

Unix 計算機を使った物理学実験 (別冊)
-Fortran77 言語によるサンプルプログラム集 2-

新井一郎・立松芳典 [担当]

(第1版)

目次

第1章 サンプルプログラム	3
1.1 研究課題: いろいろなポテンシャル -グラフィックスの応用-	4
1.1.1 概要	4
1.1.2 出力例	4
1.2 研究課題: いろいろな振り子 -微分方程式の応用-	6
1.2.1 概要	6
1.2.2 出力例	7
1.3 研究課題: Lévy フライト -モンテカルロ法の応用-	11
1.3.1 概要	11
1.3.2 出力例	12
1.4 研究課題: ロジスティック写像 -差分方程式の応用-	16
1.4.1 概要	16
1.4.2 出力例	17

第1章 サンプルプログラム

ここでは、大レポートの研究課題の候補となるようなプログラムの実例を示す。これらのプログラムをヒントにしてぜひ自分のオリジナル版を作成してほしい。

1.1 研究課題:いろいろなポテンシャル -グラフィックスの応用-

1.1.1 概要

物理学に現れるいろいろなポテンシャルは、それを適当な座標のうえに関数としてプロットすると、<力の働き>という抽象的なイメージを、「山」や「谷」といった具体的な<風景>として表象することができ、直観的な理解を容易にしてくれる。

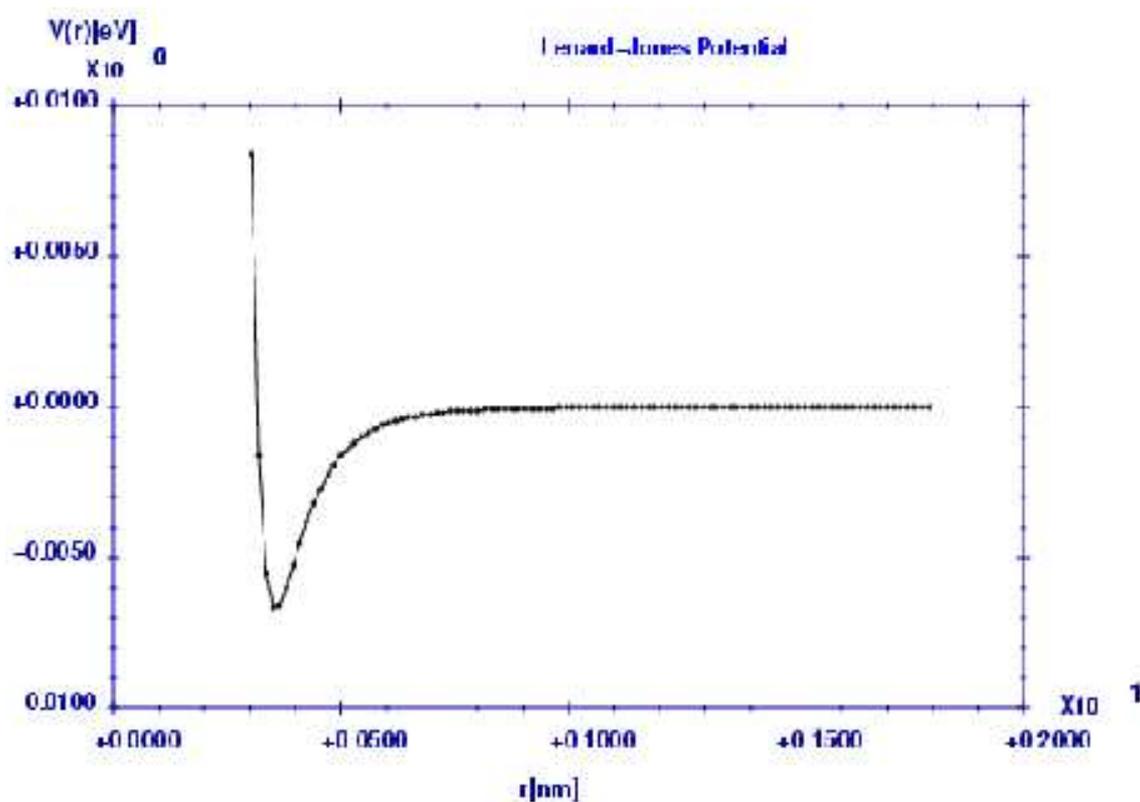
一例として、分子間力ポテンシャルとしてよく知られたレナード・ジョーンズ型ポテンシャル

$$V_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

を描画してみよう。ここで、 ϵ と σ は物質によって異なるパラメータである。水の分子 (H_2O) の場合、 $\epsilon = 0.6501696 \text{ kJ/mol}$ (0.006738535 eV) および $\sigma = 0.3165555 \text{ nm}$ となる

1.1.2 出力例

相対距離 r の関数としてプロットした水の分子間力ポテンシャル。



```
C   Sample [Chapter 1, Study 1]
C       Drawing a Lennard-Jones Potential
C
C   Parameters for Water Molecules
PARAMETER(eps =0.0067385)
PARAMETER(sig =0.3165555)
PARAMETER(r_off=0.2900000)
DIMENSION SCALE(3,2),X(100),Y(100)
CHARACTER*8 NAME(2)
DATA NAME/' r[nm] ', 'V(r)[eV]'/
DATA XBASE/ 0./,DXSTP/0.1/,XDISP/25./
DATA YBASE/-0.01/,DYSTP/0.001/,YDISP/15./
C   Initialization
CALL GRINIT
C   Write title
CALL SETCRS(300.,10.)
CALL PRTCHR('Lennard-Jones Potential',23)
C   Calculate Lennard-Jones Potential
NPNT=100
DO 10 I=1, NPNT
    X(I)=FLOAT(I)*0.015+r_off
    Y(I)=4.*eps*((sig/X(I))**12-(sig/X(I))**6)
10 CONTINUE
C   Draw graph
SCALE(1,1)=XBASE
SCALE(2,1)=DXSTP
SCALE(3,1)=XDISP
SCALE(1,2)=YBASE
SCALE(2,2)=DYSTP
SCALE(3,2)=YDISP
CALL DRWAXS(SCALE,NAME,1)
CALL DRWGRA(X,Y, NPNT,SCALE,4)
C   End graph
CALL GREND
STOP
END
```

1.2 研究課題:いろいろな振り子 -微分方程式の応用-

1.2.1 概要

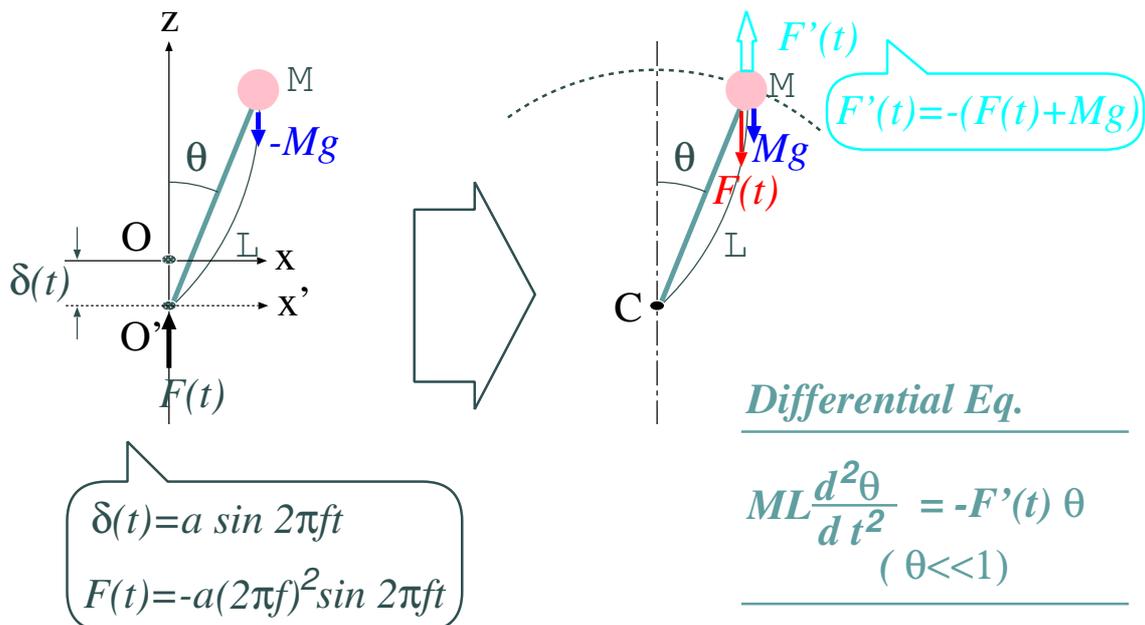
振り子は単純であるが、物理学で取り扱ういろいろな問題に現れる普遍的な力学系である。その理由は、力学的な平衡点付近の運動を表すポテンシャルが「鍋の底」のような形をしていることと深い関係がある。

ところで、振り子に外部からの駆動力が働く場合や複数の振り子を結合した場合などでは、問題は相当に複雑になって、注意深く吟味しないと直観的な理解は難しい。例えば、

- ぶらんこ
- 倒立振り子
- 二重振り子

などが挙げられる。

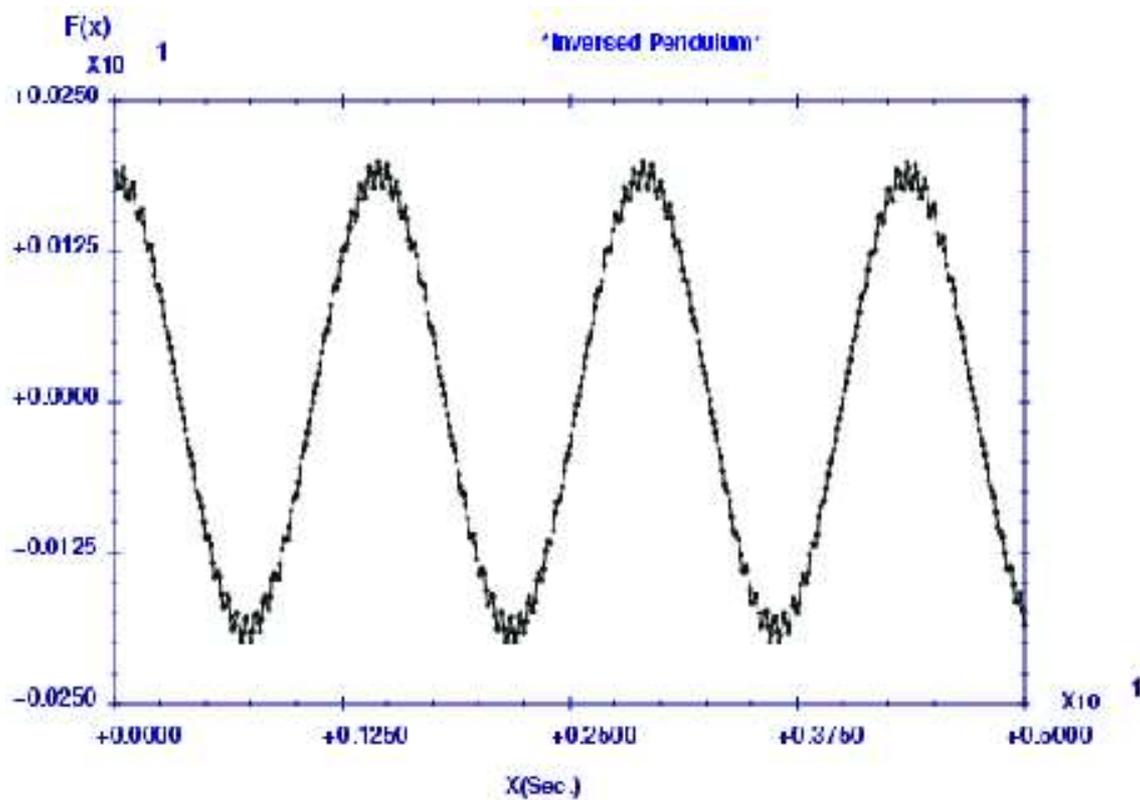
ここでは、倒立振り子を取り上げる。図のように、質量 M の質点を変形しない長さ L の棒の一端に固定し、棒のもう一端を鉛直方向に振幅 a 、周波数 f の \sin 関数にしたがって駆動する。すると、棒は質点を上にして<逆立ち>したまま、ゆっくりと左右に揺れる周期運動を始める。ここに、逆立ちして揺れる振り子 (<倒立振り子>) が出現する。力学的な取り扱いは、図に示す通りである。支点を座標原点 O' とする。支点を駆動することは座標系を動かすことに等しい。したがって、支点を C に固定した座標系では、駆動力と同じ大きさで向きが反対の慣性力が質点に作用する。



また、 a と f は系の制御パラメータである。これらを適当に変えることによって、振り子の運動は劇的に変化する。

1.2.2 出力例

$a = 6\text{cm}$ 、 $f = 20\text{Hz}$ にとった場合の倒立振り子 ($M = 1\text{kg}$, $L = 1\text{m}$) の運動。



```

C*****
C   SAMPLE PROGRAM
C       -SOLUTION OF INVERSED PENDULUM EQUATION-
C           BY RUNGE-KUTTA METHOD
C*****
C>>>>VARIABLES
      EXTERNAL F$BES
      DIMENSION SOL$Y(2)
      DIMENSION S1$DY(2),S2$DY(2),S3$DY(2),S4$DY(2)
C>>>>INITIAL CONDITIONS
C   BES$0 : Position(=Angle)
C   DBES$0: Velocity(=Angular Velocity)
      DATA BES$0/0.2/,DBES$0/0./
C>>>>INITIAL CONDITION FOR THE VARIABLE X(=Time)
      DATA X$0/0./
C>>>>PARAMETERS FOR ITERATION
      DATA D$X/0.001/,N$REP/5000/
C
C>>>>PARAMETERS FOR GRAPHICS
      DIMENSION SCALE(3,2),X(500),Y(500)
      CHARACTER*8 NAME(2)/'X(Sec.) ',' F(x) '/
      DATA XBASE/ 0. /,DXSTP/0.25 /,XDISP/25./
      DATA YBASE/-0.25/,DYSTP/0.025/,YDISP/15./
      DATA NPNT/500/
C>>>>INITIALIZATION
      CALL GRINIT
C>>>>WRITE TITLE
      CALL SETCRS(300.,10.)
      CALL PRTCHR('*Inversed Pendulum*',19)
C>>>>CALCULATE SOLUTION
C
C+++++
C+BY RUNGE-KUTTA METHOD +
C+++++
C>>>>NUMERICAL SOLUTION:BY RUNGE-KUTTA METHOD
      XS=X$0
      BES =BES$0
      BESP=DBES$0
      WRITE(6,1200)
1200 FORMAT(1H , ' -SOLUTION OF BESSEL EQUATION BY R-KUTTA METHOD',/
+         1H ,'|      X      |      Y      |      DY/DX      |',/
+         1H ,'|-----|')
      WRITE(6,2000) XS,BES,BESP
      IBUF = 1
      DO 30 I=1,N$REP

```

```

SOL$X  =XS
SOL$Y(1)=BES
SOL$Y(2)=BESP
CALL F$STEP(D$X,SOL$X,SOL$Y,S1$DY,F$BES)
SOL$X  =XS +0.5*D$X
SOL$Y(1)=BES +0.5*S1$DY(1)
SOL$Y(2)=BESP+0.5*S1$DY(2)
CALL F$STEP(D$X,SOL$X,SOL$Y,S2$DY,F$BES)
SOL$X  =XS +0.5*D$X
SOL$Y(1)=BES +0.5*S2$DY(1)
SOL$Y(2)=BESP+0.5*S2$DY(2)
CALL F$STEP(D$X,SOL$X,SOL$Y,S3$DY,F$BES)
SOL$X  =XS + D$X
SOL$Y(1)=BES + S3$DY(1)
SOL$Y(2)=BESP+ S3$DY(2)
CALL F$STEP(D$X,SOL$X,SOL$Y,S4$DY,F$BES)
XS  =XS +D$X
BES =BES +(S1$DY(1)+2.*S2$DY(1)+2.*S3$DY(1)+S4$DY(1))/6.
BESP=BESP+(S1$DY(2)+2.*S2$DY(2)+2.*S3$DY(2)+S4$DY(2))/6.
IF(MOD(I,10).EQ.0) THEN
  X(IBUF) = XS
  Y(IBUF) = BES
  IBUF = IBUF+1
  WRITE(6,2000) XS,BES, BESP
ENDIF
30 CONTINUE
2000 FORMAT(1H , '| ',F5.2,10X, '| ',2(F10.5,5X, '| '))
C>>>>>DRAW GRAPH OF SINE FUNCTION
SCALE(1,1)=XBASE
SCALE(2,1)=DXSTP
SCALE(3,1)=XDISP
SCALE(1,2)=YBASE
SCALE(2,2)=DYSTP
SCALE(3,2)=YDISP
CALL DRWAXS(SCALE,NAME,1)
CALL DRWGRA(X,Y,NPNT,SCALE,4)
C>>>>>END GRAPGH
CALL GREND
STOP
END
FUNCTION F$BES(X,B,BP,I)
C *****
C X --> t; B --> y; BP --> v=dy/dt
C 1)Pendulum : L=1m; M=1kg
C 2)Variable : Theta (=y)

```

```

C      2)Diff. Eq.: dy/dt=v
C              dv/dt=-f'(x)y
C              =(f(x)+Mg)y
C      *****
      DATA GA/9.8/
C>>>>>RIGHTHAND SIDE OF DIFF. EQUATIONS
      IF(I.EQ.1) THEN
          F$BES=BP
      ELSE IF(I.EQ.2) THEN
          F$BES=(F(X)+GA)*B
      ENDIF
      RETURN
      END
      FUNCTION F(X)
      PARAMETER (a = 0.06)
      PARAMETER (freq= 20. )
C      *****
C      1)a : Amplitude(=Maximum Drive)
C      2)freq: Frequency
C      *****
      DATA PI/3.14/
      AMP = a*(2*PI*freq)**2
      F =-AMP*COS(2*PI*freq*X)
      RETURN
      END
      SUBROUTINE F$STEP(DX,X,Y,DY,FUNC)
      DIMENSION Y(2),DY(2)
      B =Y(1)
      BP=Y(2)
      DO 10 I=1,2
          DY(I)=DX*FUNC(X,B,BP,I)
10 CONTINUE
      RETURN
      END

```

1.3 研究課題: Lévy フライト -モンテカルロ法の応用-

1.3.1 概要

酔歩は、一見したところでは乱数を使った遊びであるが、物理学で取り扱ういろいろな問題、とくにランダムな動きに起因する現象の持つ普遍性を理解するのにとても有用なモデルとなっている。それは、原子や分子含むミクロな物理世界では、〈ランダムネス〉が本質的な役割を演じていることと深い関係がある。

ところで、酔歩には歩幅が一定の値をとる〈離散的酔歩〉と歩幅が連続的に変化する値をとる〈連続的酔歩〉とがある。さらに、連続的酔歩では、たとえば、次のような確率にしたがって歩幅が変化する。

- 指数関数的確率: $P(x) \propto e^{-x/\lambda}$
- べき関数的確率: $P(x) \propto x^{-\gamma}$

ここでは、べき関数的確率にしたがう酔歩、いわゆる〈Lévy フライト〉を取り上げる。各試行のフライト距離は次の確率で計算する。

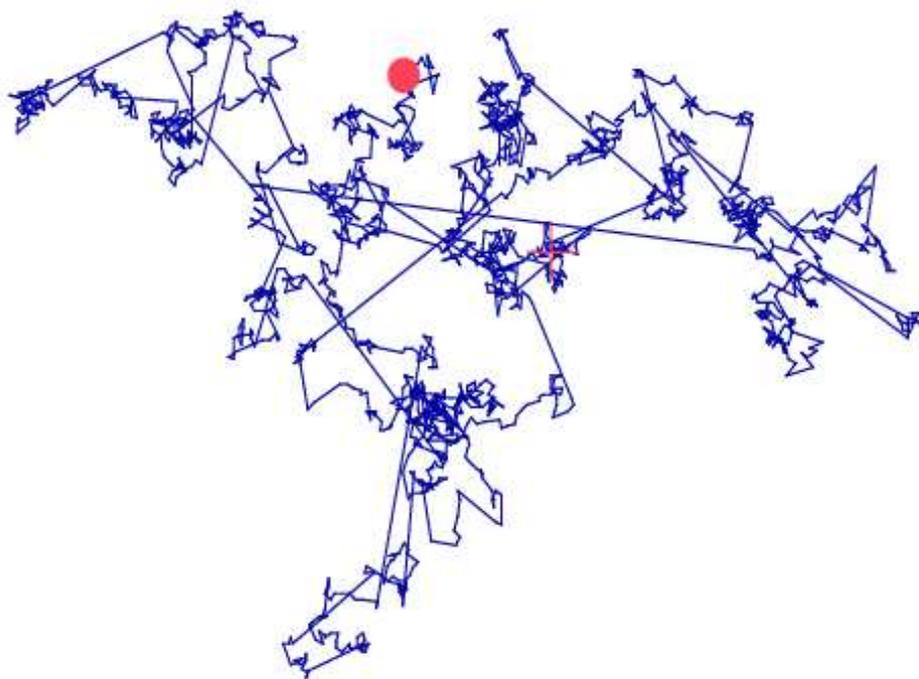
$$p(x) = \frac{\lambda - 1}{\lambda} \sum_{j=0}^{j_{max}} \lambda^{-j} \delta_{x, bj} \quad (1.1)$$

ここで、 $\lambda > b > 1$ である。複雑に見えるが、この式は $\gamma = \ln \lambda / \ln b$ としたときのべき関数的確率 $p(x) \propto x^{-\gamma}$ を与える。

1.3.2 出力例

$\lambda = 2.83$ 、 $b = 2.0$ 、 $\gamma = 1.5$ にとった場合の Lévy フライトの軌跡。

- 3-Dim Levys Flight -



```

C*****
C*   3 DIM. Levy's Flight *
C*           -MONTE CARLO SIMULATION *
C*   Probability for Flight delta *
C*       x   = b**j *
C*       P(x) =N lambda**(-j) *
C*           =N' x**(-gamma) *
C*   Parameters *
C*       lambda: U_Lamb=2.83 *
C*       b      : U_Step=2.0 *
C*           ----> gamma=ln(lambda)/ln(b) *
C*                   =1.5 *
C*****
C>>>>UNIVERSAL CONSTANT
      DATA PAI/3.14/
C>>>>INITIAL VALUE FOR RANDOM NUMBER GENERATOR
      DATA IX/100/
C>>>>PARAMETERS FOR Levi's Flight
C   -PROBABILITY
      DIMENSION P(100)
C   -INITIAL POSITION
      DATA XO/0./,YO/0./,ZO/0./
C   -Unit Step
      DATA U_Step/2.0/
C   -Unit Probability
      DATA U_Lamb/2.83/
C   -Maximum Steps
      DATA M_Step/100/
C   -NUMBER OF TRIAL
      DATA NCOLL/2000/,NPART/ 1/,NINTR/ 1/
C   -NUMBER OF TRIAL(WITH SKIP)
      DATA NCOLL/100/,NPART/ 2/,NINTR/ 5/
C>>>>PARAMETERS FOR GRAPHIC WINDOW
      DATA V_XO/- 60./,V_YO/-36./,V_X1/ 60./,V_Y1/ 36./
C>>>>INITIALIZATION
      CALL GRINIT
C>>>>DEFINE VIRTUAL GRAPHIC WINDOW
      CALL V_WIND(V_XO,V_YO,V_X1,V_Y1)
C>>>>WRITE TITLE
      CALL SETCRS(120.,30.)
      CALL PRTPCHR('- 3-Dim Levys Flight -',23)
C>>>>MONTE CARLO SIMULATION
C   -Initialization for Random Number Generator
      sum = 1.
      DO 1 I=1,M_Step

```

```

        sum = sum+U_Lamb**(-(i-1))
        p(i)= sum
1 CONTINUE
    DO 2 I=1,M_Step
        p(i)= p(i)/sum
2 CONTINUE
    DO 20 I=1,NPART
C      -INITIAL POSITION
        X=X0
        Y=Y0
        Z=Z0
C      -INITIAL DIRECTION COSINE
        CALL RAND_U(IX,RN)
        COSTH=1.-2.*RN
        SINTH=SQRT(1.-COSTH**2)
        CALL RAND_U(IX,RN)
        PHAI =2.*PAI*RN
        DX  =SINTH*COS(PHAI)
        DY  =SINTH*SIN(PHAI)
        DZ  =COSTH
C      -INITIAL POSITION FOR GRAPHICS(X-Y PROJECTION)
        XS=X0
        YS=Y0
        DO 10 J=1,NCOLL
C      -FLIGHT LENGTH
            CALL RAND_U(IX,RN)
            iflg=0
            do 5 k=1,M_Step
                if(iflg.eq.0.and.RN.le.p(k)) then
                    istp=k-1
                    iflg=1
                endif
5          continue
            FLIGHT=U_Step**istp
            X    =X+FLIGHT*DX
            Y    =Y+FLIGHT*DY
            Z    =Z+FLIGHT*DZ
            IF(MOD(J,NINTR).EQ.0) THEN
                CALL V_LINE(XS,YS,X,Y,1)
                XS=X
                YS=Y
            ENDIF
C      -CHANGE DIRECTION
            CALL RAND_U(IX,RN)
            COSTH=1.-2.*RN

```

```

        SINTH=SQRT(1.-COSTH**2)
        CALL RAND_U(IX,RN)
        PHAI =2.*PAI*RN
        DX   =SINTH*COS(PHAI)
        DY   =SINTH*SIN(PHAI)
        DZ   =COSTH
10      CONTINUE
        CALL V_LINE(-3., 0., 3., 0.,2)
        CALL V_LINE( 0.,-3., 0., 3.,2)
        CALL PAINTF(1)
        CALL V_CIRC(XS,YS,2.,2,0.,0.)
        CALL PAINTF(0)
20      CONTINUE
C>>>>>END GRAPH
        CALL GREND
        STOP
        END
        SUBROUTINE RAND_U(IX,RN)
C+++++
C+   UNIFORM RANDOM NUMBER GENERATOR           +
C+       IX = (A*IX+B) MOD N                     +
C+       RN = IX/N                               +
C+   IX      : PRECEDING/FOLLOWING RANDOM NUMBER +
C+   RN      : NORMALIZED RANDOM NUMBER         +
C+++++
        DATA MODULO/1000000/,IA/111/,IB/9/
        IX = MOD(IA*IX+IB,MODULO)
        RN = FLOAT(IX)/FLOAT(MODULO)
        RETURN
        END

```

1.4 研究課題: ロジスティック写像 -差分方程式の応用-

1.4.1 概要

ロジスティック写像は、差分方程式

$$x_{n+1} = rx_n(1 - x_n) \quad (0 \leq r \leq 4) \quad (1.2)$$

によって与えられ、カオスと深い関係がある。とくに、特定の初期値 x_0 から出発した点の集まり $\{x_i; i = 0, 1, 2, \dots, n, \dots, 0\}$ を軌道と呼ぶ。

この方程式は唯一のパラメータ r をもつ簡単な2次写像であるにもかかわらず、 r の値を変えることによって、軌道が倍周期分岐によるカオスへの推移過程を明解に示すなどの際だった特徴をもち、カオス理解にとって大変重要な課題の1つである。ここでは、パラメータ r の値を変えていったときのこの写像のアトラクタをプロットすることを通じて、カオスの生成過程を観察してみよう。アトラクタというのは十分多数回の繰り返しの後に軌道が落ち着く先のことである。アトラクタは関係式

$$x_{n+i} = f^n(x_i); f(x) = rx(1 - x) \quad (1.3)$$

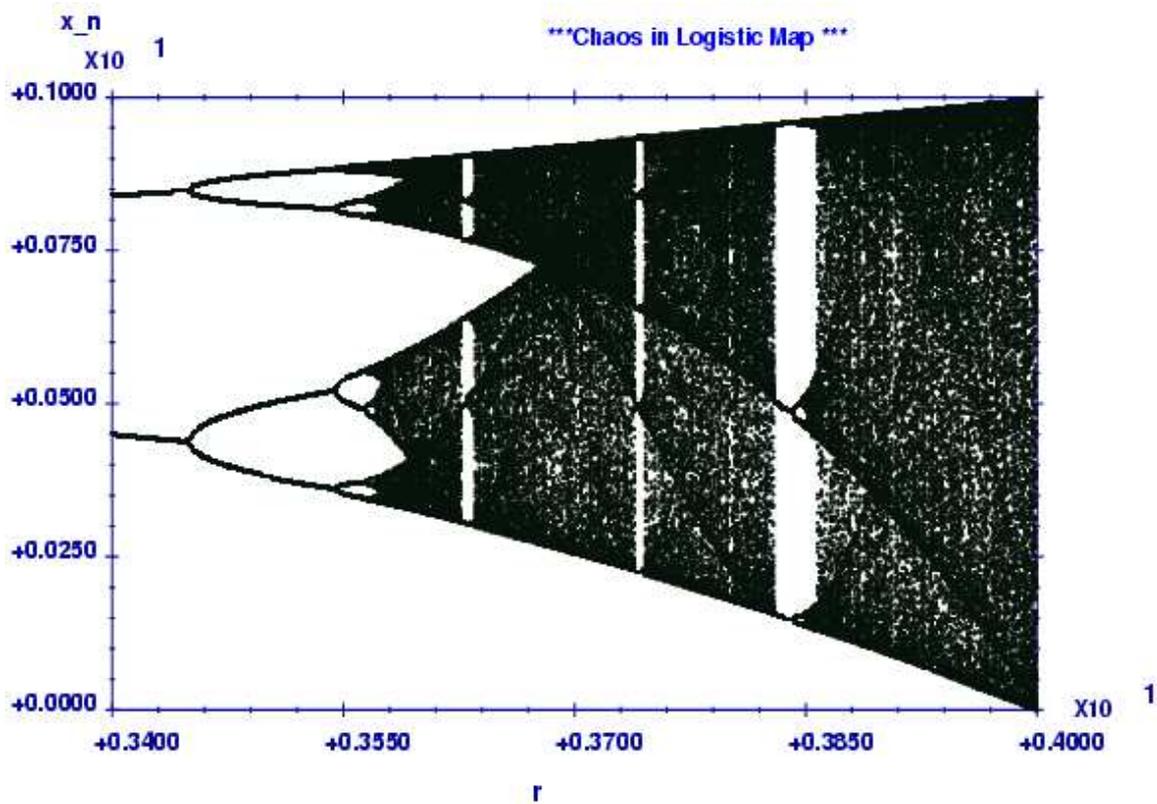
を満たすとき n -周期点と呼ばれる。一方、カオス的なアトラクタでは、このような関係を見出せない。

1.4.2 出力例

パラメータ r を 3.4 から 4.0 まで変えたときのアトラクタ。

⚠注意) この出力は、そのまま解像度が悪いので、vi エディタで出力ファイル fort.1 を一部訂正するとよいだろう。やりかたは以下の通りである。

```
%vi fort.1
:s/2.,4/1.,4/g
:wq!
%
```



```

C*****
C*          -Logistic Map-                               *
C*          <Equation>                                   *
C*                                                  *
C*           $x(n+1) = a*x(n)*(1-x(n))$                    *
C*           $0 < a \leq 4$                                    *
C*****
C>>>>PARAMETERS FOR GRAPHICS
      DIMENSION SCALE(3,2)
      CHARACTER*8 NAME(2)/' r      ',' x_n  '/
      DATA XBASE/ 3.4/,DXSTP/0.03/,XDISP/25./
      DATA YBASE/ 0./,DYSTP/0.05/,YDISP/15./
C>>>>Initial Value
      DATA AO/3.400/
      DATA DA/0.001/
      DATA XO/0.7/
C>>>>Data
      DIMENSION y(500)
C>>>>Trials
      DATA nskp/500/
      DATA ntry/200/
      DATA na /600/
C>>>>Initialization
      CALL GRINIT
C>>>>Write Title
      CALL SETCRS(300.,10.)
      CALL PRTCHR('***Chaos in Logistic Map ***',29)
      NPNT=1
C  -Simulation
C>>>>DRAW GRAPH OF UNIFORM RANDOM NUMBER
      SCALE(1,1)=XBASE
      SCALE(2,1)=DXSTP
      SCALE(3,1)=XDISP
      SCALE(1,2)=YBASE
      SCALE(2,2)=DYSTP
      SCALE(3,2)=YDISP
      CALL DRWAXS(SCALE,NAME,1)
      DO 400 ia=1,na
        a = a0+da*(ia-1)
        x$skp= X0
        DO 100 LOOP=1,NSKP
          x$skp = a*x$skp*(1.-x$skp)
100    CONTINUE
        y(1) = a*x$skp*(1.-x$skp)
        x$skp = y(1)

```

```
      DO 200 LOOP=2,NTRY
        x$skp = a*x$skp*(1.-x$skp)
        y(loop) = x$skp
200    CONTINUE
      NPNT = NTRY
      DO 300 L=1,NPNT
        CALL DRWGRA(a,y(1),1,SCALE,4)
300    CONTINUE
400    CONTINUE
C>>>>>END GRAPGH
      STOP
      END
```