

# 数式処理システム *Mathematica*について

神戸大学工学部情報知能工学科 田村直之 (tamura@kobe-u.ac.jp)

神戸大学総合情報処理センター 辻香織 (kaori@kobe-u.ac.jp)

## 1 はじめに

Wolfram Research が開発した, *Mathematica*は数式処理, 数値計算はもちろん, 3次元グラフィックス表示, アニメーション作成などの機能を持つ, 最も現代的な数学統合環境ソフトウェアです.

神戸大学は, 昨年度から *Mathematica*のキャンパスライセンスを有していますので, 神戸大学に所属している人間が, キャンパス内かつ神戸大学管理の計算機上で利用する場合, わずかな金額で, 皆さんのコンピュータ上で利用できます (ほとんどのパーソナルコンピュータやワークステーションでOK).

皆さんも *Mathematica*を使ってみましょう!

## 2 *Mathematica*の特徴

*Mathematica*は, Wolfram Research, Inc. が開発した, 最も現代的な数学統合環境ソフトウェアです (Ver 2.2 が最新). 下のマシンや環境上で利用可能です.

Macintosh (680x0, Power Mac), Microsoft Windows, IBM PC, NEC PC, Sun SPARC, Hewlett-Packard, DEC (OSF/1, RISC, VAX/VMS), IBM RISC, Silicon Graphics, NEC EWS, NEXTSTEP, Sony RISC, CONVEX, Hitachi

*Mathematica*の特徴を挙げてみましょう.

### 数式処理, 数値計算

微積分, 方程式の求解などの数式処理や, 任意精度での数値計算が可能です.

### グラフィックス, オーディオ, テキスト

2次元のプロットはもちろん, 3次元プロット, 等高線プロット, 濃淡プロットが可能です (当然カラー, ただし色の趣味は悪いと思います...). アニメーション表示もできます. また, オーディオの再生, 任意のフォントでのテキスト表示が可能です. グラフィックスは PostScript 形式で出力されますので,  $\TeX$ な

どの他のプログラムでも簡単に利用できます。数式の表示は美しくありませんが、 $\text{T}_{\text{E}}\text{X}$ 形式で出力することが可能です。

### プログラミング

*Mathematica*のプログラミング言語はかなり豊富な機能を持っています(750近くの組込関数があります)。プログラムを書くことによって、*Mathematica*の機能を拡張したり、変更したりすることが可能です。

### 会話的な利用形態

フロントエンドを用いると、マウスとキーボードから簡単に *Mathematica* を利用できます。

### ノートブック

数式、グラフィックス、サウンド、テキストなどを組み合わせた文書(ノートブックと呼ばれます)を作成できます。たとえば、オンラインの教科書を作成することが可能です。

### 多くの拡張パッケージ

*Mathematica*のプログラミング機能を利用して作成された多くの拡張パッケージが存在し、*Mathematica*の中で簡単に利用できます。

標準で付属しているのは、代数、微積分、離散数学など165のパッケージです。またWolframが公開している多数(650近く)のパッケージがあり、ネットワークを通じて無料で入手できます(このデータベースは *MathSource* と呼ばれており、CD-ROMで販売もされています)。Wolframや他の会社が販売しているパッケージもあります(電気、光学、制御、財務、3Dリアルタイム表示など)。

### リンク機能

*Mathematica*から他のプログラムを呼び出したり、反対に他のプログラムから *Mathematica*を呼び出したりすることが可能です(*MathLink*機能)。特に、Excel, Spyglass, Xmath, MATLAB, Visual Basicなどと融合的に利用する機能が用意されています(そのためのパッケージが必要です)。

### 広く用いられている

他の数式処理システム(MAPLE, REDUCEなど)と比べて、多くのユーザが利用しています。教科書も多数出版されています。

### キャンパス・ライセンス

神戸大学はキャンパス・ライセンスを持っていますので、神戸大学に所属して

いる人間は、キャンパス内かつ神戸大学管理の計算機上ならば、自由に利用できます(総合情報処理センターへの申し込みと利用料金の支払いが必要)。

### 3 Mathematicaの入手方法

Mathematicaを利用してみたい人は、神戸大学総合情報処理センターを通じて入手できます。

平成7年6月1日より、総合情報処理センターが窓口となり、利用の受付を開始しています。利用ご希望の方はまず、総合情報処理センター本館事務室にあります所定の申請書及び同意書に必要事項をご記入の上、センター業務掛にてお申し込み下さい。その際、同意書には利用機器名、IPアドレスを利用台数分ご明記下さい。電話、または電子メールでの申請書の請求も受け付けています。

なお、今年度のライセンス期限は平成8年3月31日までとなっており、申請は年度ごとに更新をして頂くこととなります。

ソフトウェアの配布方法ですが、ワークステーション用パッケージはftpにて入手して頂けますが、パソコン用パッケージはフロッピーディスクの貸し出しをしておりますので、ご足労ですがセンター本館事務室までお越し下さい。

どちらの場合も、インストールガイドとリリースノートをパッケージとともにお渡しています。その他のマニュアル類についても貸し出しを受け付けていますので、ご希望がであればお申し出下さい。

利用にあたっての負担金は、

ワークステーションライセンスサーバ	10,000円
ワークステーションクライアント	2,000円
パーソナルコンピュータ	2,000円

となっており、これは本年度センターが窓口となることに先だって、各部局宛に利用予定機器台数のアンケート調査を行った結果に基づいて決定されました。

なお、負担金はセンター利用料と同様に教官研究費の予算組み替により、年間一括払いで徴収致します。

また、利用申請されましたマシン管理者の方々全員を、後述のMathematicaユーザの為のメーリングリスト [Mathematica-admin@godel.seg.kobe-u.ac.jp](mailto:Mathematica-admin@godel.seg.kobe-u.ac.jp)へ登録しております。学内のMathematicaユーザ全員が登録されますので、情報を有効に交換、活用頂けることと思います。

Mathematica利用申請に関するお問い合わせ、ご意見ご希望などは、

総合情報処理センター 業務掛

tel : 803-0191 もしくは内線 2912  
fax : 803-0193 もしくは内線 2930  
e-mail : math-admin@kobe-u.ac.jp

までお寄せ下さい。

## 4 Mathematicaの使用例

ここでは、簡単な Mathematica を使用例を紹介します。In[n]:= の後がユーザによる入力、Out[n]= の後が Mathematica による出力です。なお入力中、(\* ... \*) の部分はコメントです。

以下で (急カーブ注意!) で始まる段落は、少し高度な内容を含んでいます。Mathematica の初心者の方はその部分を無視していただいて結構です。

### 4.1 数値計算

まず、 $e^{i\pi} + 1$  の値を計算してみましょう。

```
In[1]:= E^(I Pi)+1  
Out[1]= 0
```

E, I, Pi はそれぞれ  $e$ ,  $i$ ,  $\pi$  を表します。また $\wedge$ はべき乗、空白は積を表します (I\*Pi でも良い)。

では次に  $\frac{192}{\sqrt{4989}}$  を計算してみます。

```
In[2]:= 192/Sqrt[4989]  
Out[2]=
```

```
64 Sqrt[ $\frac{3}{1663}$ ]
```

ここで、Sqrt[x] は  $\sqrt{x}$  を表します。Mathematica は結果をできるだけ厳密なままで表現しようとするので、結果を勝手に数値に直すことはしません。

次のようにすれば、この結果を数値に直すことができます。

```
In[3]:= N[%]  
Out[3]= 2.71828
```

ここで、N[x] は  $x$  を数値に変換する Mathematica の関数です。また%は直前に実行した計算結果を表します。

関数 `N` の第 2 引数に精度を与えることによって、より高い精度の数値を求めることもできます。

```
In[4]:= N[192/Sqrt[4989], 50]
Out[4]= 2.7182817961040513631169754774169789462547621424181
```

`N` や `Sqrt` のように *Mathematica* にあらかじめ用意されている関数を組込関数と呼びます。 *Mathematica* には 750 近くの関数が組み込まれています。

## 4.2 オンライン・ヘルプ

?関数名と入れることによって、組込関数の説明を見ることができます(より詳しい説明は??関数名で表示されます)。また、関数名中に\*を含めると、\*を任意の文字列として、そのパターンに一致する関数名の一覧が表示されます。

```
In[5]:= ?Log
Log[z] gives the natural logarithm of z (logarithm to
base E). Log[b, z] gives the logarithm to base b.
In[6]:= ?Log*          (* Log で始まる関数名の一覧 *)
Log                    LogicalExpand LogIntegral
LogGamma
```

ノートブック・フロントエンドを使っている場合は、メニューから `Open Function Browser` を選ぶと組込関数の一覧が表示されます。

## 4.3 数式処理

微積分や方程式の求解などさまざまな計算が可能です。たとえば `Integrate[expr, x]` は式 `expr` を変数 `x` の元で積分し、`D[expr, x]` は偏微分します。`Simplify[expr]` は式 `expr` を単純化した結果を返します。

```
In[7]:= Integrate[ x^2 Sin[x]^2, x ]    (* ∫ x² sin² x dx *)
Out[7]=
      4 x³ - 6 x Cos[2 x] + 3 Sin[2 x] - 6 x² Sin[2 x]
      -----
                24
```

```
In[8]:= D[%, x]                        (* 上の結果の偏微分 *)
Out[8]=
      12 x² - 12 x² Cos[2 x]
      -----
                24
```

In[9]:= Simplify[%] (\* 上の結果の簡単化 \*)

Out[9]=  
 $x^2 \sin[x]^2$  (\* もとに戻った! \*)

計算した結果は=により変数に代入することができます。変数には数値、数式、グラフィックなど Mathematicaで取り扱える任意のオブジェクトを代入できます。

In[10]:= exp1 = Expand[(a+b)^4] (\* (a+b)^4の展開を exp1 に代入 \*)

Out[10]=  
 $a^4 + 4 a^3 b + 6 a^2 b^2 + 4 a b^3 + b^4$

In[11]:= Factor[exp1] (\* exp1 の因数分解 \*)

Out[11]=  
 $(a + b)^4$

関数を定義する場合は:=を使います。左辺に現れる変数には、\_ (下線)を付けなければなりません。

In[12]:= f[x\_] := x^3-7x^2+3a x (\* 関数 f(x) = x^3-7x^2+3ax を定義 \*)

In[13]:= f[1] (\* f(1) の値を求める \*)

Out[13]= -6 + 3 a

関数 Solve により方程式の解を求めることができます。同等関係を表すには=ではなく==を使います。

In[14]:= sol = Solve[ f[x]==0, x ] (\* f(x) = 0 の解を sol に代入 \*)

Out[14]=  
{ {x -> 0},  
 {x ->  $\frac{7 - \text{Sqrt}[49 - 12 a]}{2}$ },  
 {x ->  $\frac{7 - \text{Sqrt}[49 + 12 a]}{2}$ }}

3つの解が求められています。解全体が変数 sol に代入されているとき、n 番目の解を式 expr に適用した結果を求めるには、expr /. sol[[n]] と入力します。

In[15]:= Expand[ x^2 /. sol[[2]] ]

Out[15]=  
 $\frac{49}{2} - \frac{7 \text{Sqrt}[49 - 12 a]}{2} - 3 a$

In[16]:= % /. a->2 (\* さらに a を 2 とおく \*)

Out[16]= 1

以下のようにして、連立方程式も `Solve` で解くことができます。微分方程式には `DSolve` を使います。解析的に解を求められない場合は、`NSolve` や `NDSolve` により数値的に解を求めることができます。

```
In[17]:= Solve[ {x^3+y^3==1, x+y==2}, {x,y} ]
```

```
Out[17]=
```

$$\left\{ \left\{ x \rightarrow 1 - \frac{I}{\sqrt{6}}, y \rightarrow \frac{12 + 2 I \sqrt{6}}{12} \right\}, \right.$$

$$\left. \left\{ x \rightarrow 1 + \frac{I}{\sqrt{6}}, y \rightarrow \frac{12 - 2 I \sqrt{6}}{12} \right\} \right\}$$

```
In[18]:= DSolve[ y''[x]-k y[x]==1, y[x], x ]
```

```
Out[18]=
```

$$\left\{ \left\{ y[x] \rightarrow -\left(\frac{1}{k}\right) + \frac{C[1]}{E^{\sqrt{k} x}} + E^{\sqrt{k} x} C[2] \right\} \right\}$$

`DSolve` の結果中、`C[1]`、`C[2]` は積分定数です。

ラプラス変換やフーリエ変換は組込関数としては用意されていませんが、標準で添付されているパッケージを読み込むことで利用できます。パッケージを読み込むには以下のようにします (あるいは `Function Browser` から読み込むこともできます)。

```
In[19]:= Needs["Calculus`LaplaceTransform`"]
```

`Calculus`LaplaceTransform`` がラプラス変換のためのパッケージの名前です。‘はシングル・クォーテーションではなくバック・クォーテーションですので注意して下さい。

```
In[20]:= LaplaceTransform[ t^2 E^(a t), t, s ]
```

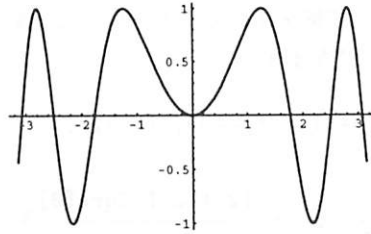
$$Out[20]= \frac{2}{(-a + s)^3}$$

どのようなパッケージがあるかは第 8 節で説明します。

#### 4.4 2次元グラフィックス

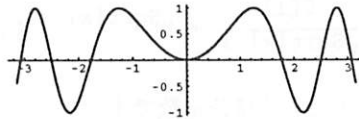
まず簡単なグラフを書かせてみましょう。  $\sin x^2$  のグラフを  $x$  が  $-\pi$  から  $\pi$  の範囲でプロットします。

```
In[21]:= Plot[ Sin[x^2], {x,-Pi,Pi} ]
```



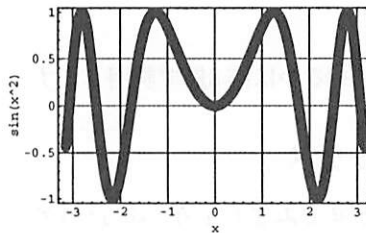
Plotにはさまざまなオプションを指定できます。たとえば図の縦横比を1対1にするには、AspectRatio オプションを Automatic と指定します。

```
In[22]:= Plot[ Sin[x^2], {x,-Pi,Pi}, AspectRatio->Automatic ]
```



どのようなオプションが指定できるかは、??Plotにより表示されます。ためしに色々なオプションを指定してみましょう。

```
In[23]:= Plot[ Sin[x^2], {x,-Pi,Pi},
    Axes->None, (* 軸を消す *)
    Frame->True, (* 枠を付ける *)
    FrameLabel->{"x","sin(x^2)"}, (* ラベルを付ける *)
    GridLines->Automatic, (* グリッドを付ける *)
    PlotStyle->{{Thickness[0.03]}} (* 線の太さの指定 *)
]
```



その他、以下のようなプロットが利用できます。

- 離散データのプロット

ListPlot, MultipleListPlot (Graphics'MultipleListPlot'パッケージ)



- パラメータ表示の関数のプロット  
ParametricPlot
- 極座標表示の関数のプロット  
PolarPlot (Graphics'Graphics'パッケージ)
- 陰関数のプロット  
ImplicitPlot (Graphics'ImplicitPlot'パッケージ)
- 棒グラフやパイチャート  
BarChart, PieChart (Graphics'Graphics'パッケージ)

実は *Mathematica* では、任意の 2 次元図形を表示することが簡単にできます。例えば、なつかしいスマイル・バッジ (でしたっけ?) を書いてみましょう。

```
In[24]:= Show[Graphics[{
    Circle[{0,0}, 1],
    Disk[{0.2,0.1}, {0.05,0.1}],
    Disk[{-0.2,0.1}, {0.05,0.1}],
    Circle[{0,0}, 0.7, {-0.85Pi,-0.15Pi} ]}],
    AspectRatio->Automatic ]
```



`Circle[{x, y}, r]` は、中心が  $(x, y)$ 、直径  $r$  の円を描きます。 `Circle[{x, y}, {rx, ry}]` ならば楕円を、 `Circle[{x, y}, r, {θ1, θ2}]` ならば角  $\theta_1$  から  $\theta_2$  の間の弧を描きます。 `Disk` も `Circle` と同様ですが、中を塗りつぶします。 `Circle` や `Disk` のほかには、直線を描く `Line`、点を描く `Point`、多角形を描く `Polygon`、塗りつぶされた矩形を描く `Rectangle`、テキストを描く `Text` などがあります。また、直線の太さを指定する `Thickness`、点線を指定する `Dashing`、点の大きさを指定する `PointSize`、グレーレベルを指定する `GrayLevel` なども利用できます。

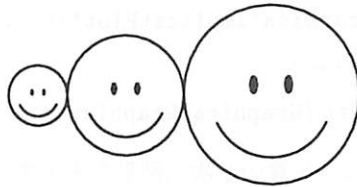
スマイル・バッジをたくさん書いてみましょう。スマイル・バッジの中心座標と大きさを指定できる関数を定義して、それを使って大きさの異なる三つのスマイルを並べます。

```
In[25]:= smiley[xy_:{0,0}, r_:1] := {
    Circle[xy, r],
```

```

Disk[xy+{ 0.2,0.1}r, {0.05,0.1}r],
Disk[xy+{-0.2,0.1}r, {0.05,0.1}r],
Circle[xy, 0.7r, {-0.85Pi,-0.15Pi}]
};
In[26]:= Show[ Graphics[Table[smiley[{i^2,0},i], {i,3}] ],
AspectRatio->Automatic ]

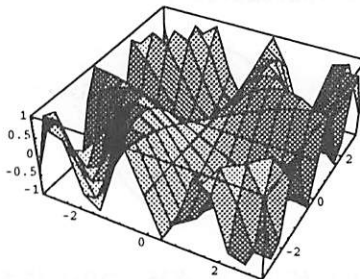
```



#### 4.5 3次元グラフィックス

3次元グラフィックスも2次元グラフィックスと同様に簡単にプロットできます。まずは  $z = \sin(xy)$  のプロットです。

```
In[27]:= Plot3D[ Sin[x y], {x,-Pi,Pi}, {y,-Pi,Pi} ]
```

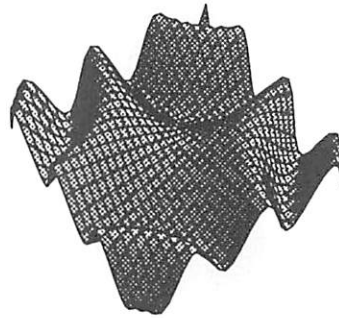


さらに、PlotPoints オプションで計算する点(デフォルトは  $15 \times 15$ )を増やし、ViewPoint オプションで視点の座標を指定してみます。

```

In[28]:= Plot3D[ Sin[x y], {x,-Pi,Pi}, {y,-Pi,Pi},
PlotPoints->30, (* 30 x 30 点で計算 *)
Axes->False, (* 軸を消す *)
Boxed->False, (* 枠を消す *)
ViewPoint->{1.5, -1, 3} (* 視点を指定 *)
]

```



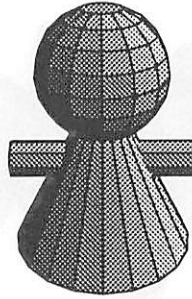
その他、以下の表示も可能です。

- 等高線表示: `ContourPlot`
- 濃度表示: `DensityPlot`
- パラメータ表示の関数: `ParametricPlot3D`
- ベクトル場: `PlotVectorField3D` (`Graphics`PlotField3D``パッケージ)

また、2次元グラフィックスと同様に任意の図形を簡単に表示できます。3次元のグラフィック原始要素としては、線を描く `Line`、点を描く `Point`、立方体を描く `Cuboid`、多角形を描く `Polygon`、テキストを描く `Text` があります。もう少し複雑な図形を描くためには `Graphics`Shapes`` パッケージがあります。これを使って、手のあるてるてる坊主(?) を書いてみましょう。

```
In[29]:= Needs["Graphics`Shapes`"]
```

```
In[30]:= Show[Graphics3D[{  
    TranslateShape[Sphere[0.7], {0,0,1}],  
    Cone[],  
    RotateShape[Cylinder[0.2,1,8], 0, Pi/2, 0] }],  
    Boxed->False, ViewPoint->{2,0,1}  
]
```



## 5 実際的な利用の例

ここでは実際的な利用の例として、ファイル (ファイル名は `data`) からデータを読み込み、処理を行う方法を紹介します。

### 5.1 データの読み込み

まず、ファイル `data` の内容を確認してみます。ファイルの内容は `!!data` で表示できます。

```
In[1]:= !!data
0.1297323638545699
0.1323127172826521
0.1446228546294113
.....
```

大きなファイルの場合、次のようにすれば最初の 10 行だけを表示できます (Unix を利用している場合)。

```
In[2]:= ColumnForm[ReadList["!head -10 data", String]]
```

一般に、`ColumnForm[ReadList["!command", String]]` によって、Unix コマンドの実行結果を表示できます。`ReadList` はファイルからデータを読み込むための関数ですが、ファイル名が `!` から始まっている場合は、以降を Unix のコマンドと解釈して、その実行結果を読み込みます (読み込んだ結果はリストとして返されます)。`String` は各行を文字列として読み込むための指定です。`ColumnForm` によりリストの要素を縦に並べて表示します。

Unix コマンドをしばしば実行する場合は、次のような関数を定義しておくくと便利です。

```
In[3]:= Unix[command_String] :=
Scan[Print, ReadList["!"<command>"|expand", String]]
```

```
In[4]:= Unix["wc -w data"]
128 data
```

ファイルからデータを読み込むには `ReadList` 関数を使います。

```
In[5]:= data = ReadList["data", Number];
```

`ReadList` は、ファイル `data` から数値データを読み込み、それをリストにして返しますので、そのリストを変数 `data` に代入しています。

各行に複数のデータがある場合の例です。次のようなファイルがあるとします。

```
In[6]:= !!data2
1 2 3
4 5
6 7 8
```

上と同じように `ReadList` を実行すると、単にすべての数値データを一列にならべたリストが返ってきます。 `RecordLists` オプションを指定すれば各行をリストにまとめることができます。

```
In[7]:= data2 = ReadList["data2", Number]
Out[7]= {1, 2, 3, 4, 5, 6, 7, 8}
In[8]:= data2 = ReadList["data2", Number, RecordLists->True]
Out[8]= {{1, 2, 3}, {4, 5}, {6, 7, 8}}
```

次のようにデータがコンマで区切られている場合はもう少しやっかいです。この場合、データの種別を `Word` と指定して、一旦データを文字列として読み込んで、`ToExpression` 関数により数値に変換する必要があります。

```
In[9]:= !!data3
1, 2, 3
4, 5
6, 7, 8
In[10]:= data3 = ToExpression[
ReadList["data3", Word, RecordLists->True,
WordSeparators->{" ", "\t", ","}]]
Out[10]= {{1, 2, 3}, {4, 5}, {6, 7, 8}}
```

読み込んだデータを表示してみます。

```
In[11]:= data
Out[11]= {0.1297323638545699, 0.1323127172826521,
0.1446228546294113, 0.183239867688719,
.....
0.02034455129999835, 0.1183108049642118}
```

関数 `Short` をもちいれば、簡略化した表示をします。

```
In[12]:= data // Short
```

```
Out[12]= {0.1297323638545699, <<126>>, 0.1183108049642118}
```

`expr // func` は、`func[expr]` と全く同じことです。上の場合は、`Short[data]` と同じことになります。

常に簡略化した表示にしたければ

```
In[13]:= $PrePrint = Short
```

とします。次のようにすれば解除できます

```
In[14]:= $PrePrint =.
```

読み込んだデータの平均値を計算してみましょう。

次のように `Statistics`DescriptiveStatistics`` パッケージを利用しても求められますが、直接求めるのも簡単にできます。

```
In[15]:= Needs["Statistics`DescriptiveStatistics`"]
```

```
In[16]:= Mean[data] (* パッケージを利用 *)
```

```
Out[16]= 0.397853
```

```
In[17]:= Plus @@ data / Length[data] (* 直接求める *)
```

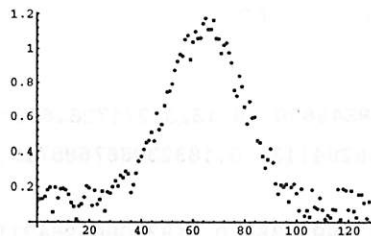
```
Out[17]= 0.397853
```

`Plus @@ data` は、`Apply[Plus, data]` と同じです。`Apply[f, { $x_1, x_2, \dots, x_n$ }]` は、 $f[x_1, x_2, \dots, x_n]$  を求めます。つまり、今の例では `data` の要素の和が求められます。`Plus[data]` ではダメなのに注意して下さい。`Length[data]` は、リスト `data` の長さを返します。つまり、今の例では `data` の要素の個数が求められます。

## 5.2 データのプロット

では、読み込んだデータをプロットしてみます。

```
In[18]:= ListPlot[data]
```



関数 `ListPlot` は、リスト  $\{y_1, y_2, \dots, y_n\}$  が与えられたとき、すべての点  $(i, y_i)$  をプロットします。

実際の  $x$ -座標が  $1, 2, 3, \dots$  なら良いのですが、そうでない場合は  $x$  座標と  $y$  座標の対のリスト  $\{\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_n, y_n\}\}$  を `ListPlot` に与えてやる必要があります。今、 $x_i = x_0 + (i-1)dx$  だとすると、 $x$ -座標なしのリストから  $x$ -座標付きのリストへの変換は以下のようにしてできます。 `Table[expri, {i, n}]` は、リスト  $\{expr_1, expr_2, \dots, expr_n\}$  を返す関数です。 `data[[i]]` は、リスト `data` の第  $i$  要素を表します。

```
In[19]:= Table[{x0+(i-1)dx, data[[i]]}, {i, Length[data]}
```

```
Out[19]= {{x0, 0.1297323638545699},
           {dx + x0, 0.1323127172826521},
           {2 dx + x0, 0.1446228546294113},
           ....}
```

しかし、次の方法のほうがエレガントです。

```
In[20]:= Transpose[{x0+dx Range[0,Length[data]-1], data}]
```

まず、関数 `Range` により、リスト `data` と同じ長さの  $0$  から始まるリストを作成し、各要素を  $dx$  倍して  $x_0$  を加えています。それと `data` を並べたリストの転置行列が求める結果です。

これを関数として定義してみましょう。

```
In[21]:= AddXcoord[list_, x0_, dx_] :=
           Transpose[{x0+dx Range[0,Length[list]-1], list}]
```

関数の定義は `function_pattern := function_body` のように、`:=` を用います (=は通常は使いません)。また `function_pattern` 中の変数には必ず下線を付けてください。

`:=` と `=` の違いは、右辺を評価するかどうかの違いです。 `=` は右辺を評価した結果を左辺に代入します。 `:=` は右辺を評価せずに左辺に代入します。関数が実行されたときに初めて右辺を評価します。以下の例を見て下さい。

```
In[22]:= a = 10
In[23]:= f1[x_] := x+a (* x+a が代入されます *)
In[24]:= f2[x_] = x+a (* x+10 が代入されます *)
In[25]:= a = 20 (* a を 20 に変更 *)
In[26]:= f1[x]
Out[26]= 20 + x (* x+a を評価した結果 *)
In[27]:= f2[x]
Out[27]= 10 + x
```

引数のデフォルト値を与えることもできます。AddXcoordで引数 dx が与えられなければ 1 とするには、以下のようにします。

```
In[28]:= AddXcoord[list_, x0_, dx_:1] :=  
          Transpose[{x0+dx Range[0,Length[list]-1], list}]  
In[29]:= AddXcoord[{10,11,12}, 1]  
Out[29]= {{1, 10}, {2, 11}, {3, 12}}
```

x-座標が-2 から 1/32 きざみだとして、実行してみます。

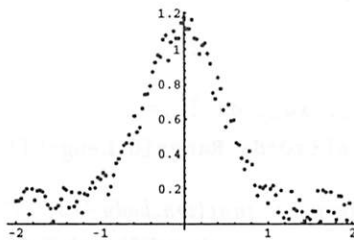
```
In[30]:= datax = AddXcoord[data, -2, 1./32]  
Out[30]= {{-2, 0.1297323638545699},  
          {-1.96875, 0.1323127172826521},  
          {-1.9375, 0.1446228546294113},  
          .....}
```

x-座標を取り除くのは簡単です。

```
In[31]:= Transpose[datax][[2]]
```

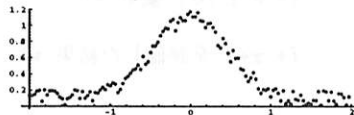
では、x-座標をつけ加えたデータ datax をプロットしてみましょう。

```
In[32]:= ListPlot[datax]
```



軸がグラフの真ん中にあるのが気になりますので、AxesOrigin->{-2,0}で軸の原点座標を指定します。ついでに AspectRatio->Automaticで、x-座標と y-座標のきざみを同一にします (Mathematicaは通常、グラフの外形を自動的に整形し、黄金分割比を持つ横長の長方形になるようにします)。

```
In[33]:= ListPlot[datax, AxesOrigin->{-2,0}, AspectRatio->Automatic]
```





## 5.3 最小 2 乗法

### 5.3.1 Fit

データを最小 2 乗法で多項式近似するには、関数 `Fit` を用います。ここでは、`datax` を  $a + bx^2 + cx^4 + dx^6$  という形の式に当てはめ、求めた多項式を関数 `fit1` と定義します。

```
In[34]:= fit1[x_] = Fit[datax, {1,x^2,x^4,x^6}, x]
```

```
Out[34]=
```

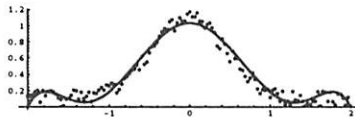
$$1.034 - 1.39553 x^2 + 0.628783 x^4 - 0.0861296 x^6$$

一般の 6 次多項式で近似するには、`Fit` の第 2 引数を  $\{1, x, x^2, x^3, x^4, x^5, x^6\}$  または `Table[x^i, {i, 0, 6}]` としてください。

求めた多項式は `Plot[fit1[x], {x, -2, 2}]` でプロットできますが、ここでは、もともとのデータ `datax` のプロットと重ねて表示してみます。複数のプロットを重ねて表示するには、`Graphics'Graphics'` パッケージにある `DisplayTogether` 関数を使います。

```
In[35]:= Needs["Graphics'Graphics'"] (* パッケージの読み込み *)
```

```
In[36]:= DisplayTogether[
    ListPlot[datax],
    Plot[fit1[x], {x, -2, 2}],
    AxesOrigin->{-2, 0}, AspectRatio->Automatic
]
```



次のようにしても複数のプロットを重ねて表示できます。

```
In[37]:= graph1 = ListPlot[datax, DisplayFunction->Identity];
In[38]:= graph2 = Plot[fit1[x], {x, -2, 2}, DisplayFunction->Identity];
In[39]:= Show[graph1, graph2, AxesOrigin->{-2, 0},
    AspectRatio->Automatic, DisplayFunction:> $DisplayFunction]
```

`DisplayFunction->Identity` はプロットを表示しないための指定です。あとから `Show` のなかで `DisplayFunction :> $DisplayFunction` とすれば表示できます。

以降、たびたび重ねた表示を行うので、関数 `plotboth` として定義しておきます。`plotboth[fit1]` で上と同じプロットが得られます。ついでに 2 乗誤差を求める関数 `squareError` も定義しておきましょう。

```
In[40]:= plotboth[f_] := Module[{x},
  DisplayTogether[
    ListPlot[datax],
    Plot[f[x], {x,-2,2}],
    AxesOrigin->{-2,0}, AspectRatio->Automatic
  ]]
```

```
In[41]:= squareError[f_] := Module[{i},
  Sum[N[(datax[[i,2]]-f[datax[[i,1]]])^2],
    {i,1,Length[datax]}]
]
```

```
In[42]:= squareError[fit1]
```

```
Out[42]= 0.743927
```

`Module[{ $x_1, x_2, \dots, x_n$ }, body]` は、局所変数  $x_1, x_2, \dots, x_n$  を宣言します (局所変数とは *body* の中だけで利用する変数です)。plotboth で Module を使わなかった場合、変数  $x$  はグローバル変数となり、 $x$  に値が代入されているときに正しく動きません。

### 5.3.2 NonlinearFit

多項式では、あまり良い近似が得られませんでした。比線形の式で近似を行う場合には `Statistics`NonlinearFit`` パッケージが使えます。

```
In[43]:= Needs["Statistics`NonlinearFit`"] (* パッケージの読み込み *)
```

```
In[44]:= fit2a[x_] = a + b Exp[-(c x)^2]; (* あてはめる式を定義 *)
```

```
In[45]:= NonlinearFit[datax, fit2a[x], x, {a,b,c}]
```

```
Out[45]= {a -> 0.11288, b -> 1.01213, c -> 1.57377}
```

```
In[46]:= fit2[x_] = fit2a[x] /. % (* これが求める近似 *)
```

```
In[47]:= plotboth[fit2] (* プロットは省略 *)
```

```
In[48]:= squareError[fit2]
```

```
Out[48]= 0.422073
```

`NonlinearFit` は、`Solve` などと同様に解を規則の形で返しますので、`/.` を使って規則を適用した結果を求める必要があります。

`NonlinearFit` を使わなくても、 $y$ -座標データの `Log` を計算すれば、`Fit` でももう少し良く近似できます。

```
In[49]:= dataxLog = Map[{#[[1]], Log#[[2]]}&, datax];
```

```
In[50]:= fit3[x_] = Exp[Fit[dataxLog, {1,x,x^2,x^3,x^4}, x]]
```

```
In[51]:= plotboth[fit3] (* プロットは省略 *)
In[52]:= squareError[fit3]
Out[52]= 0.5461
```

### 5.3.3 FFT

最後に離散型フーリエ変換で近似してみましょう。まず次のような関数を定義します(説明は省略)。

```
In[53]:= FitFFT[data_, {x_,x0_:0,dx_:1}, m_] := Module[{n,f,k},
  n = Length[data];
  f = N[InverseFourier[data] / Sqrt[n]];
  Re[f[[1]]] +
  Sum[ 2 Re[f[[k+1]]] Cos[2 Pi k (x-x0) / (n dx)] -
      2 Im[f[[k+1]]] Sin[2 Pi k (x-x0) / (n dx)] ,
      {k,1,m} ]
]
```

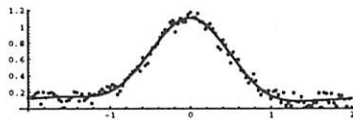
3次まで近似した式を求めます。

```
In[54]:= fit4[x_] = FitFFT[data, {x,-2,1/32}, 3]
```

この式は次のようになります。

$$\begin{aligned}
 0.397853 & - 0.442581 \cos\left(\frac{\pi(2+x)}{2}\right) & + 0.0272171 \sin\left(\frac{\pi(2+x)}{2}\right) \\
 & + 0.216662 \cos(\pi(2+x)) & + 0.0056861 \sin(\pi(2+x)) \\
 & - 0.0506513 \cos\left(\frac{3\pi(2+x)}{2}\right) & + 0.00208212 \sin\left(\frac{3\pi(2+x)}{2}\right)
 \end{aligned}$$

```
In[55]:= plotboth[fit4]
In[56]:= squareError[fit4]
Out[56]= 0.378544
```



## 5.4 データの書き出し

最後に、データをファイルに書き出すプログラムを紹介します。Mathematicaだけで利用するデータの場合は、以下のようにして簡単にファイルに保存できます。

```

In[57]:= datax >> "tmp1"          (* ファイル tmp1 に保存 *)
In[58]:= newdatax = << "tmp1"    (* ファイル tmp1 から読み込み *)
In[59]:= datax == newdatax
Out[59]= True                    (* datax と newdatax は等しい *)

```

ただし、tmp1 には *Mathematica* の形式で書き込まれるので、*Mathematica* 以外のプログラムで取り扱うのは面倒です。

```

In[60]:= !!tmp1
{-2, 0.1297323638545699}, {-1.96875, 0.1323127172826521},
{-1.9375, 0.1446228546294113}, {-1.90625, 0.183239867688719},
.....

```

このような時は、次のプログラムを利用すればうまくいくと思います(説明は省略)。

```

In[61]:= WriteArray[file_, array_List] := Module[{stream},
    stream = OpenWrite[file, PageWidth->Infinity];
    Scan[WriteToStream[stream, #]&, array];
    Close[stream];
]
WriteToStream[stream_, list_List] := Block[{},
    If[ Length[list] > 0,
        WriteString[stream, First[list]];
        Scan[WriteString[stream, "\t", #]&, Rest[list]]
    ];
    WriteString[stream, "\n"];
]
WriteToStream[stream_, x_] := Write[stream, x]

```

たとえば、datax は以下のような形式で保存されます。

```

In[62]:= WriteArray["tmp2", datax]
In[63]:= !!tmp2
-2 0.1297323638545699
-1.96875 0.1323127172826521
-1.9375 0.1446228546294113
.....

```

## 6 Mathematicaシステムの概要

### 6.1 フロントエンドとカーネル

*Mathematica*のシステムは、フロントエンドとカーネルの二つのプログラムから成り立っています。

フロントエンドは、入力編集、グラフィック表示などを行うプログラムで、通常 *Mathematica* を利用する場合は、フロントエンドを通して行います。ノートブック(後述)の表示/作成/編集を行うのもこのフロントエンドです。

カーネルは、*Mathematica*の計算を実行するプログラムです。フロントエンドを立ち上げた時点では、カーネルプログラムはまだメモリーにロードされていません。フロントエンドから何か計算式を実行しようとした時点(つまり Enter や Shift+Return を入力した時点)で、カーネルがロードされます。

カーネルを別のマシンに切り替えることも可能です(リモートカーネルといいます)。たとえば、フロントエンドは Macintosh, カーネルは Unix ワークステーションという使い方もできるわけです。

### 6.2 ライセンスサーバ

次に *Mathematica*のライセンス認証方式について説明します。

*Mathematica*のシステムには、同時に一人しか *Mathematica*を利用できないシングルユーザライセンスのものと、複数のユーザが同時に利用できるマルチユーザライセンスのものがあります。パーソナルコンピュータ用のシステムは通常シングルユーザライセンスであり、Unix ワークステーション用のシステムは通常マルチユーザライセンスです。マルチユーザライセンスの場合、何人(正確には何プロセス)までが同時に利用できるかは、購入した *Mathematica*のシステムにより異なります。神戸大学で配布しているパッケージは 99 プロセスまでが可能です。

マルチユーザライセンスにおいて、プロセス数の管理を行っているのは、ライセンスサーバと呼ばれるプログラムです(SunOS の場合、`/usr/local/math/Install/mathserver`)。ライセンスサーバが、*Mathematica*のプロセス数を把握し、ライセンス数以上の *Mathematica*が起動されないように管理しています。*Mathematica*のカーネルとライセンスサーバは必ずしも同じマシン上で動く必要はありません。その場合、カーネルを動かすマシンの方をクライアントと呼びます。

実はサーバは、プロセス数の管理だけでなく、パスワードに基づいてライセンスの認証も行います。クライアントが正しいパスワードを知っているかどうかを調べるわけです。この場合、クライアントが知っている必要があるのはサーバのパスワードです。サーバのパスワードを `mathpass` ファイル(SunOS の場合、`/usr/local/`

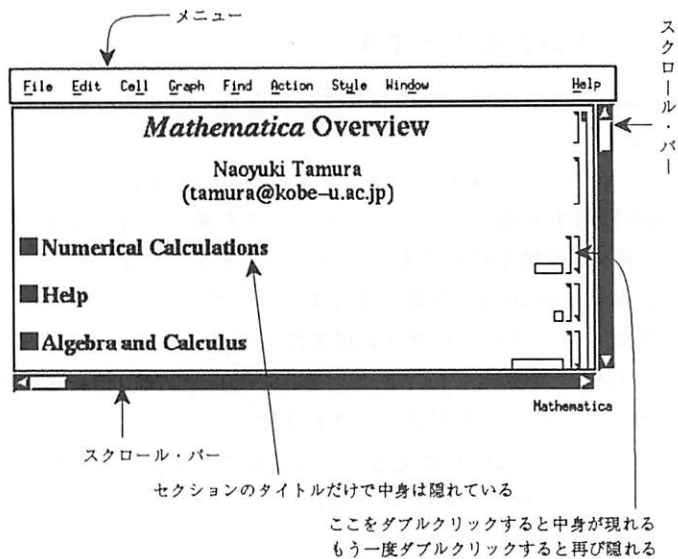


図 1: ノートブックの例

math/Install/mathpass) に記述しておきます。通常の使い方では、研究室 (あるいは学科) 単位にライセンスサーバが一台あれば十分でしょう。

## 7 ノートブック

フロントエンドを用いている場合、*Mathematica*の実行結果をノートブック (図 1 参照) としてファイルにセーブすることができます (拡張子は .ma)。通常、ノートブックには図も含めてすべての結果が保存されるためファイル容量はかなり大きくなります。Save As Special メニューで、Include formatted text をオフに、Include displayed form only をオンにすれば、図などの実行結果を含めずにセーブできます。

*Mathematica*を一旦終了してから、ノートブックを開いた場合、変数値や関数定義はすべて未定義の状態から始まります。したがって、ノートブックを開いたのち必要な入力行に対して、Enter キーでその行を再度実行する必要があります。すべての入力行を再度実行するには Evaluate Notebook が便利です。

## 8 パッケージ

*Mathematica*には多くの組込関数がありますが、それ以外にも多数の関数が標準で用意されています。これらの関数はパッケージと呼ばれるファイル (拡張子は .m) に用意されており、Needs 関数で読み込むことができます (パッケージファイルのディレクトリは SunOS では /usr/local/math/Packages です)。標準のパッケージ名は以下の通りです。

- Algebra  
CountRoots.m, Master.m, PolynomialMod.m, Relm.m, SymbolicSum.m, Trigonometry.m
- Calculus  
Common/DSolveLog.m, Common/Support.m, DSolve.m, DiracDelta.m, EllipticIntegrate.m, FourierTransform.m, LaplaceTransform.m, Limit.m, Master.m, PDSolve1.m, Pade.m, VariationalMethods.m, VectorAnalysis.m
- DiscreteMath  
CombinatorialFunctions.m, CombinatorialSimplification.m, Combinatorica.m, ComputationalGeometry.m, Master.m, Permutations.m, RSolve.m, Tree.m
- Examples  
CellularAutomata.m, Collatz.m, FileBrowse.m, IntegerRoots.m, Life.m, OneLiners.m, OptionUtilities.m, StringPatterns.m
- Geometry  
Master.m, Polytopes.m, Rotations.m
- Graphics  
Animation.m, ArgColors.m, Arrow.m, Colors.m, Common/GraphicsCommon.m, ComplexMap.m, ContourPlot3D.m, FilledPlot.m, Graphics.m, Graphics3D.m, ImplicitPlot.m, Legend.m, Master.m, MultipleListPlot.m, ParametricPlot3D.m, PlotField.m, PlotField3D.m, Polyhedra.m, Shapes.m, Spline.m, SurfaceOfRevolution.m, ThreeScript.m
- LinearAlgebra  
Cholesky.m, CrossProduct.m, GaussianElimination.m, Master.m, MatrixManipulation.m, Orthogonalization.m, Tridiagonal.m
- Miscellaneous  
Audio.m, Calendar.m, ChemicalElements.m, CityData.m, Geodesy.m, Master.m, Music.m, PhysicalConstants.m, SIUnits.m, Units.m, WorldData.m, WorldNames.m, WorldPlot.m
- NumberTheory  
Binomial.m, ContinuedFractions.m, FactorIntegerECM.m, Master.m, NumberTheoryFunctions.m, PrimeQ.m, Ramanujan.m, Rationalize.m, Recognize.m
- NumericalMath  
Approximations.m, BesselZeros.m, Butcher.m, CauchyPrincipalValue.m, ComputerArithmetic.m, GaussianQuadrature.m, InterpolateRoot.m, IntervalAnalysis.m, IntervalArithmetic.m, ListIntegrate.m, Master.m, Microscope.m, NLimit.m, NewtonCotes.m, PolynomialFit.m, SplineFit.m
- ProgrammingExamples  
AlgExp.m, ArgColors.m, Atoms.m, BookPictures.m, Collatz.m, ComplexMap.m, Com-

plexTest.m, Constants.m, ExpandBoth.m, Fibonacci1.m, FilterOptions.m, GetNumber.m, MakeFunctions.m, MyReadList.m, Newton.m, Numerical.m, OddEvenRules.m, OptionUse.m, ParametricPlot3D.m, PrimePi.m, PrintTime.m, RandomWalk.m, ReadLoop1.m, RungeKutta.m, SessionLog.m, ShowTime.m, Skeleton.m, SphericalCurve.m, Struve.m, SwinnertonDyer.m, Tensors.m, Transcript.m, TrigDefine.m, TrigSimplification.m, Until.m, VectorCalculus.m, WrapHold.m

- Statistics

Common/DistributionsCommon.m, Common/FitCommon.m, Common/HypothesisCommon.m, ConfidenceIntervals.m, ContinuousDistributions.m, DataManipulation.m, DescriptiveStatistics.m, DiscreteDistributions.m, HypothesisTests.m, InverseStatisticalFunctions.m, LinearRegression.m, Master.m, MovingAverage.m, NonlinearFit.m, NormalDistribution.m

- Utilities

BinaryFiles.m, DXF.m, FilterOptions.m, Language.m, Master.m, MemoryConserve.m, Package.m, ShowTime.m

パッケージ一覧は Function Browser で見るすることができます。また各パッケージの詳しい説明については、マニュアル“Guide to Standard Mathematica Packages”を見て下さい(このマニュアルは Wolfram に書籍として注文することも可能です)。

Mathematica標準のパッケージ以外にも、数多くのパッケージが公開されています。これらのパッケージは Wolfram により MathSourceとよばれるアーカイブにまとめられており、インターネットを通じて、あるいはCD-ROMで入手できます(詳細は次節)。

市販のパッケージとしては以下のものがあります。

- **Fuzzy Logic Pack:** Package for creating, modifying, and visualizing fuzzy sets and fuzzy-logic based systems
- **Mechanical Systems Pack:** Package with specialized Mathematica tools and examples for analyzing and designing rigid body systems
- **Electrical Engineering Pack:** Package with specialized Mathematica functions for electrical engineering tasks
- **Finance Pack:** Specialized package for financial professionals
- **Time Series Pack:** Package for analyzing time-dependent data
- **Optica:** Optic system design package
- **MathLive:** Real-time 3-D graphics package
- **Nodal:** Circuit design and analysis package
- **MathTensor:** Tensor analysis package
- **TSiDynamics:** Multibody modeling package
- **TSiControls:** Control systems design and analysis package
- **InterCall:** Package that provides access to different routines
- **Descartes-Geometry2D:** Exact drawing and symbolic package
- **Energy Worker:** Thermal and energy package
- **OPERA:** Options pricing analysis package
- **Help Stack:** On-line Mathematica help



- CalcE: Two-dimensional graphics enhancement package
- CARTAN: Tensor component package
- Calculus&Mathematica : Calculus courseware package
- Interactive Calculus: Dynamic multimedia textbook for calculus
- Joy of Mathematica: “Point and click” interface package
- Calculus Using Mathematica: Calculus courseware package
- Mathematica Labs for Calculus Instruction: Calculus instruction package
- MathLink for Excel and Mathematica: Package for linking Excel and Mathematica
- MathLink for Spyglass Transform: Package for linking Spyglass Transform and Mathematica
- MathLink for Xmath: Package for linking Xmath and Mathematica
- MathLink for MATLAB: Package for linking MATLAB and Mathematica
- Other MathLink-enabled applications: Other packages that link to Mathematica

## 9 Mathematicaの情報

[WWW]

Mathematicaについて最新の情報を得るには、WWW (World Wide Web) が最も良い方法です。

Wolfram による Mathematica ホームページの URL は <http://www.wri.com/> です (図 2 参照)。MathSource の検索 (後述) や、Mathematica のデモ、Mathematica や T シャツの抽選のページ (!) などがあります。

神戸大学内にもホームページを作成しました (<http://bach.seg.kobe-u.ac.jp/mma/>)。次のような内容を入れてありますので、ぜひご利用下さい。

- Mathematica 入門 (総合情報処理センター 夏季講習会資料)
- Mathematica 組込関数一覧
- MathSource の検索サービス
- Mathematica 関連書籍リスト
- 1994 年 Mathematica 講演会資料
- Mathematica の入手方法
- 神戸大学 Mathematica メーリングリスト

# Wolfram Research

Makers of *Mathematica*.

Wolfram Research, Inc. • Wolfram Research Europe Ltd. • Wolfram Research Asia Ltd.

[Mathematica Products Information](#)

*Mathematica* applications, publications, and technical and ordering information

[Schools Direct](#)

News releases, events calendar, Solving the Quintic with *Mathematica* <sup>NEW</sup>, and back issues of *MathUser*

[Mathematica Demo](#) <sup>NEW</sup>

An interactive demonstration of some of *Mathematica*'s capabilities

[Mathematica on the Web](#)

*Mathematica* technical support service <sup>NEW</sup>, training, user groups, mailing lists, and other customer services

[MathSource CD-ROMs Library](#)

An on-line version of the most comprehensive collection of *Mathematica* packages, notebooks, examples, and programs available today; also find out how to order the *MathSource* collection on CD-ROM

[Mathematica in Education](#)

*Mathematica* courseware directory, flexible academic purchase programs, special student version, and new version available for high schools

[Science, Research, and Academic Software](#)

Sophisticated mathematical, scientific, and technical images made by *Mathematica* users around the world

[Wolfram Research History](#)

Company history, behind-the-scenes photographic tour, job openings, unofficial employee home pages, and contact information

[Free Copies of Mathematica](#)

Download free *Mathematica* notebook reader and sample notebooks

[Training Tours](#) <sup>NEW</sup>

Become a *Mathematica* expert with these one-day training courses in Boston, New York, and Washington DC!

[Mathematica Conferences Conferences](#) <sup>NEW</sup>

Don't miss this opportunity to exchange ideas with other advanced users and meet Wolfram Research staff.

[Mathematica Merchandise Store](#)

High-quality *Mathematica*-related merchandise, including T-shirts, cap, jacket, mug, *MathSource* on CD-ROM, and Solving the Quintic poster

[Register to Win a Copy of Mathematica](#) <sup>NEW</sup>

Entry form and rules for monthly giveaway contest

[Mathematica Product Store](#)

Links to related sites on the Web

[Mathematica and Mathematica 1.0](#)

Compatibility information and installation instructions



This page was last modified on: 20 Aug 1995 at 18:48  
Copyright © 1995, 1996 Wolfram Research, Inc.

☒ 2: *Mathematica*のホームページ (<http://www.wri.com/>)

### [MathSource]

*MathSource*は *Mathematica*上の多数の応用プログラム・ライブラリです。以下の方法で入手や検索ができます。

- CD-ROM: \$45
- FTP: [mathsource.wri.com](http://mathsource.wri.com)  
国内では [ftp.osaka-u.ac.jp](http://ftp.osaka-u.ac.jp) がミラーサイトになっていますので、こちらを利用して下さい (つながらないことも多いようですが…).
- Gopher: [mathsource.wri.com](http://mathsource.wri.com)
- WWW: <http://www.wri.com/mathsource/>
- WWW: <http://bach.seg.kobe-u.ac.jp/mma/MSsearch.html>

### [インターネット・ニュース]

*Mathematica*についてのインターネット・ニュース・グループとしては次のものがあります。

- [comp.soft-sys.math.mathematica](mailto:comp.soft-sys.math.mathematica) (後述のメーリングリストの内容も流れているようです)
- [fj.sci.math](mailto:fj.sci.math), [fj.lang.misc](mailto:fj.lang.misc) (日本語のニュースグループです。ごくたまに *Mathematica*についての記事が流れています)

CompuServe にも *Mathematica*のフォーラムがあるようです (GO WOLFRAM).

### [メーリング・リスト, ユーザーズ・グループ]

メーリング・リストとしては以下のものがあります。

- Wolfram によるもの (申込先: [MathGroup@wri.com](mailto:MathGroup@wri.com))
- 神戸大学内のもの (申込先: [Mathematica-admin@godel.seg.kobe-u.ac.jp](mailto:Mathematica-admin@godel.seg.kobe-u.ac.jp))  
神戸大学の *Mathematica*ユーザの人はぜひ参加してください

### [書籍]

数多くの書籍が出版されています。以下の URL で、書籍の一覧を見ることができます。

<http://www.wri.com/mathematica/books/index.html>

<http://bach.seg.kobe-u.ac.jp/mma/publist.html>

日本語で書かれた *Mathematica* の書籍については参考文献をご覧ください。

定期刊行物としては、*The Mathematica Journal* (Miller Freeman), *Mathematica in Education* (TELOS/Springer-Verlag), *Mathematica World* (フロッピーが送付されてくるそうです。申込先は [mathematica@matilda.vut.edu.au](mailto:mathematica@matilda.vut.edu.au)), *MathUser* (Wolfram) があります。

## 謝辞

キャンパス・ライセンス取得により、神戸大学の *Mathematica* ユーザの数も本当に多くなってきました (メーリングリストへの参加者だけで現在 83 名です)。

事務的には非常に面倒な処理にもかかわらず、キャンパス・ライセンスの取得に御尽力いただいた皆様にこの場を借りて感謝いたしたいと思います。特に、工学部の田中克己先生、松田秀雄先生 (現在大阪大学)、瀬良昌憲先生が最初の導入作業を担当されたとうかがっております。今年度からは、総合情報処理センターの福島徹先生、樽磨和幸先生、大伴登業務掛長の御努力により、総合情報処理センターで事務処理を行っていただくことになりました。感謝いたします。

メーリングリストの参加者の方々も、インストール作業の分担やマニュアルの整備などに御協力いただきました。ありがとうございました。

最後になりますが、神戸大学 (および私) に *Mathematica* を紹介していただいた伊藤利明先生 (現在徳島大学)、蔡東生先生 (現在筑波大学) に感謝いたしたいと思います。

## 参考文献

- [1] S. ウルフラム (白水重明訳). *Mathematica: A System for Doing Mathematics by Computer*. アジソン・ウエスレイ, 第 2 版, 3 月 1992.
- [2] T. W. グレイ, J. グリン (榎原進訳). *Mathematica ビギナーズガイド*. トッパン, 2 月 1994. (原書名: *The beginner's guide to Mathematica version 2*).
- [3] N. ブラックマン (榎原進訳). *Mathematica 実践的アプローチ*. トッパン, 7 月 1992. (原書名: *Mathematica: a practical approach*).
- [4] N. ブラックマン (新井宏二他訳). *Mathematica 事典*. トッパン, 11 月 1994. (原書名: *Mathematica: quick reference, version 2*).
- [5] M. L. アベル, J. P. ブレイセルトン (川瀬宏海他訳). *Mathematica ハンドブック*. 東京電機大学出版局, 12 月 1994. (原書名: *The Mathematica handbook*).
- [6] M. L. アベル, J. P. ブレイセルトン (川瀬宏海, 五島奉文訳). *Mathematica 演習*. 富士ソフトウエア, 5 月 1993. (原書名: *Mathematica by example*).
- [7] S. ワゴン (長岡亮介監訳). *Mathematica で見える現代数学*. プレーン出版, 10 月 1992. (原書名: *Mathematica in action*).

- [8] R. メーダー (宇田川誠一, 時田節訳). Mathematica プログラミング技法. トッパン, 3月 1992. (原書名: Programming in Mathematica).
- [9] T. W. グレイ, J. グリン (時田節, 藤村行俊訳). Mathematica 数学の探索. トッパン, 4月 1994. (原書名: Exploring mathematics with Mathematica).
- [10] I. ヴァルデイ(時田節訳). Mathematica 計算の楽しみ. トッパン, 10月 1991. (原書名: Computational recreations in Mathematica).
- [11] R. E. クランドール (伊藤利明, 蔡東生訳). Mathematica 理工系ツールとしての. トッパン, 11月 1991. (原書名: Mathematica for the Sciences).
- [12] R. D. スキール, J. B. ケイパー (玄光男, 辻陽一, 尾内俊夫訳). Mathematica による数値計算. 共立出版, 4月 1995. (原書名: Elementary numerical computing with Mathematica).
- [13] D. C. M. バーバラ, C. T. J. ドッドソン (小林英恒訳). Mathematica 微積分入門. トッパン, 4月 1993. (原書名: Self-tutor for computer calculus using Mathematica).
- [14] S. ワゴン, E. パッケル (安田亨訳). Mathematica アニメで微積分. トッパン, 4月 1995. (原書名: Animating calculus: Mathematica notebooks for laboratories).
- [15] D. ヴィーデンスキー著 (小林英恒訳). Mathematica 偏微分方程式. トッパン, 3月 1994. (原書名: Partial differential equations with Mathematica).
- [16] S. スキエナ (植野義明訳). Mathematica 組み合わせ論とグラフ理論: 離散数学を実現する. トッパン, 7月 1992. (原書名: Implementing discrete mathematics: combinatorics and graph theory with Mathematica).
- [17] 小池慎一. Mathematica 数式処理入門. 技術評論社, 4月 1990.
- [18] 宮岡悦良. Mathematica 数学の工具箱. プレーン出版, 5月 1994.
- [19] 小林道正. Mathematica 確率・統計入門. トッパン, 8月 1994.
- [20] 守谷良二. Mathematica で数学を: 入門編. 海文堂, 9月 1993.
- [21] 守谷良二. Mathematica で数学を: 線形代数編. 海文堂, 1993.
- [22] 守谷良二. Mathematica で数学を: 微積分編 1. 海文堂, 1994.
- [23] 小野裕幸. Mathematica DSP と制御. トッパン, 10月 1992.
- [24] 阿部寛. Mathematica でみる数理物理入門 1. 講談社, 12月 1994.
- [25] 阿部寛. Mathematica でみる数理物理入門 2. 講談社, 4月 1995.