

十進 BASIC の応用

田淵豊*, 松田修†

On applications of the new educational BASIC

Yutaka, TABUCHI*, Osamu MATSUDA†

Abstract : In this paper, applications of the new educational BASIC which is called JUSSIN BASIC are introduced. JUSSIN BASIC has the precision of a thousand figures by the decimal system. From this, we are usually clear of worry to overflow. Moreover, we can easily draw the graphic of functions. Therefore, we need not get a large amount of knowledge for the programming.

One of the authors is a student in our college. His first simulation is the study on the optical orbit in an ellipse by JUSSIN BASIC. By running a simple program of JUSSIN BASIC, we experience Poncelet's theorem. Furthermore, this study expands to the study of a spheroid. We can see that a trajectory is in contact with a spheroid or a hyperboloid with the help of 3D graphics, and our simulations are very useful for education. Furthermore, these simulations suggest a direction of study of spheroid with n dimension and a study of the periodic optical orbit.

1 はじめに

工学系の高専において、数学は基礎科目として位置付けられ、その上で専門科目を学び、最終学年では卒業研究を行うシステムになっている。このような工学的教育環境の中で、数学的素養を身につけさせるために、体系的にまとめられた教科書による授業や練習問題等の演習は欠くことのできないものである。数学的素養を身に着ける別の方法として、自分で数学的な何かを発見し、発見したものをじっくりと考え、それを証明していくという方法がある。このような数学的体験は、非常に時間がかかる活動ではあるが、しかし卒業研究のプレ研究になるだけでなく、それ以上の効果をもたらす十分意義のある教育活動であることは、阿蘇 [A] の中でも主張されている。しかし現実には上のような研究活動は少人数で行われることが適切であり、40 人対象の正規の授業でその実現はなかなか困難である。そこで、数人の学生に協力を依頼し、課外活動としてこの研究活動を行った。今回の体験的教育活動のモデルは飯高ゼミ [1] の教育活動にある。この十数年一貫したスタイルの飯高ゼミの教育実践は、数学科における第 4 学年の学生に対しコンピュータを用いた体験型のゼミであり、そこには数多くの魅了される成果の蓄積がある。この論文では筆

*津山工業高等専門学校情報工学科 4 年, 岡山県

Electronics and Computer Engineering, Tsuyama National College of Technology, Japan

†津山工業高等専門学校一般科目, 岡山県

General Education, Tsuyama National College of Technology, Japan

email : matsuda@tsuyama-ct.ac.jp

者の一人である当時情報工学科の3年生であった田淵の研究活動報告とその研究方法の教育的効果について論じる。

2 研究テーマ，数学的実験，十進 BASIC

今回行った研究を主体とする教育活動は，次のステップでなされた。

- ステップ1．研究テーマを考える．
- ステップ2．テーマに応じた実験を行う．
- ステップ3．実験結果を考察する．
- ステップ4．定理を作る．
- ステップ5．証明を行う．

特に重要なポイントはステップ1とステップ2である．ステップ1では，学生がやる気になる魅力あふれるテーマが求められる．初めての研究活動の場合，やはりその研究テーマは教師が与えることになるが，研究テーマ自体の研究もそこに含まれる．テーマが決まることで教師と学生が一体となった教育的研究活動がスタートする．しかし，テーマは与えられたが一体どのような道具を使って，どのような方法で数学的な実験に取りかかるべきなのかという問題が残る．

本論の目的の1つは，十進 BASIC という非常に教育的で使用しやすいソフトウェアを提案することでもある．十進 BASIC はフリーのソフトウェアとしてインターネット [J] に公開されており，WINDOWS 版が自由に DOWNLOAD でき，即座に使用可能である．この十進 BASIC が教育上の言語として優れた性格を持つことは，白石 [SIM] や松田 [M] 等でも論じられているが，もう一度十進 BASIC の提案理由を以下に述べる．

- 1．十進 BASIC は BASIC という高級言語であるだけでなく，従来面倒であったグラフィックスのプログラムもノートにグラフを描いていく感覚でプログラミングできるなど非常に使いやすいものになっている．
- 2．実際に BASIC でプログラムを組むためにアルゴリズムをきちんと考えなくてはならず教育的である．
- 3．短時間でプログラミングできるため，プログラムへの労力は最小限であり，数学的な考察に十分時間がかけられる．
- 4．他の言語，たとえば C や FORTRAN を学ぶ上でも基礎となる．

本論では，研究テーマを楕円と回転楕円体の考察とし，その研究過程を詳細に報告することで，本教育研究活動の意義を主張する．

楕円と回転楕円体に関するシミュレーション実験を十進 BASIC を主体として行ったのは田淵である．楕円の場合と回転楕円体の場合のシミュレーション結果を用い，田淵が行うゼミの中からいくつかの定理が示され，その証明に関する議論も多くの時間を使ってなされた．

第1の研究は，楕円の中の反射を繰り返すレーザーの研究である．田淵は独自に十進 BASIC を使って，「レーザーは元の楕円と焦点を共有する楕円かまたは双曲線に接する」という命題を得た．そして初等的ではあるがその証明も与えられた．しかしこれはすでによく知られているポンスレ (Poncelet) の定理 [cf. KW] の特殊なケースである．

第2の研究は回転楕円体における反射を繰り返すレーザーの研究の考察である。これも同様な結論が出ることを期待しシミュレーションを行った。残念ではあるが十進 BASIC において3次元グラフィックスは実現できない。さらに軌道計算においても計算速度が遅いという状況があった。そこで、C 言語で軌道のデータを計算させ、そのデータを POV-RAY というフリーソフトで描画させた。しかしセクション4の十進 BASIC プログラムと最後のセクションのC プログラムを比較してわかるように、その命令文は十進 BASIC のものとほとんど変わらない。すなわち十進 BASIC はC などの言語学習に応用可能なのである。さてシミュレーションの結果等は本論で詳しく述べるが、レーザーが接している面は回転楕円面または回転双曲面と考えられる。筆者らはこの証明に取り組んだが、しかし結論を得るにいたらなかった。後でわかったことではあるが、このような研究は S. Chang, R. Friedberg [SR] にある。しかし、田淵が自力でこのような結論(予想)を得た点は、教育上非常に価値のあるものと考えられる。さらにこのような豊富な研究課題に取り組めた背景には、初めの十進 BASIC のシミュレーションによる動機付けがあった点を改めて主張する。

3 十進 BASIC の概要

十進 BASIC は JIS 規格の FULL-BASIC を基本に作られた WINDOWS 上で動くフリーソフトである。その特徴の1つは十進 1000 桁までの精度が保証されていることである。そのため比較的常識的な計算においてはオーバーフローを起こすことがなく、プログラミングの際この点にそれほど注意を払わなくてもよい。さらに初心者が非常に便利に感じるところは、関数のグラフが簡単に描けるという点にある。また複素数の演算機能も装備している。したがってガウス平面上のグラフィックスも可能である。ここでは特にグラフィックスについて簡単に説明する。

まず、 $y = x^2$ のグラフを描くプログラムを紹介する。

```
DEF f(x)=x^2
SET WINDOW -4, 4, -1, 10
DRAW axes
FOR x=-4 TO 4 STEP 0.01
    PLOT LINES: x, f(x);
NEXT x
END
```

1行目の $DEF f(x)=x^2$ で、関数を定義し、途中の FOR NEXT 文が曲線を描くプログラムである。FOR $x=-4$ TO 4 STEP 0.01 という命令文は、 x の値を -4 から 4 まで、0.01 間隔で刻んで計算せよということである。PLOT LINES: $x, f(x);$ で、点 $(x - 0.01, f(x - 0.01))$ と点 $(x, f(x))$ の短い区間を直線で結んでいくことになる。画面設定は、SET WINDOW $-4, 4, -1, 10$ で行われ、ここでは x の範囲を $-4 \leq x \leq 4$ とし、 y の範囲を $-1 \leq y \leq 10$ と設定している。また座標軸を描くために、DRAW axes という命令を使用する。このわずか7行の命令文で、 $y = x^2$ のグラフを描くことが可能になる。そしてこのプログラムは通常ノートにグラフを描く感覚に対し少しも違和感を感じない。また上のプログラム例のように、通常の BASIC と異なり行番号も不必要である。

次にパラメータで表示されたグラフのプログラムの紹介する．次は楕円を描くプログラムの例である．

```
option angle degrees
SET WINDOW -4, 4, -4, 4
DRAW axes
FOR t=0 TO 360
    PLOT LINES: 3*cos(t), 2*sin(t);
NEXT t
END
```

説明の必要がある文は1行目の `option angle degrees` である．これは角を60分法で扱うことを意味する．何も書かなければ弧度法となる．弧度法については π に誤差があることが注意点である．ついでに言えば，プログラムの `PLOT LINES: 3*cos(t), 2*sin(t);` の文を `PLOT LINES: cos(3*t), sin(2*t);` に書き換えることで，リサージュ曲線も簡単に描くことができる．

その他の命令文は通常の BASIC とほとんど変わらない．また非常に教育的な機能として「ステップ実行機能」というものが装備されている．これはプログラムを実行する際に1行ずつの命令文を確認しながら実行していく機能である．プログラムを始めて勉強する学生等は，この機能を使うことで，今コンピュータはどの命令文を行い，どのような計算を行っているのかをじっくりと観察でき，アルゴリズムの学習に役立つ．さらに得られたデータやグラフは，txt形式ファイルやbmp形式ファイルとして保存でき加工するのに非常に便利である．

4 楕円内を反射するレーザーの軌跡

xy 平面における長軸を x 軸とする楕円内を反射するレーザーの軌跡（以後レーザー反射ということにする）を考える．まず，十進 BASIC のプログラムを走らせレーザー反射のシミュレーションを試みた．その結果，発射する位置，発射する角度により，レーザーの包絡線は楕円や双曲線のようなものになって現れることがわかる．

考察の結果，レーザーを発射する位置は以下のように三通りに分類することができる．

1. 焦点から発射する場合
2. x 軸上の焦点より外側から発射する場合
3. x 軸上の2つの焦点の間から発射する場合

以下詳細に見ていく．

(1) 焦点からレーザーを発射した場合．

この場合レーザーは楕円のもうひとつの焦点を通過する．これを繰り返していくことにより，レーザービームは長軸上に収束する．この様子を図1に示す．三角はレーザーの始点を表し，黒い点は焦点を表す．

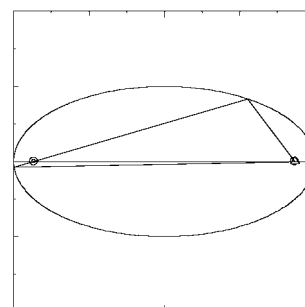


図1：焦点から発射した場合

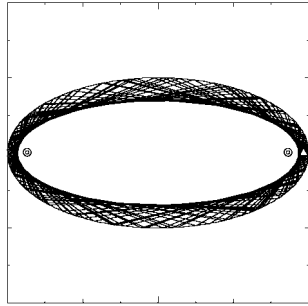


図 2 : 焦点より外側から発射した場合

(2) 焦点より外側からレーザーを発射した場合.
 この場合, 図 2 のようなシミュレーション結果が得られる. 楕円の内側で反射し, x 軸上では, 焦点と焦点の間を通ることはない. 内側にある楕円に全てのレーザーが接しているように見える.

(3) 2 つの焦点の間から発射した場合.
 この場合, 図 3 のようなシミュレーション結果が得られる. これも内側にある双曲線に全てのレーザーが接しているように見える.

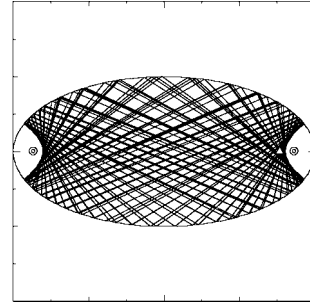


図 3 : 2 つの焦点の間から発射した場合

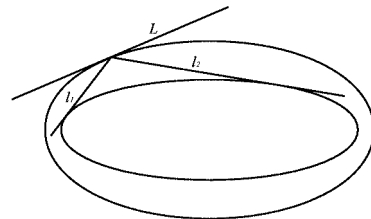
定理 xy 平面における楕円内のレーザー反射を考える.

- (1) 焦点からレーザーを発射した場合. レーザーは焦点を通過する軌道を通り, 無限回の反射において, その軌道は長軸に収束する.
- (2) 焦点より外側から発射した場合. レーザーは元の楕円の焦点を共有する楕円に接する.
- (3) 2 つの焦点の間から発射した場合. レーザーは元の楕円の焦点を共有する双曲線に接する.

□

(2),(3) はポンスレの定理の特殊なものである. ここでは著者らが行った初等的な証明を紹介する.
 (2)(3) を証明するには次の命題 1, 命題 2 を証明すればよい.

命題 1 右図のように二つの焦点が一致している楕円を, E_1, E_2 とし, E_2 は E_1 より小さい楕円とする. E_1 上の点 (x_0, y_0) の接線を L , 点 (x_0, y_0) を通り E_2 に接する 2 直線を l_1, l_2 とする. このとき l_1, l_2 は L を軸として対称となる.



証明 2 つの楕円をそれぞれ,

$$E_1 : \frac{x^2}{a_1^2} + \frac{y^2}{b_1^2} = 1 \quad E_2 : \frac{x^2}{a_2^2} + \frac{y^2}{b_2^2} = 1$$

とする. 2 つの楕円の焦点が一致していることから, 焦点の座標のひとつを $(c, 0)$ とすると, $c^2 = a_1^2 - b_1^2 = a_2^2 - b_2^2$ が成り立つ. 3 つの直線の方程式を, $l_1 : y = m_1x + n_1, l_2 : y = m_2x + n_2, L : y = m_0x + n_0$ とする.

直線 l_1 と l_2 は点 (x_0, y_0) を通り, 楕円 E_2 に接していることから,

$$(x_0^2 - a_2^2)m_i^2 - 2x_0y_0m_i + y_0^2 - b_2^2 = 0$$

$$(i = 1, 2)$$

を得る．これより，

$$m_1 = \frac{x_0y_0 + \rho \frac{(x_0y_0)^2 - (x_0^2 - a_2^2)(y_0^2 - b_2^2)}{x_0^2 - a_2^2}}{x_0^2 - a_2^2} \quad (1)$$

$$m_2 = \frac{x_0y_0 - \rho \frac{(x_0y_0)^2 - (x_0^2 - a_2^2)(y_0^2 - b_2^2)}{x_0^2 - a_2^2}}{x_0^2 - a_2^2} \quad (2)$$

を得る．命題 1 を証明するためには， l_1 と L のなす角と， l_2 と L のなす角が等しいことをいえばよい，つまり直線 l_1 の方向ベクトルは， $(1, m_1)$ であり，直線 L の方向ベクトルは $(1, m_0)$ であるから，

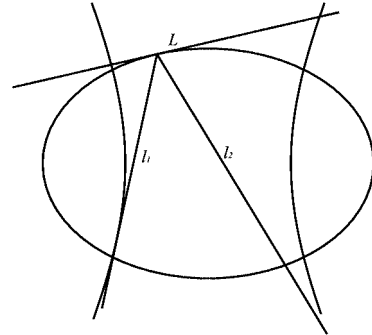
$$\rho \frac{(1, m_1)(1, m_0)}{1 + m_1^2} \rho \frac{(1, m_0)}{1 + m_0^2} = \rho \frac{(1, m_2)(1, m_0)}{1 + m_2^2} \rho \frac{(1, m_0)}{1 + m_0^2} \quad (3)$$

となることを言えばよい．ここで， $(x_1, x_2)(y_1, y_2)$ は 2 つのベクトル (x_1, x_2) と (y_1, y_2) の内積を表す．上の式に式 (1), (2) と $b_2^2 = a_2^2 - c^2$ を式 (3) に代入すると，等式

$$a_1^4 - (x_0^2 + y_0^2 + c^2)a_1^2 + c^2x_0^2 = 0 \quad (4)$$

が得られ，式 (4) を証明すればよいことになる．等式 (4) が成立することは， $x_0^2/a_1^2 + y_0^2/b_1^2 = 1$ と， $c^2 = a_1^2 - b_1^2$ を左辺に代入してみればすぐにわかる． \square

命題 2 2 つの焦点が一致している楕円と双曲線をそれぞれ E, H とする． E 上の点 (x_0, y_0) の接線を L ，点 (x_0, y_0) を通り， H に接する 2 直線を l_1, l_2 とする．このとき l_1, l_2 は L を軸として対称となる．



証明 命題 1 の証明と同様な方法で証明できる． \square

定理は初等的に証明されるが，初期値の位置とレーザーの発射方向から，どのくらいの反射を繰り返すとともに位置にもどるのか，それとも永遠にもどらないのかといったような新たな問題が浮かび上がってくる．これについては未だ結論を得ていない．

4.1 十進 BASIC のプログラム

以下がシミュレーションで使用した十進 BASIC のプログラムである．

! 楕円内のレーザー反射のシミュレーションプログラム

option angle degrees
SET WINDOW -3.5, 3.5, -3.5, 3.5

! DRAW axes

LET a=3 ! 長軸
LET b=2 ! 短軸
LET c0=60 ! 初期位置
LET f0=200 ! 初期角度
LET n=200 ! 反射回数

SET AREA COLOR 6 ! 黄色指定.
PAINT 0, 0 ! 外側を黄色で塗りつぶす.

! 楕円を描く

SET AREA COLOR 10 ! 緑色指定.
FOR t=0 TO 360 STEP 0.5
PLOT LINES: a*cos(t), b*sin(t);
NEXT t
PLOT LINES
PAINT 0, 0 ! 楕円内を緑色で塗りつぶす.

! 焦点の位置

LET f=sqr(a^2-b^2)
SET POINT STYLE 3 ! 点の形をアスタリスクに指定.
SET POINT COLOR 4 ! 点の色を赤色に指定.
PLOT POINTS: -f, 0; f, 0

SET LINE COLOR 1 ! 線の色を黒に指定.
LET x0=a*cos(c0) ! 最初の位置をセットする.
LET y0=b*sin(c0)
LET f1=f0

! ビームの軌跡.

for i=1 to n ! n回の反射を For 文で行う.
LET z0=tan(f1) ! ビームの傾きを z0 とする.
LET m=y0-x0*z0 ! ビームの切片を m とする.
LET alpha=1/a^2+z0^2/b^2 ! xに関する解と係数の関係を考える.
LET beta=(2*z0*m)/b^2 ! xに関する解と係数の関係を考える.
LET x1=-beta/alpha-x0 ! ビームの当たった x 座標を x1 とする.
LET y1=x1*z0+m ! ビームの当たった y 座標を y1 とする.
if x1-x0>0 then
LET st=0.001 ! ビームの動く速さを st で指定する.
else
LET st=-0.001
end if
for t=x0 to x1 step st
PLOT LINES: x0, y0; t, z0*t+m
next t
! (x1, y1) での接線のベクトル表示を (1, dy) とする.
LET dy=-(b^2/a^2)*(x1/y1)
LET v0=x0-x1 ! ビームベクトルの x 成分を v0 とする.
LET v1=y0-y1 ! ビームベクトルの y 成分を v1 とする.
LET vm=sqr(v0^2+v1^2) ! ビームの長さを vm とする.
LET dym=sqr(1+dy^2) ! 接線ベクトルの長さを vm とする.
LET C0=(v0+v1*dy)/(vm*dym)
LET vdyC=ACOS(C0) ! 接線とビームとのなす角を vdyC とする.
LET dyc=ATN(dy) ! 接線と x 軸とのなす角を dyc とする.
if y1>0 then
LET f1=180+vdyC+dyc

```

else
  LET f1=180-vdyc+dyc
end if
LET x0=x1
LET y0=y1
next i
END

```

5 回転楕円面の中の反射

回転楕円面について楕円の場合と同様な考察をコンピュータシミュレーションで行う。扱う回転楕円面の定義式は以下とする。

$$\frac{x^2}{2^2} + y^2 + z^2 = 1$$

十進 BASIC は 3 次元グラフィックスが出来ないことから POV-RAY[P] というフリーソフトを使ってグラフィックスを実現した。その際 3 次元データの計算は C 言語プログラムで行わせた。

5.1 長軸 (x 軸) 上の点からレーザーを発射させた場合

この場合は、平面上の楕円の場合に帰着する。

(1) x 軸上の点から発射した場合。

図 4 で示すように、焦点から発射されたレーザーはもうひとつの焦点に収束し、このレーザーの軌跡は、平面の楕円と同じく長軸上に収束する。

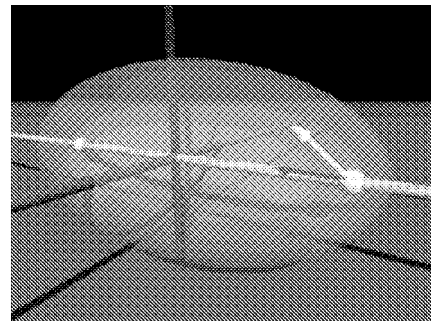


図 4：焦点から発射した場合

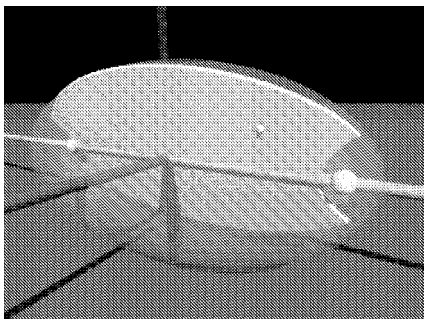


図 5： x 軸上の 2 つの焦点の間から発射

(2) x 軸上の 2 つの焦点の間から発射した場合。
この場合は、回転楕円面と焦点を共有する双曲線にレーザーは全て接する。

- (3) x 軸上の焦点より外側から発射した場合 .
 この場合は , 回転楕円面と焦点を共有する楕円
 にレーザーは全て接する .

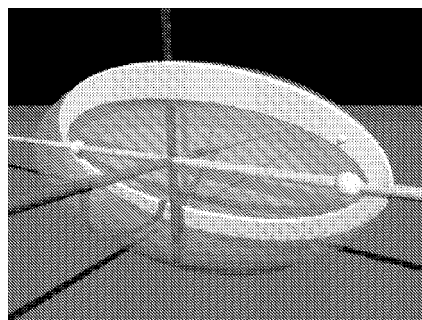


図 6 : x 軸上の焦点より外側から発射

5.2 一般の点から発射した場合

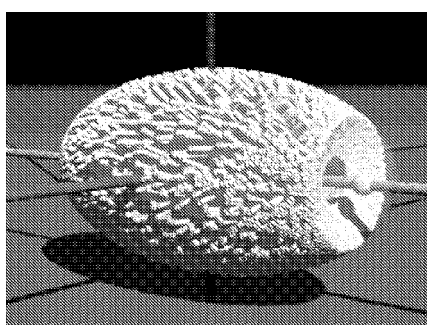


図 7 : 一般の点から発射した場合の包絡線



図 8 : 半分に切った図

一般の点から発射した包絡面は図 7 のようになる . 図 8 はそれを半分に切断したものである . これらの図から , 内側に回転楕円面と , その側面には , 回転双曲面が現れている . そこで現れた図に適当な回転楕円面 , 回転双曲面をオーバーラップさせるとほぼ一致した .

定理 回転楕円面内の一般の点から発射したレーザーは , 元の回転楕円面と焦点を共有する回転楕円面か回転双曲面に接する (cf. [SR]) . □

6 課題

本研究を通して , 楕円内 , 回転楕円面内でレーザ光線を発射したときの振る舞いをシミュレーション等により理解することができた . 計算機で算出したデータには誤差が多くつきまといその作業は大変であった . しかし現れた結果は非常に美しく , 興味深いものであったし , 教育的にも価値のある活動であったと思う . さらに今後行わなければならない研究課題も生まれた . 最後に現在の研究活動を以下に列挙しておく .

1. 初期値の位置とレーザーの発射方向と軌道の周期性の問題 .
2. 一般の楕円面内での同様な研究 .
3. 4次元空間内 , またそれ以上の n 次元空間内での同様な研究 .

7 Cプログラム

以下は回転楕円面内でのシミュレーションで使用したCプログラムである。

```
/*
 * 楕円球レーザー反射をシミュレートします
 * プログラムでは、レーザー発信元、レーザー反射先を
 * 標準出力に出力します。
 */
#include <math.h>
#include <stdio.h>
#define OUTPUT_INFO
double cdot( double a[], double b[] );
double clength( double v[] );

/* 固定パラメータの定義 */
double a = 2.0;
double b = 1.0;
double c = 1.0;
double R = 2.0;
double dt = 0.0001;

/* マクロの定義 */
#define PX (x[0])
#define PY (x[1])
#define PZ (x[2])
#define VX (v[0])
#define VY (v[1])
#define VZ (v[2])
#define SPEED (1.0) /* 遅い 速い */

int main( void )
{
    double x[] = { 0.0, -1.9, 0.0 },
           v[] = { 3.0, 2.0, 1.5 },
           ox[] = { 0.0, -1.9, 0.0 };
    int i;
#ifdef OUTPUT_INFO
    printf( "\n" );
    printf( "/*\n" );
    printf( " * r = { %+.2lf, %+.2lf, %+.2lf } \n", PX, PY, PZ );
    printf( " * v = { %+.2lf, %+.2lf, %+.2lf } \n", VX, VY, VZ );
    printf( " *\n" );
    printf( " * a = %.2lf\n", a );
    printf( " * b = %.2lf\n", b );
    printf( " * c = %.2lf\n", c );
    printf( " */\n" );
    printf( "\n" );
#endif
    {
        double k;
        k = clength( v ); k *= SPEED;
        for( i = 0; i < 3; i++ ) {
            v[ i ] /= k;
        }
    }
    while( /* 無限ループ */ 1 ) {
```

```

for( i = 0; i < 3; i++ ) {
    x[ i ] += v[ i ] * dt;
}

if( (( PX * PX ) / ( a * a )) +
    (( PY * PY ) / ( b * b )) +
    (( PZ * PZ ) / ( c * c )) >= ( R * R ) ) {
    /* ステップを後退 */
    for( i = 0; i < 3; i++ ) {
        x[ i ] -= v[ i ] * dt;
    }
}
#ifdef __DEBUG__
printf( "Vec( %13.8f, %13.8f, %13.8f, %13.8f, %13.8f, %13.8f )\n",
        ox[ 0 ], ox[ 1 ], ox[ 2 ], PX, PY, PZ );
#endif
{
    double dzx[] = { 0.0, 0.0, 0.0 }, dzy[] = { 0.0, 0.0, 0.0 };
    double n[] = { 0.0, 0.0, 0.0 }, r[] = { 0.0, 0.0, 0.0 };
    double vv[] = { 0.0, 0.0, 0.0 };
    double k, l;
    /* 偏微分した結果に値を代入 */
    dzx[ 0 ] = sqrt( R * R - (( PX * PX ) / ( a * a ))
                    - (( PY * PY ) / ( b * b ))) * ( a * a );
    dzx[ 2 ] = -c * PX;
    dzy[ 1 ] = sqrt( R * R - (( PX * PX ) / ( a * a ))
                    - (( PY * PY ) / ( b * b ))) * ( b * b );
    dzy[ 2 ] = -c * PY;
    if( PZ < 0.0 ) { /* Z < 0 の時の対処 */
        dzx[ 2 ] = -dzx[ 2 ];
        dzy[ 2 ] = -dzy[ 2 ];
    }
    /* 法線ベクトルを求める dzx × dzy */
    n[ 0 ] = dzx[ 1 ] * dzy[ 2 ] - dzx[ 2 ] * dzy[ 1 ];
    n[ 1 ] = dzx[ 2 ] * dzy[ 0 ] - dzx[ 0 ] * dzy[ 2 ];
    n[ 2 ] = dzx[ 0 ] * dzy[ 1 ] - dzx[ 1 ] * dzy[ 0 ];
    /* 途中計算に必要なベクトル r を求める */
    l = cdot( v, n ) / cdot( n, n );
    for( i = 0; i < 3; i++ ) {
        r[ i ] = v[ i ] - l * n[ i ];
    }
}
#ifdef __DEBUG__
printf( "Vec( %13.8f, %13.8f, %13.8f, %13.8f, %13.8f, %13.8f )\n",
        PX, PY, PZ, PX + n[ 0 ], PY + n[ 1 ], PZ + n[ 2 ] );
printf( "Vec( %13.8f, %13.8f, %13.8f, %13.8f, %13.8f, %13.8f )\n",
        PX, PY, PZ, PX + v[ 0 ], PY + v[ 1 ], PZ + v[ 2 ] );
printf( "Vec( %13.8f, %13.8f, %13.8f, %13.8f, %13.8f, %13.8f )\n",
        PX + v[ 0 ], PY + v[ 1 ], PZ + v[ 2 ],
        PX + v[ 0 ] - r[ 0 ], PY + v[ 1 ] - r[ 1 ], PZ + v[ 2 ] - r[ 2 ] );
#endif
/* 反射ベクトルを求める */
k = 2 * cdot( v, r ) / cdot( r, r );
for( i = 0; i < 3; i++ ) {
    vv[ i ] = k * r[ i ] - v[ i ];
}
/* もともとの速度ベクトル v に vv をコピー */
for( i = 0; i < 3; i++ ) {
    v[ i ] = vv[ i ];
}
for( i = 0; i < 3; i++ ) { /* 前の値をコピー */

```

```

        ox[ i ] = x[ i ];
    }
    {
        static int rpp = 0;
        if( rpp++ == 1500 )
            break;
    }
}
return 0;
}
double clength( double v[] )
{
    return sqrt( cdot( v, v ));
}
double cdot( double a[], double b[] )
{
    return a[ 0 ] * b[ 0 ] +
           a[ 1 ] * b[ 1 ] +
           a[ 2 ] * b[ 2 ];
}

```

参考文献

- [A] 阿蘇和寿, 数学の授業における学生の探求活動 - テクノロジーの効果的な活用に向けて - 日本数学教育学会高専・大学部会研究論文誌, VOL9, NO.1, p31-49, 2002.8
- [I] 矢萩昇太郎他, 2002年飯高ゼミ卒業研究論文集, 学習院大学理学部数学科
- [J] 十進 BASIC ホームページ, <http://hp.vector.co.jp/authors/VA008683/>
- [KW] 川又雄二郎, 射影空間の幾何学, 朝倉書店, 2001
- [M] 松田修, 川部健, 中村重之, 橋本淳, 十進 BASIC による工学へのニューアプローチ, 津山工業高等専門学校, 2003.11
- [SIM] 白石和夫, 飯高茂, 松田修, Windows 環境での新しい BASIC, 数学教育の会数学教育研究 1号, 1999.1
- [P] POV-RAY ホームページ, <http://www.povray.org/>
- [SR] Shau-Jin Chang and Richard Friedberg, Elliptical billiards and Poncelet's theorem, Journal of Mathematical Physics Vol 29(7) pp. 1537-1550. July 1988