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

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

ここでは、これまでの各章でとり上げた研究課題にたいする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:sin 関数

1.1.1　概要

関数のグラフを描くには

- x-軸の描画、目盛付け、名前（ラベル）付け
- y-軸の描画、目盛付け、名前（ラベル）付け
- グラフの名前（タイトル）
- グラフの描画（プロット）

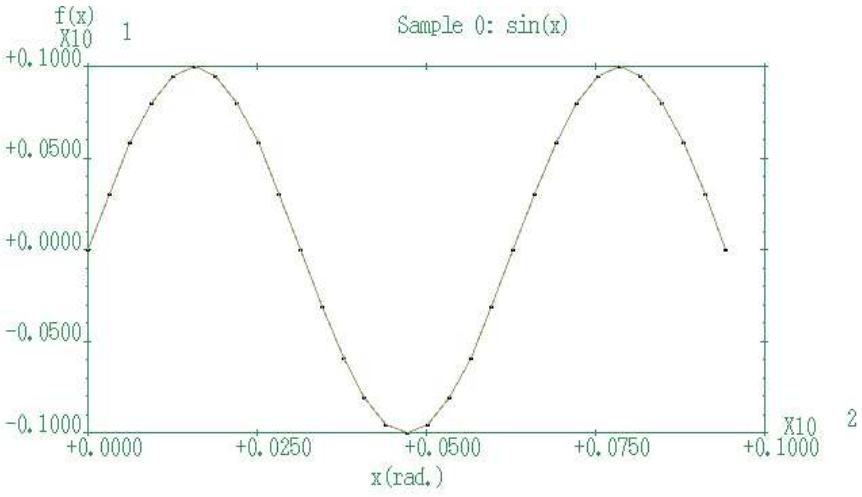
が必要となる。ここでは、サブルーチン群 *graphics.f* がこれらの仕事を行なってくれる。したがって、メイン・プログラムでは必要な情報

- x-軸の名前（xlabel)、範囲（xrange(2))、目盛の巾（dx）
- y-軸の名前（ylabel)、範囲（yrange(2))、目盛の巾（dy）
- グラフの名前（title）
- 描画データ数（n)、描画データ（data(n)）

を用意して、該当するサブルーチンをコールすればよい。

1.1.2　出力例

一例として、sin 関数の描画をとり上げる。



```
/*     Sample Program [Chapter 2, Example 1]       */
/*            Drawing a sine function              */

#include  <stdio.h>
#include  <math.h>

#define   PI      3.14159
#define   NPNT    31

main()
{
  double scale[2][3],x[100],y[100];
  char name[2][8] ={"X(RAD.) "," F(x)  "};
  double xbase= 0., dxstp=0.50, xdisp=25.;
  double ybase=-1., dystp=0.10, ydisp=15.;
  int i;
  void grinit(),setcrs(),prtchr(),drwaxs(),drwgra(),grend();

  /* Initialization */
  grinit();

  /* Write title    */
  setcrs(300.,10.);
  prtchr("*****Sample0*****",17);
  /* Calculate sin function */
  for(i=0;i<NPNT;++i){
    x[i]=(double)i*PI/10.;
    y[i]=sin(x[i]);
  }

  /* Draw graph of sin function */
  scale[0][0]=xbase;
  scale[0][1]=dxstp;
  scale[0][2]=xdisp;
  scale[1][0]=ybase;
  scale[1][1]=dystp;
  scale[1][2]=ydisp;
  drwaxs(scale,name,1);
  drwgra(x,y,NPNT,scale,4);

  /* End graph */
  grend();
}
```

1.2　研究課題：1 次元ヒストグラム

1.2.1　概要

ヒストグラム（度数分布）をとるには

- 変数の範囲（[下限:上限]）の設定とますの幅の指定
- ヒストグラムの作成
- ヒストグラムの印刷（プリント）

が必要となる。ここでは、サブルーチン $hbook(x, hpar, harray, hstat, isw)$ がこれらの仕事を行なってくれる。したがって、メイン・プログラムでは必要な情報

- 変数の値（x）
- 変数の下限、上限、ますの幅、ますの数（hpar(4)）
- ヒストグラムを格納する配列（harray()）
- ヒストグラムの統計を格納する配列（hstat(3)）
- 制御スイッチ（isw）$isw = 1$:ヒストグラムの作成、$isw \neq 1$:印刷

を用意して、サブルーチンをコールすればよい。

1.2.2　出力例

一例として、正規分布に従うヒストグラムの作成・印刷をとり上げる。

```
 VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                         0       100       200       300       400       500
  0.0000                 +----+----+----+----+----+----+----+----+----+----+
  0.0200          85     +    |   *|    |    |    |    |    |    |    |    +
  0.0400          94     +    |   *|    |    |    |    |    |    |    |    +
  0.0600         103     +    |    *    |    |    |    |    |    |    |    +
  0.0800         112     +    |    |*   |    |    |    |    |    |    |    +
  0.1000         123     +----|----|-*--|----|----|----|----|----|----|----+
  0.1200         132     +    |    |  * |    |    |    |    |    |    |    +
  0.1400         142     +    |    |   *|    |    |    |    |    |    |    +
  0.1600         154     +    |    |    *    |    |    |    |    |    |    +
  0.1800         165     +    |    |    | *  |    |    |    |    |    |    +
  0.2000         175     +----|----|----|--*-|----|----|----|----|----|----+
  0.2200         186     +    |    |    |   *|    |    |    |    |    |    +
  0.2400         196     +    |    |    |    *    |    |    |    |    |    +
  0.2600         207     +    |    |    |    |*   |    |    |    |    |    +
  0.2800         217     +    |    |    |    | *  |    |    |    |    |    +
  0.3000         228     +----|----|----|----|--*-|----|----|----|----|----+
  0.3200         237     +    |    |    |    |   *|    |    |    |    |    +
  0.3400         246     +    |    |    |    |    *    |    |    |    |    +
  0.3600         253     +    |    |    |    |    *    |    |    |    |    +
  0.3800         261     +    |    |    |    |    |*   |    |    |    |    +
  0.4000         267     +----|----|----|----|----|-*--|----|----|----|----+
  0.4200         273     +    |    |    |    |    | *  |    |    |    |    +
  0.4400         277     +    |    |    |    |    |  * |    |    |    |    +
  0.4600         280     +    |    |    |    |    |  * |    |    |    |    +
  0.4800         282     +    |    |    |    |    |  * |    |    |    |    +
  0.5000         283     +----|----|----|----|----|--*-|----|----|----|----+
  0.5200         283     +    |    |    |    |    |  * |    |    |    |    +
  0.5400         282     +    |    |    |    |    |  * |    |    |    |    +
  0.5600         280     +    |    |    |    |    |  * |    |    |    |    +
  0.5800         277     +    |    |    |    |    |  * |    |    |    |    +
  0.6000         273     +----|----|----|----|----|-*--|----|----|----|----+
  0.6200         267     +    |    |    |    |    | *  |    |    |    |    +
  0.6400         261     +    |    |    |    |    |*   |    |    |    |    +
  0.6600         253     +    |    |    |    |    *    |    |    |    |    +
  0.6800         246     +    |    |    |    |    *    |    |    |    |    +
  0.7000         237     +----|----|----|----|---*|----|----|----|----|----+
  0.7200         228     +    |    |    |    |  * |    |    |    |    |    +
  0.7400         217     +    |    |    |    | *  |    |    |    |    |    +
  0.7600         207     +    |    |    |    |*   |    |    |    |    |    +
  0.7800         196     +    |    |    |    *    |    |    |    |    |    +
  0.8000         186     +----|----|----|---*|----|----|----|----|----|----+
  0.8200         175     +    |    |    |  * |    |    |    |    |    |    +
  0.8400         165     +    |    |    | *  |    |    |    |    |    |    +
  0.8600         154     +    |    |    *    |    |    |    |    |    |    +
  0.8800         142     +    |    |   *|    |    |    |    |    |    |    +
  0.9000         132     +----|----|--*-|----|----|----|----|----|----|----+
  0.9200         123     +    |    | *  |    |    |    |    |    |    |    +
  0.9400         112     +    |    |*   |    |    |    |    |    |    |    +
  0.9600         103     +    |    *    |    |    |    |    |    |    |    +
  0.9800          94     +    |   *|    |    |    |    |    |    |    |    +
  1.0000          85     +----|---*|----|----|----|----|----|----|----|----+
  1.0200                 +----+----+----+----+----+----+----+----+----+----+
   *  0.1000e+01
   <NUMBER OF ENTRIES> TOT:    11205.
               |          |
   UNDER RANGE | IN RANGE | OVER RANGE
          624. |    9956. |     625.
```

正規分布のヒストグラム

```
/*     Sample Program [Chapter 3, Study 1] */
/*            Making a histogram           */

#include  <stdio.h>
#include  <math.h>

#define   PI      3.14159
#define   X0      -0.5
#define   DX      0.02

main()
{
  /* Arrays for histogram */
  double hpar_g[4]={0.,1.,0.02,51.};
  int harr_g[50],hsta_g[3];

  /* Gaussian distribution */
  double mean_g=0.5,var_g=0.1,norm_g=150.;

  double factor,x;
  int i,j,nloop;
  void hbook();

  /* Set Normalization Factor*/
  factor = norm_g*sqrt(1./(2.*PI*var_g));

  /* Initialize Buffer */
  for(i=0;i<50;i++)
    harr_g[i]=0;
  for(i=0;i<3;i++)
    hsta_g[i]=0;

  /* Calculation and Booking */
  for(i=0;i<100;i++){
    x=X0+DX*(double)i+0.5*DX;
    nloop=(int)floor(factor*exp(-(x-mean_g)*(x-mean_g)/(2.*var_g)));
    nloop*=3;nloop/=2;
    for(j=0;j<nloop;j++)
      hbook(x,hpar_g,harr_g,hsta_g,1);
  }

  /* Print out */
  hbook(x,hpar_g,harr_g,hsta_g,0);
}
```

1.3 研究課題:一様乱数の生成

1.3.1 概要

一様乱数を生成するには

- 乱数の種（$seed$）の入力
- 乱数の生成と $seed$ の更新
- 乱数および更新された $seed$ の出力

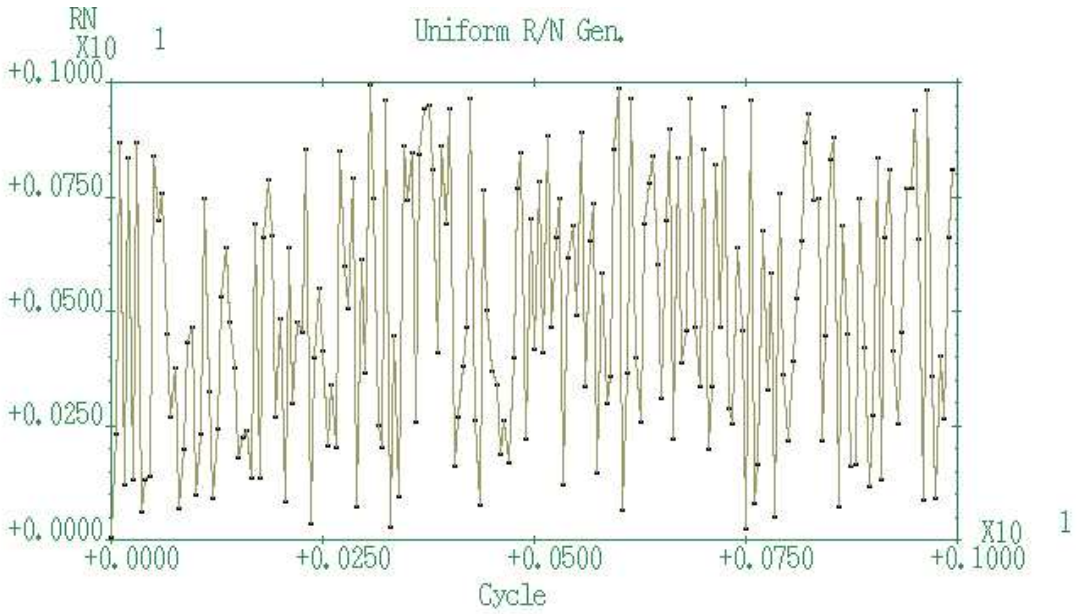
が必要となる。ここでは、サブルーチン $rand\$u(ix)$ がこれらの仕事を行なってくれる。したがって、メイン・プログラムでは必要な情報

- 現在の $seed$ の値／更新された $seed$ の値（ix）
- 生成された乱数を受け取る変数（rn）

を用意して、サブルーチンをコールすればよい。

1.3.2 出力例

一例として、一様乱数の系列プロットをとり上げる。



一様乱数の系列

```
/*     Sample Program [Chapter 4, Study 1]        */
/*           Making a uniform pseudorandom number */

#include <stdio.h>
#include <math.h>

#define  NPNT   200
#define  DX     0.005

main()
{
  /* Parameters for graphics */
  double scale[2][3],x[500],y[500];
  char name[2][8] ={" CYCLE  "," RN    "};
  double xbase= 0., dxstp=0.05, xdisp=25.;
  double ybase= 0., dystp=0.05, ydisp=15.;
  int i,ix;
  double RN();
  void grinit(),setcrs(),prtchr(),drwaxs(),drwgra(),grend();

  /* Initialize RN generator */
  ix=100;

  /* Initialization */
  grinit();

  /* Write title    */
  setcrs(300.,10.);
  prtchr("Uniform R/N Gen. ",17);

  /* Calculate sin function */
  for(i=0;i<NPNT;++i){
    x[i]=(double)i*DX;
    y[i]=RN(&ix);
  }
  /* Draw graph of uniform random number */
  scale[0][0]=xbase;
  scale[0][1]=dxstp;
  scale[0][2]=xdisp;
  scale[1][0]=ybase;
  scale[1][1]=dystp;
  scale[1][2]=ydisp;
  drwaxs(scale,name,1);
  drwgra(x,y,NPNT,scale,4);
  /* end grapgh */
```

```
  grend();
}

double RN(ix)
int *ix;
{
  *ix = 111*(*ix)+9;
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.4　研究課題:いろいろな乱数の生成

1.4.1　概要

一様乱数をもとにしていろいろな乱数が生成できる。生成方法としては相加平均法、逆関数法および棄却法などがある。テキストでは、相加平均法による正規乱数の生成と逆関数法による指数乱数の生成をとり上げたが、棄却法についても是非試してみて欲しい。

1.4.2　出力例

一例として、正規乱数の生成と指数乱数の生成をとり上げる。

```
 VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                        0        50       100       150       200       250
  0.0000                +----+----+----+----+----+----+----+----+----+----+
  0.0200        213     +    |    |    |    |    |    |    |    |  *  |    +
  0.0400        212     +    |    |    |    |    |    |    |    | *   |    +
  0.0600        191     +    |    |    |    |    |    |    |  * |     |    +
  0.0800        219     +    |    |    |    |    |    |    |    |   *|    +
  0.1000        193     +----|----|----|----|----|----|----|---*|----|----+
  0.1200        185     +    |    |    |    |    |    |    | *  |    |    +
  0.1400        173     +    |    |    |    |    |    |    *    |    |    +
  0.1600        194     +    |    |    |    |    |    |    |   *|    |    +
  0.1800        221     +    |    |    |    |    |    |    |    |   *|    +
  0.2000        196     +----|----|----|----|----|----|----|---*|----|----+
  0.2200        187     +    |    |    |    |    |    |    | *  |    |    +
  0.2400        187     +    |    |    |    |    |    |    | *  |    |    +
  0.2600        189     +    |    |    |    |    |    |    |  * |    |    +
  0.2800        200     +    |    |    |    |    |    |    |    *    |    +
  0.3000        206     +----|----|----|----|----|----|----|----|*---|----+
  0.3200        196     +    |    |    |    |    |    |    |   *|    |    +
  0.3400        184     +    |    |    |    |    |    |    | *  |    |    +
  0.3600        188     +    |    |    |    |    |    |    |  * |    |    +
  0.3800        206     +    |    |    |    |    |    |    |    |*   |    +
  0.4000        206     +----|----|----|----|----|----|----|----|*---|----+
  0.4200        195     +    |    |    |    |    |    |    |   *|    |    +
  0.4400        195     +    |    |    |    |    |    |    |   *|    |    +
  0.4600        206     +    |    |    |    |    |    |    |    |*   |    +
  0.4800        235     +    |    |    |    |    |    |    |    |    | *  +
  0.5000        207     +----|----|----|----|----|----|----|----|*---|----+
  0.5200        205     +    |    |    |    |    |    |    |    |*   |    +
  0.5400        210     +    |    |    |    |    |    |    |    | *  |    +
  0.5600        202     +    |    |    |    |    |    |    |    *    |    +
  0.5800        183     +    |    |    |    |    |    |    | *  |    |    +
  0.6000        219     +----|----|----|----|----|----|----|----|---*|----+
  0.6200        210     +    |    |    |    |    |    |    |    | *  |    +
  0.6400        195     +    |    |    |    |    |    |    |   *|    |    +
  0.6600        201     +    |    |    |    |    |    |    |    *    |    +
  0.6800        174     +    |    |    |    |    |    |    *    |    |    +
  0.7000        195     +----|----|----|----|----|----|----|---*|----|----+
  0.7200        213     +    |    |    |    |    |    |    |    |  * |    +
  0.7400        193     +    |    |    |    |    |    |    |   *|    |    +
  0.7600        227     +    |    |    |    |    |    |    |    |    *    +
  0.7800        185     +    |    |    |    |    |    |    | *  |    |    +
  0.8000        186     +----|----|----|----|----|----|----|-*--|----|----+
  0.8200        209     +    |    |    |    |    |    |    |    | *  |    +
  0.8400        196     +    |    |    |    |    |    |    |   *|    |    +
  0.8600        207     +    |    |    |    |    |    |    |    |*   |    +
  0.8800        181     +    |    |    |    |    |    |    |*   |    |    +
  0.9000        199     +----|----|----|----|----|----|----|----*----|----+
  0.9200        201     +    |    |    |    |    |    |    |    *    |    +
  0.9400        224     +    |    |    |    |    |    |    |    |    *    +
  0.9600        184     +    |    |    |    |    |    |    | *  |    |    +
  0.9800        213     +    |    |    |    |    |    |    |    |  * |    +
  1.0000        204     +----|----|----|----|----|----|----|----|*---|----+
  1.0200                +----+----+----+----+----+----+----+----+----+----+
   *  0.1000e+01
   <NUMBER OF ENTRIES> TOT:   10000.
                  |          |
   UNDER RANGE | IN RANGE | OVER RANGE
             0. |   10000. |       0.
```

一様乱数の分布

```
 VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                              0       200       400       600       800      1000
  0.0000                     +----+----+----+----+----+----+----+----+----+----+
  0.0200          0          *    |    |    |    |    |    |    |    |    |    +
  0.0400          0          *    |    |    |    |    |    |    |    |    |    +
  0.0600          0          *    |    |    |    |    |    |    |    |    |    +
  0.0800          0          *    |    |    |    |    |    |    |    |    |    +
  0.1000          0          *----|----|----|----|----|----|----|----|----|----+
  0.1200          0          *    |    |    |    |    |    |    |    |    |    +
  0.1400          0          *    |    |    |    |    |    |    |    |    |    +
  0.1600          0          *    |    |    |    |    |    |    |    |    |    +
  0.1800          0          *    |    |    |    |    |    |    |    |    |    +
  0.2000          1          *----|----|----|----|----|----|----|----|----|----+
  0.2200          2          *    |    |    |    |    |    |    |    |    |    +
  0.2400          2          *    |    |    |    |    |    |    |    |    |    +
  0.2600          6          *    |    |    |    |    |    |    |    |    |    +
  0.2800         25          +*   |    |    |    |    |    |    |    |    |    +
  0.3000         44          +-*--|----|----|----|----|----|----|----|----|----+
  0.3200         74          +   *|    |    |    |    |    |    |    |    |    +
  0.3400        131          +    | *  |    |    |    |    |    |    |    |    +
  0.3600        170          +    |   *|    |    |    |    |    |    |    |    +
  0.3800        309          +    |    |   *|    |    |    |    |    |    |    +
  0.4000        412          +----|----|----|----|*---|----|----|----|----|----+
  0.4200        526          +    |    |    |    |    |*   |    |    |    |    +
  0.4400        652          +    |    |    |    |    |    |  * |    |    |    +
  0.4600        782          +    |    |    |    |    |    |    |   *|    |    +
  0.4800        908          +    |    |    |    |    |    |    |    |   *|    +
  0.5000        936          +----|----|----|----|----|----|----|----|----|-*--+
  0.5200        935          +    |    |    |    |    |    |    |    |    | *  +
  0.5400        891          +    |    |    |    |    |    |    |    |   *|    +
  0.5600        774          +    |    |    |    |    |    |    |   *|    |    +
  0.5800        697          +    |    |    |    |    |    |   *|    |    |    +
  0.6000        556          +----|----|----|----|----|--*-|----|----|----|----+
  0.6200        389          +    |    |    |   *|    |    |    |    |    |    +
  0.6400        297          +    |    |   *|    |    |    |    |    |    |    +
  0.6600        186          +    |   *|    |    |    |    |    |    |    |    +
  0.6800        120          +    |*   |    |    |    |    |    |    |    |    +
  0.7000         81          +---*|----|----|----|----|----|----|----|----|----+
  0.7200         46          + *  |    |    |    |    |    |    |    |    |    +
  0.7400         21          +*   |    |    |    |    |    |    |    |    |    +
  0.7600         12          +*   |    |    |    |    |    |    |    |    |    +
  0.7800         12          +*   |    |    |    |    |    |    |    |    |    +
  0.8000          3          *----|----|----|----|----|----|----|----|----|----+
  0.8200          0          *    |    |    |    |    |    |    |    |    |    +
  0.8400          0          *    |    |    |    |    |    |    |    |    |    +
  0.8600          0          *    |    |    |    |    |    |    |    |    |    +
  0.8800          0          *    |    |    |    |    |    |    |    |    |    +
  0.9000          0          *----|----|----|----|----|----|----|----|----|----+
  0.9200          0          *    |    |    |    |    |    |    |    |    |    +
  0.9400          0          *    |    |    |    |    |    |    |    |    |    +
  0.9600          0          *    |    |    |    |    |    |    |    |    |    +
  0.9800          0          *    |    |    |    |    |    |    |    |    |    +
  1.0000          0          *----|----|----|----|----|----|----|----|----|----+
  1.0200                     +----+----+----+----+----+----+----+----+----+----+
   *  0.1000e+01
   <NUMBER OF ENTRIES> TOT:    10000.
               |           |
   UNDER RANGE | IN RANGE  | OVER RANGE
             0. |    10000. |        0.
```

正規乱数の分布

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                        0       200       400       600       800      1000
 0.0000                 +----+----+----+----+----+----+----+----+----+----+
 0.0080        734      +    |    |    |    |    |    |    | *  |    |    +
 0.0160        749      +    |    |    |    |    |    |    | *  |    |    +
 0.0240        677      +    |    |    |    |    |    |   *|    |    |    +
 0.0320        622      +    |    |    |    |    |    |*   |    |    |    +
 0.0400        559      +----|----|----|----|----|--*-|----|----|----|----+
 0.0480        524      +    |    |    |    |    |*   |    |    |    |    +
 0.0560        470      +    |    |    |    |   *|    |    |    |    |    +
 0.0640        425      +    |    |    |    |*   |    |    |    |    |    +
 0.0720        429      +    |    |    |    |*   |    |    |    |    |    +
 0.0800        393      +----|----|----|----*----|----|----|----|----|----+
 0.0880        373      +    |    |    |   *|    |    |    |    |    |    +
 0.0960        324      +    |    |    |*   |    |    |    |    |    |    +
 0.1040        326      +    |    |    |*   |    |    |    |    |    |    +
 0.1120        258      +    |    |  * |    |    |    |    |    |    |    +
 0.1200        231      +----|----|-*--|----|----|----|----|----|----|----+
 0.1280        237      +    |    | *  |    |    |    |    |    |    |    +
 0.1360        195      +    |    *    |    |    |    |    |    |    |    +
 0.1440        202      +    |    *    |    |    |    |    |    |    |    +
 0.1520        173      +    |   *|    |    |    |    |    |    |    |    +
 0.1600        156      +----|--*-|----|----|----|----|----|----|----|----+
 0.1680        152      +    |  * |    |    |    |    |    |    |    |    +
 0.1760        140      +    | *  |    |    |    |    |    |    |    |    +
 0.1840        124      +    |*   |    |    |    |    |    |    |    |    +
 0.1920        117      +    |*   |    |    |    |    |    |    |    |    +
 0.2000        106      +----*----|----|----|----|----|----|----|----|----+
 0.2080         91      +    *    |    |    |    |    |    |    |    |    +
 0.2160         93      +    *    |    |    |    |    |    |    |    |    +
 0.2240         84      +   *|    |    |    |    |    |    |    |    |    +
 0.2320         76      +   *|    |    |    |    |    |    |    |    |    +
 0.2400         71      +---*|----|----|----|----|----|----|----|----|----+
 0.2480         84      +   *|    |    |    |    |    |    |    |    |    +
 0.2560         51      +  * |    |    |    |    |    |    |    |    |    +
 0.2640         55      +  * |    |    |    |    |    |    |    |    |    +
 0.2720         51      +  * |    |    |    |    |    |    |    |    |    +
 0.2800         46      +-*--|----|----|----|----|----|----|----|----|----+
 0.2880         47      + *  |    |    |    |    |    |    |    |    |    +
 0.2960         51      +  * |    |    |    |    |    |    |    |    |    +
 0.3040         31      + *  |    |    |    |    |    |    |    |    |    +
 0.3120         38      + *  |    |    |    |    |    |    |    |    |    +
 0.3200         36      +-*--|----|----|----|----|----|----|----|----|----+
 0.3280         34      + *  |    |    |    |    |    |    |    |    |    +
 0.3360         36      + *  |    |    |    |    |    |    |    |    |    +
 0.3440         23      +*   |    |    |    |    |    |    |    |    |    +
 0.3520         22      +*   |    |    |    |    |    |    |    |    |    +
 0.3600         18      +*---|----|----|----|----|----|----|----|----|----+
 0.3680         25      +*   |    |    |    |    |    |    |    |    |    +
 0.3760         14      +*   |    |    |    |    |    |    |    |    |    +
 0.3840         17      +*   |    |    |    |    |    |    |    |    |    +
 0.3920         10      +*   |    |    |    |    |    |    |    |    |    +
 0.4000         15      +*---|----|----|----|----|----|----|----|----|----+
 0.4080                 +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+02
  <NUMBER OF ENTRIES> TOT:    10000.
             |          |
  UNDER RANGE | IN RANGE  | OVER RANGE
            0. |    9815.  |     185.
```

指数乱数の分布

```
/*     Sample Program [Chapter 4, Study 3-5]    */
/*     Test Program for Random Number Generator */
/*             -Uniform     Random Number       */
/*             -Gaussian    Random Number       */
/*             -Exponential Random Number       */

#include  <stdio.h>
#include  <math.h>

main()
{
  int i;
  double RN(),RN_g(),RN_e();

  /* Arrays for histogram */
  double hpar_u[4]={0.,1.,0.02,51.};
  double hpar_g[4]={0.,1.,0.02,51.};
  double hpar_e[4]={0.,4.,0.08,51.};
  int harr_u[50],hsta_u[3];
  int harr_g[50],hsta_g[3];
  int harr_e[50],hsta_e[3];

  /*Initial Value for Random Number Generator */
  int ix=100;

  /* Initialize Buffer */
  for(i=0;i<50;i++){
    harr_u[i]=0;
    harr_g[i]=0;
    harr_e[i]=0;
  }
  for(i=0;i<3;i++){
    hsta_u[i]=0;
    hsta_g[i]=0;
    hsta_e[i]=0;
  }

  /* Calculation and Booking */
  for(i=0;i<10000;i++){
    hbook(RN(&ix)  ,hpar_u,harr_u,hsta_u,1);
    hbook(RN_g(&ix),hpar_g,harr_g,hsta_g,1);
    hbook(RN_e(&ix),hpar_e,harr_e,hsta_e,1);
  }

  /* Print Out */
```

```
    hbook(RN  (&ix),hpar_u,harr_u,hsta_u,0);
    hbook(RN_g(&ix),hpar_g,harr_g,hsta_g,0);
    hbook(RN_e(&ix),hpar_e,harr_e,hsta_e,0);
}


double RN_g(ix)
     int *ix;
{
  /* Gaussian Random Number Generator      */
  /*          RN_g=((RN(1)+...+RN(12))/12. */
  /*    *ix:Seed for Uniform Random Number */
  double sr;
  int i;
  double RN();
  sr=0.;
  for(i=0;i<12;i++)
    sr+=RN(ix);
  return sr/12.;
}
double RN_e(ix)
     int *ix;
{
  /* Exponential Random Number Generator   */
  /*          RN_e  = -ln(1.-RN)           */
  /*    *ix:Seed for Uniform Random Number */
  double RN();
  return -log(1.-RN(ix)+1.e-10);
}

double RN(ix)
int *ix;
{
  /* Uniform Random Number Generator */
  /*    *ix:Seed                     */
  *ix = 111*(*ix)+9;
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.5 研究課題:モンテカルロ法による円周率 π の計算

1.5.1 概要

一様乱数の確率的な性質を利用していろいろな数値計算やシミュレーションを行なうことができる。例えば

- 多重積分計算
- 微分方程式の解法
- 確率的現象のシミュレーション

などがある。

1.5.2 出力例

二重積分の一例として、円周率 π の計算をとり上げる。

```
*****Reduction of pi by Monte Carlo method*****
# of Trial=  2000,# of Entry= 1613./Pi= 3.22600
# of Trial=  4000,# of Entry= 3169./Pi= 3.16900
# of Trial=  6000,# of Entry= 4735./Pi= 3.15667
# of Trial=  8000,# of Entry= 6301./Pi= 3.15050
# of Trial= 10000,# of Entry= 7893./Pi= 3.15720
# of Trial= 12000,# of Entry= 9455./Pi= 3.15167
# of Trial= 14000,# of Entry=11018./Pi= 3.14800
# of Trial= 16000,# of Entry=12591./Pi= 3.14775
# of Trial= 18000,# of Entry=14167./Pi= 3.14822
# of Trial= 20000,# of Entry=15748./Pi= 3.14960
# of Trial= 22000,# of Entry=17319./Pi= 3.14891
# of Trial= 24000,# of Entry=18895./Pi= 3.14917
# of Trial= 26000,# of Entry=20481./Pi= 3.15092
# of Trial= 28000,# of Entry=22046./Pi= 3.14943
# of Trial= 30000,# of Entry=23614./Pi= 3.14853
# of Trial= 32000,# of Entry=25193./Pi= 3.14913
# of Trial= 34000,# of Entry=26738./Pi= 3.14565
# of Trial= 36000,# of Entry=28299./Pi= 3.14433
# of Trial= 38000,# of Entry=29874./Pi= 3.14463
# of Trial= 40000,# of Entry=31431./Pi= 3.14310
       <<Final Result>>
     ********************
         Pi= 3.14310
     ********************
```

円周率 π の計算

```
/*     Sample Program [Chapter 5, Study 1]          */
/*            Reduction of pi by Monte Calro method */

#include <stdio.h>
#include <math.h>

#define  NPNT   40000

main()
{
  double x,y,r,pi;
  int i;
  double RN();

  /* Initialize RN generator */
  int ix=100;

  /* Initialization */
  double cnt_in=0.;

  /* Calculation */
  printf("*****Reduction of pi by Monte Carlo method*****\n");
  for(i=0;i<NPNT;i++){
    x=RN(&ix)-0.5;
    y=RN(&ix)-0.5;
    r=sqrt(x*x+y*y);
    if(r<0.5)
      cnt_in+=1.;
    if(i%2000==0&&i!=0){
      pi=4.*cnt_in/(double)i;
      printf("# of Trial=%10d,# of Entry=%10.0f,Pi=%7.6f\n",i,cnt_in,pi);
    }
  }
  /* Final result */
  pi=4.*cnt_in/(double)NPNT;
  printf("       <<Final Result>>  \n");
  printf("     ********************\n");
  printf("         Pi=%f           \n",pi);
  printf("     ********************\n");
}

double RN(ix)
int *ix;
{
  *ix = 111*(*ix)+9;
```

```
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.6　研究課題:モンテカルロ法による酔歩のシミュレーション

1.6.1　概要

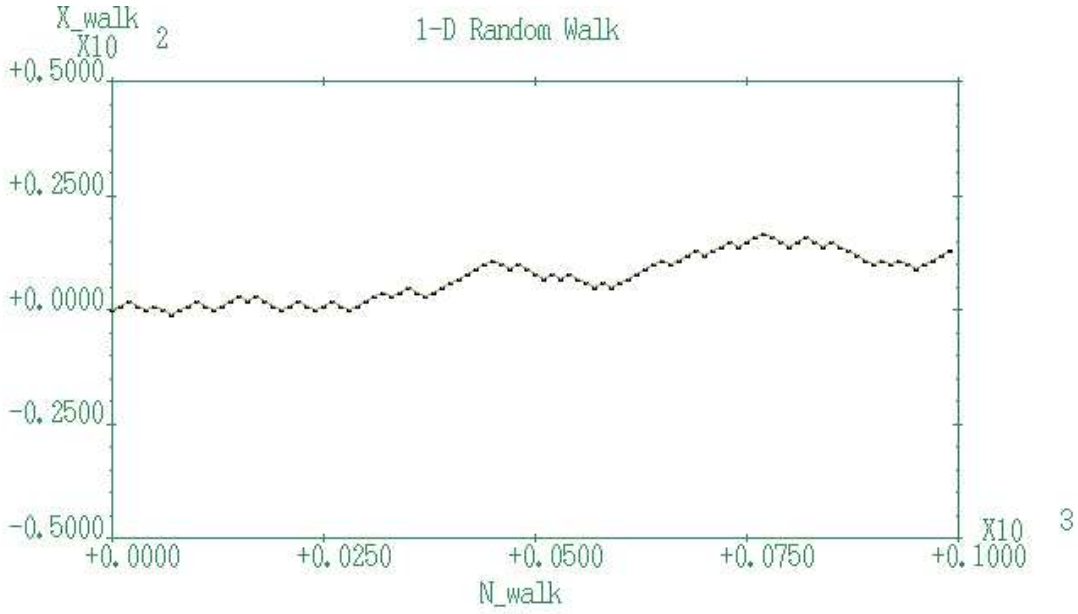
一様乱数を＜さいころ＞のように使うと酔歩のような選択肢が離散的（前／後）な場合のシミュレーションを行なうことができる。すなわち、

1. 前に進む確率を p（≤ 1）とする。
2. $rn < p$ ならば、前に一歩進む。
3. $rn \geq p$ ならば、後に一歩下がる。

として、シミュレーションを進めればよい。

1.6.2　出力例

一例として、p=0.5 の場合の計算をとり上げる。



酔歩のシミュレーション (p=0.5)

```
/*     Sample Program [Chapter 5, Study 2]     */
/*            One dimensional random walk      */
/*                  by Monte Calro method      */

#include <stdio.h>
#include <math.h>

#define  NPNT   100
#define  X0     0.
#define  DELTA  1.

main()
{
  double sgn,x_rnd;
  int i,ix;
  double RN();
  void grinit(),setcrs(),prtchr(),drwaxs(),drwgra(),grend();

  /* Parameters for graphics */
  double scale[2][3],x[101],y[101];
  char name[2][8] ={" N_walk "," X_walk "};
  double xbase=  0., dxstp=5., xdisp=25.;
  double ybase=-50., dystp=5., ydisp=15.;

  /* Initialize RN generator */
  ix=500;

  /* Initialization */
  grinit();

  /* Write title    */
  setcrs(250.,10.);
  prtchr("1-D Random Walk  ",17);

  /* Calculation */
  x_rnd=X0;
  x[0]=0.;
  y[0]=x_rnd;
  for(i=0;i<NPNT;i++){
    if(RN(&ix)<0.5)
      sgn=+1.;
    else
      sgn=-1.;
    x_rnd+=sgn*DELTA;
    x[i+1]=(double)i;
```

```
    y[i+1]=x_rnd;
  }

  /* Draw graph */
  scale[0][0]=xbase;
  scale[0][1]=dxstp;
  scale[0][2]=xdisp;
  scale[1][0]=ybase;
  scale[1][1]=dystp;
  scale[1][2]=ydisp;
  drwaxs(scale,name,1);
  drwgra(x,y,NPNT,scale,4);
  /* end grapgh */
  grend();
}

double RN(ix)
int *ix;
{
  *ix = 111*(*ix)+9;
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.7 研究課題:ベッセル方程式を解く

1.7.1 概要

物理学に現れるいろいろな常微分方程式は、変数を離散化し、それらを差分方程式に書き換えることによって、容易に数値解を求めることができる。解法としては、オイラー法、改良オイラー法およびルンゲークッタ法などがある。これらの方法を適用する場合、時間を離散化する幅と得られる数値解の収束性の間に強い相関があることを銘記すべきである。

1.7.2 出力例

一例として、$n = 0$ のベッセル方程式をとり上げ、初期条件

$$\begin{cases} y(0) & = & 1 \\ y'(0) & = & 0 \\ \lim_{x \to 0} \frac{y'(x)}{x} & = & \frac{1}{2} \end{cases} \tag{1.1}$$

のもとに数値解を求める。
ただし、ベッセルの方程式は２階常微分方程式なので

$$\begin{cases} y_1(x) & = & y(x) \\ y_2(x) & = & y'(x) \end{cases} \tag{1.2}$$

と置き換えて、連立１階常微分方程式に書き直す必要がある。それぞれの解法の収束性はどうだろうか？図書館にある関数表の数値と見比べてみよう。

```
 -Solution of bessel equation by Euler method
|      x        |     y         |     dy/dx     |
|-----------------------------------------------|
| 0.00          |   1.00000     |   0.00000     |
| 0.10          |   0.99775     |  -0.04995     |
| 0.20          |   0.99052     |  -0.09956     |
| 0.30          |   0.97836     |  -0.14845     |
| 0.40          |   0.96135     |  -0.19625     |
| 0.50          |   0.93962     |  -0.24262     |
| 0.60          |   0.91334     |  -0.28720     |
| 0.70          |   0.88269     |  -0.32966     |
| 0.80          |   0.84790     |  -0.36969     |
| 0.90          |   0.80923     |  -0.40700     |
| 1.00          |   0.76696     |  -0.44131     |
| 1.10          |   0.72140     |  -0.47237     |
| 1.20          |   0.67290     |  -0.49998     |
| 1.30          |   0.62179     |  -0.52393     |
| 1.40          |   0.56846     |  -0.54407     |
| 1.50          |   0.51329     |  -0.56026     |
| 1.60          |   0.45668     |  -0.57240     |
| 1.70          |   0.39905     |  -0.58044     |
| 1.80          |   0.34079     |  -0.58433     |
| 1.90          |   0.28234     |  -0.58409     |
| 2.00          |   0.22409     |  -0.57975     |
```

オイラー法による計算

```
 -Solution of bessel equation by I/Euler method
|      x        |     y         |     dy/dx     |
|-----------------------------------------------|
| 0.00          |   1.00000     |   0.00000     |
| 0.10          |   0.99750     |  -0.04994     |
| 0.20          |   0.99002     |  -0.09950     |
| 0.30          |   0.97763     |  -0.14832     |
| 0.40          |   0.96040     |  -0.19603     |
| 0.50          |   0.93847     |  -0.24227     |
| 0.60          |   0.91200     |  -0.28670     |
| 0.70          |   0.88120     |  -0.32900     |
| 0.80          |   0.84629     |  -0.36884     |
| 0.90          |   0.80752     |  -0.40595     |
| 1.00          |   0.76520     |  -0.44005     |
| 1.10          |   0.71962     |  -0.47090     |
| 1.20          |   0.67113     |  -0.49829     |
| 1.30          |   0.62008     |  -0.52202     |
| 1.40          |   0.56685     |  -0.54194     |
| 1.50          |   0.51182     |  -0.55793     |
| 1.60          |   0.45540     |  -0.56989     |
| 1.70          |   0.39798     |  -0.57776     |
| 1.80          |   0.33998     |  -0.58151     |
| 1.90          |   0.28181     |  -0.58115     |
| 2.00          |   0.22389     |  -0.57672     |
```

改良オイラー法による計算

```
 -Solution of bessel equation by R-Kutta method
|      x        |     y         |     dy/dx     |
|-----------------------------------------------|
| 0.00          |   1.00000     |   0.00000     |
| 0.10          |   0.99750     |  -0.04994     |
| 0.20          |   0.99002     |  -0.09950     |
| 0.30          |   0.97763     |  -0.14832     |
| 0.40          |   0.96040     |  -0.19603     |
| 0.50          |   0.93847     |  -0.24227     |
| 0.60          |   0.91200     |  -0.28670     |
| 0.70          |   0.88120     |  -0.32900     |
| 0.80          |   0.84629     |  -0.36884     |
| 0.90          |   0.80752     |  -0.40595     |
| 1.00          |   0.76520     |  -0.44005     |
| 1.10          |   0.71962     |  -0.47090     |
| 1.20          |   0.67113     |  -0.49829     |
| 1.30          |   0.62009     |  -0.52202     |
| 1.40          |   0.56686     |  -0.54195     |
| 1.50          |   0.51183     |  -0.55794     |
| 1.60          |   0.45540     |  -0.56990     |
| 1.70          |   0.39798     |  -0.57776     |
| 1.80          |   0.33999     |  -0.58152     |
| 1.90          |   0.28182     |  -0.58116     |
| 2.00          |   0.22389     |  -0.57672     |
```

ルンゲ-クッタ法による計算

```
/*     Sample Program [Chapter 6, Example 1]       */
/*            Solving Bessel Equation for n=0 case */
/*                    by   1.Euler method          */
/*                         2.Improved Euler method */
/*                         3.Runge-Kutta method    */

#include  <stdio.h>
#include  <math.h>

main()
{

  /* Variables */
  double sol_x,sol_y[2];
  double s1_dy[2],s2_dy[2],s3_dy[2],s4_dy[2];

  /* Initial condition for Bessel function(n=0) */
  double bes_0=1.,dbes_0=0.;

  /* Initial condition for the variable x */
  double x_0=0.;

  /* Parameters for iteration */
  double d_x=0.01;
  int n_rep=200;
  double xs,bes,besp;
  int i;

  /* Functions */
  double f_bes();
  void f_step();


  /* 1.Euler method */
  xs=x_0;
  bes =bes_0;
  besp=dbes_0;
  printf("-solution of Bessel equation by Euler method \n");
  printf("|      x          |     y          |     dy/dx     |\n");
  printf("|-------------------------------------------------|\n");
  printf("| %5.2f          |%10.5f     |%10.5f     |\n",xs,bes,besp);
  for(i=0;i<n_rep;i++){
    sol_x   =xs;
    sol_y[0]=bes;
    sol_y[1]=besp;
```

```
  f_step(d_x,sol_x,sol_y,s1_dy,f_bes);
  xs  =xs  +d_x;
  bes =bes +s1_dy[0];
  besp=besp+s1_dy[1];
  if((i+1)%10==0)
    printf("| %5.2f          |%10.5f     |%10.5f     |\n",xs,bes,besp);
}


/* 2.improved Euler method */
xs=x_0;
bes =bes_0;
besp=dbes_0;
printf("-solution of Bessel equation by Modified Euler method \n");
printf("|      x          |     y          |     dy/dx     |\n");
printf("|-------------------------------------------------|\n");
printf("| %5.2f          |%10.5f     |%10.5f     |\n",xs,bes,besp);
for(i=0;i<n_rep;i++){
  sol_x   =xs;
  sol_y[0]=bes;
  sol_y[1]=besp;
  f_step(d_x,sol_x,sol_y,s1_dy,f_bes);
  sol_x   =xs  +d_x;
  sol_y[0]=bes +s1_dy[0];
  sol_y[1]=besp+s1_dy[1];
  f_step(d_x,sol_x,sol_y,s2_dy,f_bes);
  xs  =xs  +d_x;
  bes =bes +(s1_dy[0]+s2_dy[0])/2.;
  besp=besp+(s1_dy[1]+s2_dy[1])/2.;
  if((i+1)%10==0)
    printf("| %5.2f          |%10.5f     |%10.5f     |\n",xs,bes,besp);
}


/* 3.Runge-Kutta method */
xs=x_0;
bes =bes_0;
besp=dbes_0;
printf("-solution of bessel equation by runge-kutta method \n");
printf("|      x          |     y          |     dy/dx     |\n");
printf("|-------------------------------------------------|\n");
printf("| %5.2f          |%10.5f     |%10.5f     |\n",xs,bes,besp);
for(i=0;i<n_rep;i++){
  sol_x   =xs;
  sol_y[0]=bes;
```

```
    sol_y[1]=besp;
    f_step(d_x,sol_x,sol_y,s1_dy,f_bes);
    sol_x   =xs  +0.5*d_x;
    sol_y[0]=bes +0.5*s1_dy[0];
    sol_y[1]=besp+0.5*s1_dy[1];
    f_step(d_x,sol_x,sol_y,s2_dy,f_bes);
    sol_x   =xs  +0.5*d_x;
    sol_y[0]=bes +0.5*s2_dy[0];
    sol_y[1]=besp+0.5*s2_dy[1];
    f_step(d_x,sol_x,sol_y,s3_dy,f_bes);
    sol_x   =xs  +    d_x;
    sol_y[0]=bes +    s3_dy[0];
    sol_y[1]=besp+    s3_dy[1];
    f_step(d_x,sol_x,sol_y,s4_dy,f_bes);
    xs  =xs  +d_x;
    bes =bes +(s1_dy[0]+2.*s2_dy[0]+2.*s3_dy[0]+s4_dy[0])/6.;
    besp=besp+(s1_dy[1]+2.*s2_dy[1]+2.*s3_dy[1]+s4_dy[1])/6.;
    if((i+1)%10==0)
      printf("| %5.2f          |%10.5f     |%10.5f     |\n",xs,bes,besp);
  }
}

double f_bes(x,b,bp,i)
double x,b,bp;
int i;
{
  /* righthand side of diff. equations */
  switch(i){
  case 0:
    return bp;
  case 1:
    if(x<1.e-10)
      return -(-0.5+b);
    else
      return -(bp/x+b);
  default:
    return 0.;
  }
}

void f_step(dx,x,y,dy,func)
double dx,x,y[],dy[];
double func();
{
  double b,bp;
```

```
  int i;
  b =y[0];
  bp=y[1];
  for(i=0;i<2;i++)
    dy[i]=dx*func(x,b,bp,i);
}
```

1.8 研究課題:人工衛星の軌道

1.8.1 概要

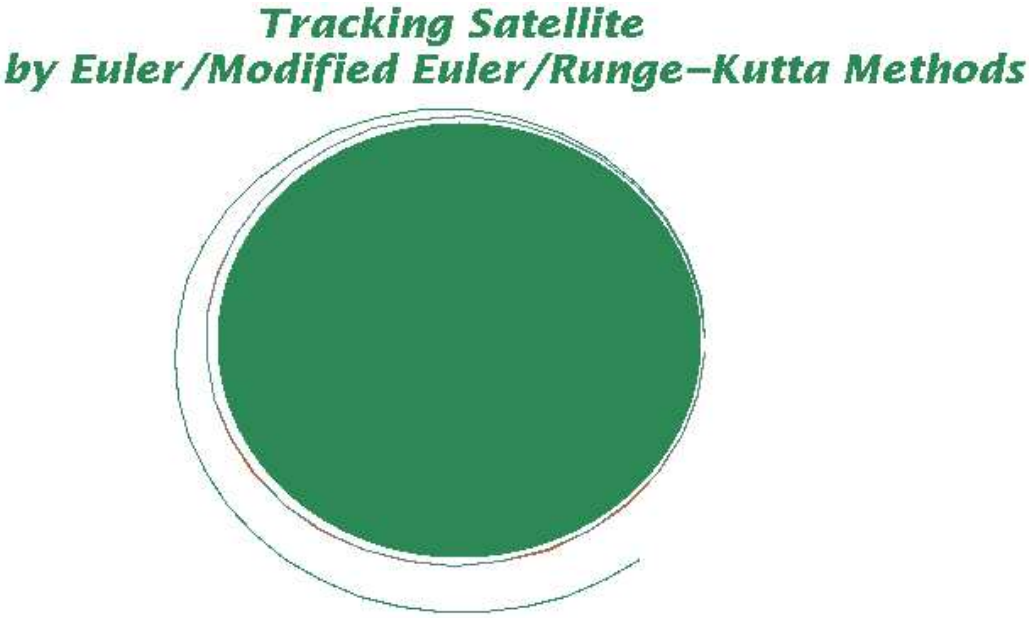
力学に現れるいろいろな常微分方程式も、基本的には前述のベッセル方程式の解法と同様にして、容易に数値解を求めることができる。具体的には、重心系で考えることにすると、$(x(t), y(t))$ についての連立２階常微分方程式を解くことに帰着する。テキストで学んだ微分方程式の数値解法を適用するには、更に

$$\begin{cases} y_1(t) &= x(t) \\ y_2(t) &= x'(t) \\ y_3(t) &= y(t) \\ y_4(t) &= y'(t) \end{cases} \tag{1.3}$$

と置き換えて、この方程式を連立１階常微分方程式に書き直す必要がある。

1.8.2 出力例

一例として、人工衛星（初期条件：高度 100km、速度 7.9km/s）の軌道の数値解をとり上げ、オイラー法、改良オイラー法およびルンゲ- クッタ法について、それぞれの解法の収束性を調べてみよう。



数値解の例；一番外側がオイラー法、内側２つが
改良オイラー法およびルンゲ-クッタ法

```
/*     Sample Program [Chapter 6, Study 1]         */
/*        finding an orbit of artificial satellite */
/*                      by   1.Euler method        */
/*                           2.Improved Euler method */
/*                           3.Runge-Kutta method  */
/*                           (in mks unit)         */

#include  <stdio.h>
#include  <math.h>

main()
{
  /* Variables*/
  double xs,ys,vx,vy,time,trc_t;
  double s_xs,s_ys,s_xn,s_yn;
  double trc_y[4];
  double t1_dy[4],t2_dy[4],t3_dy[4],t4_dy[4];
  int i;

  /* Parameters of the earth */
  double e_radi=6.38e6,e_xcen=0.,e_ycen=0.;

  /* Initial condition for the artificial satellite */
  double s_time=0.,s_x=6.48e6,s_y=0.,s_vx=0.,s_vy=7.9e3;

  /* Parameters for graphic window*/
  double v_x0=-12.e6,v_y0=-5.7e6,v_x1=12.e6,v_y1=7.7e6;

  /* Parameters for iteration */
  double d_time=16.;
  int n_rep=332;

  /* Functions */
  double f_gra();
  void f_step();
  void grinit(),setcrs(),prtchr(),paintf();
  void v_circ(),v_line();

  /* Initialization */
  grinit();

  /* Define virtual graphic window */
  v_wind(v_x0,v_y0,v_x1,v_y1);

  /* Write title */
```

```
setcrs(250.,10.);
prtchr("Tracking Satellite",18);
setcrs(150.,40.);
prtchr("by Euler/Modified Euler/Runge-Kutta methods",43);

/* Draw the earth */
paintf(1);
v_circ(e_xcen,e_ycen,e_radi,1,0.,6.29);
paintf(0);

/* 1.Euler method */
v_circ(s_x,s_y,0.2,6,0.,0.);
time=s_time;
xs  =s_x-e_xcen;
ys  =s_y-e_ycen;
vx  =s_vx;
vy  =s_vy;
s_xs=s_x;
s_ys=s_y;
for(i=0;i<n_rep;i++){
  trc_t   =time;
  trc_y[0]=xs;
  trc_y[1]=ys;
  trc_y[2]=vx;
  trc_y[3]=vy;
  f_step(d_time,trc_t,trc_y,t1_dy,f_gra);
  time=time+d_time;
  xs  =xs+t1_dy[0];
  ys  =ys+t1_dy[1];
  vx  =vx+t1_dy[2];
  vy  =vy+t1_dy[3];
  if((i+1)%10==0){
    s_xn=xs+e_xcen;
    s_yn=ys+e_ycen;
    v_line(s_xs,s_ys,s_xn,s_yn,6);
    v_circ(s_xn,s_yn,0.2,6,0.,0.);
    s_xs=s_xn;
    s_ys=s_yn;
  }
}

  /* 2.improved euler method */
  time=s_time;
  xs  =s_x-e_xcen;
  ys  =s_y-e_ycen;
```

```
    vx  =s_vx;
    vy  =s_vy;
    s_xs=s_x;
    s_ys=s_y;
    for(i=0;i<n_rep;i++){
      trc_t   =time;
      trc_y[0]=xs;
      trc_y[1]=ys;
      trc_y[2]=vx;
      trc_y[3]=vy;
      f_step(d_time,trc_t,trc_y,t1_dy,f_gra);
      trc_t   =time+d_time;
      trc_y[0]=xs  +t1_dy[0];
      trc_y[1]=ys  +t1_dy[1];
      trc_y[2]=vx  +t1_dy[2];
      trc_y[3]=vy  +t1_dy[3];
      f_step(d_time,trc_t,trc_y,t2_dy,f_gra);
      trc_t   =time+d_time;
      xs=xs  +(t1_dy[0]+t2_dy[0])/2.;
      ys=ys  +(t1_dy[1]+t2_dy[1])/2.;
      vx=vx  +(t1_dy[2]+t2_dy[2])/2.;
      vy=vy  +(t1_dy[3]+t2_dy[3])/2.;
      if((i+1)%10==0){
s_xn=xs+e_xcen;
s_yn=ys+e_ycen;
v_line(s_xs,s_ys,s_xn,s_yn,5);
v_circ(s_xn,s_yn,0.2,5,0.,0.);
s_xs=s_xn;
s_ys=s_yn;
      }
    }


    /* 3.Runge-Kutta method */
    time=s_time;
    xs  =s_x-e_xcen;
    ys  =s_y-e_ycen;
    vx  =s_vx;
    vy  =s_vy;
    s_xs=s_x;
    s_ys=s_y;
    for(i=0;i<n_rep;i++){
      trc_t   =time;
      trc_y[0]=xs;
      trc_y[1]=ys;
```

```
      trc_y[2]=vx;
      trc_y[3]=vy;
      f_step(d_time,trc_t,trc_y,t1_dy,f_gra);
      trc_t   =time+0.5*d_time;
      trc_y[0]=xs  +0.5*t1_dy[0];
      trc_y[1]=ys  +0.5*t1_dy[1];
      trc_y[2]=vx  +0.5*t1_dy[2];
      trc_y[3]=vy  +0.5*t1_dy[3];
      f_step(d_time,trc_t,trc_y,t2_dy,f_gra);
      trc_t   =time+0.5*d_time;
      trc_y[0]=xs  +0.5*t2_dy[0];
      trc_y[1]=ys  +0.5*t2_dy[1];
      trc_y[2]=vx  +0.5*t2_dy[2];
      trc_y[3]=vy  +0.5*t2_dy[3];
      f_step(d_time,time,trc_y,t3_dy,f_gra);
      trc_t   =time+    d_time;
      trc_y[0]=xs  +    t3_dy[0];
      trc_y[1]=ys  +    t3_dy[1];
      trc_y[2]=vx  +    t3_dy[2];
      trc_y[3]=vy  +    t3_dy[3];
      f_step(d_time,trc_t,trc_y,t4_dy,f_gra);
      time=time+d_time;
      xs  =xs+(t1_dy[0]+2.*t2_dy[0]+2.*t3_dy[0]+t4_dy[0])/6.;
      ys  =ys+(t1_dy[1]+2.*t2_dy[1]+2.*t3_dy[1]+t4_dy[1])/6.;
      vx  =vx+(t1_dy[2]+2.*t2_dy[2]+2.*t3_dy[2]+t4_dy[2])/6.;
      vy  =vy+(t1_dy[3]+2.*t2_dy[3]+2.*t3_dy[3]+t4_dy[3])/6.;
      if((i+1)%10==0){
s_xn=xs+e_xcen;
s_yn=ys+e_ycen;
v_line(s_xs,s_ys,s_xn,s_yn,4);
v_circ(s_xn,s_yn,0.2,4,0.,0.);
s_xs=s_xn;
s_ys=s_yn;
      }
    }

    /* End graph */
    grend();
}

double f_gra(time,x,y,vx,vy,i)
     double time,x,y,vx,vy;
     int i;
{
  /* righthand side of diff. equations    */
```

```
  /* -in present case, time independent!! */
  double e_mass=5.97e24,c_gra=6.67e-11;
  double r;
  switch(i){
  case 0:
    return vx;
  case 1:
    return vy;
  case 2:
    r    =sqrt(x*x+y*y);
    return -c_gra*e_mass*x/(r*r*r);
  case 3:
    r    =sqrt(x*x+y*y);
    return -c_gra*e_mass*y/(r*r*r);
  default:
    return 0.;
  }
}

void f_step(dtime,time,y,dy,func)
     double dtime,time,y[],dy[];
     double func();
{
  double xp,yp,vx,vy;
  int i;
  xp=y[0];
  yp=y[1];
  vx=y[2];
  vy=y[3];
  for(i=0;i<4;i++)
    dy[i]=dtime*func(time,xp,yp,vx,vy,i);
}
```

1.9 研究課題:ボイジャーと木星 -重力加速-

1.9.1 概要

力学に現れる問題で必ずしも自明でない問題の例として、惑星探査機ボイジャーの木星による重力加速の機構について検討してみよう。

原理的には、この問題は人工衛星の軌道計算と全くパラレルな問題で、初期条件と参照座標系が違うだけなのであるが、われわれの理解の困難さは相当程度に高くなってくる。実際には、いろいろな初期条件やさまざまな参照座標系をとって比較検討することによって、加速機構のの本質が次第にはっきりと見えてくるだろう。

1.9.2 出力例

一例として、初期条件

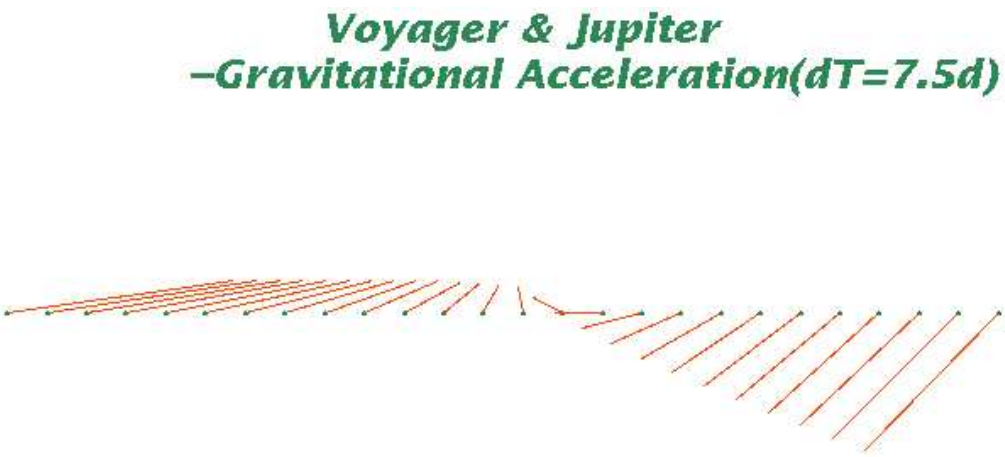
ボイジャー

$$\begin{cases} x(0) &= 50 \times 10^9 m \\ y(0) &= 8 \times 10^9 m \\ x'(0) &= 8 \times 10^3 m \\ y'(0) &= 0m \end{cases} \tag{1.4}$$

木星

$$\begin{cases} X(0) &= 0m \\ Y(0) &= 0m \\ X'(0) &= 13.5 \times 10^3 m \\ Y'(0) &= 0m \end{cases} \tag{1.5}$$

のもとに、Runge-Kutta 法を使って、軌道の数値解を求め、いろいろな参照座標系で描いてみよう。



木星の重力加速を受けるボイジャー

```
/* Sample Program [Chapter 6, Study 2]             */
/*  Examining gravitational acceleration of Voyger */
/*                    by   Runge-Kutta method      */
/*                         (in MKS unit)           */

#include  <stdio.h>
#include  <math.h>

main()
{
  /* Variables*/
  double xs,ys,vx,vy,time,trc_t;
  double s_xs,s_ys,s_xn,s_yn;
  double xz,yz;
  double trc_y[4];
  double t1_dy[4],t2_dy[4],t3_dy[4],t4_dy[4];
  int i;

  /* Parameters of the Jupiter */
  double z_radi=71.6e6;
  double z_x=0.,z_y=0.;
  double z_vx=13.5e3,z_vy=0.;

  /* Initial condition for the Voyager */
  double v_radi=0.2;
  double s_time=0.,s_x=50000.e6,s_y=8000.e6,s_vx=8.e3,s_vy=0.;

  /* Parameters for graphic window*/
  double v_x0=-1000.e6,v_y0=-56000.e6,v_x1=210000.e6,v_y1=56000.e6;

  /* Parameters for iteration */
  double d_time=300.;
  int n_rep=55000;

  /* Functions */
  double f_gra();
  void f_step();
  void grinit(),setcrs(),prtchr(),paintf();
  void v_circ(),v_line();

  /* Initialization */
  grinit();

  /* Define virtual graphic window */
  v_wind(v_x0,v_y0,v_x1,v_y1);
```

```
/* Write title */
setcrs(250.,10.);
prtchr("Voyager & Jupiter",17);
setcrs(170.,40.);
prtchr("-Gravitational Acceleration(dT=7.5d)",36);

/* Artificial Enhancement */
v_radi *=5.e9;
z_radi *=30.;

/* Tracking Voyager by Runge-Kutta method */
v_line(s_x,s_y,z_x,z_y,2);
paintf(1);
v_circ(s_x,s_y,v_radi,4,0.,6.29);
v_circ(z_x,z_y,z_radi,1,0.,6.29);
paintf(0);
time=s_time;
xs  =s_x;
ys  =s_y;
vx  =s_vx;
vy  =s_vy;
xz  =z_x;
yz  =z_y;
for(i=0;i<n_rep;i++){
  trc_t   =time;
  trc_y[0]=xs;
  trc_y[1]=ys;
  trc_y[2]=vx;
  trc_y[3]=vy;
  f_step(d_time,trc_t,trc_y,t1_dy,f_gra);
  trc_t   =time+0.5*d_time;
  trc_y[0]=xs  +0.5*t1_dy[0];
  trc_y[1]=ys  +0.5*t1_dy[1];
  trc_y[2]=vx  +0.5*t1_dy[2];
  trc_y[3]=vy  +0.5*t1_dy[3];
  f_step(d_time,trc_t,trc_y,t2_dy,f_gra);
  trc_t   =time+0.5*d_time;
  trc_y[0]=xs  +0.5*t2_dy[0];
  trc_y[1]=ys  +0.5*t2_dy[1];
  trc_y[2]=vx  +0.5*t2_dy[2];
  trc_y[3]=vy  +0.5*t2_dy[3];
  f_step(d_time,time,trc_y,t3_dy,f_gra);
  trc_t   =time+    d_time;
  trc_y[0]=xs  +    t3_dy[0];
```

```
    trc_y[1]=ys  +    t3_dy[1];
    trc_y[2]=vx  +    t3_dy[2];
    trc_y[3]=vy  +    t3_dy[3];
    f_step(d_time,trc_t,trc_y,t4_dy,f_gra);
    time=time+d_time;
    xs  =xs+(t1_dy[0]+2.*t2_dy[0]+2.*t3_dy[0]+t4_dy[0])/6.;
    ys  =ys+(t1_dy[1]+2.*t2_dy[1]+2.*t3_dy[1]+t4_dy[1])/6.;
    vx  =vx+(t1_dy[2]+2.*t2_dy[2]+2.*t3_dy[2]+t4_dy[2])/6.;
    vy  =vy+(t1_dy[3]+2.*t2_dy[3]+2.*t3_dy[3]+t4_dy[3])/6.;
    xz  =xz+d_time*z_vx;
    yz  =yz+d_time*z_vy;
    if((i+1)%2160==0){
      v_line(xs,ys,xz,yz,2);
      paintf(1);
      v_circ(xs,ys,v_radi,4,0.,6.29);
      v_circ(xz,yz,z_radi,1,0.,6.29);
      paintf(0);
    }
  }

  /* End graph */
  grend();
}

double f_gra(time,x,y,vx,vy,i)
     double time,x,y,vx,vy;
     int i;
{
  /* righthand side of diff. equations    */
  /* -in present case, time independent!! */
  double c_gra=6.67e-11;
  double z_mass=1.90e27;
  double z_x=0.,z_y=0.;
  double z_vx=13.2e3,z_vy=0.;
  double xz,yz,r;
  xz=z_x+z_vx*time;
  yz=z_y+z_vy*time;
  switch(i){
  case 0:
    return vx;
  case 1:
    return vy;
  case 2:
    r    =sqrt((x-xz)*(x-xz)+(y-yz)*(y-yz));
    return -c_gra*z_mass*(x-xz)/(r*r*r);
```

```
  case 3:
    r    =sqrt((x-xz)*(x-xz)+(y-yz)*(y-yz));
    return -c_gra*z_mass*(y-yz)/(r*r*r);
  default:
    return 0.;
  }
}

void f_step(dtime,time,y,dy,func)
     double dtime,time,y[],dy[];
     double func();
{
  double xp,yp,vx,vy;
  int i;
  xp=y[0];
  yp=y[1];
  vx=y[2];
  vy=y[3];
  for(i=0;i<4;i++)
    dy[i]=dtime*func(time,xp,yp,vx,vy,i);
}
```

1.10　研究課題:３体問題 -天体のカオス-

1.10.1　概要

古くから取り上げられている力学上の問題で、現在でもよく分かっていない問題の１つに、３体問題がある。このような問題に取り組むことにより、力学の持つ別の側面、デモーニッシュな奥深さに気付かされるだろう。

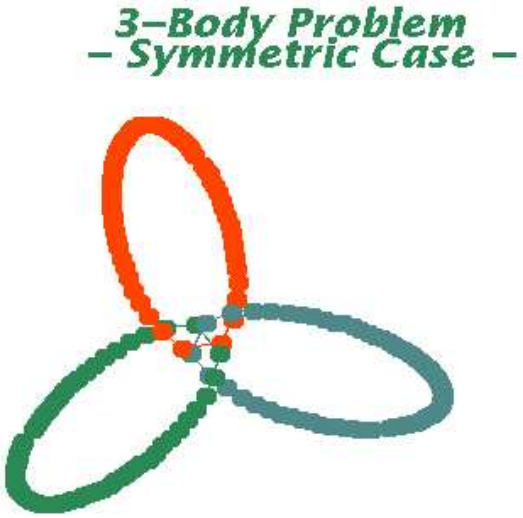
３体問題は、重心運動の自由度と、エネルギー保存則および角運動量保存則による制約を除いて、総計８個の自由度を持つ。その結果、人工衛星やボイジャーの場合（自由度は２個）と比べて、大変に複雑になる。一般的に、自由度が３以上の非線形力学系ではカオスが出現するといわれているが、３体問題でもカオスとの遭遇は避けられない。

1.10.2　出力例

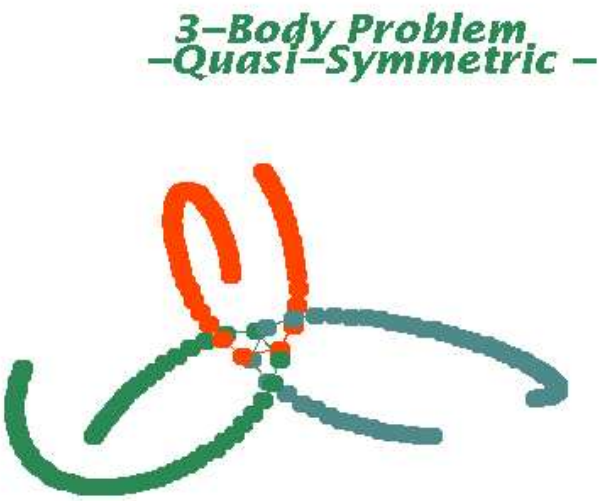
一例として、対称的な初期条件を基準にして、そこからわずかに離れた初期条件をとった場合の軌道の安定性を調べてみよう。具体的には、

1. 対称的な初期条件
   - 同一質量（地球相当）の３物体
   - 互いの距離 20 万 km の正三角形配置
   - 各辺に沿って時計周り方向に速度 1km/s で運動
2. わずかだけ 1 の配置からずれた初期条件
   （天体２が重心方向へ 200km（0.1%）移動）
3. さらに 1 の配置からずれた初期条件
   （天体２が重心方向へ 2000km（1%）移動）

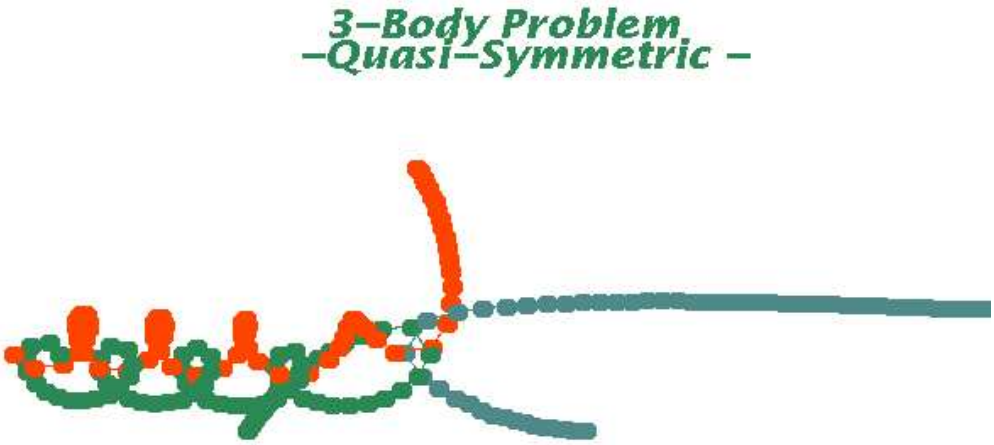
について、Runge-Kutta 法を使って、軌道の数値解を求め、結果を図に描いてみよう。



対称的な初期条件（左から物体１、２、３）



わずかにずれた初期条件



さらにずれた初期条件

```
/*     Sample Program [Chapter 6, Study 3]         */
/*            Challenging 3-body problem           */
/*                     by   Runge-Kutta method     */
/*                          (in MKS unit)          */

#include  <stdio.h>
#include  <math.h>

main()
{
  /* Variables*/
  double trc_y[4][3];
  double t1_dy[4][3],t2_dy[4][3],t3_dy[4][3],t4_dy[4][3];
  double xs[3],ys[3],vx[3],vy[3],time,trc_t;
  double s_xs[3],s_ys[3];
  int i,j;

  /* Initial condition for the Objects */
  /*-Approximately symmetric condition-*/
  double s_x[3] ={   -1.e8,      0., 1.e8};
  double s_y[3] ={      0., 1.732e8,   0.};
  double s_vx[3]={   0.5e3,   0.5e3,-1.e3};
  double s_vy[3]={ 0.866e3,-0.866e3,   0.};
  double s_time =0.;
  double s_radi[3]={6.38e6,6.38e6,6.38e6};
  int    is_col[3]={1,2,4};

  /* Parameters for graphic window*/
  double v_x0=-2.5e8,v_y0=-0.5e8,v_x1=2.5e8,v_y1=2.5e8;
  /* Parameters for iteration */
  double d_time=60.;
  int n_rep=4500;

  /* Functions */
  double f_gra();
  void f_step();
  void grinit(),setcrs(),prtchr(),paintf();
  void v_circ(),v_line();

  /* Initialization */
  grinit();

  /* Define virtual graphic window */
  v_wind(v_x0,v_y0,v_x1,v_y1);
```

```
/* Write title */
setcrs(250.,30.);
prtchr("  3-Body Problem",18);
setcrs(250.,50.);
prtchr("- Symmetric Case -",18);

/* Tracking the objects by Runge-Kutta method */
for(i=0;i<3;i++){
  paintf(1);
  v_circ(s_x[i],s_y[i],s_radi[i],is_col[i],0.,6.29);
  paintf(0);
}
time=s_time;
for(i=0;i<3;i++){
       xs[i]  =s_x[i];
       ys[i]  =s_y[i];
       vx[i]  =s_vx[i];
       vy[i]  =s_vy[i];
       s_xs[i]=s_x[i];
       s_ys[i]=s_y[i];
}
for(i=0;i<n_rep;i++){
  trc_t   =time;
  for(j=0;j<3;j++){
    trc_y[0][j]=xs[j];
    trc_y[1][j]=ys[j];
    trc_y[2][j]=vx[j];
    trc_y[3][j]=vy[j];
  }
  f_step(d_time,trc_t,trc_y,t1_dy,f_gra);
  trc_t   =time+0.5*d_time;
  for(j=0;j<3;j++){
    trc_y[0][j]=xs[j]  +0.5*t1_dy[0][j];
    trc_y[1][j]=ys[j]  +0.5*t1_dy[1][j];
    trc_y[2][j]=vx[j]  +0.5*t1_dy[2][j];
    trc_y[3][j]=vy[j]  +0.5*t1_dy[3][j];
  }
  f_step(d_time,trc_t,trc_y,t2_dy,f_gra);
  trc_t   =time+0.5*d_time;
  for(j=0;j<3;j++){
    trc_y[0][j]=xs[j]  +0.5*t2_dy[0][j];
    trc_y[1][j]=ys[j]  +0.5*t2_dy[1][j];
    trc_y[2][j]=vx[j]  +0.5*t2_dy[2][j];
    trc_y[3][j]=vy[j]  +0.5*t2_dy[3][j];
  }
```

```
    f_step(d_time,time,trc_y,t3_dy,f_gra);
    trc_t   =time+    d_time;
    for(j=0;j<3;j++){
      trc_y[0][j]=xs[j]  +    t3_dy[0][j];
      trc_y[1][j]=ys[j]  +    t3_dy[1][j];
      trc_y[2][j]=vx[j]  +    t3_dy[2][j];
      trc_y[3][j]=vy[j]  +    t3_dy[3][j];
    }
    f_step(d_time,trc_t,trc_y,t4_dy,f_gra);
    time+=d_time;
    for(j=0;j<3;j++){
      xs[j]+=(t1_dy[0][j]+2.*t2_dy[0][j]+2.*t3_dy[0][j]+t4_dy[0][j])/6.;
      ys[j]+=(t1_dy[1][j]+2.*t2_dy[1][j]+2.*t3_dy[1][j]+t4_dy[1][j])/6.;
      vx[j]+=(t1_dy[2][j]+2.*t2_dy[2][j]+2.*t3_dy[2][j]+t4_dy[2][j])/6.;
      vy[j]+=(t1_dy[3][j]+2.*t2_dy[3][j]+2.*t3_dy[3][j]+t4_dy[3][j])/6.;
    }
    if((i+1)%60==0){
      for(j=0;j<3;j++){
paintf(1);
v_circ(xs[j],ys[j],s_radi[j],is_col[j],0.,6.29);
paintf(0);
      }
    }
  }

  /* End graph */
  grend();
}

double f_gra(time,x,y,vx,vy,i,j)
     double time,x[3],y[3],vx[3],vy[3];
     int i,j;
{
  double r,f_tot;
  int k;

  /* Parameters on the objects */
  double c_gra=6.67e-11;
  double s_mass[3]={5.97e24,5.97e24,5.97e24};

  switch(i){
  case 0:
    return vx[j];
  case 1:
    return vy[j];
```

```
  case 2:
    f_tot=0.;
    for(k=0;k<3;k++){
      if(k!=j){
r=sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k]));
f_tot+=-c_gra*s_mass[k]*(x[j]-x[k])/(r*r*r);
      }
    }
    return f_tot;
  case 3:
    f_tot=0.;
    for(k=0;k<3;k++){
      if(k!=j){
r=sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k]));
f_tot+=-c_gra*s_mass[k]*(y[j]-y[k])/(r*r*r);
      }
    }
    return f_tot;
  default:
    return 0.;
  }
}
void f_step(dtime,time,y,dy,func)
     double dtime,time,y[4][3],dy[4][3];
     double func();
{
  double xp[3],yp[3],vx[3],vy[3];
  int i,j;

  for(j=0;j<3;j++){
    xp[j]=y[0][j];
    yp[j]=y[1][j];
    vx[j]=y[2][j];
    vy[j]=y[3][j];
  }
  for(i=0;i<4;i++){
    for(j=0;j<3;j++){
      dy[i][j]=dtime*func(time,xp,yp,vx,vy,i,j);
    }
  }
}
```

1.11　研究課題:ボルツマン分布則 -緩和による平衡状態の実現-

1.11.1　概要

古典的な分子運動論でよく知られているボルツマン分布則は、統計力学に現れるカノニカル分布の一例として解釈されているが、ここでは全く別の視点からこの問題にアプローチしてみよう。すなわち、極端な非平衡状態から出発し、衝突の繰返しによって分子集団の速度分布が漸近的にボルツマン分布則になることを、シミュレーションによって確かめてみる。なぜそのようにして得られた「平衡状態」がボルツマン分布則を実現しているのかは、必ずしも自明ではない。実際、個別の衝突は位相空間内に均一な粒子の再配置を起こす訳ではない。予想できることは、多数回の衝突によってある種の＜確率的な意味での固定点＞（衝突によって変わらない不変分布）に到達しているということである。

1.11.2　出力例

一例として、分子を剛体球として考えて分子同士の衝突の運動学を仮定したときの、分子の速度分布の緩和をシミュレーションし、「平衡状態」の速度分布をヒストグラムで見てみよう。具体的には、２０００個の酸素分子 O_2 について、各々運動エネルギーが $\frac{3}{2}k_BT$（一定）でランダムな運動方向を持っているという初期状態を仮定し、分子同士の衝突を３００００回起こした後の状態における速度分布を調べる。ただし、$T = 300K$ とする。

```
 VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                        0       100       200       300       400       500
  0.0000                +----+----+----+----+----+----+----+----+----+----+
  0.0787         11     +*   |    |    |    |    |    |    |    |    |    +
  0.1575         64     +    |*   |    |    |    |    |    |    |    |    +
  0.2362        169     +    |    |    | *  |    |    |    |    |    |    +
  0.3150        302     +    |    |    |    |    |    *    |    |    |    +
  0.3937        304     +----|----|----|----|----|----*----|----|----|----+
  0.4724        324     +    |    |    |    |    |    |  * |    |    |    +
  0.5512        283     +    |    |    |    |    |   * |    |    |    |    +
  0.6299        201     +    |    |    |    *    |    |    |    |    |    +
  0.7087        133     +    |    |  * |    |    |    |    |    |    |    +
  0.7874         87     +----|---*|----|----|----|----|----|----|----|----+
  0.8661         54     +    *    |    |    |    |    |    |    |    |    +
  0.9449         36     +   *|    |    |    |    |    |    |    |    |    +
  1.0236         18     + *  |    |    |    |    |    |    |    |    |    +
  1.1024          8     +*   |    |    |    |    |    |    |    |    |    +
  1.1811          2     *----|----|----|----|----|----|----|----|----|----+
  1.2598          1     *    |    |    |    |    |    |    |    |    |    +
  1.3386          1     *    |    |    |    |    |    |    |    |    |    +
  1.4173          1     *    |    |    |    |    |    |    |    |    |    +
  1.4961          1     *    |    |    |    |    |    |    |    |    |    +
  1.5748          0     *----|----|----|----|----|----|----|----|----|----+
  1.6536          0     *    |    |    |    |    |    |    |    |    |    +
  1.7323          0     *    |    |    |    |    |    |    |    |    |    +
  1.8110          0     *    |    |    |    |    |    |    |    |    |    +
  1.8898          0     *    |    |    |    |    |    |    |    |    |    +
  1.9685          0     *----|----|----|----|----|----|----|----|----|----+
  2.0473                +----+----+----+----+----+----+----+----+----+----+
   *  0.1000e+06
   <NUMBER OF ENTRIES> TOT:    2000.
                |           |
   UNDER RANGE  | IN RANGE  | OVER RANGE
             0. |    2000.  |       0.
```

$|\vec{v}|$ の分布

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                      0       100       200       300       400       500
 0.0000               +----+----+----+----+----+----+----+----+----+----+
 0.0455        280    +    |    |    |    |    |  * |    |    |    |    +
 0.0909        263    +    |    |    |    |    |*   |    |    |    |    +
 0.1364        261    +    |    |    |    |    |*   |    |    |    |    +
 0.1818        233    +    |    |    |    |  * |    |    |    |    |    +
 0.2273        196    +----|----|----|----*----|----|----|----|----|----+
 0.2728        184    +    |    |    |  * |    |    |    |    |    |    +
 0.3182        132    +    |    |  * |    |    |    |    |    |    |    +
 0.3637        109    +    |    |*   |    |    |    |    |    |    |    +
 0.4091        102    +    |    *    |    |    |    |    |    |    |    +
 0.4546         85    +----|---*|----|----|----|----|----|----|----|----+
 0.5001         52    +    *    |    |    |    |    |    |    |    |    +
 0.5455         35    +   *|    |    |    |    |    |    |    |    |    +
 0.5910         22    + *  |    |    |    |    |    |    |    |    |    +
 0.6365         17    + *  |    |    |    |    |    |    |    |    |    +
 0.6819         13    +*---|----|----|----|----|----|----|----|----|----+
 0.7274          5    +*   |    |    |    |    |    |    |    |    |    +
 0.7728          7    +*   |    |    |    |    |    |    |    |    |    +
 0.8183          0    *    |    |    |    |    |    |    |    |    |    +
 0.8638          2    *    |    |    |    |    |    |    |    |    |    +
 0.9092          1    *----|----|----|----|----|----|----|----|----|----+
 0.9547          0    *    |    |    |    |    |    |    |    |    |    +
 1.0001          0    *    |    |    |    |    |    |    |    |    |    +
 1.0456          0    *    |    |    |    |    |    |    |    |    |    +
 1.0911          0    *    |    |    |    |    |    |    |    |    |    +
 1.1365          1    *----|----|----|----|----|----|----|----|----|----+
 1.1820               +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+06
  <NUMBER OF ENTRIES> TOT:    2000.
             |          |
  UNDER RANGE | IN RANGE  | OVER RANGE
           0. |    2000.  |       0.
```

v_x の分布

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                      0       100       200       300       400       500
 0.0000               +----+----+----+----+----+----+----+----+----+----+
 0.0455        257    +    |    |    |    |    |*   |    |    |    |    +
 0.0909        253    +    |    |    |    |    *    |    |    |    |    +
 0.1364        265    +    |    |    |    |    | *  |    |    |    |    +
 0.1818        230    +    |    |    |    |  * |    |    |    |    |    +
 0.2273        209    +----|----|----|----|*---|----|----|----|----|----+
 0.2728        198    +    |    |    |    *    |    |    |    |    |    +
 0.3182        143    +    |    |   *|    |    |    |    |    |    |    +
 0.3637        120    +    |    | *  |    |    |    |    |    |    |    +
 0.4091         92    +    |   *|    |    |    |    |    |    |    |    +
 0.4546         76    +----|--*-|----|----|----|----|----|----|----|----+
 0.5001         45    +    *    |    |    |    |    |    |    |    |    +
 0.5455         26    +  * |    |    |    |    |    |    |    |    |    +
 0.5910         25    +  * |    |    |    |    |    |    |    |    |    +
 0.6365         25    +  * |    |    |    |    |    |    |    |    |    +
 0.6819         14    +*---|----|----|----|----|----|----|----|----|----+
 0.7274          4    *    |    |    |    |    |    |    |    |    |    +
 0.7728          7    +*   |    |    |    |    |    |    |    |    |    +
 0.8183          6    +*   |    |    |    |    |    |    |    |    |    +
 0.8638          3    *    |    |    |    |    |    |    |    |    |    +
 0.9092          0    *----|----|----|----|----|----|----|----|----|----+
 0.9547          2    *    |    |    |    |    |    |    |    |    |    +
 1.0001          0    *    |    |    |    |    |    |    |    |    |    +
 1.0456          0    *    |    |    |    |    |    |    |    |    |    +
 1.0911          0    *    |    |    |    |    |    |    |    |    |    +
 1.1365          0    *----|----|----|----|----|----|----|----|----|----+
 1.1820               +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+06
  <NUMBER OF ENTRIES> TOT:    2000.
             |          |
  UNDER RANGE | IN RANGE  | OVER RANGE
           0. |    2000.  |       0.
```

v_y の分布

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                        0        50       100       150       200       250
 0.0000                 +----+----+----+----+----+----+----+----+----+----+
 0.0455        214      +    |    |    |    |    |    |    |    |  *  |    +
 0.0909        246      +    |    |    |    |    |    |    |    |     |   *+
 0.1364        209      +    |    |    |    |    |    |    |    | *   |    +
 0.1818        213      +    |    |    |    |    |    |    |    |  *  |    +
 0.2273        184      +----|----|----|----|----|----|----|-*--|----|----+
 0.2728        176      +    |    |    |    |    |    |    *    |    |    +
 0.3182        147      +    |    |    |    |    |    *|   |    |    |    +
 0.3637        123      +    |    |    |    |    *    |    |    |    |    +
 0.4091         96      +    |    |    |   *|    |    |    |    |    |    +
 0.4546         87      +----|----|----|-*--|----|----|----|----|----|----+
 0.5001         65      +    |    |  * |    |    |    |    |    |    |    +
 0.5455         46      +    |   *|    |    |    |    |    |    |    |    +
 0.5910         56      +    |    |*   |    |    |    |    |    |    |    +
 0.6365         31      +    |*   |    |    |    |    |    |    |    |    +
 0.6819         24      +----*----|----|----|----|----|----|----|----|----+
 0.7274         22      +   *|    |    |    |    |    |    |    |    |    +
 0.7728         18      +   *|    |    |    |    |    |    |    |    |    +
 0.8183         12      + *  |    |    |    |    |    |    |    |    |    +
 0.8638          3      +*   |    |    |    |    |    |    |    |    |    +
 0.9092         12      +-*--|----|----|----|----|----|----|----|----|----+
 0.9547          8      + *  |    |    |    |    |    |    |    |    |    +
 1.0001          4      +*   |    |    |    |    |    |    |    |    |    +
 1.0456          1      *    |    |    |    |    |    |    |    |    |    +
 1.0911          2      *    |    |    |    |    |    |    |    |    |    +
 1.1365          1      *----|----|----|----|----|----|----|----|----|----+
 1.1820                 +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+06
  <NUMBER OF ENTRIES> TOT:    2000.
             |          |
  UNDER RANGE | IN RANGE | OVER RANGE
           0. |    2000. |       0.
```

v_z の分布

```
/*     Sample Program [Chapter 7, Study 1]      */
/*          Reduction of Boltzmann distribution */
/*                   by a relaxation model      */

#include  <stdio.h>
#include  <math.h>

#define G_MASS     32.  /* Mass number of molecule: O2 */
#define G_TEMP     300. /* Temperature: 300K           */
#define C_BOLT     1.38E-16 /* Boltzmann Constant      */
#define C_AVOG     6.24E23  /* Avogadoro Number        */
#define PI         3.1416   /* Pi                      */

main()
{
  int i,ic1,ic2;
  double sinth,costh,phai;
  double vcmx,vcmy,vcmz;
  double vrlx,vrly,vrlz;
  double va,vrl;
  double RN();

  /* Arrays for histograms */
  double hpa_va[4]={0.,0.,0.,25.};
  double hpa_vx[4]={0.,0.,0.,25.};
  double hpa_vy[4]={0.,0.,0.,25.};
  double hpa_vz[4]={0.,0.,0.,25.};
  int har_va[51],hst_va[3];
  int har_vx[51],hst_vx[3];
  int har_vy[51],hst_vy[3];
  int har_vz[51],hst_vz[3];

  /* Seed for RN generator */
  int ix=100;

  /* Tracks in velocity space */
  double vx[2000],vy[2000],vz[2000];

  /* Calculate constants in use */
  /* -Mass of molecule          */
  double p_mass=G_MASS/C_AVOG;
  /* -Initial velocity of molecule */
  double c_kt=C_BOLT*G_TEMP;
  double p_vel2=3.*c_kt/p_mass;
  double p_vel =sqrt(p_vel2);
```

```
  /* Initialize Buffer */
  for(i=0;i<50;i++){
    har_va[i]=0;
    har_vx[i]=0;
    har_vy[i]=0;
    har_vz[i]=0;
  }
  for(i=0;i<3;i++){
    hst_va[i]=0;
    hst_vx[i]=0;
    hst_vy[i]=0;
    hst_vz[i]=0;
  }


  /* Initialization */
  /* -Velocities    */
  for(i=0;i<2000;i++){
    costh=1.-2.*RN(&ix);
    sinth=sqrt(1.-costh*costh);
    phai =2.*PI*RN(&ix);
    vx[i]=p_vel*sinth*cos(phai);
    vy[i]=p_vel*sinth*sin(phai);
    vz[i]=p_vel*costh;
}

  /* -Parameters for histogram */
      hpa_va[1]=4.*p_vel;
      hpa_va[2]=hpa_va[1]/hpa_va[3];
      hpa_vx[1]=4.*p_vel/sqrt(3.);
      hpa_vx[2]=hpa_vx[1]/hpa_vx[3];
      hpa_vy[1]=4.*p_vel/sqrt(3.);
      hpa_vy[2]=hpa_vy[1]/hpa_vy[3];
      hpa_vz[1]=4.*p_vel/sqrt(3.);
      hpa_vz[2]=hpa_vz[1]/hpa_vz[3];

      /* Tracking the molecules in collisions */
      for(i=0;i<30000;i++){
/* -Select two molecules involved in collision */
            ic1=(int)(2000.*RN(&ix));
            ic2=(int)(2000.*RN(&ix));
            if(ic1!=ic2){
      /* -Calculate 2-body kinematics */
      /* ..Velocity of CMS            */
```

```
              vcmx=0.5*(vx[ic1]+vx[ic2]);
              vcmy=0.5*(vy[ic1]+vy[ic2]);
              vcmz=0.5*(vz[ic1]+vz[ic2]);
      /* ..Velocity in CMS            */
              vrlx=0.5*(vx[ic1]-vx[ic2]);
              vrly=0.5*(vy[ic1]-vy[ic2]);
              vrlz=0.5*(vz[ic1]-vz[ic2]);
      /* ..Moving direction after collision */
              costh=1.-2.*RN(&ix);
              sinth=sqrt(1.-costh*costh);
              phai =2.*PI*RN(&ix);
      /* ..Velocity after collision */
      vrl =sqrt(vrlx*vrlx+vrly*vrly+vrlz*vrlz);
              vrlx=vrl*sinth*cos(phai);
              vrly=vrl*sinth*sin(phai);
              vrlz=vrl*costh;
              vx[ic1]= vrlx+vcmx;
              vy[ic1]= vrly+vcmy;
              vz[ic1]= vrlz+vcmz;
              vx[ic2]=-vrlx+vcmx;
              vy[ic2]=-vrly+vcmy;
              vz[ic2]=-vrlz+vcmz;
    }
      }

      /* Histograming */
      for(i=0;i<2000;i++){
va=sqrt(vx[i]*vx[i]+vy[i]*vy[i]+vz[i]*vz[i]);
hbook(    va     ,hpa_va,har_va,hst_va,1);
        hbook(fabs(vx[i]),hpa_vx,har_vx,hst_vx,1);
        hbook(fabs(vy[i]),hpa_vy,har_vy,hst_vy,1);
        hbook(fabs(vz[i]),hpa_vz,har_vz,hst_vz,1);
      }

      /* Print out */
hbook(    va     ,hpa_va,har_va,hst_va,0);
        hbook(fabs(vx[i]),hpa_vx,har_vx,hst_vx,0);
        hbook(fabs(vy[i]),hpa_vy,har_vy,hst_vy,0);
        hbook(fabs(vz[i]),hpa_vz,har_vz,hst_vz,0);
}

double RN(ix)
int *ix;
{
  *ix = 111*(*ix)+9;
```

```
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.12 研究課題: 3 次元酔歩 -確率過程-

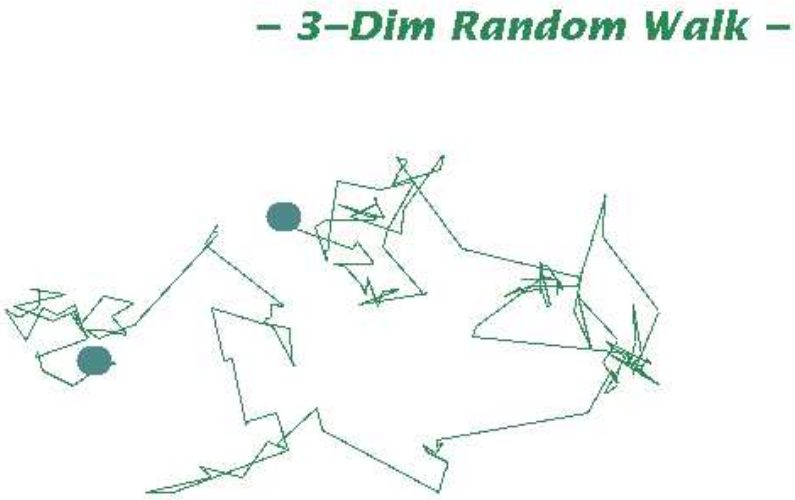
1.12.1 概要

ここでは、現実的な問題として、気体の中での電子の熱運動を 3 次元酔歩モデルによって捉えてみよう。気体分子との衝突によって電子の運動エネルギーは変わらずにその運動方向だけが変わると考えると、「一定時間毎にある距離（平均自由行程）だけ進んで運動方向を変える」という 3 次元酔歩とパラレルになっていることに気付かされる。ここでは、もう少しだけ現実的に考えて、平均自由行程の代わりに指数分布に従う自由行程を採用してシミュレーションを行なう。

1.12.2 出力例

一例として、気体分子と電子をともに剛体球とみなし、次のような仮定のもとにシミュレーションを行なってみよう。

- 気体分子の速度は、電子の速度に較べて十分に小さく、無視できる。
- 平均自由行程は電子の速度に依らず一定（$\lambda = 6cm$）である。
- 気体分子との衝突の後の電子の速度の大きさは変わらない。
- 気体分子と衝突の後の電子の運動方向は等方分布に従う。



3 次元酔歩のシミュレーション

```
/*     Sample Program [Chapter 7, Study 2a]     */
/*            3-D random walk                   */
/*                  by Monte Calro method       */

#include <stdio.h>
#include <math.h>

#define  X0     0.      /* Initial Positoin        */
#define  Y0     0.
#define  Z0     0.
#define  P_MEAN 6.      /* Mean Free Path          */
#define  NCOLL  100     /* Number of Collisions    */
#define  NPART  2       /* Number of Particles     */
#define  NINTR  1       /* Intervals to be Skipped */
#define  PI     3.14159 /* Pi                      */

main()
{
  double x,y,z,dx,dy,dz;
  double xs,ys;
  double flight;
  double sinth,costh,phai;
  int i,j,ix;

  /* Parameters for graphic window*/
  double v_x0=-70.,v_y0=-42.,v_x1=70.,v_y1=42.;

  /* Functions */
  double RN_e(),RN();
  void grinit(),grend(),setcrs(),prtchr();
  void v_wind(),v_circ(),v_line(),paintf();

  /*  Seed for RN generator */
  ix=100;

  /* Initialization */
  grinit();
  /* Define virtual graphic window */
  v_wind(v_x0,v_y0,v_x1,v_y1);

  /* Write title */
  setcrs(150.,30.);
  prtchr("- 3-Dim Random Walk -",21);

  /*  Monte Carlo simulation */
```

```
  v_line(-3., 0., 3., 0.,2);
  v_line( 0.,-3., 0., 3.,2);
  for(i=0;i<NPART;i++){
    /* -Initial position */
    x=X0;
    y=Y0;
    z=Z0;
    /* -Initial direction cosine */
    costh=1.-2.*RN(&ix);
    sinth=sqrt(1.-costh*costh);
    phai =2.*PI*RN(&ix);
    dx   =sinth*cos(phai);
    dy   =sinth*sin(phai);
    dz   =costh;
    /* -Initial position for graphics */
    /*         (x-y projection)       */
    xs=X0;
    ys=Y0;
    for(j=0;j<NCOLL;j++){
      /* -Flight length */
      flight=P_MEAN*RN_e(&ix);
      x += flight*dx;
      y += flight*dy;
      z += flight*dz;
      if(j%NINTR==0){
v_line(xs,ys,x,y,1);
xs=x;
ys=y;
      }
      /* -Change direction */
      costh=1.-2.*RN(&ix);
      sinth=sqrt(1.-costh*costh);
      phai =2.*PI*RN(&ix);
      dx   =sinth*cos(phai);
      dy   =sinth*sin(phai);
      dz   =costh;
    }
    paintf(1);
    v_circ(xs,ys,3.,4,0.,6.29);
    paintf(0);
  }

  /* End graph */
  grend();
}
```

```
double RN_e(ix)
     int *ix;
{
  /* Exponential Random Number Generator   */
  /*          RN_e  = -ln(1.-RN)           */
  /*    *ix:Seed for Uniform Random Number */
  double RN();
  return -log(1.-RN(ix)+1.e-10);
}

double RN(ix)
int *ix;
{
  /* Uniform Random Number Generator */
  /*    *ix:Seed                     */
  *ix = 111*(*ix)+9;
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.13 研究課題: 3 次元酔歩 -拡散現象-

1.13.1 概要

前項の確率過程を電子の集団に適用すると、拡散現象が現れる。

1.13.2 出力例

一例として、次のような仮定のもとにシミュレーションを行なってみよう。

- 時刻 $t = 0$ で 300 個の原子が座標原点にある。
- 気体分子の速度は、電子の速度に較べて十分に小さく、無視できる。
- 平均自由行程は電子の速度に依らず一定（$\lambda = 6cm$）である。
- 気体分子との衝突の後の電子の速度の大きさは変わらない。
- 気体分子と衝突の後の電子の運動方向は等方分布に従う。

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                       0         10        20        30        40        50
-2.0000                +----+----+----+----+----+----+----+----+----+----+
-1.8400          0     *    |    |    |    |    |    |    |    |    |    +
-1.6800          0     *    |    |    |    |    |    |    |    |    |    +
-1.5200          0     *    |    |    |    |    |    |    |    |    |    +
-1.3600          1     +*   |    |    |    |    |    |    |    |    |    +
-1.2000          3     +--*-|----|----|----|----|----|----|----|----|----+
-1.0400          3     +  * |    |    |    |    |    |    |    |    |    +
-0.8800          6     +    |*   |    |    |    |    |    |    |    |    +
-0.7200          7     +    | *  |    |    |    |    |    |    |    |    +
-0.5600         14     +    |    |   *|    |    |    |    |    |    |    +
-0.4000         25     +----|----|----|----|----*----|----|----|----|----+
-0.2400         32     +    |    |    |    |    |    | *  |    |    |    +
-0.0800         48     +    |    |    |    |    |    |    |    |    |  * +
 0.0800         37     +    |    |    |    |    |    |    | *  |    |    +
 0.2400         27     +    |    |    |    |    |    | *  |    |    |    +
 0.4000         27     +----|----|----|----|----|-*--|----|----|----|----+
 0.5600         22     +    |    |    |    |    | *  |    |    |    |    +
 0.7200         22     +    |    |    |    |    | *  |    |    |    |    +
 0.8800         11     +    |    |*   |    |    |    |    |    |    |    +
 1.0400          9     +    |   *|    |    |    |    |    |    |    |    +
 1.2000          4     +---*|----|----|----|----|----|----|----|----|----+
 1.3600          2     + *  |    |    |    |    |    |    |    |    |    +
 1.5200          0     *    |    |    |    |    |    |    |    |    |    +
 1.6800          0     *    |    |    |    |    |    |    |    |    |    +
 1.8400          0     *    |    |    |    |    |    |    |    |    |    +
 2.0000          0     *----|----|----|----|----|----|----|----|----|----+
 2.1600                +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+03
  <NUMBER OF ENTRIES> TOT:     300.
                |          |
  UNDER RANGE | IN RANGE  | OVER RANGE
           0. |     300.  |       0.
```

x の分布

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                             0         10        20        30        40        50
-2.0000                      +----+----+----+----+----+----+----+----+----+----+
-1.8400          0           *    |    |    |    |    |    |    |    |    |    +
-1.6800          0           *    |    |    |    |    |    |    |    |    |    +
-1.5200          0           *    |    |    |    |    |    |    |    |    |    +
-1.3600          1           +*   |    |    |    |    |    |    |    |    |    +
-1.2000          1           +*---|----|----|----|----|----|----|----|----|----+
-1.0400          4           +   *|    |    |    |    |    |    |    |    |    +
-0.8800          1           +*   |    |    |    |    |    |    |    |    |    +
-0.7200         17           +    |    |    | *  |    |    |    |    |    |    +
-0.5600         17           +    |    |    | *  |    |    |    |    |    |    +
-0.4000         13           +----|----|--*-|----|----|----|----|----|----|----+
-0.2400         29           +    |    |    |    |    |   *|    |    |    |    +
-0.0800         44           +    |    |    |    |    |    |    |    |   *|    +
 0.0800         36           +    |    |    |    |    |    |    |*   |    |    +
 0.2400         34           +    |    |    |    |    |    |   *|    |    |    +
 0.4000         40           +----|----|----|----|----|----|----|----|----*----|----+
 0.5600         33           +    |    |    |    |    |    |  * |    |    |    +
 0.7200         13           +    |    |  * |    |    |    |    |    |    |    +
 0.8800          6           +    |*   |    |    |    |    |    |    |    |    +
 1.0400         10           +    |    *    |    |    |    |    |    |    |    +
 1.2000          1           +*---|----|----|----|----|----|----|----|----|----+
 1.3600          0           *    |    |    |    |    |    |    |    |    |    +
 1.5200          0           *    |    |    |    |    |    |    |    |    |    +
 1.6800          0           *    |    |    |    |    |    |    |    |    |    +
 1.8400          0           *    |    |    |    |    |    |    |    |    |    +
 2.0000          0           *----|----|----|----|----|----|----|----|----|----+
 2.1600                      +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+03
  <NUMBER OF ENTRIES> TOT:      300.
              |           |
  UNDER RANGE | IN RANGE  | OVER RANGE
           0. |      300. |        0.
```

y の分布

```
VARIABLE   CONTENTS          GRAPH OF DISTRIBUTION
                             0         10        20        30        40        50
-2.0000                      +----+----+----+----+----+----+----+----+----+----+
-1.8400          0           *    |    |    |    |    |    |    |    |    |    +
-1.6800          0           *    |    |    |    |    |    |    |    |    |    +
-1.5200          0           *    |    |    |    |    |    |    |    |    |    +
-1.3600          1           +*   |    |    |    |    |    |    |    |    |    +
-1.2000          1           +*---|----|----|----|----|----|----|----|----|----+
-1.0400          1           +*   |    |    |    |    |    |    |    |    |    +
-0.8800          5           +    *    |    |    |    |    |    |    |    |    +
-0.7200         10           +    |    *    |    |    |    |    |    |    |    +
-0.5600         19           +    |    |    |   *|    |    |    |    |    |    +
-0.4000         25           +----|----|----|----|----*----|----|----|----|----+
-0.2400         29           +    |    |    |    |    |   *|    |    |    |    +
-0.0800         37           +    |    |    |    |    |    |    | *  |    |    +
 0.0800         46           +    |    |    |    |    |    |    |    |    |*   +
 0.2400         36           +    |    |    |    |    |    |    |*   |    |    +
 0.4000         28           +----|----|----|----|----|--*-|----|----|----|----+
 0.5600         23           +    |    |    |    |    |  * |    |    |    |    +
 0.7200         21           +    |    |    |    |*   |    |    |    |    |    +
 0.8800         10           +    |    *    |    |    |    |    |    |    |    +
 1.0400          5           +    *    |    |    |    |    |    |    |    |    +
 1.2000          3           +--*-|----|----|----|----|----|----|----|----|----+
 1.3600          0           *    |    |    |    |    |    |    |    |    |    +
 1.5200          0           *    |    |    |    |    |    |    |    |    |    +
 1.6800          0           *    |    |    |    |    |    |    |    |    |    +
 1.8400          0           *    |    |    |    |    |    |    |    |    |    +
 2.0000          0           *----|----|----|----|----|----|----|----|----|----+
 2.1600                      +----+----+----+----+----+----+----+----+----+----+
  *  0.1000e+03
  <NUMBER OF ENTRIES> TOT:      300.
              |           |
  UNDER RANGE | IN RANGE  | OVER RANGE
           0. |      300. |        0.
```

z の分布

```
/*     Sample Program [Chapter 7, Study 2b]     */
/*            3-D random walk                   */
/*                  by Monte Calro method       */

#include <stdio.h>
#include <math.h>

#define  X0     0.      /* Initial Positoin        */
#define  Y0     0.
#define  Z0     0.
#define  P_MEAN 6.      /* Mean Free Path          */
#define  NCOLL  100     /* Number of Collisions    */
#define  NPART  200     /* Number of Particles     */
#define  NINTR  1       /* Intervals to be Skipped */
#define  PI     3.14159 /* Pi                      */

main()
{
  double x,y,z,dx,dy,dz;
  double xs,ys;
  double flight;
  double sinth,costh,phai;
  int i,j,ix;

  /* Functions */
  double RN_e(),RN();
  void hbook();

  /* Arrays for histograms */
  double hpa_px[4]={-200.,200.,16.,25.};
  double hpa_py[4]={-200.,200.,16.,25.};
  double hpa_pz[4]={-200.,200.,16.,25.};
  int har_px[26],hst_px[3];
  int har_py[26],hst_py[3];
  int har_pz[26],hst_pz[3];

  /* Initialize Buffer */
  for(i=0;i<25;i++){
    har_px[i]=0;
    har_py[i]=0;
    har_pz[i]=0;

  }
  for(i=0;i<3;i++){
    hst_px[i]=0;
```

```
    hst_py[i]=0;
    hst_pz[i]=0;

  }

  /* Seed for RN generator */
  ix=100;

  /*  Monte Carlo simulation */
  for(i=0;i<NPART;i++){
    /* -Initial position */
    x=X0;
    y=Y0;
    z=Z0;
    /* -Initial direction cosine */
    costh=1.-2.*RN(&ix);
    sinth=sqrt(1.-costh*costh);
    phai =2.*PI*RN(&ix);
    dx   =sinth*cos(phai);
    dy   =sinth*sin(phai);
    dz   =costh;
    for(j=0;j<NCOLL;j++){
      /* -Flight length */
      flight=P_MEAN*RN_e(&ix);
      x += flight*dx;
      y += flight*dy;
      z += flight*dz;
      /* -Change direction */
      costh=1.-2.*RN(&ix);
      sinth=sqrt(1.-costh*costh);
      phai =2.*PI*RN(&ix);
      dx   =sinth*cos(phai);
      dy   =sinth*sin(phai);
      dz   =costh;
    }
    hbook(x,hpa_px,har_px,hst_px,1);
    hbook(y,hpa_py,har_py,hst_py,1);
    hbook(z,hpa_pz,har_pz,hst_pz,1);
  }

  /* Print out */
  hbook(x,hpa_px,har_px,hst_px,0);
  hbook(y,hpa_py,har_py,hst_py,0);
  hbook(z,hpa_pz,har_pz,hst_pz,0);
}
```

```
double RN_e(ix)
     int *ix;
{
  /* Exponential Random Number Generator   */
  /*          RN_e  = -ln(1.-RN)           */
  /*    *ix:Seed for Uniform Random Number */
  double RN();
  return -log(1.-RN(ix)+1.e-10);
}

double RN(ix)
int *ix;
{
  /* Uniform Random Number Generator */
  /*    *ix:Seed                     */
  *ix = 111*(*ix)+9;
  *ix = *ix-1000000*(*ix/1000000);
  return (double)(*ix)/999999.;
}
```

1.14　研究課題:気体中のブラウン粒子の運動 -座標空間像-

1.14.1　概要

ブラウン運動は、ミクロな系とマクロな系との確率的な相互作用の一つの典型である。ここでは、例えば、大気中に浮遊する煙の粒子の運動に着目して、その性質を調べてみよう。特に、運動の軌跡の持つフラクタル性やいろいろな次元量（ハウスドルフ次元、情報次元など）を見てみるのも面白いだろう。

1.14.2　出力例

一例として、1 気圧の酸素ガス中における炭素微片のブラウン運動を、次のような仮定のもとにシミュレーションしてみよう。

- 炭素微片は剛体球とみなす。
  （半径 $4.4 \times 10^{-7} cm$、重さ $6.0 \times 10^{-19} g$）
- 平均衝突時間 dt 内に 1 回衝突。
- 衝突は dt 内の任意時刻にランダムに起こる。
- $t = 0$ で座標原点にあった炭素微片に対して、$t = 10^5 dt$ までを追跡。

気体中のブラウン粒子

```
/* Sample Program [Chapter 7, Study 3] */
/*            3-D Brownian motion               */
/*                  by Monte Calro method       */
/*                  (in CGS unit)               */

#include <stdio.h>
#include <math.h>

#define  X0     0.      /* Initial Positoin        */
#define  Y0     0.
#define  Z0     0.
#define  VX0    0.      /* Initial Velocity        */
#define  VY0    0.
#define  VZ0    0.
                         /* Param. for simulation   */
#define  N_COLL 120000   /* Number of collisions    */
#define  N_PART 1        /* Number of particles     */
#define  N_INTR 1000      /* Interval of observation */
                         /* Parameters for cluster  */
#define  P_MASS 6.e-19   /* Mass:6.e-19g            */
#define  P_RADI 4.6e-7   /* Radius:46A              */
                         /* Parameters for gas      */
#define  G_MNUM 32.      /* Mass number:oxigen      */
#define  G_TEMP 300.     /* Temperature:300K        */
#define  G_PRES 1.       /* Pressure:1atm           */
                         /* Universal Constants     */
#define  PI     3.14159  /* Pi                      */
#define  C_BOLT 1.38e-16 /* Boltzmann Constant      */
#define  C_AVOG 6.03e23  /* Avogadoro Constant      */

main()
{
  double x,y,z,dx,dy,dz;
  double vx,vy,vz,v,dv,vmin,vmax;
  double vxm,vym,vzm,vcmx,vcmy,vcmz,vrlx,vrly,vrlz,vrl;
  double time,dt,dt_bef;
  double sum,area,fnorm,factor,prob,t_scal;
  double chk,fv;
  double xs,ys;
  double tk,g_mass,g_dens;
  double sinth,costh,phai;
  int i,j,k,ks;

  /* Parameters for graphic window*/
  double v_x0=-0.2e-4,v_y0=-1.2e-5,v_x1= 0.5e-4,v_y1=3.0e-5;
```

```
/* Functions */
double RN();
void grinit(),grend(),setcrs(),prtchr();
void v_wind(),v_circ(),v_line(),paintf();

/*  Seed for RN generators */
int ix=0,ix1=23450,ix2=34560,ix3=45670,ix4=56780;

/* Probability function for collision */
double p_coll[101];

/* Initialization */
grinit();

/* -Virtual graphic window */
v_wind(v_x0,v_y0,v_x1,v_y1);

/* -particle density in gas */
g_dens=C_AVOG*G_PRES/22.4e3;

/* -kBT */
tk=C_BOLT*G_TEMP;

/* Mass of molecule */
g_mass=G_MNUM/C_AVOG;

/*  -Velocity Range */
vmin=0.;
vmax=4.*sqrt(tk/g_mass);

/* -Integration */
dv  =(vmax-vmin)/100.;

/*  Reduction of probability function for collision */
sum   =0.;
area  =PI*P_RADI*P_RADI;
fnorm =sqrt(g_mass/(2.*PI*tk));
factor=area*g_dens*fnorm;
for(i=0;i<101;i++){
  v   =vmin+dv *(double)i;
  prob=factor*v*exp(-0.5*g_mass*v*v/tk)*dv;
  sum += prob;
  p_coll[i]=sum;
}
```

```
t_scal=sum;
/*  Scale probability */
for(i=0;i<101;i++){
  p_coll[i]=p_coll[i]/t_scal;
}

/* Get time slice:mean collision time */
dt=1./t_scal;

/* Write title */
setcrs(150.,10.);
prtchr("3-Dimensional Brownian Motion",30);
setcrs(150.,40.);
prtchr("- Monte Carlo Simulation -",27);

/* Monte Carlo Simulation */
/* -Reference marker around origin */
/* ..x-scale                       */
v_line(-3.3e-6,    0., 3.3e-6,    0.,0);
v_line(-3.3e-6,-3.3e-7,-3.3e-6,+3.3e-7,0);
v_line( 3.3e-6,-3.3e-7, 3.3e-6,+3.3e-7,0);
/* ..y-scale                       */
v_line(    0.,-3.3e-6,    0., 3.3e-6,0);
v_line(-3.3e-7,-3.3e-6, 3.3e-7,-3.3e-6,0);
v_line(-3.3e-7, 3.3e-6, 3.3e-7, 3.3e-6,0);

for(i=0;i<N_PART;i++){
  /* -Reset time */
  time=0.;

  /* -Initial position of particle(i) */
  x=X0;
  y=Y0;
  z=Z0;
  vx=VX0;
  vy=VY0;
  vz=VZ0;

  /* ..x-y projection */
  xs=X0;
  ys=Y0;
  for(j=0;j<N_COLL;j++){
    /* -Step DT */
    time += dt;
```

```
      /* -Before collision */
      /* ..Time interval */
            dt_bef=dt*RN(&ix);
    /* ..Drift */
            x += vx*dt_bef;
            y += vy*dt_bef;
            z += vz*dt_bef;

    /* ..Velocity of molecule */
    chk = RN(&ix);
    for(k=1;k<101;k++){
      if(chk<p_coll[k]&&chk>p_coll[k-1])
ks=k-1;
    }
            fv=dv*(chk-p_coll[ks])/(p_coll[ks+1]-p_coll[ks]);
            v =dv*(double)ks+fv;
            costh=1.-2.*RN(&ix1);
            sinth=sqrt(1.-costh*costh);
            phai =2.*PI*RN(&ix2);
            vxm  =v*sinth*cos(phai);
            vym  =v*sinth*sin(phai);
            vzm  =v*costh;

    /* -Kinematics */
    /* ..Velocity of CMS */
            vcmx=(P_MASS*vx+g_mass*vxm)/(P_MASS+g_mass);
            vcmy=(P_MASS*vy+g_mass*vym)/(P_MASS+g_mass);
            vcmz=(P_MASS*vz+g_mass*vzm)/(P_MASS+g_mass);

    /* ..Velocity in CMS */
            vrlx=vx-vcmx;
            vrly=vy-vcmy;
            vrlz=vz-vcmz;
            vrl =sqrt(vrlx*vrlx+vrly*vrly+vrlz*vrlz);

    /* ..Moving direction after collision */
            costh=1.-2.*RN(&ix3);
            sinth=sqrt(1.-costh*costh);
            phai =2.*PI*RN(&ix4);

    /* -After collision */
    /* ..Velocity */
            vrlx=vrl*sinth*cos(phai);
            vrly=vrl*sinth*sin(phai);
            vrlz=vrl*costh;
```

```
            vx  = vrlx+vcmx;
            vy  = vrly+vcmy;
            vz  = vrlz+vcmz;

    /* ..Drift */
            x += vx*(dt-dt_bef);
            y += vy*(dt-dt_bef);
            z += vz*(dt-dt_bef);
            if(j%N_INTR==0){
               v_line(xs,ys,x,y,1);
               xs=x;
               ys=y;
    }
    }
    paintf(1);
    v_circ(xs,ys,P_RADI,2,0.,6.29);
    paintf(0);
  }

  /* End graph */
  grend();
}

double RN(ix)
int *ix;
{
  /* Uniform Random Number Generator */
  /*    *ix:Seed                     */
  *ix = 111*(*ix)+9;
  *ix = *ix-10000000*(*ix/10000000);
  return (double)(*ix)/9999999.;
}
```

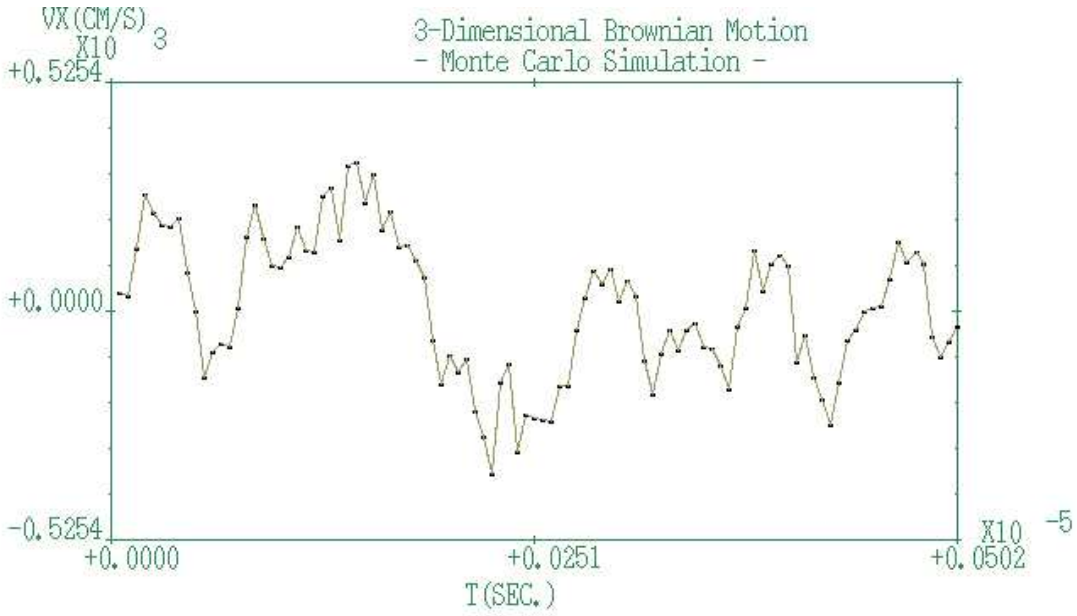
1.15　研究課題:気体中のブラウン粒子 -速度空間像-

1.15.1　概要

ブラウン粒子の運動を速度空間での運動として捉えてみよう。ここでも、座標空間での運動と全く同様に、酔歩のような軌跡を示していることが見て取れるだろう。つまり、ブラウン粒子は６次元の位相空間内を酔歩のように歩んでいるということなのである。

1.15.2　出力例

一例として、１気圧の酸素ガス中における炭素微片のブラウン運動を、前項と同じ仮定のもとにシミュレーションしてみよう。



速度空間におけるブラウン粒子の運動

```
/* Sample Program [Chapter 7, Study 3] */
/*            3-D Brownian motion               */
/*                  by Monte Calro method       */
/*                  (in CGS unit)               */

#include <stdio.h>
#include <math.h>

#define  X0     0.      /* Initial Positoin        */
#define  Y0     0.
#define  Z0     0.
#define  VX0    0.      /* Initial Velocity        */
#define  VY0    0.
#define  VZ0    0.
                         /* Param. for simulation   */
#define  N_COLL 100000   /* Number of collisions    */
#define  N_INTR 1000      /* Interval of observation */
                         /* Parameters for cluster  */
#define  P_MASS 6.e-19   /* Mass:6.e-19g            */
#define  P_RADI 4.6e-7   /* Radius:46A              */
                         /* Parameters for gas      */
#define  G_MNUM 32.      /* Mass number:oxigen      */
#define  G_TEMP 300.     /* Temperature:300K        */
#define  G_PRES 1.       /* Pressure:1atm           */
                         /* Universal Constants     */
#define  PI     3.14159  /* Pi                      */
#define  C_BOLT 1.38e-16 /* Boltzmann Constant      */
#define  C_AVOG 6.03e23  /* Avogadoro Constant      */

main()
{
  double x,y,z,dx,dy,dz;
  double vx,vy,vz,v,dv,vmin,vmax;
  double vxm,vym,vzm,vcmx,vcmy,vcmz,vrlx,vrly,vrlz,vrl;
  double time,dt,dt_bef;
  double sum,area,fnorm,factor,prob,t_scal;
  double chk,fv;
  double xs,ys;
  double tk,g_mass,g_dens;
  double sinth,costh,phai;
  double scale[2][3],xmon[100],ymon[100];
  char name[2][8]={"t(sec)  ","vx(cm/s)"};
  double p_vmax;
  int im=0;
  int i,j,k,ks;
```

```
/* Functions */
double RN();
void grinit(),grend(),setcrs(),prtchr();
void drwaxs(),drwgra();

/*  Seed for RN generators */
int ix=0,ix1=23450,ix2=34560,ix3=45670,ix4=56780;

/* Probability function for collision */
double p_coll[101];

/* Initialization */
grinit();

 /* -particle density in gas */
g_dens=C_AVOG*G_PRES/22.4e3;

/* -kBT */
tk=C_BOLT*G_TEMP;

/* Mass of molecule */
g_mass=G_MNUM/C_AVOG;

/*  -Velocity Range */
vmin=0.;
vmax=4.*sqrt(tk/g_mass);

/* -Integration */
dv  =(vmax-vmin)/100.;

/*  Reduction of probability function for collision */
sum   =0.;
area  =PI*P_RADI*P_RADI;
fnorm =sqrt(g_mass/(2.*PI*tk));
factor=area*g_dens*fnorm;
for(i=0;i<101;i++){
  v   =vmin+dv *(double)i;
  prob=factor*v*exp(-0.5*g_mass*v*v/tk)*dv;
  sum += prob;
  p_coll[i]=sum;
}
t_scal=sum;
/*  Scale probability */
for(i=0;i<101;i++){
```

```
    p_coll[i]=p_coll[i]/t_scal;
  }

  /* Get time slice:mean collision time */
  dt=1./t_scal;

  /* Write title */
  setcrs(250.,10.);
  prtchr("3-Dimensional Brownian Motion",30);
  setcrs(250.,40.);
  prtchr("- Monte Carlo Simulation[v-space] -",36);

  /* Monte Carlo Simulation */
  /* -Reset time */
  time=0.;

  /* -Initial position of particle(i) */
  x=X0;
  y=Y0;
  z=Z0;
  vx=VX0;
  vy=VY0;
  vz=VZ0;

  /* ..x-y projection */
  xs=X0;
  ys=Y0;
  for(j=0;j<N_COLL;j++){
    /* -Step DT */
    time += dt;

    /* -Before collision */
    /* ..Time interval */
    dt_bef=dt*RN(&ix);
    /* ..Drift */
    x += vx*dt_bef;
    y += vy*dt_bef;
    z += vz*dt_bef;

    /* ..Velocity of molecule */
    chk = RN(&ix);
    for(k=1;k<101;k++){
      if(chk<p_coll[k]&&chk>p_coll[k-1])
ks=k-1;
    }
```

```
    fv=dv*(chk-p_coll[ks])/(p_coll[ks+1]-p_coll[ks]);
    v =dv*(double)ks+fv;
    costh=1.-2.*RN(&ix1);
    sinth=sqrt(1.-costh*costh);
    phai =2.*PI*RN(&ix2);
    vxm  =v*sinth*cos(phai);
    vym  =v*sinth*sin(phai);
    vzm  =v*costh;

    /* -Kinematics */
    /* ..Velocity of CMS */
    vcmx=(P_MASS*vx+g_mass*vxm)/(P_MASS+g_mass);
    vcmy=(P_MASS*vy+g_mass*vym)/(P_MASS+g_mass);
    vcmz=(P_MASS*vz+g_mass*vzm)/(P_MASS+g_mass);

    /* ..Velocity in CMS */
    vrlx=vx-vcmx;
    vrly=vy-vcmy;
    vrlz=vz-vcmz;
    vrl =sqrt(vrlx*vrlx+vrly*vrly+vrlz*vrlz);

    /* ..Moving direction after collision */
    costh=1.-2.*RN(&ix3);
    sinth=sqrt(1.-costh*costh);
    phai =2.*PI*RN(&ix4);

    /* -After collision */
    /* ..Velocity */
    vrlx=vrl*sinth*cos(phai);
    vrly=vrl*sinth*sin(phai);
    vrlz=vrl*costh;
    vx  = vrlx+vcmx;
    vy  = vrly+vcmy;
    vz  = vrlz+vcmz;

    /* ..Drift */
    x += vx*(dt-dt_bef);
    y += vy*(dt-dt_bef);
    z += vz*(dt-dt_bef);
    if(j%N_INTR==0){
      xmon[im] = time;
      ymon[im] = vx;
      im++;
    }
  }
```

```
  /* Draw graph */
  p_vmax=vmax*sqrt(g_mass/P_MASS);
  scale[0][0]=0.;
  scale[0][1]=dt*(double)N_COLL/10.;
  scale[0][2]=500./10.;
  scale[1][0]=-0.5*p_vmax;
  scale[1][1]=p_vmax/10.;
  scale[1][2]=300./10.;
  drwaxs(scale,name,1);
  drwgra(xmon,ymon,100,scale,4);

  /* End graph */
  grend();
}

double RN(ix)
int *ix;
{
  /* Uniform Random Number Generator */
  /*    *ix:Seed                     */
  *ix = 111*(*ix)+9;
  *ix = *ix-10000000*(*ix/10000000);
  return (double)(*ix)/9999999.;
}
```