

## 計算機の中のカオス

山口明宏

北海道大学工学部 システム情報工学専攻 混沌系工学分野

〒060 札幌市 北区 北13条西8丁目

(e-mail: aki@complex.hokudai.ac.jp)

### 概要

カオスの力学系を計算機シミュレーションするとき計算機内部に系を模倣する有限要素のダイナミクスが形成される。この有限のダイナミクスが元の力学系とどのように対応するのか、また、計算の有限性が計算結果にどのように影響するのかを、シフト写像および Open Flow System (OFS)[1] を例に議論する。

### 1 導入

カオスの力学系において計算機を使った系の模擬実験の役割りは、非常に大きくなってきている。特に解析解が直接的に得られない場合には、計算機実験を主体にした研究が多くなされている。

特に“複雑系”と呼ばれる分野においては計算機による系の模倣実験が可能になってきたからこそ研究が発展してきたと言える面もあり、計算機の役割は非常に大きい。

そして、高速な計算資源の安価な供給に伴いこれまで行うことができなかつた大規模な数値実験も行え、ますます高度な計算機実験がおこなえるようになりつつある。

が、ここに至って計算機実験に関していくつかの不安(不信)を感じている方もおられるのではないだろうか。計算機実験については、

『計算結果は、正しいのだろうか?』

という疑問は常につきまとっているはずである。そして、研究を行っていくためには計算結果がどのような前提のもとで、どの程度正しいのかということは、必ずあきらかにしておかなければならない。このように、正しさに関する問いが生じるのは計算機による数値実験解に対して、解析的な真の解の実在というものを想定しているためである。

だが、この真の解とはいったい何者なのだろうか。我々の多くは、対象とする系を記述するために、力学系という記述言語をもちいている。そして“自然”に存在する現象などを記述しようとするとき、実数の上での連続力学系(微分方程式)をもって記述を行う。このとき真の解は、自然に存在する現象であり力学系で記述された系の表現は、力学系という言葉による真の解の模倣である。

このように真の解というものの実体が現実実在している場合には、それとの比較という意味において計算結果の正しさというものを検証することができる(見える範囲で)。

だが、力学系で記述された系とそれを模倣する計算機上に構成された系との比較を行う場合には“力学系で記述された系”の真なる軌道に関する知識が我々には無い。そのため、二つの系の振舞いを単純に比較せよとは言えない。運良くなんらかの系に固有な量を解析的に計算できた場合(エネルギー、エントロピー等)には、それらを指標として、計算結果の正しさを信じることになる。しかし、それらも検証の一部でしかなく計算結果が絶対に正しいということを保証することはない。

結局は、系の巨視的なふるまいに関する“定性的な直観”に反しない範囲で計算機実験の結果を受け入れているのが現状ではないだろうか。

この小論では、このような問題について興味深い例を二つ示す。まず2節で、実際に力学系を計算機上に表現するときに計算機内部ではどのような機械機構が実現されているのかということについて、シフト写像を例に本来の力学系としての機構と計算機上に実現される機構の違いを議論する。次に3節で、力学系を計算機上の機械機構として実現したが故に顕在化する現象の例として、一方向性の Open flow system[1] における spatial bifurcation[1] と呼ばれる分岐現象の機械的生成機構を示す。そして、これらの例をもとに4節で系を記述することについて議論する。

## 2 シフト写像

まず、なんらかの連続な力学系(無限精度)を計算機上に実現する場合、計算機は有限個の状態しか扱えないため、状態の離散化(有限精度)が必要になる。

この節では、このような無限精度の力学とその近似である有限精度の力学について、系の統計的性質は再現されるが、その再現機構は全く異なる例としてシフト写像を取り上げる。

シフト写像は、

$$f(X) = 2X \text{ mod } 1, \quad X \in [0, 1) \quad (1)$$

と定義され、区間  $[0, 1)$  の数を2倍してその値が1以上ならば1を引く写像である。そして、以下に示すように区間  $[0, 1)$  の数を2進表現した時にその表現を左へシフトしたものに対応する。

区間  $[0, 1)$  の数  $X$  を、2進小数で

$$X = (0.b_1b_2b_3\dots)_2 = b_1 \times 2^{-1} + b_2 \times 2^{-2} + b_3 \times 2^{-3} + \dots, \quad b_i \in \{0, 1\}$$

と表現する。以下2進数を表わす場合には、 $(0.b_1b_2b_3\dots)_2$  のように括弧でくくり添え字2をつける。

この表現を用いると  $\frac{1}{3}$  は、 $\frac{1}{3} = \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \dots = \sum_{i=1}^{\infty} 2^{-2i} = (0.010101\dots)_2$  と表現される。この2進表現に基づいてシフト写像を書きなおすと、

$$\begin{aligned} X &= (0.b_1b_2b_3\dots)_2, \\ f(X) &= 2X \text{ mod } 1 = 2\left(\sum_{i=1}^{\infty} 2^{-i}b_i\right) \text{ mod } 1 \\ &= 2^0x_1 + \sum_{i=2}^{\infty} 2^{-i+1}b_i \text{ mod } 1 \\ &= \sum_{i=2}^{\infty} 2^{-i+1}b_i \\ &= (0.b_2b_3b_4\dots)_2 \end{aligned}$$

となり元のビット列を左にシフトして左端を取り去ったものに対応する。この計算を小数点以下  $m$  桁の有限の精度で計算する場合には、無限精度の数を有限精度の数に近似する必要がある。これを、ここでは小数点以下  $m+1$  桁以降を切り捨てることで行う。無限精度数の数  $X$  の精度  $\delta = 2^{-m}$  での近似値  $\check{X}$  を、

$$\check{X} = \sum_{i=1}^m b_i 2^{-i} \approx \sum_{i=1}^{\infty} b_i 2^{-i} = X$$

とする。このとき数  $X$  とその近似  $\tilde{X}$  の誤差  $\varepsilon$  は、

$$0 \leq \varepsilon = X - \tilde{X} < \delta$$

である。計算機内部において数は、ビット表現され実数の計算は浮動小数点数で近似し計算するが、ここでは、簡単のために固定小数点数による近似を用いる。無限精度の数  $\frac{1}{3} = (0.010101\cdots)_2$  の精度  $\delta = 2^{-4}$  での近似は、小数点以下 5 桁目以降を 0 と置いたものに対応して  $(0.0101000\cdots)_2$  となる。この精度  $\delta = 2^{-m}$  での近似の操作は、無限精度の数の  $m+1$  桁以降に関する知識を無にする操作である。そのため  $m+1$  桁以降は 0 と仮定される。この有限桁の計算のための仮定から生じた数 “0” と無限精度を表現するために与えられた数を区別するために仮定による数には、“ $\checkmark$ ” マークを付けることにする。例えば  $\frac{1}{3}$  は、精度  $\delta = 2^{-4}$  の近似で

$$\frac{1}{3} = (0.010101\cdots)_2 \approx (0.0101)_2 = (0.0101\checkmark\checkmark\checkmark\checkmark\cdots)_2$$

仮定

と表現される。また  $(0.b_1b_2b_3\cdots)_2$  で与えられる数は、

$$(0.b_1b_2b_3b_4\cdots)_2 \approx (0.b_1b_2b_3b_4\checkmark\checkmark\checkmark\checkmark\cdots)_2$$

仮定

と近似される。この近似数にシフト写像  $f(X)$  を適用すると

$$f((0.b_1b_2b_3b_4\checkmark\checkmark\checkmark\checkmark\cdots)_2) = (0.b_2b_3b_4\checkmark\checkmark\checkmark\checkmark\cdots)_2$$

$f(X)$  を 4 回 (有効桁数) だけ適用すると

$$f(f(f(f((0.b_1b_2b_3b_4\checkmark\checkmark\checkmark\checkmark\cdots)_2)))) = (0.\checkmark\checkmark\checkmark\checkmark\checkmark\checkmark\checkmark\checkmark\cdots)_2$$

無限精度の数の場合には、結果は  $(0.b_5b_6b_7b_8\cdots)_2$  となるが、有限の計算においては初期値として与えられた情報は、有効桁数に比例する回数 of 写像の適用で失われる。そして、その後は有限桁の計算を行うために用いた仮定に依存する数となる。このことはカオスのように微細な誤差を拡大する写像の場合、その計算によって計算機の中の微細なスケールの中にある仮定をも抽出することを示唆する。シフト写像の測度を有限精度の計算によって得られる軌道の時間平均を用いて計算する場合を考えると、計算される軌道は  $(0.\checkmark\checkmark\checkmark\checkmark\checkmark\checkmark\checkmark\checkmark\cdots)_2$  の系列であるから、写像の測度は  $X=0$  において  $\delta$  関数的スパイクを持つものと計算されてしまう。これに対して数を無限精度とした理想的計算の場合、計算される系列  $X_0, X_1, X_2, \dots$  は、

$$X_0 = (0.b_1b_2\cdots)_2, \quad X_i = (0.b_{i+1}b_{i+2}\cdots)_2 = \sum_{j=1}^{\infty} 2^{-j} \times b_{i+j}; \quad i > 1.$$

となる。実数の多くは 2 進小数表現するとランダムなビット列であることから、無限精度の計算によって得られる系列  $X_0, X_1, X_2, \dots$  は、区間  $[0, 1)$  を一様におおう。よって、理想的には測度は区間  $[0, 1)$  で一様なものとなるはずである。しかし、有限の計算による結果はこれに一致しない。実際に計算機の上で乗数を 2 としてシフト写像を計算すると、一様な測度は得られない。

このように数の表現を有限に制限したことで、有限の計算で結果を得ることが可能になる反面、制限に伴う仮定が計算結果に影響を与える場合があることがわかる。ここで強調しておきたいのは、影響を受けているにもかかわらず解析解と一致する結果をもたらす場合があることである。実際、計算機上の計算において、乗数を 2 ではなく、 $2 - 10^{-5}$  や  $2 + 10^{-5}$  として計算を行うとほぼ一様な測度を得られる。このとき問題となるのは、有限精度の数を表す有限桁のビット列から、無限精度の数を表すビット列をいかに作り出すかということである。

計算機による計算では、記述できる数だけでなく計算の過程も有限である。シフト写像の例では、2倍の演算を数の2進表現に合せてビット列のシフト演算で実現できた。しかし、3倍の演算を行う場合には単純にシフト演算で表現することはできない(2進表現の場合)。3倍の演算を行う場合には、右にシフト(2倍)したビット列にもとのビット列を加えるという操作が必要になる。この場合、桁上げの操作などが計算の途中で発生するために計算は有効桁の最下位から計算を初めなければならない。被乗数が無限精度の数の場合には、有効桁の最下位は無限の下位にあるために計算を開始することができない。そのために演算にも計算の有限性にもなう仮定が含まれる。2倍の演算の場合には、小数点の位置を左に1つずらす操作だけで演算が実現できたのだが、この操作と全ての桁の値を左にずらすという操作は結果は同じとみることでもできる。後者の場合には、全ての桁において値を左にずらす操作が同時に行われるならば一回の操作で2倍の演算が実現されるが、その場合シフトの命令を全ての桁をどのようにして伝えるのかという問題が残る。この2倍の操作を前者は局所的な操作のみで実現しているのに対して、後者は全域的な操作を必要としている。これらの操作は、はたして本当に同じなのだろうか? このことは現実の現象のなかに乗算がどのようにして実現されているのか、もしそのようなものが存在するとき我々は、どのような理由でそれを乗算と見做すのかという疑問につながる。ここで示されることは、演算を有限にとどめるためには演算自体の中にもなんらかの仮定が含まれるということだけである。

その影響をみるためにシフト写像の乗数自体にも有限の仮定を導入し、乗数を2から  $(a_1 a_2 . a_3 a_4 \check{0} \check{0} \dots)_2$  に変更する。このときシフト写像は、

$$\begin{aligned} f((0.b_1 b_2 b_3 b_4 \check{0} \check{0} \dots)_2) &= (a_1 a_2 . a_3 a_4 \check{0} \check{0} \dots)_2 \times (0.b_1 b_2 b_3 b_4 \check{0} \check{0} \dots)_2 \text{ mod } 1 \\ &= a_1 \times (b_1 . b_2 b_3 b_4 \check{0} | \check{0} \check{0} \check{0} \dots)_2 + \\ &\quad a_2 \times (0 . b_1 b_2 b_3 b_4 | \check{0} \check{0} \check{0} \dots)_2 + \\ &\quad a_3 \times (0 . 0 b_1 b_2 b_3 | b_4 \check{0} \check{0} \dots)_2 + \\ &\quad a_4 \times (0 . 0 0 b_1 b_2 | b_3 b_4 \check{0} \dots)_2 + \\ &\quad \check{0} \times (0 . 0 0 0 b_1 | b_2 b_3 b_4 \check{0} \dots)_2 + \dots \text{ mod } 1 \end{aligned}$$

となる。この場合にも写像の計算において有効桁の部分(式中の縦線)に仮定の影響があらわれる。しかし、この影響は乗数を2とした場合のように常に“0”ではなく被乗数を下位桁へのフィードバックしたものと加算されたものになっている。このようにして被乗数の巨視的スケール(上位桁)の情報が桁上げされて消滅するだけでなく、微視的スケール(下位桁)にフィードバックされ計算の仮定と加算(混合)されて新しい下位桁を決定しているのである(この決定には上位桁の情報だけでなく全ての桁の情報と計算の仮定がかかわっている)。

具体的に、乗数  $a = 2 + 10^{-5}$ 、有効精度25桁の場合、乗数は2進表現すると

$$a = (10.00000, 00000, 00000, 01010, 01111, \check{0}\check{0}\check{0}\check{0}\check{0}\dots)_2$$

となり、小数点以下1~15桁目まではもとの力学( $a=2$ の場合の単純なシフト)法則にしたがい、16~25桁目で被乗数の上位桁の情報を下位桁にフィードバックし計算の仮定との混合する機構を形成する。この混合の機構は、 $a$ の微視スケールにあるビットパターンで決定される。この場合、混合の機構が疑似乱数生成系として機能するため、巨視的に無限精度のシフト写像に近い計算が近似できたように見える。つまり、有限計算のシフト写像は、その微視的スケールに疑似乱数生成系を形成するところで、0,1のランダムな片側無限列を逐次的に生成し、そのビット列を巨視的スケールでシフトするところで、見かけ上シフト写像を実現している。

このようにしてシフト写像の場合、巨視的には、もとの力学法則に従っている有限精度の力学系を構成することができる。しかし、その計算の機構は微視的スケールでは無限精度の計算とは、まったく別のものなのである。

### 3 Open flow system

前節では、有限の計算でどのように無限精度の計算を近似しているかという問題について、やや病的な側面を強調するかたちで議論を展開した。

この節では、有限の計算として実現したが故に、ある特徴的な振舞い(本来の系には存在しない)が顕在化する例として、Open flow system [1]における、spatial bifurcation[1]を示す。そして、対象とする系に有限計算の仮定を組み入れたときに系の構造がどのように変わるのかということについて議論する。

Open flow system(OFS)は、金子 [1]によって導入された一次元離散力学系の一方向性の結合系である。その時間発展規則は、

$$\begin{aligned} x_{i+1}^1 &:= (1-\epsilon)f(x_i^1) + \epsilon f(x^0), \\ x_{i+1}^k &:= (1-\epsilon)f(x_i^k) + \epsilon f(x_i^{k-1}) \quad (k=1, 2, \dots) \end{aligned} \quad (2)$$

と定義される。ここで  $f: \mathcal{R} \rightarrow \mathcal{R}$ ,  $\epsilon$  は、上位サイトとの結合係数、 $x_i^k \in \mathcal{R}$  は、サイト  $i$ , ステップ  $t$  での系の状態、 $x^0 \in \mathcal{R}$  は、定数である。OFSは、その軌道として時空間パターンを生成する。その中で各サイトで周期軌道を取り、その周期が空間方向に倍分岐していく現象 (spatial bifurcation) が、金子によって発見された [1](図 1)。

ここで問題となるのは、安定な定常解が解析的に示され、空間方向に一旦は、その定常解に収束する(図 1 のサイト 20 程度まで)にもかかわらずその定常解が空間方向に不安定化し、周期倍分岐が起きる(計算機実験で)ことである。これは、直観的に非常に奇妙な現象で、どのようにして一旦収束した、安定な定常解から、分岐(不安定化)を引き起こすのかという疑問が生じる。

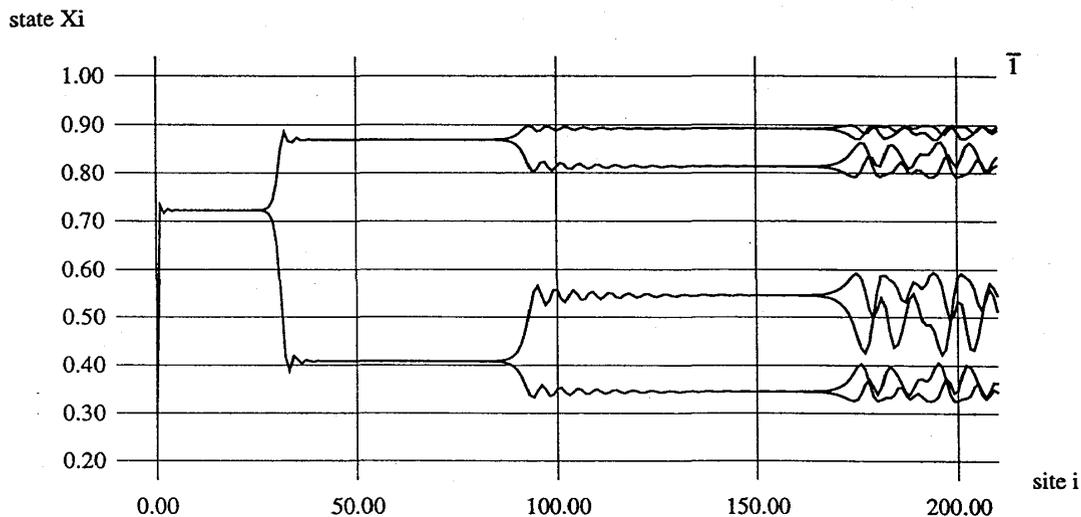


図 1  $f$ として logistic 写像  $f(x) = Ax(1-x)$  をもちいた OFS の空間方向分岐現象。  $A = 3.6, \epsilon = 0.55, x^0 = 0.3$ . 横軸が空間(サイト)方向、縦軸が各サイト  $i$  での状態  $x_i^t$ . 図では、ステップ  $t = 20003, \dots, 20019$  での系の状態を重ねて描画している。サイト 20 まで、系固有の一周期解への収束がみられ、収束後、二周期へ分岐している。そして、その後さらに、系固有の二周期解への収束、四周期への分岐が生じる。ここでは、系固有の周期解への収束は、四周期までで、八周期以降は、系固有の周期解への収束はなく、空間方向に準周期的に周期解が位置する。

この疑問について結論を先に言うと、計算機上での OFS には解析的に得られる定常解は存在せず(つまり計算機で安定定常解を記述できない)、その近似解のみが存在する。そして、この近似解と解析解との誤差が空間方向の結合により連鎖することで、定常解近傍からの分岐(不安定化)が可能となる。以下に、空間方向分岐の生成機構を示す。

### OFS の定常解

OFS が時間方向に周期軌道をとる場合、各サイトでの周期軌道の位置に関する連立方程式を解き解析解を求める事ができる [2]。ここでは、写像  $f$  固有の固定点  $x^*$  の近傍にある時間方向 1 周期解を求める。OFS は、各サイト  $i$  で系の状態が、固定点  $x_{*i} = x^* + \eta_i$  にあるとすると、各固定点の位置は、時間発展式 (2) から得られる方程式、

$$\begin{aligned} x_{*1} &= (1 - \epsilon)f(x_{*1}) + \epsilon f(x^0) \\ x_{*2} &= (1 - \epsilon)f(x_{*2}) + \epsilon f(x_{*1}) \\ &\vdots \\ x_{*^{i+1}} &= (1 - \epsilon)f(x_{*^{i+1}}) + \epsilon f(x_{*^i}) \\ &\vdots \end{aligned} \tag{3}$$

を満たす。これを  $x^*$  のまわりで展開して、サイト方向の固定点の位置の写像

$$\begin{aligned} \eta^{i+1} &= \gamma \eta^i, \\ \gamma &:= \frac{\epsilon f'(x^*)}{1 - (1 - \epsilon)f'(x^*)} \end{aligned} \tag{4}$$

を得る。 $|\gamma| < 1$  の時、各サイトでの固定点  $x_{*i}$  は、サイト方向に写像  $f$  固有の固定点  $x^*$  に収束する。この固定点の系列  $x_{*1}, x_{*2}, \dots$  は、OFS の定常解のひとつである。

実際、計算機実験により得られる OFS の振舞いは、上位サイト (図 1 のサイト 20 程度まで) では、この解析解に一致する。しかし、計算機上の OFS はその後、サイト方向に 2 周期に分岐する。この分岐は、写像  $f$  固有の 2 周期解に収束する。その後、さらに 4 周期に分岐し、以下系固有の周期解への収束、分岐を繰り返す。

### 固定点の記述不可能性

この分岐現象の生成機構を解明するためには、この一旦収束した定常解からの分岐の起源を知ることが必要である。解析解には、分岐の要素は全く無いので、計算機内部にどのような有限の力学が形成されたかが問題となる。そこで、

- 各サイトでの正確な周期
- 各サイトでの正確なセルマップ

を計算機内部の OFS について計算する。その結果を図 2、図 3 に示す。図 2 より、すでにサイト 1 において系は 2 周期解になっていることがわかる (サイト 0 は、定数)。サイト 1 での 2 周期解近傍の正確なセルマップを図示したものが、図 3 である。ここで、各サイトでのセルマップとは系の状態をビット表現の意味での各サイトでの時間発展写像をいう。図 3 では、倍精度 (IEEE 754) で表現した数をその有効数字 (仮数部) 最下位 8 ビットで番号付けて写像として図示したものである。図 3 において、無限精度の定常解に対応する固定点が存在しているが、その周りに 2 周期解が存在している。そのため固定点に収束する軌道は、ほとんどこの 2 周期解にトラップされ、固定点には到達しない。この 2 周期解は有限精度への近似誤差によるもので、無限精度の系には存在しない。

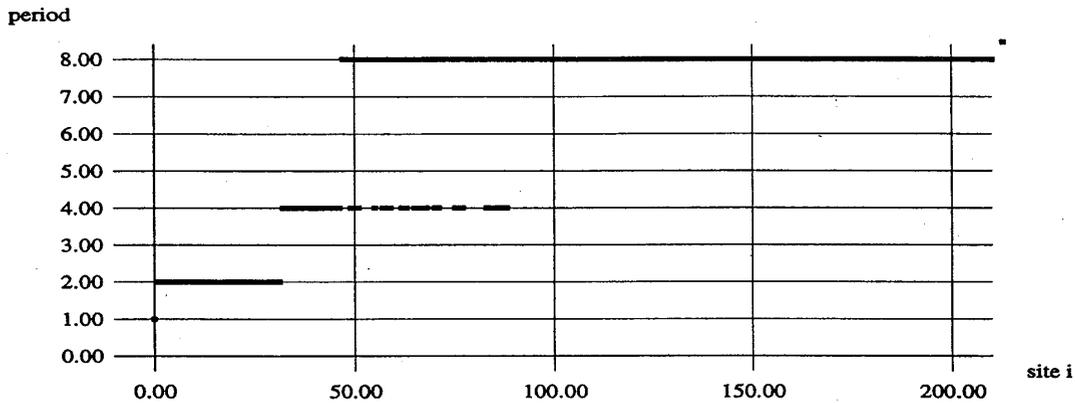


図 2 計算機上に構成された OFS で空間方向分岐が生じている場合 (図 1 に対応) の各サイトでの正確な周期。  $f$  として logistic 写像  $f(x) = Ax(1-x)$  をもちい、  $A = 3.6, \epsilon = 0.55, x^0 = 0.3$ . 横軸が空間 (サイト) 方向、縦軸が各サイト  $i$  での正確な周期。サイト 0 は、定数なので 1 周期。サイト 1 からすでに、固定点ではなく 2 周期解に収束している。図 1 において巨視的に固定点に見えたものが、微視的には 2 周期であることがわかる。巨視的に 2 周期、4 周期に見えたものも同様に正確には、4 周期、8 周期の運動をしている。

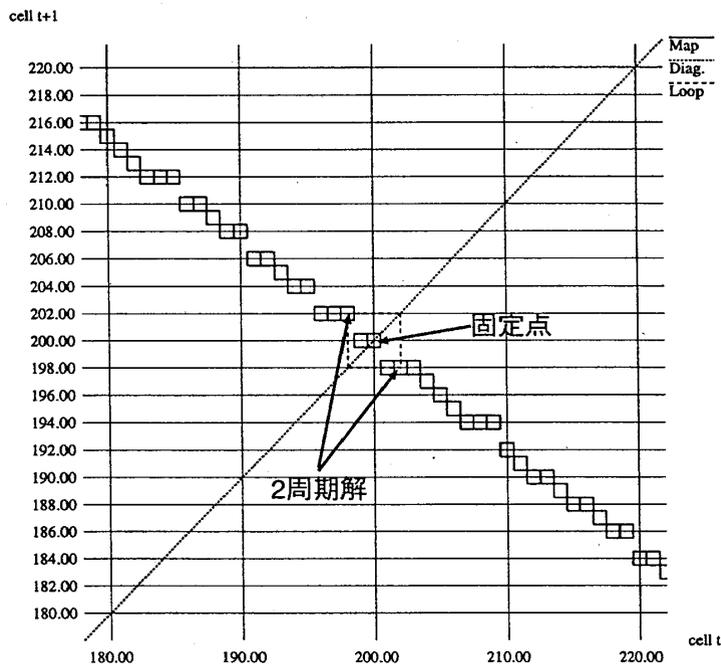


図 3 計算機上に構成された OFS のサイト 1 での固定点近傍での時間発展写像。(図 1、2 に対応  $A = 3.6, \epsilon = 0.55, x^0 = 0.3$ .) 固定点近傍の倍精度 (IEEE 754) 浮動小数点数に、その有効精度最下位 8 ビットを用い番号付けし、有限要素の時間発展写像を表現している。横軸、縦軸は、それぞれ時刻  $t, t+1$  での系の状態に対応する数の番号である。図示したように、サイト 1 での固定点のまわりに有限精度への離散化にともなう 2 周期軌道が形成されていることがわかる。

この2周期解のサイト1での固定点からのずれが、サイト方向に拡大されてサイト方向の分岐が巨視的スケールに生じる。これは、無限精度の場合と同様に、各サイトでの固定点の位置の変化の写像の構成により示すことができる。このとき、有限精度の解析として、固定点を微視的に2周期解としてあつかう必要がある。

各サイト  $i$  での固定点近傍での2周期解を  $x_{*1}^i, x_{*2}^i$  とすると、時間発展式(2)より、各サイトでの周期解は、

$$\begin{aligned} x_{*2}^{i+1} &= (1-\epsilon)f(x_{*1}^{i+1}) + \epsilon f(x_{*1}^i) \\ x_{*1}^{i+1} &= (1-\epsilon)f(x_{*2}^{i+1}) + \epsilon f(x_{*2}^i) \end{aligned} \quad (5)$$

を満たす。 $\eta_1^i := x_{*1}^i - x^*$ ,  $\eta_2^i := x_{*2}^i - x^*$  とし、(5)式を固定点のまわりで展開し、固定点からのずれの空間方向写像

$$\begin{pmatrix} \eta_1^{i+1} \\ \eta_2^{i+1} \end{pmatrix} = A^{-1} \begin{pmatrix} \eta_1^i \\ \eta_2^i \end{pmatrix} \quad (6)$$

を得る。ここで、

$$A := \frac{1}{\epsilon f'(x^*)} \begin{pmatrix} -(1-\epsilon)f'(x^*) & 1 \\ 1 & -(1-\epsilon)f'(x^*) \end{pmatrix}.$$

$A^{-1}$ の固有値は、

$$\lambda_1 = \frac{\epsilon\alpha}{1-(1-\epsilon)\alpha}, \quad \lambda_2 = \frac{-\epsilon\alpha}{1+(1-\epsilon)\alpha},$$

固有ベクトルは、

$$\xi_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \xi_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

となる。このとき、 $\xi_1$ は、固定点からのずれが1周期、つまり固定点が記述可能(無限精度)な場合に対応し、 $\lambda_1$ は、定常解の解析(4)における  $\gamma$ に対応する。

ここでの解析から、無限精度の解析で定常解が固定点へ収束 ( $|\lambda_1| < 1$ ) する場合でも、 $|\lambda_2| > 1$  の場合、有限精度の系では固定点からの分岐が生じる。ここで  $\lambda_2$  に対応する固有ベクトル  $\xi_2$  は、2周期解が固定点  $x^*$  をはさんで対称に位置する場合に対応する。固定点への収束、分岐が生じる場合、無限精度の系 ( $\eta_1^i = \eta_2^i$ ) は、固定点への安定多様体となり、空間方向の写像において1サイトでも固定点が記述不可能 ( $\eta_1^i \neq \eta_2^i$ ) なサイトがあれば、2周期への分岐がはじまり、2周期軌道は、固定点をはさんで対称(不安定方向)に拡大していく。

この収束分岐の生成機構の解析は、一般に系固有の  $n$  周期解への収束、 $2n$  周期への分岐に一般化でき ( $n$  が1または偶数の場合)、spatial bifurcation (空間方向分岐現象) が OFS に生じる条件が得られる(詳細は、[3][4])。

このように、OFS に有限精度の仮定を組み入れることにより、空間方向分岐現象を生成できる。この現象は、無限精度の系の定常解としては、存在しない。しかし、上位サイトに外乱を加えた OFS についても、過渡的に空間方向分岐的振舞いが生じるため、一概に空間方向分岐現象を計算誤差から生じる偽の現象ということとはできない。そして、巨視的な分岐の構造は、写像  $f$  固有の周期軌道の安定性、サイト間の結合係数、微視的スケールに存在する、計算機固有の有限のダイナミクスに強く依存している。このことは、これら系の潜在的性質を顕在化する性質を open flow system タイプの力学の結合系が、持つことを示唆する。

## 4 計算のリアリティーについて

計算機の中では、数は有限のビット数で表現されている。そして、 $M$ ビットで表現できる数の数は  $2^M$ 個程度である。そのため実際に計算機の中で行われている計算は、自然数上の演算と同じである。このような有限の計算の上でカオスの写像を計算して得られる結果は、いったい何を意味するのであろうか？

この小論において、シフト写像と Open flow system の二つの力学系の計算機内部における実現形式について議論してきた。ここに示したことは、無限精度の系を記述するには記述能力が足りない計算機という媒体を使って、あえて、系を記述をしようとするときに何が起こるかということである。つまり、ある記述のレベルでは、書くことができない対象を記述するという操作が何を生み出すのかということである。シフト写像の場合には、無限精度の系の有限精度での近似解釈ともいべき疑似乱数生成系を系の微視的スケールに形成し、Open flow system の場合には、無限精度の系には存在しない空間方向分岐現象を生成することで、系にある潜在的性質を顕在化させている。

これは、ひとつの例示であり問題提起でもある。このような有限システムの力学の振舞いは、その記述(離散化)のされかたに強く依存する。このような系にたいして、計算のリアリティー(正しさ)をどのように見い出せるのだろうか？

これは個々の研究の目的に依存するため、一概に結論を出すことはできない。計算機上に構成した系を真の系の模倣としてあつかう場合には、どのような前提(計算の仮定を含む)のもとでどの程度模倣できているかという指標をもっていなければならない。また、計算機上に構成した系それ自身を対象として研究する場合には、その内部に構成される有限機械としての機構がどのようにその系の振舞いに影響をしているのかを知っておかなければならない。そして、どちらの場合にせよ計算機を用いて計算を行う場合にはどのような計算の前提 [5] のもとで計算を行っているのかを知っておく必要がある。

## 参考文献

- [1] Kunihiko Kaneko, Spatial period-doubling in open flow, Phys. Lett. 111A (1985) 321.
- [2] F.H. Willeboordse, K.Kaneko, Pattern dynamics of a coupled map lattice for open flow, Physica D, (to appear).
- [3] Akihiro Yamaguchi, A mechanism of spatial bifurcations in open flow system, (in preparation).
- [4] 山口 明宏, Open flow system における空間方向分岐現象の機械機構, 物性研究 (出版予定).
- [5] David Goldberg, 西村 恕彦 訳, 浮動小数点演算について計算機科学者は何を知っておくべきか, bit 別冊 コンピューター・サイエンス. (ACM Computing Surveys, Vol. 22, No.1, pp.5-48, What Every Computer Scientist Should Know About Floating-Point Arithmetic)