

Mathematica

1 Mathematica のもつ意味

Mathematica は Wolfram Research 社(米国)が1988年6月に発売した数学とその応用のためのソフトウェアで、現在各国で非常に多くの研究者・エンジニア・学生によって使われています。はじめマッキントッシュ・コンピュータ版が出されましたが、その後さまざまなコンピュータ・システムで利用可能になっています。現在 Version 3.0 でさらに進化を続け、より充実した機能と使いやすさを追及しています。

数学処理のためのコンピュータ・ソフトウェアは他にも色々ありますが、その中での *Mathematica* の特色は操作が簡単であることと広い範囲の機能を備えていることです。操作が簡単であるということはマッキントッシュ・コンピュータ(アップル社)の基本思想でもあり、従来はある程度の「専門家」を対象にしていたコンピュータを何の予備知識もない初心者でも容易に操作できるようにした、ということです。機能の充実に関していえば、たとえば工科系の大学で4年間に学ぶ基礎および応用の数学のすべてを簡単なコマンドで処理できます。

コンピュータ・ハードウェアは急速に機能を進化させつつ価格を底廉化しています。このような状況での数学処理ソフトウェアの発展の意味するところは次のように述べることができます。

ニュートンが微分積分学を創始して(古典)物理学を体系づけたのが近代科学技術の出発点で、その後の300年間の著しい科学技術の発展はすべて微分積分学に依存してきた、とすることができます。数学とくに応用数学の観点からこの300年を眺めると、「変化を伴う現象は微分方程式で記述され、その微分方程式の解を求める方法を模索しつづけてきた」と言えます。何故かというと、具体的に関数を与えたときその微分を具体的に計算するのは比較的簡単です。それに対して、具体的な関数の積分を求める、もっと一般的にいえば具体的に与えられた微分方程式の解を既知の関数で表現する、ということは極めて難しいことです。正確に言えば「極めて難しい」というより「一般的には不可能」ということであって、種々ある微分方程式のうち解法の知られているものだけを解くことができる、ということです。極論すれば、この「解法」を色々工夫しながら蓄積してきたのが数学とくに応用数学のこの間の歴史であった訳です。

一方では、学生諸君が中学・高校の物理や化学の学習を通して体験しているように、実用上必要なのはユークリッド的厳密解ではなく常に近似解です。これが応用数学上におけるコンピュータの存在理由を与えます。方程式でも微分方程式でも、解法の知られていないタイプの方程式の解を求めることはコンピュータでもできません。しかし、近似解なら解法が知られているかいないかに無関係に求めることができ、実用上はそれで十分だからです。

とくに *Mathematica* はこれを極めて簡単に行います。後述のように、`Solve[]` または `DSolve[]` の `[]` の中に、方程式または微分方程式(常微分方程式)と何を未知数または未知関数とするかを代入してタイプし `shift` + `enter` を押すだけで、解法の知られているタイプのものの大半は即座に厳密解を返してよこします。解法の知られていないタイプのもの、あるいは解法が知られていても解法に合うように変形するのが煩雑なものに対しては、`NSolve[]`、`FindRoot[]` または `NDSolve[]` を用いれば近似解を返してくれます。

Mathematica のもうひとつの特色は、グラフィックスを容易に描くことができることです。`Plot[]` または `Plot3D[]` の `[]` の中に曲線または曲面の方程式と変数の範囲を代入してタイプし、`shift` + `enter` を押すとグラフが描かれます。このように、操作が容易であることとグラフィックスが容易に描かれることが、コンピュータを通して「数学の視覚化(可視化)」や「実験数学」を可能にしている、ということもできます。

Mathematica の類のソフトウェアは今後さらに発展・普及するものと思われます。幸い東京電機大学では Wolfram Research 社とサイトライセンス契約を結んでいるので、学内では誰でも自由に *Mathematica* を使うことができます。Version 4.1 の特徴は、入力や出力における数式の表記方法がふつう教科書に印刷されたりノートに書いたりするものに近くなっていて全体に見やすくなっていることと、計算能力がさらに進化していることです。

この章では *Mathematica* のほんの導入部のみを紹介します。参考文献として

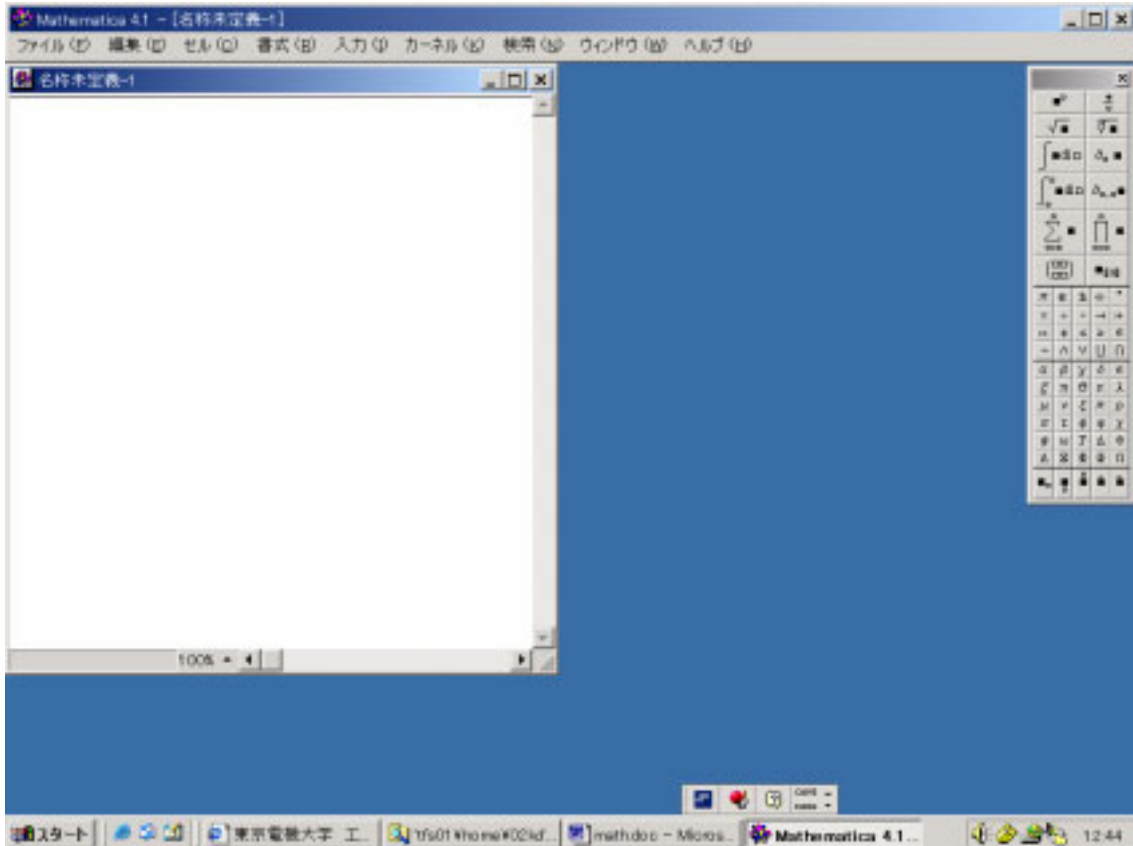
田澤義彦著「*Mathematica* による工科の数学」東京電機大学出版社

を挙げておきますが、この他にも多様な解説書が出版されていますから、さらに詳しく知りたい諸君はそれらを参照して下さい。*Mathematica* を通して、学生諸君が楽しみながら数学を理解することを希望しています。

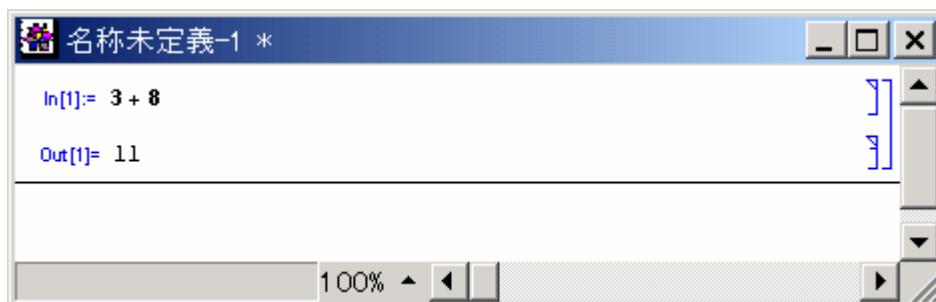
2 Mathematica の基本操作

Mathematica を起動し, 操作し, 終了するには次のように行います。

(1) [スタート][プログラム][数式処理][Mathematica 4.1][Mathematica 4.1]の順にクリックします。

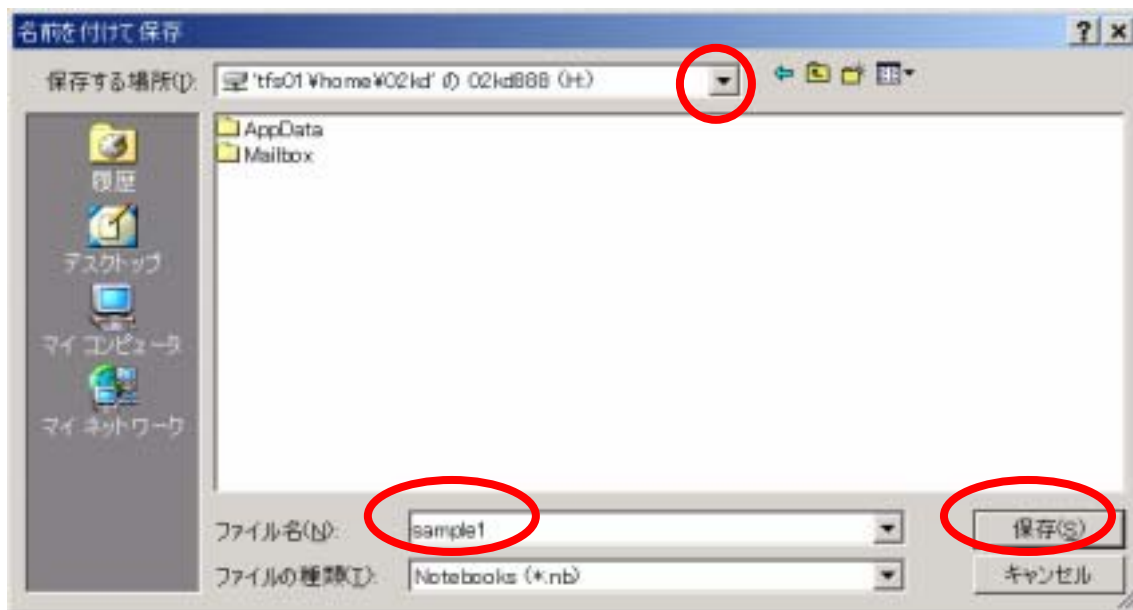


(2) キーボードからたとえば $3+8$ とタイプし, `shift` + `enter` を押すとコマンドが実行され, 計算結果の 11 が表示されます。



(4) *Mathematica* を終了したいときには、[ファイル][終了]の順にクリックします。ダイアログボックスが現れるので、計算結果を保存しないときには[保存しない]をクリックします。

(5) 入力した式を保存したいときには、[ファイル][別名で保存]の順にクリックし、つづいて現れるダイアログボックスで、保存するドライブの選択、ファイル名の入力を行い、最後に[保存]をクリックします。



3 計算

3.1 数の計算

四則演算は普通の電卓と同じように入力し, = の代わりに `shift` + `enter` をおします。乗法では \times の代わりに `*` またはスペースキーをおします。除法では \div の代わりに `/` をおします。計算結果が分数の場合, `N[]` を用いると小数で表示されます。無限小数などの場合は有効数字 6 桁の近似値が表示されますが, たとえば `N[,30]` のように用いると 30 桁まで表示されます。

```
5 6
30
```

```
7/4
7
—
4
```

```
N[100/3]
33.3333
```

```
N[100/3,30]
33.333333333333333333333333333333
```

指数(累乗)は `^` で, 根号(ルート, 2乗根, square root)は `Sqrt[]` で表記されます。入力のしかたが間違っていると, *Mathematica* は警告音とともに赤い文字でエラーメッセージを表示します。

```
3^5
243
```

```
N[Sqrt[[5]]
Syntax::sntxi:
  Incomplete expression.
N[Sqrt[5]]
2.23607
```

円周率 π は Pi で, 自然対数の底 e は E で, 虚数単位 i は I で表記されま
す。

N[Pi,200]

3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196

(1+3 I) (2-I)

5 + 5 I

FactorInteger[] を用いると, 自然数を素因数分解できます。下の例の出力は通常表記法では $2^3 \cdot 3 \cdot 5^2 \cdot 23$ と書かれます。**Sum[a_k , {k,m,n}]** は, 和の記号のように, 数列 $\{a_k\}$ の $k=m$ から $k=n$ までの項の和を計算します。

FactorInteger[13800]

{{2, 3}, {3, 1}, {5, 2}, {23, 1}}

Sum[2^k, {k,0,5}]

63

3.2 式の計算

多項式の計算において, `Expand[]` は展開(括弧をはずす)を, `Factor[]` は因数分解を, `Simplify[]` は昇べきの順の(次数の小さい方からの)整頓を行います。また, `%` は直前の出力結果を表します。有理式においては, `Simplify[]` は通分の結果を計算します。

```
Expand[(2a+b-c)^3]
      3      2      2      3      2      2      2      3
8a +12a b+6ab +b -12a c-12abc-3b c+6ac +3bc -c
```

```
Factor[2 x^2+5 x y+2 y^2+5 x+y-3]
(-1 + 2 x + y) (3 + x + 2 y)
```

```
Simplify[ 2 x^2+5 x y+2 y^2+5 x+y-3 ]
      2      2
-3 + 5 x + 2 x + y + 5 x y + 2 y
```

Mathematica では三角関数 $\sin x$, $\cos x$, $\tan x$, 指数関数 e^x , 対数関数 $\log x$ はそれぞれ `Sin[x]`, `Cos[x]`, `Tan[x]`, `Exp[x]`, `Log[x]` と表記されます。

```
Tan[Pi/3]
Sqrt[3]
```

```
Simplify[Cos[x] Cos[3y]-Sin[x]Sin[3y]]
Cos[x + 3 y]
```

```
Exp[2]^3
6
E
N[Log[2]+Log[E]]
1.69315
```

3.3 ベクトルの計算

ベクトルの線形演算(ベクトルの和, スカラー倍)・内積・外積の計算をします。線形演算は文字式の加法・乗法のように入力します。内積は . (ピリオド)を用います。

$$v1=\{2, -1, 3\}$$

$$\{2, -1, 3\}$$

$$v2=\{5, 0, -2\}$$

$$\{5, 0, -2\}$$

$$a v1 + b v2$$

$$\{2 a + 5 b, -a, 3 a - 2 b\}$$

$$v1.v2$$

$$4$$

3.4 行列の計算

行列の線形演算(行列の和, スカラー倍)・積・行列式・逆行列・固有値・固有ベクトルの計算を行います。*Mathematica* では行列は, 下の例の第1行の右辺の形で表記されますが, `MatrixForm[]` のコマンドを用いると通常のに直すことができます(行列を囲む括弧は表示されませんが)。線形演算はベクトルの場合と同様, 文字式の加法・乗法のように入力します。下の例で行列 `m1`, `m2` を定めていますが, `=`ではなく `:=`を用いているので, `shift` + `enter` をおしても画面には `m1`, `m2` の形は出力されません。行列の積は . (ピリオド)で, 行列式は `Det[]` で, 逆行列は `Inverse[]` で計算されます。また, ここでは例はあげませんが固有値は `Eigenvalues[]` で, 固有ベクトルは `Eigenvectors[]` でそれぞれ計算されます。

$$m1:=\{\{2, -3, 1\}, \{-1, 2, -4\}, \{3, 1, 2\}\}$$

$$m2:=\{\{-2, 1, 3\}, \{3, -1, 2\}, \{1, 4, 0\}\}$$

$$\text{MatrixForm}[m1]$$

$$\begin{array}{ccc} 2 & -3 & 1 \\ -1 & 2 & -4 \\ 3 & 1 & 2 \end{array}$$

MatrixForm[m2]

```
-2  1  3
3   -1 2
1   4  0
```

5 m1 -3 m2

```
{{16, -18, -4}, {-14, 13, -26},
 {12, -7, 10}}
```

MatrixForm[%]

```
16   -18   -4
-14   13   -26
12    -7    10
```

MatrixForm[m1.m2]

```
-12  9  0
4    -19 1
-1   10 11
```

m3:={{1,2,3},{2,3,4},{3,2,2}}

Det[m3]

-1

m4:=Inverse[m3]

MatrixForm[m4]

```
2   -2  1
-8  7   -2
5   -4  1
```

MatrixForm[m3.m4]

```
1  0  0
0  1  0
0  0  1
```

3.5 微分積分の計算

Mathematica は極限, 微分, 不定積分, 定積分, 偏微分, 重積分の計算を行うことができます。数列および関数の極限は `Limit[, ->]` を用いて計算します。

```
Limit[ (1+1/n)^n, n->Infinity ]
```

E

```
Limit[ Sin[x]/x,x->0 ]
```

1

微分は `D[,x]` を用いて計算します。 n 次導関数は `D[,{x,n}]` で求められます。

```
D[x^2 Sin[x],x ]
```

2

$x \cos[x] + 2 x \sin[x]$

```
D[x^2 Sin[x], x,x ]
```

2

$4 x \cos[x] + 2 \sin[x] - x^2 \sin[x]$

不定積分は `Integrate[,x]` を, 定積分は `Integrate[,{x, , }]` を用いて計算します。`NIntegrate[,{x, , }]` を用いれば, 定積分の近似値が求められます。

```
Integrate[ 3 x^2+5 x-1,x ]
```

2

$5 x^3$

$-x + \frac{5}{2} x^2 + C$

2

```
Integrate[ Sin[x], {x,0,Pi}]
```

2

```
Integrate[Log[x],{x,1,2}]
```

$-1 + 2 \log[2]$

```
NIntegrate[ Log[x],{x,1,2} ]
```

0.386294

偏微分も $D[\quad ,]$ で計算されます。ここでは例はあげませんが、重積分は累次積分に直して入力します。

$$D[x^2 \text{ Sin}[2y], x] \\ 2 x \text{ Sin}[2 y]$$

$$D[x^2 \text{ Sin}[2y], y] \\ 2 x \text{ Cos}[2 y]$$

$$D[x^2 \text{ Sin}[2y] , x, x] \\ 2 \text{ Sin}[2 y]$$

$$D[x^2 \text{ Sin}[2y] , x, y] \\ 4 x \text{ Cos}[2 y]$$

4 方程式

4.1 方程式

`Solve[f[x]==0,x]` の形で入力することにより, x を未知数とする方程式 $f(x) = 0$ を解くことができます。等号を2つ続ける点に注意して下さい。 $f(x)$ が4次までの多項式の場合は係数に文字が入っていても *Mathematica* は解を出します。ガロア理論により, 5次以上の方程式は四則演算と開平では解けないことが知られていますが, その場合には `NSolve[f[x]==0,x]` の形で入力すると *Mathematica* は近似解を出します。また, 無理方程式, 分数方程式も解くことができます。

$$\text{Solve}[a x^2+b x+c==0,x]$$
$$\frac{-b - \sqrt{b^2 - 4 a c}}{2 a}, \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}$$

$$\text{NSolve}[x^5+2 x^4+3 x^3-x^2+2 x-5==0,x]$$
$$\{x \rightarrow -1.42971 - 1.44787 I\}, \{x \rightarrow -1.42971 + 1.44787 I\},$$
$$\{x \rightarrow -0.0178731 - 1.16134 I\}, \{x \rightarrow -0.0178731 + 1.16134 I\},$$
$$\{x \rightarrow 0.895166\}$$

$$\text{Solve}[\text{Sqrt}[x-1]==x-3,x]$$
$$\{x \rightarrow 5\}$$

逆三角関数・指数関数・対数関数などの超越関数を含んでいる方程式も, 簡単な形のものそのまま解けます。

$$\text{Solve}[\text{ArcSin}[x+5]==\text{Pi}/6,x]$$
$$\{x \rightarrow -(-)\}$$

しかし, たとえば $\log(x+2) - \log x = 3$ のような場合には, `Solve` や `NSolve` ではうまくいかず, `FindRoot` を用いるか, または `Solve[Log[(x+2)/x]==3,x]` のよ

うに変形して入力するなどの工夫が必要です。三角関数を含む場合は **FindRoot** を用います。**FindRoot[f[x]==0,{x,a}]** の形で入力すると $x = a$ の近くでの $f(x) = 0$ の解を求めることができます。

```
FindRoot[Sin[x+2]+Cos[2x]==Pi/6,{x,0}]  
{x -> 0.725535}
```

4.2 連立方程式

x, y を未知数とする方程式 $f(x) = 0, g(x) = 0$ は

```
NSolve[{f[x]==0,g[x]==0},{x,y}],NSolve[{f[x]==0,g[x]==0},{x,y}],  
FindRoot[{f[x]==0,g[x]==0},{x,y}]
```

などの形で入力して解くことができます。未知数や方程式の数が多い場合もこれに準じます。連立1次方程式においては、解が自由度を持つ場合や解が存在しない場合もこめて答えを出します。

```
Solve[{ x+2y-4z== -7,  
        x+3y-6z== -11,  
        2x+4y-7z== -11  },{x,y,z}]  
{x -> 1, y -> 2, z -> 3}
```

```
Solve[{ x-2y- z== 4,  
        x+ y+2z== -1,  
        2x- y+ z== 3  },{x,y,z}]  
2          5  
{x -> - - z, y -> -(-) - z}  
3          3
```

```
Solve[{ x+2y+3z== 4,  
        2x+5y+8z==10,  
        3x+2y+ z== 0  },{x,y,z}]  
{}
```

```
Solve[{x^2+y^2==1,x+y==0},{x,y}]  
1          1  
{x -> -(-----), y -> -----},  
          Sqrt[2]      Sqrt[2]  
1          1  
{x -> -----, y -> -(-----)}}  
          Sqrt[2]      Sqrt[2]
```

```
FindRoot[{Sin[x]==x-y, Cos[y]==x+y},{x,1 },{y,0 }]  
{x -> 0.883401, y -> 0.1105}
```

4.3 微分方程式

x を独立変数, y 未知関数とする微分方程式 $f(x, y, y', y'', \dots) = 0$ は `DSolve[f[x,y,y',y'',...]==0,y[x],x]` の形で解くことができます。初期条件を与えて解を求める場合には2番目の例のように入力します。3番目の例のように解が求まらない場合でも, `NDSolve[]` を用い初期条件を与えて x の区間を限定すると, 数値解が求まります。この解は, x の区間内の分点に対し解の近似値を求め補間して得られる関数なので, 具体的な形では表示されません。それでも, 関数値を求めたりグラフを描いたりすることができます。

連立微分方程式も

```
DSolve[{f[x,y,z,y',z',...]==0,g[x,y,z,y',z',...]==0},{y[x],z[x]},x]
```

の形で解くことができます。

```
DSolve[y'[x]==-x^3 y[x]^2 ,y[x],x]
```

```
1
{{y[x] -> -----}, {y[x] -> 0}}
4
x
-- + C[1]
4
```

```
DSolve[{y''[x]+y[x]==0 ,y[0]==0, y'[0]==-1} ,y[x],x]
```

```
{{y[x] -> -Sin[x]}}
```

```
DSolve[y''[x]+Sin[y[x]]==0,y[x],x]
```

```
DSolve::dnim:
```

```
Built-in procedures cannot solve this differential equation.
```

```
DSolve[Sin[y[x]] + y''[x] == 0, y[x], x]
```

同じ微分方程式を `NDSolve` を用いて解きます。関数値やグラフを調べるために, その解を `f[x]` とおきます。関数を定義するときには下の例のように, `f[x_]` の x のあとに下線をひきます。2度目からは `f[x]` のままで使えます。

```
f[x_]=NDSolve[{y''[x]+Sin[y[x]]==0,y[0]==0,y'[0]==2 },
```

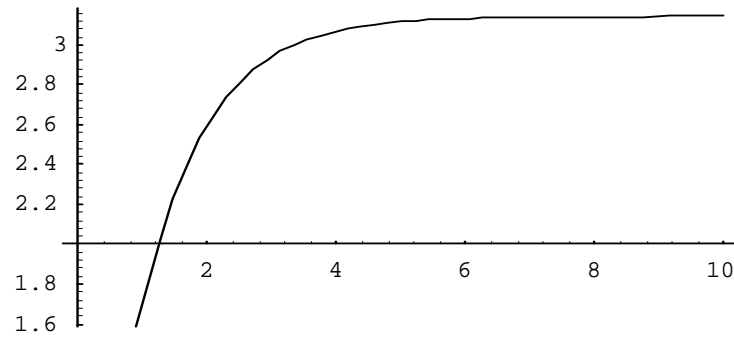
```
y[x] ,{x,0,10}]
```

```
{{y[x] -> InterpolatingFunction[{0.,10.}, <>][x]}}
```

f[3]

{{y[3] -> 2.94261}}

Plot[Evaluate[y[x]/.f[x]],{x,0,10}]



5 グラフィックス

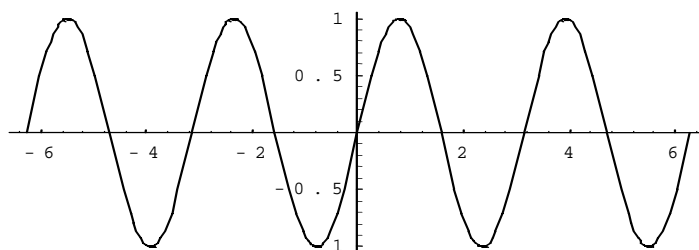
グラフィックスが簡単に描けるのは *Mathematica* の特長のひとつです。基本形は `Plot[]` のコマンドですが、色々なオプションやパッケージなどが用意されていて、2次元および3次元のグラフィックスを様々な形で描くことができます。また、出力されたグラフィックスをアニメーションの形で見することもできます。

5.1 2次元のグラフィックス

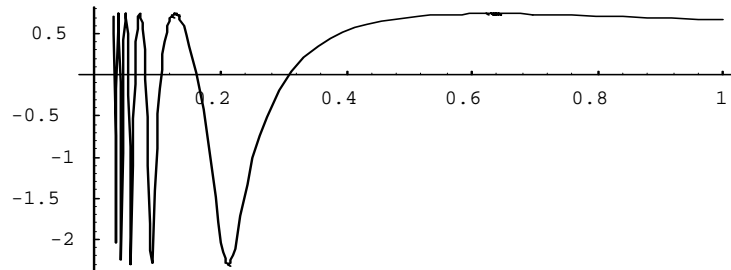
`Plot[f[x],{x,a,b}]` と入力すると関数 $y = f(x)$ のグラフが $a \leq x \leq b$ の範囲で描かれます(プロットされます)。座標軸を表す変数は示されませんので、`Plot[f[t],{t,a,b}]` としても同じ結果が得られます。その時は横軸を t 軸と見なしています。`Plot[{f[x],g[x],...},{x,a,b}]` と入力すると、複数のグラフ $y = f(x)$, $y = g(x)$, ... が同時に描かれます。

図全体の縦と横の比率はそのままでは座標軸上の目盛りに関係なく一定(いわゆる黄金分割の比)ですが、オプションを使って変えられます(後述)。図の付近にポインターを合わせてクリックすると、図のまわりが長方形の枠で囲まれます。その長方形の各頂点と各辺の midpoint にある小さな黒い正方形のどれかひとつにポインターをあわせてマウスボタンを押したままマウスを移動すると(ドラッグすると)、図の大きさが変えられます。また、枠内にポインターをおいてドラッグすると、図の位置が変えられます。画面上の他の点をクリックすると枠は消えます。

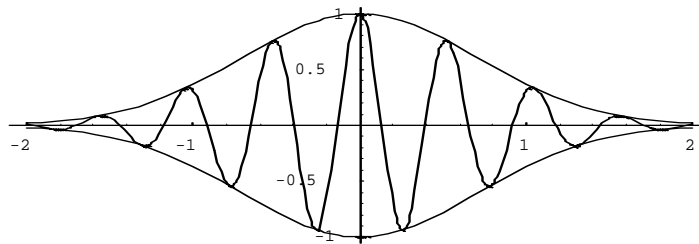
```
Plot[
  Sin[ 2x ],{ x, -2Pi, 2Pi }
]
```



```
Plot[
  Log[ Sin[ 1/x ]+11/10 ], { x, 0.03, 1 }
]
```



```
Plot[
  { Exp[ -x^2 ], -Exp[ -x^2 ], Exp[ -x^2 ] Cos[ 12 x ] },
  { x, -2, 2 }
]
```



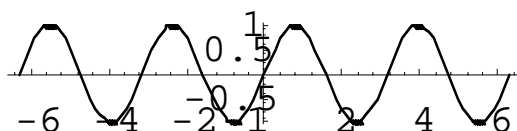
5.2 オプション

コマンドを入力すると、*Mathematica* はあらかじめ設定された最も一般的と思われる形で計算結果を出力します。この設定をデフォルト設定と言います。ユーザが自分の目的に合った形で出力させたいと思う場合にはオプションを用いて設定を変えます。これは他の場合でも同様ですが、ここでは2次元のグラフィックスのオプションを説明します。

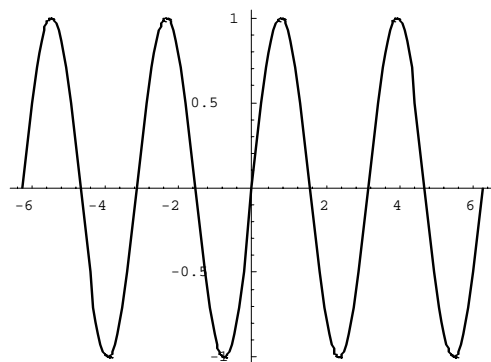
まず、前の項で述べたグラフの縦軸と横軸の比率を考えます。2110ページのコマンド `Plot[Sin[2x],{x,-2Pi,2Pi}]` に `AspectRatio->Automatic` という項を挿入して下のように入力すると、縦軸と横軸の目盛りの比率が1:1になります。

`AspectRatio->a` とすると(縦軸全体の長さ):(横軸全体の長さ)が1:a になります。

`Plot[Sin[2x], { x, -2Pi, 2Pi }, AspectRatio->Automatic]`



`Plot[Sin[2x], { x, -2Pi, 2Pi }, AspectRatio->3/2]`



その他にも多様なオプションがありますがいくつかを挙げると `Axes->False` (座標軸を表示しない), `Frame->True` (図の周りを長方形の枠で囲む), `GridLines->Automatic` (方眼紙のような縦横の線を入れる), `AxesLabel->{"x","y"}` (横軸に x, 縦軸に y と変数名を表示する), `Ticks->None` (座標軸に目盛りをつけない), `PlotRange->{a,b}` (縦軸方向に a から b までの範囲のみグラフを描く), `PlotStyle->RGBColor[a,b,c]` (描かれる曲線の色を, 赤緑青の比率が a:b:c になるように変える)などがあります。`RGBColor` では a, b, c の値は0と1の間の数値にします。これらのオプションはいくつかを組み合

わせて用いることもできます。

5.3 パッケージ

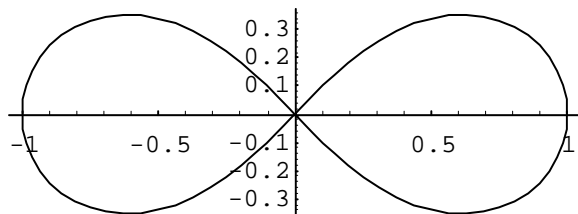
Mathematica の持っている機能のうち標準的なものは *Mathematica* を起動しただけで自動的に作動しますが、やや高度な機能はいくつかあるパッケージの中に収められていて、必要に応じて呼び出さなければなりません。

例として、`ImplicitPlot` というコマンドをあげます。陰関数の表示された曲線を描くには `Plot` でなく `ImplicitPlot` というコマンドを用いるのですが、このコマンドは `Graphics` フォルダの中の `ImplicitPlot` というパッケージに入っています。

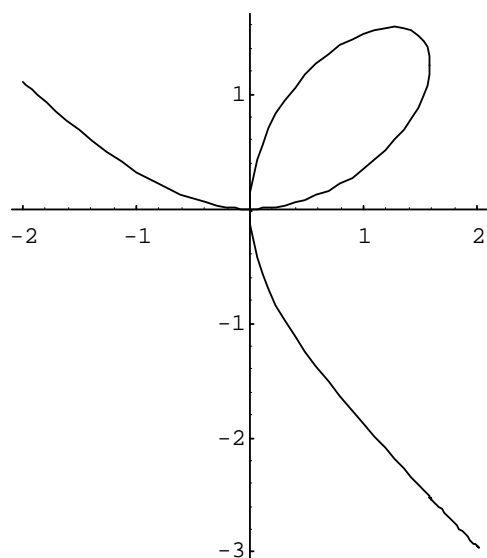
`<<Graphics`ImplicitPlot`` とキーボードでタイプし `shift` + `enter` を押すことでこのパッケージが開くことができます。画面上にはとくに何も表示されません。この手続きのあとは、`ImplicitPlot[f[x,y]==0,{x,a,b}]` と入力すると曲線 $f(x,y) = 0$ が $a \leq x \leq b$ の範囲で描かれます。一度開いたパッケージは *Mathematica* を終了するまで開かれています。終了して再び起動したときにそのパッケージが必要なならばまた同じ手続きをします。

```
<<Graphics`ImplicitPlot`
```

```
ImplicitPlot[ (x^2 + y^2)^2 == (x^2 - y^2), { x, -1, 1 } ]
```



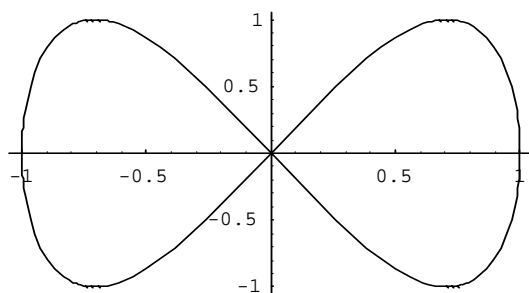
```
ImplicitPlot[ x^3 - 3 x y + y^3 == 0, { x, -2, 2 } ]
```



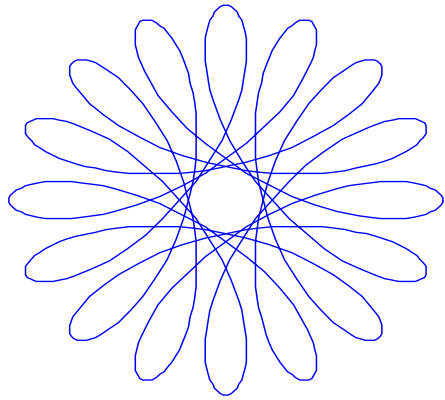
5.4 曲線のパラメータ表示

パラメータ t を用いて $x = f(t)$, $y = g(t)$, $a \leq t \leq b$ のように表示されている曲線を描くには, `Plot[{f[t],g[t]},{t,a,b}]` と入力します。

```
ParametricPlot[
  { Sin[ t ], Sin[ 2 t ] }, { t, 0, 2Pi }
]
```



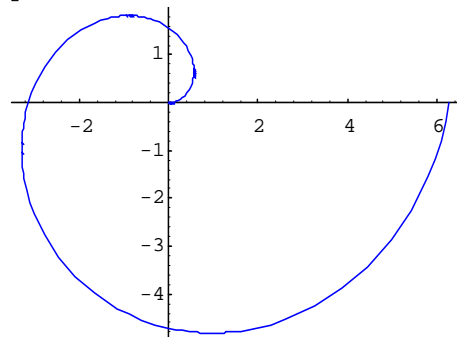
```
ParametricPlot[
  { 5 Cos[ -11/5 t ] + 7 Cos[ t ], 5 Sin[ -11/5 t ] + 7 Sin[ t ] },
  { t, 0, 10 Pi },
  AspectRatio->Automatic,
  Axes -> None,
  PlotStyle -> { RGBColor[ 0, 0, 1 ] }
]
```



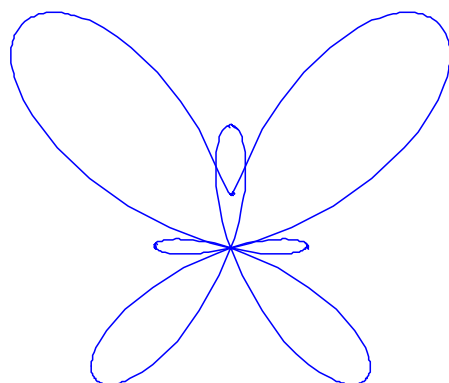
とくに曲線が極座標において $r = f(\theta)$, a b のように極方程式で表示されている場合には, `ParametricPlot[{f[t] Sin[t],f[t] Cos[t]},{t,a,b}]` のように一旦 xy 座標に直してから上記のコマンド `ParametricPlot` を用いてもその曲線を描くことはできますが, 下に示すようにコマンド `PolarPlot` を用いれば極方程式をそのまま代入することができます。ただし, `PolarPlot` は `Graphics` フォルダの `Graphics` パッケージに入っているので, まずそれを呼び出す必要があります。

```
<<Graphics`Graphics`
```

```
PolarPlot[
  theta, { theta, 0, 2 Pi },
  PlotStyle -> { RGBColor[ 0, 0, 1 ] }
]
```



```
PolarPlot[
  Exp[ Sin[ theta ] ] - 2 Cos[ 4 theta ], { theta, 0, 2 Pi },
  Axes -> None,
  PlotStyle -> { RGBColor[ 0, 0, 1 ] }
]
```



5.5 3次元グラフィックス

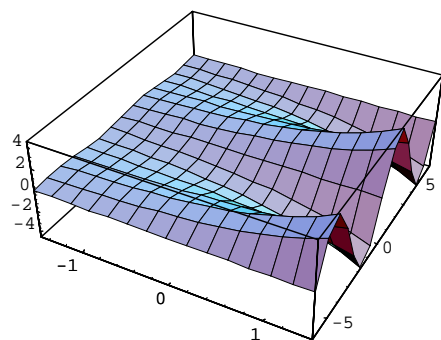
`Plot3D[f[x,y],{x,a,b},{y,c,d}]` と入力すると2変数の関数 $z = f(x, y)$ のグラフが $a \leq x \leq b, c \leq y \leq d$ の範囲で描かれます。`ParametricPlot3D` を用いて `ParametricPlot3D[{f[s,t],g[s,t],h[s,t]},{s,a,b},{t,c,d}]` と入力すると s, t をパラメータとする曲面 $(f(s, t), g(s, t), h(s, t))$ が $a \leq s \leq b, c \leq t \leq d$ の範囲で描かれます。

曲面は、いくつかのサンプル・ポイントを取り、それを線分で結んでできる網目模様面に面を張り付けたような形で表示されます。また、立体感を出すために、赤・緑・青の3原色の光源に照らされているかのように、彩色されて描かれます。

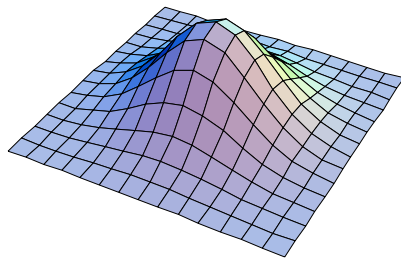
オプションとしては、`Boxed->False` (曲面をかこむ立方体の枠を表示しない)、`Axes->None` (座標軸を表示しない)、`Shading->False` (網目の間の面を彩色しない)、`Mesh->False` (網目を描かない)、`Lighting->False` (3原色の光源を用いず、高い方を明るく低い方を暗くモノクロで表示する)、`PlotPoints->{a,b}` (サンプル・ポイントの個数を $a \times b$ にする)、`ViewPoint->{a,b,c}` (点 (a, b, c) から眺めた形に描く。デフォルト値は $(1.3, -2.4, 2)$ など数多くあります。オプションではありませんが、`Show[ContourGraphics[]]` を用いて曲面の等高線を描くこともできます。

また、2次元の場合にも可能ですが、`Show[GraphicsArray[]]` を用いるといくつかのグラフを表のように縦横に配列することができます。

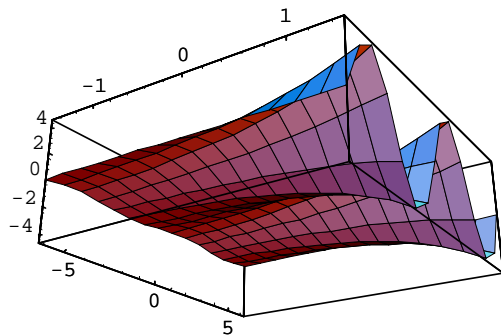
```
Plot3D[
  Exp[ x ] Sin[ y ],
  { x, -1.5, 1.5 },
  { y, -2Pi, 2Pi }
]
```



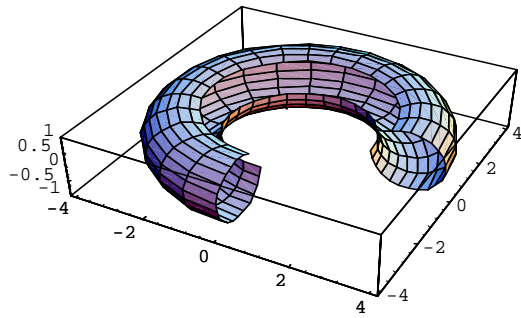

```
Plot3D[
  Exp[ -x^2 ] Exp[ -y^2 ],
  { x, -2, 2 },
  { y, -2, 2 },
  Boxed->False,
  Axes->None
]
```



```
Plot3D[
  Exp[ x ] Sin[ y ],
  { x, -1.5, 1.5 },
  { y, -2Pi, 2Pi },
  ViewPoint->{ 1.3, -2, -1 }
]
```



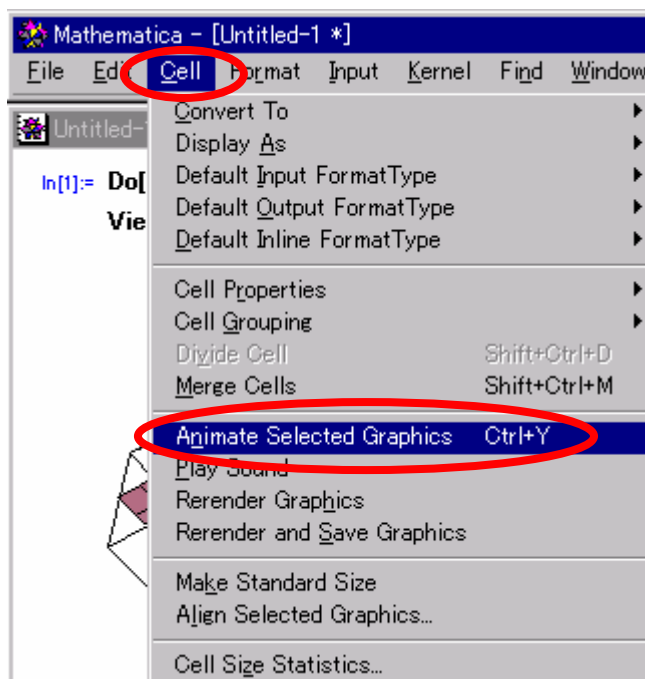
```
ParametricPlot3D[  
  { Cos[ s ]( 3+Cos[ t ] ), Sin[ s ]( 3+Cos[ t ] ), Sin[ t ] },  
  { s, 0, 3Pi/2 },  
  { t, -Pi, 2Pi/3 }  
]
```



5.6 アニメーション

$y = f(x, k)$, $z = f(x, y, k)$ のように曲面や曲線の方程式が文字の定数 k を含んでいるとき, k の値を色々変えると多数の曲線や曲面, いわゆる曲線族・曲面族が得られます。Mathematica ではこれらの曲線族や曲面族をアニメーションに見ることができます。

まず, `Do[Plot[f[x,k]],{x,a,b},{k,c,d,e}]` と入力すると, $y = f(x, k)$ において k の値を $k = c$, $k = c + e$, $k = c + 2e$, ..., $k = d$ と変えて得られる曲線をひと駒ずつ描かれます。グラフがすべて描かれたら, アニメーションの最初の駒となるグラフのセルを選択します。具体的には, そのグラフの右端にある上下に延びた線分(上端と下端がわずかに内側に鍵型に曲がっている)をクリックして白黒反転させます。次に, 画面を最後の駒となるグラフまで移動し, `shift` を押したままそのグラフのセルを選択します。これで中間の駒のセルがすべて選択されます。次に[CELL][Animate Selected Graphics] を選択すると動きはじめます。



アニメーションの動く早さを変えたいときは, 画面左下隅にカセットデッキプレーヤーの操作ボタンのようなものが表示されますから, マウスを使ってスピード調整や駒止めなどができます。



下の例ではスペースの都合上3コマしか印刷していません。

```
Do[
  Plot3D[
    Exp[ x ] Sin[ y ],
    { x, -1.5, 1.5 },
    { y, -2Pi, 2Pi },
    ViewPoint->{ 1.3, -2, a }
  ],
  { a, -3, 3, 0.5 }
]
```

