Maple，Mathematica 入門

雪江明彦

ここでは，数式処理ソフトの使いかたの基本について述べる．解説するのは，商用ソフトの Maple，Mathematica とである (バージョンはそれぞれ 13, 7)．Maple は『classical maple worksheet』を使うものとする．**バージョンによって操作が違うかもしれないので，マニュアルで確認をすることが大切である．**

注意しておくが，数式処理ソフトは計算はしてくれるが，概念を理解する助けにはならない．また，ある程度手計算もできないと，数式処理ソフトも使いこなせないようである．しかし，3 次元グラフなど，興味深いことも可能なので，ごく基本的な機能に限って解説することにする．また，線形代数についても少しだけ解説する．

基本操作

終了するには: 終了するには Maple なら `quit;`，Mathematica なら `Quit` とすればよい．

Maple の起動と初期設定: Maple を起動するには，京大のメディアセンターの Windows なら，メニューから

専門ソフト → Maple → Maple15

とする．Linux なら，コマンドラインから `xmaple` とする．その後，

ツール → オプション → 表示 (Display)

と選択し，一番上の「入力 (Input display)」を「テキスト (Maple Notation)」にする．これで「すべてのセッションに適用」を選択し，一旦 Maple を閉じる．その後再び Maple を起動する．こうしないと，入力するときに，変なふうに補完したりして入力しずらい．

ヘルプ:コマンドについて知りたい場合，Maple なら `?sin` などとする．Mathematica なら `? Integrate` などとする．

コマンド: Maple の場合，すべてのコマンドはセミコロンかコロンで終わらなくてはならない．定義，計算はプロンプト `>` で入力する．**式を入力した後 Enter を押す．** Mathematica の場合，プロンプト `In[1]:=` で**式を入力した後 Shift+Enter を押す．**

セミコロンやコロンは必要ではない．これらを同時に扱うため，以降プロンプトの部分は書かない．

両方とも，基本演算である加減乗除とべきは `+,-,*,/,^` である．ただし，Mathematica の場合，`2Pi` のように，組込み定数との積には `*` は必要ない．　例えば，

`1 + 2;` (Enter)	Maple の場合
`1 + 2` (Shift+Enter)	Mathematica の場合

と入力すると，3 という答えが返って来る．

絶対値は Maple では `abs(x)`，Mathematica では `Abs[x]` である．

1 つ前の出力:どちらのソフトでも 1 つ前の出力は `%` である．だから，`%+1` は 1 つ前のコマンドの結果に 1 を足したものである．ただし，1 つ前の出力がない場合には使えない．

優先順位:どちらのソフトでも，基本演算の中ではべき `^` が最優先される．`*`，`/` が次で，`+`，`-` の優先順位が最も低い．例えば，

`23^4*2 + 4/2;` (Mathematica なら ; なしで Shift+Enter)

は $(23)^4 \times 2 + \frac{4}{2}$ という意味である．演算の順番を強制的に指定するときは ( ) を使う．

定義と関数:定義は `:=` で与える．だから，

`eq:= x+y;`	Maple の場合
`eq:= x+y`	Mathematica の場合

は `eq` を形式的な表現 `x+y` と定義する．しかしこれは関数ではないので，`x=1,y=2` という値を `eq` に `eq(1,2)` というように代入することはできない．もしこの表現に値を代入するなら，

`subs(x=1,eq);  subs(x=1,y=2,eq);`	Maple の場合
`eq /. x->1  eq /. {x->1,y->2}`	Mathematica の場合

などとする．関数を定義するには

`eq:= x-> x^2;  eq:= (x,y) -> x+y;`	Maple の場合
`eq[x_,y_]:= x+y`	Mathematica の場合

などとする．このように，関数として定義すると，

`eq(1,2);`	Maple の場合
`eq[1,2]`	Mathematica の場合

というように値を代入することができる．

Maple では `f:= x-> x^2` と入力した後，`f:= x-> x^3` などと入力すると，関数の定義が変わる．しかし Mathematica では `Clear[f]` などとした後でないと，定義の

変更はできない.

パッケージの読み込み:どのソフトでも，最初からすべてのコマンドが使えるわけではない．必要なパッケージはその都度読み込むが，それは以下のようにして行う.

`with(linalg);`	Maple の場合
`<< Calculus‘VectorAnalysis‘`	Mathematica の場合

Mathematica の場合，これは Calculus というパッケージの中の VectorAnalysis の部分を読み込む.

微分積分に関する Maple の基本コマンド

対象・演算	コマンド (の例)
加減乗除・べき・絶対値	`+,-,*,/,^,abs(x)`
近似値	`evalf( )  evalf( ,桁数)`
円周率	`Pi`
虚数単位	`I`
無限大	`infinity`
二項係数	`binomial(n,r)`
関数の定義 (1 変数)	`f:= x -> x^2+1`
関数の定義 (多変数)	`f:= (x,y) -> x^2*y+1`
三角関数	`sin(x)  cos(x)  tan(x)`
逆三角関数	`arcsin(x)` など
指数関数	`exp(x)  2^x`
対数関数	`log(x)  log[10](x)`
極限	`value(Limit(f(x),x=点))`
左極限	`value(Limit(f(x),x=点,left))`
解を求める (明示的)	`solve({x+y=0,x^2+y^2=2},{x,y})`
解を求める (近似値)	`fsolve({x+sin(y)=0,x-y=1},{x,y})`
微分 (1 変数)	`diff(f(x),x)`
高階偏微分	`diff(f(x,y),x$回数,y$回数)`
不定積分	`int(f(x),x)`
定積分 (明示的)	`int(f(x),x=0..1)`
定積分 (近似値)	`evalf(Int(f(x),x=0..1))`
部分分数展開	`convert(f(x),parfrac,x)`
テイラー展開	`series(f(x),x=点, 次数)`
ヤコビ行列	`jacobian([x^2,x*y],[x,y])`

グラフに関する Maple の基本コマンド

対象・演算	コマンド (の例)
2 次元グラフ	`plot(x^3+1,x=-5..5)`
3 次元グラフ	`plot3d(x^2-y^2,x=0..5,y=0..5)`
パラメータ表示 (1 変数)	`plot([x,x^2,x=0..5])`
パラメータ表示 (2 変数)	`plot3d([x*y,x^2,x+y],x=0..5,y=0..5)`
極座標のグラフ	`polarplot(f(t),t=0..2*Pi)`
平面の点	`pointplot([[1,2],[3,1]])`
空間の点	`pointplot3d([[1,2,3],[2,3,4]],symbol=circle)`
2 次元陰関数	`implicitplot(x^2+y^3=1,x=-2..2,y=-2..2)`
3 次元陰関数	`implicitplot3d(x^2+y^2+z^3=1,(x,y,z の範囲))`
等高線 (平面で)	`contourplot(x*y,x=-5..5,y=-5..5)`
等高線 (立体的に)	`contourplot3d(x*y,x=-5..5,y=-5..5)`

線形代数に関する Maple の基本コマンド

対象・演算	コマンド (の例)
行列	`matrix(2,2,[1,2,3,4])`
行列の和・差・べき	`evalm(A+B),evalm(A-B),evalm(A^(-3))`
行列のスカラー倍・積	`evalm(c * A)`, `evalm(A &* B)`
転置行列	`transpose(A)`
行列の標準形	`rref(A)`
行列式	`det(A)`
ベクトル	`vector([1,2,3])`
内積, 外積 (ただし 3 次元のみ)	`dotprod(v,w)`, `crossprod(v,w)`
ベクトルの長さ	`norm(v,2)`
固有値・固有ベクトル	`eigenvects(A)`
ジョルダン標準形	`jordan(A,'P'); evalm(P);`

微分積分に関する Mathematica の基本コマンド

対象・演算	コマンド (の例)
加減乗除・べき・絶対値	`+,-,*,/,^,Abs[x]`
近似値	`N[ ]  N[ , 桁数]`
円周率	`Pi`
虚数単位	`I`
無限大	`Infinity`
二項係数	`Binomial[n,r]`
関数の定義 (1 変数)	`f[x_]:= x^2+1`
関数の定義 (多変数)	`f[x_,y_]:= x^2*y+1`
三角関数	`Sin[x]  Cos[x]  Tan[x]`
逆三角関数	`ArcSin[x]` など
指数関数	`Exp[x]  2^x`
対数関数	`Log[x]  Log[10,x]`
極限	`Limit[f[x],x->0]`
左極限	`Limit[f[x],x->0,Direction->1]` (右極限は `-1`)
解を求める (明示的)	`Solve[{x^2-2==0,y^2==3},{x,y}]`
解を求める (近似値)	微分積分に必要なコマンド (Mathematica) 参照
微分 (1 変数)	`D[f[x],x]`
高階偏微分	`D[f[x,y],{x, 回数},{y, 回数}]`
不定積分	`Integrate[f[x],x]`
定積分 (明示的)	`Integrate[f[x],{x,0,1}]`
定積分 (近似値)	`NIntegrate[f[x],{x,0,1}]`
部分分数展開	`Apart[f[x]]`
テイラー展開	`Series[f[x],{x, 点, 次数}]`
ヤコビ行列	微分積分に必要なコマンド (Mathematica) 参照

グラフに関する Mathematica の基本コマンド

グラフの種類	コマンド (の例)
2 次元グラフ	`Plot[Sin[x],{x,-Pi,Pi}`
3 次元グラフ	`Plot3D[x*y,{x,0,1},{y,0,1}]`
パラメータ表示 (1 変数)	`ParametricPlot[{x,x^2},{x,0,1}]`
パラメータ表示 (2 変数)	`ParametricPlot3D[{x,y,x*y},{x,0,5},{y,0,5}]`
極座標のグラフ	`PolarPlot[1+Cos[x],{x,0,2Pi}]`
2 次元陰関数	`ContourPlot[x^3-x*y+y^3==2,{x,-5,5},*]`
3 次元陰関数	`ContourPlot3D[関数,{x,0,1},{y,0,1},{z,0,1},*]`
平面上の点	`Graphics[{PointSize[0.05],Point[{1,2}]}]`
空間の点	`Graphics3D[{PointSize[0.05],Point[{1,2,3}]}]`
等高線 (1 変数)	2 次元陰関数と同じ．* に `Contours->{1,2,3}` など
等高線 (2 変数)	3 次元陰関数と同じ．* に `Contours->{1,2,3}` など

線形代数に関する Mathematica の基本コマンド

対象・演算	コマンド (の例)
行列	`{{1,2},{3,4}}`
行列の和・差・べき	`A+B`, `A-B`, `MatrixPower[A,n]`
行列のスカラー倍・積	`c A`, `A . B`
転置行列	`Transpose[A]`
行列の標準形	`RowReduce[A]`
行列式	`Det[A]`
ベクトル	`{1,2,3}`
内積, 外積 (ただし 3 次元のみ)	`v . w`, `Cross[v,w]`
ベクトルの長さ	`Norm[v]`
固有値・固有ベクトル	`Eigenvalues[A]`, `Eigenvectors[A]`
ジョルダン標準形	`JordanDecomposition[A]`

微分積分に必要なコマンド (Maple)

Maple での微分積分に必要なコマンドについて解説する．表のコマンドに関して，解説が必要なものについてのみ述べる．一般に，`simplify()` というコマンドは式を簡単にしてくれる．展開するときは `expand()` である．三角関数などの値の，実数としての近似値を求めるときには，`evalf(cos(1))` などと `evalf` を使う．桁数を指定するなら，`evalf(cos(1),20)` などとする．方程式を解くときには，`solve(x^2-2*x+2=0,x)` (明示的) や `fsolve(x^3-x-1=0,x)` (近似値)，`fsolve(x^3-x-1=0,x=0..2)` (範囲を指定) などとする．複数の方程式，あるいは変数なら，`{x+y=0,x-y=2}` などと `{ }` が必要である．条件，あるいは変数が 1 つだけなら必要ない．小数点の桁数をずっと変更するなら，`Digits:= 100;` などとする．

`fsolve()` を使ってある範囲での解を求めるときには，`{x=0..2,y=-5..5}` などとする．なお Maple では，近似解は必ずしもすべて求めてくれるとは限らないので，1 つ解がみつかったら，その解が入らない範囲を指定して別の解を探すなど，注意して使う必要がある．この機能に関しては，Mathematica が使い勝手がよい．ただし，Mathematica は代数方程式でない場合には，工夫が必要である．

積分はいつも明示的にできるとは限らない．また，明示的に求めることができる場合でも，ソフトに組み込まれていなくてできない場合もある．例えば，$\int \sqrt[3]{\frac{x+1}{x-1}}\,dx$ **は積分できるべきものだが，Maple 13, Mathematica 7 ではまだできない．**定積分の近似値を求めるときには，`evalf(Int(f(x),x=0..1))` などとする．重積分は `int(int(x^2+y^2,y=0..x^2),x=0..1);` などとする．近似値を求める場合も同様である．

写像のヤコビ行列を求めるには `jacobian()` というコマンドを使うが，それには `with(linalg);` として，線形代数のパッケージを読み込む必要がある．`jacobian()` は基本的にベクトル解析のコマンドなので，関数の数が 1 つでも `[x]` などとして，ベクトルとして扱わなければならない．

グラフィックスに関するコマンドは，多くの場合 `with(plots);` としてから使う．複数の対象を同時に表示するには以下のようにする．

```
with(plots);
p1:= pointplot([1,2],[3,-1]);
p2:= plot(x^3+1,x=0..5);
display(p1,p2);
```

関数を微分した後で値を代入するときには，注意が必要である．例えば，

```
f:= x-> diff(x^2,x);   f(1);
```

とすると，x^2 の微分 $2x$ に $x=1$ を代入したものになりそうなものだが，実際にはエラーになる．それは，コンピューターが $f(x)$ という式が確定した後に $x=1$ を代入しているのではないからである．このような，関数とそうでない式の違いからくる難しさというのは，Maple だけでなく Mathematica でも同様である．上の例では

```
f:= diff(x^2,x);   subs(x=1,f);
```

とするか，`f:x-> diff(x^2,x); subs(x=1,f(x));` というように，$f(x)$ を式として確定させた後，`subs( )` コマンドを使えばよい．

微分積分に必要なコマンド (Mathematica)

Mathematica での微分積分に必要なコマンドについて解説する．一般に，`Simplify[ ]` というコマンドは式を簡単にしてくれる．展開するときは `Expand[ ]` である．近似値には `N[ ]` を使う．`N[Cos[1],50]` などすれば桁数を 50 桁に指定できる．右極限は `Direction->-1` となる．方程式を明示的に解くときには，`Solve[x^2-2==0,x]` とする．方程式の近似解を求めるには，代数方程式なら `NSolve[x^2-2==0,x]` などとする．この場合はたいがいすべての近似解を求めてくれるので，Maple より使い勝手がよい．代数方程式でないなら，`FindRoot[x-Sin[x]-1==0,{x,0,2}]` などとする．ここで $0, 2$ はニュートンの近似法を使う区間 $[0, 2]$ を表し，これら 2 点での値の符号が違う場合のみ，このコマンドを適用できる．このように，代数方程式でない場合は，Maple のほうが使い勝手がよい．ヤコビ行列に関しては　`Calculus‘VectorAnalysis‘` というパッケージに `JacobianMatrix` というコマンドがあるが，変数の数も 3 に限られていて，また使い勝手もよくないので，ヤコビ行列の成分を `D[ ]` を使って入力したほうがよいだろう．この機能に関しては Maple が使い勝手がよい．

関数のグラフを表示するには，表のようにすればよい．表のコマンドのうち，パッケージを読み込まなければならないものもある．(‘ は @ を shift したキーである．)

`Graphics‘Graphics‘`	`PolarPlot` などグラフィックスの基本
`Graphics‘ImplicitPlot‘`	2 次元陰関数 (`ContourPlot` でも代用可)
`Graphics‘ParametricPlot3D‘`	`ParametricPlot3D`, `SphericalPlot3D` など
`Graphics‘ContourPlot3D‘`	等高面．3 次元陰関数もこれを使う
`Graphics‘Graphics3D‘`	空間の点や線などの 3 次元の対象

Mathematica でも以下のようにすれば，複数の対象を同時に表示できる．

```
g1 = Graphics[{PointSize[0.05],Point[{1,2}],Point[{2,3}]}]
g2 = Plot[x^2,{x,0,2}]
Show[g1,g2]   (g1 で指定された点はここで初めて表示される．)
```

Mathematica でも Maple と同様，微分した後値を代入するときには，

`f:= D[x^2,x]` (Shift+Enter) `f /. x-> 1`

とするか，`f[x_]:= D[x^2,x]` (Shift+Enter) `f[x] /. x-> 1` というように，$f(x)$ を式として確定させた後，$x = 1$ を代入すればよい．

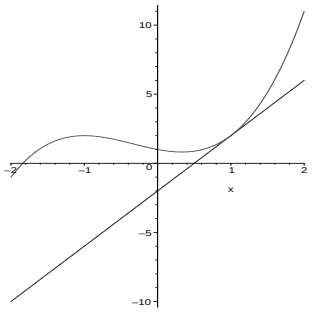
線形代数に必要なコマンド (Maple, Mathematica)

Maple では行列の演算を実際に実行するには，evalm( ) とする必要がある．例えば，A,B が行列なら，A+B; は単に A+B という表現しか返ってこない．逆行列を求めるには，inverse(A) でも A^(-1) でもよい．eigenvects(A) は便利なコマンドだが，A が整数を成分とする行列の場合は，固有値を代数的に求めようとする．だから，近似値を求めようとする場合には，成分の 1 つを 1.0 などと入力して，Maple に実数行列であることを認識させた上でこのコマンドを使う必要がある．これは Mathematica でも同様である．

Mathematica で行列表示するには，入力するとき //MatrixForm と後ろに加えればよい．逆行列は Inverse[A] である．

コマンドの例

1. 右のように，関数 $y = x^3+x^2-x+1$ のグラフと，$x = 1$ での接線 $y = 4x-2$ を同じ図に描くコマンドは以下のようになる．



plot({x^3+x^2-x+1,4*x-2},x=-2..2,*);	Maple の場合
Plot[{x^3+x^2-x+1,4*x-2},{x,-2,2},*]	Mathematica の場合

線の太さなど，* の部分にオプションもつけることができる．上のように複数の対象を表示する場合，Maple, Mathematica なら，各々でオプションを使って，display, show コマンドで後で同時に表示するのが一番わかりやすい．Maple なら，

```
thickness=2,resolution=300,color=blue
```

などとオプションを設定できる．あるいは，

```
macro(color1=RGB(1,1,0.5));
```

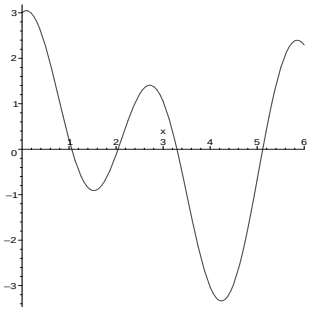
としておいて，color=color1 などとしてもよい．ここで，RGB の数字は光の三原色の赤，緑，青の割合である．

Mathematica なら，

```
PlotPoints->50,PlotStyle->{Thickness[0.01],RGBColor[1,0,0]}
```

などとオプションを設定できる．

2. $y = \sin(x) + \cos(x) + 2\cos(2.2x)$ のグラフを描き，極値の近似値を求めるコマンドは以下のようになる．

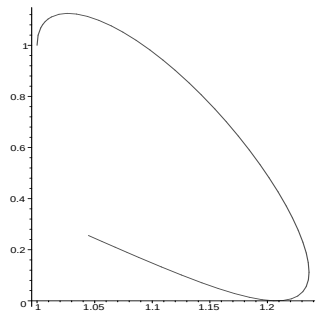


左が Maple 用，右が Mathematica 用である．

```
f:= sin(x)+cos(x)+2*cos(2.2*x);
plot(f,x=0..6);
g:= diff(f,x);
a1:= fsolve(g=0,x=0..1);
evalf(subs(x=%,f));
(略)
a5:= fsolve(g=0,x=5..6);
evalf(subs(x=%,f));
```

```
f:= Sin[x]+Cos[x]+2*Cos[2.2*x]
Plot[f,{x,0,6}]
g:= D[f,x]
FindRoot[g==0,{x,0,1}]
f /. %
(略)
FindRoot[g==0,{x,5,6}]
f /. %
```

3. パラメータ化された曲線 $x(t) = \sin(t)+\exp(-t), y(t) = \cos(2t)+\sin(t)$ $(t \in [0, 2])$ のグラフは以下のようになる．以下はこの曲線の長さの近似値を求めるコマンドである．このような曲線の長さを初等関数を使って表すことができないことは明らかだろう．



左が Maple 用，右が Mathematica 用である．

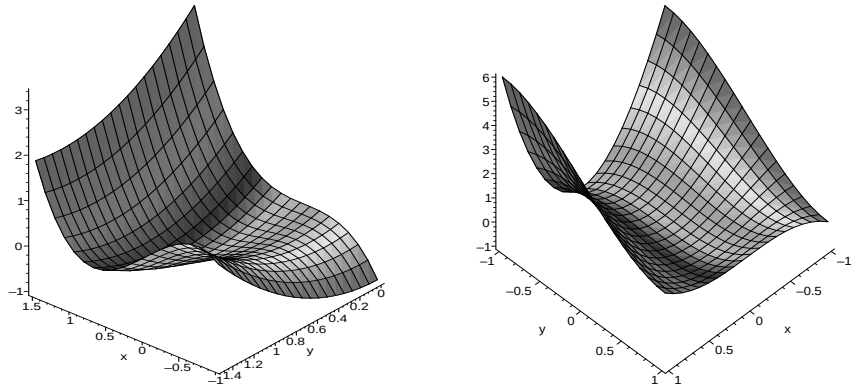
```
x:= sin(t)+exp(-t);
y:= cos(2*t)+sin(t);
F:= diff(x,t)^2+diff(y,t)^2
evalf(Int(F^(1/2),t=0..2));
```

```
x:= Sin[t]+Exp[-t]
y:= Cos[2*t]+Sin[t]
F:= D[x,t]^2+D[y,t]^2
NIntegrate[F^(1/2),{t,0,2}]
```

4. 以下，`with(plots);`(Maple)，`<< Graphics‘ContourPlot3D‘`(Mathematica) などとして，必要なパッケージは読み込むものとする．ただし，Mathematica 7 では

ContourPlot3D は読み込む必要はないようである.

関数 $z = x^3 - x * y + y^2 - y$, $z = x^3 - 3 * x^2 * y + y^3 + 3 * x^2$ のグラフは以下のようになる.



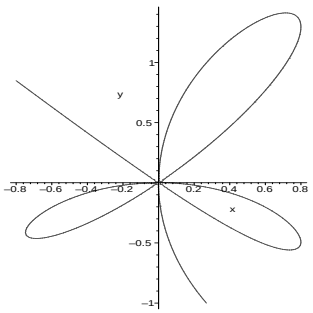
これを実現するコマンドは以下のようになる. Maple, Mathematica の順.

```
plot3d(x^3-x*y+y^2-y,x=-1..1.5,y=0..1.5,*);
plot3d(x^3-3*x^2*y+y^3+3*x^2,x=-1..1,y=-1..1,*);
* は axes=frame,grid=[20,20],color=x などとして表示を調整.
```

```
Plot3D[x^3-x*y+y^2-y,{x,-1,1.5},{y,0,1.5},*]
Plot3D[x^3-3*x^2*y+y^3+3*x^2,{x,-1,1},{y,0,1},*]
* は PlotPoints->50,ColorFunction->Hue などとして表示を調整.
```

画像として保存するには, Maple では画像を右クリックして export で eps 画像を選択すればよい. Mathematica では `Export["C:\Home\gazou.eps",%]` などとすればよい.

5. 2 次元陰関数 $y^5 - 4xy(y^2 - x^2) + 2x^5 = 0$ のグラフは右のようになる. コマンドは以下のようになる.



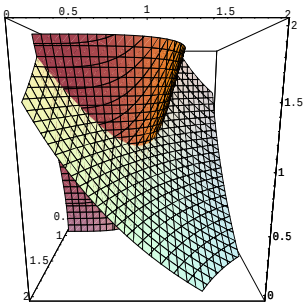
以下, Maple, Mathematica の順.

```
implicitplot(y^5-4*x*y*(y^2-x^2)+2*x^5=0,x=-0.8..1,y=-1..2,*);
```

```
ContourPlot[y^5-4*x*y*(y^2-x^2)+2*x^5==0,{x,-0.8,1},{y,-1,2},*]
```

* では `grid=[100,100]`, `PlotPoints->{20,20}` などとする.

6. 2つの3次元陰関数 $x^3+y^4-z^2=1$, $xy+xz+yz=3$ のグラフを同時に表示すると右のようになる.



コマンドは以下のようになる.
Maple の場合:

```
p1:=implicitplot3d(x^3+y^4-z^2=1,x=0..2,y=0..2,z=0..2,*):
p2:=implicitplot3d(x*y+x*z+y*z=3,x=0..2,y=0..2,z=0..2,*);
display(p1,p2);
```

* では color=red, color=blue などとする.

Mathematica の場合:

```
p1=ContourPlot3D[x^3+y^4-z^2-1,{x,0,2},{y,0,2},{z,0,2}]
p2=ContourPlot3D[x*y+x*z+y*z-3,{x,0,2},{y,0,2},{z,0,2}]
Show[p1,p2]
```

ただし, オプションとして, Contours->{0.0}, PlotPoints-{20,20,20} などとする. Mathematica は複数の曲面の表示の場合, 色を変えてくれるようである.

線形代数のコマンドの例も少しだけ考える.

Maple の場合:

```
with(linalg);
A:= matrix(3,5,[1,2,3,4,5,2,5,3,4,1,0,-2,4,3,7]);
rref(A);
```

とすると, 行列

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 3 & 4 & 1 \\ 0 & -2 & 4 & 3 & 7 \end{pmatrix}$$

の標準形が求まる.

```
with(linalg);
A:= matrix(3,3,[1,2,3,4,5,6,7,8,9]);
B:= diag(2,5,3);
evalm(A^2 &* transpose(B));
det(A);
```

とすると,

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} 2 & & \\ & 5 & \\ & & 3 \end{pmatrix}$$

とし，$A^{2t}B, \det A$ を計算する．A の最初の成分を 1 の代わりに 1.0 と入力して，`eigenvects(A)` とすると，A の固有値と固有ベクトルの近似値が全て求まる．1.0 とするのは，Maple に行列が実数行列であることを教えるためである．そうしないと，3 次方程式を解の公式を使って求めようとし，普通望ましい結果にはならない．ジョルダン標準形を求めるには,

```
with(linalg);
A:= matrix(3,3,[1,2,3,4,5,6,7,8,9]);
jordan(A,'P');
evalm(P);
```

とすれば，A のジョルダン標準形 J と，$A = PJP^{-1}$ となる P が求まる.

Mathematica の場合:
　上と同じことを実現するためには以下のようになる.

```
A= {{1,2,3,4,5},{2,5,3,4,1},{0,-2,4,3,7}}
RowReduce[A]//MatrixForm

A= {{1,2,3},{4,5,6},{7,8,9}}
B= DiagonalMatrix[{2,5,3}]
A^2 . Transpose[B]//MatrixForm
Det[A]

A= {{1.0,2,3},{4,5,6},{7,8,9}}
```

```
Eigenvalues[A]
Eigenvectors[A]

A= {{1,2,3},{4,5,6},{7,8,9}}
JordanDecomposition[A]
```

微分方程式についても少しだけ解説する.

$$y'' + y = 0,\ y(0) = c,\ y'(0) = 2$$

という初期値問題の解を求め，複数の c に対しグラフを描くということを行う.

Maple では

```
eq:= diff(y(x),x$2)+y(x) = 0;
sol:= dsolve({eq,y(0)=c,D(y)(0)=2},y(x));
with(plots);
f:= unapply(rhs(sol),x,c);
plot({f(x,c) $ c=1..5},x=0..6);
plot({seq(f(x,c),c=[1,3,4])},x=0..6);
```

とする.

Mathematica なら

```
eq= y''[x]+y[x]==0
DSolve[{eq,y[0]==c,y'[0]==2},y,x]
Last[Last[Last[Last[%]]]]
f[x_,c_]= %
Plot[Table[f[x,c],{c,1,5,1}],{x,0,6}]
Plot[Table[f[x,c],{c,{1,3,4}}],{x,0,6}]
```

とする.