Maple 10 入門

【初級】

【中級】

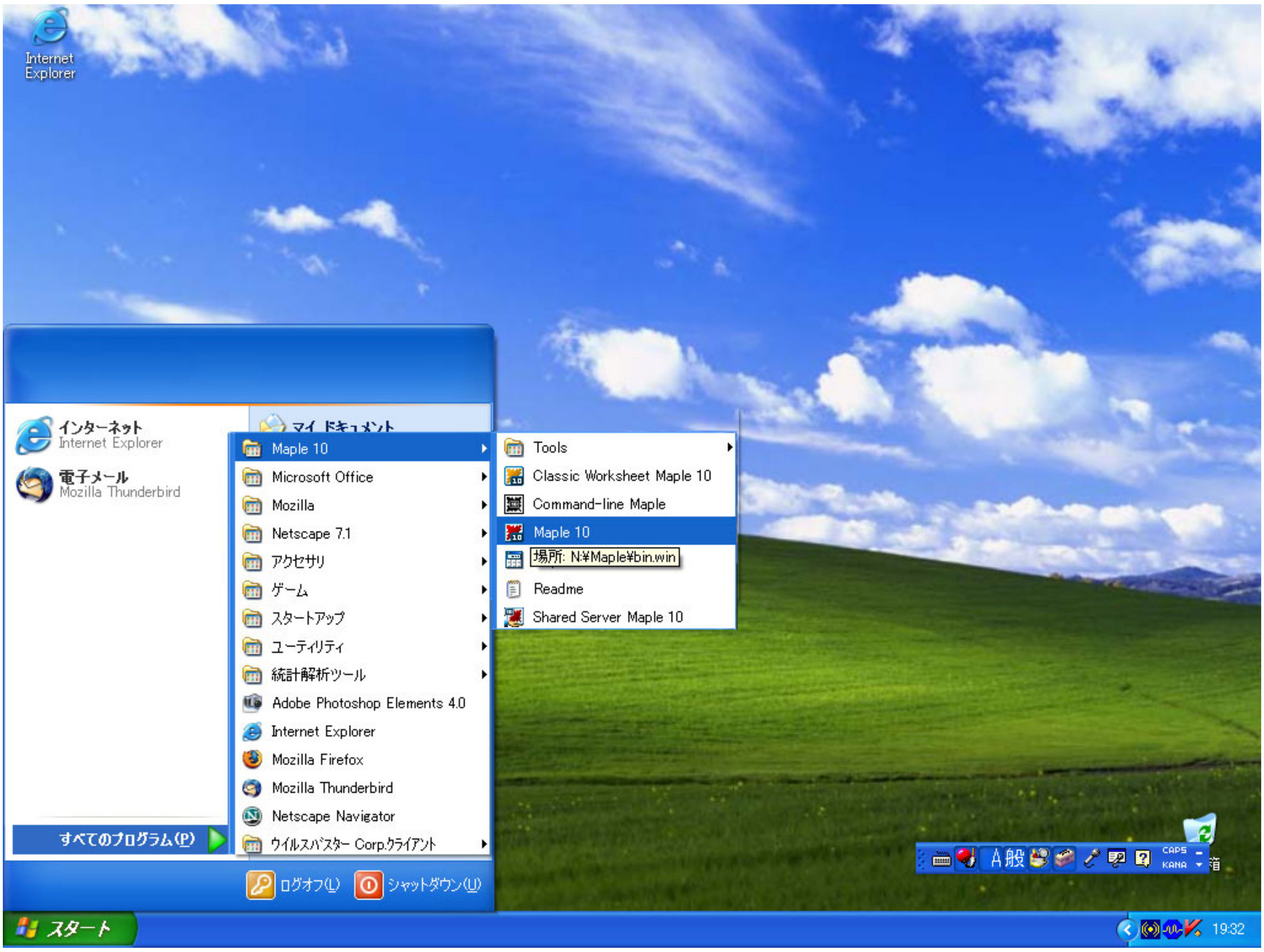
Maple チュートリアル【初級】

『Maple』は、数式処理システムと呼ばれているソフトウェアです。数式処理は、人間が行うような難解な数学の計算をコンピュータ上で実現してしまうものです。さらに、単なる計算だけではなく、2 次元や 3 次元のグラフを手軽に描いたりすることもできるので、難しくて頭の中にイメージできなかった数学を手軽に体験できるようになるでしょう。
Maple を利用して、新しい数学の世界を覗いてみましょう！

1　Maple の起動と終了

1.1　起動

Maple を起動するには、Windows の［スタート］->［プログラム］から［Maple 10］を選択し、［Maple 10］を選択します。

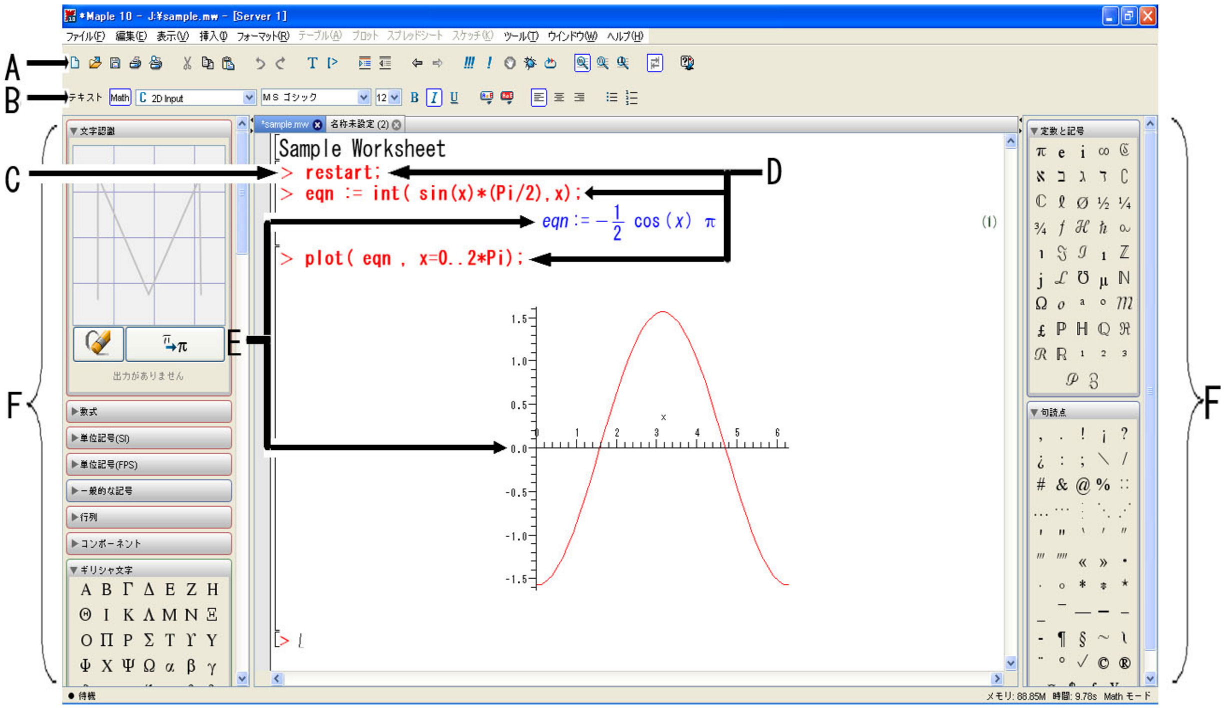


1.2　終了

Maple を終了させるには、画面右上の×をクリックします。

もしくは、［ ファイル ］メニューの［ 閉じる ］を選択することで Maple を終了させることが出来ます。

2　　ワークシート/インターフェイス



A - ツールバー
よく利用するメニューへのショートカットがアイコン化されているメニューが含まれます。

B - コンテクストバー
出力に応じて変化するメニューが含まれます。

C - プロンプト
デフォルトでは『>』で表示されています。この記号の右側にコマンドを入力していきます。

D - 入力
実行するコマンドの入力。デフォルトでは赤色の文字等で表示されます。

E - 出力
コマンド実行による出力の表示。デフォルトでは青色の文字等で表示されます。

F - パレット
キー入力の手間を省く為予めコマンドをアイコン化し用意しているテンプレート群です。パレットには、[数式], [シンボル], [行列], [ベクトル]などのパレットが用意されています。

Maple では計算が正常に実行されますと、次の入力プロンプトへカーソルが移動されますが、処理によっては中々計算が終わらないものもあります。そういった際に、計算を途中で終了させることも可能になりその際は、ツールバーの

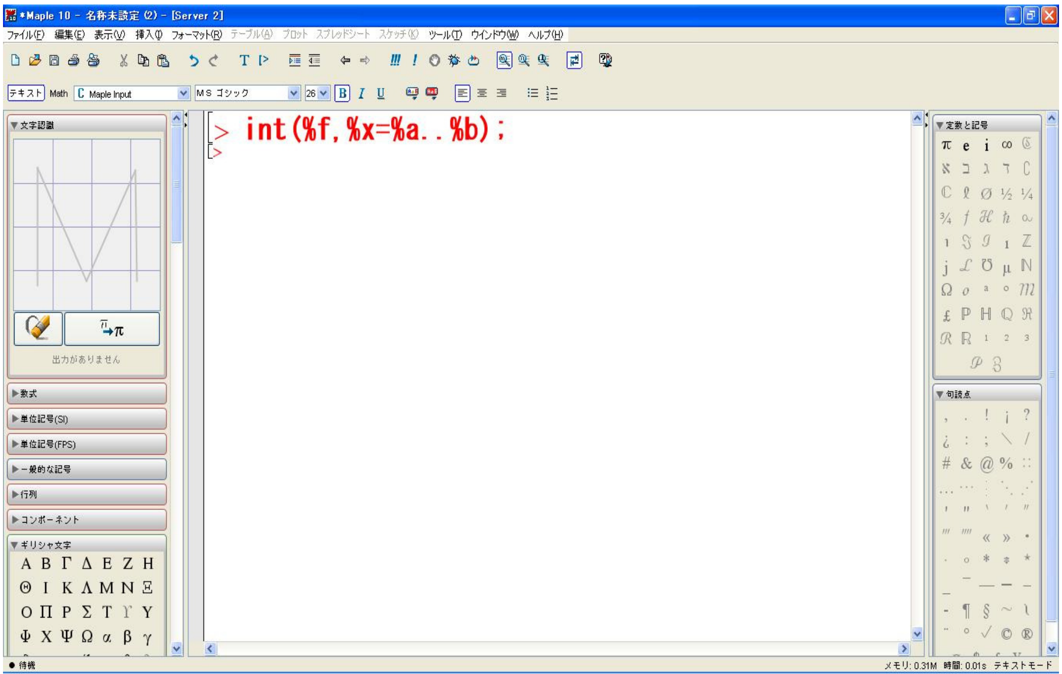
を押すことにより計算を途中で終了させることが出来ます。

3　　入力補助機能

3.1　　パレット入力

パレット入力とは予め用意されているショートカットボタンを使用し入力を行う方法です。
パレットの表示、非表示は、[表示] メニューの [パレット] より選択出来ます。

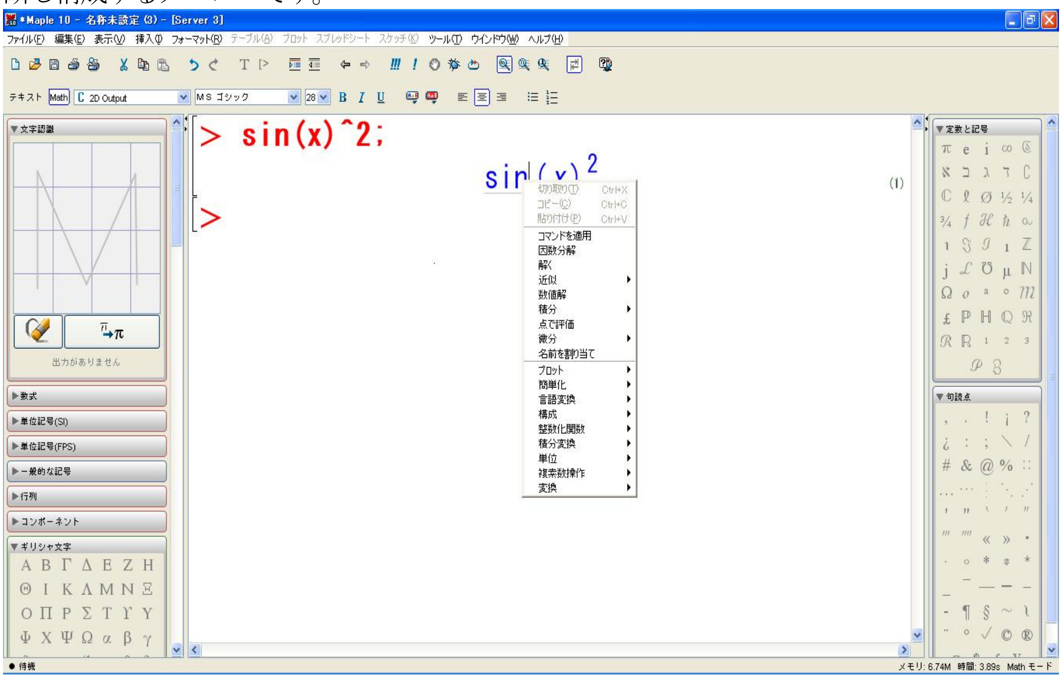
ボタンを押すと自動的にコマンドがプロンプトに挿入されますので、%?(? は f や x, a, b を指します)の部分に数式や数値などを入力し実行します。



3.2　　コンテキストメニュー

コンテキストメニューとは Maple の出力の上で右クリックすることで現れるメニューです。

コンテキストメニューでは、与えられた出力に対して「次にどのような処理を行うか」を Maple が自動的に判断し構成するメニューです。

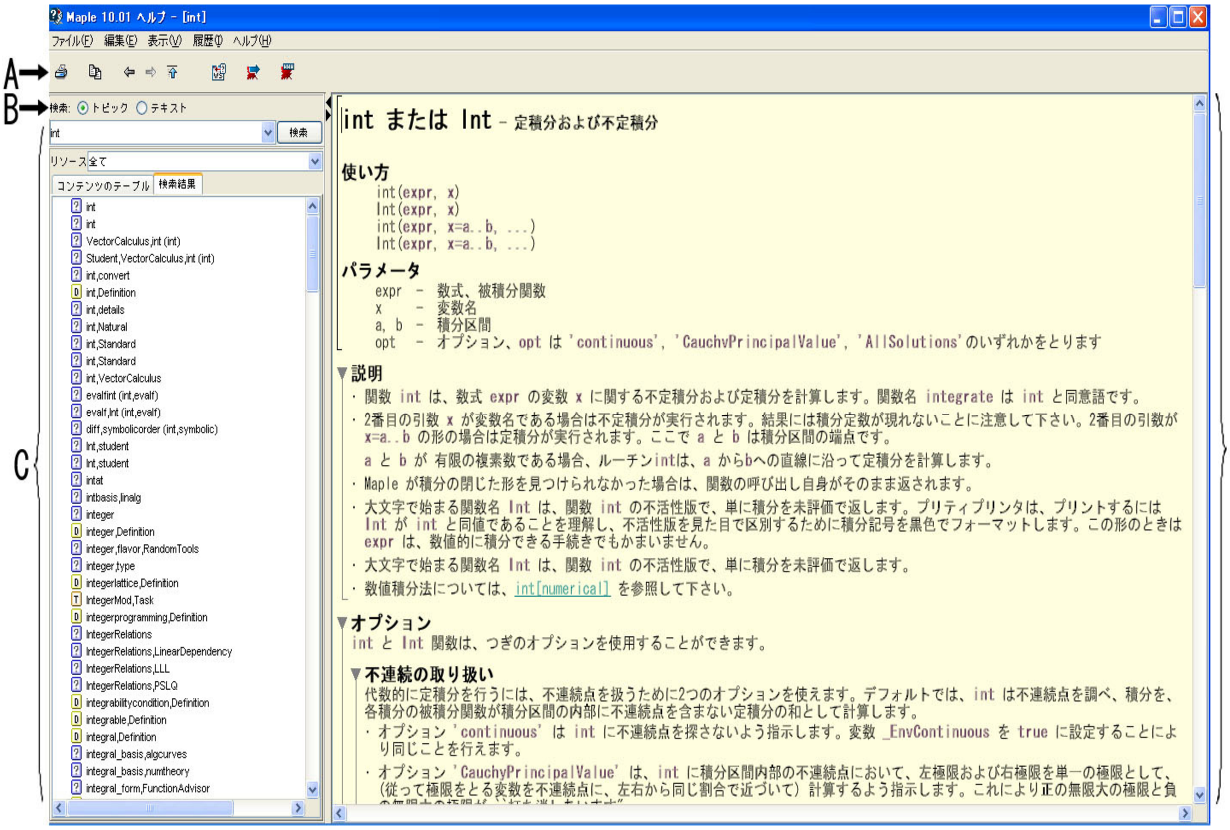


4　　ヘルプの使用方法

Maple のヘルプページには、そのコマンドの使用方法やオプションの指定方法、コマンドの使用例が記載されています。

ヘルプページへのアクセス方法は［ヘルプ］メニューよりアクセスするか、ワークシート上で下記のように ?（クエスチョン記号）に続けてコマンドを入力することでアクセスできます。

＞ ?int



A - ツールバー
現在のページの印刷やコピーなどのショートカットボタンが含まれます。

B - ヘルプナビゲータタブ
［ コンテンツ ］や［ トピック ］,［ 検索 ］,［ 辞書のコンテンツ ］,［ 履歴 ］と行ったヘルプページへアクセスする為のメニューがタブで用意されます。

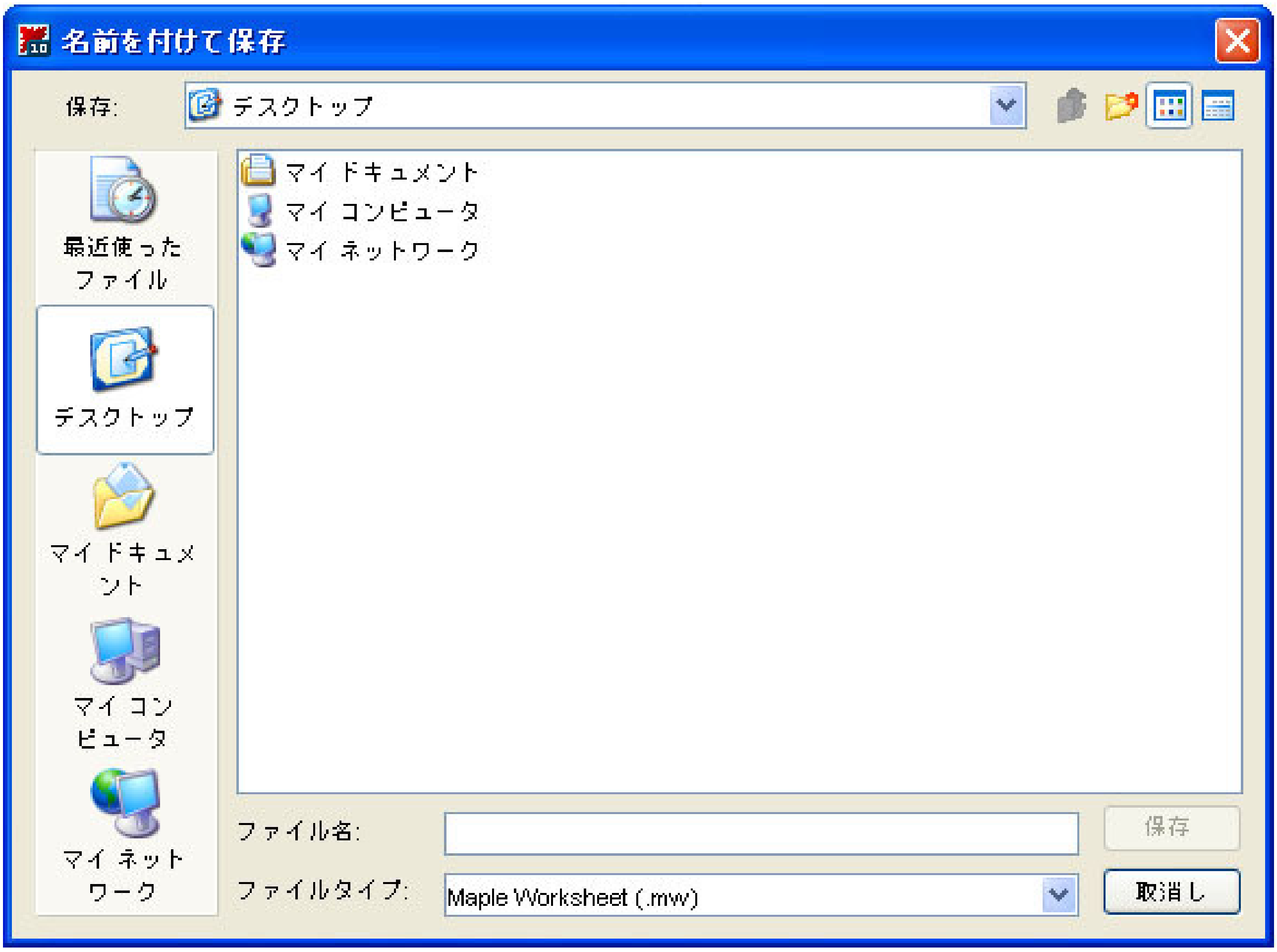
C - トピックツリー
ヘルプページをツリー構造で表示しています。

D - ヘルプページ
コマンドの使用方法や説明、オプションについて、使用例などが記載されています。

5　　ファイルの保存

5.1　　ワークシートの保存

ワークシートを保存するには、下記の手順で行うことで保存が出来ます。



1.［ ファイル ］メニューより［ 名前を付けて保存 ］を選択します。

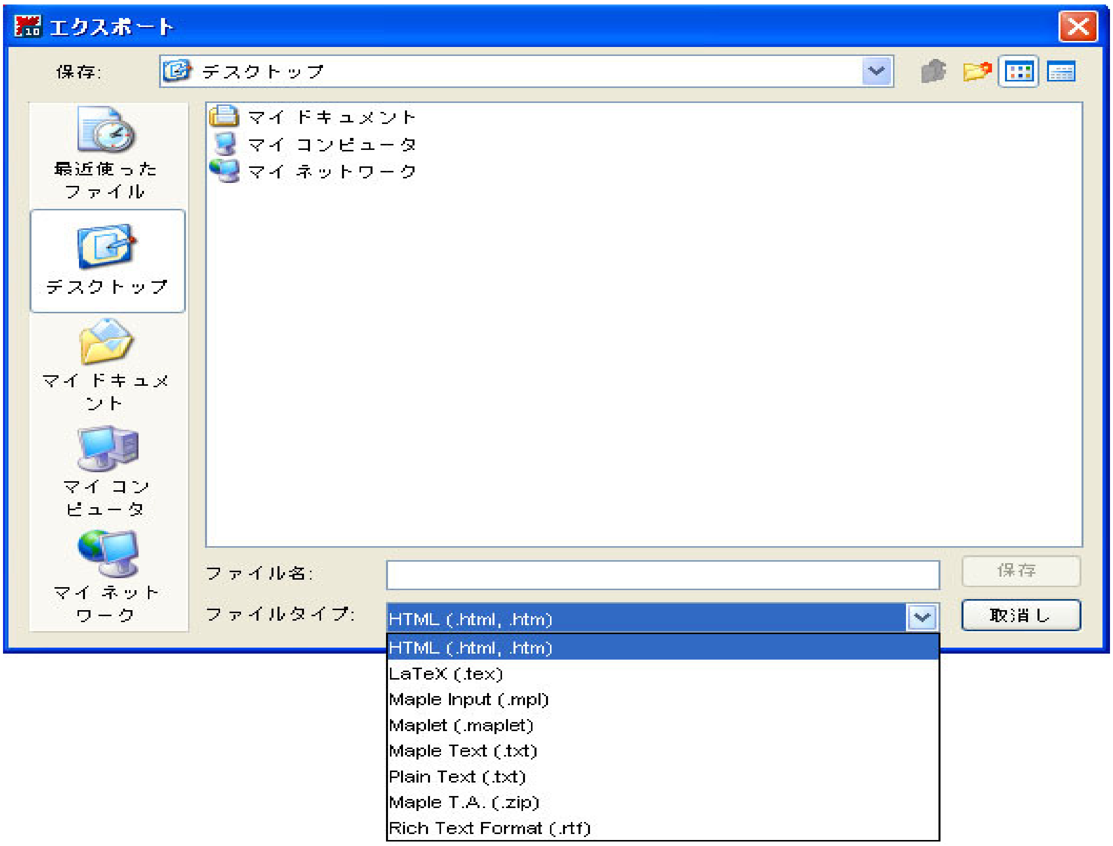
2. ファイル名及び、保存場所を指定します。

3.［ 保存 ］をクリックします。

また、保存したワークシートは、そのアイコンをダブルクリックするだけで起動することが出来ます。

5.2　　他のファイル形式への変換

Maple のワークシートは、下記の手順で html や RTF, LaTeX 等のファイル形式へ変換することが出来ます。



1. [ファイル] メニューより [ファイル変換] を選択します。

2. [ファイルタイプ] より、変換したい形式を選択します。

3. ファイル名及び、保存場所を指定します。

4. [保存] をクリックします。
[ファイルタイプ] より html を選択した場合は、下記のウィンドウが表示されます。

ここで、[イメージサブディレクトリ名] とはプロットなどの画像を保存するフォルダ名の指定になります。

また、[数式の保存形式] とは Maple の返す結果を画像ファイルとして保存するかもしくは MathML という規格を使用するかの指定になり、[GIF] を選択した場合、ファイルは [イメージサブディレクトリ名] で指定したフォルダに保存されます。[フレームを使用] にチェックを入れた場合は、html のフレーム形式で保存されます。従って、ワークシート内にセクションなどを使用している場合には有効ですが、使用していない場合はこのチェックを外されることをお勧めします。

6　　使い始めましょう

それではさっそく Maple を使ってみましょう。

Maple では入力の最後に : (コロン) もしくは ; (セミコロン) を入力し Enter キーを押すことで計算が実行されます。

この章では、まず一番最初に ： と ； の違いについて説明します。

；(セミコロン) を使用した場合は、計算が行われた後、入力した下の行に青字で出力（計算結果）が返されます。

＞ 12 ＊ 5 ;

$$60$$

一方、: (コロン) を使用した場合は、内部で計算は行われますが、出力は何も返されません。これは、長い計算結果の表示をコントロールするときに用いられます。

＞ 12 ＊ 5 :

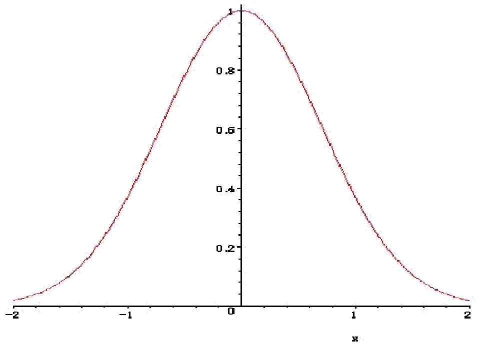
Maple では、数値だけでなく、文字（変数）などを用いた計算も可能です。

＞ a^2 + a^2 - x^2 + 3*x^2 ;

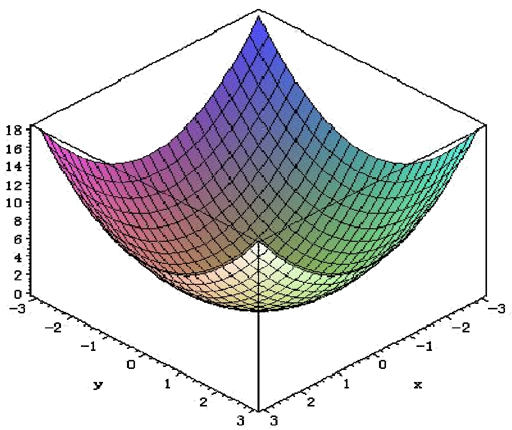
$$2a^2 + 2x^2$$

さらに、２次元・３次元のグラフも描画できます。３次元グラフは、マウスでドラッグすることで自由に回転させることができるようになっています。

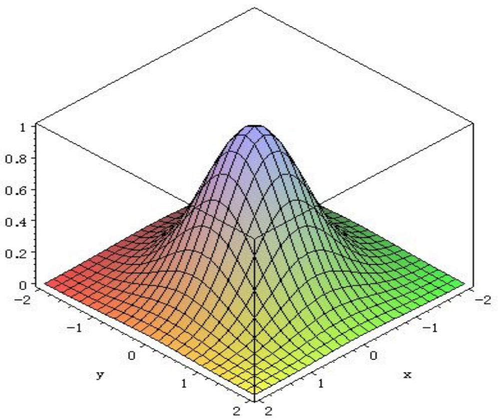
＞ plot(exp(-x^2), x=-2..2) ;



＞ plot3d(x^2 + y^2, x=-3..3, y=-3..3, axes=boxed) ;



> plot3d(exp(-x^2-y^2), x=-2..2, y=-2..2, axes=boxed) ;



それでは、Maple が持っているいくつかの機能の詳細を見ていきましょう。

7　　いろいろな数値

Maple は、電卓のように使うこともできます。つまり、

＞ 1000＋2000；

$$3000$$

といった足し算や掛け算・割り算なども実行できます。掛け算は「*」（アスタリスク記号）を、割り算は「/」（スラッシュ記号）を用いてください。

＞ 1111 ＊ 2－3333/3；

$$1111$$

普通、電卓では階乗の計算を行うことはできませんが、Maple ではコンピュータの資源（メモリや CPU）が許す限り、計算を実行できます。例えば 10! は

＞ 10!；

$$3628800$$

のように計算します。同様に、

＞ 333!；

103344654345880591560939655382975165506222600416820628234329024697831
885979142765685527001948498779298943759502525704770804183527325976587
456659256047046692271337264772438543178366351306941238937116385330019
804962298756654765985688218061703037655404898144022341599015404404321
341558445429624451536463305955882916059244292113522799434713728172799
387209748952603877845782391509318169467864162325166662519654219196518
380446180509912944035469589307454197438369665201987352011232558840892
632728298466405388269798436428857757916415751091787535095800016603920
923967986489243754010241478837022981459100468894028803941953699840000
000000000000000000000000000000000000000000000000000000000000000000000
00000000;

のような大きな数の階乗も計算できます。

また、分数の計算も可能です。

＞ 1/3 +1/5；

$$\frac{8}{15}$$

電卓で 1÷3 を計算すると 0.3333... のような近似値が計算されますが、Maple では 1/3 はあくまで 1/3 という有理数が返されます。

＞ 1/3；

$$\frac{1}{3}$$

1/3 を 3 倍すると、当然結果は 1 になります。

＞ 1/3 ＊3；

$$1$$

しかし、0.3333... という近似値を 3 倍しても厳密に 1 にはなりません。

＞ 0.3333 ＊3；

$$0.9999$$

Maple では、1/3 という有理数を近似値（実数）にする場合に evalf コマンドを用います。(evaluation by float の略)

> evalf(1/3) ;

$$0.3333333333$$

有理数以外にも、根号で表される無理数（代数的数）や数学定数はすべて Maple では「厳密数」と呼ばれ、自動的に近似値として扱われませんので注意しましょう。

数学定数：

> Pi ;

$$\pi$$

> exp(1) ;

$$e$$

無理数：

>sqrt(3) ;

$$\sqrt{3}$$

> ln(5) ;

$$\ln(5)$$

これらの数値の近似値を得るときにも evalf コマンドを用います。
なお、evalf コマンドの 2 番目の引数には近似する精度（桁数）を指定することもできます。

> evalf(Pi, 30) ;

$$3.14159265358979323846264338328$$

以下は、sqrt(2) を 100 桁まで求めています。

> evalf(sqrt(2), 100) ;

$$1.414213562373095048801688724209698078569671875376948073176679737990732478462107038850387534327641573$$

環境変数 Digits を変更しても、evalf による近似値の精度は変更できます。

> Digits := 25 ;

$$Digits := 25$$

> evalf(exp(1)) ;

$$2.718281828459045235360287$$

> Digits := 10 :

8　数式処理

数式処理とは、x や y といった変数を含んだまま行う計算のことを意味します。Maple では、方程式を解くことや式の展開や因数分解、微積分等を行うコマンドが用意されています。

```
> restart;
```

8.1　方程式を解く

solve コマンドは方程式を解くことが出来ます。

```
> eqn1 := x^2 + 4*x + 2;
```

$$eqn1 := x^2 + 4x + 2$$

```
> solve( eqn1 );
```

$$-2 + \sqrt{2}, -2 - \sqrt{2}$$

方程式に複数の変数が含まれている場合は、solve コマンドの 2 つ目の引数に "どの変数について解くか？" を指定します。

ここでは 2 次方程式の解の公式を Maple で求めています。

```
> eqn2 := a*x^2 + b*x + c;
```

$$eqn2 := ax^2 + bx + c$$

```
> solve( eqn2 , x );
```

$$-\frac{b - \sqrt{b^2 - 4ac}}{2a}, -\frac{b + \sqrt{b^2 - 4ac}}{2a}$$

8.2　式の展開と因数分解

式の展開、因数分解を行うには、expand, factor コマンドを使用します。

```
> eqn3 := (x+y)^4;
```

$$eqn3 := (x + y)^4$$

```
> expand( eqn3 );
```

$$x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$$

%と指定しているのは一つ前の結果を指します。つまりここでは、上の展開した結果を引数に指定しています。

```
> factor( % );
```

$$(x + y)^4$$

expand を使うと、各係数部分が、有名なパスカルの三角形と同じであることを実際に（手軽に）確認することができます。

```
> expand( (x+y)^3 );
```

$$x^3 + 3x^2y + 3xy^2 + y^3$$

> expand((x+y)^4) ;

$$x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$$

> expand((x+y)^5) ;

$$x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5$$

> expand((x+y)^6) ;

$$x^6 + 6x^5y + 15x^4y^2 + 20x^3y^3 + 15x^2y^4 + 6xy^5 + y^6$$

8.3 微積分

数式処理の機能は式の展開や因数分解以外にも式を微分・積分したり、極限の計算を行うことが出来ます。

> eqn4 ≔ x^2 + x + 1 ;

$$eqn4 \coloneqq x^2 + x + 1$$

微分を行うには、differential を短縮した diff というコマンドを使用します。

> diff(eqn4 , x) ;

$$2x + 1$$

積分の場合は、integral を省略した int というコマンドになります。

> int(eqn4 , x) ;

$$\frac{1}{3}x^3 + \frac{1}{2}x^2 + x$$

極限の計算は、そのまま limit というコマンドを使用します。

> limit((x^2-1)/(x^3+5) , x=infinity) ;

$$0$$

9　　グラフの作成

Maple では 2 次元グラフや 3 次元グラフを作成することが出来ます。グラフを作成する際には陽関数以外にも媒介変数からグラフを作成することもできます。

restart ;

9.1　　2 次元グラフ

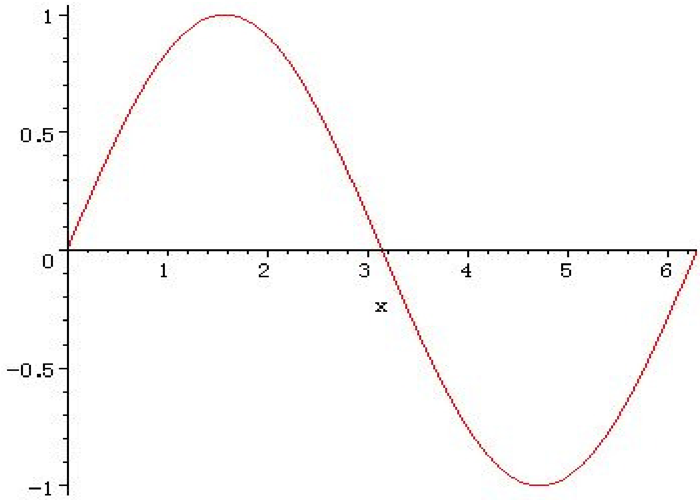
平面のグラフを作成するには plot コマンドを使用します。

コマンドは、plot(式 , 範囲) という入力を使用します。

> eq2d := sin(x) ;

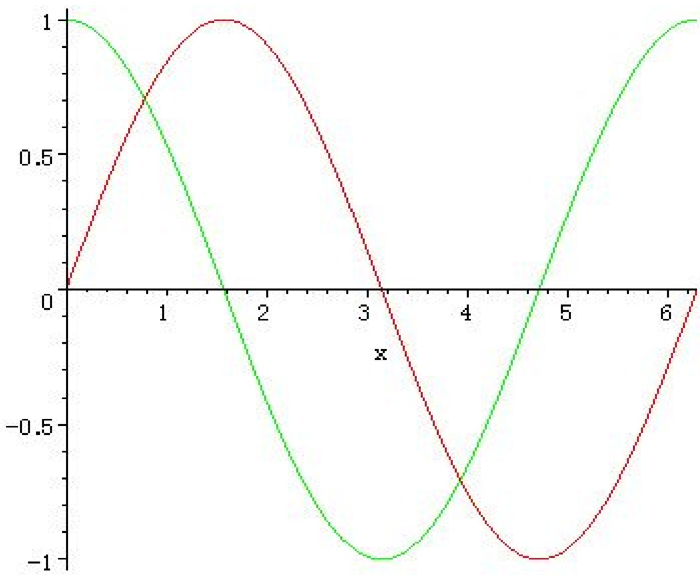
$$eq2d := \sin(x)$$

> plot(eq2d, x=0..2*Pi) ;



plot([式１, 式２], 範囲) というように式をリストで入力すると複数の式の重ね描きを行います。

> plot([sin(x), cos(x)], x=0..2*Pi) ;



9.2　　3 次元グラフ

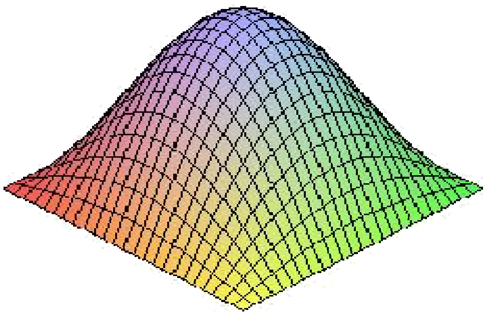
3 次元プロットは plot3d コマンドを使用します。

plot3d コマンドは plot3d(式 , 範囲１ , 範囲２) という形で使用します。

> eq3d := sin(x)*sin(y) ;

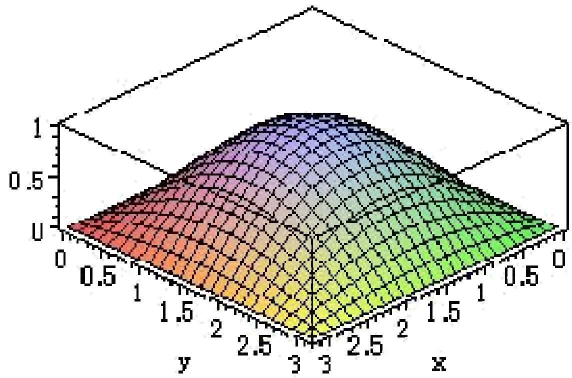
$$eq3d := \sin(x)\sin(y)$$

> plot3d(eq3d, x=0..Pi, y=0..Pi) ;



オプションを指定し軸の表示やスケールのセットを行うことが出来ます。

> plot3d(eq3d, x=0..Pi, y=0..Pi, axes=boxed, scaling=constrained) ;

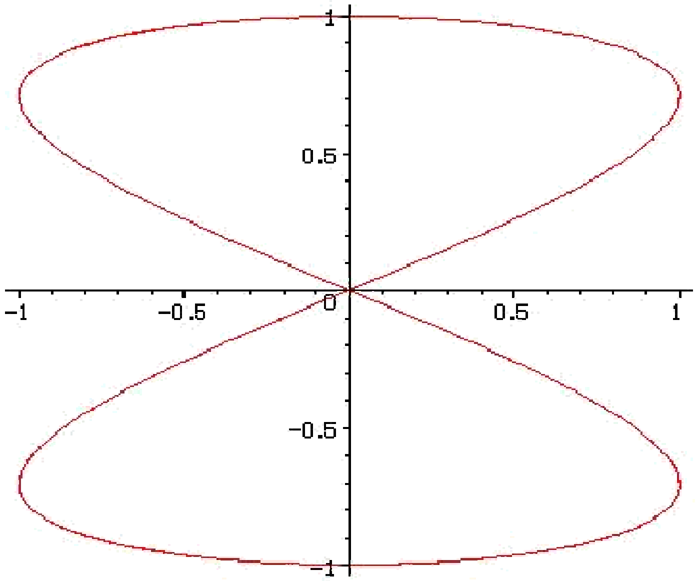


9.3　　媒介変数によるプロット

9.3.1　　2 次元

plot([x 方向の関数 , y 方向の関数 , 範囲]) という形で入力を行うと媒介変数のプロットが可能です。

> plot([sin(4*t), cos(2*t), t=-Pi..Pi]) ;



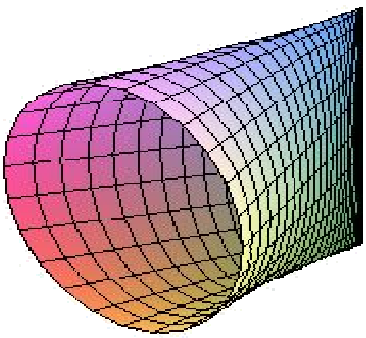
9.3.2　　3 次元

3 次元の場合は、plot3d([３次元の媒介変数] , 範囲 １, 範囲 ２）という形で入力をします。

> param3d := [s^2, s^2*cos(t), sin(t)] ;

$$param3d := [s^2, s^2 \cos(t), \sin(t)]$$

> plot3d(param3d, s=0..2*Pi, t=0..2*Pi) ;



10　プログラミング

ソフトウェアの持っている計算機能やグラフィックス機能を、ひとつの処理としてまとめておくことをプログラミングと言います。Maple では、２つのプログラミング形式を利用して、自身の関数やコマンドを作り出すことができます。

10.1　関数

関数では 「—>」記号で一つの処理を行うことが出来ます。

関数の作成は下記のように入力します；　関数名 ≔ 引数 -> 計算処理

引数が 1 個のシンプルな例は、

```
> f1≔ x -> x^2 ;
```

$$f1 := x -> x^2$$

となり、実行の際は、

```
> f1(2) ;
```

$$4$$

のように実行します。また、Maple は数式処理が可能ですので数値のみではなく数式を引数として入力することも可能です。

```
> f1(sqrt(x)) ;
```

$$x$$

引数が複数になる場合は、丸括弧を使用し複数の引数を指定します。

```
> f2 ≔ (x,y) -> x+y ;
```

$$f2 := (x, y) -> x + y$$

関数を実行し確認します。

```
> f2(1, 2) ;
```

$$3$$

10.2　プロシージャ

プロシージャが関数と違う点は複数行にまたがり処理を記述することができます。

プロシージャの入力では、proc で始まり end proc で終了するという入力が必要になります。

引数 1 つの場合のシンプルな例は、

```
> p1 ≔ proc(x)
> x^2 ;
> end proc ;
```

$$p1 := \mathbf{proc}(x)\ x^2\ \mathbf{end\ proc}$$

となります。（なお、複数行に渡ってコマンドやプログラムを記述する際は、Shift + Enter キーで改行します）

```
> p1(10) ;
```

$$100$$

```
> p1(x^2) ;
```

$$x^4$$

上記から解るように単純なプログラムを作成する場合は、関数の方が向いていることがわかります。

ではプロシージャの場合どのようなものが向いているか下記をご覧下さい。

```
> p2 := proc(x,y)
>    if x > y then
>       x ;
>    else
>       y ;
>    end if ;
> end proc ;
```

$$p2 := \textbf{proc}\ (x, y)\ \textbf{if}\ y < x\ \textbf{then}\ x\ \textbf{else}\ y\ \textbf{end if end proc}$$

このように条件分岐を含む場合や、繰り返し計算を行う場合、複数の手続きが必要な処理を行う場合はプロシージャを使用されることをお勧めします。

```
> p2(10, 4) ;
```

$$10$$

複数回の処理を行うようなときは、for - do 文を利用します。次のプロシージャは 1 から与えられた数　n までを加算するためのプログラムです。

```
> p3 := proc(n)
>    local i, tmp ;
>    tmp  : = 0 ;
>    for i from 1 to n do
>         tmp  : = tmp + i ;
>    end do ;
> end proc ;
```

p3 : = **proc** (n) **local** i, tmp ;　tmp : = 0 ;　**for** i **to** n **do** tmp : = tmp+i **end do end proc**

```
> p3(10) ;
```

$$55$$

```
> p3(100) ;
```

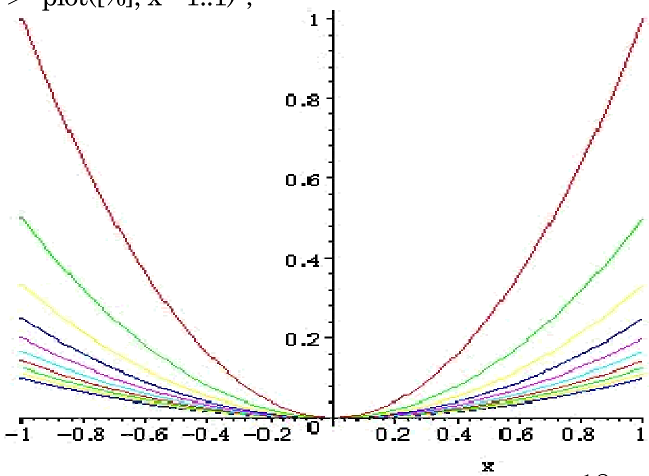
$$5050$$

同じように繰り返しを行うための関数として、seq 関数が用意されています。(sequential の略) seq 関数は、複数の数式やデータを作り出すときに用います。ここで、% 記号は直前の出力結果を指しています。

```
> seq(1/k * x^2, k=1..10) ;
```

$$x^2, \frac{x^2}{2}, \frac{x^2}{3}, \frac{x^2}{4}, \frac{x^2}{5}, \frac{x^2}{6}, \frac{x^2}{7}, \frac{x^2}{8}, \frac{x^2}{9}, \frac{x^2}{10}$$

```
> plot([%], x=-1..1) ;
```



11　　練習問題

restart ;

11.1　問題１

$y = -x^2$ と $y=x^2 + 3x - 9$ の交点の座標を求めなさい。

11.1.1　解答

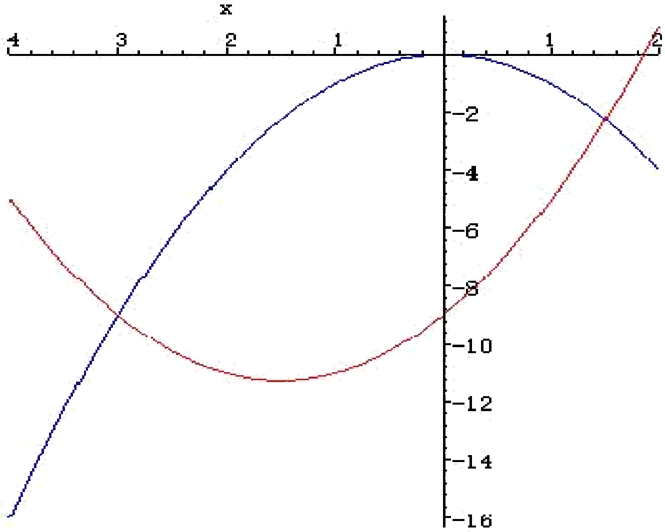
上記の連立方程式を solve コマンドで解きます。

```
> sol := solve( {y=-x^2, y=x^2 + 3*x - 9}, {x,y} ) ;
```

$$sol := \{x = -3, y = -9\}, \{x = \frac{3}{2}, y = \frac{-9}{4}\}$$

グラフで確認してみましょう。

```
> plot({-x^2, x^2+3*x-9}, x=-4..2, color=[red, blue]) ;
```



11.2　　問題２

問題１で得た交点で括られる領域の面積を求めなさい。

11.2.1　解答

変数 sol で得た交点の x 座標を取り出します。

```
> p1 := eval(x, sol[1]) ;
> p2 := eval(x, sol[2]) ;
```

$$p1 := -3$$

$$p2 := \frac{3}{2}$$

この区間では、青色の関数が上部に位置しています。したがって、面積は積分を用いて次のコマンドで計算できます。

```
> int( -x^2 - (x^2 + 3*x - 9), x=p1..p2 ) ;
```

$$\frac{243}{8}$$

なお、この近似値は、次の値となります。

＞ evalf(%) ;

$$30.37500000$$

11.3　問題３

$2ax - a^2 x^2$ の区間 [0, 1] における面積の最大値とそのときの a の値を求めなさい。

11.3.1　解答

まず、面積 S を求めます。

＞ S := int(2*a*x - a^2*x^2, x=0..1) ;

$$S := a - \frac{1}{3}a^2$$

次にこの面積を a の関数として考え、その極値（微分値）が 0 となる値を求めます。

＞ diff(S, a) ;

$$1 - \frac{2a}{3}$$

＞ solve(%=0, a) ;

$$\frac{3}{2}$$

これが面積を最大とするときの a の値であり、そのときの面積は次式で計算できます。

＞ int(subs(a=3/2, 2*a*x-a^2*x^2), x=0..1) ;

$$\frac{3}{4}$$

ここで、subs コマンドは変数に値を代入するためのコマンドです。

Maple チュートリアル【中級】

このチュートリアルでは、初級で学んだ内容をさらに発展させ、Maple をベースにして楕円曲線の解析を行うためのプログラムを作っていきます。Maple の可視化機能（グラフ作成機能）と共にユーザ独自のグラフィックスを作るためのノウハウを学んで行きましょう。

1 楕円曲線をグラフ化しよう

楕円曲線とは、次の式で与えられる関数のことを言います。

> p := y^2 = x^3 + a * x + b ;

$$p := y^2 = x^3 + ax + b$$

では、係数 a, b に適当な値を代入して、グラフを描いてみましょう。代入を行うには、subs コマンドまたは eval コマンドを用います。

例：

> subs({a=-1, b=1}, p) ;

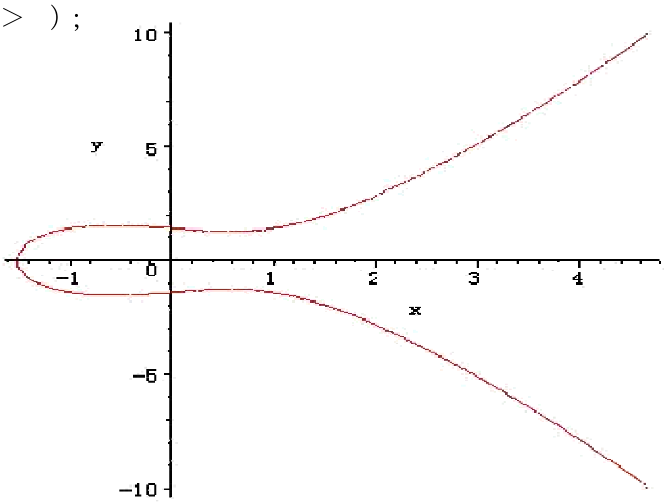
$$y^2 = x^3 - x + 1$$

> eval(p, {a=-1, b=1}) ;

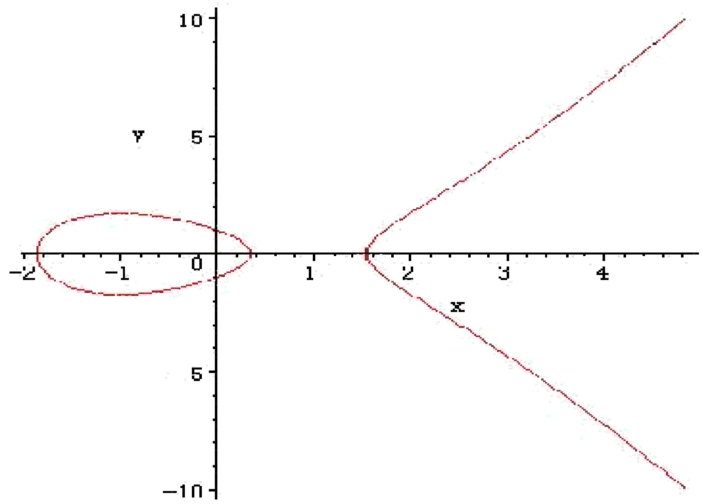
$$y^2 = x^3 - x + 1$$

グラフを描きます。陰関数を描くには、plots パッケージの implicitplot 関数を利用しましょう。

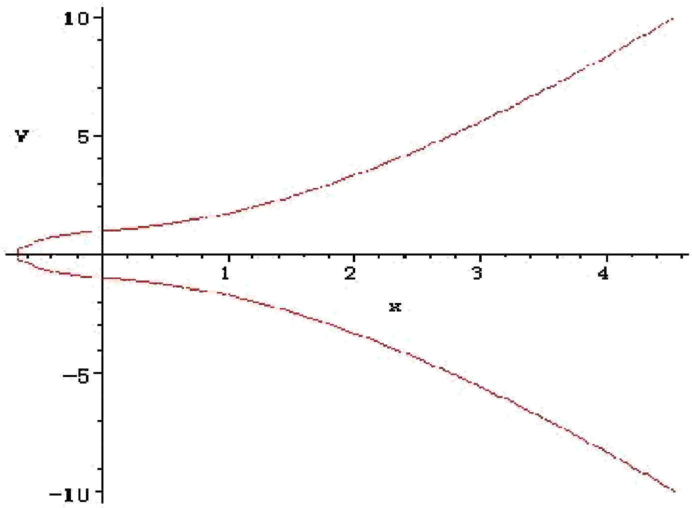
```
> with(plots) :

Warning, the name changecoords has been redefined
> implicitplot(
>   subs({a=-1,b=2}, p),
>    x=-3..6, y=-10..10, grid=[50,50]
>   ) ;
```




```
> implicitplot(
> subs({a=-3,b=1}, p),
> x=-3..6, y=-10..10, grid=[50,50]
> ) ;
```



> implicitplot(
> subs({a=1,b=1}, p),
> x=-3..6, y=-10..10, grid=[50,50]
>) ;



２　接線を描画してみよう

与えられた点 P(x, y)における楕円曲線の接線を求め、グラフで描画してみましょう。ここでは係数 a, b を適当な値に設定した楕円曲線 p1 を考えましょう。

```
> p1 := subs({a=-1, b=1}, p) ;
```

$$p1 := y^2 = x^3 - x + 1$$

ある陰関数 f(x,y)=0 の x=x[0], y=y[0] における接線の傾きは、次の微係数で求めることができます。

```
> -Diff(f(x,y),x)/Diff(f(x,y),y) ;
```

$$-\frac{\frac{\partial}{\partial x} f(x, y)}{\frac{\partial}{\partial y} f(x, y)}$$

p1 の楕円曲線に対して、 x=-1 の点での接線を求めてみます。まず、x=-1 の点における y の値を求めます。

```
> xp := -1 ;
```

$$xp := -1$$

```
> yp := solve(eval(p1, x=xp), y) ;
```

$$yp := 1, -1$$

これらの与えられた点（ここでは、x=-1, y=1, -1）での微係数を求めます。接線の傾きは、上の定義から

```
> p1zero := lhs(p1) - rhs(p1) ;
```

$$p1zero := y^2 - x^3 + x - 1$$

```
> m := - diff(p1zero, x)/diff(p1zero, y) ;
```

$$m := -\frac{-3x^2 + 1}{2y}$$

となるので、x=-1, y=1, -1 をそれぞれ代入して次を得ます。

```
> mval := [eval(m, {x=xp, y=yp[1]}), eval(m, {x=xp, y=yp[2]})] ;
```

$$mval := [1, -1]$$

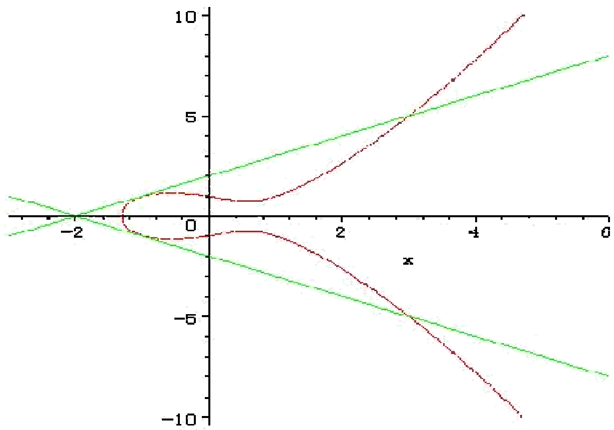
これから x=-1 での接線は以下の 2 種類が考えられます。

```
> tans := [ y - yp[1] = mval[1] * (x - xp), y - yp[2] = mval[2] * (x - xp) ] ;
```

$$tans := [y - 1 = x + 1, y + 1 = -x - 1]$$

元々の楕円曲線と一緒に描画してみます。

```
> gr_elliptic := implicitplot(p1, x=-3..6, y=-10..10, grid=[50,50]) :
> gr_tans := plot(map(p->rhs(isolate(p,y)),tans), x=-3..6, color=green) :
> display(gr_tans, gr_elliptic) ;
```



ここまで行ってきた接線描画の手順を手続き（関数）としてまとめておきます。

```
> # Compute -Diff(p,x)/Diff(p,y) at point v(x,y)
> compTangent := proc(eq)
> local p, m ;
> p := lhs(eq) - rhs(eq) ;
> m := -diff(p, x)/diff(p, y) ;
> return( m ) ;
> end proc :

> compTangent(p1) ;
```

$$-\frac{-3x^2+1}{2y}$$

```
> # Compute y point
> compYpoints := proc(p, xval)
> solve(eval(p, x=xval), y) ;
> end proc :

> # Compute tangent lines for given elliptic curve p(x,y) at x=xval
> Tangents := proc(p, xval)
>   local yp, mval;
>   yp := compYpoints(p, xval) ;
>   mval := seq(eval(compTangent(p), {x=xval, y=yp[i]}), i=1..nops([yp])) ;
>   seq( (y - yp[i] = mval[i] * (x - xval)), i=1..nops([yp])) ;
> end proc :

> p1 ;
```

$$y^2 = x^3 - x + 1$$

```
> Tangents(p1, -1) ;
```

$$y-1 = x+1, y+1 = -x-1$$

なお、ここで得られた各接線の式をプロット関数で使えるようにするには、y= の形へ書き換えて利用します。y= の形にするには isolate コマンドを、さらにその結果の右辺のみを取り出すには rhs コマンドを利用します。また、map は与えられた手続き f をリスト [x1, x2, ..., xn] の各要素に適用する（マップさせる）ためのコマンドです。

```
> map(p->rhs(isolate(p,y)), [Tangents(p1, -1)]) ;
```

$$[x+2, -x-2]$$

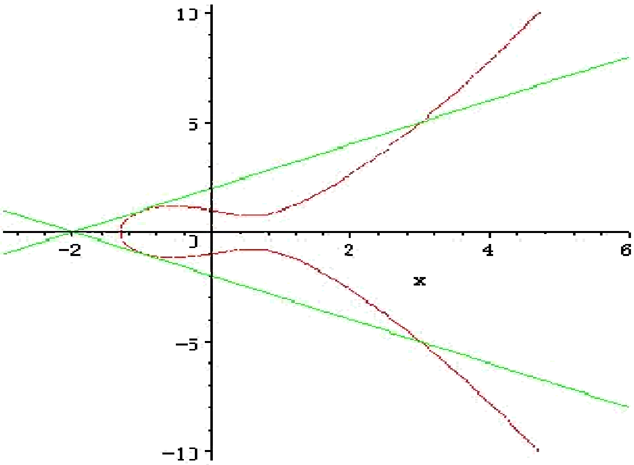
```
> getTangents := proc(p, xval)
>     map(p->rhs(isolate(p,y)), [Tangents(p, xval)]) ;
> end proc :
```

実際に使ってみましょう。

```
> tans := getTangents(p1, -1) ;
```

$$tans := [x+2, -x-2]$$

```
> gr_tans := plot(tans, x=-3..6, color=green) :
> display(gr_tans, gr_elliptic) ;
```



上のグラフを見たときに、多少接点が見えにくいかもしれません。そこで、接点位置を明確にする（接点をプロットする）ための描画関数を用意します。

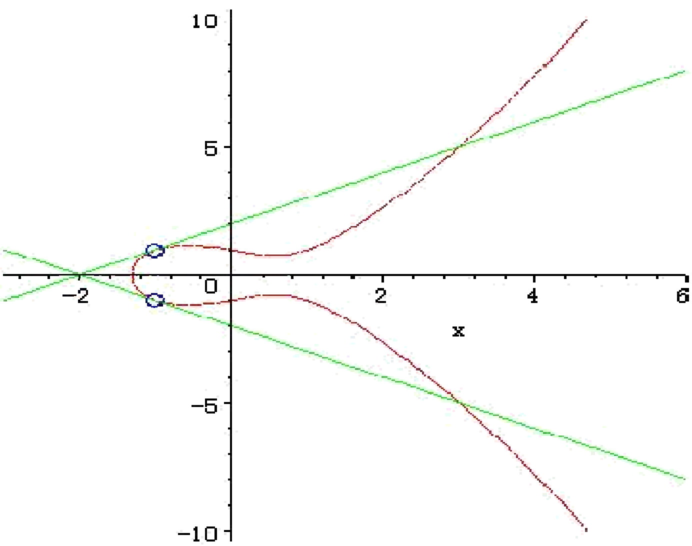
```
> compYpoints(p1, -1) ;
```

$$1, -1$$

```
> getTangentPoints := proc(p, xval)
>   local yp ;
> 
>   yp := compYpoints(p, xval) ;
>   map(y->[xval, y], [yp]) ;
> end proc :
> plotTangentPoints := proc(p, xval)
>   local xylist ;
> 
>   xylist := getTangentPoints(p, xval) ;
>   plots[pointplot](xylist, symbol=circle, symbolsize=18, color=blue) ;
> end proc :
```

この関数を今までのグラフィックスとまとめて描画してみましょう。

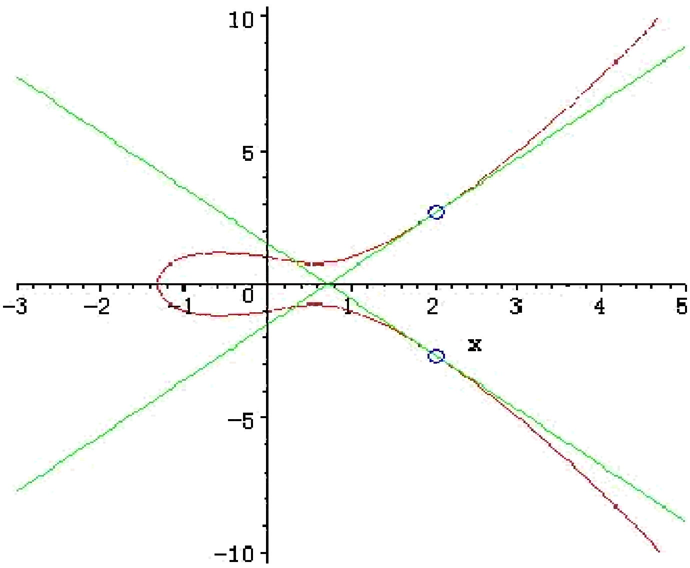
```
> gr_tans_point := plotTangentPoints(p1, -1) :
> display(gr_tans, gr_elliptic, gr_tans_point) ;
```



次の関数は、ここまでの処理をまとめた関数です。引数として楕円曲線を受け取り、与えられた x 座標値の接線の関数を求め、またその点でのグラフを描画します。rangex と rangey は x および y の描画範囲です。

```
> plotTangents := proc(p, xval, rangex, rangey)
> local tans, gr_ellip, gr_tans, gr_tans_point ;
>
> gr_ellip := plots[implicitplot](p, x=rangex, y=rangey, grid=[50,50]) ;
> tans := getTangents(p, xval) ;
> gr_tans := plot(tans, x=rangex, color=green) ;
> print(getTangentPoints(p, xval)) ;
> gr_tans_point := plotTangentPoints(p, xval) ;
> plots[display](gr_tans, gr_ellip, gr_tans_point) ;
> end proc :
> plotTangents(p1, 2, -3..5, -10..10) ;
```

$$[[2, \sqrt{7}], [2, -\sqrt{7}]]$$



３　接線と楕円曲線の交点を求めると・・・

前節で定義した Tangents 関数を使えば、与えられた楕円曲線と適当な点での接線が手軽に計算できるようになりました。では、この関数を利用して得られた接線と元の楕円曲線との交点を求めてみます。

＞ p1 ;

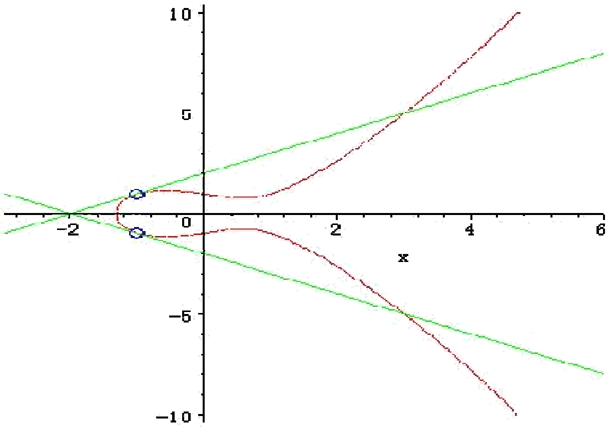
$$y^2 = x^3 - x + 1$$

＞ tans := [Tangents(p1, -1)] ;

$$\text{tans} := [y - 1 = x + 1, y + 1 = -x - 1]$$

＞ plotTangents(p1, -1, -3..6, -10..10) ;

$$[[-1, 1], [-1, -1]]$$



tans の１番目と２番目それぞれの接線の式と楕円曲線との交点を solve 関数で求めます。

＞ eqs1 := {p1, tans[1]} ;
＞ eqs2 := {p1, tans[2]} ;

$$eqs1 := \{y - 1 = x + 1, y^2 = x^3 - x + 1\}$$

$$eqs2 := \{y + 1 = -x - 1, y^2 = x^3 - x + 1\}$$

＞ solve(eqs1, {x,y}) ;

$$\{x = 3, y = 5\}, \{y = 1, x = -1\}, \{y = 1, x = -1\}$$

＞ solve(eqs2, {x,y}) ;

$$\{y = -5, x = 3\}, \{y = -1, x = -1\}, \{y = -1, x = -1\}$$

同様に、別の接線との交点も計算してみましょう。（以下は、x=3 の点における接線とその楕円曲線との交点です）

＞ tans := Tangents(p1, 3) ;

$$tans := \ y - 5 = \frac{13x}{5} - \frac{39}{5}, \ \ y + 5 \ = \ -\frac{13x}{5} + \frac{39}{5}$$

＞ eqs1 := {p1, tans[1]} ;
＞ eqs2 := {p1, tans[2]} ;

$$eqs1 := \{y - 5 = \frac{13x}{5} - \frac{39}{5},\ y^2 = x^3 - x + 1\}$$

$$eqs2 := \{y - 5 = -\frac{13x}{5} + \frac{39}{5},\ y^2 = x^3 - x + 1\}$$

> solve(eqs1, {x,y}) ;

$$\{y = \frac{-103}{125}, x = \frac{19}{25}\}, \{x = 3, y = 5\}, \{x = 3, y = 5\}$$

> solve(eqs2, {x,y}) ;

$$\{y = \frac{103}{125}, x = \frac{19}{25}\}, \{y = -5, x = 3\}, \{y = -5, x = 3\}$$

４　　Maple 上での合同計算・素数計算

Maple では、整数や多項式の合同式も計算することができます。

合同計算は、modp コマンドまたは mods コマンドを用います。次は 111222333 を 332211 で割ったときの余りです。

＞ 111222333 mod 332211；

$$263859$$

関数の形式で記述することもできます。

＞ r := modp(111222333, 332211)；

$$r := 263859$$

iquo は商を求めます。

＞ iquo(111222333, 332211)；

$$334$$

＞ % * 332211 + r；

$$111222333$$

modp や mods は、多項式の演算と組み合わせることもできます。

＞ f := x^3 + 5；

$$f := x^3 + 5$$

整数上ではこの多項式は因数分解できません。

＞ factor(f)；

$$x^3 + 5$$

しかし、11 を法とした場合、次のような因数分解が可能です。

＞ mods(Factor(f), 11)；

$$(x+3)(x^2-3x-2)$$

しかし、まだ２次因子が含まれています。そこで、$x^2 - 3x - 2$ の根を代数的数として変数 α に割り当てます。

＞ alias(alpha = RootOf(y^2 - 3*y - 2))；

$$\alpha$$

このとき、1/α（α の逆数は、11 を法とした場合に次の代数的数として表現できます。

＞ mods(Normal(1/alpha), 11)；

$$-5\alpha+4$$

つまり、代数的数 α を追加した有理数体 Q(α) では、多項式を f を次のように因数分解することができるようになります。

＞ mods(Factor(f, alpha), 11)；

$$(x-\alpha)(x+3)(x+\alpha-3)$$

最終的に一次因子に分解できました。さて、この多項式は、11 を法とした世界で展開すると、元の多項式と一致します。

> mods(Expand(%), 11) ;

$$x^3+5$$

5　練習問題

次の練習問題を行ってみてください。

5.1　問題１

次の楕円曲線 P を考えます；

＞ P := (y^2 = x^3 - 2*x + 2) ;

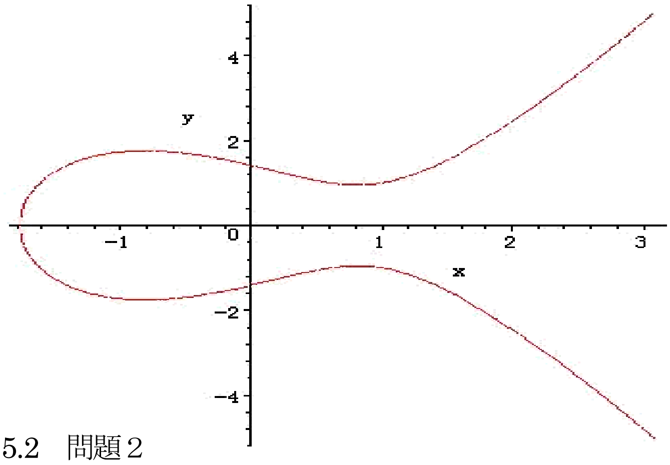
$$P := y^2 = x^3 - 2x + 2$$

この楕円曲線 P のグラフを作成しなさい。

5.1.1　解答

plots パッケージの implicitplot 関数（陰関数の描画関数）を用います。

＞ plots[implicitplot](P, x=-4..4, y=-5..5, grid=[50,50]) ;



5.2　問題２

問題１で定義した楕円曲線 P に対して、

＞ xp := -1 ;

$$xp := -1$$

のときの接線の方程式を、このワークシートで定義した Tangents 関数で求めなさい。また、そのグラフを描画しなさい。

5.2.1　解答

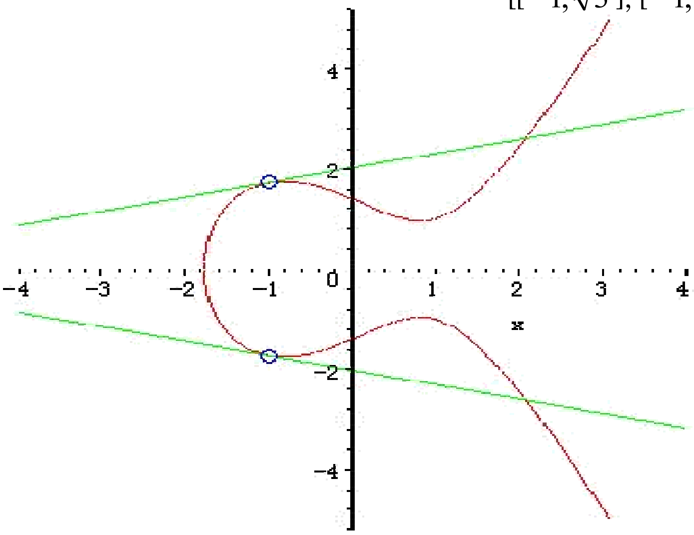
次のように実行します。

＞ tans := Tangents(P, xp) ;

$$tans := \ y - \sqrt{3} \ = \ \frac{\sqrt{3}(x+1)}{6}, y + \sqrt{3} \ = \ -\frac{\sqrt{3}(x+1)}{6}$$

> plotTangents(P, xp, -4..4, -5..5) ;

$$[[-1,\sqrt{3}\,],[-1,-\sqrt{3}\,]]$$



5.3　問題３

問題２で得た接線の方程式と、元々の楕円曲線 P との交点を求めなさい。

5.3.1　解答

すでに変数 tans に接線の方程式が含まれているので、それぞれの接線と元の楕円曲線を連立方程式にして solve コマンドで求めます。

> eqs1 := {P, tans[1]} ;
> eqs2 := {P, tans[2]} ;

$$eqs1 := \{ y^2 = x^3 - 2x + 2,\ y - \sqrt{3} = \frac{\sqrt{3}(x+1)}{6}$$

$$eqs2 := \{ y^2 = x^3 - 2x + 2,\ y + \sqrt{3} = -\frac{\sqrt{3}(x+1)}{6}$$

> sol1 := solve(eqs1, {x,y}) ;

$$sol1 := \{x = \frac{25}{12},\ y = \frac{109\sqrt{3}}{72}\}, \{x = -1, y = \sqrt{3}\,\}, \{x = -1, y = \sqrt{3}\,\}$$

> sol2 := solve(eqs2, {x,y}) ;

$$sol2 := \{x = \frac{25}{12},\ y = -\frac{109\sqrt{3}}{72}\}, \{y = -\sqrt{3}, x = -1\}, \{y = -\sqrt{3}, x = -1\}$$