

数式処理ソフトMaple入門(I)

—微積分とplot—

関西学院大学理工学部 西谷滋人

Copyright ©2007 by Shigeto R. Nishitani

最初の一步

中学までは数学が好きだったけど、高校に進んで嫌いになったという人もいます。大学に行くともっと嫌いというか、そんな人には訳が分からなくなってきます。でも、理系の場合はそうも言ってもらえないので、数式をいじくり回して適当に答案書いてなんとか単位をもらったりなんかしてます(私のことではありません...)。高校の微積分と大学の微積分は違うのでしょうか？ 逆行列とか固有値とかは難しいもんなんですか？ もっと簡単にできないかなと私も20年前まで思っていました。そんなときに数式処理ソフトに出会いました。衝撃でした。それ以来、数式処理ソフトは私のコンピュータに常駐しています。製品とか版とかは変わりましたが、数式処理ソフトは、数学の天才でない人が、理系で生きていくために必須の道具だと思います。Mapleはそんな数式処理ソフトのひとつです。

どんな感じか見ていきましょう。Mapleを立ち上げて、次のような文を打ち込んでください。

```
> eq1:=- (1/2)*(x-2)^2+2;
```

ただし、Maple10以降では入力モードのデフォルトが「Math」に変わっています。左上にある、「テキスト」というボタンを押してください。毎回するのが面倒な場合は、環境設定→「表示」タブ→入力方法で「テキスト」を選択して、「全体に適用」ボタンを押して、Mapleを再起動させてください。

そして、キーボードのEnterを押してください。これは次の式

$$-\frac{1}{2}(x-2)^2 + 2$$

をeq1としなさいということをMapleに命令したことになります。最後の” ; ”は、ここが文章の区切りですというしるしです。このような文をコマンドと呼びます。では次のコマンドを打ち込んでください。

```
> expand(eq1);
```

$$-\frac{1}{2}x^2 + 2x \tag{2.1}$$

展開したのわかります？ そう。これは、高校数学の最初に出てくる「次の式を展開せよ」という問題をMapleに解かせたのです。expandというのが展開せよという命令

で、括弧” () ”で囲まれた箇所にある式を入力として受け取って、展開した結果を出力してくれます。英語で「展開せよ」という動詞がexpandです。Mapleの生まれ故郷はカナダなので英語で命令が書かれています。近い将来「展開(式1)」とかでやってくれるようになるかも知れませんが、それより英単語を覚えたほうがカッコいいと思いますよ (ぜったい)。

では逆の操作、因数分解をさせてみます。「展開はなんにも頭を使わんでもできるけど、因数分解はそうはいかんぜよ」ということを皆さん実感されているでしょう。でもMapleでは、なんの違ひもありません。因数の事を英語でfactorと言います。こいつをMapleに教えてやると

```
> factor(eq1);
```

$$-\frac{1}{2}x(x-4) \quad (2.2)$$

となります。こんなコンピュータをわざわざ出さんでもと思ったあなた、これならどう?

```
> factor((a+b+c)^3-(a^3+b^3+c^3));
```

$$3(b+c)(a+b)(a+c) \quad (2.3)$$

(でや、まいったか)。

次にeq1をF1という関数と定義しましょう。変数をxとして定義するのは以下のようなコマンドになります。

```
> F1:=unapply(eq1,x);
```

$$F1 := x \rightarrow -\frac{1}{2}(x-2)^2 + 2 \quad (2.4)$$

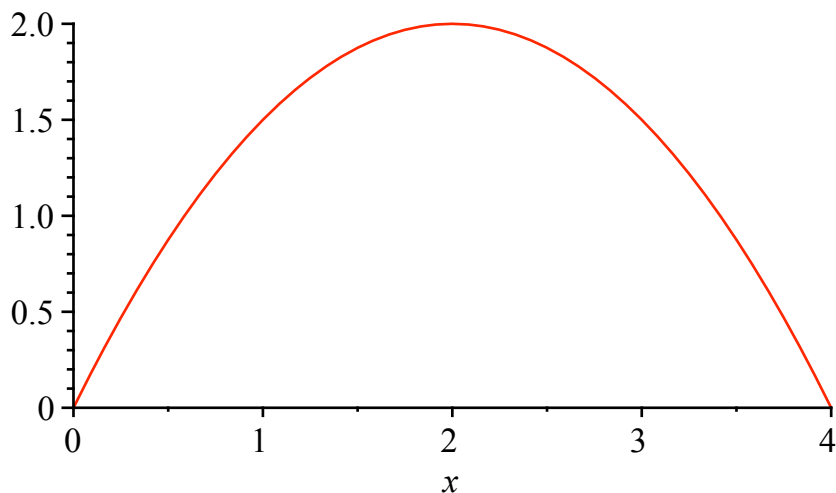
このコマンドunapplyを辞書でひいてもありません。語源はちょっと凝っています。変数を関数に適用するというのをapplyと言いますが、関数を作るのはその逆の操作なのでというらしいです。わからなかったら覚えてください。どうせfactorも初めは覚えるのですから。英語圏の人はこんなのを覚える必要もないので楽ですね。英単語を覚えるのが苦手な人は

<http://ist.ksc.kwansei.ac.jp/~nishitani/Maple/JtoE.html>

を参照してください。ちょっとは役にたつでしょう。

この関数をプロットしてみましょう。関数F1の変数xを0から4までの範囲でplotしなさいというコマンドは次のようになります。

```
> plot(F1(x),x=0..4);
```



これで高校の数学の半分が終わっちゃいました（ウソです）。

▼ 球を放り投げたら

ちょっと動かしてみましよう。

「宇宙のどっかには重力が地球の9.8分の一の星があるはずで、そこでボールを放り投げたらどうなるか？」

という問題です。つぎのようなコマンドを打ち込んでください。とその前に注意。いままでは一行に入るコマンドしか打ち込んでいませんでしたが、長くなると見にくいです。そこで改行したくなります。このときは、`shift+enter`を入力してください。

まずはアニメにするためのおまじない(`with`)です。

> `with(plots);with(plottools);`

あっ。出力は余り意味ありません。びびらせただけです。気にさわるようでしたら、最後の”`;`”を”`:`”にすれば「代入だけ、出力はなし」というコマンドになり、消えます。

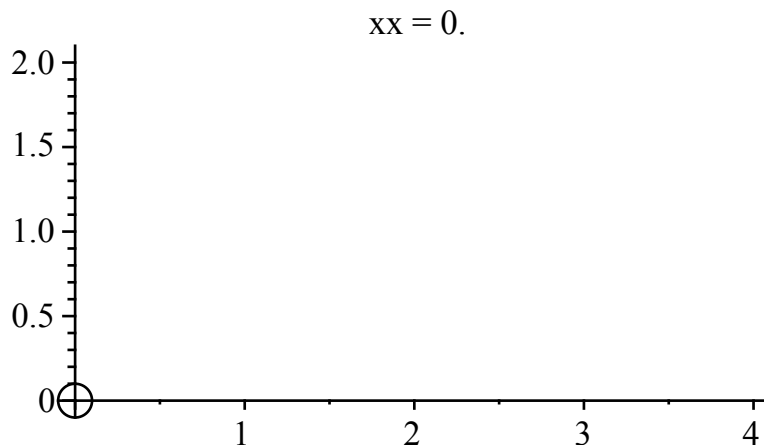
[`animate`, `animate3d`, `animatecurve`, `arrow`, `changecoords`, `complexplot`,
`complexplot3d`, `conformal`, `conformal3d`, `contourplot`, `contourplot3d`,
`coordplot`, `coordplot3d`, `densityplot`, `display`, `fieldplot`, `fieldplot3d`,
`gradplot`, `gradplot3d`, `graphplot3d`, `implicitplot`, `implicitplot3d`,
`inequal`, `interactive`, `interactiveparams`, `intersectplot`, `listcontplot`,
`listcontplot3d`, `listdensityplot`, `listplot`, `listplot3d`, `loglogplot`, `logplot`,
`matrixplot`, `multiple`, `odeplot`, `pareto`, `plotcompare`, `pointplot`,
`pointplot3d`, `polarplot`, `polygonplot`, `polygonplot3d`,
`polyhedra_supported`, `polyhedraplot`, `rootlocus`, `semilogplot`,
`setcolors`, `setoptions`, `setoptions3d`, `spacecurve`, `sparsematrixplot`,
`surfdata`, `textplot`, `textplot3d`, `tubeplot`]

[`arc`, `arrow`, `circle`, `cone`, `cuboid`, `curve`, `cutin`, `cutout`, `cylinder`, `disk`,

(3.1)

dodecahedron, ellipse, ellipticArc, hemisphere, hexahedron, homothety, hyperbola, icosahedron, line, octahedron, parallelepiped, pieslice, point, polygon, project, rectangle, reflect, rotate, scale, semitorus, sphere, stellate, tetrahedron, torus, transform, translate, vrm/]

```
> ball := proc(x)
  plots[display](
    circle([x,F1(x)],0.1));
end proc:
> animate(ball, [xx], xx=0..4, frames=50, scaling=constrained);
```



できあがった画面をクリックして、メニューバーの下に出てきた右向き矢印 (>) を押してください。球が放り投げられているのが分かると思います。すこし「とる」そうだったら、FPSの右にある数字を上げてください。FPSはframes per secondの略です。ゼルダとかでアイテムが落ちていくときの動きとかはまさにこうやって表示させています (たぶん)。

微分

もうすこし数学っぽいことをさせてみましょう。高校数学の大敵のひとつが微積分です。これをMapleにやらせます。これができれば高校の数学は半分終わりです (ホント)。いつものとおり英語を調べれば、「微分する」はdifferentiateと言います。長いですが、ね。「diffる」としたいところです。そうします。Mapleでも。「る」はないけど

```
> df1:=diff(F1(x),x);
                                df1 := -x + 2
(4.1)
```

ちなみに2次微分は次の通りです。

```
> diff(F1(x),x,x);
                                -1
(4.2)
```

いままでと同じようにすぐにプロットさせてみましょう。関数dF1を定義します。

```
> dF1:=unapply(df1,x);
```

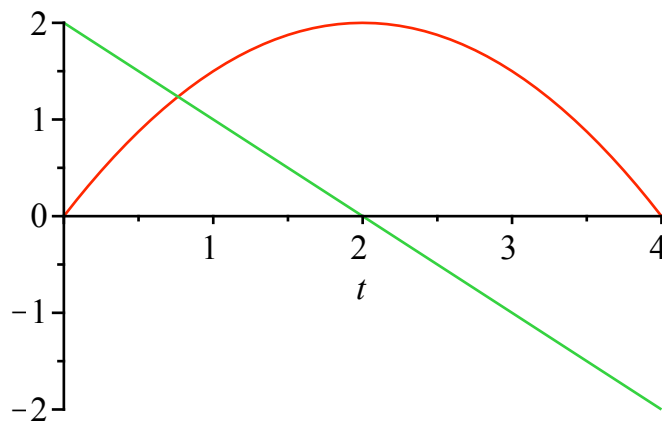
$$dF1 := x \rightarrow -x + 2$$

(4.3)

次のようにして2つの関数を同時にplotさせます。

```
> plot([F1(t),dF1(t)],t=0..4);
```

あ。ここで変数をtに変えています。これは、球を斜めに投げ上げるかわりに真上に投げたときの球の高さxの変化を時間(time, t)で追っかけたと考えていることに相当します。その時の1次微分、つまり傾きが、その速度になります。



【課題1】 2次微分

では2次微分も同時にplotしてみてください。どうなった？ 予測通り？これが物理でいう「重力加速度一定の法則」です。

これをぐちゃぐちゃやっている間に必ずちよんぼります。ここでMapleの隠された真実をひとつお教えします。それは

「Mapleは、コマンドを上から順ではなく、入れた順に解釈する」

です。だから、見た目はあっているのにうまく行かないということが頻発します。そんなときは、入力値をすべて忘れさせるコマンドrestartを一番上に置いて最初から始めます。メニューバーの「挿入」→「実行グループ」を使ったり、enterを入れる順とかを意識して、どの順にコマンドを受け取るべきかというこちらの意思を断固としてきゃつらに示さねばならないのです。

微積分の例

「単純な1次や2次の数式だけしか扱えないのでは」とあなどるなかれ。次みたいな複雑な関数の微分や積分もすぐです。

```
> F2:=unapply(x/(x^2-2*x+4),x);
```

$$F2 := x \rightarrow \frac{x}{x^2 - 2x + 4}$$

(6.1)

```
> diff(F2(x),x);
```

$$\frac{1}{x^2 - 2x + 4} - \frac{x(2x - 2)}{(x^2 - 2x + 4)^2} \quad (6.2)$$

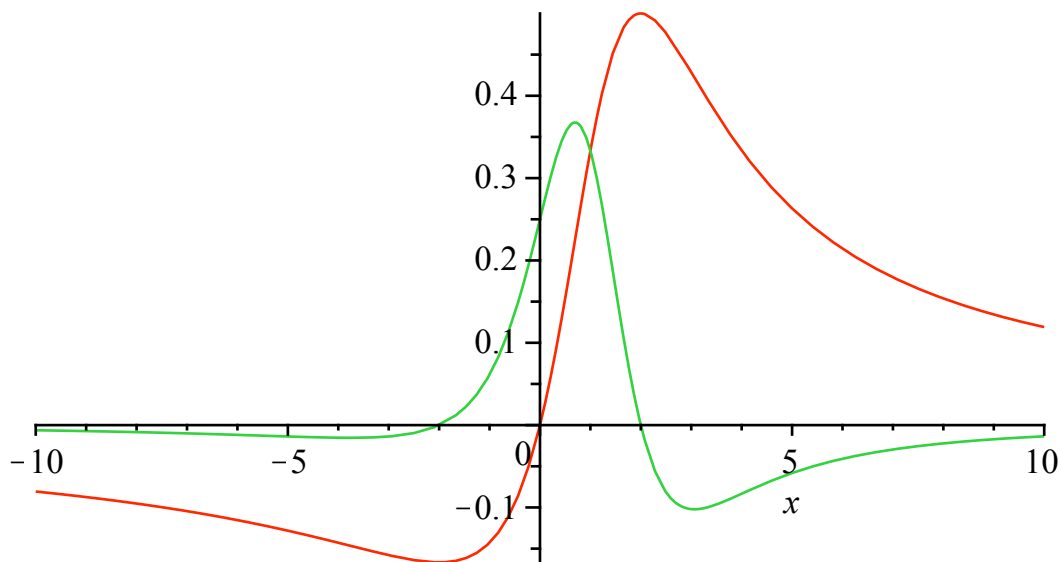
積分する(integrate)は, Mapleでint・るです.

> int(F2(x),x);

$$\frac{1}{2} \ln(x^2 - 2x + 4) + \frac{1}{3} \sqrt{3} \arctan\left(\frac{1}{6} (2x - 2) \sqrt{3}\right) \quad (6.3)$$

おっとこれは習ってないかもしれませんね. でも知らなくても合っているはずですよ(たぶん). plotも簡単です. 赤いほうがもとのF2で, 緑がその微分です.

> plot([F2(x),diff(F2(x),x)],x);



x		-2		2	
f(x)	↘		↗		↘
f'(x)	-	0	+	0	-

こんな増減表なるものをしょっちゅう書かされますが, +や-, やじるしですませているところのビミョーな変化も一目です. ちなみに関数の一次微分が0となるxを求めるには,

「関数の一次微分 (diff(F2(x),x)) が0となる (=0) 式 (eq2) を, xについて解く (solve(eq2,x))」

と考えて, 次の通りです.

> eq2:=diff(F2(x),x)=0;
solve(eq2,x);

$$\text{eq2} := \frac{1}{x^2 - 2x + 4} - \frac{x(2x - 2)}{(x^2 - 2x + 4)^2} = 0$$

-2, 2

(6.4)

先ほどのplotの緑がx軸と交差する点の座標にご注目あれ.

このあと...

どうです. すごいでしょ (えへん). Mapleのコマンドを身に付けたい人は <http://ist.ksc.kwansei.ac.jp/~nishitani/Lectures/Maple/BottomLine0.html> へ行ってみてください. ここには基本から応用までの簡単な解説と演習問題が用意してあります. 初級の欄には高校の数学の問題をたくさん載せています. 例えば, 「接線と逆関数」には高校数学によくでてくる接線の式を求める問題があります. もう少し簡単な課題をいくつか次にあげました. 特に微分・積分と現実世界での動きとの対応に注意してください. ややこしいことを省いて手短かにいうと, 「微分は傾き, 積分は面積」ですので, それを実感してください.

【課題2】ビリヤード球

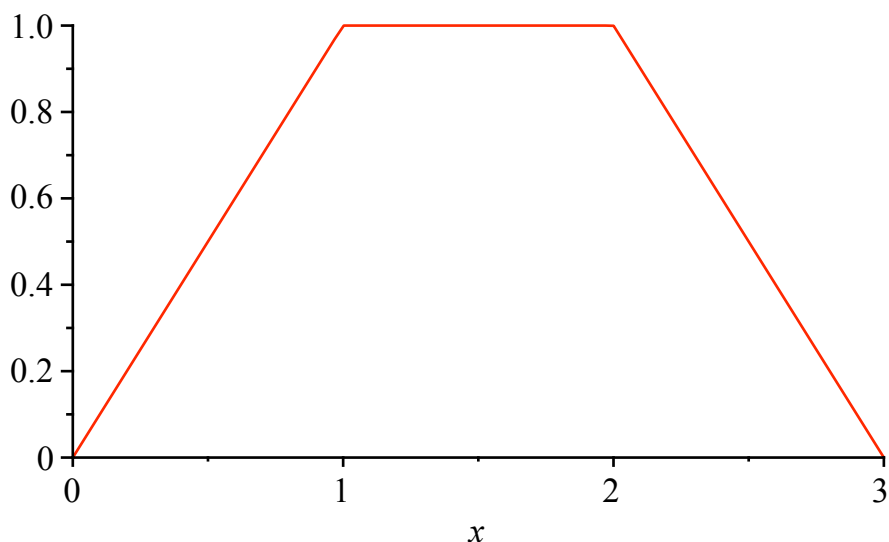
ビリヤード球の跳ね返りは以下のような関数で定義できる. 球を動かしてみよ.

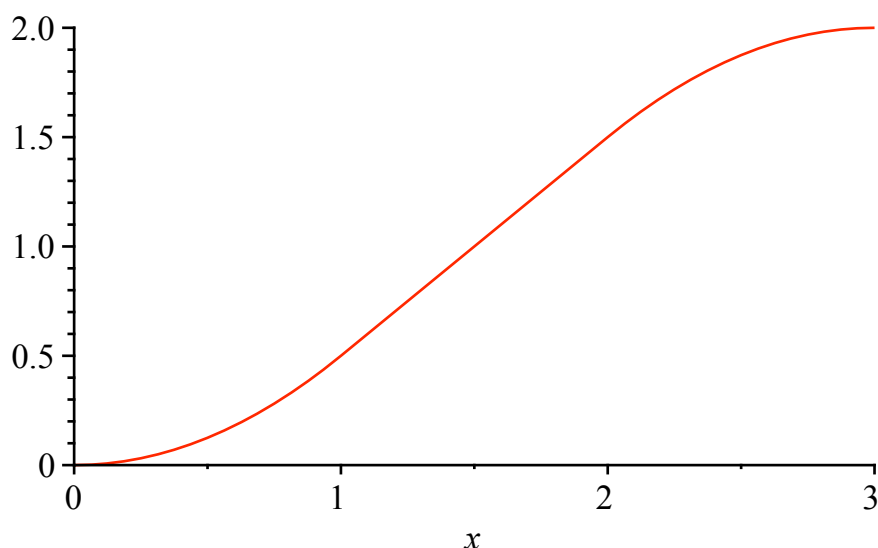
```
> F3:=x->piecewise(x<0,x,x>0,-x);  
F3:=x→piecewise(x<0,x,0<x,-x) (8.1)
```

【課題3】車の距離・速度・加速度

次のような速度変化をさせた車は時間とともにどれだけ進む? 答えは下側の図の通り. では加速度の変化はどうよ.

```
> F3:=x->piecewise(x<1,x, x>1 and x<2,1, x>2,3-x);  
F3:=x→piecewise(x<1,x,1<x and x<2,1,2<x,3-x) (9.1)  
> plot(F3(x),x=0..3);
```





【課題4】人工衛星(惑星)の動き

もうすこし複雑なコマンドです。地球の重力圏に引き込まれないように宇宙船を操縦するときに必要な重力ポテンシャルの視覚化です。ちょっとややこしいので、planet.mwというファイルを

<http://ist.ksc.kwansei.ac.jp/~nishitani/Lectures/Maple/BottomLine0.html>

からダウンロードして、ぐりぐり回して遊んでください。なぜこの絵がそれらしく見えるか考えてください。

```
> restart;
a:=1;
b:=0.8;
with(plots):with(plottools):
e:=sqrt(1-b^2/a^2);
                                a:= 1
                                b:= 0.8000000000
                                e:= 0.6000000000
```

(10.1)

```
> potential:=plot3d(-1/sqrt((x-e*a)^2+y^2),x=-1.2..1.2,y=
-1.2..1.2,view=-3..0,style=wireframe);
                                potential:= PLOT3D(...)
```

(10.2)

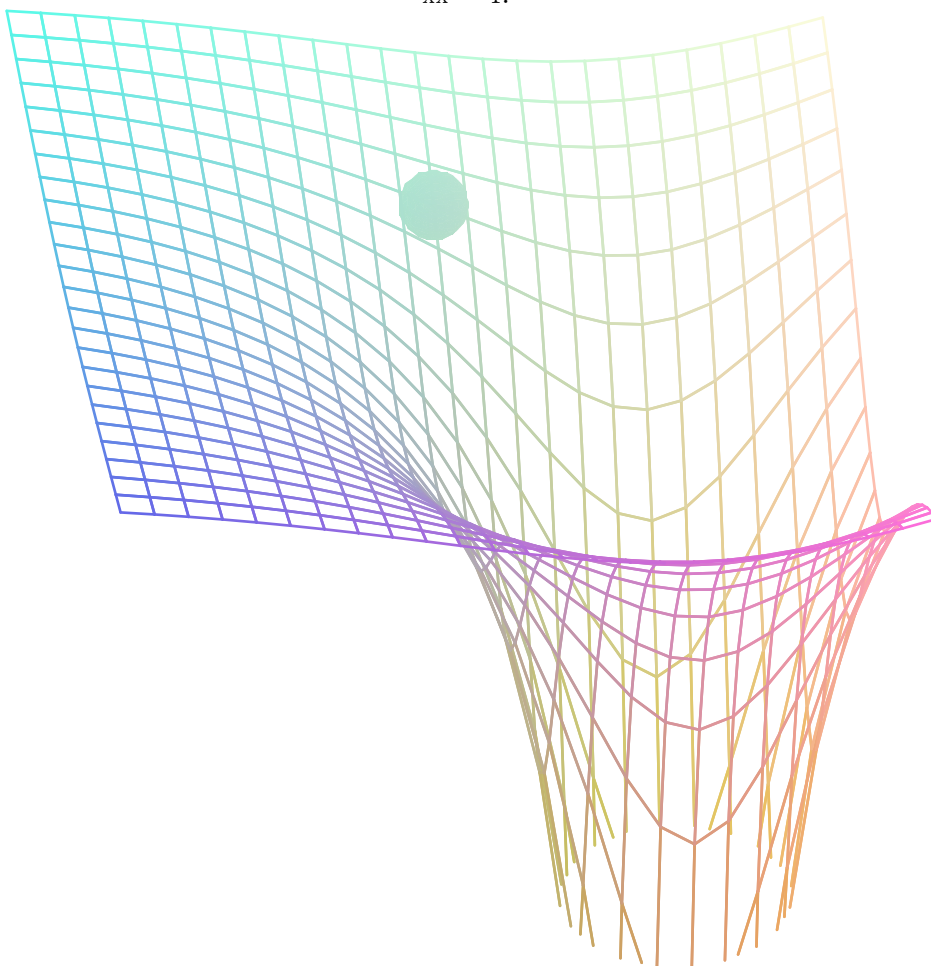
```
> theta40:=[0, .1702850391, .3621441466, .5843340015,
.8499000261, 1.173249127, 1.548794064, 1.927856352,
2.258146178, 2.529619975, 2.755986277, 2.950772893,
3.123171483, 3.279183922, 3.422833759, 3.556924188,
3.683483607, 3.804031515, 3.919742362, 4.031550481,
4.140219423, 4.246389688, 4.350612453, 4.453374408,
4.555116757, 4.656250390, 4.757168589, 4.858258539,
4.959912319, 5.062538193, 5.166573260, 5.272498249,
```



```
5.380855976, 5.492275381, 5.607504105, 5.727454284,  
5.853268956, 5.986421978, 6.128873603, 6.283321858]:
```

```
> planet2:=proc(x)  
  local t;  
  t:=theta40[round(x)];  
  plots[display](  
    sphere([a*sin(t),b*cos(t),-1/sqrt((a*sin(t)-e*a)^2+(b*  
cos(t))^2)],0.1), style=patchnogrid );  
  end:  
animate(planet2,[xx],xx=1..39,frames=39,  
  background=potential, orientation=[-90,0],  
  scaling=constrained,axes=none);
```

xx = 1.



知っている人はケプラーの第一，第二法則との対応を確かめてください。

ダウンロードは右クリックして「ファイルを保存」として，mwという拡張子にします。