

数式処理システム Mathematica の応用

山本昌志*

2006 年 8 月 28 日

概要

数式処理システム Mathematica について、一通り学習する。電卓のように気軽に計算に使えるようになることが最初の目標である。さらに、グラフィックスが使い、プログラムができるようになるまでを目指す。

1 目的

いろいろな場面で、エンジニアや理工系の学生、研究者は複雑な計算を迅速に行う必要が生じる。例えば、非線形な素子が含まれる回路の応答を示す微分方程式を解く場合などである。この場合、運がよければ紙と鉛筆により理論式を導くことができるであろう。あるいは、近似計算ができるかもしれない。この方法は手軽だが、複雑な問題には適さないし、計算できる問題に限られる。紙と鉛筆による計算が適さないと分かると、プログラムを作成して、コンピューターによる数値計算を行うことになるだろう。この方法は複雑な問題の精度の良い近似解を求めるには良いが、プログラムの作成に時間がかかる。いずれにしても、一長一短がある。

これらの方法の欠点をカバーするものとして、Mathematica に代表される数式処理システムが使われるようになってきている。ここでは、理工系の諸問題の計算を行うツールとして Mathematica の使い方を学習する。

2 Mathematica とは

2.1 はじめに

Mathematica とは、Wolfram Research 社が提供している科学技術計算のアプリケーションソフトウェアである。世界中の教育機関や研究所、企業で使われている。秋田高専では、情報処理センターにの 20 台のパソコンにインストールされており、マニュアル 3 冊が常設されている。

* 国立秋田工業高等専門学校 専攻科 生産システム工学専攻

2.2 何ができるか

通常の研究や科学技術で使われる範囲の数学の問題に適用できる。微分や積分，線形代数，記号演算，他ほとんどの数学の分野で使うことができる。また，グラフィック簡単に使えるので，計算結果を図示するのも容易である。理工系の諸問題は数学で表すことができるので，それを解くツールとして使われている。

また，音声や画像を取り扱うこともできる。音声や画像の研究が可能である。

ただ，計算の速度はコンパイラ言語である C や FORTRAN よりも遅い。しかし，問題を解く場合，例えば微分方程式，そのプログラムは非常に簡単である。プログラム作成の時間を考えると，計算結果を得るまでの時間は，C 言語でプログラムを書くよりは圧倒的に早い。

3 使用方法

3.1 ドキュメント類

使い方が分からない場合，まずは情報処理センターに常設してある以下の書名の Mathematica のマニュアルを参照するのが良いであろう。

- MATHEMATICA ブック 5

また，オンラインドキュメントも充実している。Mathematica のメニューの [ヘルプ] → [ヘルプ] を選択すると，図 1 のようなヘルプブラウザが立ち上がるので，ほとんどのコマンドとその使い方が参照できる。

さらに，Wolfram Research 社の web ページ (<http://www.wolfram.com>) も充実しており，そこからいろいろと調べることができる。これも Mathematica のメニューから選択することにより，そのホームページにアクセスできる。メニューの [ヘルプ] → [Wolfram Research のホームページ] を選択する。くわえて，Mathematica のユーザーは世界中に広がっており，web から検索することにより，多くの問題点は解決するであろう。

3.2 10 分間のチュートリアル

まずは，チュートリアル¹で，Mathematica の概要を見ることにする。以下の通り，パソコンを操作し，全てのチュートリアルを実行すること。10 分間のチュートリアルと書いてあるが，10 分では絶対にできない。

1. Mathematica を立ち上げる

- Widows の [スタート] → [プログラム] → [Mathematica 5] → [Mathematica 5]

2. 10 分間のチュートリアルの開始

- Mathematica のメニューの [ヘルプ] → [チュートリアル]
- チュートリアル開始ボタンを右クリック

¹tutorial:[形]1. 家庭教師の 2. (英国の大学の)(個別)指導の [名](英国の大学の指導教員の)指導時間 [コンピューター]ハードウェアやソフトウェアの使い方を教える教材プログラム



図 1: ヘルプブラウザ

3. チュートリアル通りに計算を行う

- 細かいことは気にしないで，チュートリアルの文章を読み，その In[1] と書かれている文字と同じものを，Mathematica の計算させるウィンドウに書きます．または，コピー (ctrl+c) ペースト (ctrl+v) でもよい．
- 計算内容を書いたならば，Shift+Enter を押し実行させる．
- 計算結果が表示されるので，チュートリアルと同じであることを確認する．
- チュートリアルに書かれている全ての計算を行う．上方にある ▶ ボタンを右クリックすると次ページに進むことができる．

3.3 デモ

チュートリアルでだいたいの雰囲気をつかめたならば，デモプログラムを動作させてみよう．メニューの [ヘルプ]→[ヘルプ]→[デモ] から，おもしろそうなプログラムを実行してみよう．

4 基本的な約束

4.1 文法他

Mathematica は、C や FORTRAN のようにコンパイルする必要が無く、その場ですぐに計算ができる。使い方は簡単であるが、ここで学習する上で、以下の注意事項を覚えておく必要があります。

- 計算すべき内容 (プログラムやコマンドなど) を書いたならば、Shift+Enter で実行する。
- 計算の結果やプログラムは、メニューの [保存] または [別名で保存] を選択することによりセーブできる。
- キーをタイプすることにより、コマンドや文字はテキストで示すこともできるが、パレット (図 2) を使うことも可能である。もし、パレットのウィンドウが開かれていなければ、[ファイル]→[パレット]→[BasicInput(基本的な入力)] を選択すれば良い。
- 文法は、C 言語に似ている。
- 関数名やコマンドは、全て大文字から始まる。2 つ以上の単語から構成されるコマンドは、それぞれ大文字から始まる。
- 引数は、全てかぎ括弧 [引数] 内に書く。
- 式の間には 1 つ以上のスペースがあると、それは積の演算になる。

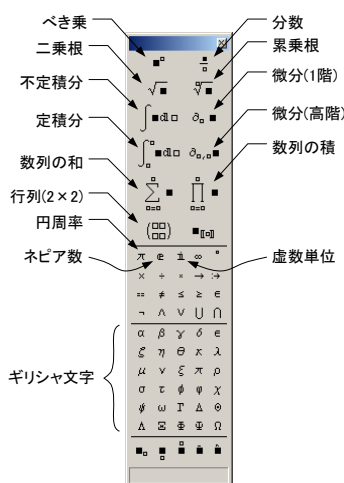


図 2: Mathematica のパレット

4.2 カーネルの強制終了

Mathematica は、フロントエンドとカーネルから構成されている。フロントエンドは、式の入力や結果の出力を行うユーザーインターフェースである。それに対して、カーネルは計算を行い、ユーザーからは見えない。これらのモジュールは、Mathlink で結ばれている。

カーネルは、以前の計算結果の内容も記憶している。たとえば、変数 a を使った場合、その内容は、クリアー命令²を使わない限り、次の計算にも継承される。これは、便利な機能である反面、問題が生じることがある。

次々に計算結果が、継承されるので、新たな計算をする場合、思わぬ間違いが生じたり、計算エラーを起こすことがある。その場合、カーネルの記憶内容をクリアーすれば良い。最も簡単な方法は、一度、カーネルを終了させることである。これは、

- メニューの [カーネル]→[カーネルの終了]→[Local(ローカル)] を選択

のようにする。カーネルの再起動は、フロントエンドから次の計算を入力すると、自動的に行われる。

Mathematica を使っていて、挙動がおかしいと感じたら、まずはカーネルを終了させるのがコツである。

5 使い方

5.1 高級電卓

まずは、円周率 π の値を 1000 桁計算してみよう。入力ウィンドウに

$$N[\pi, 1000]$$

と入れて、[Shift]+[Enter] と計算を実行させてみよう。すると、1000 桁の円周率が打ち出されたはずである。ここで使ったコマンドは N で、N[計算式, 桁数] と使う。

次に、簡単な分数の計算

$$\frac{987654321}{123456789}$$

を行う。計算の結果は分数で、正確な値が得られる。これは非常に驚くべきことで、C や FORTRAN では不可能である。正確な値ではなく、近似値が欲しい場合は、

$$N\left[\frac{987654321}{123456789}, 1000\right]$$

とすればよい。これで、1000 桁の近似値が得られる。

もう少し便利な電卓としての使い方は、

$$\theta_1 = \frac{\pi}{6};$$

$$\theta_2 = \frac{\pi}{4};$$

$$\text{Sin}[\theta_1]\text{Cos}[\theta_2] - \text{Sin}[\theta_2]\text{Cos}[\theta_1]$$

²この場合は、Clear[a] と書く。

である。1行目と2行目のセミコロン「;」は、C言語の行の区切りではない。その行の計算結果を出力しないという意味である。計算結果である3行目は出力させたいため、セミコロンが無い。3行目を

```
Simplify [Sin[θ1]Cos[θ2] - Sin[θ2]Cos[θ1]]
```

と書き換えれば、整理した結果が得られる。言うまでも無いが、3行目を

```
N [Sin[θ1]Cos[θ2] - Sin[θ2]Cos[θ1]]
```

と書き換えれば、近似値を求めることができる。

複素数の計算も問題なくできる。複素数が関係する最も美しい式

$$e^{i\pi}$$

を計算してみよう。ネピア数 e と虚数単位 i は、パレットから選択する必要がある。計算の結果は、-1になるはずである。

線形代数の計算も簡単にできる。例えば、

$$A = \{a_x, a_y, a_z\};$$

$$B = \{b_x, b_y, b_z\};$$

$$A.B$$

$$\text{Cross}[A, B]$$

でベクトル A と B の内積と外積が計算できる。固有値や固有ベクトル、行列式の計算もできる。行列は、メニューの [入力] → [表・行列・パレットの作成] を用いて入力する。次の

$$U = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix};$$

$$\text{Eigenvalues}[U]$$

$$\text{Eigenvectors}[U]$$

$$\text{Det}[U]$$

のようにすれば行列の計算ができる。

5.2 微分と積分

5.2.1 微分

微分は、D というコマンドを用いても、パレットから ∂ を使っても可能である。次の例のように、

$$D[\text{Sin}[x], x]$$

$$\partial_x \text{Sin}[x]$$

できる．高次の微分は，

$$D[x^n, \{x, 4\}]$$

のようにする． $\frac{\partial^3 \sin(x^2 y)}{\partial x \partial^2 y}$ のような偏微分は

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

とすればよい．また，

$$\partial_x f[g[x]]$$

のようなこともできる．

5.2.2 積分

不定積分は，コマンド `Integrate`，あるいはパレットから \int で可能である．たとえば，

$$\text{Integrate}[x^n, x] \\ \int x^n dx$$

のようである．次のような複雑な積分

$$\int \sqrt{a + b \text{Cos}[c x]} dx$$

も可能である．結果は楕円関数になっている．積分記号も根号もパレットから選択する．

定積分は，

$$\text{Integrate}\left[\frac{1}{1+x^2}, \{x, 0, 1\}\right] \\ \int_0^1 \frac{1}{1+x^2} dx$$

のようにしてできる．

数値積分は，

$$\text{NIntegrate}[(\text{Sin}[x])^2, \{x, 0, 2\pi\}]$$

のように，`NIntegrate` コマンドを使う．

通常のプログラミング言語で計算できるのは，この数値積分だけである．Mathematica では不定積分や定積分の計算が可能である．これは，驚くべきことである．

5.3 グラフィックス

グラフィックの機能は豊富で，通常必要なグラフはほとんど Mathematica で書くことができる．C 言語などの計算結果を Mathematica で出力することも行われており，グラフ作成が非常に容易である．

5.3.1 2次元のグラフ

まずは、非常に単純な三角関数を例にして、グラフの書き方を示す。最初の例は、

```
Plot[Sin[x], {x, 0, 2π}]
```

である。

計算結果をグラフにする場合は、

```
f[x_] = D[Sin[x^2], x];  
Plot[f[x], {x, -2π, 2π}]
```

のようにする。計算結果である微分を関数と定義して、プロットしている。関数を定義するときは、変数の後に下線をつける。

媒介変数を使ったプロットは、

```
ParametricPlot[{Sin[2t], Sin[3t]}, {t, 0, 2π}]
```

のようにする。これは、リサージュの図形である。

グラフの形に不満があれば、オプションを使って、整形できる。論文などに計算結果のグラフを貼り付ける場合、オプションを駆使して、見栄えを整えるべきである。

5.3.2 3次元のグラフ

3次元グラフは

```
Plot3D[ $\frac{\text{Sin}[\sqrt{x^2 + y^2}]}{\sqrt{x^2 + y^2}}$ , {x, -4π, 4π}, {y, -4π, 4π}]
```

と書けばよい。このままだと少し分かり難いので、オプションをつけるて分かり易くできる。

```
Plot3D[ $\frac{\text{Sin}[\sqrt{x^2 + y^2}]}{\sqrt{x^2 + y^2}}$ , {x, -4π, 4π}, {y, -4π, 4π},  
PlotRange -> All,  
PlotPoints -> 100  
]
```

媒介変数による3次元プロットは、

```
ParametricPlot3D[{Cos[t] (3+Cos[u]), Sin[t] (3+Cos[u]), Sin[u]}, {t, 0, 2π}, {u, 0, 2π}]
```

のようにする。

その他、濃淡プロットや等高線プロットもできるので、各自、必要に応じて勉強するのが良いだろう。

5.4 方程式

Mathematica での方程式の解き方は、いろいろある。ここでは、Solve を用いた非線形方程式と連立方程式、NDSolve を用いた微分方程式の計算方法を示す。

5.4.1 非線形方程式

普通，紙と鉛筆を用いて非線形方程式を解くことはできない．通常であれば，プログラムを作成して，ニュートン法等のアルゴリズムで方程式の近似解を得ることになる．

まず，簡単なところからはじめる．二次方程式は

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

とする．二次方程式の解析解が得られるはずである．このように解析解がない場合は，ニュートン法を使うことになる．例えば $\cos x - x = 0$ のような方程式は

$$\text{FindRoot}[\text{Cos}[x]-x== 0, \{x, 0.5\}]$$

とする．最初に式を書き，次に変数とその初期値をあたえる．これでニュートン法により計算できる．

5.4.2 連立方程式

連立方程式は，Solve コマンドで計算できる．例えば，

$$\text{Solve}[\{x+2y==3, 4x+5y==6\}, \{x, y\}]$$

のようにする．これは，

- Solve は，方程式を解くためのコマンドである．引数は，以下の通りである．
 - 第 1 引数は，解くべき方程式である．これが複数ある場合，括弧 $\{$ で囲んで，リストにする．
 - 第 2 引数は，解を求めるべき変数である．この場合も，複数ある場合，括弧 $\{$ で囲んで，リストにする．

となっている．

5.4.3 常微分方程式

次は，常微分方程式の計算方法を示す．解くべき常微分方程式は

$$\begin{cases} \frac{d^2x}{dt^2} + x = \sin t \\ \frac{dx}{dt} = 0 \quad (t = 0) \\ x = 0 \quad (t = 0) \end{cases}$$

とする．これは，強制振動の方程式である．この系の固有振動数は， $\omega_0 = 1$ である．そして，外力の角振動数も $\omega = 1$ となって，共振している．

Mathematica で数値計算して解くには，

```
result=NDSolve[
  {x'[t]+x[t]==Sin[t], x[0]==0, x'[0]==0},x,{t,0,500}
];
f[t_]=x[t] /. result;
Plot[f[t],{t,0,500}]
```

とする．ここで，使われているコマンドの内容は，次の通りである．

- NDSolve コマンドで，微分方程式を数値計算する．計算された結果は，solve に代入される．このコマンドの引数は，
 - － 第 1 引数は，解くべき方程式である．初期条件も入れて，連立微分方程式の形にし，リストで記述する．
 - － 第 2 引数は解くべき関数である．
 - － 第 3 引数は，独立変数とその範囲をリストで記述する．

である．

- 次にコマンド /. で，計算結果 result の関数 x[t] を f[t] に置き換えている．
- Plot コマンドで，計算結果をグラフにしている．このコマンドの引数は，
 - － 第 1 引数は，プロットする関数である．
 - － 第 2 引数は，関数の独立変数とプロットの範囲である．

である．

さあ，実行してみよう．どのような解が得られるか?．次に，外力の振動数を少し変化させた場合どのようになるか，調べよ．

先ほどのコマンドは，パレットを用いて

```
result=NDSolve[
  {∂t,tx[t]+x[t]==Sin[t],
  x[0]==Sin[t],
  x'[0]==0},
  x,{t,0,500}
];
f[t_]=x[t] /. result;
Plot[f[t],{t,0,500}]
```

としても良い。この方が、式の内容が分かりやすい。パレットを使って、数学で使う表記にした方が、ミスが少ないし、後で内容を再考しやすい。

C 言語などで、プログラムを作成して計算することを考えると、いかに Mathematica が簡単かが分かるであろう。

5.5 音

任意の波形の音を出すこともできる。例えば、音階のラの音—440[Hz]—は

```
ra = 440;  
Play[Sin[2  $\pi$  ra t], {t, 0, 5}]
```

とする。これで、440[Hz] の sin 波の音が 5 秒間鳴る。

[練習 1] 440[Hz] と 441[Hz] の音を重ね合わせて、1[Hz] のうなりを聞いてみよう。

5.6 その他

Mathematica では画像処理もできる。ここでは、割愛するので、興味がある人は、自分で学習して欲しい。

6 ファイル処理

実験データを整理したり、計算結果を他のプログラムで処理したりする場合、ファイル入出力の処理が必要不可欠である。ファイル処理の基本について学習する。

6.1 ファイルヘデータ出力

先ほどの微分方程式 (5.4.3) の数値計算の結果は、グラフに表示させた。ここでは、グラフに表示のみならず、解の値をファイルに出力する方法を示す。コンピュータープログラムでファイルを取り扱う場合、Open と Close の処理が必要である。FORTRAN や C 言語でその処理が必要なように、Mathematica でも必要である。Mathematica では、次の手順でファイルヘデータを書き出す。

1. OpenWrite コマンドで、ファイルをオープンする。
2. Write コマンドで、データをファイルに書き込む。
3. Close コマンドで、ファイルをクローズする。

先ほどの常微分方程式の結果を、ファイルに書き出すことを練習する。プログラムの動作は次のようにする。

- 計算すべき微分方程式は、式 (5.4.3) の通りとする。

- 微分方程式の解 $x(t)$ は, $t_{\min} \leq t \leq t_{\max}$ の範囲計算する .
- 計算した範囲はプロットするとともに , num 個の数値データとしてファイルに書き込む .

Mathematica のプログラムは , 次のようになる . このプログラムの大まかな動作を理解して , 実行させてみよう . そして , 書き込まれたファイルの中身を調べてみよう . このプログラムの各行の動作をプログラムの後に , 説明している .

```
tmin=0.0;
tmax=10.0;
num=1000;

result=NDSolve[
  {x'[t]+ x'[t]+x[t]==0,x[0]==0,x'[0]==1},
  x,{t,tmin,tmax}
];

f[t_]:=x[t]/.result[[1]];

Plot[f[t],{t,tmin,tmax}];

wfile=OpenWrite[
  "u:/temp/numerical_result.txt",
  FormatType->OutputForm
];

For[i=0,i<=num, i=i+1,
  t=(tmax-tmin) i/num+tmin;
  Print[t,"\t",f[t]];
  Write[wfile,t,"\t",CForm[f[t]]];
];

Close[wfile];

Clear[tmin,tmax,num,t,i,f,x,result,wfile];
```

このプログラムの動作について , 細かく説明しておく .

- 最初の 3 行で , 変数 tmin,tmin,num の値を設定している .
- 次の NDSolve コマンドで , 微分方程式の解を数値計算している . 微分方程式と初期条件 , 独立変数 , 解を計算する領域がそのコマンドの引数として与えられている . 解は , Mathematica の形式になっており , それを変数 result に代入している .
- Mathematica の形式である解は , x[t]/.result[[1]] で関数に直している . これについては , 今はおまじないと思って欲しい³ . その解の関数は , f[t_]:=とすることにより , 関数 f[t] と定義している . 以降 , 解の値が必要なときは , 関数 f[t] を使う .

³気になる人はマニュアルを見て欲しい

- 次の Plot コマンドにより，解をグラフ表示している．
- 次の OpenWrite コマンドにより，解の値を書き出すファイルをオープンしている．その第 1 引数がファイル名，第 2 引数はオプションで，ここでは出力形式をしている．このファイルは，wfile というストリーム⁴を使って，データを書き込むことができる．
- 次の FOR 文で num 個の解のデータをファイルに書き出す．FOR 文はループ制御で，初期値 $i=0$ ，ループ条件 $i \leq \text{num}$ ，ループ後処理 $i=i+1$ で，それ以降の文を繰り返す．
- データを書き出す時刻は t で， t_{\min} から t_{\max} まで，num 個に分割している．
- Print 文で，データをディスプレイに書き出している．各行は，最初に時刻 t ，そして''\ t''によりタブ (Tab) を空け，関数 $f[t]$ の値を書き出している．時刻と関数の値の間にタブを入れるのは，データを書き出すときの常套手段である．
- ファイルを使い終わったら，クローズしなくてはならない．Close コマンドを使う．
- もう一度，そのプログラムを実行させる場合，前回の変数が残っていると，思わぬ動作をすることがある．そのため，使った変数はクリアするのが望ましい．Mathematica では，そのためのコマンド Clear が用意されている．

6.2 ファイルからデータ入力

次に，ファイルからのデータの入力方法について，説明する．これは，実験データなどを Mathematica で処理する場合，必要不可欠なテクニックなので，覚えておくといい．

先ほど作成したデータを Mathematica で読み込み，処理をすることを考える．処理は，フーリエ変換を行うことにする．ここで，作成するプログラムの流れは，

- ファイルからデータを読み込む．
- 読み込んだデータをプロットする．
- データのうち，式 (5.4.3) の解の $x(t)$ の数値を抽出する．
- $x(t)$ の数値の列をフーリエ変換する．
- フーリエ変換されたデータをプロットする．

である．実際の，プログラムは次のようになる．このプログラムの流れを理解して，実際に打ち込んで実行させよう．各行の説明は，プログラムリストの後に記述している．

```
data=ReadList[
  "u:/temp/numerical_result.txt",
  {Number,Number}
];
```

⁴ファイルを指定する変数みたいなもの

```

ListPlot[data];

trdata=Transpose[data];
xlist=trdata[[2]];

ft=Fourier[xlist];

ListPlot[
  Abs[ft],
  PlotRange->{{0,100},{0,5}},
  PlotJoined->True
];

```

このプログラムの動作について、細かく説明しておく。

- ReadList コマンドで、データを読み込んでいる。第 1 引数でファイルを指定している。第 2 引数が、データの種類とその記述方法を示している。ここでは、各行に数値 (Number) が 2 つ並んでいることを示している。読み込んだ結果は、リスト (行列みたいなもの) として、data に格納される。
- 読み込まれたデータは、ListPlot コマンドでグラフ化されている。
- Transpose コマンドで、行列 data の転置行列を trdata としている。
- trdata[[2]] で第 2 行、すなわち、 $x(t)$ のデータの数値列を抽出している。抽出された数値列は、xlist に格納される。
- Fourier コマンドにより、xlist の数値列をフーリエ変換している。フーリエ変換されたデータは、ft 変数に格納される。フーリエ変換された結果も数値列である。
- ListPlot コマンドにより、フーリエ変換の結果をプロットしている。
 - 第 1 引数が、プロットするデータである。一般に、フーリエ変換では複素数に変換されるので、絶対値化 Abs コマンドを用いて、実数に直している。
 - PlotRange オプションで、グラフの表示領域を指定している。横軸は 0 から 100 まで、縦軸は 0 から 5 までとしている。
 - PlotJoined オプションで、離散的なプロット点を直線で接続している。

7 プログラミング

先のファイル処理では、いろいろなコマンドが並んでいた。そして、それは上から下へと処理が進められた。これは、プログラムに他ならない。プログラムに必要な構文は、

- コメント文
- 制御文

- サブルーチン

等である。Mathematica には、これらに関する全てのコマンドが用意されている。それも、FORTRAN は言うに及ばず、C 言語よりも、そのコマンドは高度で幅広い。コマンドが多すぎるのが欠点であるが、マニュアルやヘルプを参照して必要なコマンドを使うよう心がけるべきである。

全てを学習することは時間的に不可能なので、ここでは概略のみを述べる。諸君は Mathematica でプログラムが書けることを覚えておき、研究で必要になればそれを使うようにすれば良い。

7.1 コメント文

プログラムの動作には全く関係ないが、プログラマーのメモがコメント文である。Mathematica では、次のように書く。

```
書式
(*      ここにコメントを書く      *)
```

(*) がコメントのはじまりで、*) が終わりである。コメント文は、複数の行にわたってもよい。C 言語の /*と*/ と全く同じである。

7.2 制御文

一般に、コンピュータープログラムの制御構造は分岐と繰り返し (ループ) から成る。Mathematica でも、それらの構造はサポートされている。

7.2.1 分岐

以下の分岐の構文が使える。通常のプログラミング言語と似ているので、すぐに理解できるであろう。

```
書式
If [判定式,
    判定式が正しい時の文 1;
    判定式が正しい時の文 2,
    判定式が誤り時の文 1;
    判定式が誤り時の文 2
];
```

Mathematica ではブロック構造が使えないので、複数の実行文があるときはセミコロン (;) を使う。そして、ブロックの区切りにカンマ (,) を使っている。これを使った関数

$$f(x) = \begin{cases} \sin x & x < 0 \text{ のとき} \\ x & 0 \leq x \text{ のとき} \end{cases}$$

の例を示す。

```
f[x_]:=If[x<0, Sin[x],x];  
Plot[f[x],{x,-10,10}];  
Plot[f'[x],{x,-10,10}];
```

この例で分かるように、普通の関数のように微分ができるのである。

If のほかに、分岐を行う構文に Which や Switch がある。ヘルプを見て、各自が必要になったときに学習せよ。

7.2.2 繰り返し (ループ)

C 言語と全く同じ、For というループ文が用意されている。

書式

```
For[初期の式, 判別式, ループ後の式,  
  文 1;  
  文 2;  
  文 3;  
];
```

例えば、次のように書く。

```
For[i=1, i<20, i++,  
  Print[N[Pi,i]];  
];
```

For 以外に、Do や While を使ったループ文も使える。また、Break や Continue, Return など C 言語とほとんど同じ構文が使えるようになっている。必要になったときに学習せよ。

7.3 サブルーチン

サブルーチン—C 言語の関数—を使いたい場合は、Module を使う。これを使ったサブルーチンの書き方は、次のようにする。

書式

```
関数名 [引数並び_] := Module[{ローカル変数並び},  
  文 1;  
  文 2,  
  文 3;  
  Return[返却値];  
];
```

これで、サブルーチンでの処理を定義することができる。プログラム中では、サブルーチンは、使用前に定義する必要がある。これは C 言語と同じであるが、C 言語の場合はプロトタイプ宣言が使えるので、定義を書く場所は関数はその関数が最初に現れるところに書かなくても良い。

つぎに、サブルーチンを使ったプログラム例を示す。ここで取り扱っている例は、

$$\frac{d^2f}{dt^2} + \frac{df}{dt} + f = \sin t \qquad f'(0) = 0 \qquad f(0) = 0$$

という微分方程式の数値計算を行う。そして、呼出元で与えられた値を中心にして、全幅 2 でグラフを描く。サブルーチンの返却値は、呼出元で与えられた t の場合の微分方程式の近似値である。

```
(* ===== Subroutine ===== *)  
  
func[x_] := Module[{xlow, xhigh},  
  xlow = x - 1;  
  xhigh = x + 1;  
  Print["xlow = ", xlow];  
  result = NDSolve[{f''[t] + f'[t] + f[t] == Sin[t],  
    f'[0] == 0, f[0] == 0}, f, {t, xlow, xhigh}  
  ];  
  ans[t_] = f[t] /. result;  
  Plot[ans[t], {t, xlow, xhigh}];  
  Return[ans[x]];  
];  
  
(* ===== Main routine ===== *)  
  
value1 = func[3.4];  
value2 = func[6.5];  
Print["value1 = ", value1];  
Print["value2 = ", value2];
```

8 演習問題

8.1 データのフーリエ変換

データを与えるので，以下の処理を行うプログラムを作成せよ．これは，データの読み込みのプログラムとほとんど同じなので簡単である．

データは， $y = f(x)$ の値がテキストファイルで書かれている．各行に， x のと $f(x)$ の値が書かれている．ただし， x は等間隔に書かれている．

- ファイルからデータを読み込む．
- 読み込んだデータをプロットする．
- データのうち， $f(x)$ の値を抽出する．
- $f(x)$ の数値の列をフーリエ変換する．
- フーリエ変換されたデータをプロットする．ただし，プロットする範囲を考えて，フーリエ変換のピークが分かるようにすること．

8.2 データのフィット

データを与えるので，以下の処理を行うプログラムを作成せよ．データの与え方は，フーリエ変換の問題と同じである．ただし，内容は異なる．作成するプログラムは，以下の内容であること．

- ファイルからデータを読み込む．
- 読み込んだデータをプロットする．
- 読み込んだデータを 2 次関数で，最小自乗フィットする．
- フィットの結果をグラフにする．
- 元のデータとフィットした結果を同じグラフに重ねて，プロットする．

ここのプログラムは説明していないので，マニュアルの P.900 ~ 903 を参考にすること．

9 考察課題

- Mathematica と通常のプログラミング言語 (C や FORTRAN など) との違いについて述べよ．
- 自分の研究に Mathematica は役立つか? . 役立つ場合は，どのように利用できるか述べよ．役立つない場合は，その理由を述べよ．
- Mathematica 以外にも数式処理システムがある．フリーのものを含めて，どのようなシステムがあるか，調べよ．

10 レポート

レポートは、以下の通りとする。

- 提出期限は、9月13日(水)のPM1:00とする。
- ワードプロを用いて作成すること。手書きは不可とする。
- 用紙は A4 サイズとする。
- 表紙には、以下の項目が含まれること。これもワードプロで書くこと。
 1. 実験課題 数式処理システム Mathematica の応用
 2. 実験日 (8/29, 8/30, 9/4, 9/6)
 3. 学籍番号
 4. 氏名
 5. レポート提出日
- レポートの内容が含まれること。
 1. 目的
 2. Mathematica について
 - Mathematica は、どのようなソフトウェアなのか、自分で調べて記述すること。実験の指導書を写すことは厳禁である。
 3. 練習問題のプログラムと結果
 - 練習問題の内容
 - フローチャートとアルゴリズム
 - プログラム
 - 出力結果を載せること。
 4. 考察課題
 5. 感想
- 番号の書き方
 - 各頁の下に、頁番号をつけること。ただし、表紙には番号をつけなくて、その次の頁を 1 とする。
 - セクション毎に、番号をつけること。サブセクション、サブサブセクションも同様である。
 1. 目的
 2. Mathematica について
 - 2.1 数式処理システムとは
 - 2.2 Mathematica とは
 - ⋮

3. 練習問題

⋮

— 図や表，式には番号をつけること．

* 図は，その下に以下のように書く．

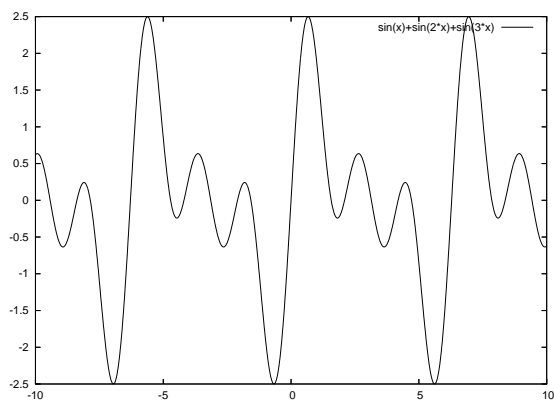


図 3: 図の例

* 表は，その上に以下のように書く．

表 1: 排他的論理和の真理値表

A	B	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

* 式は，その横に以下のように書く．

$$e^{i\theta} = \cos \theta + i \sin \theta \quad (1)$$