テーマ：ラチェット模型による分子モーターの研究

高知工科大学

知能機械システム工学科

４年

福岡猛志

目次

緒言

微小粒子の大きさにモーターを作るとき、微小粒子はいろいろな物から常に影響を受けているため制御することが非常に困難である。そこで分子モーターを作るために単純なモデルでの操作を考える。時間的および空間的に周期的に力が働くような条件の下で拡張標準写像で表されるような微小粒子の式を、プログラムを使って解析する。そしてその結果から微小粒子の動きを推察して微小粒子の動きを制御できるかをシュミレートをする。

序章

１９５９年にリチャード・ファインマンが初めてナノテクノロジーという概念を提唱してから半世紀近く経たった。現在ではコンピュータやDNAなどさまざまな分野でナノテクノロジーが注目されてきている。これは今までの技術が徐々に通用しなくなってきているからである。今までの技術とはトップダウンと呼ばれる大きい物をどんどん細かく削って微細な物を作るという技術でこの技術は近い将来限界がくると言われている。それに対しナノテクノロジーを使う方法はボトムアップと呼ばれる。これは原子や分子などを組み合わせて微細な物を作るという方法で、現在この技術が使われている物には「カーボンナノチューブ」などがある。(3)

実際にナノがどの程度の大きさかというと$10^{-9}m$という極微小な長さで、物質を構成している原子の大きさがだいたい$10^{-10}m$であることからナノという大きさがとても小さいものだとわかる。今後このような技術が発展してくると思うが、このような小さい物の場合、我々が普段生活している中で影響を受けている物理法則以外にもさまざまなものから影響を受けている、例えば熱ゆらぎがある。ナノのような大きさになると空気中にあるさまざまな微粒子がランダムに絶えず当たり、微小な物が壊れやすくなったり制御しにくくなったりする。またこれらのような微小な物の場合こちらから操作できる方法というのも限られており、さらに分子や原子単位にまで小さくすると操作の方法が単純なものでなければできなくなってくる。しかしこれらの現象を上手く利用しているものがある、それは生き物である。

人間や動物、植物にいたるまですべてが精密にできた機械と考えることができる。人間や動物は物を動かす時などに手や足の筋肉を使って動かしている。筋肉はアクチンとミオシンという直進分子モーターを使い、ATP というエネルギーを貯めている物質を分解することでエネルギーをもらいフィラメントの上を移動している。アクチン、ミオシンの大きさはそれぞれ数十ｎｍととても小さく、複雑な操作ができるほどの大きさがありません。(1, 2)

ではアクチンやミオシンがどのようにして動いているのだろうか、それらについてモデルを作りシュミレーションしてみる。

分子モーターモデル

２．１分子モーター

独立で物を動かす場合それにはエネルギーを自ら生み出すためにモーターが必要になる。自動車や飛行機にもモーターが必要で自転車のライトのように自転車を動かす力を使っているものもある。しかしナノサイズのもののように小さいときはそれを動かすモーターはもっと小さくなければならない。そこでアクチンやミオシンがどのようにしてエネルギーを作っているかを参考にしてモデルを作ってみる。

アクチンやミオシンなどはその大きさから中にあるモーターを複雑な操作を使って制御することができない。またこれらのモーターは絶えず熱ゆらぎなどの力を受けている。この熱ゆらぎはブラウン運動をしており、短時間で見ればランダムに動いているが長時間で見ると動いていない。そしてこの熱ゆらぎとアクチンなどが ATP を分解して得られるエネルギーはあまり差が無い。そのためアクチンなどはこの熱ゆらぎなどの力を利用して一方向に制御する方法を使っている。今回の実験ではアクチン、ミオシンを参考にし時間的および空間的に周期的に力が働くような、最も簡単なモデルを作りこれを制御できるかを実験する。(2)

ラチェットによる制御

ラチェットとは方向を制御して一方向にしか動かなくするような歯車のことで、自転車のようにペダルを前にしか動かなくしているのもラチェットである。

２．２拡張標準写像について

今回のシュミレーションには式（１．１)、（１．２）を使う。(5)

拡張標準写像

$$x_{n+1} = x_n + p_{n+1}(1+\varepsilon s_n)$$
$$p_{n+1} = p_n - K\sin x_n - As_n \qquad (1.1)$$

（ｘ＝位置　ｐ＝速度　ε，Ｋ，Ａ＝パラメーター　Ｓｎ＝－１のｎ乗）

標準写像

$$x_{n+1} = x_n + p_{n+1}$$

$$p_{n+1} = p_n - K \sin x_n \qquad (1.\ 2)$$

２．１．１拡張標準写像の式の説明

拡張標準写像とは時間的および空間的に周期的に力が働き、なおかつその平均が０になるような最も簡単なモデルである。これは

$F\ddot{x} = F(x,t)$

で$F(x,t)$をｘ、ｔに関して周期的かつ平均が０になるような簡単なモデルで

$F(x,-\pi) = F(x)$

$F(x,2) = F(x)$

で表される。よってこれらは－ａπからａ πの間で同じことを繰り返していることになる。そしてプラス、マイナスから周期的に同じように働いているので平均が０になる。よって

$$\int_{-a\pi}^{a\pi} F(x,t)dx = 0$$

$$\int_{-a\pi}^{a\pi} F(x,t)dt = 0$$

になる。今回この$F(x,t)$を運動方程式を使って表す。

$F = m\ddot{x}$

（F＝力、m＝質量、 $\ddot{x}$ ＝加速度）

まずモデルとして（図１）のようなものを考える。これは微小粒子にある程度の力を加えた時だけ微小粒子が移動するというように仮定したものである。図１の上に微小粒子を置き、左右に振るような形で力を加えるようなモデルである。これを式にすると

$F = -k\sin x + A$

（k＝凹凸の高さを決めるパラメーター、A＝微小粒子に加える力）

となる。さらに摩擦を加えると

$F = -k\sin x - \gamma\dot{x} + A$

（γ ＝摩擦、 $\dot{x}$ ＝速度）

となる。ただし微小粒子に加える力は一瞬で両方から均等に力をくわえる。加える力は微小粒子を動かすというよりも弾くように力を加える。一度力を加えて次の力を加えるまでの時間を１秒と仮定し式にすると

$F = (-k\sin x - \gamma\dot{x} - (-1)^n A) f_{(t)}$

（n＝回数、 $f_{(t)}$ ＝時間）

で表される。ただし $f_{(t)}$ の時間は一度力を加えて次の力を加えるまでを１秒とおいた時の時間である。

次に $m\dot{x} = p$ とおくと

$m\ddot{x} = \dot{p}$

となる。ここで微小粒子の質量を１と仮定すると

$\dot{x} = p$

$\ddot{x} = \dot{p}$

となり。

$\dot{x} = p$

$\dot{p} = (-k \sin x - \gamma \dot{x} - (-1)^n A) f_{(t)}$

という二つの微分方程式ができる。この微分方程式を積分する。しかしこの時摩擦を考えると計算式がとても難しくなるので今回は摩擦を考えないことにする。

$\dot{x}$ を時間ｎ→ｎ＋１で積分すると

$\dot{x} = x_{n+1} - x_n$

$\dot{x} = \int_n^{n+1} p_{(t)} dt$

になる。この時ｐは運動量を表すもので値が決まっているので

$\dot{x} = p \cdot \int_n^{n+1} dt$

となり

$\dot{x} = p(n+1-n)$

となる。ここで（ｎ＋１－ｎ）は図２のように力を加えた点から次に力を加えた点を表しているので（ｎ＋１＋ε―ｎ）と表すことが出来る。よって

$x_{n+1} - x_n = p(1+\varepsilon)$

になる。この時ｐは微小粒子に力を加えた瞬間を過ぎると p_{n+1} になる。力を加えるのは一瞬なのでｎ＜n_a≦ｎ＋１を満たす時、ｐはすべて p_{n+1} となる（図２）よって

$x_{n+1} = x_n + p_{n+1}(1+\varepsilon)$

となる。次に $\dot{p}$ を時間ｎ→ｎ＋１で積分すると

$$\dot{p} = p_{n+1} - p_n$$

$$\dot{p} = \int_n^{n+1} (-k\sin x - (-1)^n A)\, f_{(t)} dt$$

になる。この時$(-\sin x - (-1)^n A)$は時間の関数ではないので

$$\dot{p} = (-k\sin x - (-1)^n A) \cdot \int_n^{n+1} f_{(t)} dt$$

となる。$\int_n^{n+1} f_{(t)} dt$は先ほども書いたように力が瞬間的にしか働かないのでn＜n_a≦n＋1を満たすn_aの値までならばどの値で積分をしてもその値は変わらない。また微小粒子に加える力は必ず１とする。よって

$$\int_n^{n+1} f_{(t)} dt = 1$$

となる。したがって

$$\dot{p} = (-k\sin x - (-1)^n A)$$

になる。よって

$$p_{n+1} - p_n = (-k\sin x - (-1)^n A)$$

$$p_{n+1} = p_n + (-k\sin x - (-1)^n A)$$

になる。$(-1)^n = s_n$とおくと。

$$x_{n+1} = x_n + p_{n+1}(1+\varepsilon s_n)$$

$$p_{n+1} = p_n - k\sin x_n - As_n$$

と式変形ができ拡張標準写像を求めることが出来る。

　この拡張標準写像のパラメーターであるA、ε＝０とすると

$$x_{n+1} = x_n + p_{n+1}$$

$$p_{n+1} = p_n - k\sin x_n$$

になり、標準写像になる。標準写像は拡張標準写像のパラメーターA、εを考えない式で、つまり微小粒子に両端から衝撃を与えずにただ図１のような力を加えているものである。

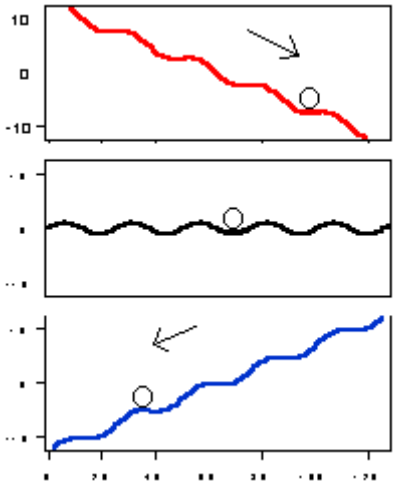

図１．拡張標準写像のモデルのイメージ（8）

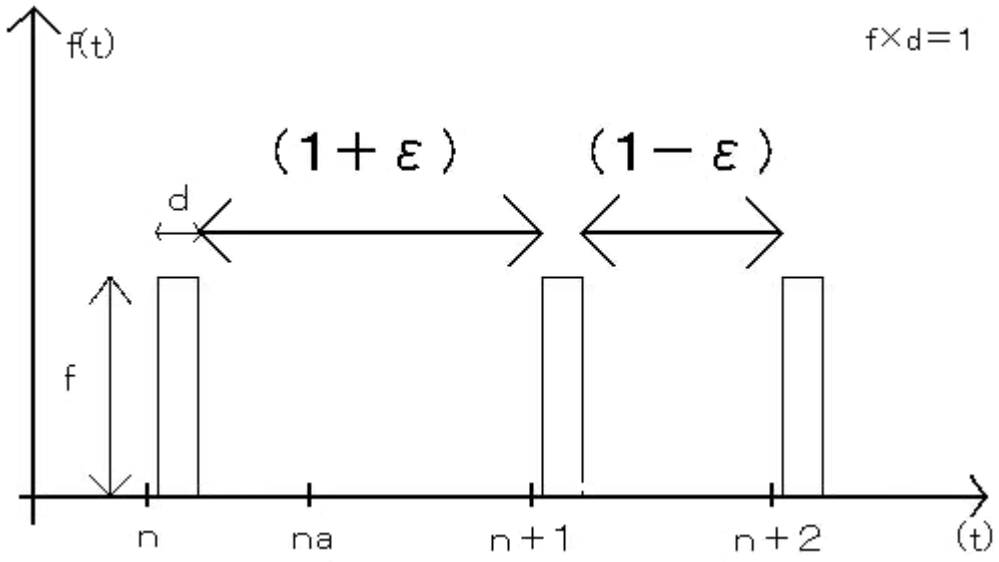

図２．瞬間的に力を加えるイメージ

数値実験方法

このシュミレーションをするために今回はＣ言語を使いプログラムを組む（付録）そしてそのプログラムを使い拡張標準写像について位相空間を調べることでその特性を見てみる。次にｋ、Ａ、εのパラメーターの値を変え、その時ｐがどのように変えていけば一番安定して分子を制御できるかを調べる。この数値実験では式の繰り返し回数をｎ、初期値の数をｍとおき、ｘの初期値の数をｍｘ、ｐの初期値の数をｍｐで表す。ｘやｐのとる初期値は必ずー πからπの間の値で等分する。

３．１位相空間について（4）

位相空間とは力学系の状態を表す仮想的な空間で、その状態を特定するために自由度と言われるものがある。この自由度は力学系によって異なってくる。今回のように位置と速度を考えて時間軸を無視する場合、自由度は２となり次元は２次元になる。空間３次元で考えた場合６次元必要になり表示することができなくなるので今回は考えないものとする。

位相空間からさまざまな運動系統がみることができ、その特性は主に振動、減衰、発散、カオスなどがある。（図３）

３．２拡張標準写像の特性について

３．２．１ｘ、ｐの範囲について

ｘ、ｐの範囲を調べるために標準写像である式（１．２）にｋ＝０．６を代入し、ｘ、ｐの初期値をそれぞれｘ＝００．２、ｐ＝－４とする。そしてｍｘ、ｍｐを２０にしそれぞれの点ごとにｐ＝０．０５ずつ増加させｘは固定させておく。これらの実験をｘ、ｐ方向で共に－３ πから３πの間でグラフを作成する。グラフを作成する時 Sn＝１の時と Sn＝－１の時を別々に作成させる。これは１と－１を同じグラフに表示した場合グラフがとても分かりにくくなるのでこれらは別々に表示することにする。

次に同じ作業をｘ、ｐそれぞれー πから πの範囲内で決めたものをグラフにし、比較してみる。（図４）

図４より式（１．２）はー πから πまでの間でひとつのグラフを作っている。そのためー πから πの範囲を超えた場合は同じグラフを描くので意味がないと判断し今回は省くことにする。この結果はε、Ａの値が０以外の時も成り立つ（図５）

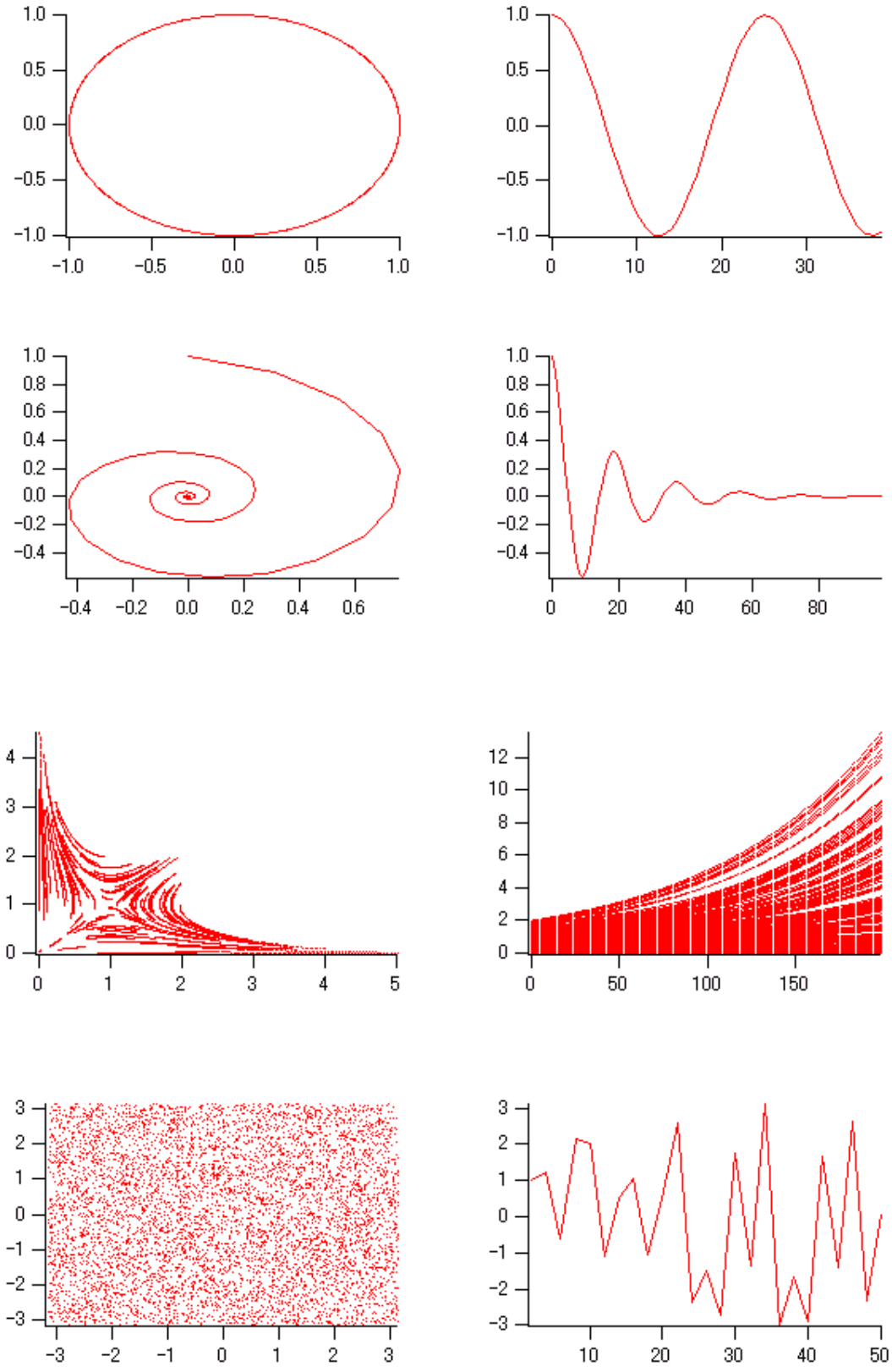

図3 振動 減衰 収束 カオス

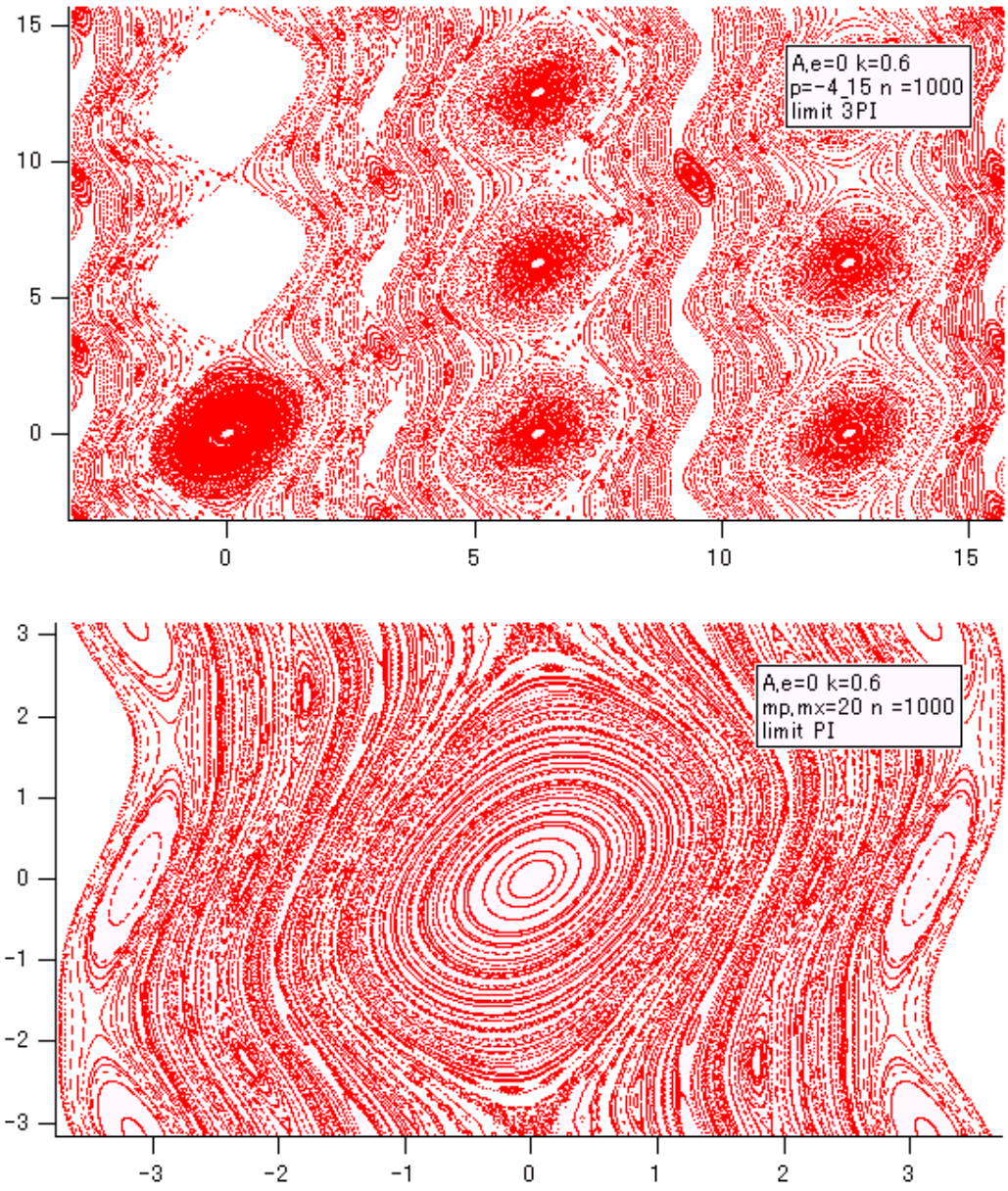

図4　位相空間からグラフに表示させる範囲を決める．1

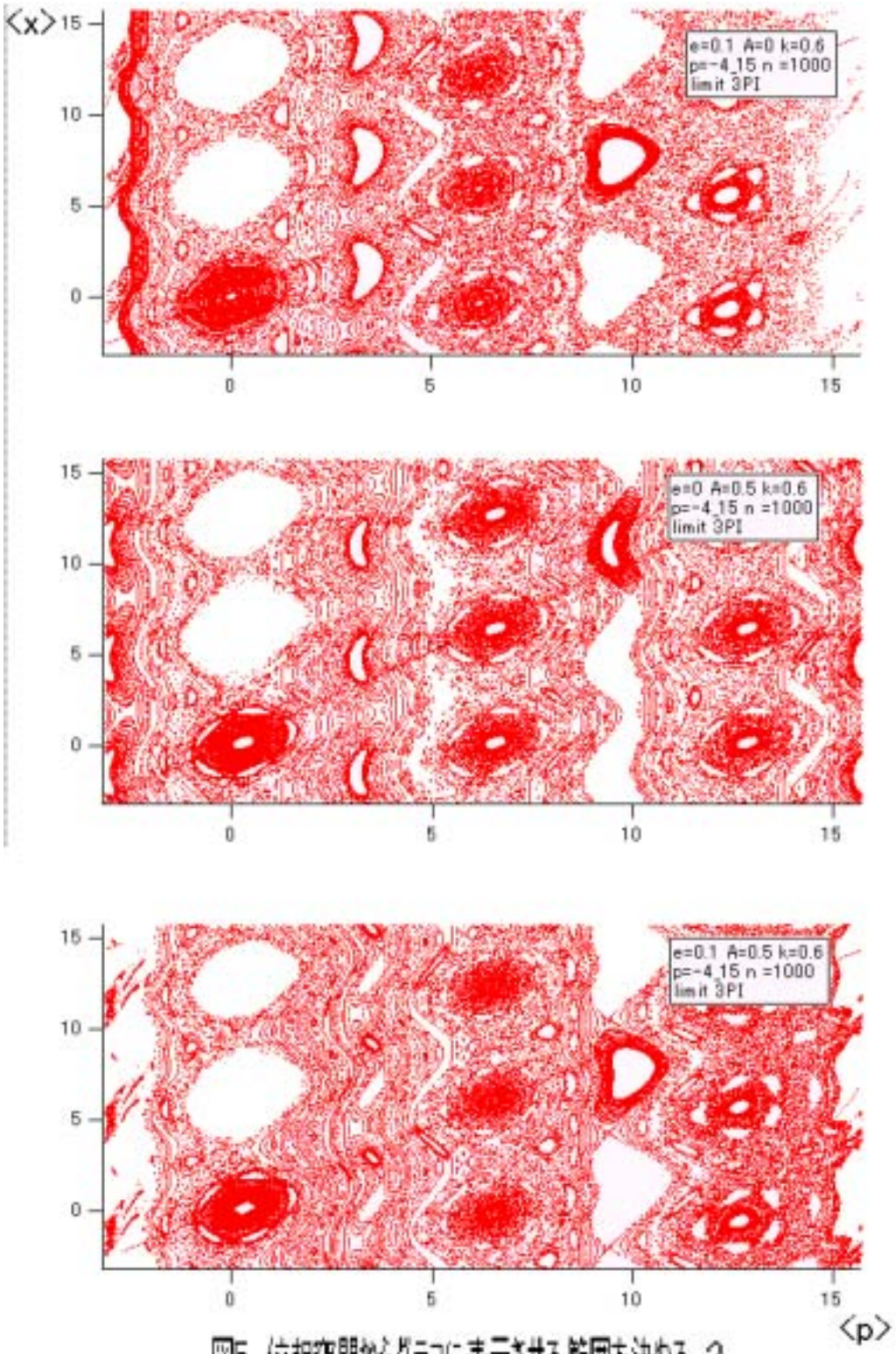

図5 位相空間からグラフに表示させる範囲を決める. 2

３．２．２数値実験の精度

このシュミレーションの精度について調べる。拡張標準写像の A と εの値がそれぞれ０の時、ｐの平均速度は０になる。これは位相空間を見ることによってわかる。A＝０、 ε＝０をそれぞれの拡張標準写像に代入する。この時写像が線対称や点対称のような対称性を持つ時、これらのグラフはすべて同じ動きをする。よってｐの平均速度は０となる。

対称性

　対称性を調べるために

$x_{n+1} = x_n + p_{n+1}(1+\varepsilon s_n)$

を

$-x_{n+1} = -(x_n + p_{n+1}(1+\varepsilon s_n))$

に変形させることによってｙ座標に対しての線対称であるかを調べ

$p_{n+1} = p_n - k\sin x_n + As_n$

を

$-p_{n+1} = -(p_n - k\sin x_n + As_n)$

に変形させることによってｘ座標に対しての線対称であるかを調べる。そして

$-x_{n+1} = -(x_n + p_{n+1}(1+\varepsilon s_n))$

$-p_{n+1} = -(p_n - k\sin x_n + As_n)$

に変形させることで点対称であるかを調べる。

　図６～８より、A か εが０の時、グラフはすべて対称性を持っている。よって A か εが０の時はｐの平均速度が０になることがわかる。次に本当にｐの平均速度が０になるかを調べる。まずｋを０から１５まで０．５づつ増加させ、ｍｘ、ｍｐを１５にする。そしてｎ＝５０００にし A＝０、 ε＝０，A=０．５、 ε＝０，A＝０、 ε＝０．１，A＝０．５、ε＝０．１の４つのパターンでグラフを作成してみる。

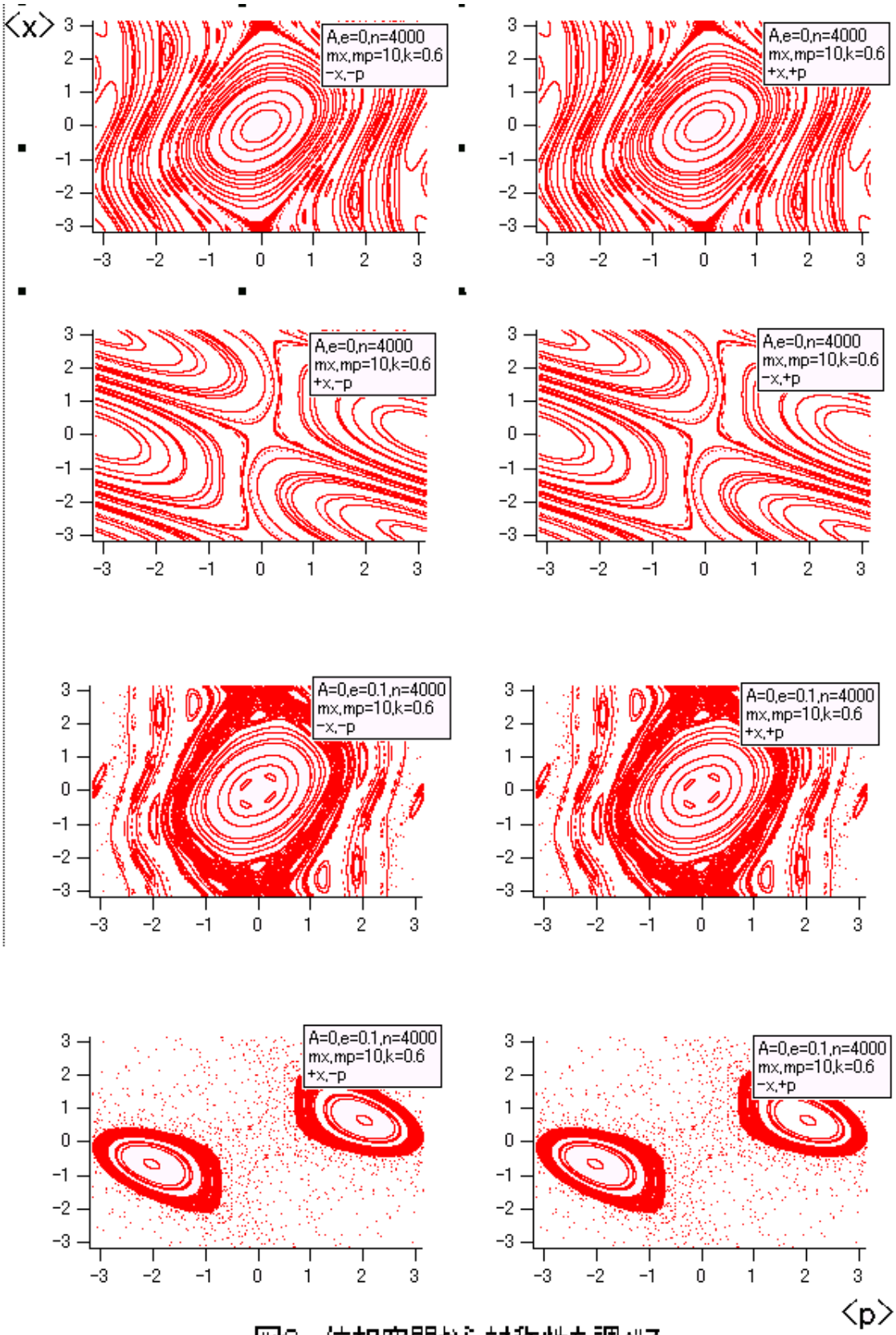

図6　位相空間から対称性を調べる

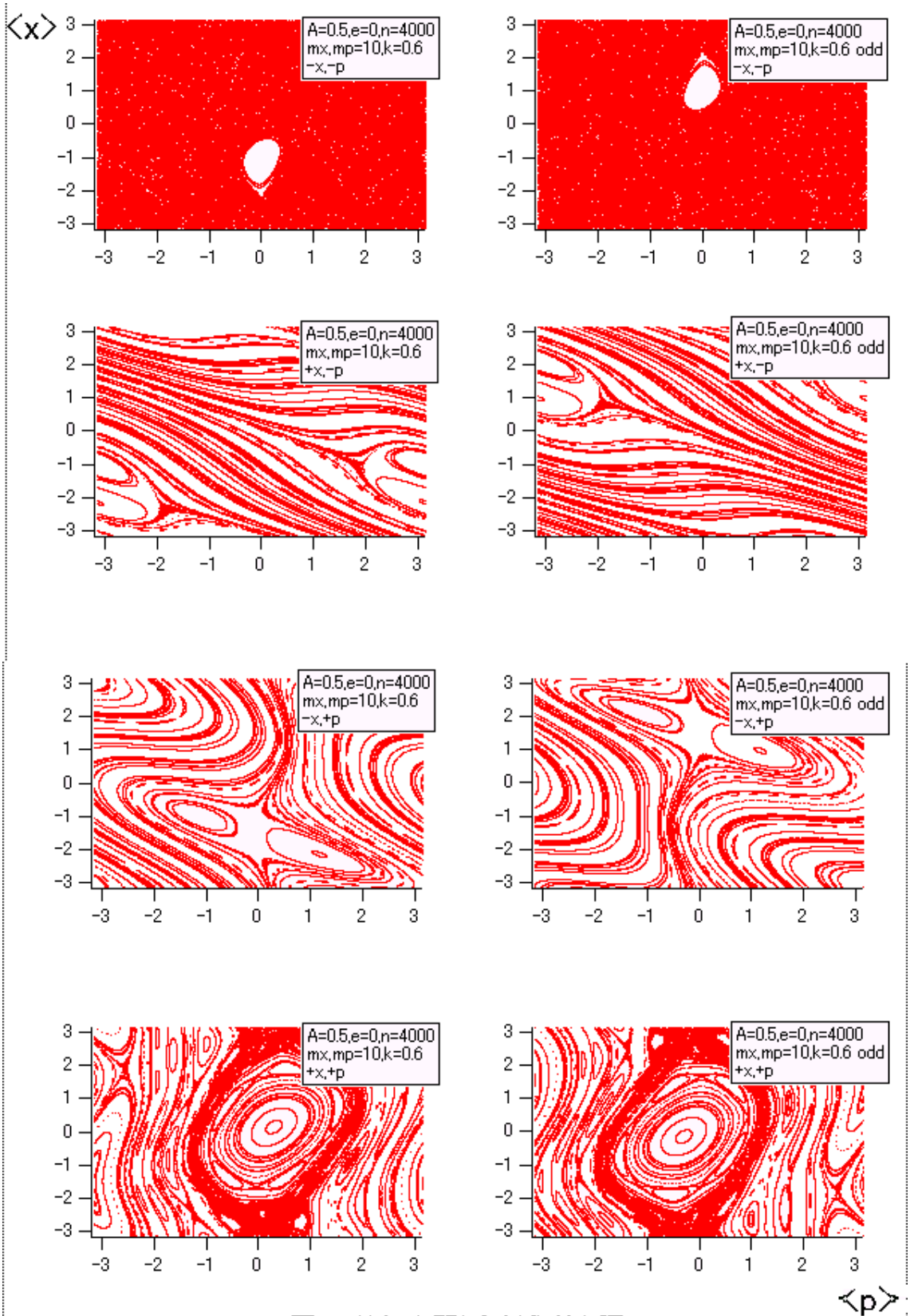

図7　位相空間から対称性を調べる

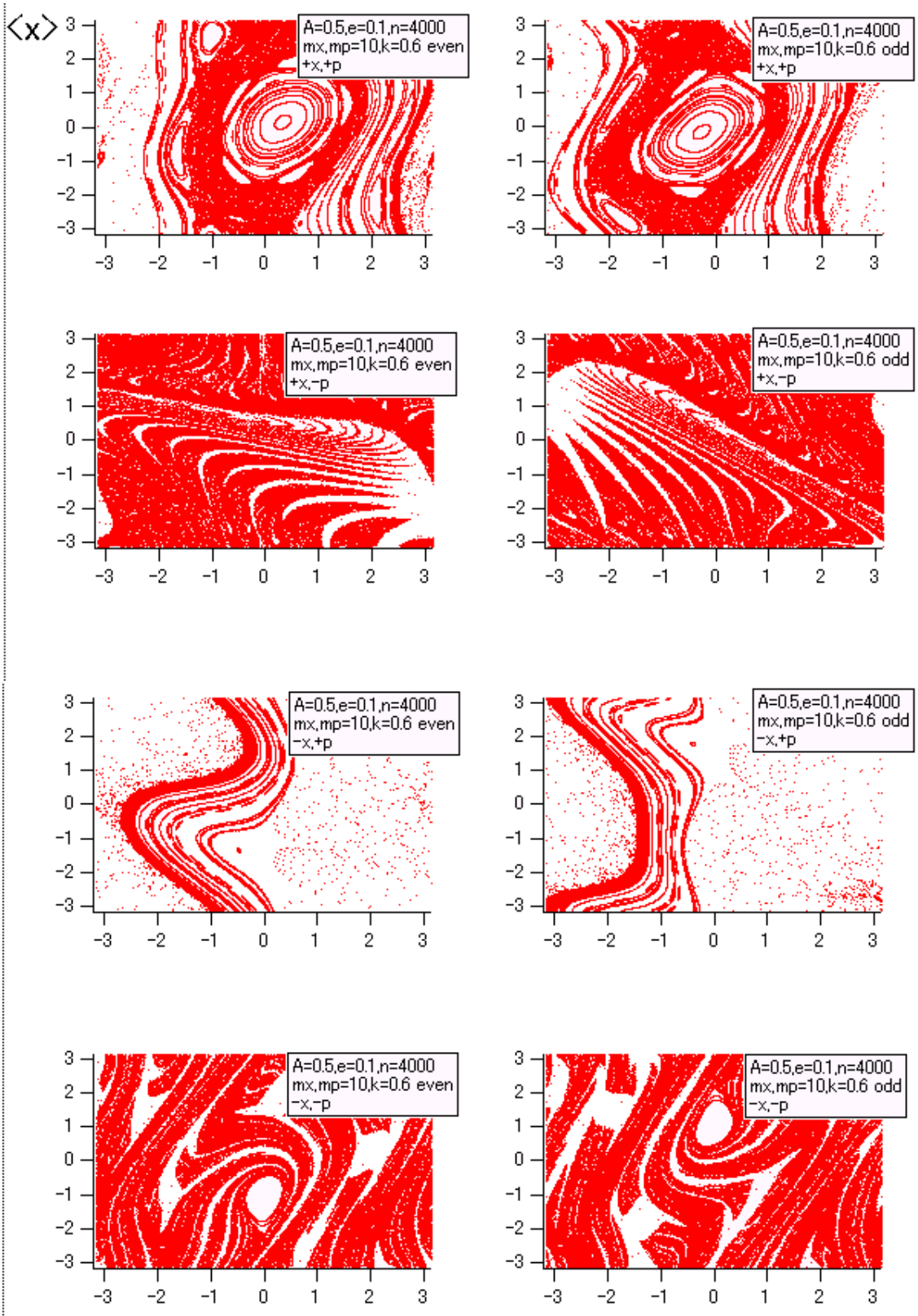

図8　位相空間から対称性を調べる

図９からわかるようにA、εがそれぞれ０ではない時でもｐの平均速度が０になっていない。特にA＝０．５、ε＝０の時とA＝０．５、ε＝０．２の時は差が見られない。これは原因としてｍｘ、ｍｐ、ｎの数によるものと思われる。プログラムで繰り返すごとに毎回わずかながら誤差が出る。なので少しの誤差がわからなくなるぐらいまで数値を大きくし、数値実験をｍｘ、ｍｐを１５０にしｎを５００にし、ｋを０から１５まで行ってみる。（図１０）するとA＝０．５、ε＝０の時ｐの平均速度が０に近づいている。よって今回のシュミレーションは誤差が生じるのでそれを考慮しなければならない。

３．２．３拡張標準写像の位相空間

拡張標準写像の位相空間について調べる。まずｋ＝０からｋ＝１．２までｋを０．３づつ増加させてその結果を見る。その時ｍｘ、ｍｐを１５、ｎ＝２０００とし、ε＝０．１、A＝０．５とする。（図１１）

図１１より拡張標準写像は微小粒子が振動やカオスを繰り返しながら全体的にプラスのところで安定している。よって拡張標準写像はプラスの方向やマイナスの方向に振動しながらも全体的にプラス、マイナスの方向に動く式である。

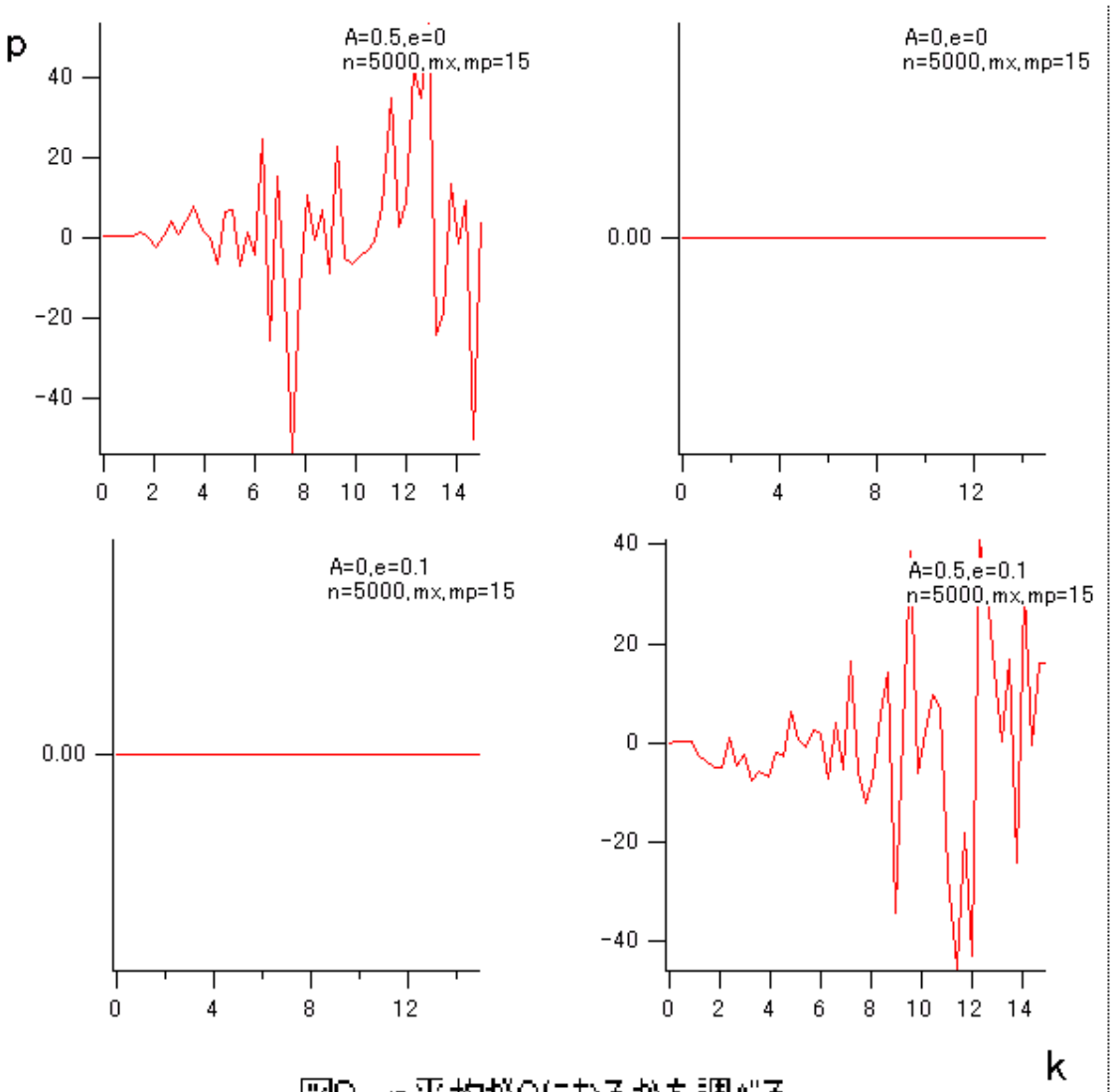

図9　p平均が0になるかを調べる

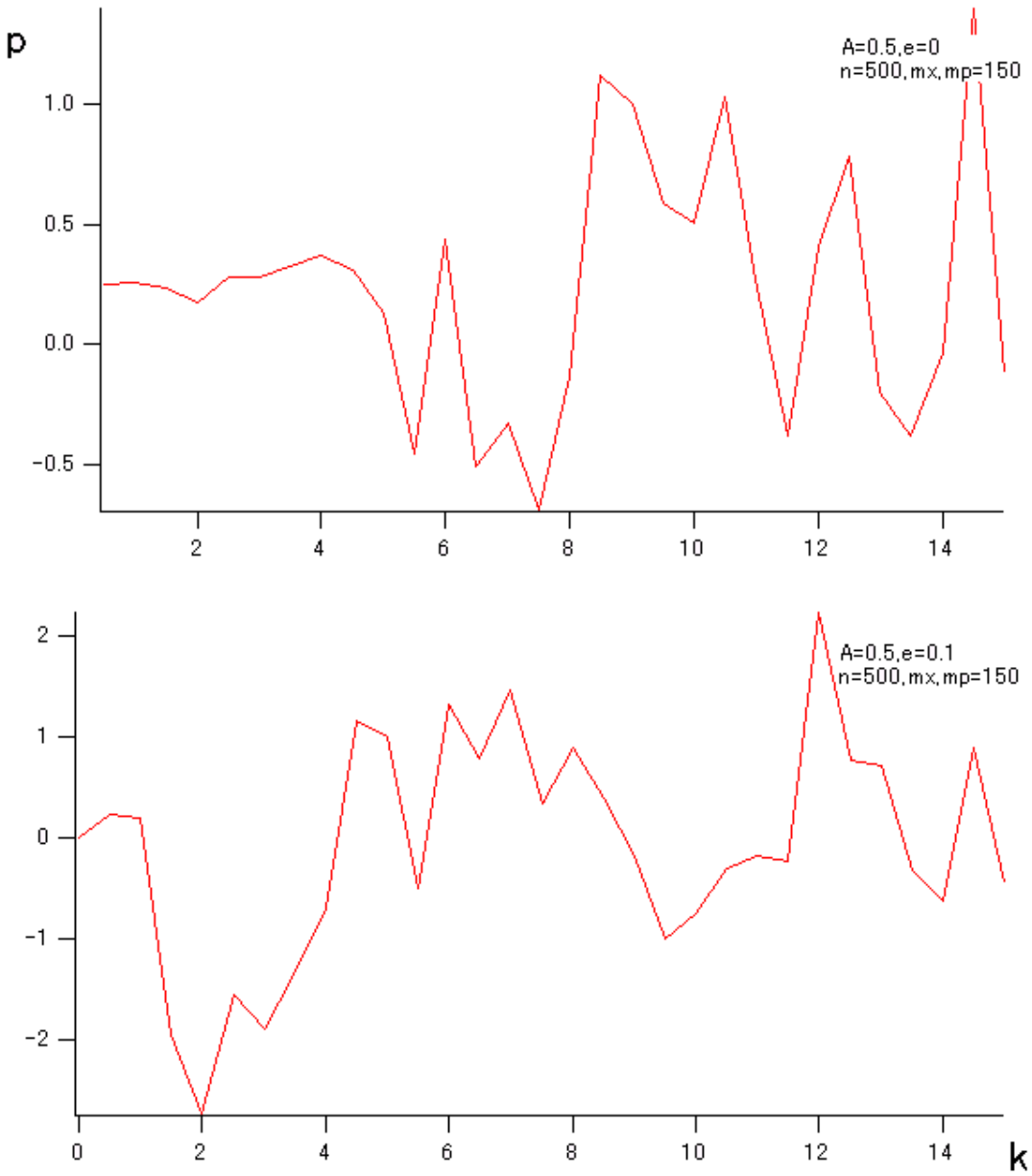

図10　pの平均の誤差を調べる

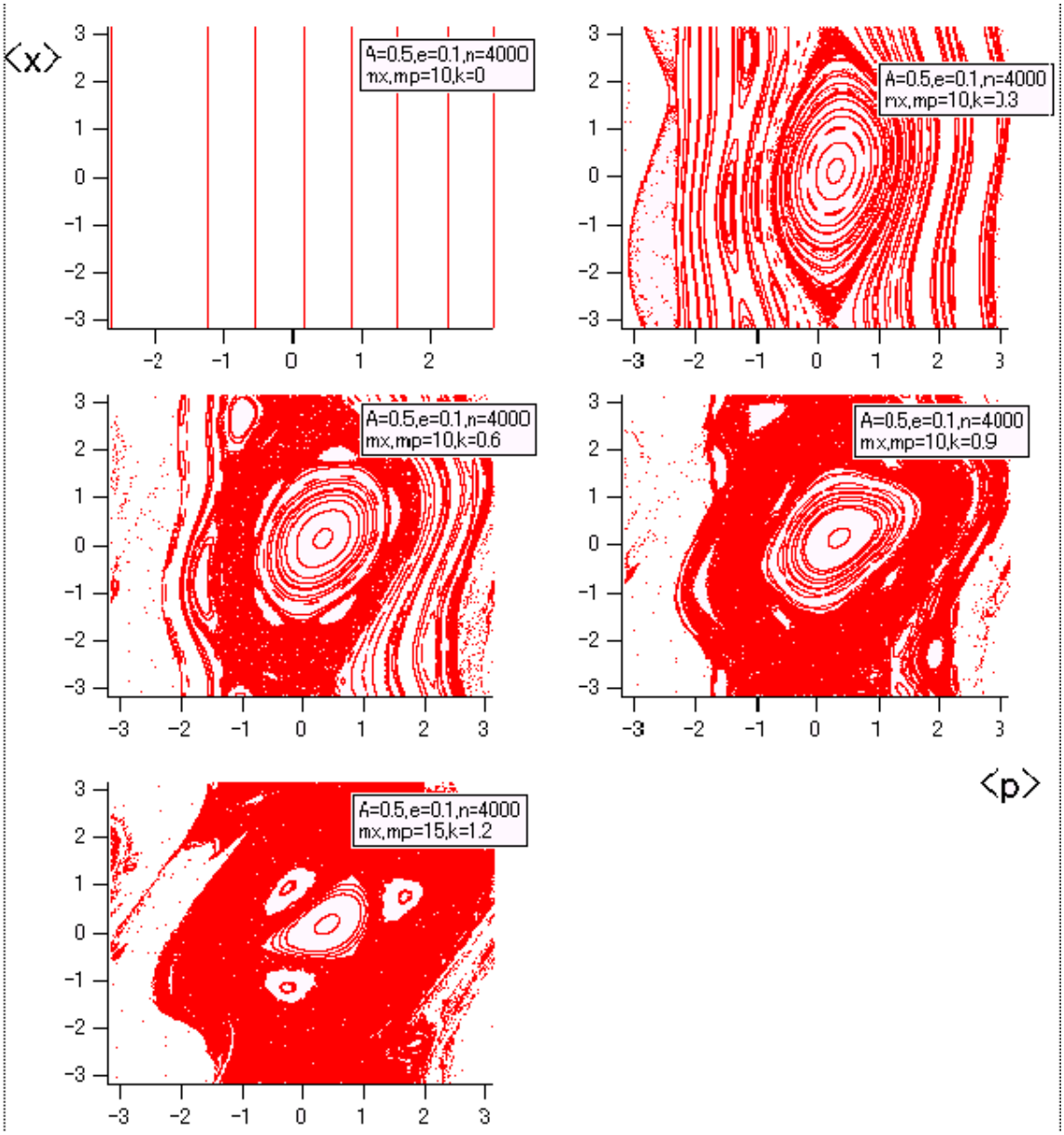

図11　拡張標準写像の特性について

３．３ｐの平均を求める

式（１．１）を使い ε、A を変化させながらグラフの変化を見ていく。グラフはｋを０から１５まで０．３ずつ増加させ、それを横軸にする。そしてｋの変化に応じて各点でそれぞれｐの平均値を出しそれを縦軸にする。そうすることによって ε、A のパラメーターを変化させ値がいくつの時にｐが平均してプラスの方に進むのか、マイナスの方に進むのかを見ることができる。その結果より ε、A、ｋのパラメーターがどの値の時にｐが最もプラス、マイナスの方向に動くかを調べる。

このときｎ＝２０００、ｍｘ、ｍｐ＝５１にし、 εを０．１から０．８まで０．１ずつ増やし、A を０．３から３．５まで０．２ずつ増やすことにする。またこの数値実験では誤差が出る。（図１２）よってｋ＝４以上の数値は無視する。

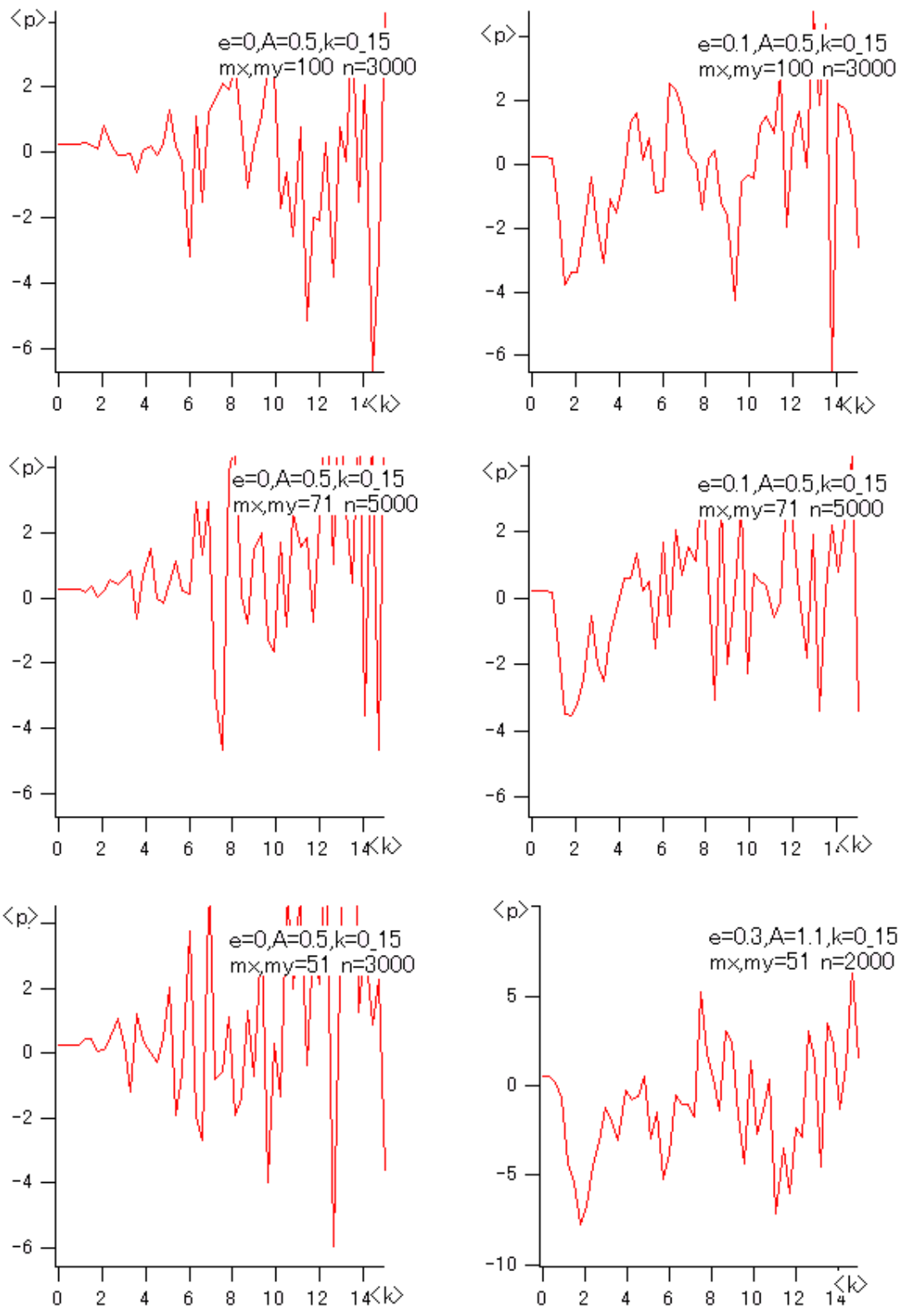
e=0,A=0.5,k=0_15
mx,my=100 n=3000
e=0.1,A=0.5,k=0_15
mx,my=100 n=3000
e=0,A=0.5,k=0_15
mx,my=71 n=5000
e=0.1,A=0.5,k=0_15
mx,my=71 n=5000
e=0,A=0.5,k=0_15
mx,my=51 n=3000
e=0.3,A=1.1,k=0_15
mx,my=51 n=2000
<p>
<k>

図12　k＝4以上の精度

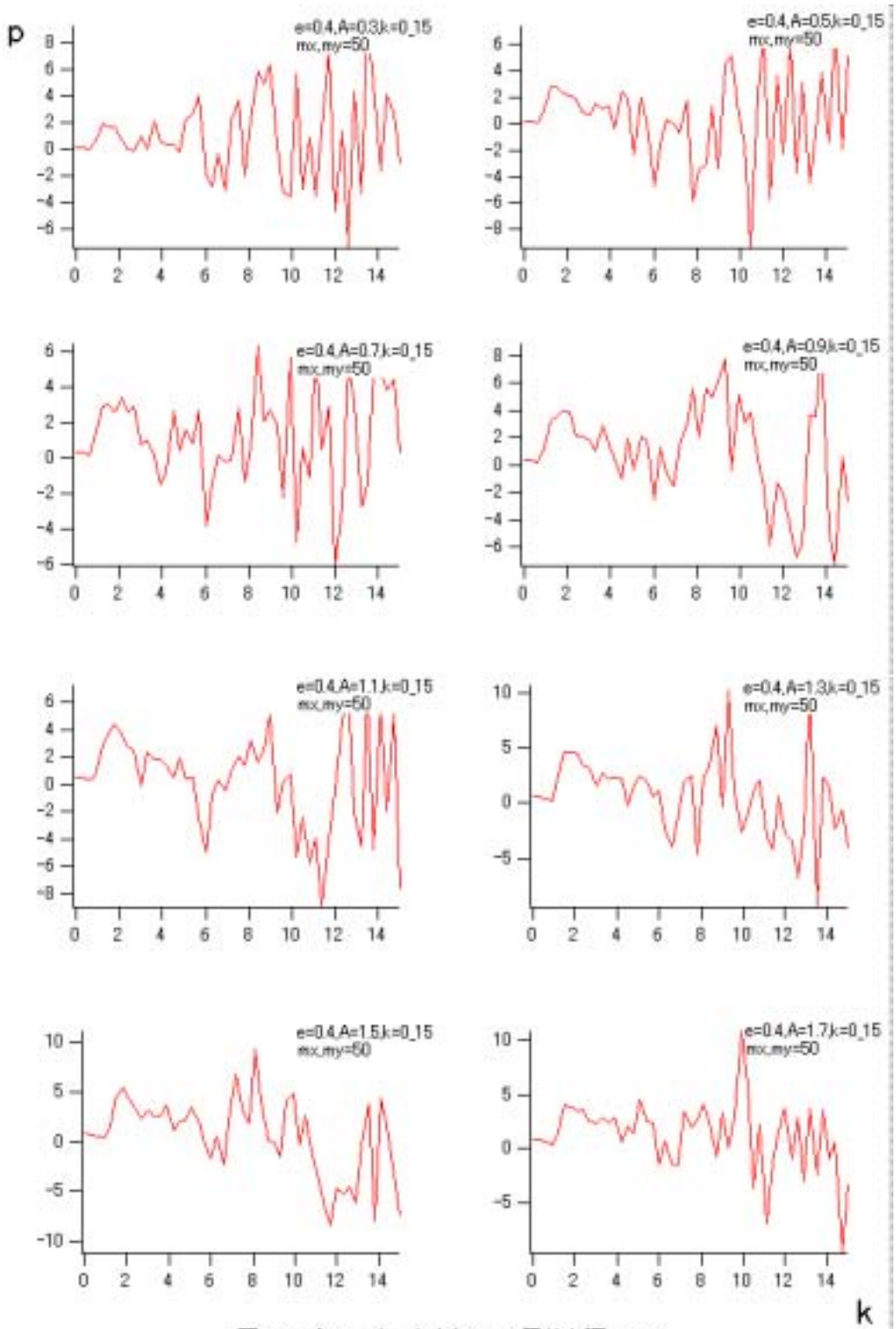

図13 パラメーターを変えてpの平均を調べる．1

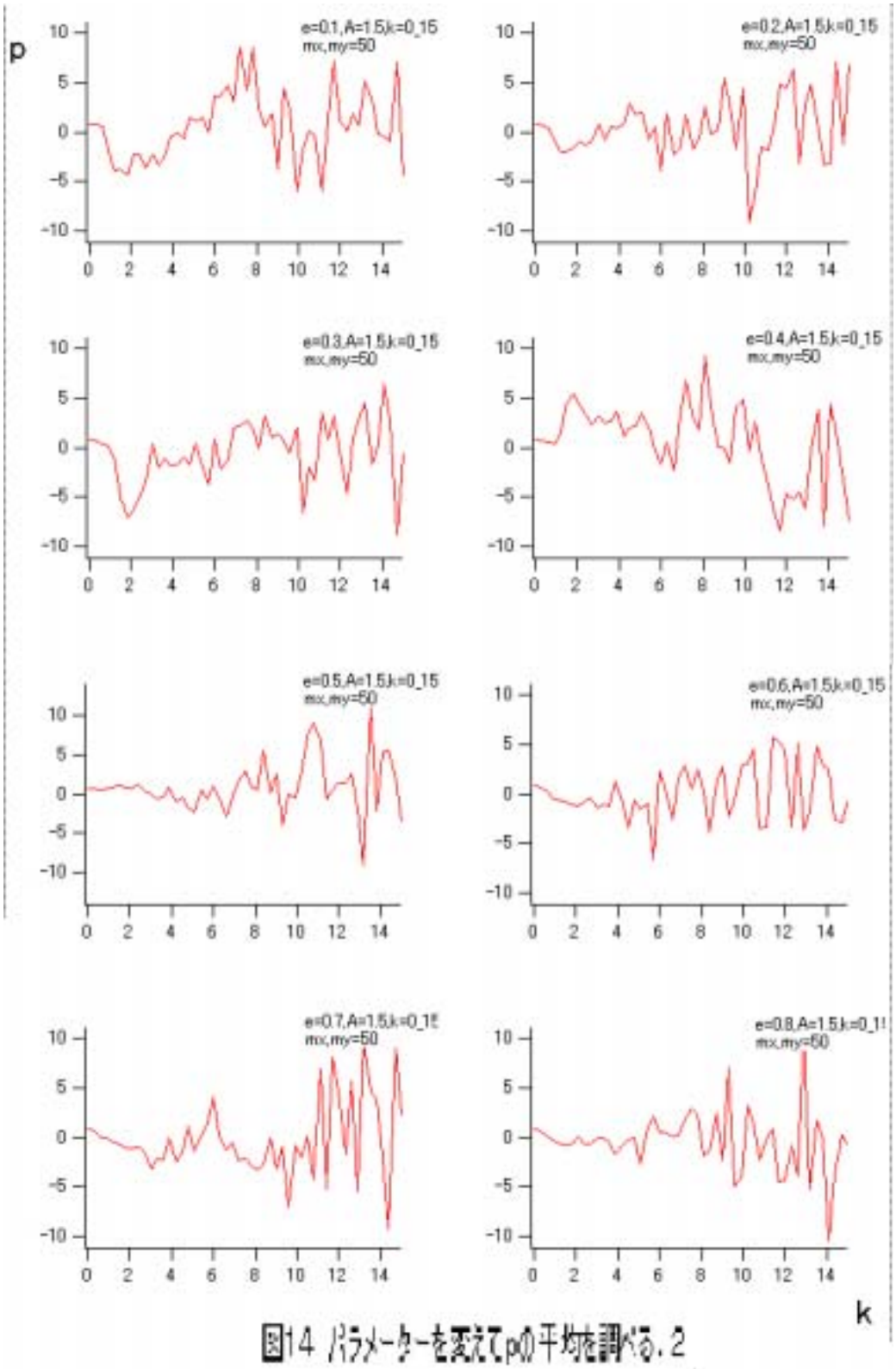

図14 パラメーターを変えてpの平均を調べる. 2

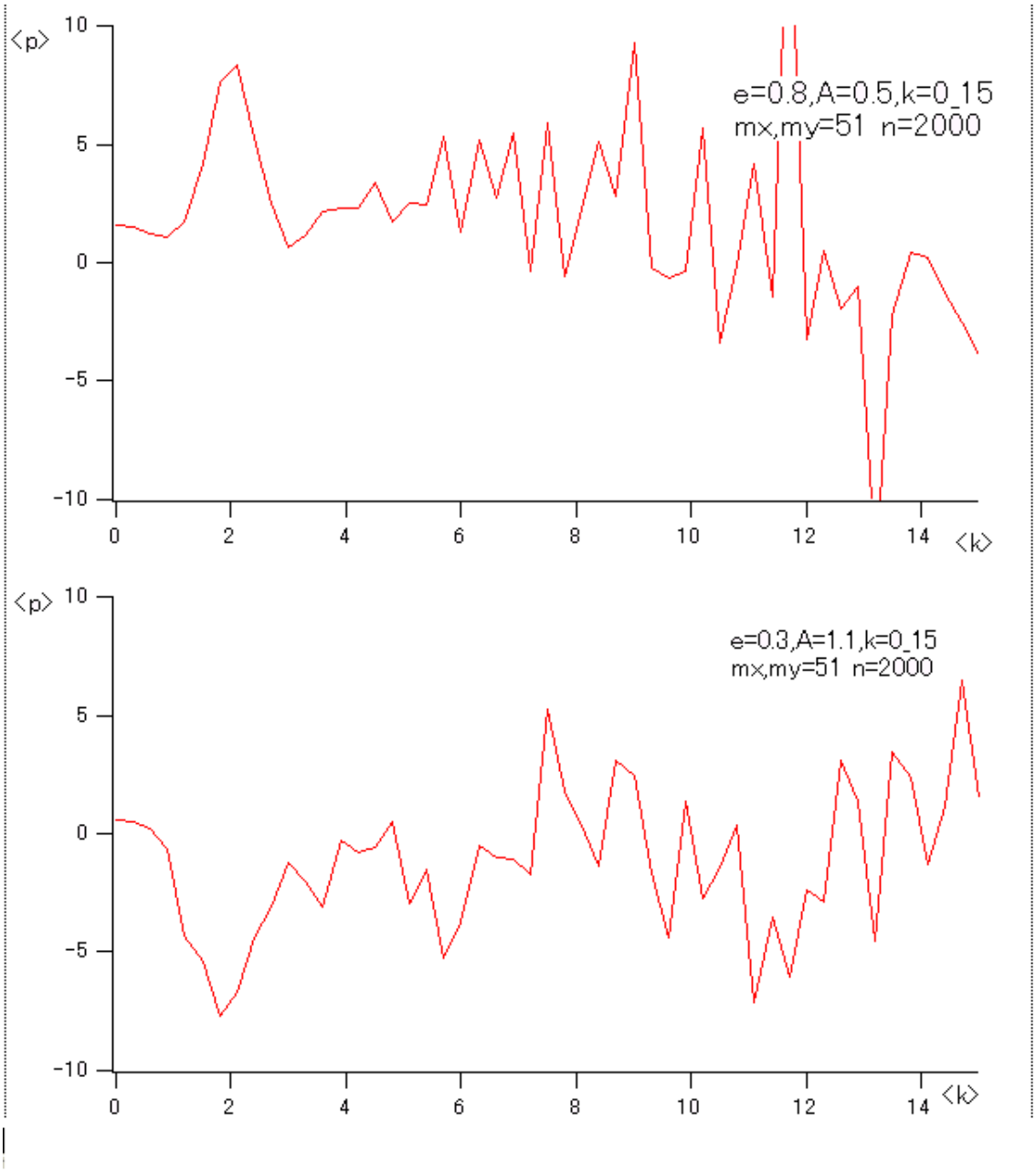

図15 pの最大値、最小値を求める

考察

結果よりパラメーター ε、A、ｋがそれぞれ０．３、１．１、１．８の時に一番マイナスの方向に動き、０．８、０．５、１４．７の時に一番プラスの方向に動くことがわかった。このことから微小粒子は拡張標準写像で表される周期的な力を使って動かすことができ、またその方向も変化させることができることがわかった。これらの結果からアクチンやミオシンのような微小な物から磁気浮揚などの非接触移動をするようなものの最も簡単なモデルができた。今回は精度があまり高くなかったので今後はもう少しアルゴリズムを改造し、精度の高いシュミレートを行おうと思う。

謝辞

今回の数値実験にあたり適切な助言を下さった全助教授、ならびにお世話になった方々に深く御礼申し上げます。

参考文献

1．新生物物理の最前線　　日本生物物理学会　　講談社

2．生物分子モーター　　柳田敏雄　　岩波書店

3．http://www5.ocn.ne.jp/~report/index.html こちらは気になる科学探検隊

4．http://www001.upp.so-net.ne.jp/seri-cf/reference.html　カオス&フラクタル術語集

5．Extended Standard Map with Spatio-Temporal Asymmetry　Taksu Cheon, Pavel Exner, and Petr seba

6．Cold　atoms in optical lattices: a Hamiltonian ratchet　T.S. Monteiro, P.A. Dando, N.A.C.Hutchings, and M. R. Isherwood

7．A chaotic 'turnstile' for atoms in periodic potentials　M. R. Isherwood and T. S. Monteiro

8．http://www.mech.kochi-tech.ac.jp/cheon/13/ratchet-yitp.files/frame.htm　ラチェット・マップのダイナミクス

付録

プログラム

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define PI M_PI

int main(void);
float DEp(float p,float x,float s,float k,float e,float A);
float DEx(float p,float x,float s,float k,float e,float A);
float amod(float x,float a);
float bmod(float x,float y);




int main(void){

    int   n,i,n1,at,bt;
    float mp,mx,b,bb,s,e,A,k,p,x,t,p0,x0,p1,x1,p00,x00,p11,x11,hei1,hei2;

    FILE *fp1,*fp2,*fp3;



    fp1 = fopen("bunshimotertxt_p.txt","wb");
    fp2 = fopen("bunshimotertxt_x.txt","wb");
    fp3 = fopen("bunshimotertxt_t.txt","wb");

    e = 0;
    A = 0;
    printf("ｋの値を決めてください¥n");
    scanf("%f",&k);
```

```
printf("p の分割回数を決めてください¥n");
scanf("%f",&mp);
printf("x の分割回数を決めてください¥n");
scanf("%f",&mx);



n = 5000;

if(mp == 0){
	mp = 1;
	at = 1;
}
if(mx == 0){
	mx = 1;
	bt = 1;
}

for(n = 0; n1 < 20; n1++){
	k += 0.3;
	hei1=0;
	hei2=0;


	for(b = 1; mp >= b; b++){
		p00 = bmod(mp,b);

	for(bb = 1; mx >= bb; bb++){
		x00 = bmod(mx,bb);

	p0 = p00;
	x0 = x00;


	for(i = 0; i < n; i++){
		s = -pow(-1,i);
```

```
            hei1 += p0;
            hei2 += x0;


            p11 = p0;
            x11 = x0;

//            p11 = amod(p11,PI);
//            x11 = amod(x11,PI);

/*            if(i < (kuri/2)){
                if(s == 1){
                        fprintf(fp1,"%f    %f¥n",x11,p11);
                        }
                /* else{
                    //      fprintf(fp2,"%f    %f¥n",x11,p11);
                }*/

/*            }
            if(i > (kuri/2)){
                if(s == 1){
                        fprintf(fp3,"%f %f¥n",x11,p11);
                        }
                /* else{
                    //      fprintf(fp2,"%f    %f¥n",x11,p11);
                } */
 //           }




                p1 = DEp(p11,x11,s,k,e,A);
                x1 = DEx(p1,x11,s,k,e,A);
```

```
                p0 = p1;
                x0 = x1;


        }
    //          hei1 = hei1/kuri;
    //          hei2 = hei2/kuri;

     /*       fprintf(fp1,"%f    ", hei1);
            fprintf(fp2,"%f    ", hei2);
            fprintf(fp1,"%f¥n", k);
            fprintf(fp2,"%f¥n", k);    */
        }
        }

        hei1 = hei1/(n*mp*mx);
        hei2 = hei2/(n*mp*mx);


    fprintf(fp1,"%f    ", hei1);
    fprintf(fp2,"%f    ", hei2);
    fprintf(fp1,"%f¥n", k);
    fprintf(fp2,"%f¥n", k);
    }

    scanf("%d",&x0);
    fclose(fp1);
    fclose(fp2);
    fclose(fp3);
    return(0);
}
```

```
float DEp(float p,float x,float s,float k,float e,float A){

	float X;
	X = p - k*sin(x) - A*s;
	return(X);
}

float DEx(float p,float x,float s,float k,float e,float A){

	float Y;
	Y = x + p*(1+e*s);
	return(Y);
}


float amod(float x,float a){


	while(x > a){
		x = x-2*a;
	}
	while(x < -a){
		x = x+2*a;
	}
	return(x);
}

float bmod(float x,float y){

	x = -1*PI+(y-1)*((2*PI)/(x-1));
	return(x);
}
```