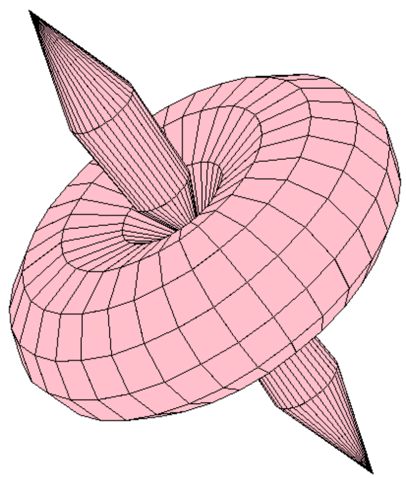
はやわかり鱒の助入門

目　次



北陸職業能力開発大学校 情報技術科
新浜 裕幸　　田中 信博
相川 政和(サポート)

1．鱒の助の概要

鱒の助は、表 1-1 に示す機能を提供する数式処理ソフトです。

表 1-1　鱒の助の提供機能

#	機能
1	算術式の解析
2	変数、組込み定数
3	組込み関数（三角関数、他）
4	関数定義
5	総和(∑)、直積(Π)
6	微分、積分、微分方程式
7	回帰分析（最小２乗法)、相関係数
8	グラフ表示、グラフ操作
9	コマンド履歴
10	マクロ、プロファイル
11	入力支援
12	オプション設定
13	グラフ画像出力
14	編集（切取、複写、貼付）
15	ログ出力
16	その他各種コマンド

この入門書では多くの方が鱒の助を愛用して下さることを願い、鱒の助の代表的な機能を紹介していきます。

2．メインウィンドウ

鱒の助を起動すると、図 2-1 のメインウィンドウが表れます。

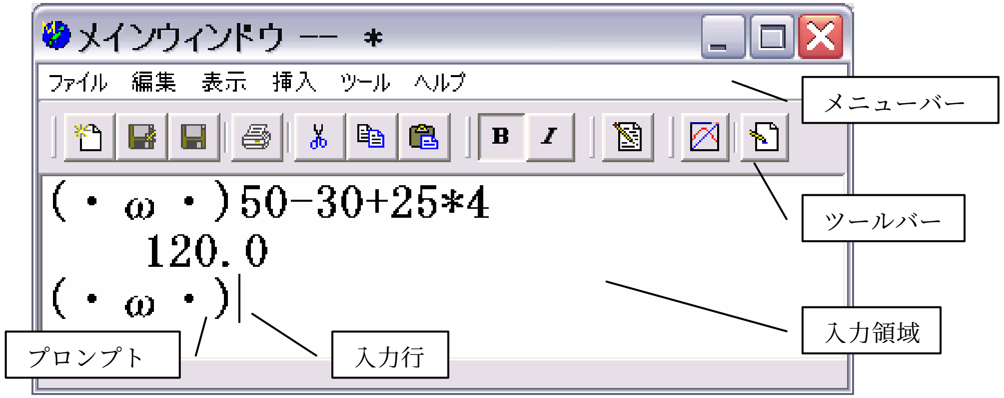


図 2-1　メインウィンドウ

メインウィンドウには、メニューバー、ツールバー、入力領域があります。

ツールバーの左側から（新規作成）、（上書き保存）、（テキストに名前を付けて保存）、（印刷）、（切り取り）、（コピー）、（貼り付け）、（太字）、（斜体）、（入力支援）、（グラフ表示/非表示）、（ロギングモード）となっています。詳細は取扱説明書をご覧下さい。

入力領域に表示されている（・ω・）はプロンプト（入力促進記号）で、この隣に、数式やコマンドを入力します。プロンプトの文字列はメニューバーの ツールーオプションープロンプト で変更できます。

次の章からプロンプトを(^_^)に変更して説明していきます。

３．算術演算

鱒の助が対応している演算記号と結合規則の一覧表を表 3-1 に示します。

表 3-1　演算記号と結合規則

演算方法	演算記号	結合規則
加算（たしざん）	+	→
減算（ひきざん）	-	→
乗算（かけざん）	*	→
除算（わりざん）	/	→
剰余（あまり）	%	→
累乗、べき乗	^	←
階乗	!	←
二重階乗	!!	←
素数階乗	#	←
絶対値	|　|	→
括弧	(　)	→
代入	=	←

・加減乗除

加減乗除を行う場合、次のように数式を入力します。

```
(^_^) 5+10-8
          7.0
(^_^) 20*12/4
          60.0
```

また、負数や小数の演算を行うこともできます。

```
(^_^) 5.2+2.2-8.5
          -1.1
(^_^)1/3
          0.333333333333
```

・剰余

x÷y の余りを求める場合、x%y と入力します。

```
(^_^) 10%3
          1.0
```

・累乗

x の s 乗、即ち x^s を求める場合、x^s と入力します。

```
(^_^) 2^3
          8.0
```

・階乗

n の階乗を求める場合、n!と入力します。

```
(^_^) 5!
          120.0
```

また、二重階乗・素数階乗もできます（詳細は取扱説明書をご覧下さい）

・絶対値

x の絶対値をとる場合、|x|と入力します。

```
(^_^) 2+|-3|
          5.0
```

・優先順位

演算子には優先順位があります。例えば、+,-よりも*,/の方が優先度が高いため、次式を入力した場合、２＋３よりも３＊４の演算が先に行われ、結果は 20 ではなく、14 となります。

```
(^_^)2+3*4
          14.0
```

演算子の優先順位表は取扱説明書にあります。そちらをご覧下さい。

４．変数、組込定数

・変数

変数に値を代入して残すことができます。(鱒の助を終了するまで有効)

　　変数名＝式

この式は変数に式の計算結果を代入（コピー）することを意味します。

```
(^_^) abc=20
        20.0
(^_^) abc-2*10
        0.0
```

　変数名としては半角の英数字と_(アンダースコア)だけを使うことができます。ただし、先頭が英字または_で始まらなければいけません。また、全体で 32 文字以内が有効です。

・組込定数

　円周率や万有引力定数などシミュレーションで良く使われる物理定数を組込定数として提供しています。組込定数値を変更する事はできません。

表 4-1　定数表

定数名	値	意味
PI	3.141592653589793	円周率
E	2.718281828459045	自然対数の底
G	6.67E-11	万有引力定数
g	9.81	重力加速度
C	299792458	真空中の光速
R	6.378E+6	地球の赤道半径

大文字と小文字は区別されます。（g と G は別の定数）

```
(^_^)g
        9.81
```

５．組込関数

　鱒の助は三角関数や統計関数などの便利な関数を提供しています。提供されている関数の一覧については取扱説明書をご覧下さい。

・三角関数

　sin,cos,tan,…,arcsin,arccos,…sinh,cosh など、三角関数及びその関連関数を利用することができます。

例えば、弧度法で sinπ/2 を計算する場合、次のように入力します。

```
(^_^) sin(PI/2)
        1.0
```

・一般数学関数

x の平方根を求める場合 sqrt(x)と入力します。

```
(^_^) sqrt(9)
        3.0
```

e^x を計算する場合、exp(x)と入力します。

```
(^_^)y=exp(3)
        20.08553692319
(^_^)E*E*E
        20.08553692319
```

常用対数は log(x)、自然対数は ln(x)で計算できます。

```
(^_^)ln(y)
        3.0
 (^_^)log(10000)
        4.0
```

切上げは ceil(x)、切捨ては floor(x)、四捨五入は round(x)で計算します。

```
(^_^)ceil(3.2)
          4.0
(^_^)floor(3.9)
          3.0
(^_^)round(4.5)
          5.0
```

2 から n までの素数を知りたい場合、prime(n)と入力します。

```
(^_^) prime(41)
     2      3      5      7     11     13     17
    19     23     29     31     37     41
          13.0
```

素数が表示された後に、n までの素数の個数が最終行に表示されます。

整数 n を素因数分解する場合、factor(n)と入力します。

```
(^_^) factor(546)
     2      3      7     13
          4.0
```

素因数が表示された後に、その素因数の個数が次の行に表示されます。

順列 $_nP_r$ を計算する場合、P(n,r)と、
組み合わせ $_nC_r$ を計算する場合、C(n,r)と入力します。

```
(^_^) P(5,2)
          20.0
(^_^)C(5,2)
          10.0
```

・統計関数

データ列 x1,x2,...,xn の平均を求める場合
mean(x1,x2,x3, ... xn) と入力します。
同様に、中央値は median,分散は variance,標準偏差は stdev を使います。
例として、以下のデータの平均、中央値、分散、標準偏差を順に求めます。

データ：80,75,68,90,92,80,76,50,66,99

```
(^_^) mean(80,75,68,90,92,80,76,50,66,99)
          77.6
(^_^) median(80,75,68,90,92,80,76,50,66,99)
          78.0
(^_^) variance(80,75,68,90,92,80,76,50,66,99)
          182.84
(^_^) stdev(80,75,68,90,92,80,76,50,66,99)
          13.52183419511
```

・その他の関数

AP 関数で等差数列の和を、
GP 関数で等比数列の和を計算できます。

```
(^_^)AP(1,2,10)
          100.0
(^_^)1+3+5+7+9+11+13+15+17+19
          100.0
```

６．関数定義

既存関数や数式を自由に組み合わせて新しい関数を定義することができます。次章以降で説明している数値計算や関数のグラフ化とも関係します。

関数定義の構文は次の通りです。

def 関数名(引数) = 関数式

例えば、f(x)=x^2-1 の関数を定義する場合、

```
(^_^) def   f(x) = x^2-1
```

と入力すると、関数 f が定義され、組込関数と同様に扱われます。
従って、関数値 f(x)を計算する場合は、f(x)と入力します。

```
(^_^) f(3)
        8.0
```

関数名の規則は変数の規則と同様です。
ただし、組込関数やコマンドと同じ名前の関数は定義できません。

f(z,x)=|x+z|のような二変数関数を定義する場合、次のように入力します。

```
(^_^) def   f(z,x)=|x+z|
```

z が 2、x が-3 の場合の関数値(2,-3)は次のように計算します。

```
(^_^) f(2,-3)
        1.0
```

同じ名前の関数を定義すると、以前定義した関数は無効化されます。
上の例では f(x)=x-2-1 は無効化されました。

７．数値計算

先程の関数定義を利用していろいろな計算をしてみます。

・総和

$\sum_{x=a}^{b} f(x)$ を計算する場合、sigma(a,b,f)と入力します。

例えば、 $\sum_{k=2}^{5}(3k^2-1)$ は次のように計算します。

```
(^_^) def sig(k)=3*k^2-1
(^_^) sigma(2,5,sig)
        158.0
```

・方程式の数値解を求める

f(x) = 0 の数値解を求める場合、solve 関数を使います。
ただし、ヒントとして解の範囲を与えなければいけません。
例えば、x^3+x^2-5=0 の解が 1~5 の間にある場合、次のように入力します。

```
(^_^) def f(x)=x^3+x^2-5
(^_^) solve(1,5,f)
        1.43342757225
```

x-cos(x)=0 も解いてみましょう。

```
(^_^)def f(x)=x-cos(x)
(^_^)solve(0,PI/2,f)
        0.739085500758
```

・微分係数

関数 f(x)の微分係数 f'(a)は differ(f(a))と入力して求めます。

例えば、$f(x)=x^2+x-1$ の f'(3)の値は次のように計算します。

```
(^_^) def f(x)=x^2+x-1
(^_^) differ(f(3))
        7.0
```

・定積分

$\int_a^b f(x)dx$ を計算するには、integral(a,b,f)と入力します。

例えば、$f(x)=x^2+x-1$ を区間[0,3]で定積分するには次のように入力します。

```
(^_^) def f(x)= x^2+x-1
(^_^) integral(0,3,f)
        10.5
```

・微分方程式

鱒の助は次の形で与えられる微分方程式を解くことができます。

詳細は省略します。（取り扱い説明書をご覧下さい）

$$\begin{cases} \dfrac{dy}{dx} = f(x,y) \\ (x_0, y_0) : \text{初期値} \end{cases}$$

一階だけでなく、二階微分方程式を解くこともできます。

８．魔方陣、パスカルの三角形、カレンダー

鱒の助は魔方陣やパスカルの三角形等を出力することもできます。

・魔方陣

魔方陣は、縦・横・斜めの和が全て同じになります。例えば、5×5 の魔方陣を表示する場合、次のように入力します。

```
(^_^) magic 5
     3    16     9    22    15
    20     8    21    14     2
     7    25    13     1    19
    24    12     5    18     6
    11     4    17    10    23
```

・パスカルの三角形

例えば、4 のパスカルの三角形を表示する場合、次のように入力します。

```
(^_^) pascal 4
            1
          1   1
        1   2   1
      1   3   3   1
    1   4   6   4   1
```

・カレンダー

例えば、2004 年 6 月のカレンダー表示する場合、次のように入力します。

```
(^_^) cal Jun 2004
     西暦 2004 年  6 月
 Su Mo Tu We Th Fr Sa
        1  2  3  4  5
  6  7  8  9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30
```

９．グラフウィンドウ

　メインウィンドウのメニューバーにあるボタンを押すと､図 9-1 のグラフウィンドウが表れます。(graph コマンドを入力しても表示されます)

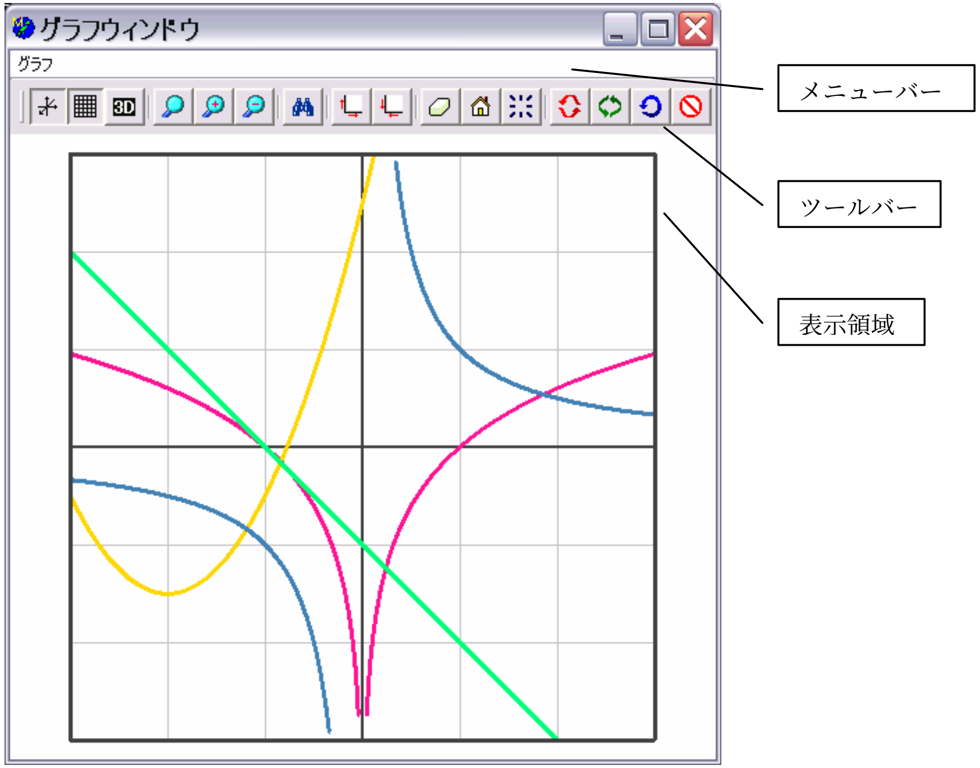


図 9-1　グラフウィンドウ

　グラフウィンドウには､メニューバー､ツールバー､表示領域があります。

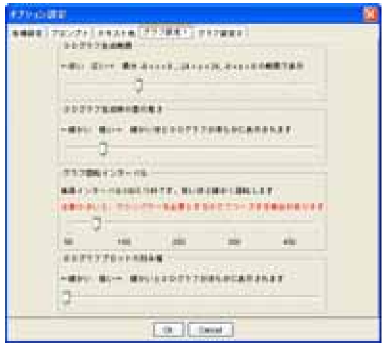
　ツールバーに描かれているアイコンは左側から（軸）､（グリッド）､（2D/3D モード切替）､（拡大・縮小）､（視野の移動）､（表示領域の広さ変更）､（クリア）､（ホーム）､（リセット）､（グラフ回転）､（停止）となっています。

・グラフのオプション

　生成されるグラフの粗さや範囲は

　　　メニューバー　―　ツール　―　オプション　―　グラフ設定 1

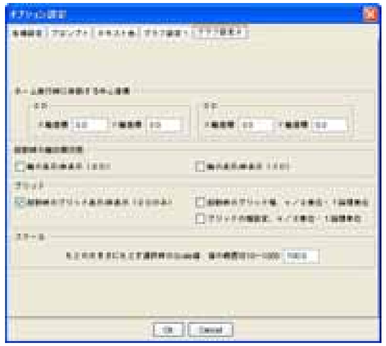
で設定できます。面の粗さを細かく設定すると、動きが重くなります。



　グラフ領域の初期値はメニューバーの

　　　メニューバー　―　ツール　―　オプション　―　グラフ設定 2

で設定できます。



グラフの線色／面色

　グラフの線色／面色は graph コマンドで変更します。

　　　graph　color　orange

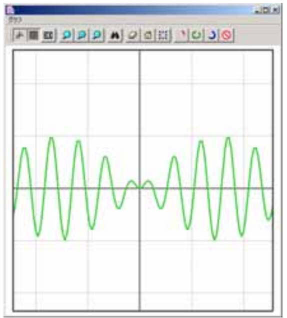
１０．関数のグラフ化｛ y=f(x) ｝

　y = f(x) の形式の関数を簡単にグラフ化できます。
手順は次の通りです。

（１）def コマンドで関数を定義する。
（２）graph コマンドで関数をグラフ化する。

例えば、 $y = \sin(12x) \cdot \sin(x)$ をグラフ化する場合、次のように入力します。

```
(^_^) def　f(x) = sin(12*x)*sin(x)
(^_^) graph　f
```



上の例では、関数名が f、パラメータが x ですが、
別の名前を使っても構いません。
パラメータが横軸、関数値が縦軸となります。

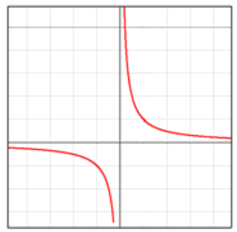
グラフ表示領域で、
・マウス左ボタンでドラッグすると領域が回転します。
・マウス右ボタンでドラッグすると表示領域が移動します。

ツールバーのボタン操作により、以下の機能を実行できます。
・グラフクリア　・回転／停止　・ホームポジションへ戻す
・拡大／縮小　・升目の表示／非表示　・領域の拡大／縮小

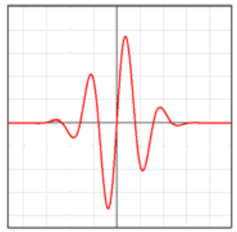
グラフの細かさ（プロット間隔）は、
　メニューバー　―　ツール　―　オプション　―　グラフ設定１
で設定できます。

グラフ出力例をいくつか示します。

```
(^_^) def　y(x)=1/x
(^_^) graph　y
```

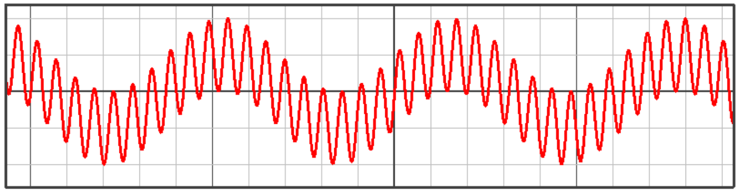


```
(^_^) def g(x)=4*sin(4*x)*exp(-0.5*x^2)
(^_^) graph　g
```



```
(^_^) graph color green
(^_^) def y1(x)=-x+3
(^_^) graph y1
(^_^) def y2(x)=(x+2)*(x-2)
(^_^) graph color red
(^_^) graph y2
```

```
(^_^) def　f(x)=sin(12*x)+sin(x)
(^_^) graph　f
```

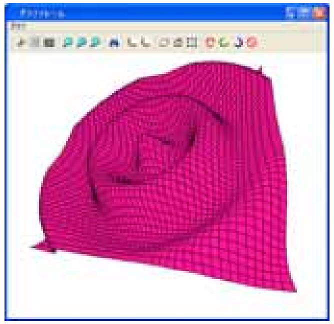


１１．二変数関数のグラフ化 ｛ y=f(z,x) ｝

　二変数関数 y=f(z,x) 形式の関数も簡単にグラフ化できます。
出力結果は曲面となります。
出力手順は y=f(x) 形式の関数と同様です。

例えば、 $y=\cos(2\sqrt{x^2+z^2})$ をグラフ化する場合、次のように入力します。

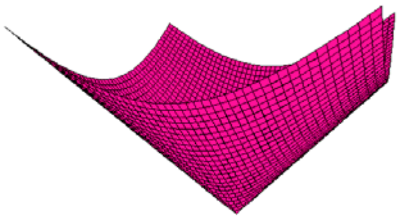
```
(^_^) def   y(z,x)=cos(2*sqrt(x^2+z^2))
(^_^) graph   y
```



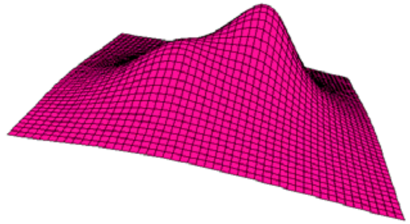
右側に実行結果がありますが、
図形の生成範囲と生成時の面の粗さは
オプション設定で変えることができます。

この機能を使いグラフ出力した例をいくつか示します。

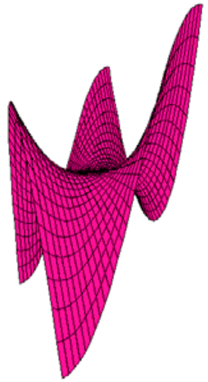
```
(^_^) def   y(z,x)=sqrt(x^2+z^2))
(^_^) graph   y
```



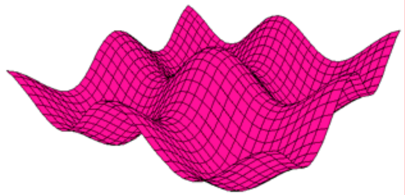
```
(^_^) def   f(x,y) = 2/((x+0.5)^2+y^2+1)+1/((x-1)^2+(y-1)^2+1)
(^_^) graph   f
```



```
(^_^) def   f(x,y) = 0.1*(y^3-3*x^2*y)
(^_^) graph   f
```



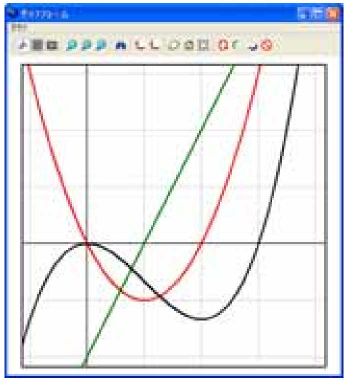
```
(^_^) def   y(z,x) = cos(x+z)*cos(x-z)
(^_^) graph   y
```



１２．微分・積分

y=f(x) の形式の関数を微分・積分してグラフ化するのも簡単です。

```
(^_^) def f(x)=x*(x-2)
(^_^) graph f
(^_^) graph color black
(^_^) graph integral(f)
(^_^) def g(x)=differ(f(x))
(^_^) graph color green
(^_^) graph g
```



・微分

def　g(x) = differ(f(x))

この定義により、f(x)の導関数を g(x)と定義します。

更に、

graph　g

とすれば、f の導関数 g がグラフ化されます。

・積分

def　h(x) = integral(0, x, f)

この定義により、関数 f の定積分を定義してからグラフ化することもできますが、この方法ではグラフ化に時間が掛かります。

そこで、積分グラフを表示する場合は次のように入力します。

graph　integral(f)

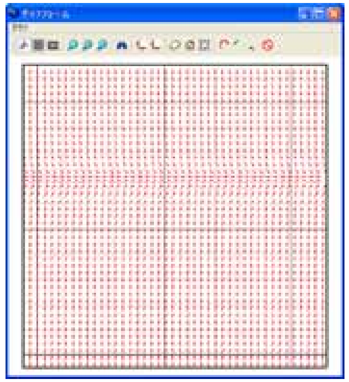
これは、$\int_0^x f(t)dt$ を表示するものです。（0 からの定積分値）

１３．ベクトル場の表示

dy/dx = f(x,y) の式は二次元平面上のベクトル場を表します。

このベクトル場をグラフ化する場合、次のように行います。

```
(^_^) k=0.2
        0.2
(^_^) m=1
        1.0
(^_^) def f(t,v)=g-k*v^2/m
(^_^) graph field(f)
```



dy/dx = f(x,y) のベクトル場を表示する場合、

右辺の関数を定義し、

graph　field(f)

と入力します。ご覧のように、ベクトル場が表示されます。

１４．微分方程式の解グラフ

微分方程式の解グラフを表示することもできます。

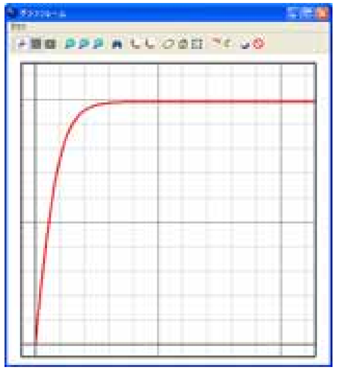
例えば、物体の落下運動について考えて見ましょう。
物体には重力と速度の二乗に比例する空気抵抗が働くものとします。
この場合の運動方程式は次のように表されます。

$$m\frac{dv}{dt} = mg - kv^2$$

m：質量
v：落下速度
g：重力加速度
k：空気抵抗係数

質量を 1[Kg]、重力加速度を 9.8[m/s/s]、空気抵抗係数を 0.2 として
解グラフを表示してみます。

```
(^_^) k=0.1
          0.1
(^_^) m=1
          1.0
(^_^) def f(t,v)=g-k*v^2/m
(^_^) graph diffsolve(0,0,f)
```



・一階微分方程式を解く場合

微分方程式を dy/dx = f(x,y) の形に変形し、次のように入力します。

graph　diffsolve(x0,y0,f)

(x0,y0)は初期値です。
この結果、解グラフが表示されます。

・二階微分方程式を解く場合

二階の微分方程式を解く場合、微分方程式を次の形式に変形します。

$$\frac{d^2y}{dx^2} + P(x,y)\frac{dy}{dx} + Q(x,y) = 0$$

初期値：x_0, y_0, y'_0

そして、関数 P(x,y)と関数 Q(x,y)を定義した後、次のように入力します。

diffsolve2(x0,y0,y'0,P,Q)

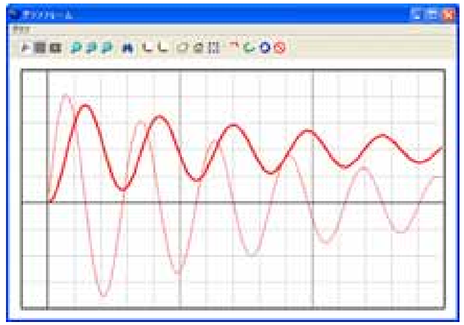
この結果、y と y'の変化がグラフ化されます。

ばねの運動を解析する例を以下に示します。
ばね定数 K のばねの先端に質量 m のおもりをつけます。
ばねの自然長の位置におもりを置き、静かに手を放した場合

$$\frac{d^2x}{dt^2} + \frac{h}{m}\frac{dx}{dt} + \frac{kx}{m} - g = 0$$

x：ばねの変位
t：経過時間
h：空気抵抗係数
g：重力加速度

```
(^_^) h=0.2
          0.2
(^_^) m=1
          1.0
(^_^) k=5
          5.0
(^_^) def p(t,x)=h/m
(^_^) def q(t,x)=k*x/m-g
(^_^) graph diffsolve2(0,0,0,p,q)
```



１５．二次元パラメトリック曲線

y = f(x) 形式の曲線だけでなく、媒介変数（パラメータ）を用いた次の形式の曲線を描くこともできます。

$$\begin{cases} x = g(t) \\ y = h(t) \end{cases}$$

上の式で表される曲線を描くには、関数 x(t)と関数 y(t)を定義した後、次のように入力します。

graph　x : y [a, b]

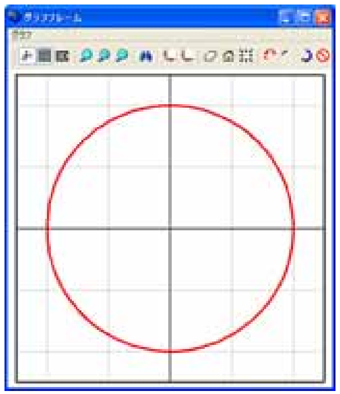
x: x(t)の関数名　y: y(t)　の関数名　a: t の最小値　b: t の最大値

例えば、半径 r の円は次式で表されます。

$$\begin{cases} x = r \cdot \cos(t) \\ y = r \cdot \sin(t) \\ 0 \leqq t \leqq 2\pi \end{cases}$$

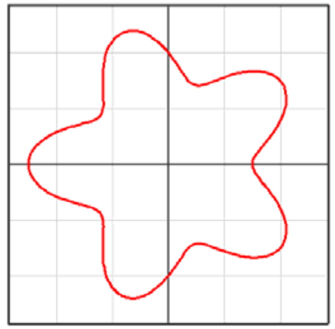
従って、次の要領で円を描きます。

```
(^_^) r=2
          2.0
(^_^) def x(t)=r*cos(t)
(^_^) def y(t)=r*sin(t)
(^_^) graph x:y[0,2*PI]
```

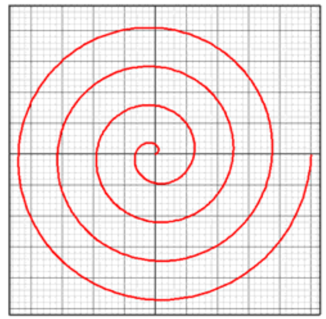


半径を一定ではなく、変化させれば次のような図形も描けます。

```
(^_^) def r(t)=2-0.5*cos(5*t)
(^_^) def x(t)=r(t)*cos(t)
(^_^) def y(t)=r(t)*sin(t)
(^_^) graph x:y[0,2*PI]
```

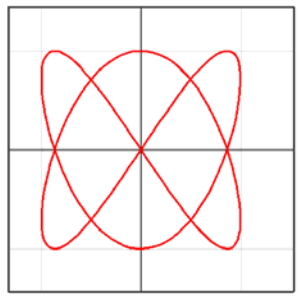


```
(^_^) def r(t)=t
(^_^) graph x:y[0,8*PI]
```



次の図はリサージュ図形と呼ばれるものです。

```
(^_^) def x(t)=sin(2*t)
(^_^) def y(t)=sin(3*t)
(^_^) graph x:y[0,2*PI]
```



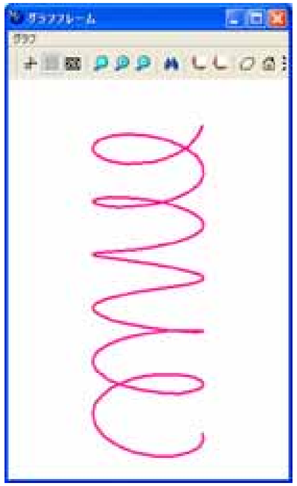
１６．三次元パラメトリック曲線

次式で表される三次元の媒介変数を用いた曲線を描くこともできます。

$$\begin{cases} x = f(t) \\ y = g(t) \\ z = h(t) \end{cases}$$

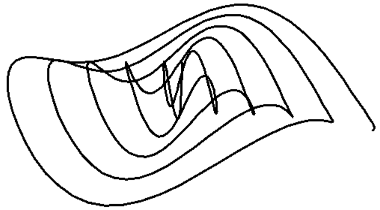
描き方は二次元の場合と同様です。
ばねを描く例を示します。

```
(^_^) def x(t)=2*cos(t)
(^_^) def y(t)=0.3*t
(^_^) def z(t)=2*sin(t)
(^_^) graph x:y:z[-6*PI,6*PI]
```



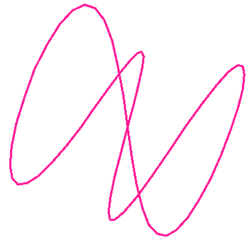
平面上の螺旋を面と垂直な方向へ
波打たせています。

```
(^_^)def x(t)=0.1*t*cos(t)
(^_^)def y(t)=0.5*sin(3*t)
(^_^)def z(t)=0.1*t*sin(t)
(^_^)graph x:y:z[0,10*PI]
```



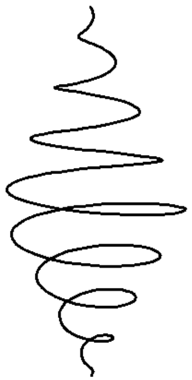
次の例は三次元のリサージュ図形です。

```
(^_^) def x(t)=sin(t)
(^_^) def y(t)=sin(2*t)
(^_^) def z(t)=sin(3*t)
(^_^) graph x:y:z[0,2*PI]
```



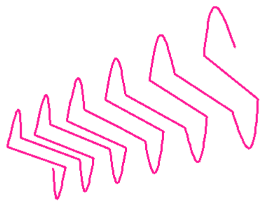
半径を変化させれば、
右の形のばねを描くことができます。

```
(^_^)def r(t)=2-|0.1*t|
(^_^)def r(t)=2.5-|0.1*t|
(^_^)def x(t)=r(t)*cos(t)
(^_^)def y(t)=0.2*t
(^_^)def z(t)=r(t)*sin(t)
(^_^)graph x:y:z[-8*PI,8*PI]
```



工夫すれば直線と曲線を組み合わせることもできます。

```
(^_^) def x(t)=sign(sin(t))
(^_^) def y(t)=0.2*t
(^_^) def z(t)=sin(t)
(^_^) graph x:y:z[-6*PI,6*PI]
```

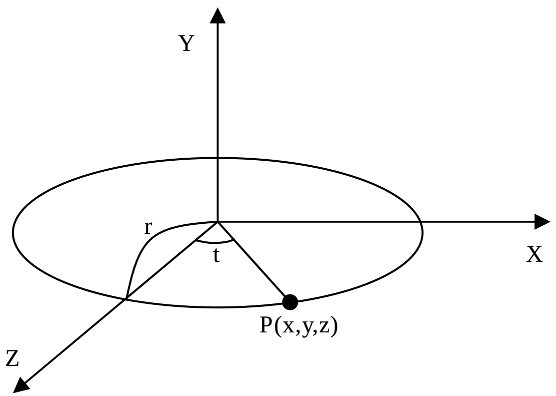


１７．三次元パラメトリック曲面

二つの媒介変数 u,v により表現される次の式は曲面を表します。

$$\begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases}$$

円盤を例にとり、鱒の助で曲面を表示する方法について説明します。

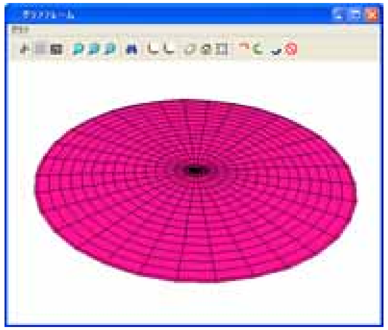


円盤上の点$P(x, y, z)$は、半径rと中心角tを使い、
次のように表されます。

$$\begin{cases} x(r,t) = r\sin(t) \\ y(r,t) = 0 \\ z(r,t) = r\cos(t) \end{cases}$$

従って、鱒の助では次のように入力します。

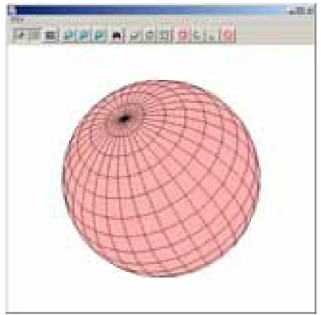
```
(^_^) def x(r,t)=r*sin(t)
(^_^) def y(r,t)=-1
(^_^) def z(r,t)=r*cos(t)
(^_^) graph x:y:z[0,2][0,2*PI]
```



次に球面を表示してみましょう。

球面上の点の座標は極座標を使えば、次式で表されます。

$$x = r\cos(\phi)\cdot\sin(\theta)$$
$$y = r\sin(\phi)$$
$$z = r\cos(\phi)\cdot\cos(\theta)$$

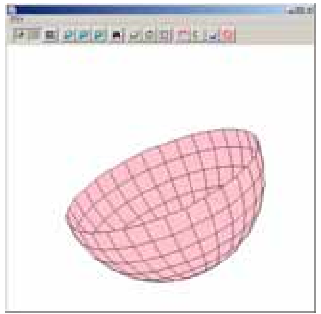


従って、鱒の助では次のように入力します。

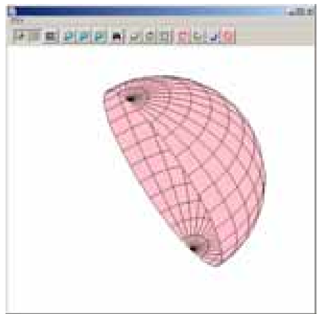
```
(^_^) def x(s,t)=cos(t)*sin(s)
(^_^) def y(s,t)=sin(t)
(^_^) def z(s,t)=cos(t)*cos(s)
(^_^) graph x:y:z[0, 2*PI][-PI/2, PI/2]
```

s と t の範囲を変えれば、半球も描けます。

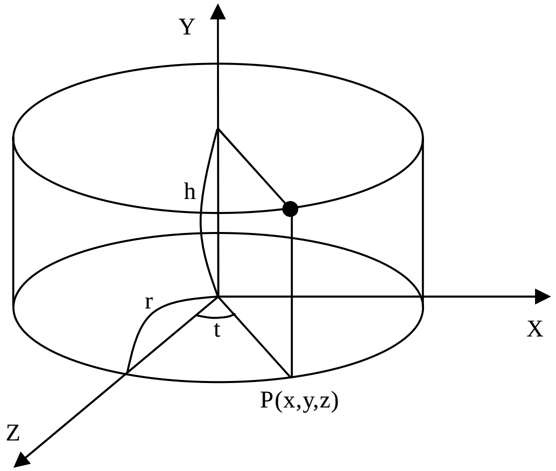
```
(^_^) graph x:y:z[0, 2*PI][-PI/2, 0]
```



```
(^_^) graph x:y:z[0, PI][-PI/2, PI/2]
```



もう一例紹介しましょう。今度は円筒です。



円筒上の点$P(x,y,z)$は、半径r、中心角t、高さtを用いて次のように表される。

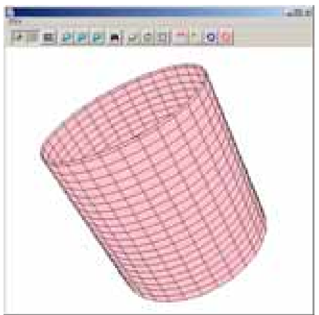
$x = r\sin(t)$

$y = h$

$z = r\cos(t)$

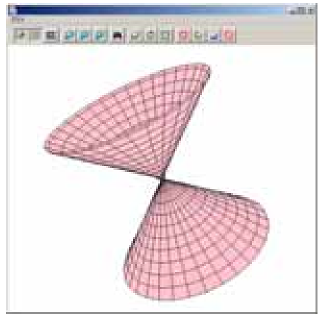
従って、鱒の助では次のように入力します。

```
(^_^) r=2
        2.0
(^_^) def x(t,h)=r*sin(t)
(^_^) def y(t,h)=h
(^_^) def z(t,h)=r*cos(t)
(^_^) graph x:y:z[0,2*PI][-2,2]
```



半径を変化させれば、いろんな形ができます。

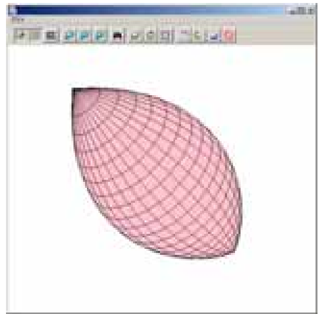
```
(^_^) def r(t,h)=|h|
(^_^) def x(t,h)=r(t,h)*sin(t)
(^_^) def y(t,h)=h
(^_^) def z(t,h)=r(t,h)*cos(t)
(^_^) graph x:y:z[0,2*PI][-2,2]
```



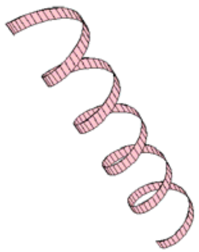
中心角度に依り、半径を変化させると

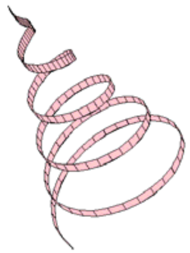
```
(^_^) def r(t,h)=0.5*t/PI
(^_^) graph x:y:z[0,4*PI][-2,2]
```

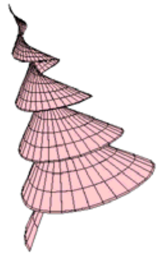
```
(^_^) def r(t,h)=-(h-2)*(h+2)/4
(^_^) graph x:y:z[0,2*PI][-2,2]
```

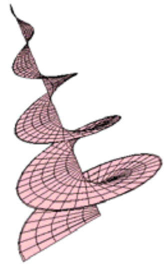


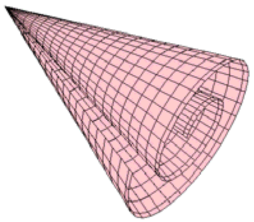
最後に、鱒の助を使って描いたグラフを載せておきます。
皆さんも面白い形状の生成に挑戦してみましょう。

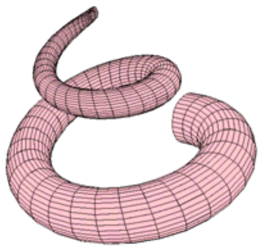


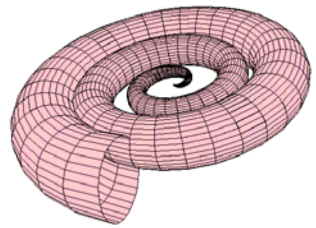


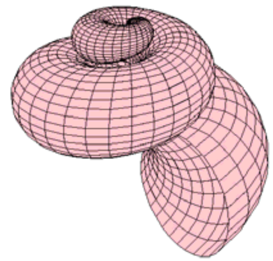


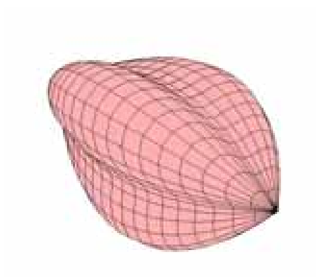


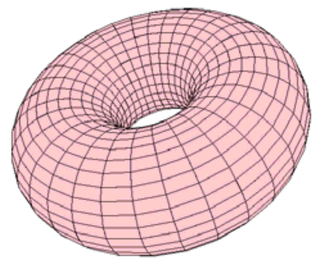












はやわかり鱒の助入門

平成 16 年 7 月 4 日(日) 第一版発行

著者：新浜裕幸、田中信博、相川政和