様乱数」として用いるには、ロジスティック写像はかなり精度が悪い。

しかし、被積分関数にロジスティック写像の不変測度である $1/\sqrt{x(1-x)}$ の因子が含まれるような定積分の近似解法として用いる場合には事情が変わってくる可能性がある。それを調べるのが、この課題 6 である。

なお、 x_n に関する漸化式と y_n についてのそれは完全に同じ形をしており、その違いは微小な初期条件のみである。従って、直観的には得られる点列も同じであり、点列は $y = x$ の直線上に分布しそうに思われるが、実際には上図からわかるように「ほぼ」一様に分布し、そうはなっていない。これは本講義で学んだカオスの持つ「初期値鋭敏性」が反映された結果であることはすぐに理解できるであろう。

注意:

課題レポートの提出期限は **8/12(金) 午後 5 時 (厳守)**、提出先は情報科学研究科棟 8-13 のポスト

⁴⁰ この図には書き入れていないが、 $1/4$ 円を外れた点も描き入れてみると、 $(x_n, y_n) = (1, 1)$ の密度も高くなっている。

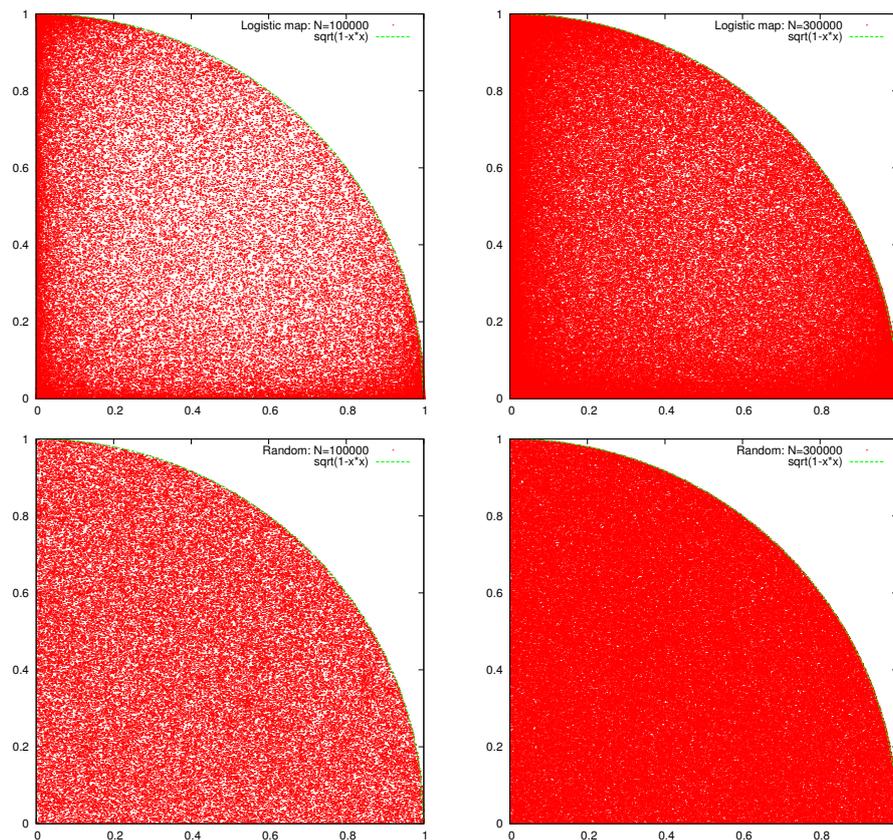


図 72: 上段は問題に与えられた漸化式を用いた場合の分布図であり, 下段は C 言語で定義されている `rand()` という疑似乱数生成関数を用いたときの分布図である. 右側が $N = 10^7$, 左側が $N = 3 \times 10^7$ と選んである.

まで. 全てのコードおよび, 結果のグラフ/図などを全て提出. また, 各課題に関し, 独自の考察などを含めても良い.

[カオス・フラクタル] 今後数回の予定:

- 7月19日: 通常講義. 「マルチ・フラクタル」について
- 7月26日: 最終回. 本講義に関連する研究紹介(1時間程度を予定).

なお, 後期開講の「情報工学演習 II」では, 本講義では扱うことのできなかつた, **臨界現象に関するフラクタル図形とその特徴付け**について学習することになります. 本講義はこの演習とのセットで, カオスとフラクタルについての基本的な知識が得られるように設計しました.

17 マルチフラクタル

今回の講義ではフラクタル図形をある種の「フィルタ」を介して見なおした場合に得られる「マルチフラクタル」という概念について学んでいく。

17.1 一般化次元の復習

第9回講義ノートでは相関次元をパラメータ q を用いて一般化した。今回はまず、それを思い出してみよう。半径 ϵ の球内に時系列の点が存在する確率を p_i とすれば、その一般化された相関次元 D_q は

$$D_q = \frac{1}{q-1} \lim_{\epsilon \rightarrow 0} \frac{\log \sum_{i=1}^{n(\epsilon)} p_i^q}{\log \epsilon} \quad (250)$$

で定義された。

この一般化された相関次元はフラクタル図形を特徴つける際にも用いることができる。例えば、対象となるフラクタル図形をメッシュに切り、各セルを長さ ϵ の立方体で定義する。このとき、各セル i に図形を構成する点が存在する確率を $p_i(\epsilon)$ と定める。このとき、上記の相関次元は $q=0$ の場合には簡単な計算から

$$D_{q=0} = - \lim_{\epsilon \rightarrow 0} \frac{\log \sum_{i=1}^{n(\epsilon)} 1}{\log(\epsilon)} = - \lim_{\epsilon \rightarrow 0} \frac{\log n(\epsilon)}{\log(\epsilon)} \quad (251)$$

となり、これは ϵ の立方体で対象図形を覆った場合 (図 73 参照)、セルの一边の長さ ϵ と、その長さを一辺とする立方体のセル群に含まれる点の総数 $n(\epsilon)$ の間の関係が

$$n(\epsilon) \simeq \epsilon^{-D_{q=0}} \quad (252)$$

であることを意味するので、この $D_{q=0}$ は対象図形のフラクタル次元 (容量次元, ボックスカウント次元) である。また、 $q=1$ のときには「情報次元」、 $q=2$ の場合には「相関次元」となったこと

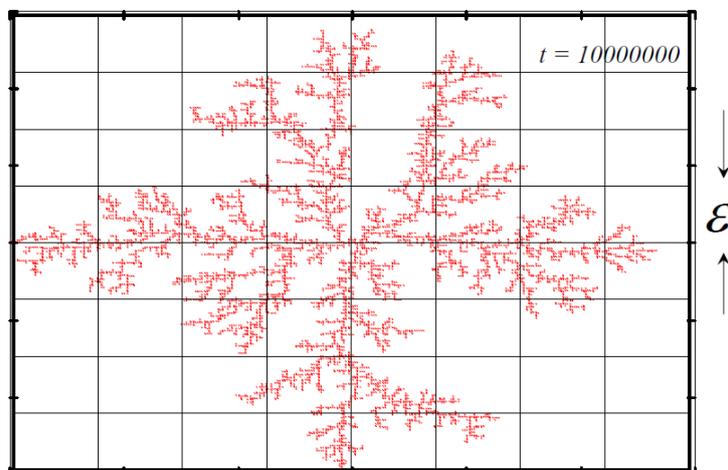


図 73: フラクタル図形をサイズ ϵ のセルで覆う。

を思い出そう。

17.2 パラメータ q で特徴つけられる「フィルタ」を介して図形を見る

ここで一般化次元の定義式 (250) から, $q > 1$ ならば点が存在する確率が比較的大きなセル i ほど, その確率は (250) 式の和の中で相対的に大きく寄与することになる. 従って, このようにフラクタル次元をパラメータ q を介して一般化することで, 対象とする図形の点の存在確率が大きいところと小さいところをそれぞれ強調してフラクタル次元を測りなおすことができることになる. つまり, q を大きくすることで, 点の存在確率が大きいところを強調し, 逆に q を小さくすることで点の存在確率が小さいところを強調してフラクタル性の度合いが測れることになる. フラクタル図形が複雑になればなるほど $q = 0$ で定義される次元ではなく, より広い q の値である種の「フィルタ」をかけて図形を捉え, その次元を考えた方が適切な場合が多い (単一の q では捉えられないほど複雑な図形)⁴¹. このようにパラメータ q を用いて一般化されたフラクタル次元 (250) を **マルチフラクタル次元** と呼び, そのような尺度で測られたフラクタルを **マルチフラクタル** と呼ぶ. 一方, ここで定義されたマルチフラクタル次元が用いたフィルタに依らない, つまり, パラメータ q の値に依存しない図形もあり, それを単にフラクタル, あるいは「モノフラクタル」と呼ぶ場合がある. ここで学ぶマルチフラクタルは自然現象のみならず, 株や為替の時系列などのデータ解析にも用いられる重要な概念となってきた⁴².

17.3 二項分岐過程とそのマルチフラクタル次元

ここではマルチフラクタルの概念とマルチフラクタル次元の計算方法に慣れるために, **二項分岐過程** と呼ばれる単純なプロセスに従って描かれる自己相似図形を考えよう.

17.3.1 描画アルゴリズム

このプロセスは図 74 のように一次元の数直線上 $[0, 1]$ に確率を割り当てる. この割り当て方は, まず, この線分を 2 分割し, その左側に p , 右側に $1-p$ を割り当てる. 次いで, この左側の $[0, 0.5]$ を

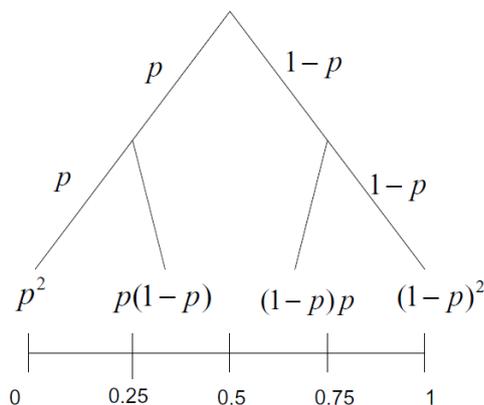


図 74: 二項分岐過程. $n = 2$ の場合.

⁴¹ 信号処理や応用数学で学んだかもしれないが, 周波数を制限して画像や音声を切り出す「低周波通過フィルタ」「高周波通過フィルタ」などを思い出す (比較してみる) と良い.

⁴² このようなマルチフラクタルでヒトの生体情報を計測し, その知見を工学に生かす「感性工学」という分野もある.

さらに 2 分割し、やはり、左側に p 、右側に $1 - p$ を割り当て、1 回目の分割でできた $[0, 5, 1]$ の区間についても同様に右側に p を、左側に $1 - p$ を割り当てる。このような割り振りのできる「2 分木」の末端にはそれぞれの経路（「枝」）に割り振られた確率の積が与えられる。従って、2 回目の分割では区間 $[0, 0.25], [0.25, 0.5], [0.5, 0.75], [0.75, 1]$ の 4 区間にそれぞれ $p^2, p(1 - p), (1 - p)p, (1 - p)^2$ が割り振られることになる（図 74 参照）。

このプロセスを n 回繰り返すと線分 $[0, 1]$ は 2^n 等分され、その中で確率 $p^k(1 - p)^{n-k}$ が割り振られている小区間は ${}_n C_k$ 個となる。このようにしてできる小区間を横軸、各小区間に割り当てられた確率を縦軸にプロットすると図 75 のようになる。この図中の組み込み図には区間 $[0, 0.1]$ を拡大

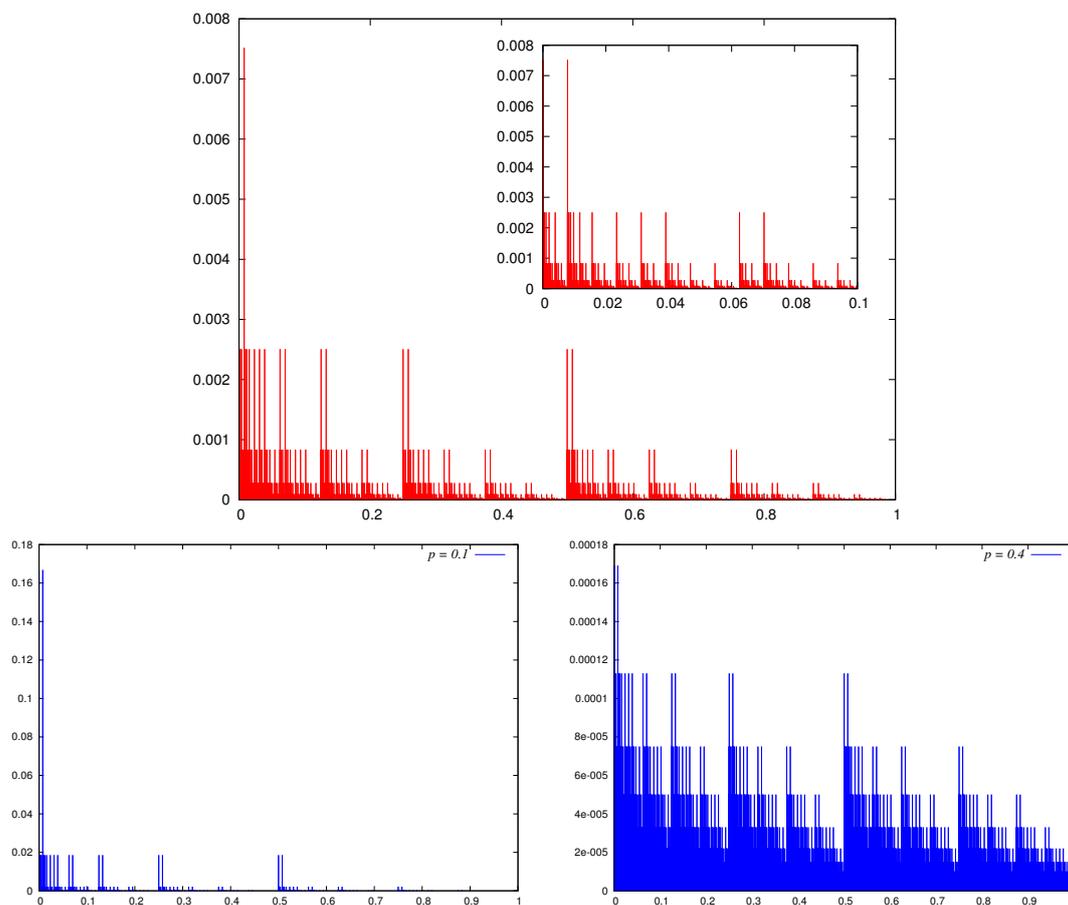


図 75: 二項分岐過程の密度分布. 上図では $p = 0.25$ と選んである. 上図の組み込み図は同図のスケールを変えてプロットしたもの. 下図はそれぞれ $p = 0.1, 0.4$ の場合の二項分岐過程の密度分布. $n = 18$ と選んでいる.

した部分が載せてある。この図からわかるように、この図形は「自己相似性」を持つ。

従って、具体的に計算機で実行するためのコーディングの際には「0」で 2 分岐の左側を「1」で右側を表すことに約束すると、 n ステップ後の 2 分木の末端（いわゆる「葉」）は $(000 \cdots 000)_2, (000 \cdots 001)_2, \dots, (111 \cdots 111)_2$ 等と 2 進数でコードされる。よって、 n 個の 0, 1 の並びのうち k 個の 1 を含む葉には確率 $p^k(1 - p)^{n-k}$ を割り当てることにし、この葉の位置する $[0, 1]$ 区間の値はこの葉の 2 進数 $(x_{n-1}x_{n-2} \cdots x_0)_2$, ($x_i = 1$ の個数が k 個) を 10 進数表示したものを 2^n でスケー

ルしなおせばよい。すなわち、この葉の $[0, 1]$ 区間内の位置 X は

$$X = \frac{\sum_{i=0}^{n-1} x_i 2^i}{2^n} \tag{253}$$

で与えられる。まとめると

$$(x_{n-1}x_{n-2}\cdots x_0)_2 = \underbrace{(0110\cdots 01)}_{k \text{ 個の「1」, } n-k \text{ 個の「0」}}_2 \rightarrow X = \frac{1}{2^n} \sum_{i=0}^{n-1} x_i 2^i \tag{254}$$

$$\rightarrow Y = p^k(1-p)^{n-k} \tag{255}$$

となる。この (X, Y) をプロットしたものが図 75 である⁴³。

17.3.2 マルチフラクタル次元の計算

このようにしてできる図形のマルチフラクタル次元を議論するために、 D_q が次の**分配関数**:

$$Z_q(\epsilon) = \sum_{i=1}^{n(\epsilon)} [p_i(\epsilon)]^q \tag{256}$$

を用いて

$$D_q = \frac{1}{q-1} \lim_{\epsilon \rightarrow 0} \frac{\log Z_q(\epsilon)}{\log \epsilon} \tag{257}$$

のように書き直せることに注意しておこう⁴⁴。ここで、上記の分配関数はデルタ関数 (もしくは、クロネッカのデルタ) を用いて次のように書き換えることができる。

$$Z_q(\epsilon) = \sum_p \sum_{i=1}^{n(\epsilon)} \delta(p_i(\epsilon) - p) [p_i(\epsilon)]^q = \sum_p \sum_{i=1}^{n(\epsilon)} \delta(p_i(\epsilon) - p) p^q \tag{258}$$

このとき、 $\sum_{i=1}^{n(\epsilon)} \delta(p_i(\epsilon) - p)$ は、 $n(\epsilon)$ のセルのうち p の確率を持つものの個数を意味するので、これを $\rho(p)$ と置けば、上記の分配関数 (258) はさらに次のように書き直すことができる。

$$Z_q = \sum_p \rho(p) p^q \tag{259}$$

つまり、ここで定義した $\rho(p)$ は確率 p の「縮退度」を表す。この分配関数を今考えている二項分岐過程に対して計算すると、 $p \rightarrow p^k(1-p)^{n-k}$, $\rho(p) \rightarrow {}_n C_k$ なる対応関係があることに注意して

$$\begin{aligned} Z_q &= \sum_{k=0}^n {}_n C_k [p^k(1-p)^{n-k}]^q \\ &= \sum_{k=0}^n {}_n C_k p^{qk} (1-p)^{q(n-k)} \\ &= \sum_{k=0}^n {}_n C_k (p^q)^k \{(1-p)^q\}^{n-k} = [p^q + (1-p)^q]^n \end{aligned} \tag{260}$$

⁴³ 数直線上には同じ Y を持つ X が ${}_n C_k$ 個あることに注意。また、 n の選び方には注意が必要である。担当者のパソコンでは $n = 18$ 程度までが我慢できるくらいの計算時間であり、また、これを超えると出来上がる図形の容量も大きくなる。

⁴⁴ ここで言う「分配関数」とは統計力学と呼ばれる方法論で用いられる用語である。パラメータ q を温度 T の逆数 $\beta = T^{-1}$ 、セル i に粒子が存在する確率をセル i のもつエネルギーを E_i として $p_i = e^{-E_i}$ とすると、 $Z_\beta = \sum_i [p_i]^\beta = \sum_i e^{-\beta E_i}$ がその分配関数であり、ここで議論するマルチフラクタルの一般論は統計力学との形式的類似点を持つ。

となる。よって今の場合、一つのセルサイズは $\epsilon = 2^{-n}$ であるから、二項分岐過程のマルチフラクタル次元は

$$D_q = \frac{1}{q-1} \frac{\log[p^q + (1-p)^q]^n}{\log 2^{-n}} = \frac{1}{q-1} \frac{\log[p^q + (1-p)^q]}{\log(1/2)} \quad (261)$$

で与えられる。いくつかの p に対して D_q の振る舞いをプロットすると図 76 のようになる。この

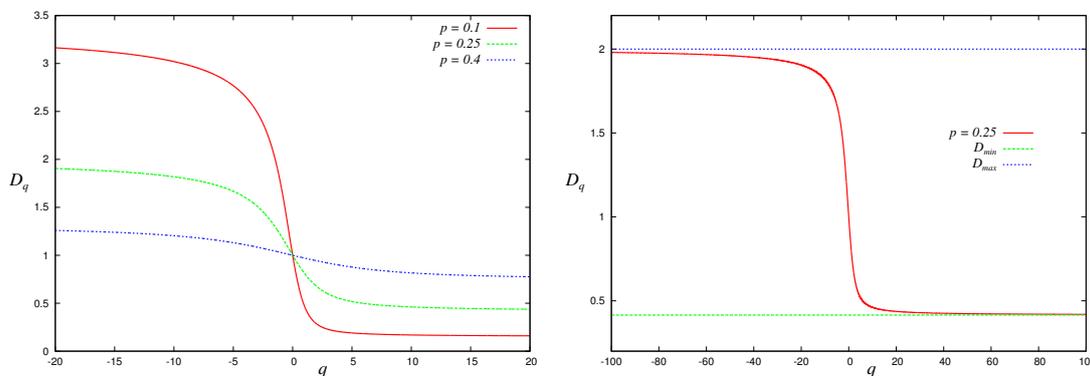


図 76: 二項分岐過程におけるマルチフラクタル次元 D_q . p の値は $p = 0.1, 0.25, 0.4$ に選んでいる。右図は $p = 0.25$ の場合についての D_q の $q \rightarrow \pm\infty$ での振舞い。それぞれ、一定値: $D_{\max} = \log p / \log(1/2)$, $D_{\min} = \log(1-p) / \log(1/2)$ に漸近していく。

図からわかるように、 q の値を -20 から $+20$ 程度のレンジで振ると、マルチフラクタル次元 D_q の値は「両端」でそれぞれある一定値に近づいて行き、結局、 $[D_{q=+\infty}, D_{q=-\infty}]$ の有限範囲に収まることがわかる。この有限値は (261) 式から次のように評価することができる。まず、 $p < 1/2$ ならば $q \rightarrow +\infty$ の場合、(261) 式右辺対数の 2 項のうち $(1-p)^q$ が支配的になるので

$$D_{q=+\infty} \simeq \frac{q}{q-1} \frac{\log(1-p)}{\log(1/2)} = \frac{\log(1-p)}{\log(1/2)} \quad (262)$$

であり、逆に $q \rightarrow -\infty$ の場合、(261) 式右辺対数の 2 項のうち p^q が支配的になるので

$$D_{q=-\infty} \simeq \frac{q}{q-1} \frac{\log p}{\log(1/2)} = \frac{\log p}{\log(1/2)} \quad (263)$$

である。これは図 76(右) で数値的にも確認できる。

この二項分岐過程で描かれるフラクタル次元は $q = -\infty$ で小さな確率で現れる点を強調して次元を測ることで大きくなり、逆に、 $q = +\infty$ で大きな確率で現れる点を強調して次元を測ると、フラクタル次元は小さくなる。これは大きな確率で得られる部分だけを描画すると、図形は「線状」あるいは「点」のみが強調された画像となるので、その次元は 0-1 間の「低次元」な画像になることから容易に予想できる結果である。

17.4 マルチフラクタル次元の一般論

前節で求めた二項分岐過程から得られるフラクタル図形に対するマルチフラクタル解析は、より一般的な形式で議論することができる。また、既に脚注で述べたようにパラメータ q を「逆温度」とみなした場合、ここでの一般論は「統計力学」と呼ばれる方法論と類似した数理構造を持っている⁴⁵。ここではそれについて簡単にみておこう。

⁴⁵ ただし、この講義では統計力学、および、それとの関係については触れない。

17.4.1 特異性の強さと大域スペクトル

まず, 長さ ϵ を持つセル i の中にフラクタル図形を構成する点が含まれる確率は ϵ の関数として

$$p_i(\epsilon) \propto \epsilon^{\alpha_i} \tag{264}$$

のようにセルサイズの減少: $\epsilon \rightarrow 0$ とともに減少していくことが予想される. ここで, その「減少の速さ」を表す α_i はセルごとに異なるが, ある密度を持って分布することになる. 具体的には $[\alpha, \alpha + d\alpha]$ の区間に α が存在する確率が

$$\rho(\alpha) \epsilon^{-f(\alpha)} d\alpha \tag{265}$$

で与えられるものとする. つまり, セルサイズに依存しない部分を $\rho(\alpha)$, セルサイズに依存する部分を $\epsilon^{-f(\alpha)}$ とおくことにする. ここで, このパラメータ α を**特異性の強さ**, 関数 $f(\alpha)$ を**大域スペクトル**と呼ぶ. このとき $\alpha \geq 0$ に注意すると, この系の分配関数は

$$Z_q = \int_0^\infty \rho(\alpha) \epsilon^{-f(\alpha)} [\epsilon^\alpha]^q d\alpha = \int_0^\infty \rho(\alpha) \epsilon^{-f(\alpha)+q\alpha} d\alpha \tag{266}$$

で与えられる. ここで, この (266) 式が分配関数の定義 (256) に則っているかを確認するためには

$$\rho(\alpha) \epsilon^{-f(\alpha)} = \sum_{i=1}^{n(\epsilon)} \delta(\alpha - \alpha_i) \tag{267}$$

を上式に代入すればよい. するとデルタ関数の性質に注意して直ちに

$$Z_q = \int_0^\infty \sum_{i=1}^{n(\epsilon)} \delta(\alpha - \alpha_i) \epsilon^{q\alpha} d\alpha = \sum_{i=1}^{n(\epsilon)} \epsilon^{q\alpha_i} = \sum_{i=1}^{n(\epsilon)} [p_i(\epsilon)]^q \tag{268}$$

が得られる. これは我々の分配関数の定義 (256) である.

さて, ここからマルチフラクタル次元を求めるためには, 具体的に Z_q に含まれる α の積分を実行しなければならないが, セルのサイズが $\epsilon \rightarrow 0$ となる極限では被積分関数のうちの $\epsilon^{-f(\alpha)+q\alpha}$ 肩の極小値で積分の値が支配されることに注意すればこの積分を精度よく実行することができる (鞍点近似). この極小値を与える α を $\alpha(q)$ とおくと, 極値条件より

$$\left. \frac{\partial}{\partial \alpha} \{-f(\alpha) + q\alpha\} \right|_{\alpha=\alpha(q)} = 0 \tag{269}$$

すなわち

$$q = \left. \frac{\partial f(\alpha)}{\partial \alpha} \right|_{\alpha=\alpha(q)} \tag{270}$$

が成り立つ. 従って, この条件の下で $-f(\alpha) + q\alpha$ は

$$-f(\alpha) + q\alpha \simeq -f(\alpha(q)) + q\alpha(q) + \frac{1}{2!} \left. \frac{\partial^2 f(\alpha)}{\partial \alpha^2} \right|_{\alpha=\alpha(q)} (\alpha - \alpha(q))^2 \tag{271}$$

と展開されるので (ここに, 積分を有限値にするため $\partial^2 f(\alpha)/\partial \alpha^2|_{\alpha=\alpha(q)} < 0$ であることに注意), ここでの分配関数を

$$Z_q = \rho(\alpha(q)) \epsilon^{-f(\alpha(q))+q\alpha(q)} \int_0^\infty \epsilon^{\frac{1}{2} \left. \frac{\partial^2 f(\alpha)}{\partial \alpha^2} \right|_{\alpha=\alpha(q)} (\alpha - \alpha(q))^2} d\alpha \tag{272}$$

のように書き直すことができる. ここに

$$\begin{aligned} \int_0^\infty \epsilon^{\frac{1}{2} \frac{\partial^2 f(\alpha)}{\partial \alpha^2} |_{\alpha=\alpha(q)} (\alpha - \alpha(q))^2} d\alpha &= \int_0^\infty \exp \left[\frac{1}{2} \log \epsilon \frac{\partial^2 f(\alpha)}{\partial \alpha^2} \Big|_{\alpha=\alpha(q)} (\alpha - \alpha(q))^2 \right] d\alpha \\ &= \sqrt{-\frac{\pi}{2(\log \epsilon) \frac{\partial^2 f(\alpha)}{\partial \alpha^2} |_{\alpha=\alpha(q)}}} \end{aligned} \quad (273)$$

であることに注意すると⁴⁶, マルチフラクタル次元 D_q は

$$\begin{aligned} (q-1)D_q &= \lim_{\epsilon \rightarrow 0} \frac{\log Z_q}{\log \epsilon} \\ &= \lim_{\epsilon \rightarrow 0} \frac{\log \rho(\alpha(q))}{\log \epsilon} - f(\alpha(q)) + q\alpha(q) + \lim_{\epsilon \rightarrow 0} \frac{\sqrt{-\frac{\pi}{2(\log \epsilon) \frac{\partial^2 f(\alpha)}{\partial \alpha^2} |_{\alpha=\alpha(q)}}}}{\log \epsilon} \\ &= -f(\alpha(q)) + q\alpha(q) \end{aligned} \quad (274)$$

となる. ここで $\lim_{\epsilon \rightarrow 0} (\log \epsilon)^{-1} = 0$, $\lim_{\epsilon \rightarrow 0} (\log \epsilon)^{-3/2} = 0$ を用いたことに注意されたい.
そこで, 簡単のため

$$\tau(q) \equiv (q-1)D_q \quad (275)$$

と定義すると

$$\tau(q) = -f(\alpha(q)) + q\alpha(q) \quad (276)$$

が得られる. 従って

$$f(\alpha(q)) = q\alpha(q) - \tau(q) \quad (277)$$

であるから

$$\begin{aligned} \frac{d\tau}{dq} &= \frac{\partial \tau}{\partial q} + \left(\frac{\partial \tau}{\partial \alpha} \right) \left(\frac{\partial \alpha}{\partial q} \right) \\ &= \alpha + \left\{ q - \frac{\partial f}{\partial \alpha} \right\} \frac{\partial \alpha}{\partial q} = \alpha \end{aligned} \quad (278)$$

が得られる. 従って, 以上をまとめると

$$f(\alpha(q)) = q\alpha(q) - \tau(q) \quad (279)$$

$$\frac{d\tau}{dq} = \alpha \quad (280)$$

が得られる⁴⁷. ここで, $\partial^2 f(\alpha)/\partial \alpha^2 |_{\alpha=\alpha(q)} < 0$ であったから, 関数 $f(\alpha)$ は上に凸の関数である.
また, $\alpha \geq 0$ と (280) 式から関数 τ は q の増加関数であるので (279) 式から

$$f(q\alpha) = q\alpha(q) - \tau(q) \leq 0 \cdot \alpha(0) - \tau(0) = -\tau(0) = D_{q=0} = f(0) \quad (281)$$

⁴⁶ ここで

$$\int_0^\infty \epsilon^{-b\alpha^2} d\alpha = \int_0^\infty e^{\log \epsilon^{-b\alpha^2}} d\alpha = \int_0^\infty e^{-b(\log \epsilon)\alpha^2} d\alpha = \frac{1}{2} \sqrt{\frac{\pi}{b \log \epsilon}}$$

を用いたことに注意.

⁴⁷ 応用数学で学習したと思うが, この関係 (279)(280) をもって「関数 f は関数 τ のルジャンドル変換である」という.
また, $df/d\alpha = y$ とおくと

$$P(y) = \alpha y - f = \alpha \frac{df}{d\alpha} - f = \alpha q - f = \tau$$

であるから, 「関数 τ は関数 f のルジャンドル変換」でもある.

が得られる. 従って, 関数 f は $q = 0$ のときに最大値 $D_{q=0}$ をとる.

また, 関数 $f(\alpha)$ は上に凸の関数なので, (270) 式から $q \rightarrow +\infty$ のとき $\alpha = \alpha_{\min}$, $q \rightarrow -\infty$ のとき $\alpha = \alpha_{\max}$ である. 従って, (280) より, $\tau(q) \simeq \alpha q$ として線形近似すると

$$\tau(q \rightarrow +\infty) \simeq \alpha_{\min} \cdot (+\infty) \quad (282)$$

$$\tau(q \rightarrow -\infty) \simeq \alpha_{\max} \cdot (-\infty) \quad (283)$$

に注意すると, $D_q = \tau(q)/(q-1)$ であるから $D_{q \rightarrow -\infty} = \{\alpha_{\max} \times (-\infty)\}/(-\infty - 1) = \alpha_{\max}$, $D_{q \rightarrow +\infty} = \{\alpha_{\min} \times (+\infty)\}/(+\infty - 1) = \alpha_{\min}$ であり, $\alpha_{\min}, \alpha_{\max}$ の両端で大域スペクトル f の値は

$$\begin{aligned} f(\alpha_{\min}) &= f(\alpha(q \rightarrow +\infty)) = (+\infty) \cdot \alpha(q \rightarrow +\infty) - \tau(q \rightarrow +\infty) \\ &= (+\infty) \cdot \alpha_{\min} - \alpha_{\min} \cdot (+\infty) = 0 \end{aligned} \quad (284)$$

$$\begin{aligned} f(\alpha_{\max}) &= f(\alpha(q \rightarrow -\infty)) = (-\infty) \cdot \alpha(q \rightarrow -\infty) - \tau(q \rightarrow -\infty) \\ &= (-\infty) \cdot \alpha_{\max} - \alpha_{\max} \cdot (-\infty) = 0 \end{aligned} \quad (285)$$

となることがわかる.

さらに, (270)(279) 式から

$$\begin{aligned} f(q(\alpha)) &= q\alpha(q) - \tau(q) \\ &= \left(\frac{\partial f(\alpha)}{\partial \alpha} \Big|_{\alpha=\alpha(q)} \right) \alpha(q) - \tau(q) \end{aligned} \quad (286)$$

より, 原点を通り, 傾き:

$$q = \left(\frac{\partial f(\alpha)}{\partial \alpha} \Big|_{\alpha=\alpha(q)} \right) = 1 \quad (287)$$

の直線: $f = \alpha$ と (279) 式の交点の α 座標を求めると, $(1-q)\alpha = -\tau(q) = (1-q)D_q$ より

$$\alpha = D_{q=1} \quad (288)$$

であり, このときの f は

$$f(\alpha(1)) = \alpha = D_{q=1} \quad (289)$$

である.

17.4.2 二項分岐過程再考

上で展開された一般論は当然, 個別の現象である二項分岐過程に対しても当てはまる. 具体的なケースを考えることで理解も深まるので, ここでは一般論をこの二項分岐過程に適用してみよう.

まずは (261) より, マルチフラクタル次元が

$$D_q = \frac{1}{q-1} \frac{p^q + (1-p)^q}{\log(1/2)} \quad (290)$$

で与えられたことを思い出すと, $\tau(q)$ は

$$\tau(q) = (q-1)D_q = \frac{p^q + (1-p)^q}{\log(1/2)} \quad (291)$$

である。従って, (280) 式は

$$\alpha = \frac{d\tau}{dq} = \frac{1}{\log(1/2)} \left[\frac{p^q \log p + (1-p)^q \log(1-p)}{p^q + (1-p)^q} \right] \quad (292)$$

となる。

ここで簡単のため

$$\xi = \frac{p^q}{p^q + (1-p)^q} \quad (293)$$

とおくと

$$1 - \xi = \frac{(1-p)^q}{p^q + (1-p)^q} \quad (294)$$

となることに気づけば (292) 式は

$$\alpha = \frac{1}{\log(1/2)} (\xi \log p + (1-\xi) \log(1-p)) \quad (295)$$

と書き直すことができ, 従って, (279) 式から大域スペクトルは

$$\begin{aligned} f &= \frac{q}{\log(1/2)} (\xi \log p + (1-\xi) \log(1-p)) - \frac{1}{\log(1/2)} \log(p^q + (1-p)^q) \\ &= \frac{1}{\log(1/2)} (\xi \log \xi + (1-\xi) \log(1-\xi)) \end{aligned} \quad (296)$$

として求めることができる。以上をまとめると, 二項分岐過程の特異性の強さ α と大域スペクトルは

$$\alpha = \frac{1}{\log(1/2)} (\xi \log p + (1-\xi) \log(1-p)) \quad (297)$$

$$f = \frac{1}{\log(1/2)} (\xi \log \xi + (1-\xi) \log(1-\xi)) \quad (298)$$

である。ここに, $0 \leq \xi \leq 1$ に注意して, いくつかの確率 p に対して大域スペクトルをプロットすると図 77(左上) のようになる。

ところで, $D_{q=0}, D_{q=1}$ の値, および, そのときの特異性の強さ α は次のようにして陽に求めることができる。まずは (293)(294) 式を辺々割ると

$$\frac{\xi}{1-\xi} = \left(\frac{p}{1-p} \right)^q \quad (299)$$

が得られる。従って, $q=0$ の場合には, $\xi=1/2$ であり, 従って, このときの f の値は (298) 式から $f = D_{q=0} = 1$ であり, これを与える α の値は (297) 式から

$$\alpha = \frac{\log p(1-p)}{2 \log(1/2)} \quad (300)$$

で与えられる。

一方, $q=1$ の場合には, (299) 式から $\xi=p$ であり, このときの大域スペクトル値, および, 特異性の強さは (298)(297) 式から

$$f = \alpha = \frac{p \log p + (1-p) \log(1-p)}{\log(1/2)} = D_{q=1} \quad (301)$$

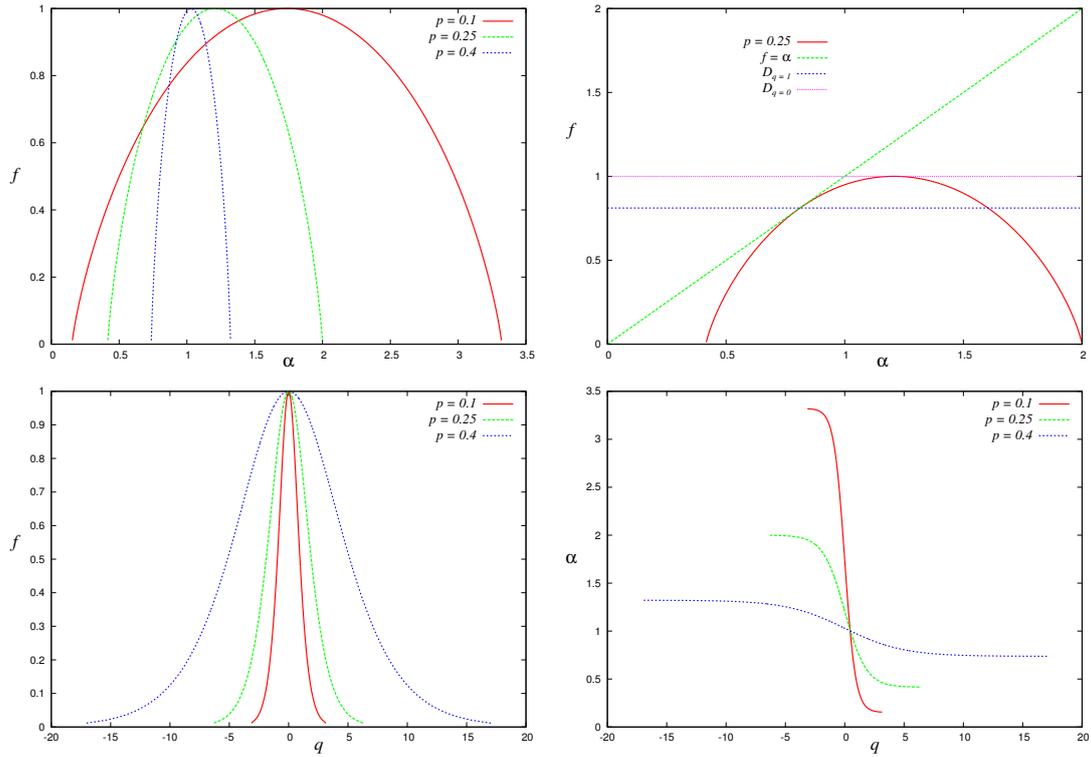


図 77: 二項分岐過程における f の α 依存性. p の値は $p = 0.1, 0.25, 0.4$ に選んでいる (左図). 右図は $p = 0.25$ に関し, f と $D_{q=0}, D_{q=1}$ の値の関係をプロットしてある. 下段は f の q 依存性 (左), および, α の q 依存性 (右).

で与えられる. この関係を $p = 0.25$ に対して図 77(右上) に載せた.

以上がマルチフラクタルとその次元の計算方法である. 興味のある者はこの講義で作図したいいくつかのフラクタル図形に対し, マルチフラクタル次元を q の関数として数値計算してみると良いであろう.

レポート課題

本講義で学習したカオスやフラクタルが役に立つような事例があればその理由とともに挙げ、A4用紙数枚程度で説明せよ。ちなみに、役に立たないと思う場合にその理由を書いても良い。なお、事例を多数あげるより、少ない事例に関しての考察が深い方が点数は良い。これと、**講義ノート #11** に載せた課題をまとめて提出する。

全てのレポートの〆切: **平成 23 年 8 月 12 日 (金) 午後 5 時 (厳守)**, 提出先は情報科学研究科棟 8-13 のポストまで。

連絡:

最終回である次週 7/26 は本講義に関連する「フラクタル」「スケールフリー性」に関連する担当者らの研究グループ (情報統計力学グループ):

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/major.html#group

の研究紹介を行います。



図 78: 上海. 平成 23 年 6 月 (井上撮影).

関連する参考書を何点か挙げておきます。下記リストのなかで入手困難なもので、読んでみたいものがあれば、数週間程度ならばお貸しできます。

参考文献

- [1] 「カオス入門：現象の解析と数理」長島弘幸, 馬場良和 共著 培風館 (1992).
- [2] *Fractals for the classroom*, by H. Peiger, H. Jurgen and D. Saupe, part I,II, Springer-Varlag (1993).
- [3] *Modeling reality: How computer mirror life*, by Iwo Bialynicki-Birula and Iwona Bialynicka-Birula (2004).
- [4] 「非線形力学とカオス: 技術者・科学者のための幾何学的手法」J.M.T. Thompson, H.B. Stewart 著, 武者利光, 橋口住久訳 オーム社 (1988).
- [5] 「カオス時系列解析の基礎と応用」池口徹, 山田泰司, 小室元政 共著, 合原一幸編, 産業図書 (2000).
- [6] 「フラクタル」 高安秀樹 著 朝倉書店 (1986).
- [7] 「フラクタル科学」高安秀樹 編著 朝倉書店 (1987).
- [8] 「フラクタル成長現象」タマス・ヴィチェック 著, 宮島佐介 訳, 朝倉書店 (1990).
- [9] 「フラクタル」 本田勝也 著 朝倉書店 (2002).
- [10] 「フラクタルと数の世界」西沢清子 他著 海文堂 (1991).
- [11] 「パーコレーションの科学」小田垣孝 著 嘗華房 (1993).
- [12] *Introduction to percolation theory : revised 2nd edition*, by D. Stauffer and A. Aharonoy, (Tayler & Francis, 1994).
- [13] 「パーコレーション：ちよつと変わった確率論入門」樋口保成 著 遊星社 (1996).
- [14] 「確率モデルって何だろう」今野紀雄 著 ダイアモンド社 (1995).
- [15] 「複雑系を解く確率モデル」香取真理 著 講談社ブルーバックス (1997).
- [16] 「秩序・無秩序の世界」黒田耕嗣 著 丸善 (1987).
- [17] 「カオスとフラクタル」山口昌哉 著, ちくま学芸文庫 (1986).

カオス・フラクタル 2011 ガイダンス 配布資料

担当：井上 純一 (情報科学研究科棟 8-13)

平成 23 年 4 月 12 日

1. この講義の目標

複雑系研究の代表的なトピックスである「カオス」「フラクタル」の基本的概念を習得する。特に、複雑なシステムを非線形力学系、セル・オートマトン等で計算機上に再現し、そこに現れる複雑な系の挙動・性質がカオスやフラクタルの概念によって特徴付けられることを、2 年次に既習の初等的な C 言語プログラミングを実際に行ってもらうことで学習する。

2. 到達目標

カオス、フラクタルの概念を簡単な系の数値実験 (シミュレーション) 例を通じて習得する。その際、それらの特徴付けるリアプノフ指数、フラクタル次元などの物理量の計算手法のいくつかを学び、実際に計算機上でこれらの系をシミュレートし、特徴量を評価するための数値計算技法もあわせて学習する。また、時間があれば、これらカオス、フラクタルの応用例についても触れたいが、当講義は時間的にみて (半期の全 15 回前後)、カオス・フラクタルの「入門編」にとどまるため、より進んだ内容 (工学的応用例を含む) は大学院情報科学研究科複合情報学専攻において開講されるであろういくつかの講義・演習等で学ぶことになる (例えば、混沌系工学特論)。

※ なお、当講義で学んだ内容を定着させるために、後期に開講される「情報工学演習 II」では、カオス・フラクタルに関連する、ある程度まとまった計算機実験をしてもらう。

3. 対象学生

工学部情報工学コース 3 年生 (必修科目)。 ※ 基本的にどなたの受講も歓迎です。

4. 授業計画

大まかに言って当講義は I. カオス編, II. フラクタル編の 2 つのパートに分かれる。

I. カオス編

- (1) 初等力学 (物理学 I) の復習, 運動方程式 (微分方程式) の解析的解法, 数値的解法
- (2) 非線形項の入った運動方程式の数値的解法, 写像力学系 (ロジスティック写像など)
- (3) カオスの特徴付け (リ・ヨークの定理, リアプノフ指数, 位相エントロピーなど)
- (4) 位相空間, 奇妙なアトラクタ, リアプノフ指数計算, 高次元データの低次元への埋め込み
- (5) カオス応用 (時間があれば (多分, ない.), カオス・ニューラルネットを用いた最適化など)

II. フラクタル編

- (1) 自然界に現れる様々なフラクタル図形と非フラクタル図形 (「特徴的な長さ」の有無, 「スケール・フリー性」について)
- (2) 確定的なルールで作られるフラクタル図形 (コッホ曲線, シェルピンスキーガスケット)
- (3) 確率的なルールで作られるフラクタル図形 (セル・オートマトン, 菌糸成長など)
- (4) 複素力学系とフラクタル (ジュリア集合など)

- (5) フラクタル次元とその数値計算法 (ボックスカウント次元など)
 - (6) カオスとの関係について
 - (7) フラクタル応用 (時間があれば (多分, ない.), 情報圧縮など)
- 以上を各項目 1-2 回程度で行う.

5. 成績の付け方

各回に簡単なレポートを出す. 全てのレポートの内容で成績をつける. レポートの提出状況などは逐次, 学籍番号にて講義ホームページに掲示し, 各自がチェック可能にしておく. 各先生からの成績の平均をとって総合成績とする.

6. 講義の進め方

毎回配布の講義ノートに沿って授業を進める. それに基づき (スライド/板書で) 解説する.

7. 注意事項

初等力学 (物理学 I) の内容, 常微分方程式, 確率・統計の基本的な考え方, 複素数などを良く復習しておくこと. また, 当講義ではシミュレーションを介して直観的にカオス・フラクタルを理解してもらうために, 計算機によるプログラミングを頻繁に行ってもらおう. また, 毎週のレポート課題もその手の問題を出します. このため, 当講義を受講する前に 2 年次に履修したであろう「計算機プログラミング I,II および同演習」で学んだプログラミング技法を各自が思い出し, 基本的事項を十分に復習しておくこと. 当講義内で別途に C 言語プログラミングの復習のための時間を確保することは難しいですから, 忘れている場合, 講義そのものについていけない可能性があります. ご注意ください.

8. 担当教員の連絡先

電話 : 011-706-7225

電子メール : j_inoue@complex.ist.hokudai.ac.jp

講義ホームページ:

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/ChaosFractal2011/ChaosFractal2011.html



図 1: 米フォートワース・ストックヤード (平成 23 年 3 月). 担当者撮影.