

# College Analysis レファレンスマニュアル

—教育・科学 1—

## 目次

1. フラクタルビューア 2 D .....	1
2. フラクタルビューア 3 D .....	12
3. カオスビューア (未完成) .....	18
4. 3 Dモデルビューア (未完成) .....	26
5. くるくるエディット .....	28
6. おもしろグラフ .....	30
7. 3 Dフォトメーカー .....	32
8. 3 Dバナーメーカー .....	34
9. 惑星シミュレーション .....	37
10. 電荷と電場 .....	46
11. 質点系の運動 .....	62
12. 電流と磁場 .....	78
13. 特殊相対論的視覚効果 .....	85

## 1. フラクタルビューア 2D

フラクタルビューア 2D の実行画面を図 1 に示す。



図 1 フラクタルビューア 2D

このプログラムは複素数列の収束問題を利用したフラクタルと再帰的な手法によるフラクタルの 2 種類を扱う。複素数列の収束問題は基本的に  $z_n = f(z_{n-1}) + c$  の形の代表的な力学系について、Mandelbrot 集合のように、 $z_0$  を固定して  $c$  の収束、周期振動、発散領域を調べる場合と、充填 Julia 集合のように  $c$  を固定して  $z_0$  の収束、周期振動、発散領域を調べる場合を扱っている。収束と周期振動は複数の点の周りで生じることがあるので、よく利用される色付け方法である収束までのループの回数による色分けの他に、どの点に収束するのか、または振動の周期はいくらかによる色付け方法も加えている。また、色付けに用いる色の種類は予め登録してある色パターンによって指定できる。収束、周期振動、発散を分けて表示したい場合は、それぞれ赤系の「Autumn」、緑系の「Spring」、青系の「Aqua」などの色パターンを選ぶとどの部分がどんな特性を持っているのかをはっきり見ることができる。これらの色パターンのサンプルは、メニュー「ヘルプー色パターン」を選択すれば表示される。

2 次元の再帰的手法には、線、円、正多角形（正三角形から正六角形）の描画要素によってフラクタルを描く機能がある。描画手順は Logo のタートルグラフィックスに似た独自のフラクタル記述言

語によって記述される。その簡単な仕様は「書式」ボタンをクリックすることにより表示される。描画要素の色は、フラクタルの次数と色パターンで指定される。

## 1.1 複素数収束によるフラクタル

複素数の収束を利用したフラクタルでは、Mandelbrot 集合と充填 Julia 集合の他に我々のプログラムには以下の力学系が含まれている。

- 1)  $z_n = z_{n-1}^k + c$  (Mandelbrot 集合, 充填 Julia 集合)
- 2)  $z_n = cz_{n-1}(1 - z_{n-1}^k)$
- 3)  $z_n = cz_{n-1}(1 - 1/z_{n-1}^k)$
- 4)  $z_n = z_{n-1}^k (c - z_{n-1})/(1 - \bar{c}z_{n-1})$
- 5)  $z_n = (|\operatorname{Re}(z_{n-1})| - i|\operatorname{Im}(z_{n-1})|)^2 + c$  (Burning Ship Fractal)

ここに、 $k = 1, 2, \dots$ ,  $z_i \in \mathbb{C}$ , ( $i = 1, 2, \dots$ ),  $c \in \mathbb{C}$ である。また、1) ~ 4) については、Mandelbrot 集合のように、 $z_0$ を固定して $c$ の収束、周期振動、発散領域を調べる場合と、充填 Julia 集合のように $c$ を固定して $z_0$ の収束、周期振動、発散領域を調べる場合が含まれている。

これらを図に描く際、美しさを強調するために、例えば $|z_n - z_{n-1}| < 0.00001$ ,  $|z_n| > 1000$ のように収束と発散の値を定め、これに到達した $n$ の値を用いて色分けする。繰り返し回数 $n$ の最大値は、「回数」テキストボックスに記入する。これを超える場合は収束も発散もせず、周期振動する場合と判定するが、これにも繰り返しの最終的な到達点の近傍に何回目で近づくかを再計算することによって色を指定する。色の指定は、「色収束」、「色周期」、「色発散」コンボボックスを使って既定の色の組み合わせから選ぶ。繰り返し回数については、描画の正確さと美しさを優先させた値を取っているが、描画に時間がかかる場合は、回数を 1000 回から 100 回程度に減らしてよい場合もある。

この他に収束と周期振動については、もう 1 つ色付けの方法がある。収束については収束点が複数ある場合、どちらの収束点に収束するか、周期振動については周期がいくつかということによる色付けである。これは、「色収束」、「色周期」コンボボックスの上にある「種類」チェックボックスをチェックすることで選択される。

図 2.1 のメニュー画面では、まず集合の種類を選択する。力学系の数式の他に、上で説明した Mandelbrot 集合的な描画であれば「M 集合」、充填 Julia 集合的な描画であれば「J 集合」になっている。その他特殊なものもある。これらの集合の選択をすると次数 $n$ の値、描画範囲、色の種類など代表的な例の設定が行われるので、そのまま「描画」ボタンをクリックしてもサンプル画像が見られる。これ以外はユーザーの設定となる。

「描画」ボタンで表示された画面上の一部を拡大したい場合は、画面上でマウスをドラッグすると、描画範囲が赤い四角形で選択され、そのまま「描画」ボタンをクリックすると選択範囲が拡大表示さ

れ、その領域が描画画面の他に、メニュー画面のテキストボックスにも表示される。「範囲指定」チェックボックスにチェックがない場合、領域を選択していないときにはデフォルトの範囲になるが、「範囲指定」はチェックしておく方が分り易い。実部と虚部の範囲をデフォルトに直す時には「Reset」ボタンをクリックする。

何度も拡大するといくつか前の段階の画像を見たくなることがある。その際には、「<<」、「>>」ボタンで前後の画像を表示することができる。

例として一部分を拡大していった Mandelbrot 集合の図を図 1.2a、図 1.2b、図 1.2c、図 1.2d に示す。

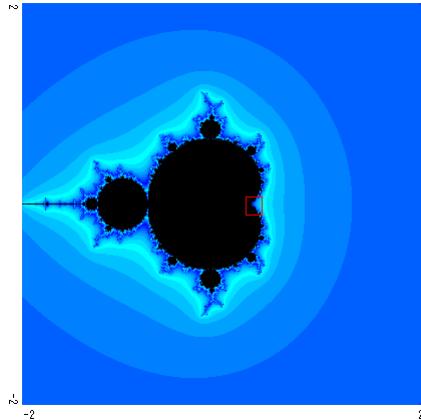


図 1.1a Mandelbrot 集合

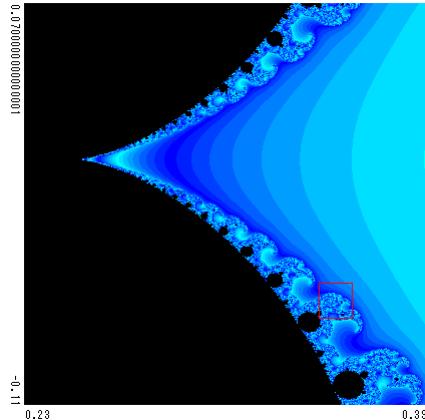


図 1.1b 拡大 1

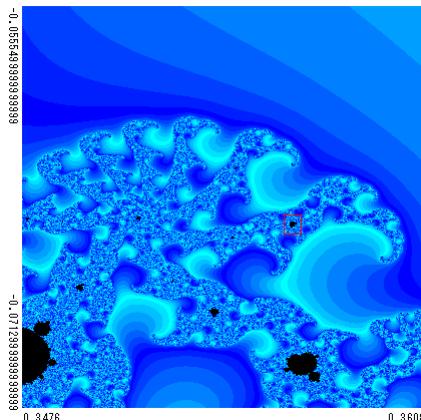


図 1.1c 拡大 2

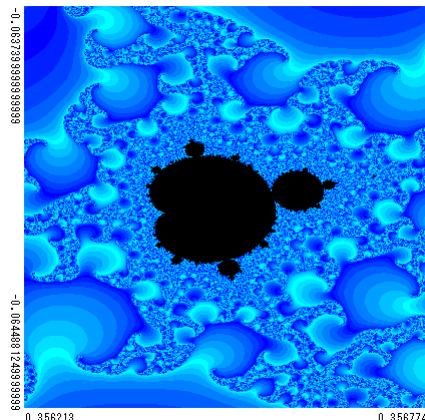


図 1.1d 拡大 3

次に収束の種類による塗り分けの例を示す。図 2.1.2 は Mandelbrot 集合で、収束、1 周期、2 周期、…のように塗り分けた例である。中央左側（2 周期）や上下の丸い部分（3 周期）の色が変わっている

ことが分る。16 周期以上はすべて白色にするように設定している。

通常、充填 Julia 集合の図は、例えば  $c = 0.3 + 0.3i$  などのように、定数の値を図 1.2 の Mandelbrot 集合の中央部分の収束領域に与えて描くことが多いが、ここでは 2 周期振動となるその左側の領域内の値  $c = -1$  に取って図 1.3 を描いてみる。これは周期振動の種類を色分けする設定で実行したが、2 周期点が 1 種類であることから、収束領域は単一色となっている。

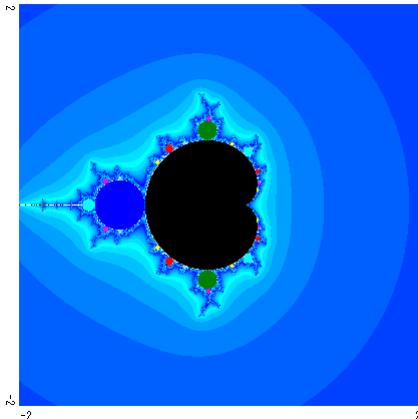


図 1.2 収束・周期で色分けした Mandelbrot 集合

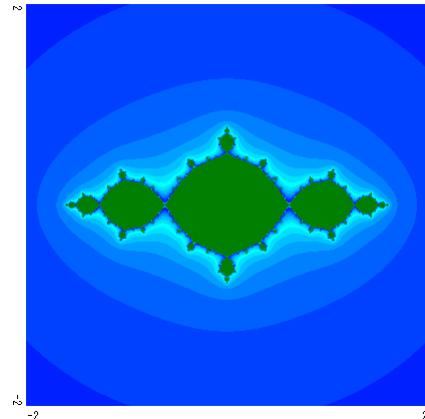


図 1.3 2 周期の定数を用いた充填 Julia 集合

図 1.4 は  $J$  集合  $z_n = c(z - 1/z)$ ,  $c = 0.2 + 0.5i$  の場合の 2 周期振動領域の周期振動の種類（収束する周期振動点）によって塗り分けた例である。この場合画面全体が 2 周期振動領域で、収束点の種類が 2 種類（周期振動する 2 つの点が 2 種類）あることが分る。印刷の画像では見えないと思うが、図の中に無限遠の引力圏が点在する。抜き出して表示すると図 1.5 のようになる。これらは図 1.4 の複雑な模様の部分に点在していることが分る。この図もフラクタルのメニューに含まれている。

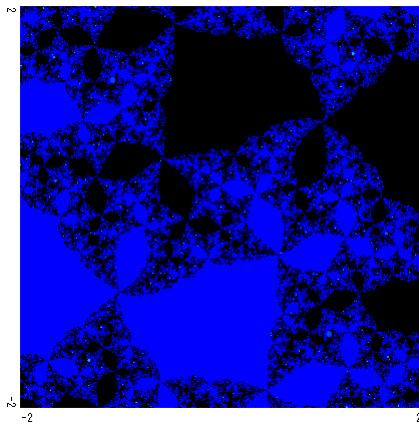


図 1.4 周期振動の収束点による色分け

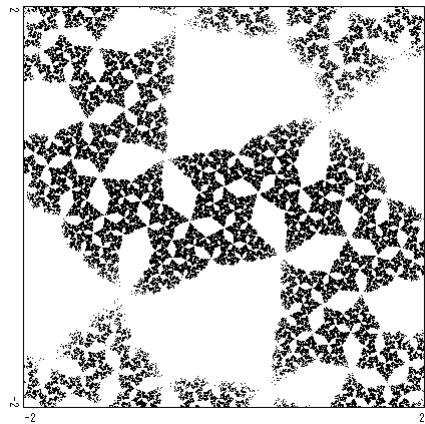


図 1.5 無限遠の引力圏

これらの集合以外で、我々のプログラムに含まれる集合とそのサンプルを以下に示す。図 1.6 は M 集合で、 $z_n = cz_{n-1}(1 - z_{n-1}^3)$ ,  $z_0 = 0.63$  の発散領域とそれ以外の塗り分け、図 1.7 は M 集合で、 $z_n = cz_{n-1}(1 - z_{n-1}^{-2})$ ,  $z_0 = i$  の周期による塗り分けである。

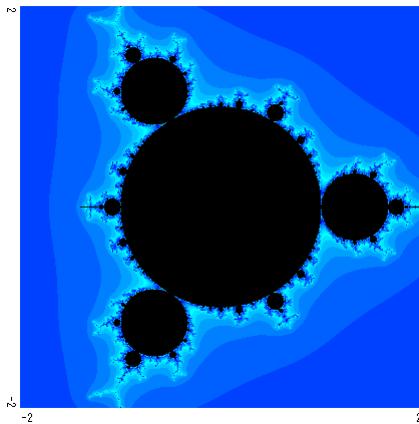


図 1.6 サンプル 1

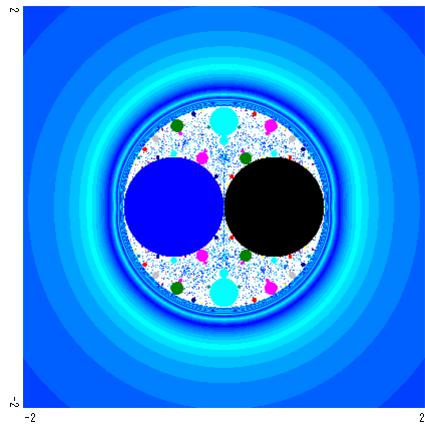


図 1.7 サンプル 2

図 1.8 は M 集合で、 $z_n = z_{n-1}(z_{n-1} - c)/(1 - \bar{c}z_{n-1})$ ,  $z_0 = 0.1i$  の 3 領域による塗り分け、図 1.9 は充填 Julia 集合で、 $z_n = z_{n-1}^2 + c$ ,  $c = 0.3 + 0.3i$  の発散と他の領域による塗り分けである。ここに  $\bar{c}$  は  $c$  の複素共役である。

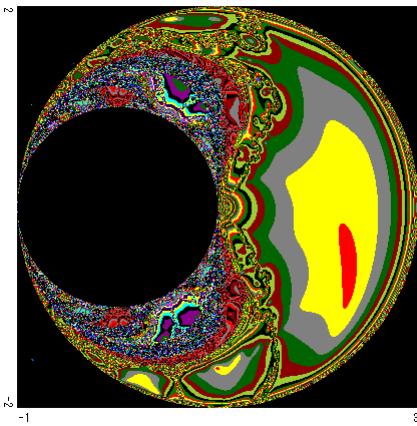


図 1.8 サンプル 3

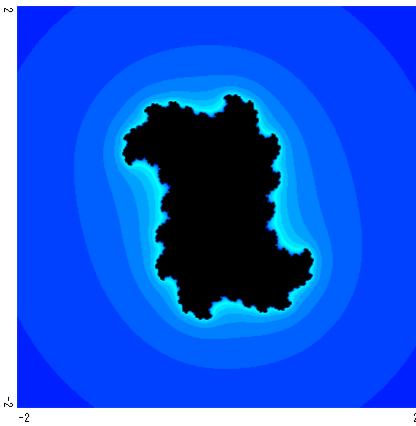


図 1.9 サンプル 4

図 1.10 は J 集合で、 $z_n = cz_{n-1}(1 - z_{n-1}^3)$ ,  $c = 1.2 + 0.1i$  の発散と他の領域による塗り分けで、図 1.11 は J 集合で、 $z_n = z_{n-1}^2(z_{n-1} - c)/(1 - \bar{c}z_{n-1})$ ,  $z_0 = 1.1 + 2.1i$  の 2 つの領域による塗り分けである。

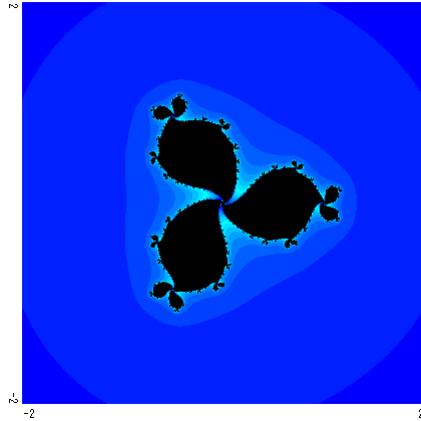


図 1.10 サンプル 5

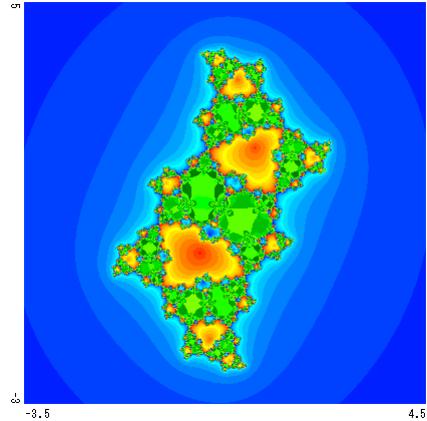


図 1.11 サンプル 6

## 2.2 再帰的方法によるフラクタル

再帰によるフラクタル画像の描画はプログラミングの基礎として多くのプログラマが経験する。しかし Windows 環境ではプログラム処理系のグラフィックの扱いに多少の基礎知識を必要とするので、多くの人が手軽に、というわけには行かない。そこで我々はこれらの知識に煩わされることなく、再帰処理だけを頭に入れてフラクタル画像が作れる簡単なマクロ言語を開発した。これを用いることで、何の準備もなく 2 D のフラクタル画像を作成することができる。また、この言語はかなりの部分で 3

Dのフラクタルに応用できるので、次章の内容と合わせて読んでもらいたい。

まず簡単な例を示す。図 1 のメニューで「テキストから」ラジオボタンをチェックし、「テキストエディタ」ボタンをクリックする。テキストエディタが開かれるが、この中にプログラムを書き込み、完成したら「テキスト入力」ボタンをクリックして、プログラムをグリッドへコピーし保存するのがよい。グリッドからテキストへプログラムを書き出すのは、「テキスト出力」ボタンを利用する。

プログラムは基本的に LOGO のタートルグラフィックスのようになっており、左端 (0,0) から右端 (1,0) へ向けてスタートする。命令はコマンドとパラメータからなっているが、パラメータがない場合もある。パラメータには数式も使える。プログラムのサンプルを図 2.1 に示す。

```
turn 45
frac 1/2^0.5
turn -90
frac 1/2^0.5
```

図 2.1a サンプル 1



図 2.1b 次数 1

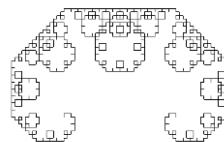


図 2.1c 次数 12

最初の turn 45 は反時計回りで角度  $45^\circ$  、frac  $1/2^{0.5}$  はスケール  $1/\sqrt{2}$  の相似図形を現在の角度で貼り付けることを意味する。メニューに「開始角度」テキストボックスがあるが、これを  $90^\circ$  にすると、上向きに描画が始まり、図 2.1b と図 2.1c が反時計回りに  $90^\circ$  回転した図形となる。

次の例は有名なコッホ曲線である。図 2.2 にプログラムと描画サンプルを示す。

```
frac 1/3
turn 60
frac 1/3
turn -120
frac 1/3
turn 60
frac 1/3
```

図 2.2a サンプル 2



図 2.2b 次数 1



図 2.2c 次数 7

これは説明の必要がないであろう。同じような例が続いたので、次は分岐がある場合の例である。

```
go 0.3
separate
turn 45
frac 0.6
return
turn -45
frac 0.6
```

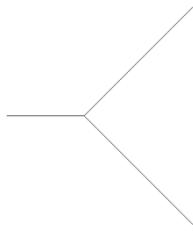


図 2.3a サンプル 3

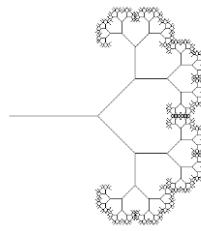


図 2.3c 次数 9

図 2.3b 次数 1

goto 0.3 は 0.3 のスケールで線を引く命令である。次の separate はこの時のタートルの位置と向きを記憶する命令である。分岐する場合によく使われるので separate にした。5 行目にある return は、前の separate で記憶した状態に戻す命令である。残念ながら現在は separate ~ return のネスト構造には対応していない。

次は、4 角形を用いたサンプルである。

```
cfix
polygon4 1
turn 20
fracp 0.7
```



図 2.4a サンプル 4

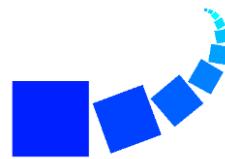


図 2.4c 次数 10

図 2.4b 次数 1

cfix は描画の際に図形の中心を基準にすることを意味する。これがなければ左下が基準である。次の polygon4 1 はスケール 1 (1 辺の長さが 1) の四角形を描画する命令である。描画が終わった段階でタートルは図形の中心から、その時のタートルの向きに図形のスケールだけ進んでいる。最後の fracp 0.7 は通常の frac 0.7 とは異なる。frac の場合は次数 1 の場合にも 2 つの四角形が描かれてしまう。fracp はこれを止めるためのフラクタルの予定地のようなものである。フラクタルは置くが、次の次数から表示する命令である。ちなみに色パターンは「Aqua」を利用している。

次の例は 3 角形で構成されるよく知られたシェルピンスキーのギャスケットである。

```

separate
fract3 0.5
fract3 0.5
return
turn 60
jump 0.5
turn -60
fract3 0.5

```

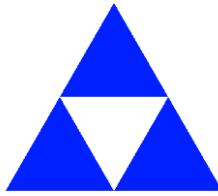


図 2.5a サンプル 5

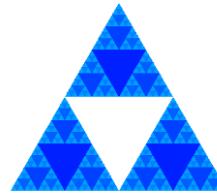


図 2.5b 次数 1

図 2.5c 次数 6

ここでは `fract3 0.5` があるが、これは 3 角形も書いてフラクタルも貼り付けるという意味である。このような書式を利用すると、次数 1 で 3 角形が 3 つになり、それに合わせて各次数で 3 角形が増える。また `jump 0.5` は線を引かずにタートルを飛ばす命令である。

次数 1 で 3 角形を 1 つにしたい場合は、少し長くなるが、図 2.6 のようなプログラムにする。4 行目の `separate` はなくてもよい。

```

separate
polygon3 1
return
separate
fracp 0.5
fracp 0.5
return
turn 60
jump 0.5
turn -60
fracp 0.5

```

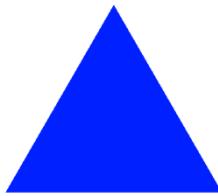


図 2.6a サンプル 6

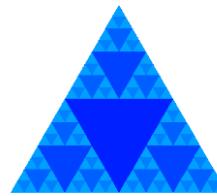


図 2.6b 次数 1

図 2.6c 次数 6

図 2.7 は円を使ったサンプルである。`cfract 1/3` は比率  $1/3$  の円を書いて、フラクタルを貼り付ける命令である。この場合比率は直径となる。`cfix` は図形の中心を基準にすることを意味する。プログラムは途中から省略している。

```

cfix
separate
cfract 1/3
return
jump 1/3
cfract 1/3
return
turn 60
jump 1/3
cfract 1/3
return
...

```

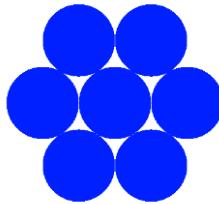


図 2.7a サンプル 7

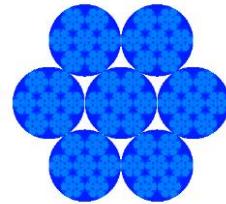


図 2.7b 次数 1

図 2.7c 次数 4

図 2.8 は色に乱数を使った例である。

```

drep
cfix
separate
color int(16*rnd)
circle 1
return
color 14
pentagon 1
return
fractp 0.81

```

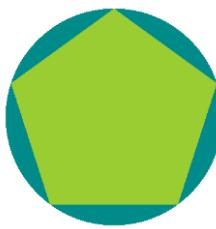


図 2.8a サンプル 8

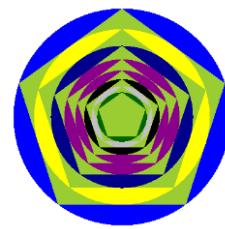


図 2.8b 次数 1

図 2.8c 次数 8

最初の `drep` は正多角形の比率を外接円の直径にする宣言である。これがなければ比率は 1 辺の長さとなる。図形の色は色パターンと描画の次数によって自動的に決まるが、自分で指定することもできる。`color` 命令は直後の図形描画の色を指定する。ここでは円の色指定で、パラメータの中に乱数 `rnd` を使っている。

以上いくつかサンプルを示したが、マクロの簡単なまとめを表 2.1 に示す。

表 2.1 フラクタルビューア 2D の再帰処理書式

CFIX	正多角形の起点を中心に設定（お勧め）【宣言】
DREP	多角形の比率を外接円の直径表示に設定【宣言】
CONNECT [色番号]	フラクタル同士を色番号の線でつなぐ 【宣言】

FRACP	比率	描画をしない再帰処理
FRACT3	比率	常に正三角形を描画する再帰処理
POLYGON3	比率	正三角形の描画
TRIANGLE	比率	正三角形の描画
FRACT4	比率	常に正方形を描画する再帰処理
POLYGON4	比率	正方形の描画
SQUARE	比率	正方形の描画
FRACT5	比率	常に正五角形を描画する再帰処理
POLYGON5	比率	正五角形の描画
PENTAGON	比率	正五角形の描画
FRACT6	比率	常に正六角形を描画する再帰処理
POLYGON6	比率	正六角形の描画
HEXAGON	比率	正六角形の描画
CFRACT	比率	常に円を描画する再帰処理
CIRCLE	比率	円の描画
FRAC	比率	最後の次数だけ直線描画の再帰処理
FRACT	比率	常に直線を描画する再帰処理
FRACC		強制的な終点へ連結する再帰処理
GO	比率	常に直線描画（標準/GOT）
GOF	比率	最後の次数だけの直線描画
GOC		強制的な終点へ連結する直線描画
TURN	角度	進行方向の角度変化[度]
REVERSE		裏返し
SEPARATE		分岐の開始点・状態の保存（ネスト構造はまだ未対応）
RETURN		分岐への状態の戻り
RESET		進行方向・回転角の初期値再設定

注) 多角形の場合、比率は1辺の長さで表す。

(DREP があるときは外接円の直径)

注) 円の場合、比率は直径で表す。

## 2. フラクタルビューア 3D

3次元空間へのフラクタルの描画は、反復関数による点の描画と再帰的手法による直線の描画で行われる。図1にフラクタルビューア 3Dのメニュー画面を示す。

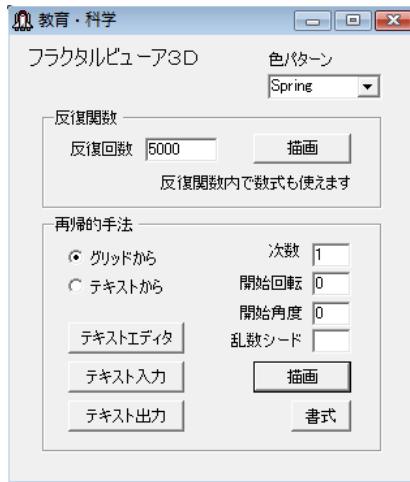


図1 フラクタルビューア 3D メニュー

ここでは反復関数による方法と再帰的手法を順番に説明する。

### 2.1 反復関数による方法

2次元の場合 $r$ 種類の反復関数は以下の行列計算で表される。

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} a_{11}^\alpha & a_{12}^\alpha \\ a_{21}^\alpha & a_{22}^\alpha \end{pmatrix} \begin{pmatrix} x_{n-1} \\ y_{n-1} \end{pmatrix} + \begin{pmatrix} b_1^\alpha \\ b_2^\alpha \end{pmatrix} \quad (\alpha = 1, 2, \dots, r)$$

これは複素平面上における以下の演算と同等であり、古くからその性質が調べられてきた。

$$z_n = a^\alpha z_{n-1} + b^\alpha \bar{z}_{n-1} + c^\alpha \quad (\alpha = 1, 2, \dots, r)$$

ここに

$$z_n = x_n + iy_n, \quad z_{n-1} = x_{n-1} + iy_{n-1}, \quad \bar{z}_{n-1} = x_{n-1} + iy_{n-1},$$

$$a^\alpha = (a_{11}^\alpha + a_{22}^\alpha)/2 + i(a_{21}^\alpha - a_{12}^\alpha)/2, \quad b^\alpha = (a_{11}^\alpha - a_{22}^\alpha)/2 + i(a_{12}^\alpha + a_{21}^\alpha)/2,$$

$$c^\alpha = b_1^\alpha + ib_2^\alpha.$$

これを3次元に拡張すると、反復関数は以下のように表される。

$$\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} a_{11}^\alpha & a_{12}^\alpha & a_{13}^\alpha \\ a_{21}^\alpha & a_{22}^\alpha & a_{23}^\alpha \\ a_{31}^\alpha & a_{32}^\alpha & a_{33}^\alpha \end{pmatrix} \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix} + \begin{pmatrix} b_1^\alpha \\ b_2^\alpha \\ b_3^\alpha \end{pmatrix} \quad (\alpha = 1, 2, \dots, r)$$

この $r$ 種類の反復関数から、確率的に1つ選んで計算を実行し、それを繰り返して点を打って行く。

以下具体的に例を示しながら結果を見て行こう。図1.1に非常に有名なC曲線のデータと実行例を

示す。実行例は 10000 点を打ったものである。描画結果は紙面（スクリーン面）上方向が  $z$  軸正の方向、紙面右方向が  $x$  軸正の方向である。

	確率	$a_1$	$a_2$	$a_3$	$a_0$
▶ 初期値		1	0	1	
反復関数1	0.5	0.5	0	-0.5	0
		0	0	0	0
		0.5	0	0.5	0
反復関数2	0.5	0.5	0	0.5	0.5
		0	0	0	0
		-0.5	0	0.5	0.5

図 1.1a C 曲線データ



図 1.1b 実行結果

データでは、1 行目に初期値として  $z_0$  の値、1 列目に反復関数を選択する確率、2 行 2 列目以降が反復関数の係数行列の値である。同様にして、コッホ曲線の例を図 1.2 に示す。

	確率	$a_1$	$a_2$	$a_3$	$a_0$
▶ 初期値		1	0	0	
反復関数1	0.5	1/2	0	3^0.5/6	0
		0	0	0	0
		3^0.5/6	0	-1/2	0
反復関数2	0.5	1/2	0	-3^0.5/6	1/2
		0	0	0	0
		-3^0.5/6	0	-1/2	3^0.5/6

図 1.2a コッホ曲線データ

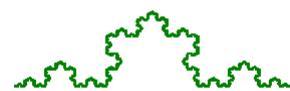


図 1.2b 実行結果

図 1.2 ではデータに数式を用いている。以上 2 つは 2 次元の例であったが、図 1.3 に 3 次元の例を示す。

	確率	$a_1$	$a_2$	$a_3$	$a_0$
▶ 初期値		1	1	1	
反復関数1	0.25	0.5	0	0	0
		0	0.5	0	0
		0	0	0.5	0
反復関数2	0.25	0.5	0	0	60
		0	0.5	0	0
		0	0	0.5	0
反復関数3	0.25	0.5	0	0	30
		0	0.5	0	60
		0	0	0.5	0
反復関数4	0.25	0.5	0	0	30
		0	0.5	0	30
		0	0	0.5	60

図 1.3a 3 次元ギャスケットデータ



図 1.3b 実行結果

これらの他にも文献などで紹介されている例を図 1.4、図 1.5 にあげておこう。

	確率	a1	a2	a3	a0
▶ 初期値		0	1	1	
反復関数1	0.1	0.05	0	0	0
		0	1	0	0
		0	0	0.6	0
反復関数2	0.2	0.05	0	0	0
		0	1	0	0
		0	0	-0.5	1
反復関数3	0.2	0.46	0	-0.32	0
		0	1	0	0
		0.39	0	0.5	0.6
反復関数4	0.2	0.47	0	-0.15	0
		0	1	0	0
		0.17	0	0.42	1.1
反復関数5	0.2	0.43	0	0.28	0
		0	1	0	0
		-0.25	0	0.45	1
反復関数6	0.1	0.42	0	0.26	0
		0	1	0	0
		-0.35	0	0.31	0.7

図 1.4a 木のようなフラクタル



図 1.4b 実行結果

	確率	a1	a2	a3	a0
▶ 初期値		0.5	0	0.5	
反復関数1	0.73	0.856	0	0.0414	0.007
		0	1	0	0
		-0.0205	0	0.858	0.147
反復関数2	0.13	0.244	0	-0.385	0.393
		0	1	0	0
		0.176	0	0.224	0.102
反復関数3	0.13	-0.144	0	0.039	0.527
		0	1	0	0
		0.181	0	0.259	-0.014
反復関数4	0.01	0	0	0	0.496
		0	1	0	0
		0.355	0	0.216	0.05

図 1.5a 葉のようなフラクタル



図 1.5b 実行結果

## 2.2 再帰的方法

再帰的方法はフラクタルビューア 2D のところでも述べたが、2 次元でタートルは平面上を移動するので左右に向きをえるとき `turn` 命令を用いるが、3 次元ではタートルの背中方向（初期値は y 軸負の方向）に法線ベクトルを考え、この向きを `rotate` 命令で変更する。ここで反復関数のときと同様に紙面（スクリーン面）上方向が z 軸正の方向、紙面右方向が x 軸正の方向である。しかし考えにくい場合は、タートルを  $90^\circ$  ひねり、背中を z 軸方向に向け、法線ベクトルの方向をタートルの右脇方向と考え、利用者はタートルに乗った状態をイメージすればよい。これで、`rotate` はタートルの回転（右ねじ方向）、`turn` は上下方向への向き変えとなる。以後この考え方に基づいて解説する。

この方法を用いた 3 次元フラクタルの例を図 2.1 に示す。

```

go 0.4
separate
turn -30
frac 0.6
return
rotate 60
turn 30
frac 0.6
return
rotate -60
turn 30
frac 0.6
return

```

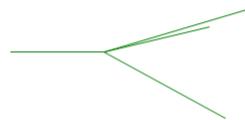


図 2.1a サンプル 1

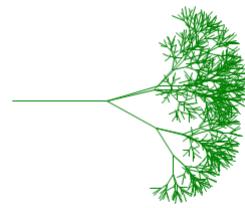


図 2.1b 次数 1

図 2.1c 次数 6

初期状態で進行方向は右横（x 軸方向 0 から 1 へ）で、タートルの右脇に当たる法線ベクトルは、y 軸負の方向である。紙面は x-z 平面で、タートルを右横から見ている状態になる。go 0.4 は 0.4 だけ進んで線を描く命令である。separate はその状態を記憶する。turn -30 で  $30^\circ$  下を向き、frac 0.6 で、0.6 倍に縮小したフラクタルを貼り付ける。return で記憶した位置に戻り、rotate 60 で  $60^\circ$  右ねじの方向に回転、turn 30 で  $30^\circ$  上を向いて、0.6 倍に縮小したフラクタルを貼り付ける、等々である。

次に図 2.2 にフラクタルを貼り付ける際に直線を表示しない frACP を用いた例を示す。

```

separate
rotate 90
go 1
turn 90
go 1
turn 90
go 1
turn 90
go 1
return
jump 0.5
turn -90
jump 0.25
turn 90
rotate 90
turn 45
rotate -90
frACP 0.7071

```



図 2.2a サンプル 1

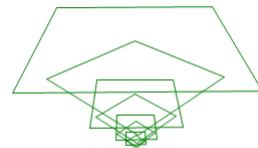


図 2.2b 次数 1

図 2.2c 次数 8

今回のサンプルは真横からだと見にくいので、少し傾けて表示してある。最初に `rotate 90` することによって、描画の方向が x-y 平面上になり、正方形を描いている。始点まで戻ったら、最初の辺の中央にジャンプし、 $90^\circ$  傾けて 0.25 ジャンプする。元に戻して  $45^\circ$  傾けて 0.7071 倍のフラクタルを表示せずに貼り付ける。

最後にマクロは付けないが、いくつかの例を示しておく。次数はいずれも 5 である。図 2.3 は図 1.3 で描いたギャスケットを再帰的手法で描いたものである。描画の要素は直線になっている。図 2.4 はフラクタル同士をつなぐ `connect` 命令を利用したヒルベルト曲線である。図 2.5 は同じフラクタルを複数同時に表示するために作られた `copy` 命令を使ったチリのような図形である。

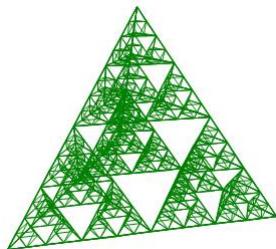


図 2.3 ギャスケット

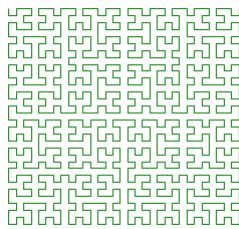


図 2.4 ヒルベルト曲線

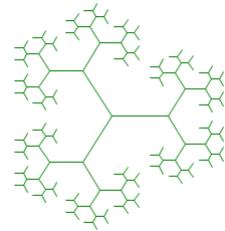


図 2.5 チリ

これらのマクロはホームページ上の `Sample.zip` 中の `フラクタル 3D (再帰).txt` に含まれている。

以上いくつかサンプルを示したが、マクロの簡単なまとめを表 2.1 に示す。

表 2.1 フラクタルビューア 3D の再帰処理書式

CONNECT	フラクタル同士を線でつなぐ 【宣言】 例：ヒルベルト曲線	
FRAC	比率	最後の次数だけ直線描画の再帰処理 (標準/FRACF)
FRACP	比率	直線描画をしない再帰処理
FRACT	比率	常に直線描画する再帰処理
GO	比率	常に直線描画 (標準/GOT)
GOF	比率	最後の次数だけの直線描画
TURN	角度	進行方向の角度変化 [度])
ROTATE	角度	進行方向の回転 [度] (/ROUND)
FRACC		強制的な終点へ連結する再帰処理
GOC		強制的な終点へ連結する直線描画

SEPARATE	分岐の開始点・状態の保存（ネスト構造はまだ未対応）
RETURN	分岐への状態の戻り
RESET	進行方向・回転角の初期値再設定
START	進行方向・回転角・位置の初期値再設定
COPY      比率	同じフラクタルの描画（2つ目以降）

我々のフラクタルビューアでは、2次元は通常のビットマップ画像として、3次元は3 Dビューアを用いた空間データとして出力した。3次元では動きを重視するために、描画要素数 10000 程度までが望ましい。

フラクタルビューア 2 D のマクロに比べ、フラクタルビューア 3 D のマクロは単純である。描画も直線を集めたもので、面の概念もない。そのため四角形 1 つにも 10 行近くの命令が必要である。我々は、単純な多角形を描画要素に加えたいと考えるが、面にはその法線方向のデータが必要となり、拡張の仕様が固まっていない。これは今後の課題である。

我々のプログラムはフラクタルアートを作成するようなものではなく、基本的に原理を学ぶためのツールである。それゆえ作る画像は、完全な自己相似形であり、それらを組み合わせて表示することや視覚的な効果を加えることは殆ど考えていない。ただ、3 D のフラクタルについては「Copy 比率」命令を加えて、大きさを変えた同じフラクタルを組み合わせることができるようしている。

### 3. カオスビューア（未完成）

フラクタルと関係の深いカオスの画像を見るためのプログラムがカオスビューアである。ファイル [分析－教育・科学他－カオス・フラクタル－カオスビューア] を選択して表示されるカオスビューアの画面を図 1 に示す。

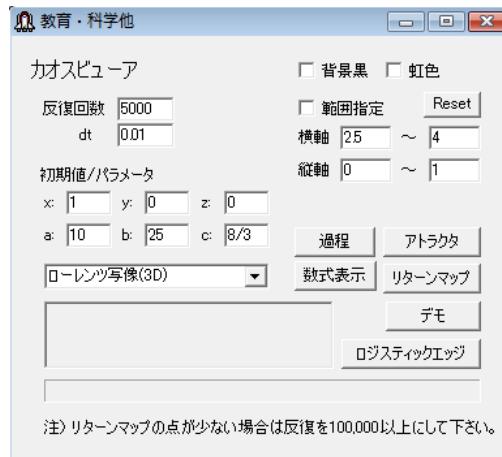


図 1 カオスビューア画面

選択できるカオスは、ローレンツ写像（3D）、レスラー写像（3D）、エノン写像（2D）、グモウスキー・ミラ写像（2D）、ロジスティック写像（1D）である。順次これらの写像とその性質などを説明する。これらの写像で、使えるボタンと使えないボタンがあるので注意してもらいたい。

#### ローレンツ写像（3 次元）

ローレンツ写像は、流体力学の方程式から導出された方程式で、その特別な解は乱流を表すカオスとして知られている。それは以下の式で与えられる。

$$x' = -a(x - y)$$

$$y' = -xz + bx - y$$

$$z' = xy - cz$$

ここで、

$a = 10, b = 15, c = 8/3$  のとき、ベルナール対流、

$a = 10, b = 25, c = 8/3$  のとき、ローレンツ・アトラクタ（カオス）

と呼ばれる。

初期値として  $x = 1, y = 0, z = 0$  、パラメータとして上の数値を設定し、「アトラクタ」ボタンをクリックして得られるベルナール対流とローレンツ・アトラクタを図 2 と図 3 に示す。

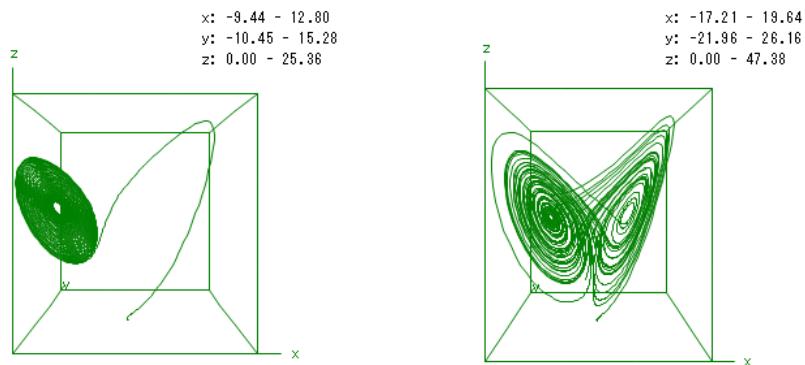


図 2 ベルナール対流

図 3 ローレンツ・アトラクタ

これらは3Dビューアの機能を用いて自由な角度と距離で見ることができる。これらの描画過程は、「過程」ボタンをクリックするとアニメーションで表示される。例えば視点を近づけて、角度を変えると図4のようなものが見える。図中では球状のものが線を描いて行っている。

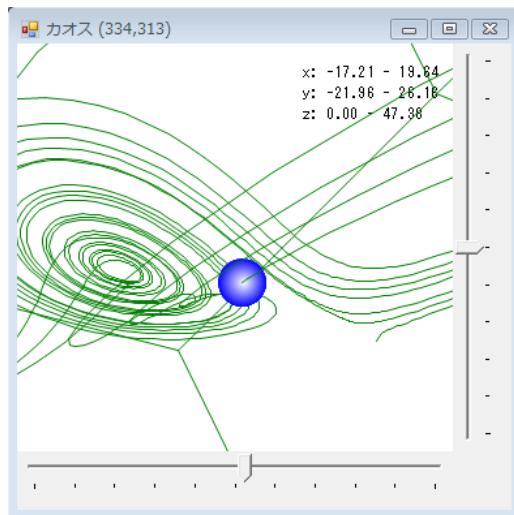


図 4 ローレンツ・アトラクタ描画過程

「数式表示」ボタンをクリックすると、それぞれの写像に対応した数式が、図4のように表示される。

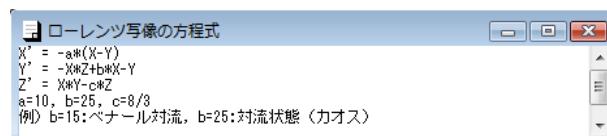


図 4 数式表示

カオスの描画過程のある部分を取り出してカオスの特徴を描きだすのがリターンマップである。例えばローレンツ・アトラクタでは、 $z$  方向で極値を取る際の  $z$  の値を取り出して、その  $z$  値を縦軸に、前回の  $z$  値を横軸にプロットして行くと図 5 のような図が得られる。これは反復回数を 10 万回にして「リターンマップ」ボタンをクリックして得られた図である。

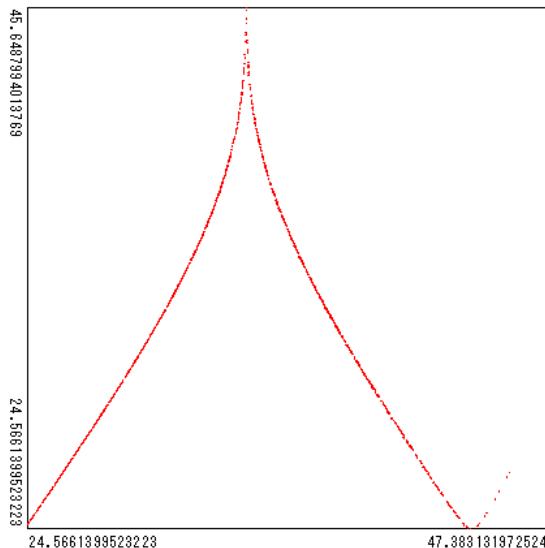


図 5 ローレンツ写像のリターンマップ

連続的な時間経過でカオスを生じるのは、3 次元以上であることが知られている。

### レスラー写像 (3 次元)

レスラー写像は以下の方程式で与えられる。

$$x' = -(y + z)$$

$$y' = x + ay$$

$$z' = b - cz + xz$$

$$a = 0.15, b = 0.2, c = 10$$

初期値として  $x = 1, y = 0, z = 0$  、パラメータとして上の数値を設定し、「アトラクタ」ボタンをクリックすると図 6 のレスラー・アトラクタを得る。

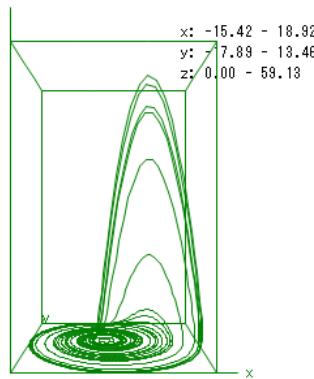


図 6 レスラー・アトラクタ

反復回数を 20 万回にしたレスラー写像の「リターンマップ」は図 7 のようになる。

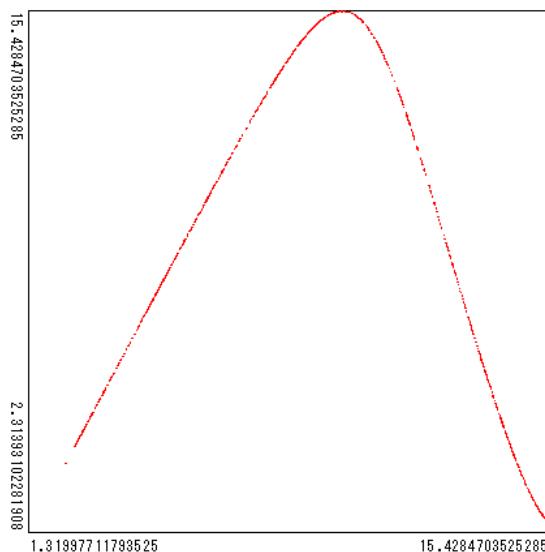


図 7 レスラー写像のリターンマップ

これは、アトラクタが  $y=0, x<0$  の平面を切る際の原点からの距離  $r$  を取り出し、その  $r$  値を縦軸に、前回の  $r$  値を横軸にプロットして行くと得られる。

### エノン写像 (2 次元)

エノン写像は以下の方程式で与えられる。

$$x_n = 1 - ax_{n-1}^2 + y_{n-1}$$

$$y_n = bx_{n-1}$$

$$a = 1.5, b = 0.25$$

初期値として  $x_0 = 1, y_0 = 0$  、パラメータとして上の数値を設定し、「アトラクタ」ボタンをクリックすると図 8 のエノン・アトラクタを得る。

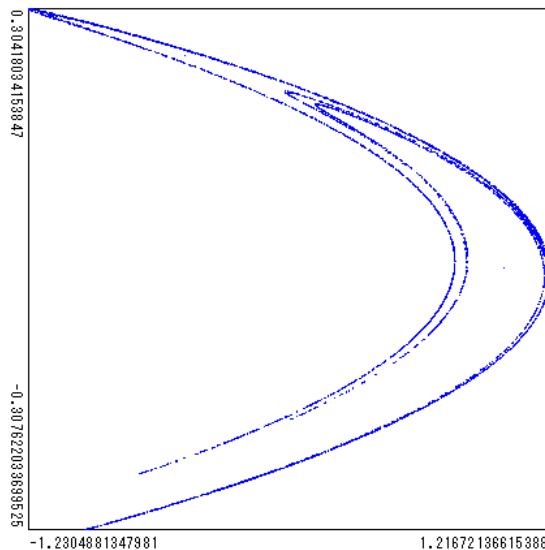


図 8 エノン・アトラクター

### グモウスキー・ミラ写像

グモウスキー・ミラ写像は以下の方程式で与えられる。

$$x_n = y_{n-1} + a(1 - b y_{n-1}^2) y_{n-1} + c x_{n-1} + 2(1 - c) x_{n-1}^2 / (1 + x_{n-1}^2)$$

$$y_n = -x_{n-1} + 2(1 - c) x_{n-1}^2 / (1 + x_{n-1}^2)$$

$$a = 0.008, b = 0.05, c = -0.8 \quad (-1 < c < 1)$$

これはパラメータの値によって大きく形を変えるアトラクターを持つ。

初期値として  $x_0 = 0.1, y_0 = 0$  、パラメータとして  $a, b$  は上の値、 $c = -0.8, 0, 0.8$  とした場合のアトラクタの変化を図 9 に示す。

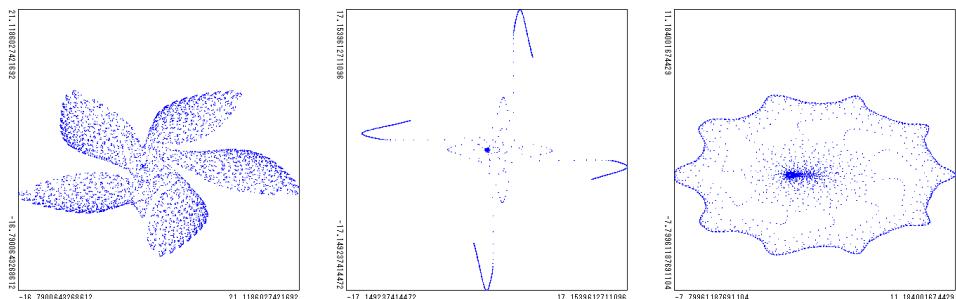


図 9 グモウスキー・ミラ・アトラクタ

「デモ」ボタンをクリックすると、パラメータ  $c$  の値を -1 から 1 まで 0.01 ずつ変化させたアトラクターの変化を表示するデモが表示される。

### ロジスティック写像

最も古くから知られているカオスで、以下の式で与えられる。

$$x_n = ax_{n-1} - bx_{n-1}^2$$

特に  $a = b$  がよく利用される。アトラクタは、1 次元のため示されないが、 $a = b = 3.78$  のときのリターンマップは図 10 のようになる。

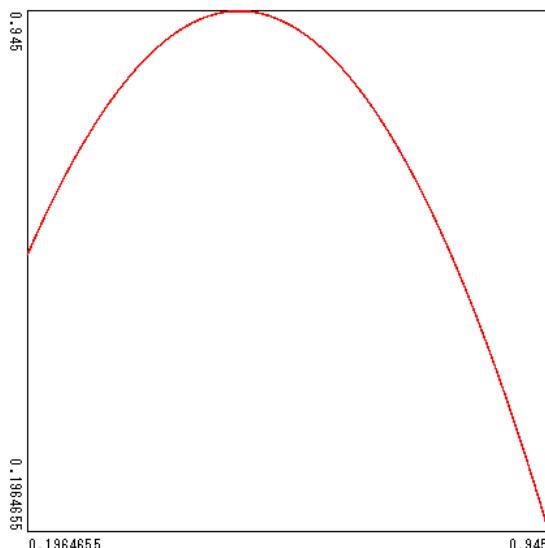


図 10 ロジスティック写像のリターンマップ

これは、 $x_n$  の値を y 軸に、 $x_{n-1}$  の値を x 軸に取って描いたものである。

この写像は、パラメータの値によって、複数の収束値（振動する点に収束）を持ったり、ランダムに散らばったりするが、その振動値の数がパラメータの値によって大きく変わるものである。特に無数に散らばる場合がカオスである。横軸に取ったパラメータ  $a (= b)$  の値によって、最終的に  $x_n$  が振動する様子が、「ロジスティックエッジ」ボタンをクリックすることによって、図 11 に示される。この図は、それぞれのパラメータ値について、 $x_0 = 0.5$  を始点として、繰返し 501 回目から 2500 回目までをプロットしている。

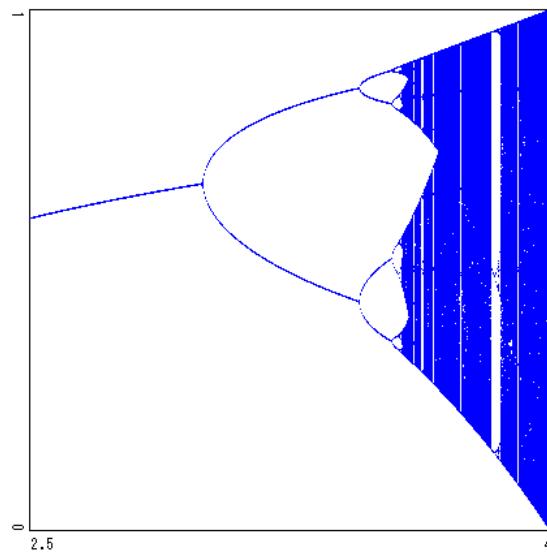


図 11 ロジスティックエッジ

横軸がパラメータ  $a (= b)$  の値で、2.5 から 4 まで変化させている。それに応じて収束点の数が、1 つから 2 つになり、4 つ、8 つと増えて行っていることが分かる。これはさらに増えて行き、無限に振動するカオス構造になる。これはフラクタル構造となっており、例えば図 12a の四角の部分を拡大すると図 12b のようになり、さらに、図 12b の四角の部分を拡大すると図 12c のようになる。

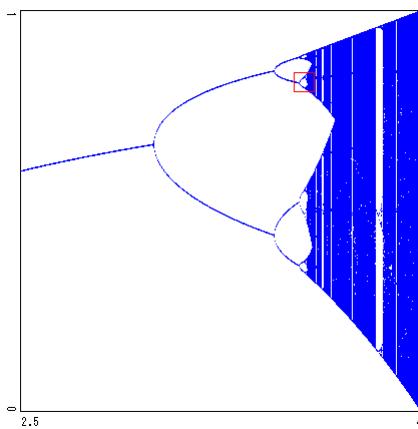


図 12a ロジスティックエッジ

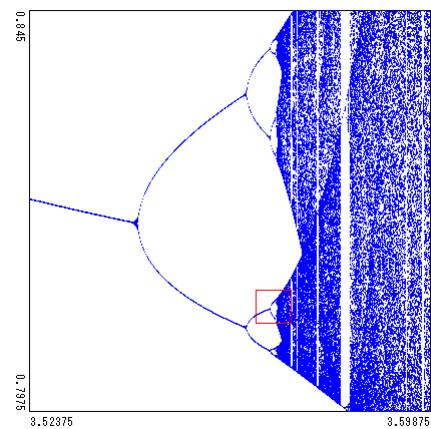


図 12b ロジスティックエッジ (拡大 1)

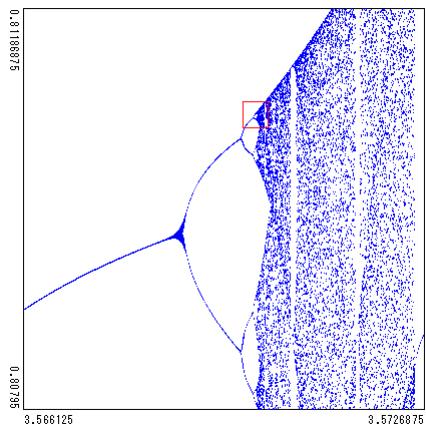


図 12a ロジスティックエッジ (拡大 2)

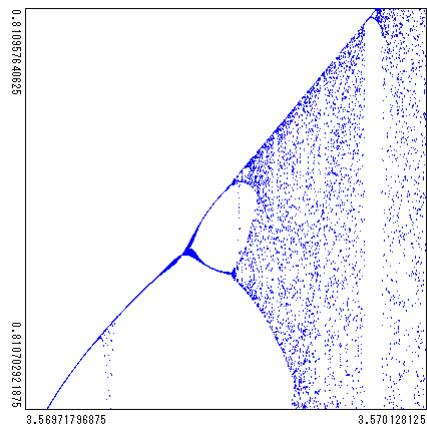


図 12b ロジスティックエッジ (拡大 3)

最後の方は計算の精度が落ちているが、フラクタル構造になっていることは分かる。

#### 4. 3Dモデルビューア（未完成）

これは3Dで与えられたデータから、立体図形を描画する基本的なプログラムである。しかし、現在は描画ツールがないため、テスト用のツールとなっている。メニュー【分析一科学・教育一お楽しみ3D-3Dモデルビューア】を選択すると図1の実行画面が表示される。



図1 3Dモデルビューア画面

既存のデータをグラフィックエディタに読み込み、「描画」ボタンをクリックすると、図2や図3のようなグラフィックモデルが表示される。

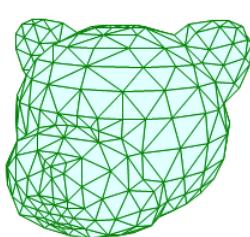


図2 サンプル1

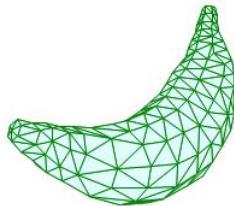


図3 サンプル2

単純に線で表わされた図形は、エディタに数値を直接入力して作ることもできる。

学習用にデータを作成する場合、面を三角形の集合として表示する場合がある。その際には、例えば「2変量関数グラフ」の中で描いた図形の四角形を三角形の集合に変換することもある。「□→△変換」ボタンをクリックすると、四角形のデータを三角形に変換して出力してくれる。図3aに元のデータからの図形、図3bに新しく変換した三角形で構成された図形を示す。

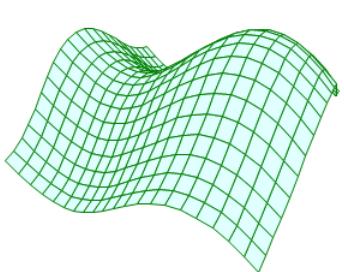


図 3a 四角形による図形

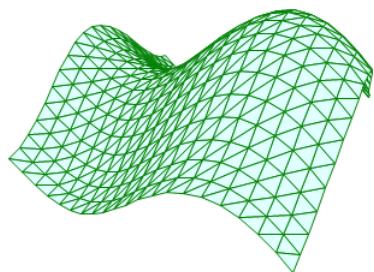


図 3b 三角形による図形

## 5. くるくるエディット

これは後に述べるおもしろグラフのデータを作るために作成した中心軸周りの回転図形を描くプログラムである。メニュー [分析－科学・教育－お楽しみ 3D－くるくるエディット] を選択すると図 1 の実行画面が表示される。

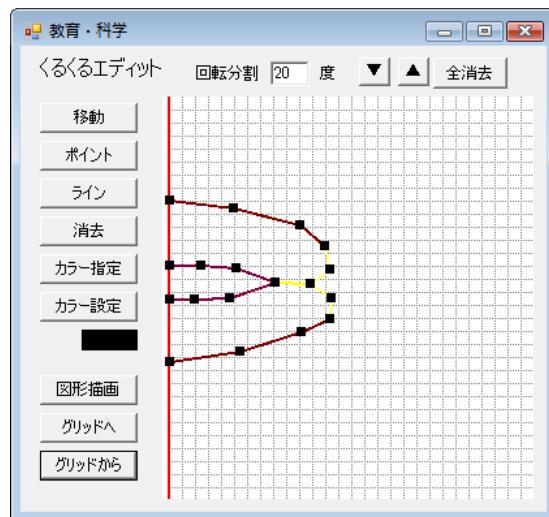


図 1 くるくるエディット画面

左側の「ポイント」ボタンを選択して画面上をクリックすると、画面にポイントが打たれ、「ライン」ボタンを選択してポイントからポイントにドラッグするとその間にラインが引かれる。「消去」ボタンを選択して、描画要素をクリックするとその要素が消える。「カラー設定」ボタンをクリックすると標準の色指定画面が開き、色を選択することができる。以後はラインを引くとその色で描かれる。「カラー指定」ボタンを選択して、ライン上をクリックするとすでに描かれているラインの色が選択された色に変わる。右上の「▼」「▲」ボタンは図形の位置を下や上に移動させる。

「図形描画」ボタンをクリックすると、左端の線を回転軸として、ラインの色を面の色とした図 2 のような回転図形が描かれる。「回転分割」の角度を小さく取って、描画すると図 3 のように細かく分割された図形になる。

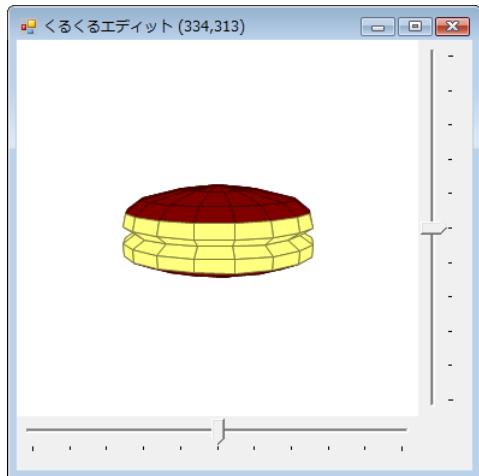


図 2 図形描画結果

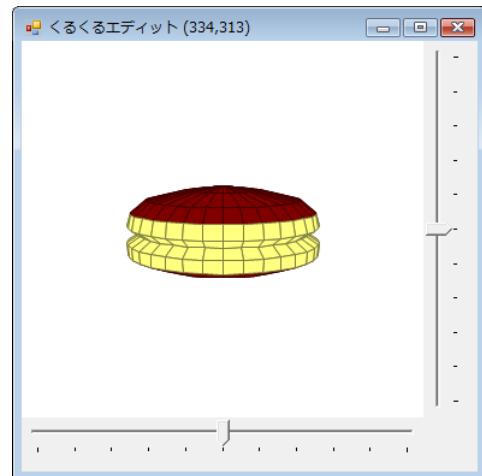


図 3 図形描画結果 2

これらの図形の色や表示法を変える方法は、ツール編の3Dビューアを参照してもらいたい。  
くるくるエディットのデータは「グリッドへ」ボタンでグリッドに移して保存する。その際、後の  
おもしろグラフで利用するために、ページに図形を表す「どら焼き」などの名前を付けておくとよい。

## 6. おもしろグラフ

おもしろグラフは統計グラフとしては殆ど役に立たないが、楽しめるグラフである。現在、種類は棒グラフ、折れ線グラフ、円（パイ）グラフに対応して、「どら焼き棒グラフ」、「ヘビ線グラフ」、「アップルパイグラフ」の3種類である。メニュー[分析－科学・教育－お楽しみ3D－おもしろグラフ]を選択すると図1の実行画面が表示される。

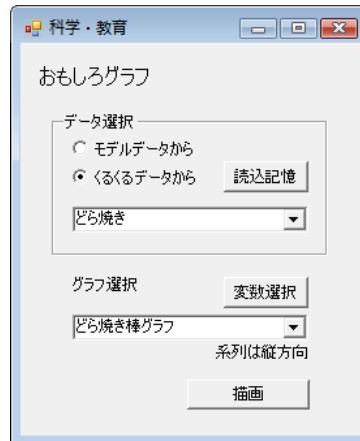


図1 おもしろグラフ実行画面

どら焼き棒グラフのグラフィックデータは、モデルデータからでもくるくるエディットで作られたデータからでもよい。また、アップルパイグラフのグラフィックデータはくるくるエディットのデータを利用する。ラジオボタンで利用するモデルの種類を選択し、データファイルを指定して、「読み込み」ボタンでデータを読み込む。その後は普通のグラフ描画プログラムと同じである。

まずグラフデータのファイルを開き、「変数選択」で利用する変数を選択する。グラフの種類を選択して、「描画」ボタンをクリックするとグラフが表示される。図2はどら焼き棒グラフのサンプルである。

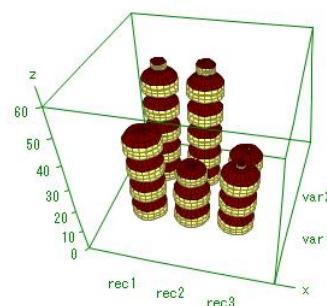


図 2 どら焼き棒グラフ結果

図 3a と図 3b はヘビ線グラフのサンプルである。

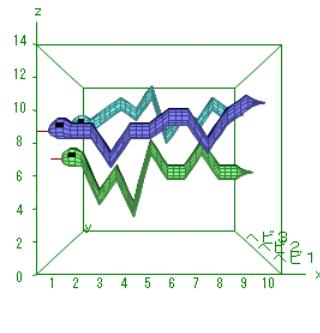


図 3a ヘビ線グラフ 1

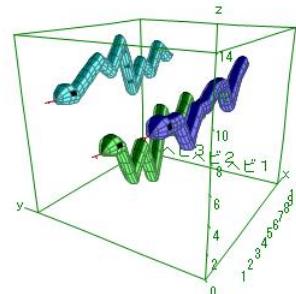


図 3b ヘビ線グラフ 2

図 4a と図 4b はアップルパイグラフのサンプルである。

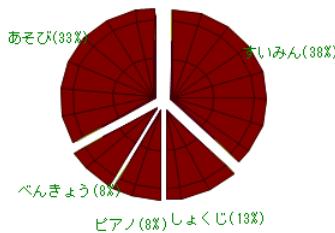


図 4a アップルパイグラフ 1

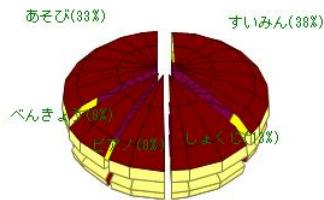


図 4b アップルパイグラフ 2

特にヘビ線グラフとアップルパイグラフでは、最初表示されるのは図 2a と図 3a であり、グラフのように見えるが、図形を回転させるとヘビや回転物体がよく見えてくるという、だまし絵のようになっている。

## 7. 3Dフォトメーカー

3Dフォトメーカーは、少し横にずらして撮影した写真を重ねて、アナグリフを作成するプログラムである。アナグリフは左目が赤、右目がシアンである。メニュー [分析一科学・教育一お楽しみ3D-3Dフォトメーカー] を選択すると図1の実行画面が表示される。

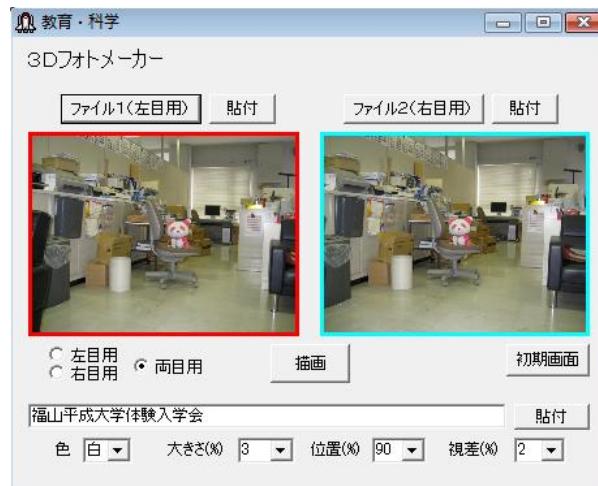


図1 3Dフォトメーカー実行画面

このメニューは何となく赤青（シアン）メガネをかけた顔をイメージしている。「ファイル1」と「ファイル2」に左目用と右目用のデータを読み込む。必要なら下のテキストボックスに文字列を書き込み、色や大きさ位置（画面上端からの割合を%で表わす）、視差などを入力する。デフォルトが適当な値になっているので調節する。左の両目用のラジオボタンを選択し、中央の「描画」ボタンをクリックすると図2のようなアナグリフが表示される。



図2 アナグリフ

左目用と右目用のラジオボタンを選択した画面をそれぞれ図 3a と図 3b に示す。



図 3a 左目用画像



図 3b 右目用画像

楽しんでもらえたらと思う。

## 8. 3Dバナーメーカー

3Dバナーメーカーは、クリップボード上の絵をアナグリフ化したり、立体図形に作り替えるプログラムである。アナグリフは左目が赤、右目がシアンである。メニュー【分析-科学・教育-お楽しみ3D-3Dバナーメーカー】を選択すると図1の実行画面が表示される。

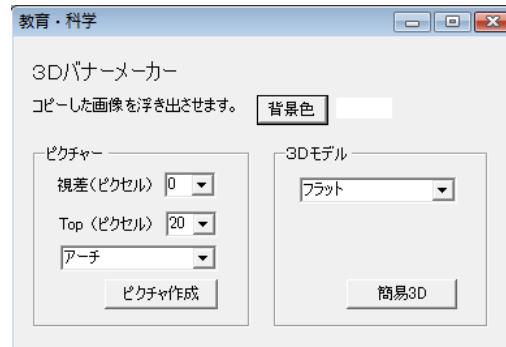


図1 3Dバナーメーカー実行画面

「ピクチャー」グループボックス内は、2次元のアナグリフを作るものである。視差（ベースとなる飛び出し）とTop（ベースと最も飛び出す部分の差）を決め、形状のコンボボックスから、すきなものを選んで「ピクチャ作成」ボタンをクリックする。形状には、フラット、アーチ、ウェーブ、マウンテン、がある。今作成中の画面をコピーして、マウンテンを選択し、描画した結果を図2に示す。

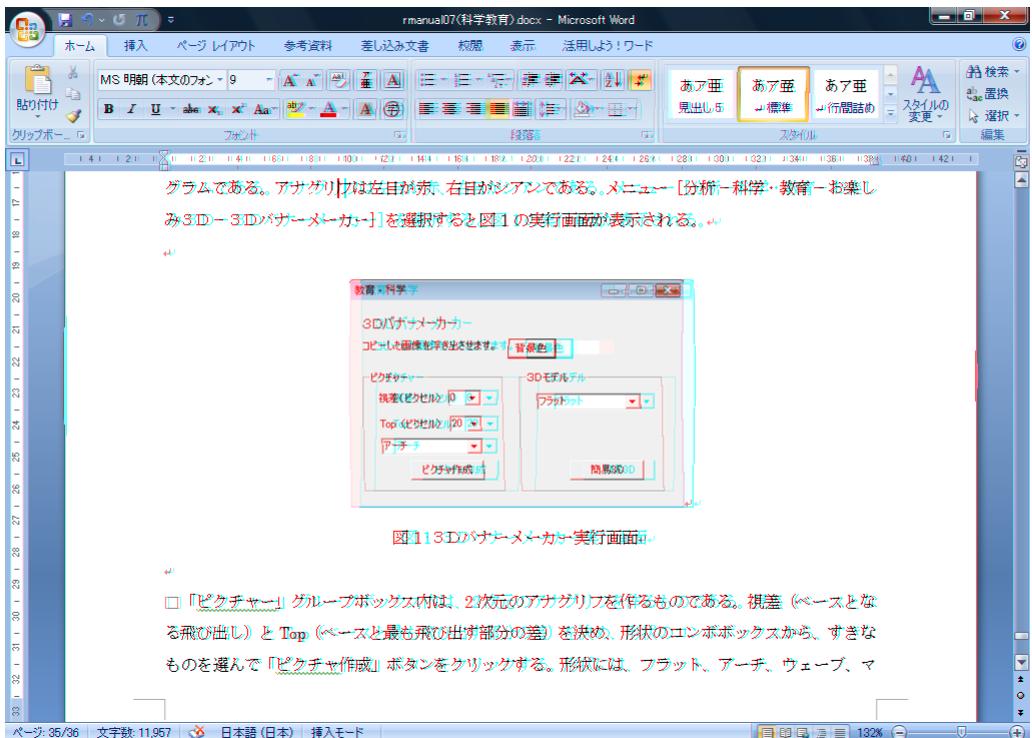


図1 3Dバナーメーカー実行画面

- 「ピクチャーチャンプ」ボックス内は、2次元のアアアリフを作るものである。視差（ベースとなる飛び出し）と Top（ベースと最も飛び出る部分の差）を決め、形状のコンボボックスから、すぎなものを選んで「ピクチャ作成」ボタンをクリックする。形状には、フラット、アーチ、ウェーブ、マ

図2 描画結果

赤青メガネで見ると、中央が盛り上がっている。

3Dモデルグループボックスは、画面のドットを図形に変えて表示するので、ドット数にして3万点位が動かす限界である。ただ、白地の線画はデータ数が少なくなるので使い易い。例えば図3のような絵をコピーして、フラットやリングで描画すると図4や図5のようになる。



図3 コピー元画像

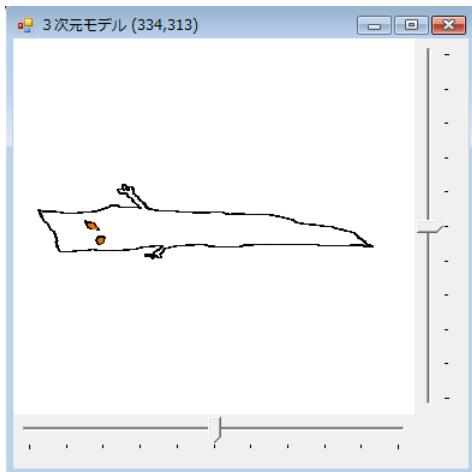


図4 フラット描画

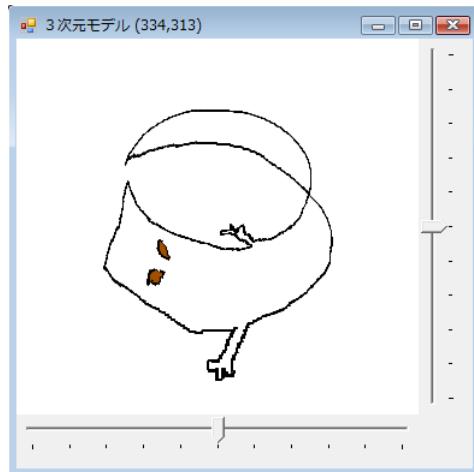


図5 リング描画

これらは、3Dビューアのアクション機能で、ひらひら飛んだり、くるくる回ったりする。

描画方法は、フラット、リング、ハーフリング、ボール、ハーフボール、メビウスがある。最後に、図6と図7にメビウスとボールのサンプル（いずれも元絵は小さい）を示しておく。



図6 メビウス



図7 ボール

楽しんでもらいたい。

## 9. 惑星シミュレーション

ここでは常微分方程式の数値解法の例として、惑星シミュレーションを取り上げる。メニュー〔数学－常微分方程式〕のところで扱った一般的な常微分方程式では、式の形が決まっていないため、記述された方程式を文字列として処理し計算する必要があり、計算の高速化は望めなかつたが、モデルが定まっている場合は、予めプログラム中に数式が組み込めるので、実行速度は格段に速くなる。惑星のシミュレーションについては物理学者からアマチュアのプログラマまで、多くの人が様々な目的や精度で実行しているが、我々のプログラムの目的は、基礎教育の授業で、物理の知識の殆どない学生に、1コマ分の夢のある話題を提供することである。

我々が扱う問題は、 $n$  個の星を考えて、以下で与えられる。

$$\frac{d^2\mathbf{r}_i}{dt^2} = - \sum_{j \neq i} Gm_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} \quad (i = 1, \dots, n)$$

ここに、 $m_i$  は  $i$  番目の星の質量である。

我々は利用者のイメージを考えて、太陽と地球の平均距離（1天文単位）を1、地球の速さを1とする単位系を採用する。そのため太陽の質量を $M$ とすると、 $v^2/r = GM/r^2$  より、 $GM = 1$  となり、方程式は以下のように書き換えられる。

$$\frac{d^2\mathbf{r}_i}{dt^2} = - \sum_{j \neq i} \frac{m_j}{M} \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

質量は比として表れるので、自由度が残るが、分かり易いように、地球質量を1とする。この単位系では地球の1年は $2\pi$ で表される。しかし、利用者には分かりにくいので、シミュレーション内ではこれを改めて1年と計算し直している。

この式にルンゲークッタ (Runge-Kutta) 法<sup>1)</sup> を適用して数値解を求めるが、そのために以下のよな連立1階常微分方程式に変形する。

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i$$

$$\frac{d\mathbf{v}_i}{dt} = - \sum_{j \neq i} \frac{m_j}{M} \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

この方程式の変数  $(\mathbf{r}_i, \mathbf{v}_i)$  に  $t = 0$  での初期条件を付けて、計算を実行する。

メニュー〔分析－教育・科学－物理シミュレーション－惑星シミュレーション〕を選択すると、図1のような実行メニューが表示される。

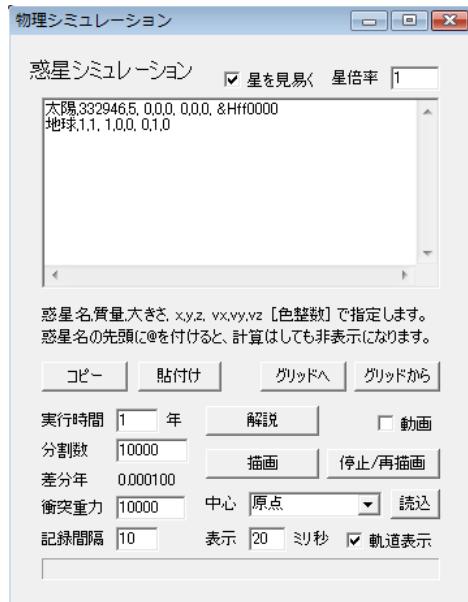


図 1 実行メニュー

実行メニューには、「星を見易く」表示するモードと実サイズで表示するモードがある。メニュー右上の「星を見易く」チェックボックスをチェックすると、星の大きさの単位が太陽と地球の平均距離である天文単位となり、星の半径は 0.02 天文単位×指定された星の大きさとなる。太陽の大きさを 5 と設定した場合、描画される半径は 0.1 天文単位であり、実際の大きさの 20 倍ほどの大きさになっている。地球の場合、データで大きさ 1 を指定したならば、実際の大きさの約 400 倍で描かれることがある。これ程の大きさで描かないと、球体の周りを球体が回っているようには見えない。これでもまだサイズが小さい場合は、右上の「星倍率」テキストボックスの数値を適当に大きくする。

星を実サイズで表示するモードは、「星を見易く」のチェックボックスのチェックを外して設定する。このモードでは、星の大きさは地球の大きさの倍数で指定する。例えば木星軌道から眺めると、太陽はかすかな点として表示されるが、惑星は全く見えない。拡大しても動きがある星は非常に捉えにくい。しかし後に示すように、殆ど見えにくい太陽の公転運動など、面白い結果も示すことができる。星を見易く表示するモードと同様、「星倍率」テキストボックスで、見易い大きさまで拡大することもできる。

データは、上部のデータ入力用のテキストボックスに記述する。記述方法は、その下にあるように、「惑星（恒星）名、質量、大きさ（半径）、x, y, z, vx, vy, vz [色整数]」の順番にカンマ区切りで入力する。ここに、質量は地球質量の倍数、大きさは先に述べたように、星の見易いモードか否かによって基準が異なるが、基準値の倍数で指定する。x, y, z は初期座標値、vx, vy, vz は初期速度を表す。色整数は特に指定する必要はないが、色を表す 10 進整数または&H を付けて 16 進表示で指定する。例

えば青色は&H0000ff、緑色は&H00ff00、赤色は&Hff0000で表わされる。惑星（恒星）データの単位系や星の大きさの記述方法、シミュレーションの設定については、「解説」ボタンをクリックすると説明が表示される。図2にその画面を示す。

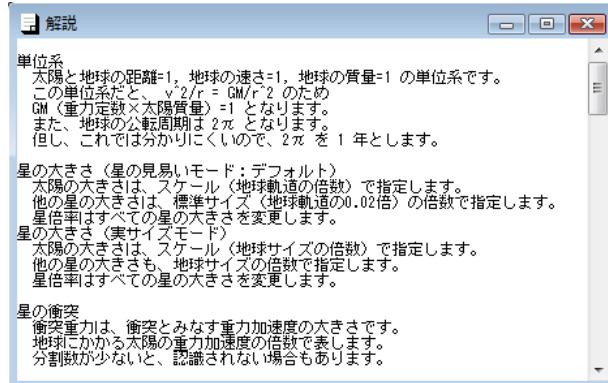


図2 解説画面

「実行年」テキストボックスはシミュレーションの実行年数（小数も可）を記入する。「分割数」テキストボックスは、実行年をいくつに分けてシミュレーションするかを表す。実行年÷分割数は時間の「差分年」 $\Delta t$ であり、この値が大きいとシミュレーションの精度が悪くなる。但し、重力加速度の大きさにもよるので、一定の基準はない。少なくとも惑星の周期を $T$ 年として、 $\Delta t < T/1000$ （太陽と地球の場合は $T = 1$ 年）とする必要がありそうである。このように考えると、公転半径を $r$ 、中心の星の質量を $m$ として、周期が $2\pi\sqrt{r^3/Gm}$ であるから、 $\Delta t \propto \sqrt{r^3/m}$ と考えてもよいであろう。パソコン画面上できれいな絵を描くには、描画間隔（計算間隔とは異なる）として経験上1/50周期以下であろうか。

シミュレーションを実行すると星同士の強い反発現象が見られることがある。これは強い重力で引き寄せられた星が加速し、1つの差分時間で相手の星をまたぎ越してしまうことに起因すると思われる。この現象の対処には強い重力加速度の状況では差分値を小さくする方法が考えられるが、このプログラムには導入されていない。通常強い重力下では星は衝突すると考えられるので、「衝突重力」テキストボックスに限界と考えられる値を入れておき、それよりも重力加速度が強い場合、星は衝突したものと判定する。強さの単位は、太陽が地球に及ぼしている重力加速度の倍数で指定する。デフォルトは10000（ほぼ太陽半径×2の距離での太陽重力）となっているが、実はこれが妥当かどうか分からぬ。また、時間差分の値によってはこの範囲をまたぎ越してしまうことも起こるので、注意が必要である。

シミュレーションの時間差分間隔でグラフィックデータを作成した場合、データ数が多すぎて、マウスで図を動かすことが困難になる。我々のプログラムで、現在の一般的なパソコン環境では、描画

要素 1 万以下に抑えるのが望ましい。そのため計算したデータの中から、一定の間隔で選んでデータを記録し、グラフィック表示させるようにしている。「記録間隔」テキストボックスは、元データの中からどれだけの間隔で記録させるのかを与える。デフォルトの 10 の場合は、10 個間隔で記録し、表示するという意味である。

「表示」テキストボックスにはグラフィックを何ミリ秒に1回表示するのかを記入する。グラフィック表示が追いつかない場合は、この通りにはならず、描画が遅れて行く。デフォルトの20ミリ秒は、College Analysisで他の描画にも採用しているスピードである。星をゆっくり表示させる方がよければ大きな値にする。最後に「軌道描画」チェックボックスは、星の動画で星の軌道を表示するか否かを決める。星の軌道を描くと、時間経過とともに描画要素数が多くなり、描画スピードが落ちて行く。その欠点をなくすために、軌道を描かないモードも設けた。このモードではあまり描画スピードが落ちることはない。実際の星のスピード（もちろん、時間的な縮尺は含まれるが）も再現している。

データはメニュー画面上部の大きなテキストボックスに、「惑星(恒星)名, 質量, 大きさ, x, y, z, vx, vy, vz [,色整数]」の順番にカンマ区切りで記入するが、「コピー」(範囲を選択して実行) や「貼付け」で、他のソフトとのデータの受け渡しができる。また、「グリッドへ」と「グリッドから」ボタンで、データをグリッドデータへ移して、何ページかまとめて保存することもできる。Samples.zip に含まれるファイル「惑星シミュレーション 1.txt」には、幾つかのサンプルが入っている。データの行の先頭に「#」を付けるとコメント行、「@」を付けると計算には使用するが、表示しないデータとなる。用途に応じて使ってもらいたい。「停止/再描画」ボタンはシミュレーション実行後、上で述べた「@」を付けて再表示させる場合や、一時停止や再実行する場合などに利用する。

以下、データの例を挙げながらグラフィック出力結果を紹介する。断らない限り「星を見易く」のモードで表す。分割数、実行時間は、見出しの右に括弧付きで示す。

例 1 太陽と地球（分割数 10000, 実行時間 1）

地球,1,1, 1,0,0, 0,1,0

これは地球が太陽の周りを回転するモデルである。我々の単位系で、地球の質量は 1、位置の初期値は  $x$  軸上として  $(1,0,0)$ 、速度の初期値は  $(0,1,0)$  である。地球の半径は、実サイズモードでは 1 (地球の大きさの基準値)、星が見易いモードでは適当な数値にする。ここでは見易いモードで地球のサイズ 1 (0.02 天文単位に相当)、太陽のサイズ 5 に設定している。図 3 に「動画」チェックボックスを外した場合と付けた場合の、「描画」ボタンで表示される画面の例を示す。動画には軌道を描画するモードと描画しないモードがある。動画は「停止/再描画」ボタンをクリックすると停止する。

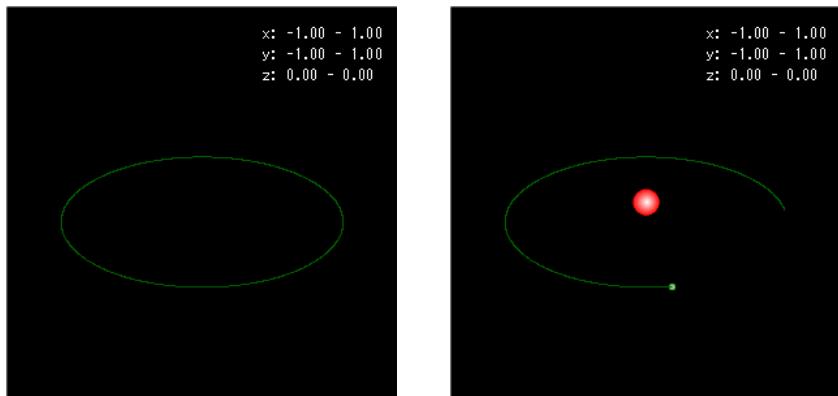


図 3 太陽と地球のシミュレーション結果

## 例 2 太陽と連星（分割数 10000, 実行時間 1）

太陽,332946.5, 0,0,0, 0,0,0, &Hff0000

p1,10000,1, 1,0,0, 0,0.7,0

p2,10000,1, 1.1,0,0, 0,1,0

これは太陽のまわりを地球質量の 10000 倍の惑星が連星として回転するシミュレーションである。

図 4 にシミュレーション結果と描画過程の動画画面を示す。

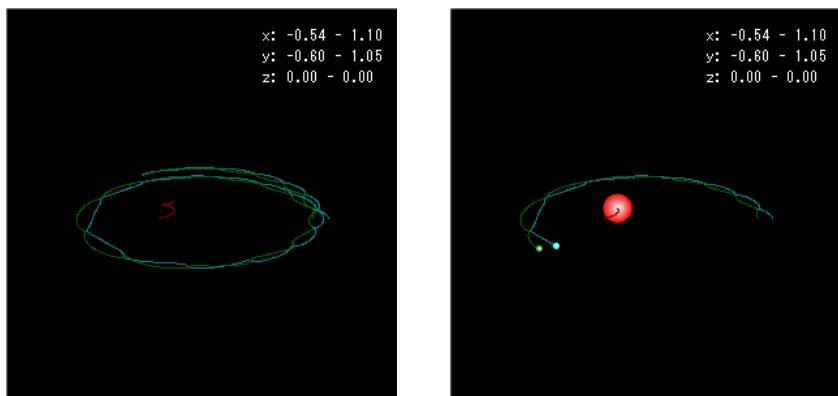


図 4 太陽と連星のシミュレーション結果

これを見ると、初期速度によって系全体が移動していることがわかる。系全体を静止させるためには、「中心」コンボボックスに「読込」ボタンで選択肢を読み込み、重心を中心とした図を「再描画」させるとよい。図 5 にその描画画面を示す。

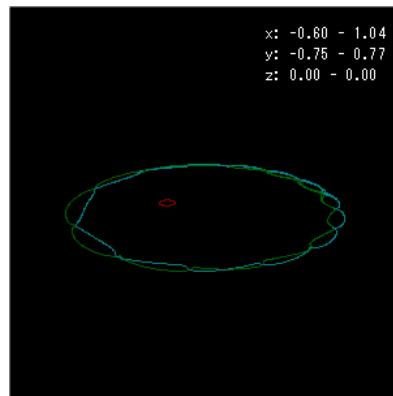


図 5 重心を中心とした描画

## 例 3 連星をなす恒星と 1 つの惑星 (分割数 10000, 実行時間 8)

```
Sun1,166473,2, 0.5,0,0, 0,0.5,0
```

```
Sun2,166473,2, -0.5,0,0, 0,-0.5,0
```

```
p1,1,1, 0,2,0, -0.8,0,0
```

これは太陽の半分の質量の 2 つの恒星が、公転半径 0.5 天文単位で連星をなしている周りを、地球質量の惑星が 2 倍の公転半径で回っている状況を表している。図 6 にシミュレーション結果を示す。星の大きさが分かりにくいので、「星倍率」 2 倍のサイズで描いている。

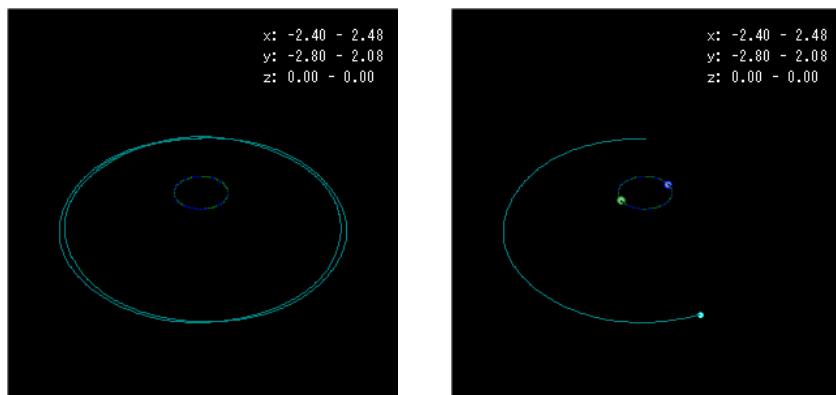


図 6 連星をなす恒星と 1 つの惑星

## 例 4 4 連星 (分割数 10000, 実行時間 1)

```
p1,100000,1, 0.5,-0.5,0, 0,0.5,0
```

```
p2,100000,1, 0.5,0.5,0, -0.5,0,0
```

```
p3,100000,1, -0.5,0.5,0, 0,-0.5,0
```

```
p4,100000,1, -0.5,-0.5,0, 0.5,0,0
```

これは4つの同じ質量の星が、全くタイミング良く連星をなしているモデルである。図7aにそのシミュレーション結果を示す。しかし、これは不安定な解で、例えばp1の初期位置をx方向に0.0001ずらすと図7bのように全く異なる軌道になる。

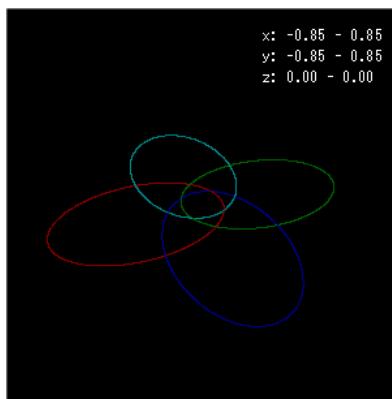


図 7a 不安定な4連星解

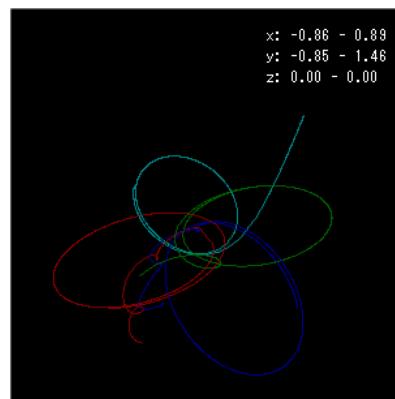


図 7b 4連星の崩壊

## 7つの星の衝突 (分割数 10000, 実行時間 1)

```
p1,100000,1, 0.5,-0.5,1, 0,0.5,0
```

```
p2,100000,1, 0.5,0.5,1, -0.5,0,0
```

```
p3,100000,1, -0.5,0.5,1, 0,-0.5,0
```

```
p4,100000,1, -0.5,-0.5,1, 0.5,0,0
```

```
p5,100000,1, 0.5,-0.5,-1, -0.5,0,0
```

```
p6,100000,1, 0.5,0.5,-1, 0,-0.5,0
```

```
p7,100000,1, -0.5,0.5,-1, 0.5,0,0
```

これは4つと3つの星が回転しながら重力で衝突するシミュレーションである。星は非常に複雑な動きをしている。図8にシミュレーション結果とその過程を示す。

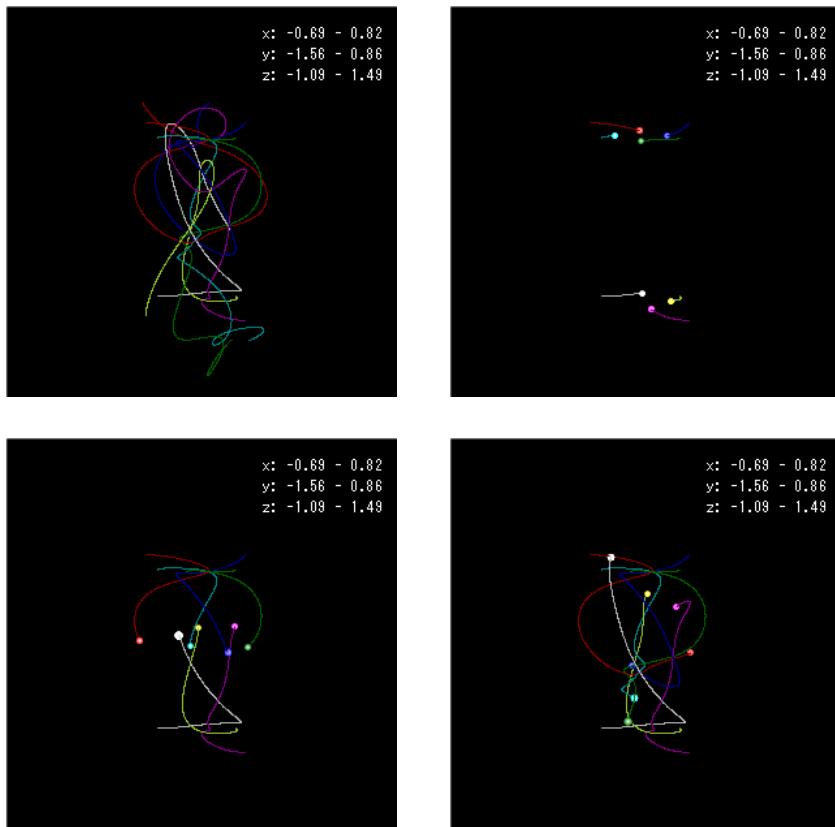


図 8 7つの星の衝突

太陽の公転運動（分割数 10000, 実行時間 11.86）

太陽,332946,109,0,0,0,0,-0.00044,0, &Hff0000

地球,1,1, 1,0,0, 0,1,0, &H00ffff

木星,318,10, 5.2,0,0, 0,0.4385,0

これは太陽、地球、木星の実際の重さ、半径、公転半径を使ったシミュレーションである。実行時間は木星の周期である。この場合、惑星が公転するのと同様に、太陽も木星からの重力で公転する。しかしその公転半径は極めて小さく、太陽半径にほぼ等しい。このシミュレーションでは、「星を見易く」のチェックを外して実行する。シミュレーション結果は図 9a のように木星軌道から眺めた図になり、動画で星は全く見えない。しかし、マウスホイールを使って太陽ヘズームして行くと（最後の段階ではズームを微細に変更している）、図 9b のように、太陽の公転運動が見えてくる。上の 2 本の線は地球と木星の公転軌道である。

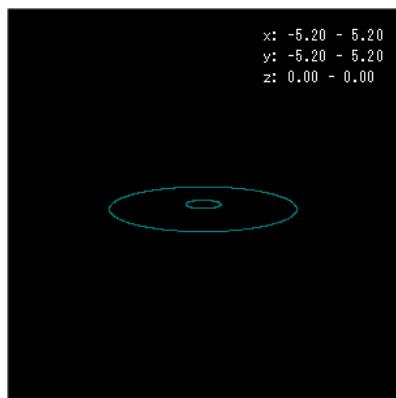


図 9a 太陽、地球、木星軌道

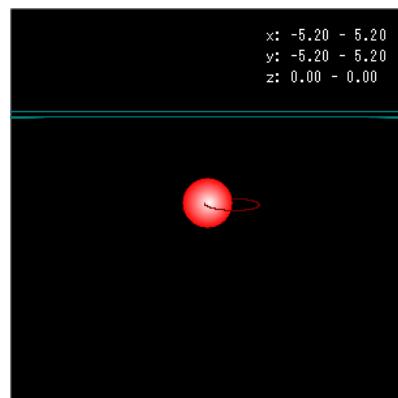


図 9b 太陽の公転運動

「中心」コンボボックスに「読み込」ボタンで選択肢を読み込み、地球を中心とした図を描画させることもできる。図 10 にその結果を表示する。

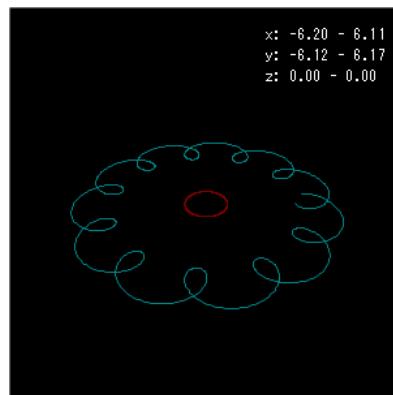


図 10 地球を中心とした太陽と木星の軌道

## 参考文献

- 1) 理工系の基礎数学 8 数値計算, 高橋大輔, 岩波書店, 1996.

## 10. 電荷と電場

今回の物理シミュレーションでは、静止点電荷からの電気力線、空間中の電場の向き、等電位面の描画を取り上げる。電気力線は、電荷の近傍でクーロン力を計算し、電場の向きを繋いで行く。空間中の電場の向きは、空間の指定点での電場の向きを矢印で表わす。その際、上限は付けるが、矢印の長さで相対的な電場の強さを近似的に表現する。等電位面の描画は、直接クーロンポテンシャルを計算する方法の他に、多重極展開を用いた電荷から遠方での近似法も加えた。さらにこの多重極展開法を拡張したツリー法や高速多重極展開法のアルゴリズムをプログラムに組み込んだ。

この章では、最初に点電荷からの電場とポテンシャルについての理論を与える。次に、ツリー法や高速多重極展開法の理論についても解説する。その後、実行メニューに従って、プログラムの利用法を説明する。

### 10.1 静止電荷と電場の理論

電荷分布  $\rho(\mathbf{r}')$  [C/m<sup>3</sup>] や点電荷  $q_i$  [C] からの電場  $\mathbf{E}(\mathbf{r})$  [V/m] は以下のように計算される。

電荷分布の場合

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int \frac{\rho(\mathbf{r}')(\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} d\mathbf{r}'$$

点電荷の場合（このプログラムの場合）

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^n \frac{q_i(\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^3}$$

ここに  $\epsilon_0 = 8.854188 \times 10^{-12}$  [F/m] は真空の誘電率である。電気力線と電場はこの関係を用いて計算する。また、等電位面を計算する際の厳密な電位  $V(\mathbf{r})$  (V) は以下のように計算される。

電荷分布の場合

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}'$$

点電荷の場合

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^n \frac{q_i}{|\mathbf{r} - \mathbf{r}_i|}$$

また電荷分布  $\rho(\mathbf{r}')$  の作るポテンシャルは、以下のように多重極展開や局所展開の式を用いて近似的に表される（ $l$  を有限で計算した場合）。今回我々は上側の定義に基づいてプログラムを作成しているが、下側（括弧内）の定義が示されている教科書もある。

$\mathbf{r} > \mathbf{r}'$  のとき（多重極展開）

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{M_{l,m} Y_{l,m}(\theta, \varphi)}{r^{l+1}} \quad (1)$$

$$\left( V(\mathbf{r}) = \frac{1}{\epsilon_0} \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{2l+1} \frac{M_{l,m} Y_{l,m}(\theta, \varphi)}{r^{l+1}} \right)$$

ここに、 $M_{l,m}$  は  $2^l$  重極子 ( $2^l$  重極モーメント) と呼ばれ、以下のように定義される。

電荷分布の場合

$$M_{l,m} = \int \rho(\mathbf{r}') r'^l Y_{l,m}^*(\theta', \varphi') dv'$$

点電荷の場合

$$M_{l,m} = \sum_{i=0}^n q(r_i) r_i^l Y_{l,m}^*(\theta_i, \varphi_i) \quad (2)$$

$r < r'$  のとき (局所展開)

$$\left( V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{l=0}^{\infty} \sum_{m=-l}^l L_{l,m} Y_{l,m}(\theta, \varphi) r^l \right)$$

$$\left( V(\mathbf{r}) = \frac{1}{\epsilon_0} \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{2l+1} L_{l,m} Y_{l,m}(\theta, \varphi) r^l \right)$$

ここに、 $L_{l,m}$  は局所展開係数と呼ばれ、以下のように定義される。

電荷分布の場合

$$L_{l,m} = \int \rho(\mathbf{r}') r'^{-(l+1)} Y_{l,m}^*(\theta', \varphi') dv'$$

点電荷の場合

$$L_{l,m} = \sum_{i=0}^n q(r_i) r_i^{-(l+1)} Y_{l,m}^*(\theta_i, \varphi_i)$$

$Y_{l,m}(\theta, \varphi)$  は球面調和関数で、 $Y_{l,m}^*(\theta, \varphi)$  はその複素共役である。今回は上側の定義に基づいている。

$$Y_{l,m}(\theta, \varphi) = \sqrt{\frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\cos \theta) e^{im\varphi}$$

$$\left( Y_{l,m}(\theta, \varphi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta) e^{im\varphi} \right)$$

ここに、 $m < 0$  に対しては以下の式が成り立つ。

$$Y_{l,-|m|}(\theta, \varphi) = (-1)^m Y_{l,|m|}^*(\theta, \varphi)$$

また、 $P_l^m(x)$  は Legendre の陪多項式で、 $m \geq 0$  に対して以下のように定義される。

$$P_l^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_l(x)$$

ここに、 $P_l(x)$  は Legendre の多項式である。

Legendre の陪多項式の計算には、以下の関係式を用いる。

$$P_m^m = (-1)^m (2m-1)!! (1-x^2)^{m/2}$$

$$(l-m)P_l^m = x(2l-1)P_{l-1}^m - (l+m-1)P_{l-2}^m$$

但し、 $P_{m-1}^m = 0$  である。

## 10.2 ツリー法と高速多重極展開法

電気力のポテンシャル計算において、電荷数が多くなると、1箇所での電位計算はその数に比例して時間がかかる。これに対して、近傍からの電位は直接計算し、遠方からの電位はまとめて多重極展開や局所展開を用いて計算し、計算時間の短縮化を図る方法が、ツリー法や高速多重極展開法である。ここではこれらについてその詳細を参考文献 1), 2) に習って説明する。

### ツリー法

まず、電荷分布の領域を座標軸ごとに  $d$  分割（このプログラムでは  $d = 2$  または  $d = 3$  としている）する。これにより、3次元空間を  $d^3$  分割することになるが、その中の1つの領域を第1階層のセルと呼ぶ。その第1階層のセルをさらに  $d^3$  分割し、その中の1つの領域を第2階層のセルと呼ぶ。第2階層のセルは  $d^6$  個になる。これを繰り返して、第  $n$  階層まで進める（このプログラムでは、第4階層まで想定している）。

第  $n$  階層の各セルの座標の中心を展開の中心として、そのセル内の電荷による多重極子を(2)式によって計算する。次に第  $n-1$  階層の各セル内で同様に多重極子を計算するが、直接計算は時間がかかるので、第  $n-1$  階層のセルに含まれる  $d^3$  個の第  $n$  階層のセルから、多重極子の中心の移動公式を用いて計算する。その方法を以下に示す。

今、第  $i$  階層のセル A の中心  $O'$  にある多重極子をそのセルを含む第  $i-1$  階層のセル B の中心  $O$  に移動させる問題を考える。セル B の中心  $O$  から、セル A の中心  $O'$  までのベクトルを極座標で  $\rho = (\rho, \alpha, \beta)$  と表す。また、 $O$  から電位を求める位置  $Q$  までのベクトルを極座標で  $\mathbf{r} = (r, \theta, \varphi)$  、 $O'$  から  $Q$  までのベクトルを極座標で  $\mathbf{r}' = (r', \theta', \varphi')$  とする。但し、 $\rho \ll r, r'$  とする。これを図で表わすと図 1 のようになる。

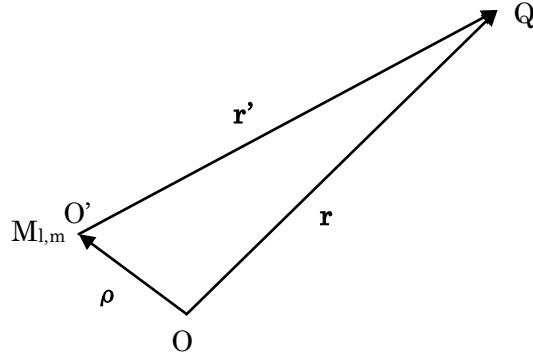


図 1 M(i)2M(i-1)

第  $i$  階層の中心にある多重極子が点  $Q$  の位置に作るポテンシャルは多重極展開によって以下のように書かれる。

$$V(r') = \frac{1}{4\pi\epsilon_0} \sum_{p=0}^{\infty} \sum_{q=-p}^p \frac{M_{p,q} Y_{p,q}(\theta', \varphi')}{r'^{p+1}}$$

この  $Y_{p,q}(\theta', \varphi')/r'^{p+1}$  に対して、球面調和関数の変換公式 (4) を代入する。

$$\frac{Y_{p,q}(\theta', \varphi')}{r'^{p+1}} = \sum_{r=0}^{\infty} \sum_{s=-r}^r \frac{J_s^q A_r^s A_p^q \rho^r Y_{r,-s}(\alpha, \beta)}{A_{r+p}^{s+q}} \frac{Y_{r+p,s+q}(\theta, \varphi)}{r^{r+p+1}} \quad (4)$$

ここに、

$$J_s^q = i^{|q+s|-|q|-|s|} = \begin{cases} (-1)^{\min(|q|, |s|)} & \text{if } q \cdot s < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$A_p^q = \begin{cases} \frac{(-1)^p}{\sqrt{(p-q)!(p+q)!}} & \text{if } p \geq q \\ 0 & \text{otherwise} \end{cases}$$

その後、 $j \equiv r+p$ ,  $k \equiv s+q$ ,  $l \equiv r$ ,  $m \equiv s$  とすると、以下のようになる。

$$V(r') = \frac{1}{4\pi\epsilon_0} \sum_{p=0}^{\infty} \sum_{q=-p}^p \sum_{r=0}^{\infty} \sum_{s=-r}^r \frac{M_{p,q} J_s^q A_r^s A_p^q \rho^r Y_{r,-s}(\alpha, \beta) Y_{r+p,s+q}(\theta, \varphi)}{A_{r+p}^{m+q}} \frac{Y_{r+p,s+q}(\theta, \varphi)}{r^{r+p+1}}$$

$$= \frac{1}{4\pi\epsilon_0} \sum_{j=0}^{\infty} \sum_{k=-j}^j \sum_{l=0}^j \sum_{m=-l}^l \frac{M_{j-l,k-m} J_m^{k-m} A_l^m A_{j-l}^{k-m} \rho^l Y_{l,-m}(\alpha, \beta) Y_{j,k}(\theta, \varphi)}{A_j^k} \frac{Y_{j,k}(\theta, \varphi)}{r^{j+1}}$$

ここで、新しく

$$M'_{j,k} \equiv \sum_{l=0}^j \sum_{m=-l}^l \frac{M_{j-l,k-m} J_m^{k-m} A_l^m A_{j-l}^{k-m} \rho^l Y_{l,-m}(\alpha, \beta)}{A_j^k} \quad (5)$$

と定義すると、ポテンシャルは新たに定義された多重極子  $M'_{j,k}$  を用いて、以下のように書かれる。

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{j=0}^{\infty} \sum_{k=-j}^j \frac{M'_{j,k} Y_{j,k}(\theta, \varphi)}{r^{j+1}}$$

これにより、多重極展開の展開の中心が変更されたことになり、その際、多重極子は(5)式のように変換されることが分かった。このように第  $i$  階層の多重極子をまとめて第  $i-1$  階層の多重極子にする処理を慣例により  $M(i)2M(i-1)$  と表す。ここで、多重極子の次数は理論的には  $\infty$  までとなっているが、現実の計算では、ある次数まで（例えば  $j \leq 3$ ）の計算で打ち切ることになる。

以上により、第  $n$  階層のセルの中心にある多重極子をそのセルを含む第  $n-1$  階層のセルの中心に移動させることができるようになった。この方法を利用して、さらに第  $n-1$  階層のセルの中心にある多重極子をそのセルを含む第  $n-2$  階層のセルの中心へ移動させ、これを繰り返して、第 1 階層の  $d^3$  個のセルまで続ける。これらの多重極子を用いて遠距離への効果をまとめて扱うようとする。

ツリー法は、ある位置でのポテンシャルを、近距離では厳密に、それより遠い所では第  $n$  階層のセルの中心にある多重極子を使って、さらに遠い所では、第  $n-1$  階層のセルの中心にある多重極子を使って、というように、距離によって階層別に計算し、それを足し合わせて求める。このプログラムでは、基本的な距離を第  $n$  階層のセルの対角長とし、電位を求める点と各階層のセルの中心との距離が、その何倍以上かでどの階層の多重極子を使うかを決められるようになっている。精度と実行速度を秤にかけて、データ数に応じて、セルの階層、分割数、多重極展開の次数、距離を決めるパラメータ等を効果的に決めて行く必要がある。

### 高速多重極展開法

高速多重極展開法はツリー法をさらに進めた方法である。ツリー法において、ある点でのポテンシャルは、直接計算と距離による様々な階層の多重極子からの寄与の合計で与えられるが、これらは1つの位置につき、その都度計算を実行する必要があった。高速多重極展開法は、最初にこの複数の階層の多重極子からの寄与を第  $n$  階層の各セルの中心に集約しておく、その後の各点でのポテンシャル計算は、直接計算とその点が含まれる第  $n$  階層のセルの中心にある多重極子の寄与だけから計算する。これにより、多くの位置のポテンシャルを計算する場合は、ツリー法に比べ、計算効率がさらに向上する。但し、最初の集約段階でかなりの計算量を必要とするので、ポテンシャルを求める位置が少ない場合は、逆に効率が悪くなる。我々のプログラムでは、位置の数が 10,000 点程度から効果が表れて来るようである。

高速多重極展開法では、多重極子の集約に局所展開の方法を用いるが、これには 2 通りの方法が考えられる。遠方の多重極子を、ポテンシャルを調べる位置の第  $n$  階層のセルの中心に直接局所展開する方法と、遠方の多重極子が含まれる階層と調べる点が含まれる同じ階層のセルに一度局所展開し、さらに上の階層から順番に第  $n$  階層まで、局所展開を行う方法である。前者は局所展開が一度で済

む分、計算精度の上で有利であるが、後者は、同じ階層のセルに局所展開する際、一度計算した値を記憶させておくことで計算時間の短縮を図れるという利点がある。我々のプログラムでは、この問題についても、どの程度時間短縮が図れるのか調べられるようになっている。

ここでは、第  $i$  階層のセル A の中心  $O'$  にある多重極子を遠く離れた第  $j$  階層のセル B (階層は同じでなくてもよい) の中心  $O$  に局所展開する問題を考える。セル B の中心  $O$  から、セル A の中心  $O'$  までのベクトルを極座標で  $\rho = (\rho, \alpha, \beta)$  と表す。また、 $O$  から  $Q$  までのベクトルを極座標で  $\mathbf{r} = (r, \theta, \varphi)$  、 $O'$  から  $Q$  までのベクトルを極座標で  $\mathbf{r}' = (r', \theta', \varphi')$  とする。但し、この場合は、 $r \ll \rho, r'$  である。これを図で表わすと図 2 のようになる。

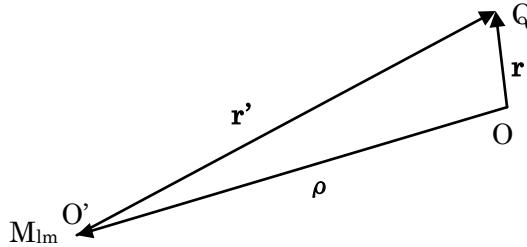


図 2 M(i)2L(j)

セル A の中心  $O'$  にある多重極子が点  $Q$  の位置に作るポテンシャルは多重極展開によって以下のように書かれる。

$$V(\mathbf{r}') = \frac{1}{4\pi\epsilon_0} \sum_{p=0}^{\infty} \sum_{q=-p}^p \frac{M_{p,q} Y_{p,q}(\theta', \varphi')}{r'^{p+1}}$$

この  $Y_{p,q}(\theta', \varphi')/r'^{p+1}$  に対して、球面調和関数の 2 番目の変換公式 (6) を代入する。

$$\frac{Y_{p,q}(\theta', \varphi')}{r'^{p+1}} = \sum_{r=0}^{\infty} \sum_{s=-r}^r \frac{J_{p,s}^q A_r^s A_p^q Y_{p+r, q-s}(\alpha, \beta)}{A_{r+p}^{s-q} \rho^{r+p+1}} Y_{r,s}(\theta, \varphi) r^r \quad (6)$$

ここに、

$$J_{p,s}^q = (-1)^p i^{|s-q|-|s|-|q|} = \begin{cases} (-1)^p (-1)^{\min(|q|, |s|)} & \text{if } q \cdot s > 0 \\ (-1)^p & \text{otherwise} \end{cases}$$

その後、 $j \equiv r$ ,  $k \equiv s$ ,  $l \equiv p$ ,  $m \equiv q$  とすると、以下のようなになる。

$$\begin{aligned} V(\mathbf{r}') &= \frac{1}{4\pi\epsilon_0} \sum_{p=0}^{\infty} \sum_{q=-p}^p \sum_{r=0}^{\infty} \sum_{s=-r}^r \frac{M_{p,q} J_{p,s}^q A_r^s A_p^q Y_{p+r, q-s}(\alpha, \beta)}{A_{r+p}^{s-q} \rho^{r+p+1}} Y_{r,s}(\theta, \varphi) r^r \\ &= \frac{1}{4\pi\epsilon_0} \sum_{j=0}^{\infty} \sum_{k=-j}^j \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{M_{l,m} J_{l,k}^m A_l^m A_{l+j}^m Y_{l+j, m-k}(\alpha, \beta)}{A_{j+l}^{k-m} \rho^{j+l+1}} Y_{j,k}(\theta, \varphi) r^j \end{aligned}$$

ここで、新しく

$$L_{j,k} \equiv \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{M_{l,m} J_{l,k}^m A_j^k A_l^m Y_{l+j,m-k}(\alpha, \beta)}{A_{j+l}^{k-m} \rho^{j+l+1}} \quad (7)$$

と定義すると、ポテンシャルは新たに定義された局所展開係数  $L_{j,k}$  を用いて、以下のように書かれる。

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{j=0}^{\infty} \sum_{k=-j}^j L_{j,k} Y_{j,k}(\theta, \phi) r^j$$

この展開を慣例として  $M(i)2L(j)$  と呼ぶ。

上の局所展開が同階層で行われた場合、階層間の局所展開を続けて行う。その理論を説明しておく。

ここでは、第  $i$  階層のセル A の中心  $O'$  にある多重極子をそれに含まれる第  $i+1$  階層のセル B の中心  $O$  に局所展開する問題を考える。セル B の中心  $O$  から、セル A の中心  $O'$  までのベクトルを極座標で  $\rho = (\rho, \alpha, \beta)$  と表す。また、 $O$  から電位を求める位置  $Q$  までのベクトルを極座標で  $\mathbf{r} = (r, \theta, \varphi)$  、 $O'$  から  $Q$  までのベクトルを極座標で  $\mathbf{r}' = (r', \theta', \varphi')$  とする。但し、この場合は、セル A の中にセル B があるため、 $r < \rho, r'$  であるが、これまでのよう大きな差はない。そのため、局所展開の際の誤差は大きくなる。これを図で表わすと図 3 のようになる。

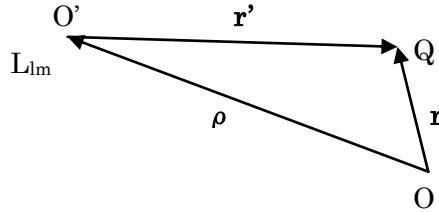


図 3  $L(i)2L(i+1)$

セル A の中心  $O'$  にある局所展開係数が点  $Q$  の位置に作るポテンシャルは局所展開によって以下のように書かれる。

$$V(\mathbf{r}') = \frac{1}{4\pi\epsilon_0} \sum_{p=0}^{\infty} \sum_{q=-p}^p L_{p,q} Y_{p,q}(\theta', \varphi') r'^p$$

この  $Y_{p,q}(\theta', \varphi')/r'^{p+1}$  に対して、球面調和関数の 3 番目の変換公式 (8) を代入する。

$$Y_{p,q}(\theta', \varphi') r'^p = \sum_{r=0}^{\infty} \sum_{s=-r}^r \frac{J_{r,s}^q A_r^s A_{p-r}^{q-s} \rho^r Y_{r,s}(\alpha, \beta)}{A_p^q} Y_{p-r, q-s}(\theta, \varphi) r^{p-r} \quad (8)$$

ここに、

$$J_{r,s}^q = (-1)^r i^{|q|-|s|-|q-s|} = \begin{cases} (-1)^r (-1)^s & \text{if } q \cdot s < 0 \\ (-1)^r (-1)^{q-s} & \text{if } q \cdot s \geq 0 \text{ and } |q| < |s| \\ (-1)^r & \text{otherwise} \end{cases}$$

その後、 $j \equiv p-r$ ,  $k \equiv q-s$ ,  $l \equiv p$ ,  $m \equiv q$  とすると、以下のようになる。

$$\begin{aligned}
 V(\mathbf{r}') &= \frac{1}{4\pi\epsilon_0} \sum_{p=0}^{\infty} \sum_{q=-p}^p \sum_{r=0}^{\infty} \sum_{s=-r}^r \frac{L_{p,q} J'_{r,s} A_r^s A_{p-r}^{q-s} \rho^r Y_{r,s}(\alpha, \beta)}{A_l^m} Y_{p-r, q-s}(\theta, \varphi) r^{p-r} \\
 &= \frac{1}{4\pi\epsilon_0} \sum_{j=0}^{\infty} \sum_{k=-j}^j \sum_{l=j}^{\infty} \sum_{m=-l}^l \frac{L_{l,m} J'_{l-j, m-k} A_{l-j}^{m-k} A_j^k \rho^{l-j} Y_{l-j, m-k}(\alpha, \beta)}{A_l^m} Y_{j,k}(\theta, \varphi) r^j
 \end{aligned}$$

ここで、新しく

$$L'_{j,k} \equiv \sum_{l=j}^{\infty} \sum_{m=-l}^l \frac{L_{l,m} J'_{l-j, m-k} A_{l-j}^{m-k} A_j^k \rho^{l-j} Y_{l-j, m-k}(\alpha, \beta)}{A_l^m} \quad (9)$$

と定義すると、ポテンシャルは新たに定義された局所展開係数  $L'_{j,k}$  を用いて、以下のように書かれる。

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{j=0}^{\infty} \sum_{k=-j}^j L'_{j,k} Y_{j,k}(\theta, \varphi) r^j$$

展開の次数は  $\infty$  までになっているが、現実の計算ではある次数までとする。この展開を慣例として  $L(i)2L(i+1)$  と呼ぶ。

### 10.3 プログラムの利用法

メニュー [分析－教育・科学他－物理シミュレーション－電荷と電場] を選択すると、図 4 のような実行メニューが表示される。



図 4 実行メニュー

データの入力形式はグリッド入力で、「入力形式」ボタンをクリックすると図 5 のように例が表示される。

	電荷	X	Y	Z
▶ 電荷1	1	5	0	0
▶ 電荷2	-1	-5	0	0

図 5 入力形式

これは、 $x = \pm 5 \text{ m}$  のところに、 $\pm 1 \times 10^{-6}$  クーロンの電荷がある例である。メニューの最下部に書いているように、電荷の単位は、 $10^{-6}$  クーロンとする。

図 5 のような形にデータをグリッド入力し、描画範囲をすべての軸で-10m から 10m に設定し、「電気力線」ボタンをクリックすると、図 6 のような結果が表示される。電気力線は、出て行く（入って来る）本数だけが設定され、方向はプログラムで決まった設定による。

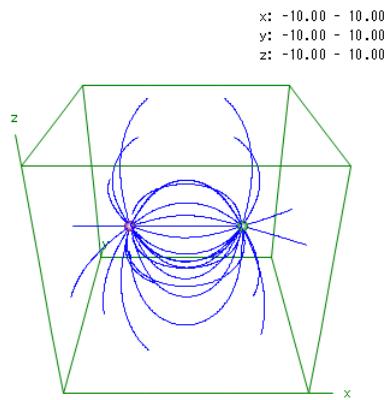


図 6 電気双極子の電気力線

電荷から出る（入る）電気力線の本数は、「単位電荷当」（ $10^{-6}$  クーロン当り）テキストボックスに入力する。図 4 のメニューでは、12 本に設定されており、正の電荷から 12 本出て、負の電荷に 12 本入っている。それぞれの電気力線は、異符号の電荷に入って終わるが、途中で描画範囲を超える場合は中断する。電気力線は「微小区間」の長さの線分を繋ぎ合わせて表示するが、繋いだ本数が、「最大連鎖」数を超える場合も中断する。必要な場合は、最大連鎖数を増やしてきれいな図を描くことができる。電荷の大きさは、「電荷半径」で指定することができる。色は正電荷が紫、負電荷が緑に設定されている。

図 7 に本数を 30 本に増やした例、図 8 に電気 8 重極子からの電気力線の例を示す。

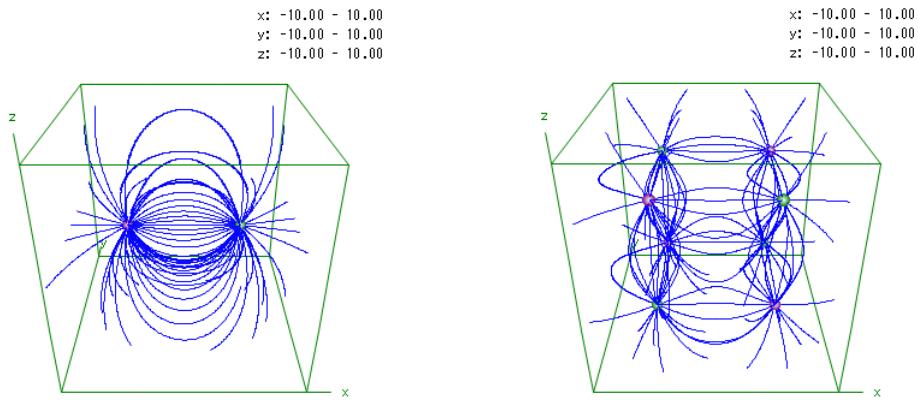


図 7 本数を増やした電気力線

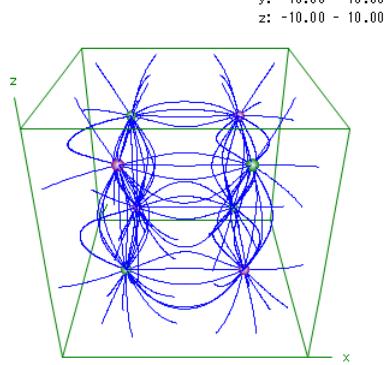


図 8 重極子からの電気力線

電気力線の他に、我々は空間中の電場を矢印で表現することにした。図 2.9 にその例を示す。矢印の向きは電場の向き、長さは空間内の電場の強さに比例するようにし、平均+標準偏差の 2 倍以上を基準の長さ、-2 倍以下を 0 になるように描画している。また、図 9 では、電場の矢印が重なって見にくないので、平面を切り出す機能も付けることにした。図 10 にその例を示す。この図は、描画範囲の y 軸の幅を 0 にして、「電場表示」を「8,1,8」に設定して得られたものである。

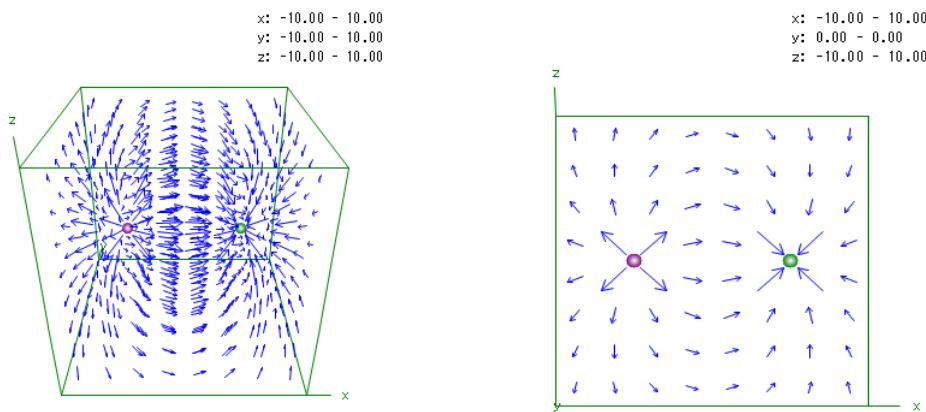


図 9 電場の向きと強さ

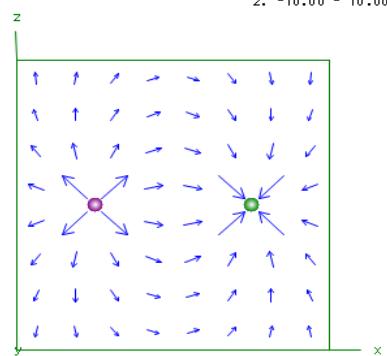


図 10 電場の平面表示

次に等電位面の描画について説明する。電荷量は  $10^{-6}$  クーロンで位置を  $x = \pm 1m$  にする。等電位面を描くには、まず「端点電位」ボタンで、領域の端の電位を調べておく。図 11 に双極子の端点での電位を示す。

端点電位		電位	X	Y	Z
▶	端点1	34.5164	-10	-10	-10
	端点2	34.5164	-10	-10	10
	端点3	34.5164	-10	10	-10
	端点4	34.5164	-10	10	10
	端点5	-34.5164	10	-10	-10
	端点6	-34.5164	10	-10	10
	端点7	-34.5164	10	10	-10
	端点8	-34.5164	10	10	10

図 11 双極子の領域の端点での電位

等電位面は、「開始」電位、電位の「幅」、「枚数」を指定し、「等電位面」グループボックス内の「厳密計算」ボタンをクリックして描画する。図 12 に、範囲を各軸-10m から 10m に設定した 100V の等電位面、図 13 に、-200V, -100V, 0V, 100V, 200V の 5 枚の等電位面をまとめて示す。図 2.13 で中央の平面は 0V 等電位面である。ここで電荷半径は 0.2 に設定している。ここでは電位表示の「分割」が 40 のデフォルトの設定で、約 64,000 ( $40^3$ ) 点の電位を調べ、それを用いて電位の等高線を描いている。

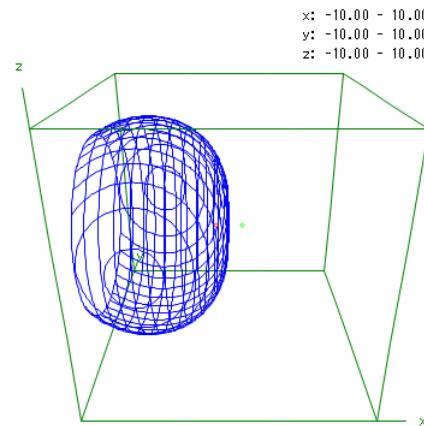


図 12 双極子からの 100V 等電位面

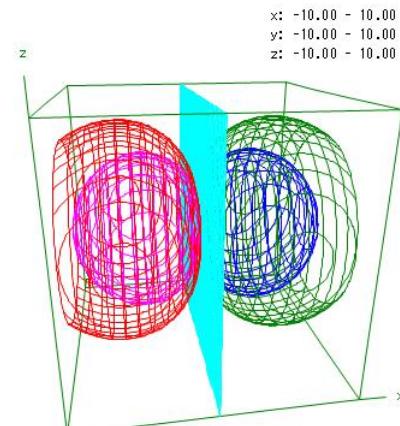


図 13 5 枚の等電位面

図 14 に  $y$  平面上の 4 重極子(電荷量は  $10^{-6}$  クーロンで、位置は  $x, z = \pm 1$  )からの 50V 等電位面、図 15 に 8 重極子 (電荷量は  $10^{-6}$  クーロンで、位置は  $x, y, z = \pm 1$  ) からの 10V 等電位面を示す。

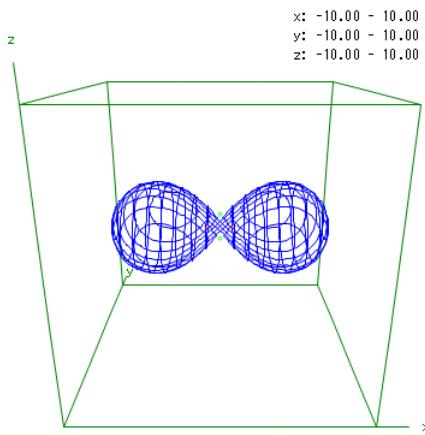


図 14 4 重極子からの 50V 等電位面

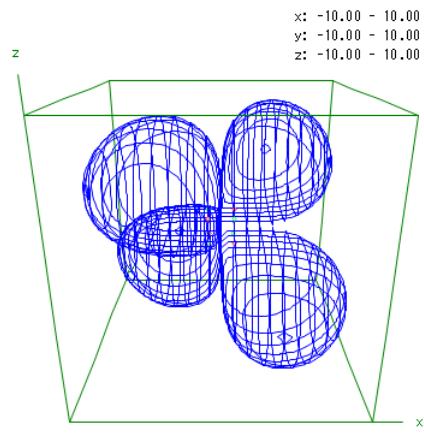


図 15 8 重極子からの 10V 等電位面

描画の精度は「分割」と「表示間隔」で与えられる。分割は、領域を分割して各点で電位を測る各軸の分割数、表示間隔は、分割をいくつ毎表示するかの間隔を表す。「分割」を 50、「表示間隔」を 1 にして、上と同じ図を図 2.16 と図 2.17 に描いておく。

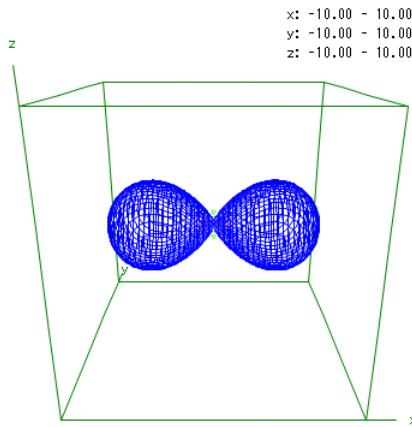


図 2.16 4 重極子からの等電位面詳細表示

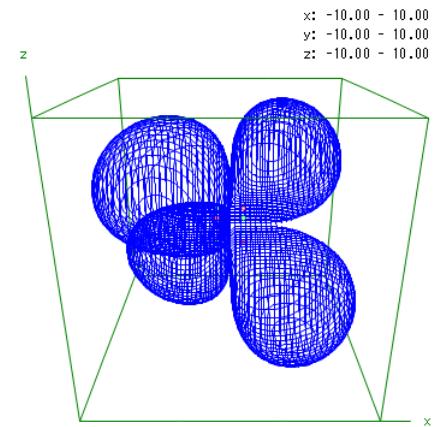


図 2.17 8 重極子からの等電位面詳細表示

次に、電荷から遠方の近似を与える多重極展開を用いた等電位面は、「多重極展開」ボタンをクリックすると得られる。その際、多重極展開の次数は「多重極次数」テキストボックスに入力する。図 18 に上で述べた双極子からの 3 次の多重極展開による 100V 等電位面を示す。図 19 に上で述べた 8 重極子からの 3 次の多重極展開による 10V 等電位面を示す。図 12 と図 15 と比べて良い近似になっていることが分かる。但し、この場合電荷数が少ないので、計算時間は直接計算の方がかなり速い。

x: -10.00 - 10.00  
y: -10.00 - 10.00  
z: -10.00 - 10.00

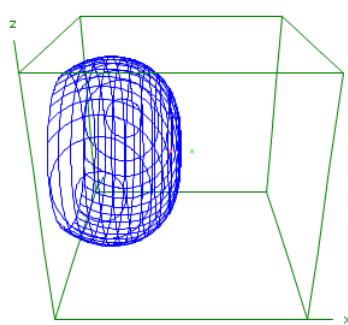


図 2.18 双極子からの 100V 等電位面

x: -10.00 - 10.00  
y: -10.00 - 10.00  
z: -10.00 - 10.00

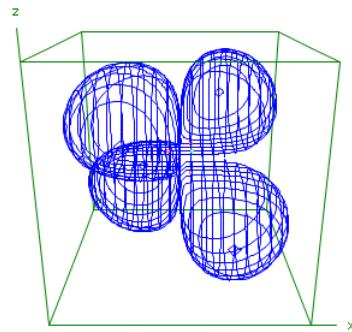


図 2.19 8 重極子からの 10V 等電位面

最後に、各軸とも-2m から 2m までの区間に分布する 1,000 個の電荷の場合について、厳密計算と多重極展開の近似との比較を行っておく。この電荷数では、多重極展開近似がかなり速くなる。図 20a に厳密計算の等電位面、図 20b に 3 次までの近似、図 20c に 2 次までの近似、図 20d に 1 次までの近似の等電位面を示す。遠方では 3 次までの近似で、比較的良い結果が与えられる。

x: -10.00 - 10.00  
y: -10.00 - 10.00  
z: -10.00 - 10.00

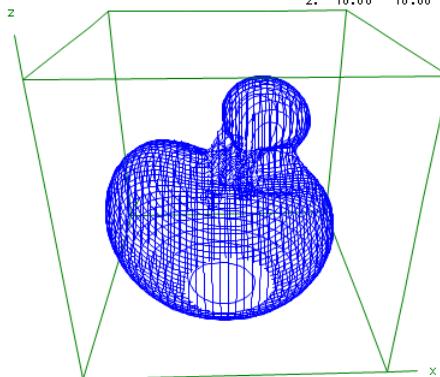
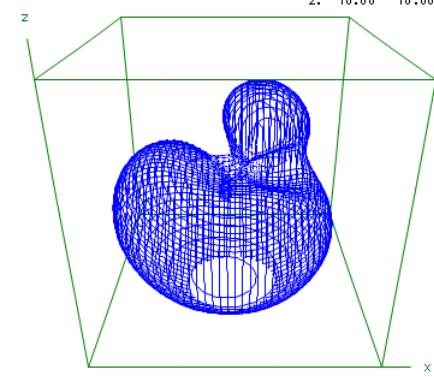
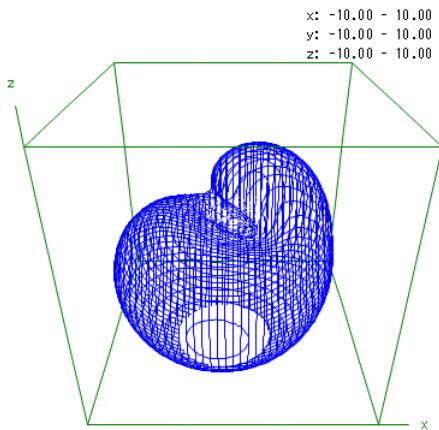
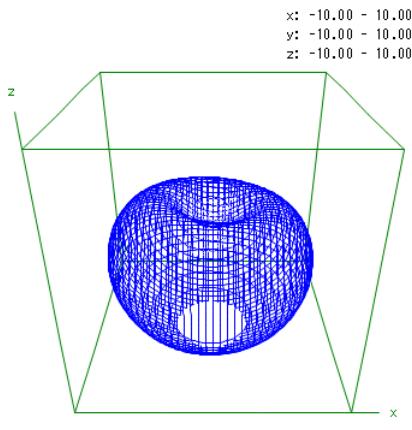


図 20a 厳密計算の等電位面

x: -10.00 - 10.00  
y: -10.00 - 10.00  
z: -10.00 - 10.00

図 20b  $l \leq 3$  の多重極展開近似

図 20c  $l \leq 2$  の多重極展開近似図 20d  $l \leq 1$  の多重極展開近似

次に、ツリー法や高速多重極展開法（FMM）に関係する部分について説明するが、設定が複雑になるので、もう一度プログラム実行画面から、これらに関係のある部分を抜き出して、図 21 に示しておく。



図 21 ツリー法と高速多重極展開法実行画面

これまで述べて来た等電位面の描画は、デフォルトの設定で約 64,000 点のポテンシャルを求めて、描画するプログラムである。電荷の数が、1,000 点位までは厳密計算で対応可能であるが、10,000 点を超えると、時間がかかり、ツリー法や高速多重極展開法が有利になる。

図 21 のメニューで、「階層」はこれらの計算のセルの階層である。2 階層から 4 階層にまで対応している。「分割」は、1 つ上の階層のセルの軸をいくつに分割するかを決める。通常は 2 分割か 3 分割であるが、パソコンで取り扱える範囲で、4 階層 3 分割は計算時間が余計にかかってしまい、現実的ではない。10,000 点で 3 階層 2 分割、100,000 点で 4 階層 2 分割当りが適当ではないかと思われる。

「多重極次数」はどの次数まで多重極展開や局所展開を考えるかを決める。「高速化」は、次数が 3 次まで、球面調和関数の具体的な形を用いる、 $M(i)2L(i)$  の上位 3 階層で計算結果を保存しておき、

計算量を減らすように工夫する方法である。これによって、2割程度高速化される。「近傍決定パラメータ」は、精度とスピードに大きく影響する。上位階層から見て行き、最下位階層のセルの対角長の何倍までその階層の多重極子の結果を用いるかを指定する。大きな値を用いるほど精度は上がるが、スピードは落ちる。

「M2M」は  $M(i)2M(i-1)$  の処理を行うか、直接電荷から多重極子を計算するかを決める。通常はチェックを入れて、多重極子の移動によって階層的に計算する方法を選んでおく。「L2L」は  $M2L$  の局所展開を同じ階層で行うか、直接下の階層で行うかを決める。チェックを入れると同じ階層で行い、 $L2L$  の処理を追加する。

通常は、図 21 の「ツリー法」や「FMM」（高速多重極展開）等のボタンで、等電位面が描かれるが、「電荷位置電位」をチェックすると、各電荷の位置の電位を表形式で表示する。計算精度を厳密計算と詳しく比較する場合に利用する。このプログラムでは処理時間の測定が重要なので、一番下に、画面表示などを除いた処理時間を表示するようにしている。

計算の例として、10,000 個の電荷が領域全体に分布している場合の約 64,000 点での電位の計算を 4 つの計算法で比較する。計算結果を図 22a～22d に 1 つの等電位面（この例では-50,000V）として与える。多重極展開は 3 次の多重極子まで求めているが、電荷から遠方での近似でしか利用できないため、結果は正しくない。ツリー法と高速多重極展開法では、ほぼデフォルトの設定で計算しているが、「多重極次数」を 2 次までとし、「高速化」の処理を行っている。この電荷数では、ツリー法と高速多重極展開法でさほど計算時間に差はないが、精度においては、局所展開の近似が無い分、ツリー法が上回っている。設定によって、高速多重極展開法では、さらに計算時間を短縮することもできる（我々の計算では約半分程度）。今、電位を計算する地点の数を固定しているが、一般には電荷が存在する位置の電位計算をする場合が多い。そのような場合、電荷数がさらに増えると、高速多重極展開法の計算は有利になる。電荷数に応じて、どんな設定でどのような計算法を利用するのが有利か、今後議論しておく必要がある。

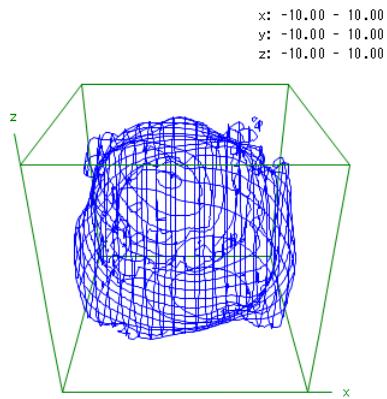


図 22a 厳密計算 (564 秒)

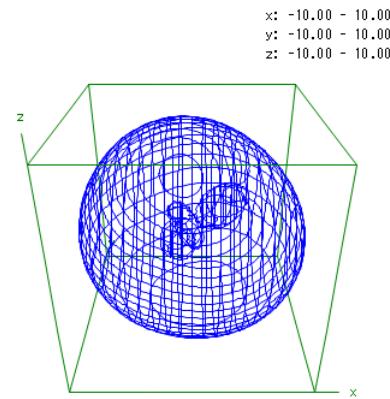


図 22b 多重極展開法 (5 秒)

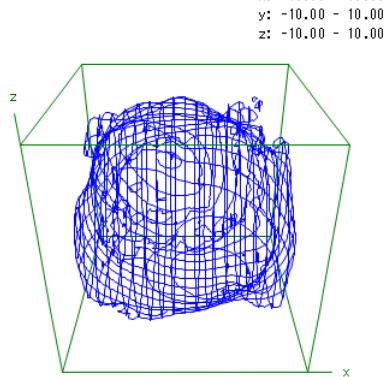


図 22c ツリー法 (153 秒)

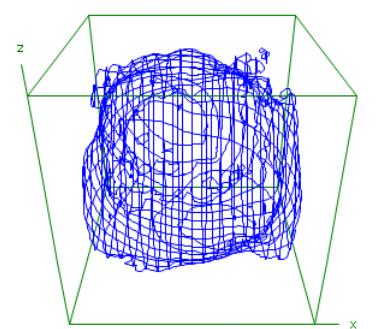


図 22d 高速多重極展開法 (127 秒)

## 参考文献

- 1) コンピュータ・シミュレーションの基礎 [第 2 版] , 岡崎進・吉井範行, 化学同人, 2011.
- 2) The Rapid Evaluation of Potential Fields in Particle Systems, Leslie F. Greengard, MIT Press.

## 1.1. 質点系の運動

### 1. 束縛運動

最初に、ある曲面に束縛された質点の運動について考える際の、法線ベクトルとそれに関する角度の関係を与える。その曲面を以下の関数で表わす。

$$z = f(x, y)$$

外力は $z$ 軸下向きに、 $mg$ の力が働くものとする。ここに $m$ は運動する質点の質量である。 $x$ 方向に $dx$ 進む際の $z$ 方向の変化 $dz_x$ 、及び $y$ 方向に $dy$ 進む際の $z$ 方向の変化 $dz_y$ は以下で与えられる。

$$dz_x = f_x dx \equiv \tan \theta_x dx$$

$$dz_y = f_y dy \equiv \tan \theta_y dy$$

ベクトル  $\mathbf{ds}_x = (dx, 0, dz_x)$  と  $\mathbf{ds}_y = (0, dy, dz_y)$  が作る平面の法線ベクトル  $\mathbf{n}$  は、

$$\mathbf{n} \propto \mathbf{ds}_x \times \mathbf{ds}_y \propto (-\tan \theta_x, -\tan \theta_y, 1)$$

より、以下となる。

$$\mathbf{n} = \left( -\frac{\sin \theta_x \cos \theta_y}{d}, -\frac{\cos \theta_x \sin \theta_y}{d}, \frac{\cos \theta_x \cos \theta_y}{d} \right)$$

$$d = \sqrt{\cos^2 \theta_x + \cos^2 \theta_y + \cos^2 \theta_x \cos^2 \theta_y}$$

法線を含む鉛直面内で法線の傾き $\theta'$ は、以下のように与えられる。

$$\tan \theta' = \frac{\cos \theta_x \cos \theta_y}{\sqrt{\sin^2 \theta_x \cos^2 \theta_y + \cos^2 \theta_x \sin^2 \theta_y}} = \frac{1}{\sqrt{\tan^2 \theta_x + \tan^2 \theta_y}}$$

これから、平面の傾き $\theta$ は以下となる。

$$\tan \theta = \sqrt{\tan^2 \theta_x + \tan^2 \theta_y}$$

この関係を利用すると、

$$d = \frac{\cos \theta_x \cos \theta_y}{\cos \theta}$$

これより、法線ベクトルは以下のようにも書かれる。

$$\mathbf{n} = (-\tan \theta_x \cos \theta, -\tan \theta_y \cos \theta, \cos \theta)$$

法線ベクトルの $x$ 軸の負の向きからの角度を $\varphi$ とすると、これは以下のように表わされる。

$$\cos \varphi = \frac{\sin \theta_x \cos \theta_y}{\sqrt{\sin^2 \theta_x \cos^2 \theta_y + \cos^2 \theta_x \sin^2 \theta_y}} = \frac{\tan \theta_x}{\tan \theta}$$

$$\sin \varphi = \frac{\cos \theta_x \sin \theta_y}{\sqrt{\sin^2 \theta_x \cos^2 \theta_y + \cos^2 \theta_x \sin^2 \theta_y}} = \frac{\tan \theta_y}{\tan \theta}$$

この角には、以下の関係もある。

$$\tan \theta_x \cos \varphi + \tan \theta_y \sin \varphi = \tan \theta$$

$$\tan \theta_y \cos \varphi - \tan \theta_x \sin \varphi = 0$$

もう少しベクトルの関係式を与えておく。ある単位ベクトルを、 $\mathbf{n}_1 = (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$  とすると、このベクトルに直交し、互いに直交する 2 つの単位ベクトル  $\mathbf{n}_2, \mathbf{n}_3$  は、以下のように与えることができる。

$$\mathbf{n}_2 = (-\sin \varphi, \cos \varphi, 0)$$

$$\mathbf{n}_3 = (-\cos \theta \cos \varphi, -\cos \theta \sin \varphi, \sin \theta)$$

またこれに対して、任意の角度  $\alpha$  について、以下も直交系をなす。

$$\mathbf{n}_1 = (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$$

$$\mathbf{n}'_2 = (-\sin \varphi \cos \alpha - \cos \theta \cos \varphi \sin \alpha, \cos \varphi \cos \alpha - \cos \theta \sin \varphi \sin \alpha, \sin \theta \sin \alpha)$$

$$\mathbf{n}'_3 = (\sin \varphi \sin \alpha - \cos \theta \cos \varphi \cos \alpha, -\cos \varphi \sin \alpha - \cos \theta \sin \varphi \cos \alpha, \sin \theta \cos \alpha)$$

(1)

これらの関係はプログラムで利用される。特に後に出るバネの描写には非常に役に立つ。

さて、床束縛問題をラグランジュの未定乗数法で考えてみる。ポテンシャルを  $mgz$ 、束縛面を  $z = f(x, y)$  とすると、ラグランジアンは以下となる。

$$L = \frac{m}{2} (v_x^2 + v_y^2 + v_z^2) - mgz + \lambda(z - f(x, y))$$

ここに  $\lambda$  は未定乗数である。運動方程式は以下となる。

$$\begin{aligned} m \frac{dv_x}{dt} + \lambda f_x &= 0 \\ m \frac{dv_y}{dt} + \lambda f_y &= 0 \\ m \frac{dv_z}{dt} + mg - \lambda &= 0 \end{aligned}$$

$$z - f(x, y) = 0$$

束縛条件を  $z$  方向の運動方程式に代入すると、

$$\begin{aligned} v_z &= f_x v_x + f_y v_y \\ \frac{dv_z}{dt} &= f_x \frac{dv_x}{dt} + f_y \frac{dv_y}{dt} + f_{xx} v_x^2 + 2f_{xy} v_x v_y + f_{yy} v_y^2 \end{aligned}$$

より、

$$\lambda = m \left( f_x \frac{dv_x}{dt} + f_y \frac{dv_y}{dt} + f_{xx} v_x^2 + 2f_{xy} v_x v_y + f_{yy} v_y^2 + g \right)$$

これに  $x, y$  方向の運動方程式を代入すると

$$\lambda = mg - (f_x^2 + f_y^2) \lambda + m(f_{xx} v_x^2 + 2f_{xy} v_x v_y + f_{yy} v_y^2)$$

より、以下となる。

$$\lambda = m \frac{g + (f_{xx} v_x^2 + 2f_{xy} v_x v_y + f_{yy} v_y^2)}{1 + f_x^2 + f_y^2}$$

以前の記号を利用すると、 $f_x = \tan \theta_x$ ,  $f_y = \tan \theta_y$  より、

$$\lambda = m[g + (f_{xx}v_x^2 + 2f_{xy}v_xv_y + f_{yy}v_y^2)]\cos^2\theta$$

これを $\lambda \equiv T\cos\theta$ と定義すると、運動方程式は以下となる。

$$\begin{aligned} m\frac{dv_x}{dt} &= -T\cos\theta\tan\theta_x = -T\sin\theta\cos\varphi \\ m\frac{dv_y}{dt} &= -T\cos\theta\tan\theta_y = -T\sin\theta\sin\varphi \\ m\frac{dv_z}{dt} &= -mg + T\cos\theta \end{aligned}$$

$$T = m[g + (f_{xx}v_x^2 + 2f_{xy}v_xv_y + f_{yy}v_y^2)]\cos\theta$$

ここで、 $T$ は床からの抗力に相当する。これより、束縛条件は床面鉛直方向で、重力と遠心力につり合いを持たせる力を生じさせることを意味していることが分かる。

抗力 $T$ を使って、摩擦のある場合の運動方程式は、摩擦係数を $\mu$ として以下で与えられる。

$$\begin{aligned} m\frac{d^2x}{dt^2} &= -T\sin\theta\cos\varphi - \mu T\frac{v_x}{v} \\ m\frac{d^2y}{dt^2} &= -T\sin\theta\sin\varphi - \mu T\frac{v_y}{v} \\ m\frac{d^2z}{dt^2} &= -mg + T\cos\theta - \mu T\frac{v_z}{v} \end{aligned}$$

但し、 $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$ ， $v_z = \tan\theta_x v_x + \tan\theta_y v_y$  である。

次に質点の反発について考える。床面の法線ベクトルは以下のようにあたえられるので、

$$\mathbf{n} = (-\tan\theta_x\cos\theta, -\tan\theta_y\cos\theta, \cos\theta)$$

これに直交するベクトルで、同一鉛直面内のものを $\mathbf{n}_1$ 、法線ベクトルとこのベクトルに垂直のものを $\mathbf{n}_2$ とすると、以下のように表わされる。

$$\begin{aligned} \mathbf{n}_1 &= \left(-\tan\theta_x \frac{\cos^2\theta}{\sin\theta}, -\tan\theta_y \frac{\cos^2\theta}{\sin\theta}, -\sin\theta\right) \\ \mathbf{n}_2 &= \left(-\tan\theta_y \frac{\cos\theta}{\sin\theta}, \tan\theta_x \frac{\cos\theta}{\sin\theta}, 0\right) \end{aligned}$$

入射する質点の速度を $\mathbf{v}$ 、反発係数を $e$ とすると、反発後の速度 $\mathbf{v}'$ は以下で与えられる。

$$\mathbf{v}' = -e(\mathbf{v} \cdot \mathbf{n})\mathbf{n} + (\mathbf{v} \cdot \mathbf{n}_1)\mathbf{n}_1 + (\mathbf{v} \cdot \mathbf{n}_2)\mathbf{n}_2$$

1 質点の束縛問題をさらに一般化する。ポテンシャルを $g(x, y, z)$ 、束縛面を $f(x, y, z) = 0$ とすると、ラグランジアンは以下となる。

$$L = \frac{m}{2}(v_x^2 + v_y^2 + v_z^2) - g(x, y, z) + \lambda f(x, y, z)$$

これを用いると運動方程式は以下となる。

$$\begin{aligned} m\frac{dv_x}{dt} + g_x - \lambda f_x &= 0 \\ m\frac{dv_y}{dt} + g_y - \lambda f_y &= 0 \\ m\frac{dv_z}{dt} + g_z - \lambda f_z &= 0 \end{aligned}$$

$$f(x, y, z) = 0$$

束縛条件から

$$f_x v_x + f_y v_y + f_z v_z = 0$$

$$f_x \frac{dv_x}{dt} + f_y \frac{dv_y}{dt} + f_z \frac{dv_z}{dt} + f_{xx} v_x^2 + f_{yy} v_y^2 + f_{zz} v_z^2 + 2f_{xy} v_x v_y + 2f_{yz} v_y v_z + 2f_{zx} v_z v_x = 0$$

これに3つの方程式を代入して、 $\lambda$ について求めると以下のようになる。

$$\lambda = \frac{(f_x g_x + f_y g_y + f_z g_z) - m(f_{xx} v_x^2 + f_{yy} v_y^2 + f_{zz} v_z^2 + 2f_{xy} v_x v_y + 2f_{yz} v_y v_z + 2f_{zx} v_z v_x)}{f_x^2 + f_y^2 + f_z^2}$$

これを用いて運動方程式を解くことができる。

最後に複数の質点系で複数の束縛条件のある問題について考える。この質点系のラグランジアンは以下で与えられる。

$$L = \sum_{i=1}^N \left[ \frac{m_i}{2} \mathbf{v}_i^2 - g(\mathbf{r}_i) \right] + \sum_{k=1}^p \lambda_k f_k(\mathbf{r})$$

運動方程式は $3N$ 個で、以下となる。下は質量で割った式である。

$$m_i \frac{d\mathbf{v}_i}{dt} + \frac{\partial g}{\partial \mathbf{x}_i} - \sum_{k=1}^p \lambda_k \frac{\partial f_k}{\partial \mathbf{x}_i} = 0$$

$$\frac{d\mathbf{v}_i}{dt} + \frac{1}{m_i} \frac{\partial g}{\partial \mathbf{x}_i} - \sum_{k=1}^p \frac{\lambda_k}{m_i} \frac{\partial f_k}{\partial \mathbf{x}_i} = 0$$

束縛条件から、 $p$ 個の式を得る。

$$\sum_{i=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i} \mathbf{v}_i = 0$$

これをもう一度時間微分して、

$$\sum_{i=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i} \frac{d\mathbf{v}_i}{dt} + \sum_{i=0}^N \sum_{j=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \mathbf{v}_i \mathbf{v}_j = 0$$

この式に運動方程式を代入すると、

$$\sum_{i=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i} \frac{1}{m_i} \left( -\frac{\partial g}{\partial \mathbf{x}_i} + \sum_{l=1}^p \lambda_l \frac{\partial f_l}{\partial \mathbf{x}_i} \right) + \sum_{i=0}^N \sum_{j=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \mathbf{v}_i \mathbf{v}_j = 0$$

最終的に未定乗数 $\lambda_l$ を与える以下の方程式が得られる。

$$\sum_{l=1}^p \left( \sum_{i=0}^N \frac{1}{m_i} \frac{\partial f_k}{\partial \mathbf{x}_i} \frac{\partial f_l}{\partial \mathbf{x}_i} \right) \lambda_l - \sum_{i=0}^N \frac{1}{m_i} \frac{\partial f_k}{\partial \mathbf{x}_i} \frac{\partial g}{\partial \mathbf{x}_i} + \sum_{i=0}^N \sum_{j=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \mathbf{v}_i \mathbf{v}_j = 0$$

これは、

$$a_{kl} = \sum_{i=0}^N \frac{1}{m_i} \frac{\partial f_k}{\partial \mathbf{x}_i} \frac{\partial f_l}{\partial \mathbf{x}_i}, \quad b_k = \sum_{i=0}^N \frac{1}{m_i} \frac{\partial f_k}{\partial \mathbf{x}_i} \frac{\partial g}{\partial \mathbf{x}_i} - \sum_{i=0}^N \sum_{j=0}^N \frac{\partial f_k}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \mathbf{v}_i \mathbf{v}_j$$

と定義すると、次の連立1次方程式であり、

$$\sum_{l=1}^p a_{kl} \lambda_l = b_k$$

未定乗数  $\lambda_l$  が求められる。これを用いて運動方程式を解くことができる。

## 2. プログラムの動作

メニュー「分析－教育・科学他－物理シミュレーション－質点系の運動」を選択すると、図 1 のような実行メニューが表示される。

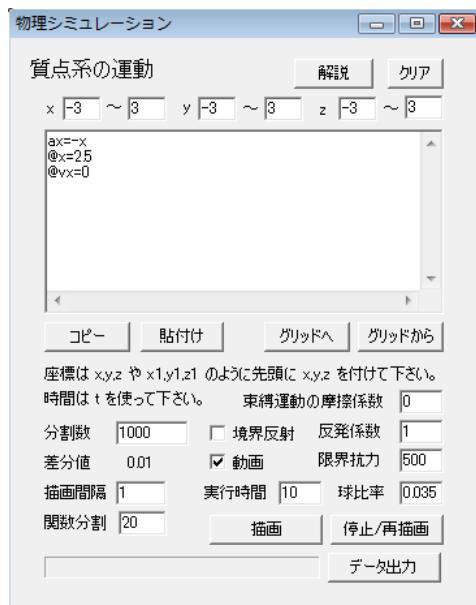


図 1 実行メニュー

プログラムを実行するには、まず、最上部の「x」「y」「z」となっている描画範囲を指定する。その下の大きなテキストボックスに描画コマンドを記述する。プログラムの概要とコマンドの書き方などについては、「解説」ボタンをクリックすると図 2 のように表示される。

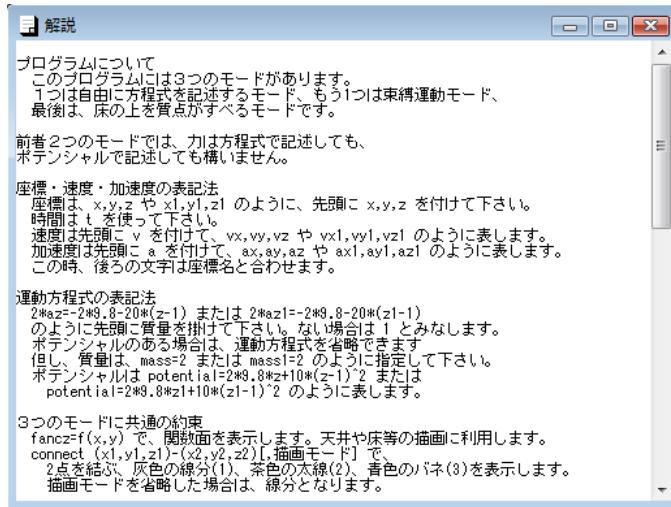


図 2 解説画面

サンプルプログラム（Sample.zip の中の質点系の運動という名前の付いたファイル）には様々な例がある。ここでは、最初に解説画面を元にコマンドの記述法とその利用法を解説し、その後、例を元に具体的に実行メニュー中の設定について説明する。

このプログラムには大きく分けて 3 つのモードがある。1 つは束縛のないモード（通常モードと呼ぶ）、もう 1 つは束縛運動モード（束縛モードと呼ぶ）、最後は床の上を質点が滑るモード（床上モードと呼ぶ）である。床上モードは、束縛モードのひとつであるが、このプログラムでは、質点のジャンプやバウンドを考えているので、特別に分けています。通常モードと束縛モードには、質点に働く力をそのまま運動方程式に記述する場合とポテンシャルで記述する場合の 2 種類がある。一般に複数の質点を考えるが、床上モードでは、特別に、質点は 1 つ、力の方向は z 軸下向きと規定している。

コマンド入力について、最初に、座標、速度、加速度の表記法を示す。まず時間は必ず  $t$  を使う。座標は、 $x, y, z$  や  $x1, y1, z1$  のように、先頭に  $x, y, z$  を付ける。速度は先頭に  $v$ 、続けて座標の名前を付けて、 $vx, vy, vz$  や  $vx1, vy1, vz1$  のように表す。加速度は先頭に  $a$ 、続けて座標の名前を付けて、 $ax, ay, az$  や  $ax1, ay1, az1$  のように表す。

運動方程式は（ポテンシャルを記入しない場合）、以下のように記述する。

$$2*az=-2*9.8-20*(z-1)$$

これは重力場中のばね運動を表している。左辺は質量×加速度で、先頭に質量（この場合 2）を付けて記述する。質量の記述がない場合は 1 とみなす。方程式は 3 次元すべて記入する必要はない。

同じ問題をポテンシャルで記述すると以下となる。

```
potential=2*9.8*z+10*(z-1)^2
```

```
mass=2
```

ポテンシャルで記述する場合、方程式を入力する必要はないが、質量が分離できないので、必ず質量を指定する。指定の方法は質点の座標が  $(x, y, z)$  の場合 `mass=1`、 $(x_1, y_1, z_1)$  の場合 `mass1=1` などとする。

直接的な運動の記述以外の、描画に関するコマンドを紹介する。これらのコマンドは3つのモードで共通に使える。

実行結果に天井や床などを加えたい場合は、以下のように指定する。

```
funcz=f(x, y)
```

このコマンドで、 $z=f(x, y)$  で表わされる関数面を表示する。但し、 $z$  方向に極端に傾いた部分は表示できない。

実行結果に糸やバネを加えたい場合は、以下のように指定する。

```
connect (x1, y1, z1)-(x2, y2, z2) [, 描画モード]
```

描画モード 1:線分, 2:太い線分, 3:バネ

このコマンドにより、描画に座標  $(x_1, y_1, z_1)$  と  $(x_2, y_2, z_2)$  を結ぶ、線分（灰色）、太い線分（茶色）、バネ（青）を追加する。どれを描画するかは描画モードの数字によって使い分ける。描画モードを指定しない場合は線分となる。

運動方程式には必ず初期条件が必要である。初期条件は以下のように、先頭に `@` を付けて指定する。但し、方程式にない変数については指定しない（これらの変数については常に0の値を取る）。

```
@x1=0
```

```
@vx1=1
```

`define` コマンドで、運動方程式やポテンシャルなどの複雑な式を以下のように分けて表示することができる。

```
define s=(x1^2+y1^2+(z1-3)^2)^0.5
```

```
potential=2*9.8*z1+10*(s-2)^2
```

床上モードでは、質点への重力加速度は  $z$  方向に働き、床面は以下のように指定する。床上モードで扱える質点は1個である。

```
floorz=f(x, y)
```

重力加速度は以下のように指定する。

```
gravz=g(x, y, z)
```

運動方程式は入力せず、初期値のみ入力する。変数名には必ず  $x, y, z$  を使う。

以後、例を見ながら説明をする。実行はテキストボックスにコマンドを打ち込むか、グリッドにあるデータを「グリッドから」ボタンで読み込んで、「描画」ボタンをクリックする。運動を一時停止する場合や、再度運動を開始する場合は、「停止/再描画」ボタンをクリックする。また、質点の軌道を表示させる場合は、実行中（一時停止中）に「動画」チェックボックスのチェックを外し、「停止/再描画」ボタンをクリックする。質点の軌道の数値データを出力したい場合は、「データ出力」ボタンをクリックする。時間、質点の位置、質点の速度、未定乗数（床上モードの場合は抗力）の値が表データで出力される。

### 例 1 楕円運動

```
# 楕円運動
connect (0, 0, 0)-(x1, y1, z1)
ax1=-2*x1/(x1^2+y1^2)^1.5
ay1=-2*y1/(x1^2+y1^2)^1.5
@x1=3
@y1=0
@vx1=0
@vy1=0.5
```

これは原点からの距離の逆 2 乗則による楕円軌道を表している。力の方向を明らかにするために、原点と質点を線で結んである。行の先頭に#を付けるとコメント行になる。図 3a に実行例を示す。また、図 3b に「動画」チェックボックスのチェックを外して描いた軌道を示す。

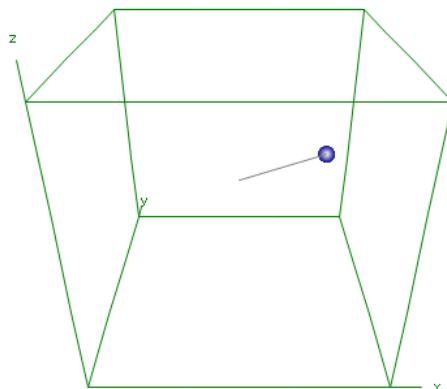


図 3a 質点の楕円運動

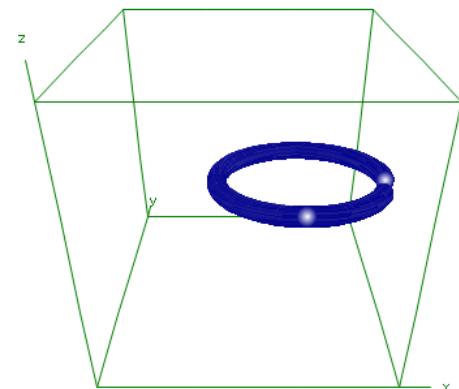


図 3b 質点の楕円運動の軌道 1

質点の大きさは、「球半径」テキストボックスで、上の図はデフォルトの 0.2 である。軌道表示などでは小さく設定した方がよい。また、描画のスピードは「描画間隔」テキストボックスで、計算結果

をいくつおきに表示するか決める。またこれは軌道表示へも影響する。図 3c に「球半径」を 0.1 に、描画間隔を 20 に設定した例を示す。

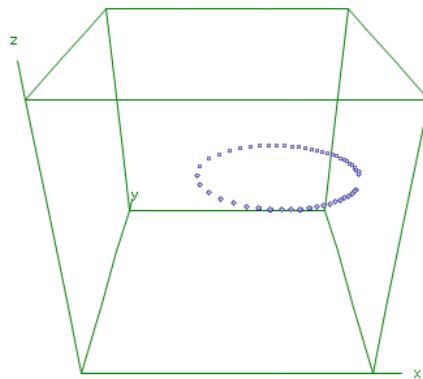


図 3c 質点の椭円運動の軌道 2

## 例 2 单振子

```
funcz=3
connect(0, 0, 3)-(x1, 0, z1)
ax1=-(9.8*(3-z1)+(vx1^2+vz1^2))*x1/25
az1=-(9.8*x1^2-(3-z1)*(vx1^2+vz1^2))/25
@x1=5*sin(pi/4)
@vx1=0
@z1=3-5*cos(pi/4)
@vz1=0
```

これは  $x$ - $z$  面での糸の長さ 5 の单振子を表している。運動方程式には、円運動をするための糸の張力を加えてある。描画には、天井  $z=3$  と糸（運動している場合のみ）を追加している。図 4a に実行結果、図 4b に軌道を表示する。

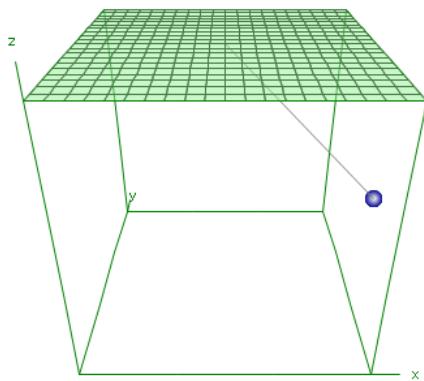


図 4a 単振子

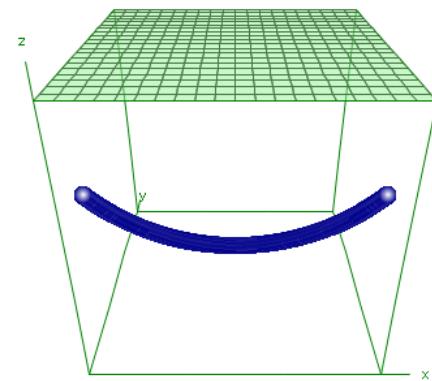


図 4b 単振子の軌道

この運動は以下のように束縛運動としても記述できる。

```

funcz=3
connect (0, 0, 3)-(x1, y1, z1)
bind x1^2+y1^2+(3-z1)^2-25=0
ax1=0
ay1=0
az1=-9.8
@x1=5*sin(pi/4)
@vx1=0
@y1=0
@vy1=0
@z1=3-5*cos(pi/4)
@vz1=0

```

bind の式が束縛条件を表している。方程式の解法は Lagrange の未定乗数法を利用している。このようにした場合、y 方向の運動にも対応可能である。図 4c に y 方向の速度の初期値を 2 に書き換えた場合 (@vy1=2) の z 方向から見た軌道を示しておく。

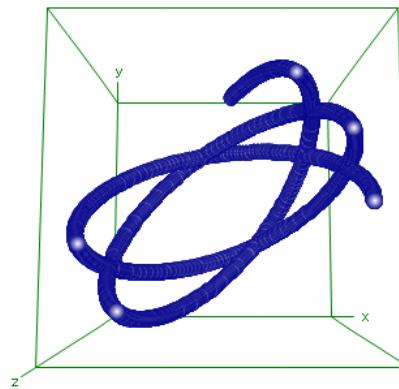


図 4c 上から見た 3 次元の单振子の軌道

Lagrange の未定乗数法を利用する場合、初期条件の与え方に注意する必要がある。束縛条件を破る形で与えた場合、束縛条件は全く意味をなさなくなる。何故なら、未定乗数法の場合、例えば距離一定の束縛条件は、距離方向の力を打ち消すように働き、その方向に初速度がある場合、決してそれにならないからである。試しに  $z$  方向に  $@vz1=0.3$  で初速度を与えた場合、糸の長さは短くなつて行き、計算は破たんする。この状況を図 4d に示す。

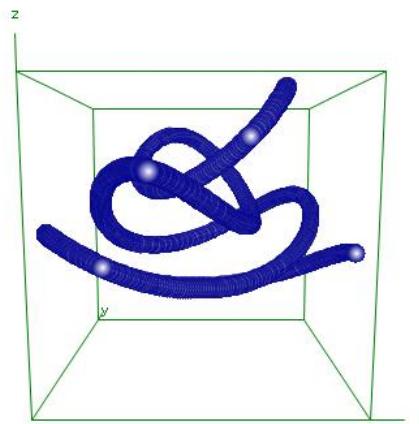


図 4d 束縛条件を破る方向に初速度を与えた場合の計算結果

### 例 3 2 連振子

```
funcz=3
connect (0, 0, 3)-(x1, y1, z1)
```

```

connect (x1, y1, z1)-(x2, y2, z2)
bind x1^2+y1^2+(3-z1)^2-9=0
bind (x1-x2)^2+(y1-y2)^2+(z1-z2)^2-4=0
ax1=0
ay1=0
az1=-9.8
2*ax2=0
2*ay2=0
2*az2=-19.6
@x1=3*sin(pi/6)
@vx1=0
@y1=0
@vy1=0
@z1=3-3*cos(pi/6)
@vz1=0
@x2=5*sin(pi/6)
@vx2=0
@y2=0
@vy2=2
@z2=3-5*cos(pi/6)
@vz2=0

```

これは、2 連振子に Lagrange の未定乗数法を適用した例である。振動には y 軸方向も加えてある。

図 5a に実行結果、図 5b に質点の軌道を示す。

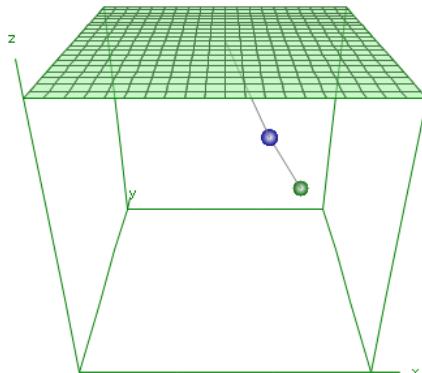


図 5a 2 連振子

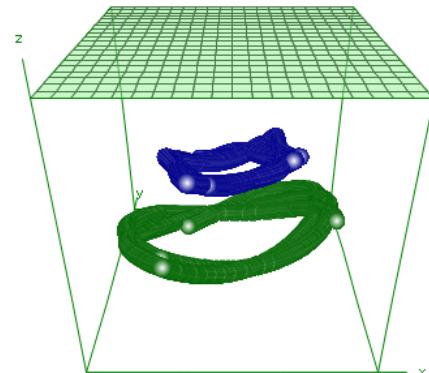


図 5b 2 連振子の軌道

#### 例 4 バネ振子

```

#define funcz=3
define s1=(x^2+y^2+(3-z)^2)^0.5
connect (0, 0, 3)-(x, y, z), 3

```

```

potential= 9.8*z+4*(s1-2)^2
mass=1
@x=2*sin(pi/4)
@vx=0
@y=0
@vy=1
@z=3-2*cos(pi/4)
@vz=0

```

これはバネ振子の場合の例である。バネ振子には束縛条件は付かない。すべてポテンシャルで記述する。ここではポテンシャルを記述する場合、バネの長さを  $s1$  と定義して利用している。たくさんのバネなどが繋がっている場合、ポテンシャルの記述が長くなるので、簡単な文字列で定義しておくと分かり易い。図 6a に実行結果、図 6b に質点の軌道を示す。

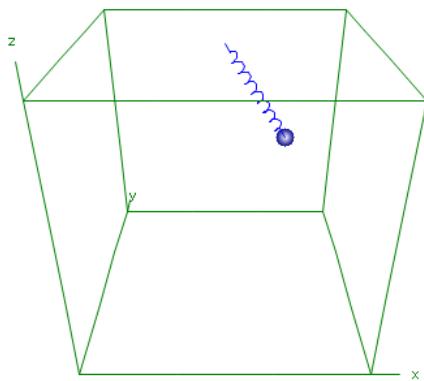


図 6a バネ振子

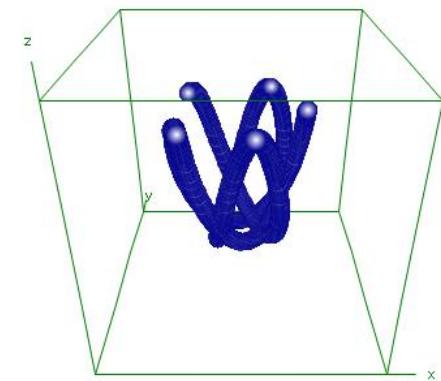


図 6b バネ振子の軌道

このバネの描画には前に示した(1)式を  $\alpha = \varphi$  として利用している。

### 例 5 床上運動

```

floorz=-1.5*(sin(x+2)+cos(y))
funcz=-1.5*(sin(x+2)+cos(y))-0.5
gravz=-9.8
@x=0
@vx=0
@y=0.5
@vy=0
@z=3
@vz=0

```

これは、 $floorz=-1.5*(\sin(x+2)+\cos(y))$  で与えられた面への質点の自由落下（または面上での滑り運動）を表している。その際、反発係数と面上での滑り運動を起こす際の摩擦係数とを実行メニュー

一で設定する。重力加速度は  $gravz=-9.8$  で与えられ、下向き 9.8 である。反発係数 0.7、摩擦係数 0.2 とした場合の結果を図 7a、質点の軌道を図 7b に示す。

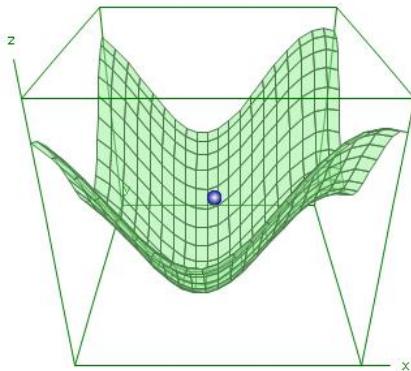


図 7a 床上運動

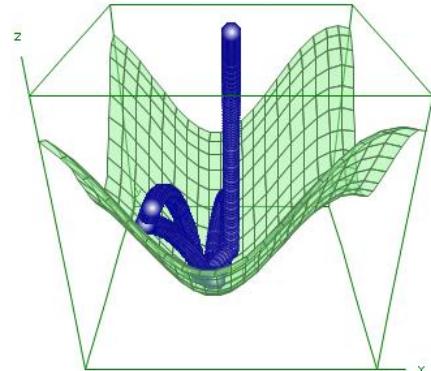


図 7b 床上運動の軌道

この問題では、床からの抗力を計算し、抗力が負になった時点で自由落下に切り替える、自由落下の床と垂直方向の速さが小さくなかった場合は滑り運動に切り替える、床と垂直方向の速さが大きい場合は反発係数に応じた反射をするなどの条件を加えて、できるだけ、実際の運動に近づけたシミュレーションである。ただ、多少の計算誤差も含まれており、反発係数を 1 (0.99 位で 1 に相当) にすると少し反発が強いようである。

#### 例 6 バネ振子の組み合わせ

```

define 11=((x1+3)^2+y1^2+(z1-3)^2)^0.5
define 12=((x2-x1)^2+(y2-y1)^2+(z2-z1)^2)^0.5
define 13=((3-x2)^2+y2^2+(3-z2)^2)^0.5
define 14=((x3-x1)^2+(y3-y1)^2+(z3-z1)^2)^0.5
define 15=((x4-x2)^2+(y4-y2)^2+(z4-z2)^2)^0.5
connect (-3, 0, 3)-(x1, y1, z1), 3
connect (x1, y1, z1)-(x2, y2, z2), 2
connect (x2, y2, z2)-(3, 0, 3), 3
connect (x1, y1, z1)-(x3, y3, z3), 3
connect (x2, y2, z2)-(x4, y4, z4), 3
potential=9.8*z1+9.8*z2+9.8*z3+9.8*z4+32*(11-2)^2+32*(13-2)^2+16*(14-1)^2+16*(15-1)^2
bind 12-2=0mass1=1
mass2=1
mass3=1
mass4=1
@x1=-1
@vx1=1

```

```

@y1=0
@vy1=0
@z1=3
@vz1=0
@x2=1
@vx2=0
@y2=0
@vy2=0
@z2=3
@vz2=0
@x3=-1
@vx3=1
@y3=0
@vy3=1
@z3=2
@vz3=0
@x4=1
@vx4=0
@y4=0
@vy4=-1
@z4=2
@vz4=0

```

これはバネ振子を組み合わせた質点系の例である。結果を図 8a、あまり意味がないが、質点の軌道を図 8b に示す。

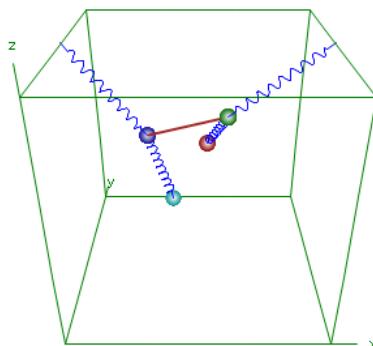


図 8a バネ振子の組み合わせ

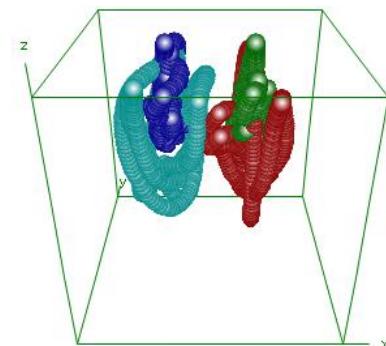


図 8b バネ振子の組み合わせの軌道

上の 2 つの質点の間の線は、堅い棒を表している。これは `connect (x1, y1, z1)-(x2, y2, z2), 2` で描かれている。この棒は束縛条件で、Lagrange の未定乗数法により表現できるが、バネ定数を大きくして（実際の計算上、他のものの数百倍程度が限界、それ以上はエラーの原因となる）ポテンシャル問題として表してもよい。後者の方が計算時間は幾分短い。

最後に実行メニューのまだ説明をしていない機能について述べる。「分割数」テキストボックスと「実行時間」テキストボックスの関係であるが、運動の速さにもよるが、「差分値」の値が 0.01 以下になるように選ぶことが望ましい。束縛運動では計算時間が長くかかるため、注意する必要がある。描画は 0.02 秒に 1 回で設定しているので、「描画間隔」が 1 の場合、差分値の値が 0.02 で描画時間が実時間になるはずであるが、描画スピードが追い付けない場合は時間が長くなる。

「関数分割」テキストボックスは、天井や床を描画する関数の分割数を表す。ある座標軸について、関数の描画要素の長さは、描画範囲 ÷ 関数分割数である。「境界反射」チェックボックスは、質点が描画範囲の境界で反射されるかどうかを表す。反射は、境界の垂直成分について反発係数 1 としている。その他の成分について速度変化はない。「限界抗力」テキストボックスについては、床上運動の場合に対応できる抗力の上限を与えている。抗力が大きすぎると計算誤差が生じるので、ある段階で計算を停止させている。

## 12. 電流と磁場

ここでは定常電流によって作られる磁場とその中の荷電粒子の運動について考える。簡易言語を用いて、電流を3次元パラメータ関数によって表し、それによって作られる磁力線や磁場を電流も含めて3Dビューア上に表現する。その際、磁場中の荷電粒子の運動についてもシミュレーションできるようになる。

微小な長さの定常電流のつくる磁場  $d\mathbf{H}$  [A/m] は、以下のビオ・サバール (Biot-Savart) の法則によって与えられる。

$$d\mathbf{H} = \frac{1}{4\pi} \frac{Id\mathbf{s} \times \mathbf{r}}{r^3}$$

ここに、 $I$  [A] は電流、 $d\mathbf{s}$  [m] は電流の微小な長さ、 $\mathbf{r}$  [m] は電流から磁場を作る点への位置ベクトルである。微小な長さの定常電流のつくる真空中の磁束密度  $d\mathbf{B}$  [Wb/m<sup>2</sup>] は、真空の透磁率を  $\mu_0 = 4\pi \times 10^{-7}$  [H/m] とすると、以下となる。

$$d\mathbf{B} = \mu_0 d\mathbf{H} = \frac{\mu_0}{4\pi} \frac{Id\mathbf{s} \times \mathbf{r}}{r^3}$$

ここに、ある地点における磁場や磁束密度はこれらの足し合わせによって得られる。

磁束密度  $\mathbf{B}$  の空間で、電荷量  $q$  [C]、速度  $\mathbf{v}$  [m/s] の荷電粒子に働く力  $\mathbf{F}$  [N] は、以下で与えられる。

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}$$

ここでのシミュレーションはこれらの関係を利用して計算される。

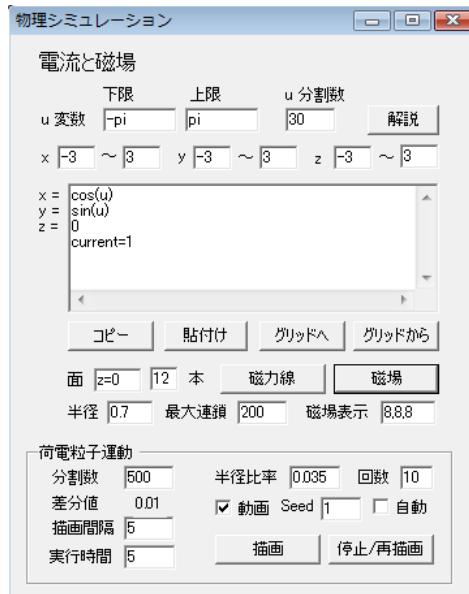


図 1 分析メニュー

電流は「u 変数」で与えられるパラメータによって、パラメータ関数として式入力用のテキストボックスに記述する。u 変数の「下限」と「上限」はパラメータの設定範囲、「u 分割数」は、関数の微小量を表すが、この場合には、電流の微小量である。分析メニューの例は、デフォルトで与えられるもので、後に述べる円形コイルを表す。磁力線や、磁場、粒子の運動の描画範囲は、「x」、「y」、「z」で与えられ、初期値はいずれも-3m～3m となっている。

磁力線を表示するには、まず描画を開始する「面」を記述する。面は、z=0 のような式で表わすが、磁力線は、その中の描画領域の中央を中心とする「半径」（デフォルトで 0.7）で示される円内でバランスが取れるように（0.5×「半径」のところに本数の 1/3、「半径」のところに残りの本数）開始点が設定される。磁力線の本数は初期設定では 12 本になっている。微小な長さの磁力線の「最大連鎖」は 200、または描画領域外に到達するまでである。以後の図はすべて磁力線のデフォルト値を用いている。

### 例 1 円形コイル

```
cos(u)
sin(u)
0
current=1
```

これは円形コイルの例である。上から 3 行はパラメータ関数で表わした電流の経路で、current=1 は電流が 1 A 流れていることを示す。メニュー中央のテキストボックスに上の設定を書き込み、「磁

「力線」ボタンをクリックすると図 2 のような円形コイルの作る磁力線が表示される。

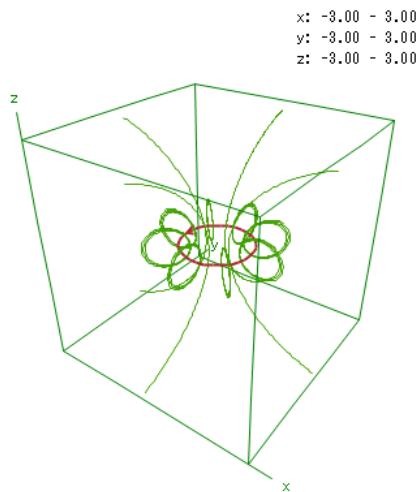


図 2 円形コイルの磁力線

ここに、current=1 は、電流が 1A 流れていることを示す。中心の茶色の円が電流を表す。電流の向きは、電流上に矢印で示されている。円形コイルの場合、本来磁力線は閉じた曲線で表わされるが、ここでは誤差があるため、曲線が少しずれている。

### 例 2 バネ状コイル

```
cos(10*u)
sin(10*u)
u/2
current=1
```

これはバネ状コイルを表している。このコイルの作る磁力線を図 3 に表示する。ここで、u 分割数は 150 に設定している。

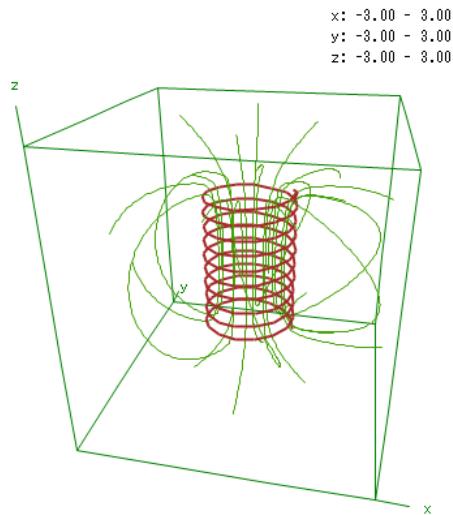


図 3 バネ状コイルの磁力線

## 例 3 2 つの円形コイル

```

cos(u)
sin(u)
0
-3
3*cos(u)
3*sin(u)
current=1,1
  
```

これは 2 種類の円形コイルを表している。電流は上から 3 行ずつを 1 組にして、2 種類の経路を示している。current=1, 1 は経路順に、2 つの電流の強さを与えている。最後に 2 つの円形コイルの作る磁力線を図 4 に示す。

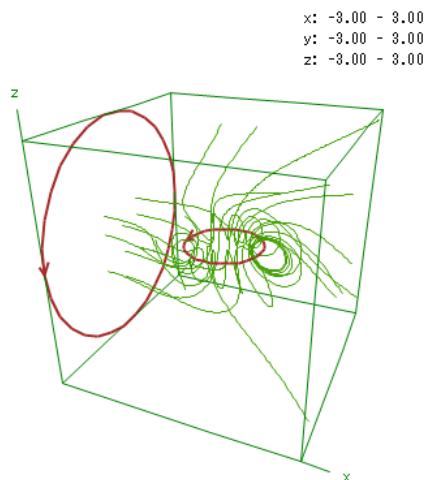


図 4 2つの円形コイルが作る磁力線

磁場は、領域内で x, y, z 軸方向をそれぞれ、「磁場表示」で指定した回数で分割して、その分割点に表示する。「磁場表示」を 8,8,8 とした例 1 の円形コイルの場合の磁場を図 5a に、「磁場表示」を 8,1,8 として、描画領域で y を 0 – 0 に指定した平面表示を図 5b に示す。矢印の中央が磁場を求める点である。

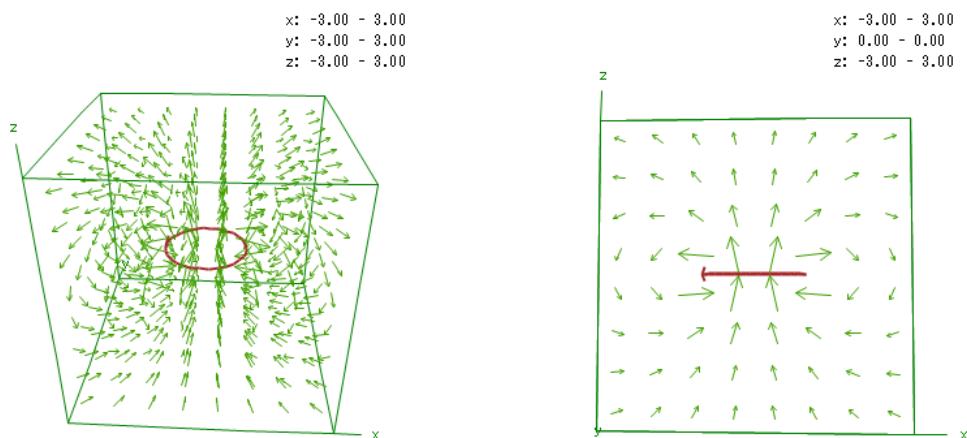


図 5a 円形電流の磁場

図 5b 円形電流の平面表示

#### 例 4 円コイルの作る磁場の中の荷電粒子の運動 1

$$\begin{aligned} \cos(u) \\ \sin(u) \end{aligned}$$

```

0
current=1
mass=0.0000001
charge=1
@x=3
@y=0
@z=0
@vx=-2
@vy=0
@vz=0

```

これは円形コイルの中を荷電粒子が運動するプログラムである。この荷電粒子の運動を見るためには、粒子の質量と電荷、及び初期位置と初速度が必要になる。最初の 4 行は電流を表すコマンドであり、「mass=0.0000001」と「charge=1」は、粒子の質量 [kg] と電荷 [C] を表す。粒子の初期値は、`@x=`, `@y=`, `@z=`, `@vx=`, `@vy=`, `@vz=` の形で入力する。図 6 に上の設定で、「動画」 チェックボックスを外して、「描画」 ボタンをクリックした描画結果を示す。画面には粒子の軌道が描かれている。ここで粒子の「粒子半径」は 0.2 している。「動画」 チェックボックスにチェックを入れると、粒子の運動を見ることができる。

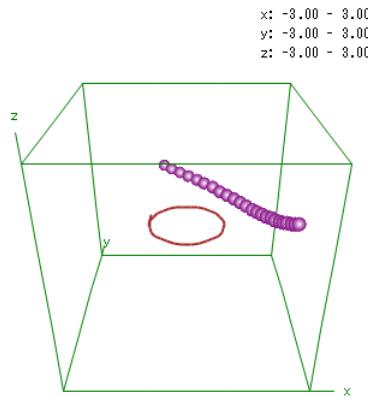


図 6 磁場中の正電荷の運動

#### 例 5 円形コイルの作る磁場中の荷電粒子の運動 2

```

cos(u)
sin(u)
0
current=1
mass=0.0000001
charge=-1+2*theta(rnd-0.5)
@x=3

```

```

@y=0
@z=0
@vx=-1
@vy=0
@vz=-1+2*rnd

```

これは電荷を $\pm 1C$  の乱数で与え、 $x$  方向から粒子を初速度  $1m/s$  で入射させ、 $z$  方向の初速度を  $-1m/s \sim 1m/s$  の一様乱数に設定した場合の例である。乱数は「Seed」で乱数のシードを固定したり、「自動」で変動したりすることができる。結果を図 7a に示す。ここで、電荷が正の荷電粒子は紫、負の荷電粒子は緑で表わしている。荷電粒子が磁場ではじかれている様子がよく分かる。

同様にして、 $z$  方向から粒子を初速度  $1m/s$  で入力させ、 $y$  方向の初速度を上と同様に変動させた例を図 7b に示す。

```

@x=0
@y=0
@z=3
@vx=0
@vy=-1+2*rnd
@vz=-1

```

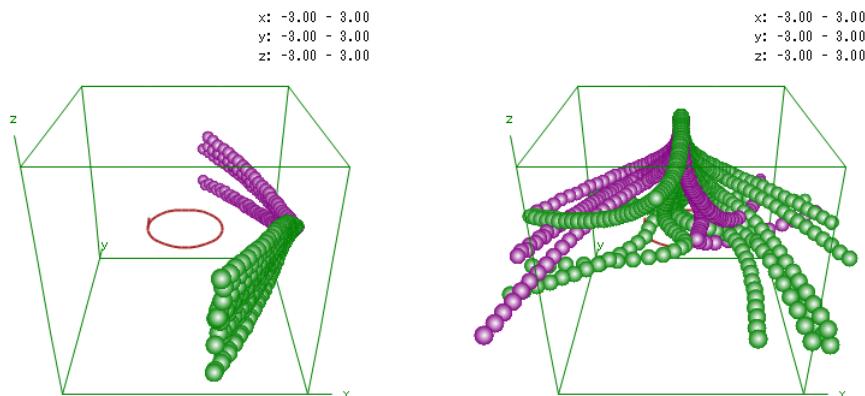


図 7a 荷電粒子の軌道（x 方向）

図 7b 荷電粒子の軌道（z 方向）

### 13. 特殊相対論的視覚効果

我々は3次元グラフや物理シミュレーションの表示用に、College Analysis の中で3Dビューアを開発した。今回はこれを用いて、動いている物体の長さや時間が変化する、特殊相対性理論の世界を視覚的に表現することを試みる。

光速に近い速さで動く物体はローレンツ収縮し、運動方向の長さが縮むことはよく知られている。しかし、これがそのまま観測者に見えるわけではない。厳密にいうと、物体の各部位から発せられた光が観測者に達する時刻（見える時刻）を同一に考えると、光が発せられる時刻は、物体の部位によって異なってくる。例えば、物体の観測者に近い部位からの光は、遠い部位からの光に比べて遅く発せられた光である。通常この差はごくわずかで、その間の物体の移動は当然無視できるが、光速に近い速さで動く物体の場合、この差は無視できなくなる。即ち、物体が近づく場合、物体の前方は（遅く光が出るわけであるから）、観測者にとってより近く見える。これは、いわゆる「鼻デカワソコ」の写真に似ている。ただこの映像が見えた刹那（10mの距離で見たとして、約  $3 \times 10^{-8}$  秒後）、観測者または観測機器の命運は尽きる。

このプログラムではまた、光のドップラー効果についても視覚効果を入れている。一般に物体が近づく場合青方変異、遠ざかる場合赤方変異となることが知られている。青方変異の場合、赤色の可視光は青くシフトするが、その分赤外線が赤色にシフトしてくる。我々はこの赤外線の成分について知識がないので、可視光の範囲内だけで考えており、実際の色を再現しているわけではない。あくまで色が変わる効果があるという程度のものである。

このようなことから、これは物理シミュレーションとは言いづらい。そのため、論文としては物理シミュレーションのシリーズに入れず、単独の研究報告とした。とにかく、この不思議な感覚を体験してもらえば幸いである。

特殊相対性理論は、互いに相対的に運動する慣性系の間での光速度不変の原理と特殊相対性原理を基にして、構成される。それによると1つの慣性系 S (S系と呼ぶ) に対して z 方向に速さ  $v$  で運動する慣性系 S' (S'系と呼ぶ) のそれぞれの座標  $(t, x, y, z)$  と  $(t', x', y', z')$  は以下の Lorentz 変換によって関係づけられる。ここに  $c$  は光速度を表す。

$$t' = \frac{t - (v/c^2)z}{\sqrt{1 - (v/c)^2}}, \quad x' = x, \quad y' = y, \quad z' = \frac{z - vt}{\sqrt{1 - (v/c)^2}}$$

まず、動いている棒を観測する問題を考える。S'系の長さ  $z'_2 - z'_1$  を S 系で観測する場合、S 系においては同時刻に測ることが必要であるから、 $t$  を同じ値として、

$$z'_2 - z'_1 = \frac{(z_2 - vt) - (z_1 - vt)}{\sqrt{1 - (v/c)^2}} = \frac{z_2 - z_1}{\sqrt{1 - (v/c)^2}}$$

より、 $l = l_0\sqrt{1 - (v/c)^2}$  となる。ここに、 $l_0 \equiv z'_2 - z'_1$  は静止系で測定した距離、 $l \equiv z_2 - z_1$  は運動している系で測定した距離を表している。すなわち、動いている棒には Lorentz 収縮が起こる。

時間については、同時刻は同じ位置で測定して、それを他の系から眺める必要があるため、上の Lorentz 変換を逆に考えて、

$$t_2 - t_1 = \frac{\{t'_2 + (v/c^2)z'\} - \{t'_1 + (v/c^2)z'\}}{\sqrt{1 - (v/c)^2}} = \frac{t'_2 - t'_1}{\sqrt{1 - (v/c)^2}}$$

より、 $\tau = \tau_0/\sqrt{1 - (v/c)^2}$  となる。ここに、 $\tau_0 = t'_2 - t'_1$  は静止系で測定した時間、 $\tau = t_2 - t_1$  は運動している系で測定した時間を表している。これは時間の遅れを意味する。これらは、光速度不変の原理を基に求められる結論である。

光速度に近い速さで運動する物体を観測する場合、特殊相対論的効果だけでなく、光速度が有限に感じられるようになることの効果も大きい。即ち、運動している物体の 2 点からの光を同時に観測した時、観測者から近い点はより遅く、観測者から遠い点はより早く光を出しているはずで、その効果によって 2 点の位置が、静止しているときよりずれて見える。例えば物体が観測者に向かって動いている場合、観測者に近い側は遠い側に比べてより近くに見えるはずである。この効果は物体が光速に近い速さになれば顕著となる。我々のプログラムは、物体が観測者に対して角度を持って、近づいたり遠ざかったりする場合にも対応している。

この Lorentz 収縮の効果と有限光速度の効果を計算する。まず図 1 に光速に近い速さ  $v$  で、観測者 ( $z$  方向) に対して角度  $\theta$  で近づく物体 (四角形) とその Lorentz 収縮を与える。

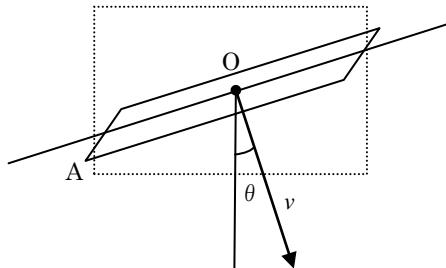


図 1 Lorentz 収縮

この Lorentz 収縮は物体を一度、角度  $-\theta$  だけ回転させて、 $z$  方向で Lorentz 収縮させ、もう一度角度  $\theta$  戻すことによって簡単に得ることができる。

次に、観測点  $Z_0 (0, 0, z_0)$  に点  $O (0, 0, o_z)$  からの光が届く際に点  $A (x, y, z)$  からの光はどの位置から発せられるかを考える。その位置を  $A' (x', y', z')$  とする。その位置関係を図 2 に表す。

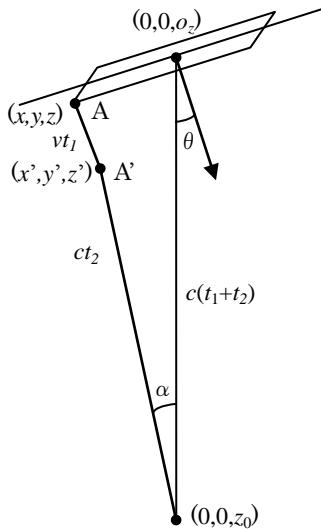


図 2 光速度有限化の影響

ここに  $y$  方向は紙面に垂直な方向とする。

この関係を式で書くと、以下となる。

$$\begin{aligned}
 x' &= x + vt_1 \sin\theta \\
 y' &= y \\
 z' &= z + vt_1 \cos\theta \\
 (ct_2)^2 &= x'^2 + y'^2 + (z_0 - z')^2 \\
 c(t_1 + t_2) &= z_0 - o_z
 \end{aligned}$$

簡単のため  $c = 1$  の単位系を用いてこれをまとめると、 $t_1$  に対して、以下の 2 次方程式を得る。

$$(1 - v^2)t_1^2 - 2[(z_0 - o_z) + \{(z - z_0)\cos\theta + x\sin\theta\}v]t_1 + [(z_0 - o_z)^2 - \{x^2 + y^2 + (z - z_0)^2\}] = 0$$

この解のうち、小さい方は物体の進行方向と同じ方向から光が到着する解で、もう 1 つは逆向きから到着する解である。プログラムはこれらの解を用いて、光の発射位置  $A'$  を求めている。

次に相対論的な効果として求められるものは、物体が光の速さに近づく際の光のドップラー効果である。速度  $v$  で移動する光源から振動数  $f$  で発射された光が、図 2 のように観測者に対して角度  $\alpha$  で入射する際の、振動数  $f'$  は以下のように与えられる。

$$f' = f \frac{\sqrt{1 - v^2}}{1 - v \cos\alpha}$$

ここで、光のドップラー効果は、光源の運動による時間の遅れが含まれており、角度 90 度からの光でもドップラー効果を受ける。

我々はこの効果をイメージするため、可視光のみ考え、色を光の 3 要素に分解し、それぞれのドップラー効果による色の変化を、再度光の 3 要素に分解して、強度に応じて足し合わせ、新しい色を作

り上げている。この場合強度が設定された許容範囲（0～255）を超える場合もあるが、許容範囲で打ち切りしている。また、紫は純粋な短波長のものと赤と青の合成されたものと区別が付かないとして、後者を採用した。おそらく現実の色とは違が生じているであろうが、可視光のみ考えるという設定からして現実的でないので、あくまでドップラー効果のイメージとして捉えてもらいたい。

相対論的な効果を表すには、まず3Dビューアを用いて3Dモデルや3次元関数グラフを描画する。例として図3に  $z = \sin x + \cos y$  の形のグラフを示す。

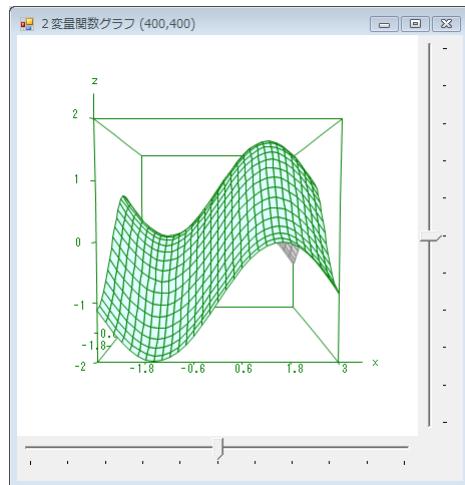


図3 2変量関数グラフ

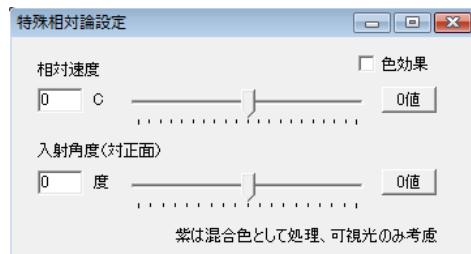
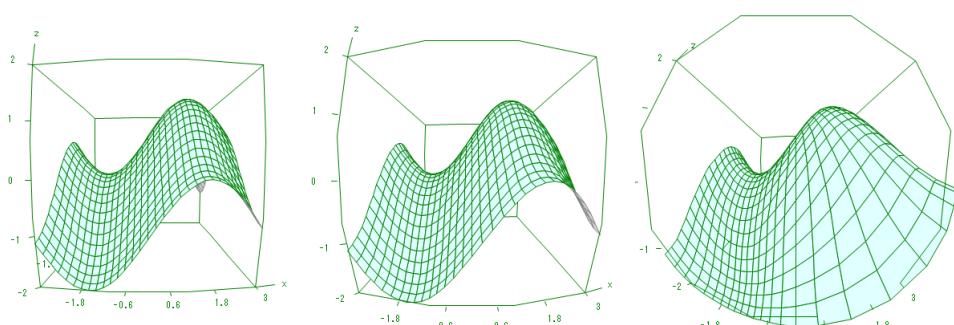


図4 特殊相対論設定画面

3Dビューアのメニュー「表示-相対論的効果[ON/OFF]」を選択して、デフォルトのOFFからONに切り替えると、図4の特殊相対論設定画面が表示される。ここでは、物体の速さ  $v$  (画面前方を正、後方を負とする) と画面前面垂直方向に対する進行角度  $\theta$  (画面に向かって右が正、左が負) が設定できる。その他に、ドップラー効果による色の変化の効果を加えるかどうかのチェックボックスがある。図5に、 $\theta=0^\circ$ で、 $v=0.5, 0.7, 0.9$  の場合の相対論的な効果を示す。

図5  $\theta=0^\circ, v=0.5, 0.7, 0.9$  の場合の結果

本来、Lorentz 収縮によって物体は前後に縮むが、光速度の有限性の効果によって、前方に飛び出したように見えている。次に、図 6 に、 $\theta=0$  で、 $v=-0.5, -0.7, -0.9$  の場合の相対論的な効果を示す。

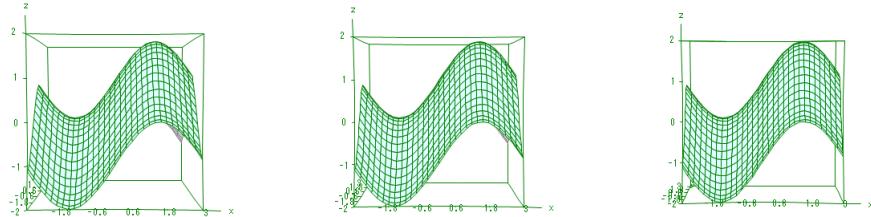


図 6  $\theta=0^\circ, v=-0.5, -0.7, -0.9$  の場合の結果

遠ざかる場合は近づく場合に比べて距離の変化は少なく、Lorentz 収縮の効果がよく見える。

次に、物体が観測者に対して角度を持って近づいて来る場合について、図 7 に  $v=0.9, \theta=30^\circ, 60^\circ, 90^\circ$  の場合の相対論的な効果を示す。

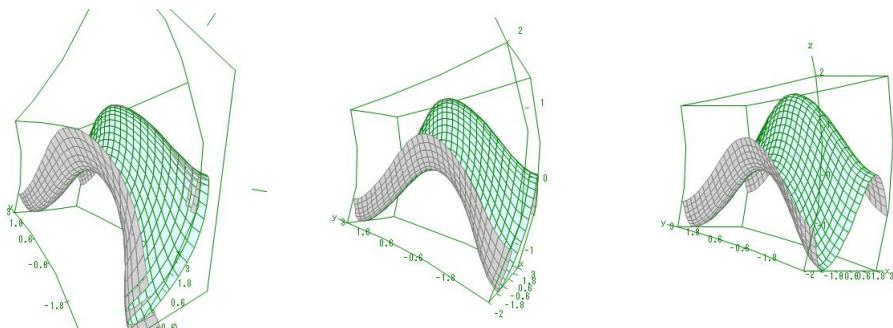


図 7  $v=0.9, \theta=30^\circ, 60^\circ, 90^\circ$  の場合の結果

ここで、 $\theta=30^\circ$ の図は多少乱れている。90°の場合は、物体は右向きに移動していることになる。図 8 に  $v=-0.9, \theta=30^\circ, 60^\circ, 90^\circ$  の場合の相対論的な効果を示す。

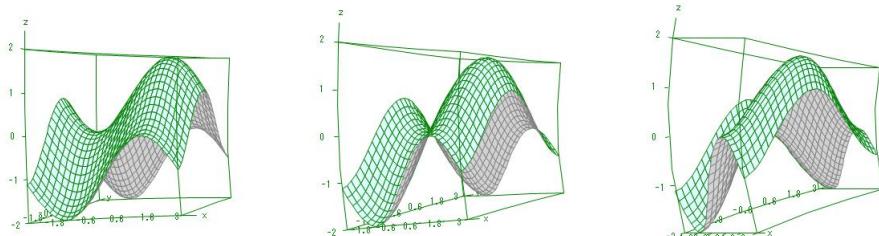
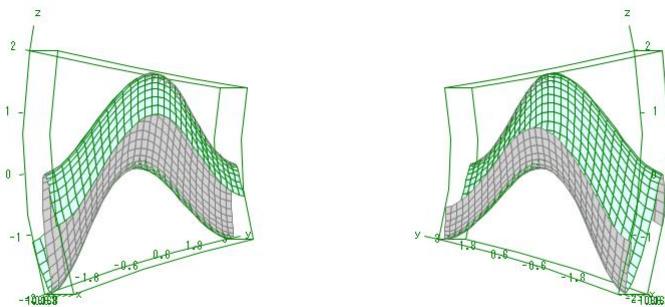


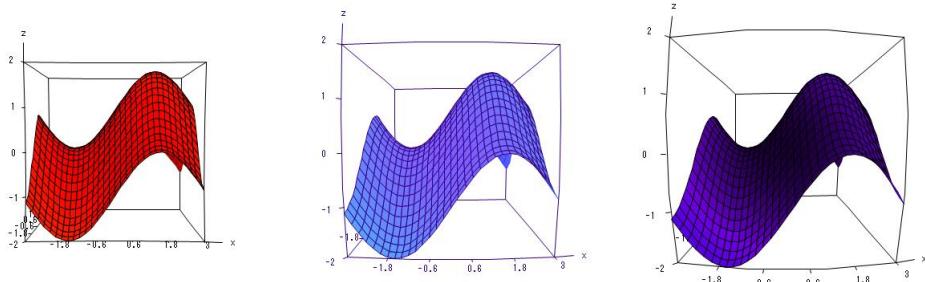
図 8  $v=-0.9, \theta=30^\circ, 60^\circ, 90^\circ$  の場合の結果

特に極端な場合、図 9 に  $v=0.99, \theta=-90^\circ, 90^\circ$  の場合の結果を示す。

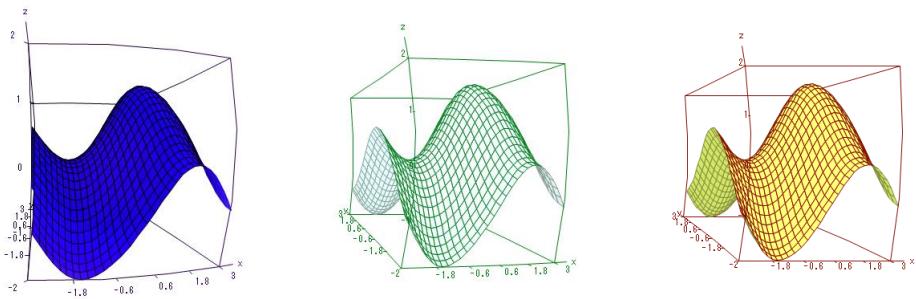
図 9  $v=0.99, \theta=-90^\circ, 90^\circ$  の場合の結果

これは Lorentz 収縮により横方向に縮むが、光の到達時間の差によって正面方向がほぼ左右を向く様子を表したものである。

次に光のドップラー効果により、色の変わらる様子を示す。この効果は図 4 のメニューで「色効果」チェックボックスをチェックすることで調べることができる。図 10 に  $\theta=0^\circ, v=-0.4, 0.35, 0.55$  の場合の結果を示す。

図 10  $\theta=0^\circ, v=-0.4, 0.35, 0.55$  の場合の光のドップラー効果

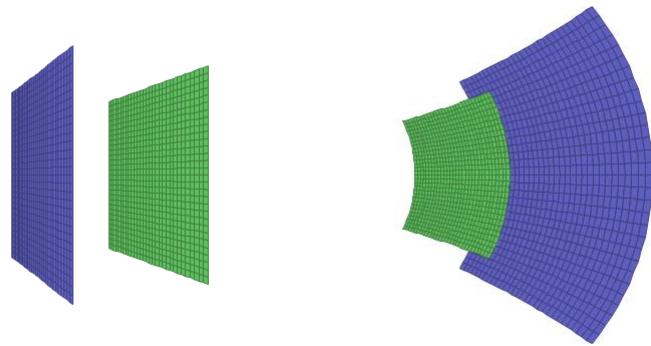
観測者への光の入射角度によりドップラー効果に影響が表れるので、右側 2 つの図では、中央の色に違いがある。これをより明らかにするために物体の進行角度を変更させる。図 11 に  $v=0.55, \theta=30^\circ, 70^\circ, 90^\circ$  の結果を表示する。

図 11  $v=0.55, \theta=30^\circ, 70^\circ, 90^\circ$  の場合の光のドップラー効果

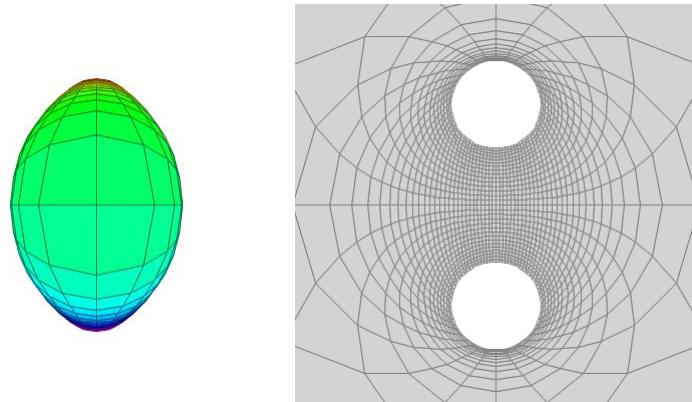
この場合は、物体の進行方向の角度により、一度  $70^\circ$ 当りでドップラー効果が無くなり、 $90^\circ$ では横ドップラー効果と呼ばれる赤色方向へのシフトが起きている。

光速度に非常に近く、ある程度角度を付けて物体が近づく場合、図 12 のように裏表が逆になって見えることがある。これはもちろん裏表が逆になっているのではなく、図形の左右反転が起こって、プログラム上の裏表の表示の規則から反転しているように見えるだけである。仮に図 13 のように、2 枚の平面を並べて、裏表の反転が起こっているときと同じ条件で見てみると、実際に左右反転が起こっていることが分かる。3D ビューアのメニュー [色一表裏[ON/OFF]] で、表裏を無くしておくといい。

図 12  $v=0, \theta=0^\circ, v=0.99, \theta=45^\circ$  の場合の結果

図 13  $v=0, \theta=0^\circ, v=0.99, \theta=45^\circ$  の場合の左右反転

最後に、近づいて来る図 12 の円筒の内側に入った場合の画像を図 14 に載せておく。2 つの出口が同時に見えるという不思議な現象が見えている。

図 14  $v=0.95, \theta=0^\circ$  の場合の入る直前と内部に入った後