まえがき

このテキストは，「**オープンソースソフトウェアによる情報リテラシー**」ー第 2 版ー の「第 11 章 数式処理とグラフ作成」に加筆したものです。

Maxima は数式処理，数値計算，関数やデータのグラフ作成などを行うことができるコンピュータ代数システムです．最初の章では，Maxima のグラフィカルなフロントエンドである wxMaxima を使って，主に数式処理について説明します．また次の章では，wxMaxima を使ったグラフ作成について説明しています。

別途，同じ内容を Python の Sympy などを使って実行した例を用意しています。ですから，一から Maxima や SymPy を始めようとしている人だけでなく，Maxima と SymPy のどちらかは知っているが，同じことをもう一方ではどのように計算するかということに興味を持っている人にも読んでもらえるのではないかと思います。

なお，wxMaxima の実行結果は本稿執筆時点（2012 年頃）のものです。その後の Maxima バージョンアップにより，よりスマートな結果を出力するようになっているかも知れません。

葛西 真寿（KASAI Masumi）

弘前大学 大学院理工学研究科
情報連携統括本部 情報基盤センター

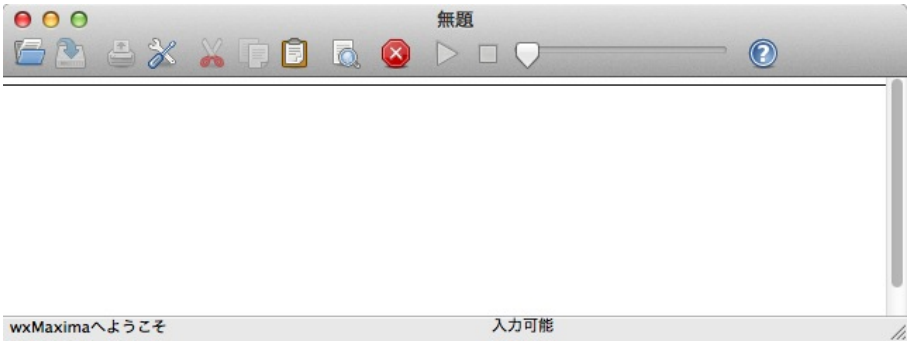
目　次

第 1 章
wxMaxima による数式処理

1.1 基本操作

1.1.1 wxMaxima の起動

wxMaxima を起動すると，以下のようなウィンドウが開きます．



なお，wxMaxima は Windows や Mac OS X，そして Linux でも使えるクロスプラットフォームなグラフィカル・フロントエンドです．使用するプラットフォームによって，若干の外見の違いがありますが，基本的な使い方は共通です．

1.1.2 計算の実行と改行

計算の実行は，数値や式などを入力して**最後にセミコロン ; を入力後，Shift キーを押しながら Enter キーを押します**．（セミコロンを忘れた場合は，Shift + Enter キーの後，wxMaxima が自動的にセミコロンを付加します．）

まずは簡単なかけ算 `123*456` を計算してみます．（かけ算の記号は*です．）行頭の `(%i..)` が入力を，`(%o..)` が出力結果を表します．`".."` には通し番号が自動的に付けられます．

2 の 100 乗 (2^{100}) のような大きい数でも，厳密な結果を出力します．（^は累乗を表します．）また，`float` 関数を使って浮動小数点表示で近似値を出力させることもできます．

Enter キーのみでは改行されるだけで計算が実行されません．Shift+Enter キーが入力されるまで，wxMaxima は式の入力が継続していると判断します．

例として，以下のような積分を計算してみます．Shift キーを忘れて Enter キーのみを押すと単に改行されます．**慌てずにセミコロン ；を入力後， Shift キーを押しながら Enter キーを押すと実行します．**（セミコロンを忘れて Shift+Enter キーを押しても，wxMaxima が付けたしてくれます．）答えは $\int 3x^2 dx = x^3$ です．

1.1.3　計算結果の保存と読み込み

wxMaxima で計算した結果を保存するには，「ファイル」メニューから「保存」コマンドを使います．保存したファイルを閉じた後，また「ファイル」から「開く」を選んで，読み込んでみると，以下のように入力部分のみが保存されていることがわかります．このとき，各行を

Shift+Enter キーで実行していくこともできますが，「Cell」メニューから「全てのセルを評価」を選ぶと，一挙に実行できて便利です．

1.2　表記などの約束事

Maxima の実行に際して，いくつか表記についての約束事があります．代表的なものを以下に示します．

コメント (実行に影響しない注釈) を入れたいときは，`/* ... */` でコメント部分を囲みます．

```
(%i1)  1 + 2 + 3;
(%o1)  6
(%i2)  1 + 2 /* + 3 */;
(%o2)  3
```

加法 `+`，減法 `-`，乗法 `*`，除法 `/` における優先順位は数学と同じです．優先して計算したい箇所は （　） で囲みます．

```
(%i3)  1 + 2 * 3;
(%o3)  7
(%i4)  (1 + 2) * 3;
(%o4)  9
```

ただし，大カッコ `[ ]` や中カッコ `{ }` は優先順位の指定には用いません．

1.2.1　代入

変数に式や値を代入するときは `:`（コロン）を使います．

変数 `a` に，$100+3/21$ を代入して，`a` を使って計算を続けます．

```
(%i5)  a: 100 + 3/21;
(%o5)  701/7
(%i6)  (a-100) * 7;
(%o6)  1
```

変数 `eq` に 式 x^2+3x+2 を代入します．

`eq` を `factor` 関数で因数分解します．

```
(%i7)  eq: x^2 + 3*x + 2;
(%o7)  x^2+3*x+2
(%i8)  factor(eq);
(%o8)  (x+1)*(x+2)
```

1.2.2　関数の定義

関数の定義には := を使います.

$f(x) := x^2$

$f(4) = 4^2 = 16$

$f(3\sqrt{A}) = (3\sqrt{A})^2 = 9A$ です.

$\frac{d}{dx}(x\cos x)$ を計算します.

この例のように f(x):= % のような仕方で (直前の) 計算結果を参照して関数を定義することはできません.

```
(%i9)  f(x):= x^2;
(%o9)  f (x) := x^2
(%i10) f(4);
(%o10) 16
(%i11) f(sqrt(A)*3);
(%o11) 9 A
```

```
(%i12) diff(x*cos(x), x);
(%o12) cos (x) − x sin (x)
(%i13) g(x):= %;
(%o13) g (x) := %
(%i14) g(x);
(%o14) g (x) := %
```

このような場合には define を使います. %o12 は 12 番目の出力結果 (今の場合は $\cos(x) - x\sin(x)$ です.)

```
(%i15) define(g(x), %o12);
(%o15) g (x) := cos (x) − x sin (x)
(%i16) integrate(g(x), x);
(%o16) x cos (x)
```

1.2.3　実行結果の非表示

文末に $ をつけると実行結果の出力を省略します. 変数 eq を出力させると…
確かに式 $x^2 + 3x + 2$ ですね.
2 次方程式 $x^2+3x+2 = 0$ を solve を使って解いていますが, 結果は出力されません. $ ではなく ; を文末につけると結果が出力されます.

```
(%i17) eq: x^2 + 3*x + 2$
(%i18) eq;
(%o18) x^2 + 3 x + 2
```

```
(%i19) solve(eq = 0, x)$
(%i20) solve(eq = 0, x);
(%o20) [x = −2, x = −1]
```

1.2.4 前の出力結果の参照

直前の出力結果は % と表します．これを参照してさらに計算することができます．また，以前の結果を参照する場合は %o22 のように通し番号を指定することもできます．

expand 関数を使って展開します．

```
(%i21) expand( (x-2)^4 );
(%o21) x^4-8*x^3+24*x^2-32*x+16
```

直前の結果を factor 関数で因数分解します．

```
(%i22) factor(%);
(%o22) (x-2)^4
```

行番号 %o22 の結果を展開します．

```
(%i23) expand(%o22);
(%o23) x^4-8*x^3+24*x^2-32*x+16
```

1.2.5 一時的代入

ev 関数を使うことで，一時的に式などに値を代入する事ができます．

変数 c に 2*x を代入します．

```
(%i24) c: 2*x;
(%o24) 2*x
```

x=3 を一時的に代入して c の値を評価します．

```
(%i25) ev(c, x=3);
(%o25) 6
```

c の定義そのものは変わりません．

```
(%i26) c;
(%o26) 2*x
```

1.2.6 リスト成分や右辺・左辺の取り出し

方程式の解などは，リスト形式（[ ] で囲まれた形）で出力します．リスト成分は [ ] で取り出すことができます．

関数 f(x) を定義します．

```
(%i27) f(x):= x^2 + 3*x + 2;
(%o27) f(x):=x^2+3*x+2
```

solve で方程式 f(x) = 0 を解きます．

```
(%i28) sol: solve( f(x) = 0, x );
(%o28) [x=-2,x=-1]
```

リストとして表されている 2 個の解のうち，1 番目を `sol[1]` として．.. 2 番目を `sol[2]` としてそれぞれ取り出します．

```
(%i29) sol[1];
(%o29) x = -2
(%i30) sol[2];
(%o30) x = -1
```

`rhs` は右辺 (right hand side) を取り出す関数です．また，左辺を取り出す関数は `lhs`(left hand side) です．

`x = -2` の右辺を取り出して `ans1` に代入します．

二つめの解も同様に `ans2` に代入します．

```
(%i31) ans1: rhs(sol[1]);
(%o31) -2
(%i32) ans2: rhs(sol[2]);
(%o32) -1
```

関数 `f(x)` に解を入れると確かに 0 になります．

`(%i35)` は見慣れない書式ですが，`ev(f(x), sol[1])` の簡略形です．

```
(%i33) f(ans1);
(%o33) 0
(%i34) f(ans2);
(%o34) 0
(%i35) f(x), sol[1];
(%o35) 0
```

1.3　数の計算・基本的な関数

1.3.1　基本的な演算

1 から 10 までの和を求めてみます．

同様の計算は公式 $n*(n+1)/2$ を使ってもできます．

`(%i4)` は `ev(n*(n+1)/2, n=10)` の簡略形でしたね．

```
(%i1)  1+2+3+4+5+6+7+8+9+10;
(%o1)  55
(%i2)  n: 10$ n*(n + 1)/2;
(%o3)  55
(%i4)  n*(n + 1)/2, n = 10;
(%o4)  55
```

基本的な四則演算 (`+ - * /`) 以外によく使われる演算記号を示します．

2^4 の計算．累乗は ^ で表します．

^ のかわりに ** でもよいようです．

```
(%i5)  2^4;
(%o5)  16
(%i6)  2**4;
(%o6)  16
```

4 の階乗の計算をします．
4! = 4*3*2*1 = 24 です．

```
(%i7)  4!;
(%o7)  24
```

!!は，一つおきの階乗の計算をします．5!! = 5*3*1 です．

```
(%i8)  5!!;
(%o8)  15
```

（練習）3, 2, 1 でつくる最大の数

数字の 3 と 2 と 1 を 1 個ずつ使った演算でつくることができる最大の整数は？

ヒント: $(1+2)\times 3 = 9$ じゃ全然だめ．そのまま並べると 321 ですが，累乗を使うともっと大きい数をつくることができます．21^3 とか，31^2 とか，2^{31} とか…

1.3.2 基本的な定数

Maxima の基本的定数を示します．

円周率 π	無限大 ∞	自然対数の底 e	虚数単位 i
%pi	inf	%e	%i

無理数 e，π と虚数単位 i と整数 -1 の間のすてきな関係．

オイラーの等式 $e^{i\pi} = -1$.

```
(%i9)  %e^(%i * %pi);
(%o9)  -1
```

浮動小数点で近似値を表示するには float や bfloat を使います．

```
(%i10) %pi;
(%o10) π
(%i11) float(%pi);
(%o11) 3.141592653589793
(%i12) bfloat(%pi);
(%o12) 3.1415926535897932b0
```

float は 16 桁の値を返します．それ以上の桁数を見たいときには，bfloat を使います．bfloat の表示桁数は fpprec で指定します．

有効数字 32 桁で円周率 π を表示する例です.

```
(%i13) fpprec: 32$  bfloat(%pi);
(%o14) 3.1415926535897932384626433832795b0
```

1.3.3　基本的な関数

基本的な関数は以下のように表記されます.

平方根	指数関数	自然対数	絶対値	三角関数	逆三角関数	双曲線関数
sqrt	exp	log	abs	sin, cos, tan	asin, acos, atan	sinh, cosh, tanh

うっかり $\sqrt{x^2} = x$ としないように.
$\sqrt{x^2} = |x|$ です.

```
(%i15) sqrt(x^2);
(%o15) |x|
```

ピタゴラスの定理をみたす整数の例.
$\sqrt{3^2 + 4^2} = 5$ ですね.

```
(%i16) sqrt(3^2 + 4^2);
(%o16) 5
```

$e^1 = e$ のこと.

```
(%i17) exp(1) - %e;
(%o17) 0
```

$e^{(\log a)}$

$\log(e^2)$

```
(%i18) exp( log(a) );
(%o18) a
(%i19) log(%e^2);
(%o19) 2
```

(練習) ピタゴラス数

$a^2 + b^2 = c^2$ を満たす正の整数の組をピタゴラス数といいます. ピタゴラス数は二つの任意の正の整数 m, n から以下のように何組でもつくることができます.

$$a = |m^2 - n^2|, \quad b = 2mn, \quad c = \sqrt{a^2 + b^2}$$

このとき, c は必ず整数になることを示しなさい.

試しにやってみます. a に m と n の表式を代入します.
おっと, 以前 (%i2 で) n に数値を代入していたのが悪さをしていますね.
kill(n) で n の定義を消して...

もう一度 a の定義.

```
(%i20) a: abs(m^2 - n^2);
(%o20) |m^2 - 100|
(%i21) kill(n);
(%o21) done
(%i22) a: abs(m^2 - n^2);
(%o22) |n^2 - m^2|
```

b にも m と n の表式を代入し…

c の定義もしておきます．

```
(%i23) b: 2*m*n;
(%o23) 2 m n
(%i24) c: sqrt(a^2 + b^2)$
```

直接証明することも簡単ですが，疑似乱数を使って，ちょっとした数値実験をしてみます．

random(100) は 0 から 99 までの整数疑似乱数をつくる関数です．

$c = \sqrt{a^2 + b^2}$ は確かに整数になっています．証明も簡単ですね．

```
(%i25) ev([m, n, a, b, c],
          m=random(100), n=random(100));
(%o25) [12, 2, 140, 48, 148]
(%i26) ev([m, n, a, b, c],
          m=random(100), n=random(100));
(%o26) [34, 85, 6069, 5780, 8381]
```

参考: random(100) は 0 から 99 までの整数値の疑似乱数を与えます．random(1.0) ならどうなるか，試してみましょう．

（練習）星の明るさと等級

夜空に見える星の明るさ L は等級 m で表され，以下のような関係があります．

$$m = -2.5 \log_{10} L + \text{const.} \quad （\text{const.} は定数のこと）$$

1 等星と 6 等星では，どちらが何倍明るいですか？

load(log10); とすると，常用対数 log10 が使えます．
$\log_{10} 1000 = \log_{10} 10^3 = 3$ ですね．

```
(%i34) load(log10)$
(%i35) log10(1000);
(%o35) 3
```

1.3.4　三角関数の値・引数の単位

三角関数の値を数値で求めたいときには，以下のようにします．

$\sin \frac{\pi}{3}$ の値は…
$\frac{\sqrt{3}}{2}$ です．

```
(%i27) sin(%pi/3);
(%o27) sqrt(3)/2
```

float 関数を使って，直前の結果を浮動小数点表示します．

```
(%i28) float(%);
(%o28) .8660254037844386
```

変数 pi に浮動小数点表示の π の値を代入します．
浮動小数点表示で数値を入れると，sin も浮動小数点表示で答えを返します．

```
(%i29) pi: float(%pi);
(%o29) 3.141592653589793
(%i30) sin(pi/3);
(%o30) .8660254037844386
```

三角関数の引数はラジアンです．度を使って求めたいときは以下のようにします．

60° をラジアンに直した引数を使って計算します．

```
(%i31) sin( 60/180*%pi );
```

$$(\%o31)\ \frac{\sqrt{3}}{2}$$

度をラジアンに変換する関数 degrad を定義します．

```
(%i32) degrad(x):= x/180*%pi;
```

$$(\%o32)\ \mathrm{degrad}\,(x) := \frac{x}{180}\,\pi$$

degrad(60) を引数にして sin の値を求めます．

```
(%i33) sin( degrad(60) );
```

$$(\%o33)\ \frac{\sqrt{3}}{2}$$

（練習）屋根勾配

家の屋根の勾配は角度ではなく，伝統的に 3 寸勾配，4 寸勾配のように水平方向に 1 尺（10 寸）に対して垂直方向に何寸立ち上がるかで示します．図のログハウスの屋根は 8 寸勾配です．傾斜角度にすると何度ですか？
以下に各寸勾配の角度の表があります．10 寸勾配まで表を完成させましょう．

勾配	1 寸勾配	2 寸勾配	3 寸勾配	...
	atan(0.1)	atan(0.2)	atan(0.3)	...
rad	0.0997	0.1974	0.2915	...
度	5.7106°	11.3099°	16.6992°	...

1.3.5　数列の和

数列の和を計算する関数 sum の書式は，sum（一般項 a_i，i，はじめの i，最後の i）です．

$\sum_{i=1}^{10} i^2$ の計算をします.

```
(%i36) sum(i^2, i, 1, 10);
(%o36) 385
```

一般の n までの和は $\sum_{i=1}^{n} i^2$ ですが, 答えがわからないときはそのままの形で返します.

```
(%i37) sum(i^2, i, 1, n);
(%o37) sum(i^2, i, 1, n)
```

nusum は n までの和を知っています. $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ です.

```
(%i38) nusum(i^2, i, 1, n);
(%o38) n (n+1) (2 n+1)/6
```

(練習) 数列の和

1. 初項 a_1, 公差 d の等差数列の第 1 項から第 n 項までの和を求めよ.
2. 初項が 3, 公比が 2 の等比数列について, 第 10 項までの和を求めよ.
3. $\sum_{i=1}^{n} b_1 r^{i-1}$ を求めよ. また, 無限等比数列の和 $\sum_{i=1}^{\infty} b_1 r^{i-1}$ を求めよ. ただし, $0 < r < 1$.

ヒント: r の範囲を宣言して...
一般項を定義して…
数列の和をとって…
$n \to \infty$ の極限の計算.

```
(%i39) assume(0<r, r<1);
(%i40) b(i):= b[1]*r^(i-1);
(%i41) nusum( b(i), i, 1, n );
(%i42) limit(%, n, inf);
```

1.3.6 関数のテイラー展開

$\sin x$ を $x = 0$ のまわりで x^5 までテイラー展開する例.

```
(%i43) taylor( sin(x), x, 0, 5 );
(%o43)/T/ x - x^3/6 + x^5/120 + ...
```

e^{ix} (i は虚数単位) のテイラー展開.

```
(%i44) taylor( exp(%i*x),  x, 0, 5 );
(%o44)/T/ 1 + %i x - x^2/2 - %i x^3/6 + x^4/24 + %i x^5/120 + ...
```

その虚部を imagpart でとると…
(ちなみに, 実部は realpart です.)

```
(%i45) imagpart(%o44);
(%o45) x^5/120 - x^3/6 + x
```

$\cos x, \tan x$ 等のテイラー展開もやってみましょう.

1.4　式の整理・簡単化

Maxima では，多項式やいろいろな関数を含んだ式の変形や整理・簡単化が可能です．

1.4.1　展開・因数分解・簡単化

$(x+1)^2$ を展開します．

```
(%i1)  expand( (x + 1)^2 );
```

$$(\%o1)\quad x^2+2\,x+1$$

x^3-1 を因数分解します．

```
(%i2)  factor( x^3 - 1 );
```

$$(\%o2)\quad (x-1)\left(x^2+x+1\right)$$

$\dfrac{3}{x^3-1}$ を部分分数に展開する例．

```
(%i3)  partfrac( 3/(x^3 - 1), x );
```

$$(\%o3)\quad \frac{-x-2}{x^2+x+1}+\frac{1}{x-1}$$

ratsimp を使って，上式を「簡単化」します．

```
(%i4)  ratsimp(%);
```

$$(\%o4)\quad \frac{3}{x^3-1}$$

1.4.2　三角関数を含む式の簡単化

三角関数や指数関数を含む式の簡単化には trigsimp 関数などを使います．

$\cos^2 x-\sin^2 x$ を「簡単化」します．

```
(%i5)  trigsimp( cos(x)^2 - sin(x)^2 );
```

$$(\%o5)\quad 2\cos\left(x\right)^2-1$$

trigreduce は三角関数同士の積をなるべく減らして「簡単化」します．

```
(%i6)  trigreduce( cos(x)^2 - sin(x)^2 );
```

$$(\%o6)\quad \frac{\cos\left(2\,x\right)+1}{2}+\frac{\cos\left(2\,x\right)}{2}-\frac{1}{2}$$

通分すればもっと簡単になりそうですね．「簡単化」してみます．

```
(%i7)  trigsimp(%);
```

$$(\%o7)\quad \cos\left(2\,x\right)$$

trigexpand 関数は加法定理や倍角の公式を使って「展開」します．

```
(%i8)  trigexpand( cos(3*x) );
```

$$(\%o8)\quad \cos\left(x\right)^3-3\,\cos\left(x\right)\,\sin\left(x\right)^2$$

trigsimp でもう少し簡単にします．

```
(%i9)  trigsimp(%);
```

$$(\%o9)\quad 4\cos\left(x\right)^3-3\,\cos\left(x\right)$$

（練習）加法定理・三倍角の公式

1. 三角関数及び双曲線関数の加法定理を確認しなさい．
2. $\sin 3x$ を $\sin x$ だけを使って表しなさい．また，$\tan 3x$ を $\tan x$ だけを使って表しなさい．

`trigexpand` で展開します．
`sin(x)` と `cos(x)` が混在する場合，`cos(x)^2` を `1-sin(x)^2` で置き換える例です．

```
(%i10) trigexpand( sin(x + y) );

(%i11) trigexpand( sin(3*x) );

(%i12) ratsubst(1-sin(x)^2, cos(x)^2, %);
```

1.5　微分・積分・数値積分

微分・積分を行う関数の表記を以下に示します．

操作	表記例
微分	`diff( f(x),  x )`
n 階微分	`diff( f(x),  x, n )`
不定積分	`integrate( f(x),  x )`
定積分	`integrate( f(x),  x, a, b )`
数値積分	`romberg( f(x),  x, a, b )`

1.5.1　微分・積分

x^2+5x+2 を x で微分します．

```
(%i1)  diff( x^3 + 5*x + 2, x );
```

(%o1)　$3\,x^2+5$

x^2+5x+2 を x で 2 階微分します．

```
(%i2)  diff( x^3 + 5*x + 2, x, 2 );
```

(%o2)　$6\,x$

$x^2\cos x+2x\sin x$ を x で積分します．

```
(%i3)  integrate(x^2*cos(x)+2*x*sin(x),x);
```

(%o3)　$2\,\left(\sin\left(x\right)-x\,\cos\left(x\right)\right)+\left(x^2-2\right)\,\sin\left(x\right)+2\,x\,\cos\left(x\right)$

もう少し簡単になりそうですね．

```
(%i4)  trigsimp(%);
```

(%o4)　$x^2\,\sin\left(x\right)$

定積分の例：$\displaystyle\int_0^{\pi/2} \sin x\,dx = 1$

```
(%i5)  integrate( sin(x), x, 0, %pi/2 );
(%o5)  1
```

（練習）関数の傾き

$y = \dfrac{2x-1}{x+1}$ が $x \neq -1$ で増加関数であることを示しなさい．

（練習）積分

1. $\displaystyle\int e^x \sin x\,dx$ を計算しなさい．
2. $\displaystyle\int \frac{x^2-2x-2}{x^3-1}\,dx$ を計算しなさい．

1.5.2　数値積分

解析的に積分できない場合は，数値積分によって近似値を求めることができます．

$\displaystyle\int_0^{\pi} \sin(\sin x)\,dx$ の計算をします．

解けないとそのままで返します．

```
(%i6)  integrate(sin(sin(x)), x, 0, %pi);
(%o6)  ∫_0^%pi sin(sin(x)) dx
```

romberg 関数を使うと，数値的に積分した結果を表示します．

```
(%i7)  romberg(sin(sin(x)), x, 0, %pi);
(%o7)  1.786487487137068
```

数値積分 romberg の精度は，大域変数 rombergtol で決まります．rombergtol の値を小さくすることで精度をあげることができます．

$\displaystyle\int_0^1 e^{-x^2}\,dx$ の数値積分

```
(%i8)  romberg(exp(-x^2), x, 0, 1);
(%o8)  .7468241699098985
```

デフォルトの rombergtol の値は…

```
(%i9)  rombergtol;
(%o9)  1.0e-4
```

rombergtol を 10^{-12} にして…

```
(%i10) rombergtol: 1.e-12;
(%o10) 9.999999999999999e-13
```

もう一度数値積分を行うと…

```
(%i11) romberg(exp(-x^2), x, 0, 1);
(%o11) .7468241328124268
```

1.6 方程式

1.6.1 多項式方程式・連立方程式

Maxima は，式の計算だけでなく，代数方程式を解くことができます．例を以下に示します．

$x^2+2x-2=0$ を x について解きます．

```
(%i1)  solve(x^2 + 2*x - 2 = 0, x);
(%o1)  [x = -sqrt(3) - 1, x = sqrt(3) - 1]
```

連立方程式 $x^2+y^2=5,\ x+y=1$ を x, y について解きます．

```
(%i2)  solve([x^2+y^2=5, x+y=1], [x, y]);
(%o2)  [[x = -1, y = 2], [x = 2, y = -1]]
```

`solve` の代わりに `algsys` も使えます．

```
(%i3)  algsys([x^2+y^2=5, x+y=1], [x, y]);
(%o3)  [[x = -1, y = 2], [x = 2, y = -1]]
```

3 次方程式 $x^3=8$ を解きます．

```
(%i4)  solve(x^3 = 8, x);
(%o4)  [x = sqrt(3) i - 1, x = -sqrt(3) i - 1, x = 2]
```

念のため，1 番目の答えを 3 乗して簡単化すると…

```
(%i5)  rhs(%o4[1])^3, ratsimp;
(%o5)  8
```

実数解のみを求める場合は，`realroots` を使います．

```
(%i6)  realroots(x^3 = 8);
(%o6)  [x = 2]
```

（練習）鶴と亀と蟻

鶴と亀と蟻，個体数の合計は 10．足の数は全部で 34 本．蟻は亀より 1 匹少ない．鶴，亀，蟻，それぞれ何羽・何匹？ 鶴と亀と蟻の個体数をそれぞれ x, y, z として連立方程式をたてて `solve` します．蟻の足は何本かわかりますよね？

1.6.2 方程式の数値解

1 変数多項式方程式の数値解を求める関数として，`allroots` や `find_root` があります．

$f(x) := x^3+8x-3$ と定義します．

```
(%i7)  f(x):= x^3 + 8*x - 3;
(%o7)  f(x) := x^3 + 8 x - 3
```

$f(x)=0$ の全数値解を求めます．

```
(%i8)  allroots(f(x) = 0);
(%o8)  [x = .3687331875693762,
x = 2.846396515370145 i - .1843665937846881,
x = -2.846396515370145 i - .1843665937846881]
```

`realroots` を使って $f(x) = 0$ の実数解を求めます.

```
(%i9)  realroots(f(x) = 0);
```

$$(\%o9)\quad [x = \frac{12372633}{33554432}]$$

`realroots` で求めた実数解を浮動小数点表示してみます.

```
(%i10) float(%);
```

$$(\%o10)\ [x = .3687331974506378]$$

`allroots` で求めた (`%o8`) の結果と少し違いますね.

```
(%i11) rhs(%o10[1]) - rhs(%o8[1]);
```

$$(\%o11)\ 9.881261586741631\,10^{-9}$$

精度を指定して再計算してみます.

```
(%i12) realroots(f(x)=0, 1e-16)$ float(%);
```

$$(\%o13)\ [x = .3687331875693762]$$

念のために, 答えを代入してみます.

```
(%i14) ev(f(x), %);
```

$$(\%o14)\ 0.0$$

多項式方程式でない場合には `find_root` 関数を使って, 範囲を指定して解を探します.

関数 $g(x) := (x-1)\,e^x$ の定義.

```
(%i15) g(x):= (x - 1)*exp(x);
```

$$(\%o15)\ \mathrm{g}\,(x) := (x-1)\,\exp(x)$$

$(x-1)\,e^x = 1$ の解は `solve` などでは求めることができません.

```
(%i16) solve(g(x) = 1, x);
```

$$(\%o16)\ [x = e^{-x}\ (e^x + 1)]$$

`find_root` を使って $0 \le x \le 2$ の範囲で解を探してみます.
出た答えを代入して確かめると…

```
(%i17) find_root(g(x) = 1, x, 0, 2);
```

$$(\%o17)\ 1.278464542761074$$

```
(%i18) g(%);
```

$$(\%o18)\ 1.0$$

(練習) 関数の極大値

次の関数が極大値をとる x の値およびその極大値を求めよ.

$$1.\ \ f(x) = \frac{x^3}{e^x - 1}, \qquad 2.\ \ g(x) = \frac{x^5}{e^x - 1}$$

ヒント：

$f(x)$ を定義して．..
それを微分して．..

```
(%i19) f(x):= x^3/(exp(x) - 1);
(%i20) diff(f(x), x);
(%i21) ratsimp(%);
```

導関数が 0 になる x を数値的に求める．

```
(%i22) define(df(x), %);
(%i23) find_root(df(x)=0, x, 1, 5);
```

1.6.3 常微分方程式：ode2

Maxima は 1 階および 2 階の常微分方程式も扱うことができます．まず ode2 関数を使って解く例を示します．

1 階常微分方程式と初期条件

微分方程式を記述する際は diff の前の’ に注意．
ode2 で $\frac{dy}{dt} = ay$ を解きます．

```
(%i24) 'diff(y, t) = a*y;
```

(%o24) $\frac{d}{dt}y = a\,y$

```
(%i25) ode2(%, y, t);
```

(%o25) $y = \%c\,e^{a\,t}$

%c は積分定数です．1 階常微分方程式の初期条件には ic1 を使い，例えば初期条件を $t = 0$ で $y = 1$ とすると積分定数 %c の値が決まります．

```
(%i26) ic1(%, t = 0, y = 1);
```

(%o26) $y = e^{a\,t}$

2 階常微分方程式と初期条件

2 階の常微分方程式 $\frac{d^2x}{dt^2} + Kx = 0$ を解く例です．

最初に $K > 0$ を仮定しておきます．これをしないと後で聞かれます．

```
(%i27) assume(K > 0);
```

(%o27) $[K > 0]$

ode2 関数を使って解くときの書式．%k1，%k2 は積分定数です．

```
(%i28) ode2('diff(x,t,2) + K*x = 0, x, t);
```

(%o28) $x = \%k1\,\sin\left(t\sqrt{K}\right) + \%k2\,\cos\left(t\sqrt{K}\right)$

2 階常微分方程式の初期条件は ic2 で設定します．初期条件を $t = 0$ で $x = x_0, v = v_0$ とすると %k1，%k2 が決まります．

```
(%i29) ic2(%, t=0,x=x[0],'diff(x,t)=v[0]);
```

(%o29) $x = \frac{v_0\,\sin\left(t\sqrt{K}\right)}{\sqrt{K}} + x_0\,\cos\left(t\sqrt{K}\right)$

細かいことですが，x[0] のようにすると Maxima は下付きの添字 x_0 として表示します．

1.6.4　常微分方程式：desolve

常微分方程式を解く関数としては desolve もあります． ode2 と書式が若干違います．また，初期条件は atvalue 関数で先に設定しておきます．2 階の常微分方程式 $\frac{d^2x}{dt^2} + Kx = 0$ を解く例を以下に示します．

初期条件の設定． $x(0) = x_0$

```
(%i30) atvalue(x(t), t = 0, x[0]);
```

(%o30) x_0

および $x'(0) = v_0$

```
(%i31) atvalue('diff(x(t), t), t=0, v[0]);
```

(%o31) v_0

微分方程式の左辺を定義しておきましょう．

```
(%i32) eq: 'diff(x(t),t,2) + K*x(t) = 0;
```

(%o32) $\mathrm{x}\,(t)\ K + \frac{d^2}{d\,t^2}\,\mathrm{x}\,(t) = 0$

desolve 関数を使って解くときの書式．

```
(%i33) desolve(eq, x(t));
```

(%o33) $\mathrm{x}\,(t) = \frac{v_0\,\sin\left(t\,\sqrt{K}\right)}{\sqrt{K}} + x_0\,\cos\left(t\,\sqrt{K}\right)$

ちなみに，$K \to 0$ の極限で，この解がどうなるかというと...

```
(%i34) limit(%, K, 0);
```

(%o34) $\mathrm{x}\,(t) = v_0\,t + x_0$

最初から $K = 0$ とした微分方程式 $\frac{d^2x}{dt^2} = 0$ を解いても...

```
(%i35) desolve('diff(x(t),t,2) = 0, x(t));
```

(%o35) $\mathrm{x}\,(t) = v_0\,t + x_0$

（練習）重力場中の投げ上げ運動

1. 高さ 0 から初速度 v_0 で鉛直上方に投げ上げた物体の時刻 t における高さ y を求めなさい．運動方程式は，
$$\frac{d^2y}{dt^2} = -g$$
2. 速度に比例する空気抵抗がある場合，運動方程式は以下のようになる．
$$\frac{d^2y}{dt^2} = -g - \beta\frac{dy}{dt}$$
このとき，高さ 0 から初速度 v_0 で鉛直上方に投げ上げた物体の時刻 t における高さ y を求めなさい．
3. 前問 2. で求めた解が $\beta \to 0$ のとき，前問 1. の答えに一致することを示しなさい．
（ヒント： limit(%, beta, 0)）

1.7　ベクトルと行列

1.7.1　3 次元ベクトルの表記

3 次元ベクトル $\boldsymbol{a} = (a_x, a_y, a_z)$ の表記例.

$\boldsymbol{a}$ の x 成分（第 1 成分）を取り出すには…

ベクトル $\boldsymbol{b}$ も同様に定義.

```
(%i1)  a: [a[x], a[y], a[z]];
(%o1)  [a_x, a_y, a_z]
(%i2)  a[1];
(%o2)  a_x
(%i3)  b: [b[x], b[y], b[z]];
(%o3)  [b_x, b_y, b_z]
```

1.7.2　内積

ベクトルの内積 $\boldsymbol{a} \cdot \boldsymbol{b}$ には `.`（ドット）を使います.

内積を利用して，ベクトルの大きさを返す関数 `norm` を定義します.

```
(%i4)  a . b;
(%o4)  a_z b_z + a_y b_y + a_x b_x
(%i5)  norm(v):= sqrt(v . v);
(%o5)  norm(v) := sqrt(v.v)
(%i6)  norm(a);
(%o6)  sqrt(a_z^2 + a_y^2 + a_x^2)
```

1.7.3　外積

ベクトルの外積 $\boldsymbol{a} \times \boldsymbol{b}$ を返す関数 cross を定義します.

```
(%i7)  cross(u, v):=
           [u[2]*v[3] - u[3]*v[2],
            u[3]*v[1] - u[1]*v[3],
            u[1]*v[2] - u[2]*v[1]];
```

$$(\%o7)\quad \mathrm{cross}\,(u, v) := [u_2\,v_3 - u_3\,v_2, u_3\,v_1 - u_1\,v_3, u_1\,v_2 - u_2\,v_1]$$

成分ごとに見てみます.

```
(%i8)  cross(a, b)[1];
```

$$(\%o8)\quad a_y\,b_z - b_y\,a_z$$

```
(%i9)  cross(a, b)[2];
```

$$(\%o9)\quad b_x\,a_z - a_x\,b_z$$

```
(%i10) cross(a, b)[3];
```

$$(\%o10)\quad a_x\,b_y - b_x\,a_y$$

（練習）ベクトルの内積，外積

$\boldsymbol{a} = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \sqrt{3})$, $\boldsymbol{b} = (-1, 1, -\sqrt{2})$ について以下の量を計算しなさい.

1. $|\boldsymbol{a}|$,　　2. $|\boldsymbol{b}|$,　　3. $\boldsymbol{a} \cdot \boldsymbol{b}$,　　4. $\boldsymbol{a} \times \boldsymbol{b}$

また二つのベクトルのなす角を θ としたときの $\cos\theta$ および $\sin\theta$ を求めなさい.

（練習）ベクトルの 3 重積

1. スカラー 3 重積の以下の性質を確かめなさい.

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b} \cdot (\boldsymbol{c} \times \boldsymbol{a}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b})$$

ヒント: `a.cross(b, c) - b.cross(c, a), ratsimp;`

2. ベクトル 3 重積の以下の性質を確かめなさい.

$$\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} - (\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{c}$$

1.7.4　行列の簡単な演算

変数 a, b を再利用する際には，以前の定義を消しておきましょう．

```
(%i11) kill(a, b);
```

(%o11) *done*

2 行 2 列の行列 A の定義．

```
(%i12) A: matrix([a, b], [c, d]);
```

(%o12) $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

transpose は転置行列．

```
(%i13) transpose(A);
```

(%o13) $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$

determinant は行列式．

```
(%i14) determinant(A);
```

(%o14) $a\,d - b\,c$

逆行列は A^^-1 のように書きます．

```
(%i15) A^^-1;
```

(%o15) $\begin{pmatrix} \frac{d}{a\,d-b\,c} & -\frac{b}{a\,d-b\,c} \\ -\frac{c}{a\,d-b\,c} & \frac{a}{a\,d-b\,c} \end{pmatrix}$

$A^{-1}A$ が単位行列になることを確かめます．行列の積も　.　（ドット）で表します．

```
(%i16) A^^-1 . A, ratsimp;
```

(%o16) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

回転行列 O の定義．

```
(%i17) O: matrix([cos(x), -sin(x), 0],
                 [sin(x),  cos(x), 0],
                 [0,       0,      1]);
```

(%o17) $\begin{pmatrix} \cos(x) & -\sin(x) & 0 \\ \sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{pmatrix}$

直交行列であること．$O\,O^T$ が単位行列になることをを示します．

```
(%i18) O. transpose(O)$
       trigsimp(%);
```

(%o19) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

3 次元ベクトルを，3 行 1 列の行列として定義します．このように成分を縦に並べたものを列ベクトルと言うんでしたね．

```
(%i20) r: matrix([x], [y], [z]);
```

$$(\%o20)\quad \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

3 次元ベクトルの直交変換の例．

```
(%i21) O . r;
```

$$(\%o21)\quad \begin{pmatrix} x\,\cos(x) - \sin(x)\,y \\ \cos(x)\,y + x\,\sin(x) \\ z \end{pmatrix}$$

（練習）直交変換されたベクトルの大きさ

$\boldsymbol{r}' \equiv O\,\boldsymbol{r}$ の大きさは $\boldsymbol{r}$ の大きさと同じであることを示しなさい．

ヒント： `(O.r).(O.r)`

第 2 章
wxMaxima によるグラフ作成

wxMaxima を使って，関数のグラフを描くことができます．また，数値データもグラフにするすることができます．

2.1　2 次元グラフ plot2d の例

2 次元グラフ $y = f(x)$ を描く一般的な表記例は以下の通りです．

数式	表示範囲	wxMaxima での一般的表記例
$y = f(x)$	$a \leq x \leq b$	`plot2d(f(x), [x, a, b])`

例として，$y = \sin x$ のグラフを $-2\pi \leq x \leq 2\pi$ の範囲で描きます．基本的な定数の一つである円周率 π は Maxima では `%pi` と書くのでしたね．

```
(%i1)  plot2d(sin(x), [x, -2*%pi, 2*%pi])$
```

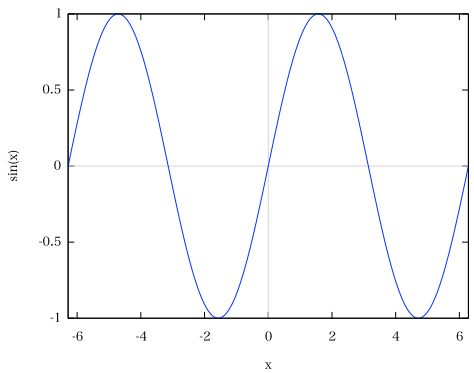


図 2.1　`(%i1)` の実行結果

2.2　3 次元グラフ plot3d の例

3 次元グラフ $z = g(x, y)$ を描く一般的な表記例は以下の通りです．

数式	表示範囲	wxMaxima での一般的表記例
$z = g(x, y)$	$a \leq x \leq b,\ c \leq y \leq d$	`plot3d(g(x,y), [x, a, b], [y, c, d])`

例として，$z = x \sin y$ のグラフを $-3 \leq x \leq 3,\ -4 \leq y \leq 4$ の範囲で描きます．

```
(%i2)  plot3d(x*sin(y), [x, -3, 3], [y, -4, 4])$
```

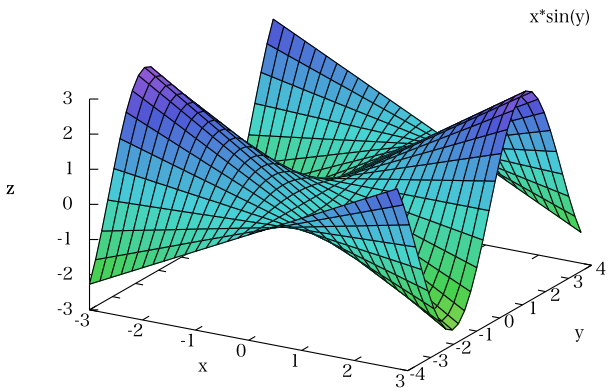


図 2.2　(%i2) の実行結果

2.3　数値データ・リストのグラフ作成例

wxMaxima は，リスト形式のリスト形式の数値データもグラフ化することができます．以下の例では，最初（左側）の数値を x 座標の値，次（右側）の数値を y 座標の値として 6 個の点からなるリスト `data` の各点をつないだ折れ線グラフを描きます．

```
(%i3)  data: makelist([i, i^2], i, 0, 5);
(%o3)  [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25]]
(%i4)  plot2d([discrete, data])$
```

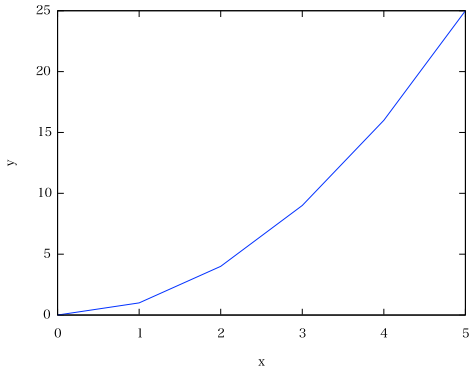


図 2.3　(%i4) の実行結果

オプションを設定してプロットする例．ここでは，[style，points] を設定して線でつながずに点で描きます．プロットオプションの style には，points の他にも，lines，linespoints，dots が設定できます．

```
(%i5)  plot2d([discrete, data], [style, points])$
```

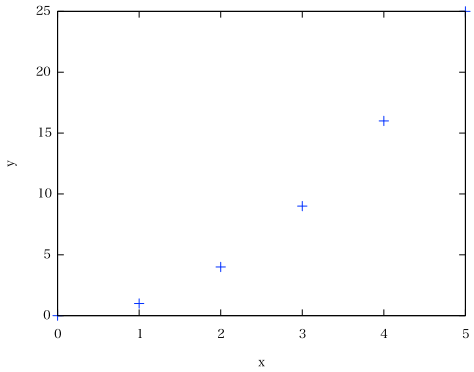


図 2.4　(%i5) の実行結果

2.4　複数のグラフを重ねて表示

wxMaxima で複数のグラフを重ねて表示する例を示します．$y = x^2 - 1$ と $y = 4x - 5$ の 2 つのグラフを重ねて描きます．x の範囲は $-5 \leq x \leq 5$ です．以下の例でわかるように，重ねて描きたい関数を [x^2-1,4*x - 5] のように [と] で囲んだリスト形式にして plot2d コマンドを使います．

```
(%i6)  plot2d([x^2 - 1, 4*x - 5], [x, -5, 5])$
```

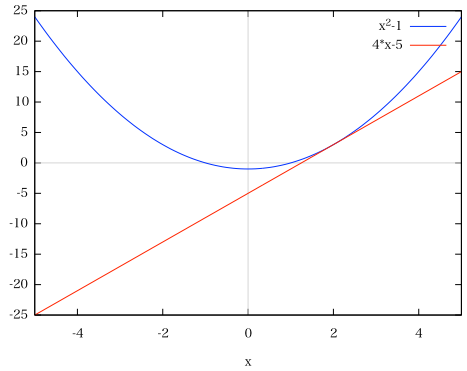


図 2.5　(%i6) の実行結果

いくつかオプションを設定して描く例です．以下の例では，縦軸 (y 軸) の表示範囲を制限したために，`plot2d: some values were clipped.` というメッセージが出ています．

```
(%i7)  plot2d([x^2 - 1, 4*x - 5], [x, -5, 5],
              [y, -5, 10],                      /* 縦軸の表示範囲の設定 */
              [color, red, black],                    /* 線の色の設定 */
              [style, [lines, 4], lines],       /* 片方の線の太さを 4 に */
              [gnuplot_preamble, "set key bottom"])$      /* 凡例の位置 */

plot2d: some values were clipped.
plot2d: some values were clipped.
```

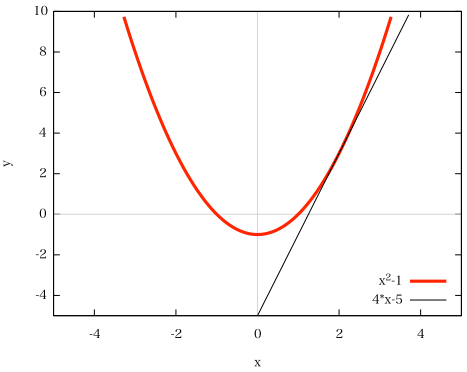


図 2.6　(%i7) の実行結果

上のグラフをみると，2 つのグラフは 1 点で接しているように見えます．実際に $y = 4x - 5$ が接線であることを示してみましょう．

```
(%i8)  /* 2 次曲線 f(x) = x^2 -1 及びその微分 */
       f(x):= x^2 - 1; define(df(x), diff(f(x), x));
```

(%o8) $\mathrm{f}(x) := x^2 - 1$
(%o9) $\mathrm{df}(x) := 2\,x$

```
(%i10) /* 交点を求める */   kai: solve( f(x) = 4*x-5, x );
```

(%o10) $[x = 2]$

```
(%i11) /* 接点 P の座標及び接線の傾き */
       P: ev([x, f(x)], kai)$   m: ev(df(x), kai)$
(%i13) /* 最終的に接線の方程式は... */
       y = m*(x-P[1]) + P[2], ratsimp;
```

(%o13) $y = 4\,x - 5$

できたら，$y = x^2 - 1$ の $x = 1$ 等における接線のグラフも描いてみましょう．

2.5 数値データファイルを読み込んでグラフにする

wxMaxima では，あらかじめ作成された数値データファイルを読み込んでグラフを描くこともできます．
右のような内容のファイル `test.dat` が`/Users/kasai` ディレクトリにあるとします．実際にやってみる場合は，以下の例のディレクトリ名 (`/Users/kasai`) は各自の環境に合わせて変更して下さい．

```
0	0
1	1
2	4
3	9
4	16
5	25
```

データファイルを読み込む wxMaxima の関数は，`read_nested_list` を使います．以下の例のように数値データを読み込んで，グラフを描くことができます．

```
(%i14) dat: read_nested_list("/Users/kasai/test.dat")$
(%i15) plot2d([discrete, dat],
              [x, 0, 6], [y, 0, 28],          /* x, y の表示範囲を設定 */
              [style, [points, 2, 1, 5]],      /* 大きさ，色，点の種類 */
              [gnuplot_preamble, "set key top left"],   /* 凡例の位置 */
              [legend, "data"])$                      /* 凡例表示の設定 */
```

上の例で，`[points, 2, 1, 5]` の部分の最初の数値は大きさを表します．2 番目の数値は色を表し，1：青，2：赤，...などです．3 番目の数値は点の種類を表します．それぞれの数値を変えてみて，各自の環境でどのように見えるか確認してみましょう．

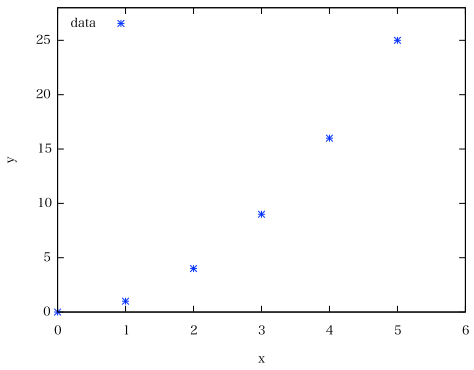


図 2.7　(%i15) の実行結果

2.6　数値データと理論曲線を重ねて表示する

前節の数値データファイル `test.dat` と理論曲線 $y = x^2$ の 2 つのグラフを重ねて表示してみます．ここでは，凡例がグラフと重ならないように縦軸の表示範囲を `[y, 0, 28]` と設定しています．

```
(%i16) plot2d([[discrete, dat], x^2], [x, 0, 5], [y, 0, 28],
              [style, points, lines], [legend, "data", "y=x^2"])$
```

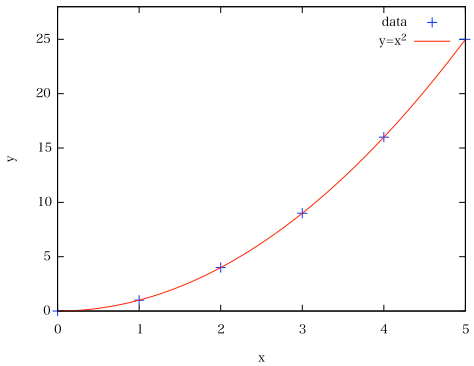


図 2.8　(%i16) の実行結果

2.7　媒介変数表示の 2 次元グラフ

半径 1 の円の方程式は $x^2+y^2=1$ です．この円を描くには，$y = \pm\sqrt{1-x^2}$ として $y = f(x)$ の形にするよりも，以下のような媒介変数表示にしたほうが簡単でしょう．

$$x = \cos t, \quad y = \sin t, \quad (0 \le t \le 2\pi)$$

媒介変数表示の 2 次元グラフを wxMaxima で描くには以下のようにします．(媒介変数は `t` にします．それ以外ではエラーが出てしまうようです．)

```
(%i17) plot2d([parametric, cos(t), sin(t), [t, 0, 2*%pi]])$
```

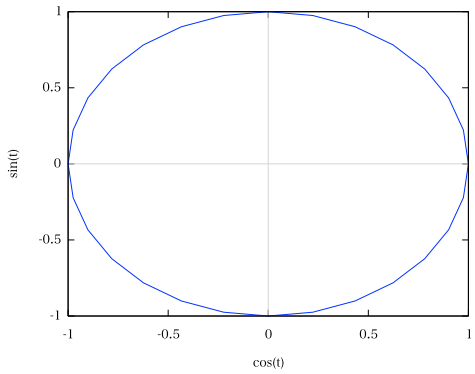


図 2.9　(%i17) の実行結果

これは媒介変数表示の円のグラフですが，少しカクカクしていて横につぶれて見えます．せっかくですから，もう少し滑らかになるように nticks の値を大きめに設定し，グラフの縦横の比を 1:1 にして円らしく見えるように，"set size square"を設定してみます．

```
(%i18) plot2d([parametric, cos(t), sin(t), [t, 0, 2*%pi], [nticks, 100]],
              [gnuplot_preamble, "set size square"],
              [xlabel, "x"], [ylabel, "y"], [legend, "x^2+y^2=1"],
              [x,-1.1, 1.1], [y, -1.1, 1.1])$
```

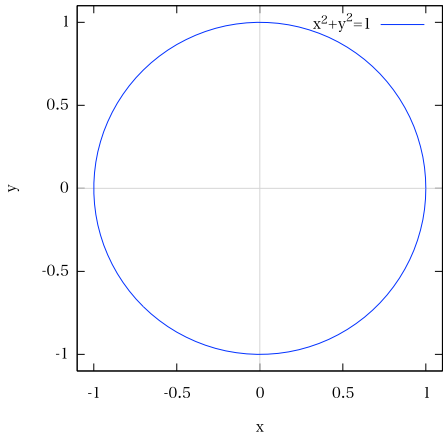


図 2.10　(%i18) の実行結果

（練習）楕円のグラフ（楕円中心を原点にして）

同様にして，楕円のグラフを描くこともできます．原点を中心とし，長半径 a、単半径 b の楕円の式は

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

です．媒介変数表示では，$x=a\cos t,\quad y=b\sin t,\quad (0\le t\le 2\pi)$ と書けます．a，b に適当な値を入れて楕円のグラフを描きなさい．

2.8　海王星と冥王星の軌道（焦点を原点とした楕円のグラフ）

太陽からの万有引力を受けて運動する惑星は，太陽（二体問題では質量中心）を焦点の一つとした楕円運動を描いて運動します．焦点の一つを原点とし，長半径 a，離心率 e の楕円の方程式は，極座標 r，φ を使って以下のように表すことができます．

$$r=\frac{a(1-e^2)}{1+e\cos\varphi}$$

例として海王星の軌道を描きます．海王星の軌道長半径 a_N は 30.1 天文単位，離心率 e_N は 0.009 であり十分小さいので，以下では簡単のため海王星の離心率は 0 として扱います．
まず，極座標表示の楕円の式を関数として定義します．

```
(%i19) r(a,e,t):= a*(1-e^2)/(1+e*cos(t));
```

$$(\%o19)\ \mathrm{r}\,(a,e,t):=\frac{a\,\left(1-e^2\right)}{1+e\,\cos\left(t\right)}$$

曲線が滑らかになるように，あらかじめ `nticks` を 100 に設定しておきます．

```
(%i20) set_plot_option([nticks,100])$
```

$x=r(a,e,t)\cos t,\ y=r(a,e,t)\sin t$ とした媒介変数表示で楕円を描きます．

```
(%i21) plot2d([parametric, r(30.1, 0, t)*cos(t), r(30.1, 0, t)*sin(t),
               [t, 0, 2*%pi]],
               [gnuplot_preamble, "set size square"],
               [xlabel, ""], [ylabel, ""])$
```

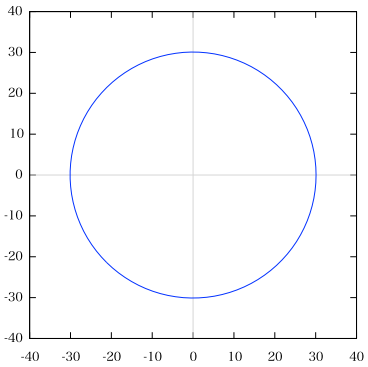


図 2.11　(%i21) の実行結果

では，この海王星の軌道と，その外側をまわる準惑星，冥王星の軌道を重ねて描いてみましょう．冥王星の軌道長半径 a_P は 39.6，離心率 e_P は 0.249 です．実際の二つの軌道は同じ平面上にありませんが，太陽からの距離のみをみるために，同一平面上に描きます．

うまくできましたか？ 以下のような結果が出るように設定を考えて下さい．

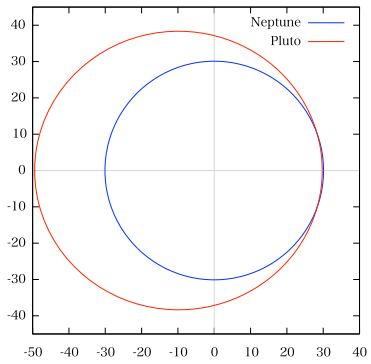


図 2.12　天王星と冥王星の軌道

ヒント：半径 1 と 2 の二つの円を重ねて描く例：

```
(%i22) plot2d([
               [parametric,   cos(t),   sin(t), [t, 0, 2*%pi]],
               [parametric, 2*cos(t), 2*sin(t), [t, 0, 2*%pi]]
              ],
              [gnuplot_preamble, "set size square"],
              [legend, "r=1", "r=2"])$
```

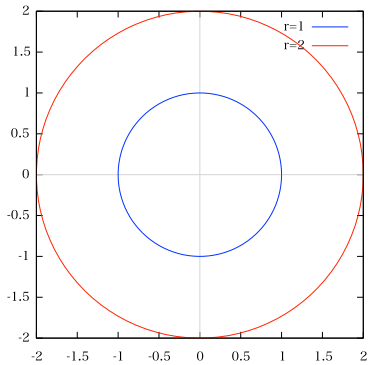


図 2.13　(%i22) の実行結果

さて，図 2.12 の海王星と冥王星の軌道のグラフを見ると，軌道が交差しているように見えます．このことを確かめてみます．海王星の軌道は円としますから，以下の式を満たす角度 ϕ を求めればよいことになります．

$$r(a_P, e_P, \phi) = a_N$$

ここで，$a_P = 39.6$，$e_P = 0.249$，$a_N = 30.1$ でしたね．$0 < \phi < \pi/2$ の範囲のどこかで交差

しているようですから，この範囲で数値的に解を求める方法は以下の通りです．

```
(%i23) aN: 30.1$ eN: 0$
       aP: 39.6$ eP: 0.249$
(%i27) phi1: find_root(r(aP,eP,t) = aN, t, 0, %pi/2);
(%o27) 0.348340241522416
```

$-\pi/2 < \phi < 0$ の範囲でも交差していますね．対称性から答えは (%o27) に負号をつけたものになるのは明らかですが，念のために確かめます．

```
(%i28) phi2: find_root( r(aP,eP,t)=aN, t, -%pi/2, 0);
(%o28) − 0.348340241522416
```

求めた角度はラジアン単位ですから，度に直してみます．

```
(%i29) fpprintprec: 3$   /* 表示するケタ数を 3 に */
       float(phi1*180/%pi);
(%o30) 20.0
```

この角度の 2 倍，つまり一周 360 度のうちに約 40 度にあたる分は海王星のほうが冥王星より太陽から遠い位置にあることがわかります．

原点と 2 つの軌道の交点を結ぶ直線を描く方法は以下の通りです．

```
(%i31) plot2d([ [discrete, [[0, 0], [aN*cos(phi1), aN*sin(phi1)]]],
                [discrete, [[0, 0], [aN*cos(phi2), aN*sin(phi2)]]]
              ],
              [gnuplot_preamble, "set size square"],
              [x,-50,40], [y,-45,45],
              [xlabel, ""], [ylabel, ""], [legend, ""],
              [color, black, black])$
```

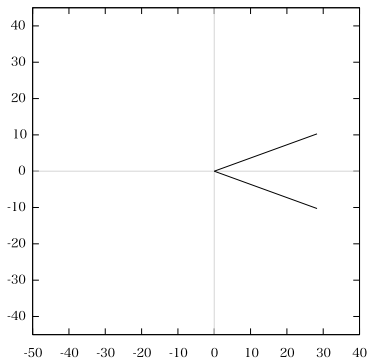


図 2.14　(%i31) の実行結果

最終的に 2 つの軌道も重ねて描いて以下のようなグラフを完成させてみましょう．

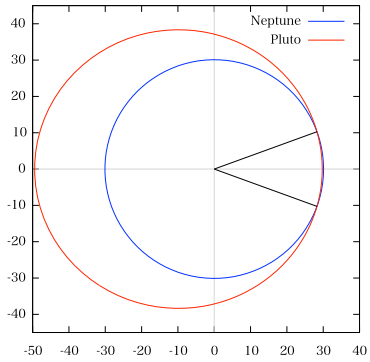


図 2.15　海王星と冥王星の軌道及び太陽からの距離が等しくなる点

2.9　月別平年気温のグラフと関数フィット

青森市の月別平年気温のデータを使って行列 M を作ります.

```
(%i32) M: matrix([ 1, -1.4],
                 [ 2, -1.1],
                 [ 3,  2.0],
                 [ 4,  7.9],
                 [ 5, 13.1],
                 [ 6, 17.0],
                 [ 7, 21.1],
                 [ 8, 23.0],
                 [ 9, 18.9],
                 [10, 12.6],
                 [11,  6.4],
                 [12,  1.3])$
```

このデータをグラフに描いてみます. 数値データをグラフにする際には, すでに説明したように plot2d([discrete, ...]) を使いますが, 使用するデータはリスト形式である必要があります. 今回は, すぐ後で述べる理由により, 行列形式でデータを準備したので, args 関数を使って行列をリストに変換して plot2d に渡します.

```
(%i33) plot2d([discrete, args(M)], [style, points])$
```

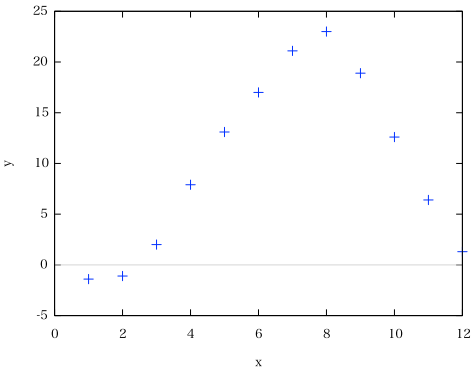


図 2.16　(%i33) の実行結果

グラフを見ると，月別平年気温は 12 ヶ月周期の正弦関数または余弦関数のように見えます．では，以下のように関数フィットをしてみましょう．

まず，最小二乗法のパッケージ `lsquares` をロードします．これは `lsquares_estimates` を初めて使う際，最初に 1 回だけ行えばよいです．

```
(%i34) load(lsquares)$
```

次に，次のような関数を定義して，うまくフィットするように定数 `a`，`b`，`c` を求めます．最小二乗法を行う関数 `lsquares_estimates` には数値データを行列として入れてやる必要があります．このために行列形式で月別平年気温のデータを準備したのでした．

```
(%i35) f(x):= a - b*cos(2*%pi*(x-c)/12);
```

$$(\%o35)\ \mathrm{f}\,(x) := a - b\,\cos\left(\frac{2\,\pi\,(x-c)}{12}\right)$$

```
(%i36) lsquares_estimates(M, [x, y], y=f(x), [a, b, c]);
```

（出力メッセージは省略）...

$$(\%o36)\ [[a = 10.1, b = 11.8, c = 1.48]]$$

最小二乗法によって求めた `a`，`b`，`c` の値を `f(x)` に代入し，この関数を描きます．

```
(%i37) fit: ev(fit(x), %);
```

$$(\%o37)\ 10.1 - 11.8\,\cos\left(\frac{\pi\,(x-1.48)}{6}\right)$$

```
(%i38) plot2d(fit, [x, 0.5, 12.5], [y, -5, 28],
              [style,lines], [color, red],
              [gnuplot_preamble, "set xtics 1"],
              [xlabel, "month"], [ylabel, "average temp."])$
```

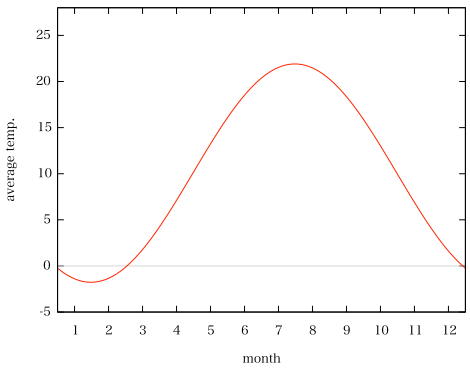


図 2.17　(%i38) の実行結果

最終的に，月別平年気温の数値データと余弦関数によるフィット曲線を重ねて描いて以下のようなグラフを完成させて下さい．

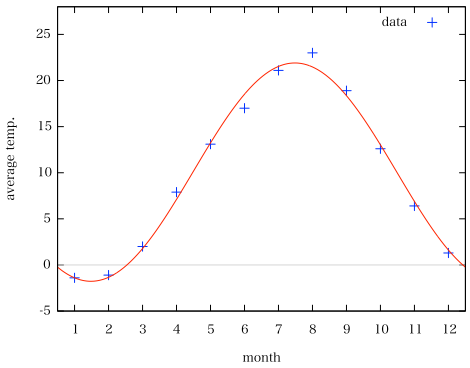


図 2.18　月別平年気温の数値データと余弦関数によるフィット曲線

ちなみに，関数フィットした際の定数 a は月別平均気温の平均値に相当します．平均値を求めるだけなら，最小二乗法などやらなくても以下のようにして求めることができます．行列 M の転置行列 transpose(M) の 2 行目（[2] で表します）が月別平年気温のデータです．それらを足し合わせ，個数 length(temp) で割ります．

```
(%i49) temp: transpose(M)[2];
(%o49) [−1.4, −1.1, 2.0, 7.9, 13.1, 17.0, 21.1, 23.0, 18.9, 12.6, 6.4, 1.3]
(%i50) sum(temp[i], i, 1, length(temp))/length(temp);
(%o50) 10.1
```

索　引