

[資 料]

高機能数値計算・可視化機能ソフト MATLAB の基本的な使い方

秋田県立大学 陳 国躍
東北大学情報部情報基盤課 共同利用支援係

概要

MATLAB は、高機能な数値計算機能と多彩な可視化機能を備えた技術計算ソフトウェアです。科学的、工学的分野の様々な数値計算(特に行列演算)、データ解析、シミュレーション、およびビジュアライゼーションのための統合環境を提供しています。また、MATLAB は高性能な数値計算機能を持つばかりでなく、多彩な可視化機能を備えており、計算結果を簡単にグラフに表示することができます。

今回は、MATLAB の基本的な使い方を紹介します。

キーワード：高機能な数値計算、行列演算、可視化機能、グラフ処理

1 はじめに

MATLAB は、専門のプログラマやコンピュータに詳しい人だけではなく、一般の利用者にも高次の機能が簡単に使えるように作られています。このため、次にあげるような特徴をもっています。

- MATLAB という名称が “matrix laboratory” の略称というだけあって、特に、行列とベクトルの計算は、数学的な記述に近い表現で簡単に演算できます。このため、Fortran、Basic、C といった言語でプログラムを記述するのに要する時間と比べ、MATLAB は非常に少ない時間で多くの問題を解くことが可能となります。特に、大規模な行列計算は、MATLAB が本来得意とするところでもあります。
- 数値実験の結果を 2 次元および 3 次元図面で簡単に表示できると同時に、ファイル出力や印刷出力ができます。ファイル出力した図面を文書編集清書プログラム TeX の入力とすることができます。
- 対話型の実行により、モデルの変数やパラメータを種々変更して数値実験ができます。
- 使い方については、必要な時点で案内情報を画面に出すことによって、極力手引書に頼らずにすむよう help コマンドによるオンラインマニュアルが完備しています。
- ツールボックスと呼ばれる特定分野の解法を集めたジュールを組み込むことにより、容易に機能拡張を行うことができ、専門的な利用にも柔軟に対応できます。

MATLAB はコアモジュールの MATLAB を中心として、ブロックダイアグラムシミュレータの Simulink、および Toolbox と呼ばれるアプリケーションモジュールから構成されています。本センターでは次の Toolbox を使用できます。

- * Communications System Toolbox : 通信システムの設計、解析
- * Control System Toolbox : 線形制御システムの設計、解析
- * Curve Fitting Toolbox : 数式近似ツール
- * DSP System Toolbox : 信号処理システムの設計とシミュレーション
- * Fixed-Point Toolbox : 固定小数点データタイプと演算機能を提供
- * Fuzzy Logic Toolbox : ファジー理論に基づくシステムの設計、解析
- * Image Processing Toolbox : 画像処理、解析、可視化およびアルゴリズム開発のためのグラフィカル ツール
- * MATLAB Coder : MATLAB コードからスタンドアロンの C コードと C++ コードを生成
- * MATLAB Compiler : MATLAB アプリケーションを実行ファイル、共有ライブラリとして生成
- * Model Predictive Control Toolbox : 予測制御システムの設計、解析
- * Neural Network Toolbox : ニューラルネットワーク理論を用いた複雑非線形事象のモデリング
- * Optimization Toolbox : 標準的および大規模な最適化に広く使用されるアルゴリズムを提供
- * Partial Differential Equation Toolbox : 偏微分方程式の解法ツール
- * Robust Control Toolbox : ロバスト性の評価、解析
- * Signal Processing Toolbox : アナログおよびデジタル信号処理 (DSP) のアルゴリズムを提供
- * Simulink Coder : Simulink ダイアグラム、および MATLAB 関数から C コードと C++ コードを生成
- * Simulink Control Design : Simulink でモデル化された制御システムの設計と解析
- * Simulink Design Optimization : Simulink でモデル化された制御システムの数値最適化
- * Simulink Fixed Point : Simulink プロダクトファミリーで固定小数点機能が利用可能
- * Simulink Verification and Validation : Simulink でモデル化された制御システムの確認、検証およびテスト
- * Statistics Toolbox : データの整理、解析、およびモデリングのためのアルゴリズムとツールを提供
- * Symbolic Math Toolbox : 数式処理と可変精度演算を行うためのツールを提供
- * System Identification Toolbox : 測定された入出力データから動的システムの数学モデルを構築
- * Wavelet Toolbox : ウェーブレット変換を用いた画像処理、解析、可視化ツール

今回、MATLAB の入門として、その基本的な利用法と使用例を幾つか紹介します。Simulink と Toolbox については、次の機会に紹介することにいたします。

2 MATLAB を使うために

2.1 基本的な環境設定

MATLAB は、サイバーサイエンスセンターの gen という名の並列コンピュータにインストールされています。ここでは、利用者の方が、自分の使用しているワークステーションあるいは X 端末から gen へリモートログインして利用することを想定して話を進めます。具体的な例を下に示します。ssh コマンドの X11Forwarding 機能を有効にして、gen に接続します。

```
% ssh -X 利用者番号@gen.isc.tohoku.ac.jp <R>          注：遠隔ホスト gen へ接続

The authenticity of host 'gen.isc.tohoku.ac.jp (xx.xx.xx.xx)' can't be established. RSA key fingerprint is
fd:c2:9a:11:xx:xx:xx:xx:xx:xx:xx:xx:xx:cd:53:9f.
Are you sure you want to continue connecting (yes/no)? yes          注：初回接続時のみの問合せ

Warning: Permanently added 'gen.isc.tohoku.ac.jp,xx.xx.xx.xx' (RSA) to the list of known hosts.

利用者番号@gen.isc.tohoku.ac.jp's password: パスワード          注：パスワードを入力

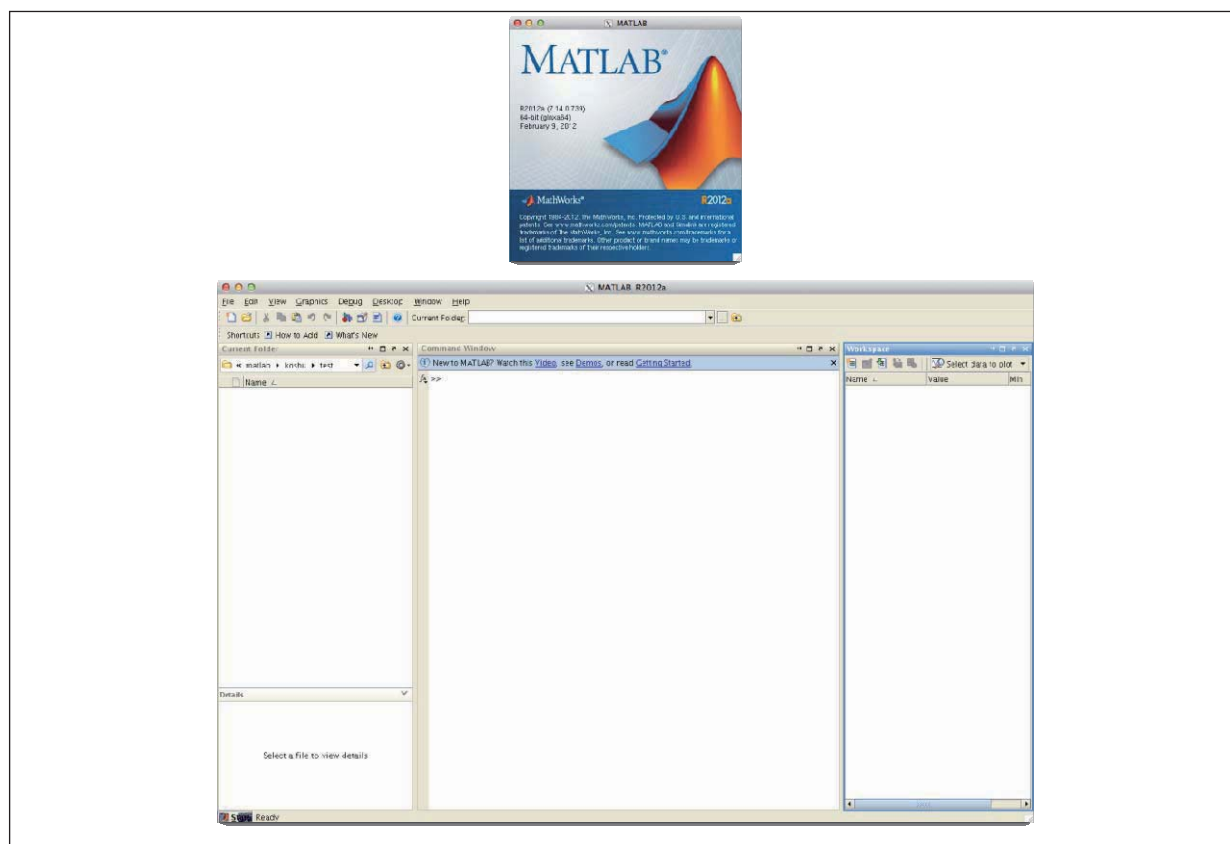
Last login: Mon Apr 1 09:26:11 2012 from yyy.yyy.tohoku.ac.jp 4 月 1 日 13:29:59 JST 2012

[利用者番号@gen ~]$          注：コマンド入力待ち
```

注：<R>はリターン(改行)の意味で、以下でも同じ意味を表します

2.2 MATLAB の起動と終了

MATLAB の起動には、`matlab` というコマンドを使用します。スプラッシュ画面の表示の後、GUI ウィンドウが表示されます。



スプラッシュ画面が表示されない場合は、ローカル端末で X 環境が使用できることと、`ssh` コマンドを再確認するようにしてください。ここで、>>は MATLAB の入力促進記号(プロンプト)です。例えば、MATLAB を終了したい場合には、`quit` あるいは `exit` を入力すれば終了します。このような `quit` や `exit` などを、コマンドといいます。

3 数値計算

3.1 基本演算

数値計算の場合は、MATLAB 上でそのまま入力して簡単に演算できます。利用可能な演算子と関数を付録に記します。以下では、式(1) (2)を利用して、計算方法を示します。

$$\frac{5(3+4)}{2} \tag{1}$$

```
>> 5*(3+4)/2    <R>
ans =
    17.5000
>> x = 3,   y = 2,   z = 1.5    <R>
X =
     3
y =
     2
Z =
    1.5000
>> (x^2+y^5)/sin(z)    <R>
ans =
    41.1030
```

$$\frac{(x^2 + y^5)}{\sin(z)} \quad (\text{初期値 } x = 3 \ y = 2 \ z = 1.5) \tag{2}$$

3.2 ベクトル・行列の演算

ベクトル・行列の演算が、数学的な記述に近い形で行えるのが、MATLAB の大きな特徴の一つです。そこで、以下では、式(3)に示す行列演算を、MATLAB 上で実現する方法について紹介します。

$$AX = B \quad \begin{pmatrix} 1 & 3 & 2 \\ 5 & 1.2 & 3.4 \\ 1.5 & 4 & 5.1 \end{pmatrix} \begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} 6.1 \\ 4 \\ 5.8 \end{pmatrix} \tag{3}$$

以下に示すように、MATLABは行列の左除算(\: バックスラッシュ)を提供しています。Aが正方行列ならば、解XはA\B(方法2)で求めたものとinv(A)*B(方法1)で求めたものとほぼ同じですが計算方法は異なります。Aがn行n列でBがn個の列ベクトルかn個の列行列ならば、X = A\Bは方程式AX = Bをガウス消去で計算した解になります。Aのスケールが悪かったり、特異点に近かったりすると警告メッセージが表示されます。

Aがm行n列(m≠n)でBがm個の列ベクトルかm個の列行列ならば、X = A\Bは未知数の数と方程式の数不一致な方程式AX = Bの最小二乗的な解になります。Aのランクkはピボットを行ったQR分解から求められます。解Xは、k個のゼロでない成分を持ちます。k<nであれば、X = A\Bは通常、最小残差ノルム||AX - B||による最小二乗解であるpinv(A)*Bと同じ解にはなりません。

```

>> C=[6.1 4 5.8] <R>          注：ベクトルの入力（方法1）
C =
    6.1000    4.0000    5.8000      注：' は転置の演算子
>> B = C' <R>
B =
    6.1000
    4.0000
    5.8000
>> B2 = [ 6.1 <R>
4 <R>
5.8] <R>          注：ベクトルの入力（方法2）
B2 =
    6.1000
    4.0000
    5.8000
>> A = [1 3 2; 5 1.2 3.4; 1.5 4 5.1] <R>          注：行列の入力（方法1）
A =
    1.0000    3.0000    2.0000
    5.0000    1.2000    3.4000
    1.5000    4.0000    5.1000
>> A1 = [1 3 2; 5 1.2 3.4; 1.5 4 5.1] <R>          注：行列の入力（方法1）
注：文の終わりに ; を入れるとその値は画面に現れない
>> A2 = [1 3 2 <R>
5 1.2 3.4 <R>
1.5 4 5.1] <R>          注：行列の入力（方法2）
A2 =
    1.0000    3.0000    2.0000
    5.0000    1.2000    3.4000
    1.5000    4.0000    5.1000
>> eig(A) <R>          注：行列 A の固有値
ans =
    9.0284
   -2.9432
    1.2148
>> X = inv(A)*B <R>          注：ベクトル X の解(方法1)
X =          (この方法は、通常良い方法ではありませんので、ご注意！)
    0.9166
    2.4089
   -1.0217
>> X = A \ B <R>          注：ベクトル X の解（方法2）
X =
    0.9166
    2.4089
   -1.0217
>> max(X) <R>          注：ベクトル X の要素の最大値
ans =
    2.4089
>> mean(X) <R>          注：ベクトル X の要素の平均値
ans =
    0.7679
>> n=length(X) <R>          注：ベクトル X の要素の長さ
n =
    3
>> x2 = X(2) <R>          注：ベクトル X の2番目の要素
x2 =
    2.4089

```

3.3 複素数

MATLABでは、複素数をすべての演算や関数において使用することができます。虚数表現としては、 i と j のどちらも許容されます。一方、 i や j を実変数として利用することも可能です。どちらで利用しているのかは、表現方法やそれまでの流れなどからMATLABが自動的に解釈しますので、それらを混同しないように注意する必要があります。

以下では、MATLAB上で複素数演算の例を式(4)(5)を用いて示します。

$$Z1 = 3 + 4i \qquad Z2 = 1 - 7i \qquad (4)$$

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + i \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \qquad B = \begin{pmatrix} 3 & -2 \\ -4 & 8 \end{pmatrix} + i \begin{pmatrix} 7 & -3 \\ 1 & -2 \end{pmatrix} \qquad (5)$$

```

>> z1 = 3+4*i
z1 =
    3.0000 + 4.0000i
>> z2 = 1-7*j
z2 =
    1.0000 - 7.0000i
>> z1+z2
ans =
    4.0000 - 3.0000i
>> z1*z2
ans =
    31.0000 - 17.0000i
>> z1 / z2
ans =
   -0.5000 + 0.5000i

>> A = [ 1 2 ; 3 4 ] + i*[ 5 6 ; 7 8]           注：複素数行列の入力(方法1)
A =
    1.0000 + 5.0000i    2.0000 + 6.0000i
    3.0000 + 7.0000i    4.0000 + 8.0000i
>> B = [ 3+7i -2-3i ; -4+1j 8-2j]           注：複素数行列の入力(方法1)
B =
    3.0000 + 7.0000i   -2.0000 - 3.0000i
   -4.0000 + 1.0000i    8.0000 - 2.0000i
>> eig(A*B)                                   注：複素数行列 A*B の固有値
ans =
   -0.7101 - 4.5909i
   17.7101 +37.5909i
>>real(eig(A*B))                             注：固有値の実数部
ans =
   -0.7101
   17.7101
>> imag( eig(A*B) )                          注：固有値の虚数部
ans =
   -4.5909
   37.5909
>> abs( eig(A*B) )                           注：固有値の絶対値
ans =
    4.6455
   41.5539
    
```

4 ファイルとデータの入出力

MATLABでは、`save`と`load`というコマンドによって、MATLABで使用していたときの変数をディスク上に書き込んだり、読み込んだりすることができます。以下では、`load`コマンドによりファイルからMATLAB上へASCIIデータを読み込む方法を説明します。いま、データファイルは`sam1.dat`, `sam2.dat`, `sam3.dat`とします。

1) `sam1.dat`というファイルの内容(次のデータが一行に書かれていることに注意) :

```
0.0 0.3 0.95 -0.4 0.4 0.2 -0.3 0.0 0.3 0.0 -0.2 0.1 0.0 0.0 0.0 0.0
```

2) `sam2.dat`というファイルの内容(次のデータが7行に書かれていることに注意) :

```
0
3
9
5
-4
2
-3
```

3) `sam3.dat`というファイルの内容(5行3列のデータです) :

```
1.1 1.2 1.3
2.1 2.2 2.3
3.1 3.2 3.3
4.1 4.2 4.2
5.1 5.2 5.3
```

以上3つのデータファイルについて、MATLAB上へコマンド`load`によりASCIIデータを読み込んだ結果を次頁に示します。

また、MATLABでもC言語のように、`fread`と`fscanf`コマンドにより、ファイルからバイナリデータや書式付きデータを読み込むことができます。逆に、ファイルへバイナリデータや書式付きデータを書き込む場合には、`fwrite`と`fscanf`コマンドを用います。詳しい使用方法については、マニュアルを参照してください。

```

>> load sam1.dat
>> sam1
sam1 =
  Columns 1 through 7
         0    0.3000    0.9500   -0.4000    0.4000    0.2000   -0.3000
  Columns 8 through 14
         0    0.3000         0   -0.2000    0.1000         0         0
  Columns 15 through 16
         0         0
>> load sam2.dat
>> sam2
sam2 =
         0
         3
         9
         5
        -4
         2
        -3
         0
>> load sam3.dat
>> sam3
sam3 =
    1.1000    1.2000    1.3000
    2.1000    2.2000    2.3000
    3.1000    3.2000    3.3000
    4.1000    4.2000    4.3000
    5.1000    5.2000    5.3000

```

5 スクリプトファイル

MATLAB上で、制御コマンドを一行一行入力するのは、かなり面倒な作業です。そこで、シェルスクリプトやバッチファイルのように、予め実行したいコマンド群をまとめてファイルに書いておき、まとめて実行させることが可能です。環境変数MATLABPATHで指定されたディレクトリに、.m という拡張子を持つファイルを作成すると、そのファイルはMATLABのスクリプトファイルとみなされます。

sample.m というファイルの内容が以下のものであるとします。

```

load sam1.dat
sam1
load sam2.dat
sam2
load sam3.dat
sam3

```

MATLAB上では “>>sample <R>” と入力すると、前節に示したものと同じ結果になります。

6 グラフィックス表示機能

6.1 簡単なグラフィックスの表示

MATLAB は、数値計算の他にも、データの可視化に関して優れた機能を持っています。座標系などを指定しない場合は、自動的に見やすい範囲で表示してくれます。もちろん、表示方法を個人の好みにより細かく指定することも可能です。

以下では、母音/a/のデータ（サンプリング周波数 16kHz, 60ms）を `speech.dat` というデータファイルを使って、時間波形とその周波数特性（スペクトル）を描く例を示します。

```
>> load speech.dat
>> plot(speech)
>> close
>> freq = 20*log10(abs(fft(speech)))
>> plot( freq(1:length(speech)/2))
```

`plot`, `close`, `log10`, `abs`, `fft`, `length` などコマンドの機能については、参考書あるいはオンラインマニュアルを参照してください。

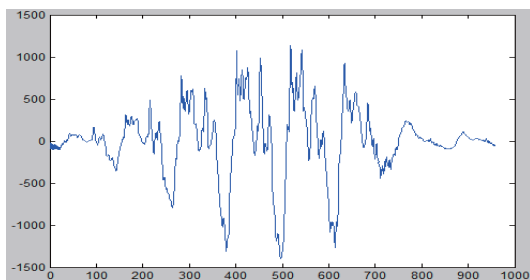


図 1 : 母音/a/の音声データの時間波形

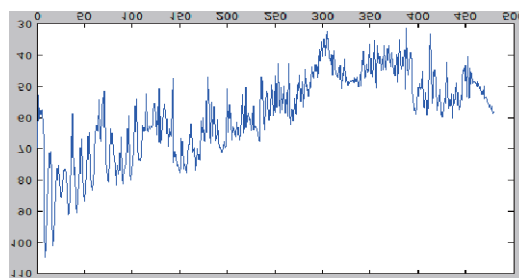


図 2 : 母音/a/の音声データの周波数特性

次の例は、図 2 に示したような周波数特性を、図 3 に示したように AR モデルにより、その周波数特性の包絡線を計算して表示させたものです。（なお、コマンド `lpc` と `freqz` は System Identification Toolbox の機能です）。このような仕事も大変簡単にできてしまうことがお分かりいただけるでしょう。

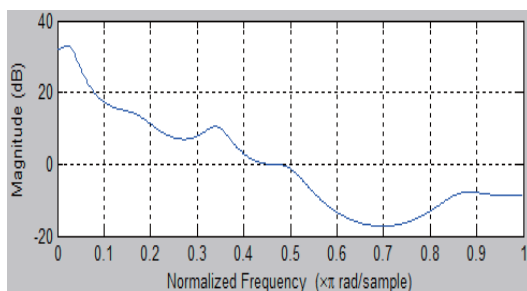


図 3 : 図 2 に示した周波数特性の包絡線

```
load speech.dat
a = lpc( speech, 12 );
freqz( 1, a );
```

6.2 カラーグラフィックス

MATLAB では、カラーグラフィックスの表示も容易に行えます。`plot(x,y,'linetype')` は、ベクトル `x` を x 軸、ベクトル `y` を y 軸とした曲線を 'linetype' で指定されたタイプの線種でプロットします。X を省

略場合には、y のインデックスが x 軸に対応することになります。'linetype'は、線のタイプと色 (表 1) を指定します。'linetype'を省略した場合、自動的に青色の実線を使用します。同じ図面上で異なるラインタイプと同じ色を指定することもできます。図 4 にその例を示します。

表 1 : ラインのタイプと色

シンボル	ラインタイプ	シンボル	ラインタイプ	シンボル	色
-	実線	.	点	b	青
:	点線	o	円	g	緑
-.	鎖線	x	x 印	r	赤
--	破線	+	プラス記号	c	シアン
(none)	線なし	*	星印	m	マゼンタ
		s	正方形	y	黄
		d	ダイヤモンド	w	白
		v	三角形 (上向き)	k	黒
		^	三角形 (下向き)		
		<	三角形 (左向き)		
		>	三角形 (右向き)		
		p	五角形		
		h	六角形		

6.3 グラフィックスの出力

作成した図面を出力するには、**print** というコマンドを使用します。利用者が自分のところにあるプリンタから出力する場合には、以下に示すオプションを利用して、一度ポストスクリプトファイルなどへ出力し、そのファイルを自分のところに **sftp** で送ることになります。

```
print [ -option ] filename
      option:    dps:  白黒   ps   ファイル          dpsc: カラー  ps   ファイル
                deps: 白黒   eps  ファイル          depsc: カラー eps  ファイル
```

これ以外のデバイスも指定可能ですので、マニュアルを参照してください。

6.4 3次元グラフィックス

この説は、MATLAB に組み込まれている **peaks** という特殊な多峰型関数を用いて、3次元グラフィックスの作成を説明します。**Peaks** 関数は、x と y の関数で、次の形をしています。

$$f(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-x^2-y^2} - \frac{1}{3} e^{-(x+1)^2-y^2} \tag{6}$$

MATLAB で **peaks(n)** を実行すると、n×n の正方行列が得られます。引数の n を省くと、標準値 49 が採用されます。

以下は、`Z=peaks(7)`の結果です。この行列の各要素は、`x`および`y`を-3から3まで変化させた場合の値になります。列方向が`x`、行方向が`y`の変化に対応します。

```
>> z = peaks(7)
z =
    0.0001    0.0034   -0.0299   -0.2450   -0.1100   -0.0043    0.0000
    0.0007    0.0468   -0.5921   -4.7596   -2.1024   -0.0616    0.0004
   -0.0088   -0.1301    1.8559   -0.7239   -0.2729    0.4996    0.0130
   -0.0365   -1.3327   -1.6523    0.9810    2.9369    1.4122    0.0331
   -0.0137   -0.4808    0.2289    3.6886    2.4338    0.5805    0.0125
    0.0000    0.0797    2.0967    5.8591    2.2099    0.1328    0.0013
    0.0000    0.0053    0.1099    0.2999    0.1107    0.0057    0.0000
```

図5～図6に、3次元のカラーグラフィックスの例を示します。また、図5に示す3次元グラフィックスは、図7～図11に示すような様々な方法によって等高線表示をすることも可能です。

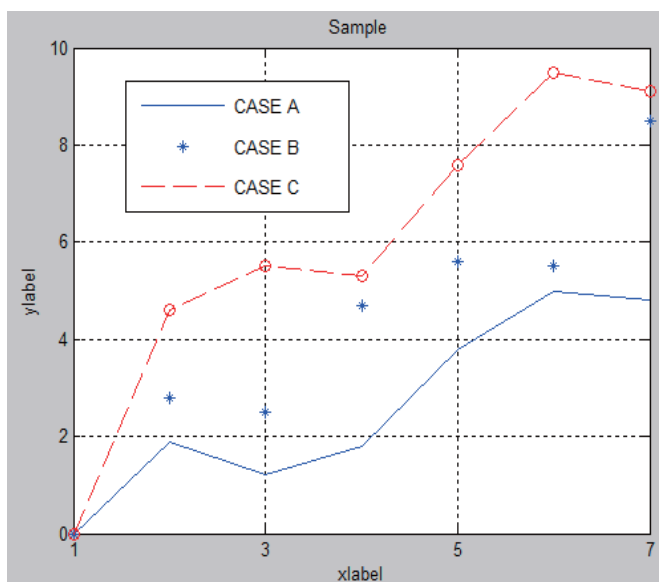


図 4

```
c1=[0.0 1.9 1.2 1.8 3.8 5.0 4.8];
c2=[0.0 2.8 2.5 4.7 5.6 5.5 8.5];
c3=[0.0 4.6 5.5 5.3 7.6 9.5 9.1];
plot(c1)
hold on
plot(c2, 'b*')
plot(c3, 'r--')
plot(c3, 'ro')
hold off
axis([1 7 0 10])
set(gca,'XTick',[1:2:7])
set(gca,'YTick',[0:2:10])
xlabel('xlabel');
ylabel('ylabel');
title('Sample ');
grid
h=legend('CASE A','CASE B','CASE C');
set(h,'Position',[0.2,0.65,0.3,0.22]);
axes(h)
print -depsc fig4
```

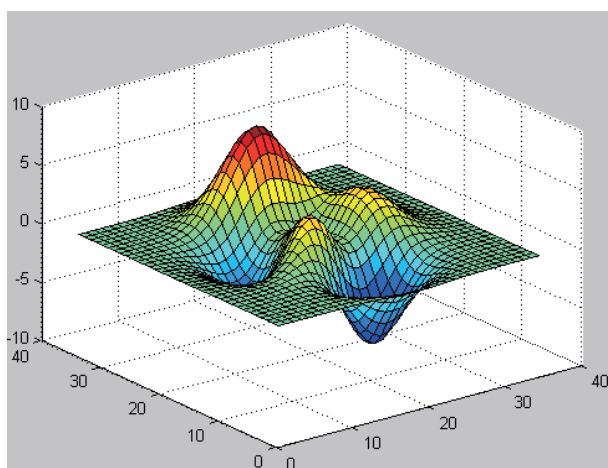


図 5

```
surf(peaks(35))
print -depsc fig5
```

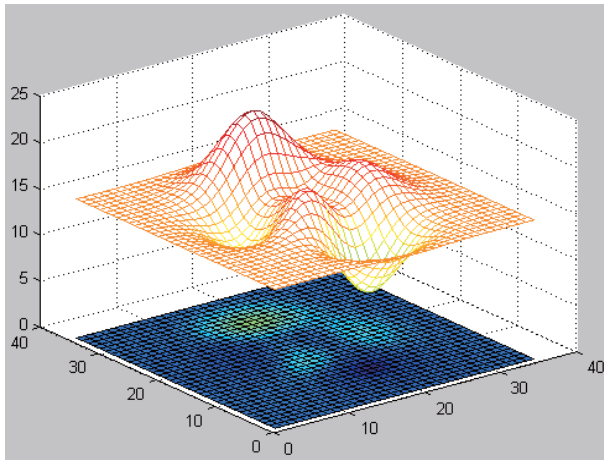


图 6

```
mesh(peaks(35)+15)  
hold  
pcolor(peaks(35))  
hold off  
print -depsc fig6
```

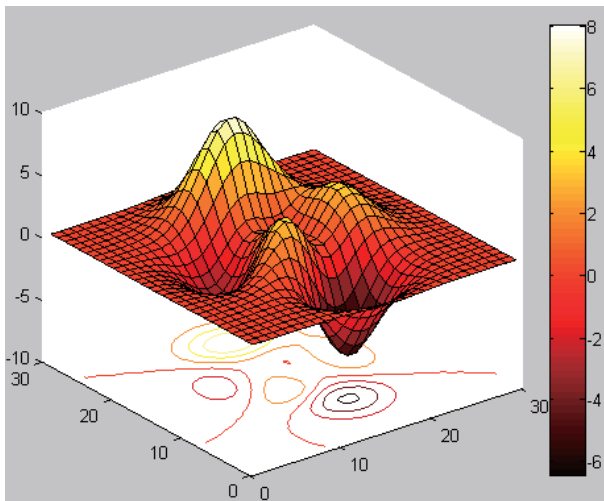


图 7

```
surf(peaks(30))  
colormap(hot)  
colorbar('vert')  
grid  
print -depsc fig7
```

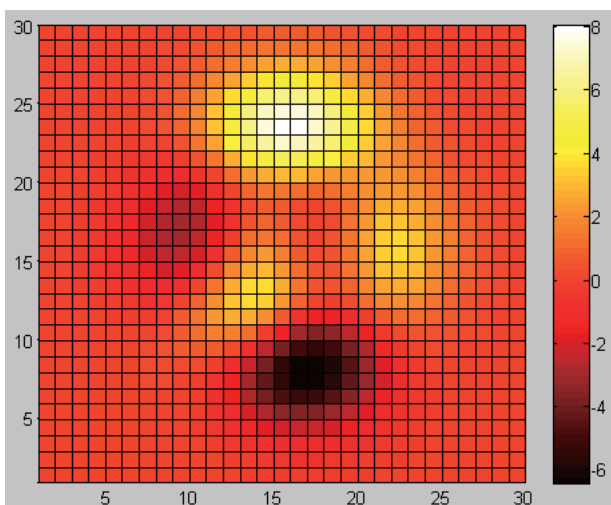


图 8

```
pcolor(peaks(30))  
colormap(hot)  
colorbar('vert')  
print -depsc fig8
```

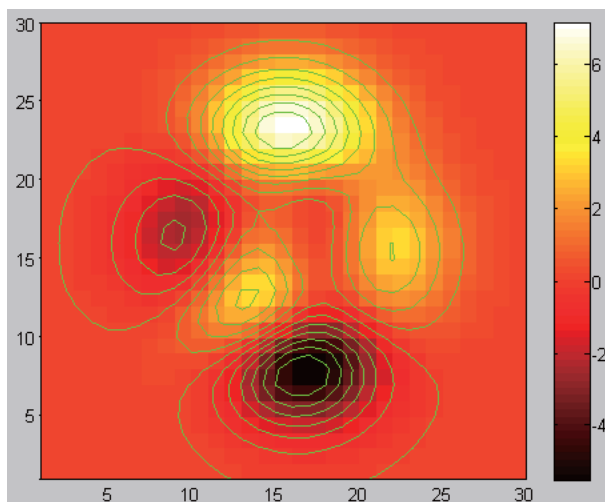


図 9

```

pcolor(peaks(30))
colormap(hot)
shading flat
hold on
contour(peaks(30),15,'g')
hold off
colorbar('vert')
print -depsc fig9
    
```

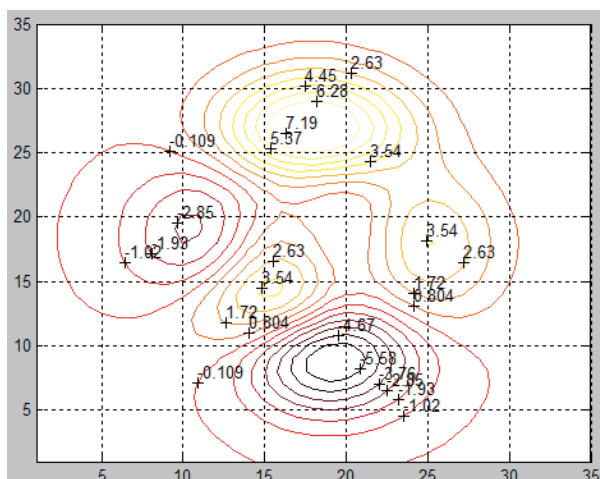


図 10

```

b=contour(peaks(35),15);
clabel(b)
grid
print -depsc fig10
    
```

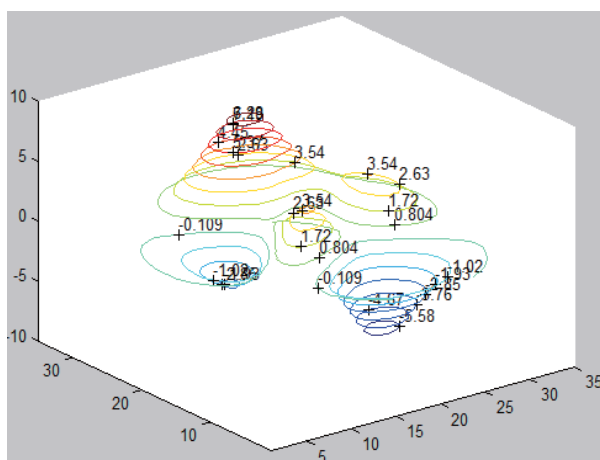


図 11

```

b=contour3(peaks(35),15);
clabel(b)
grid
print -depsc fig11
    
```

6.5 視点

3次元グラフィックスの表示において重要な、視点の角度の調節には `view` というコマンドを用います。
`view(Azimuth,Elevation)` コマンドは、水平方向と垂直方向の回転角を図 1 2 に示すように指定することにより、視点の角度を調節します。

例えば、図 13 は `speaks` コマンドにより得られた行列を異なる 4 つの視点から表示したものです。

```
view(-37.5,30) view(-7,80)
view(-90,0) view(-7,-10)
```

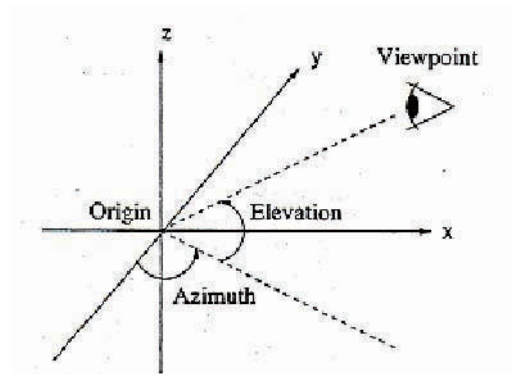


図 1 2 : 3次元プロットの視点角度の設定

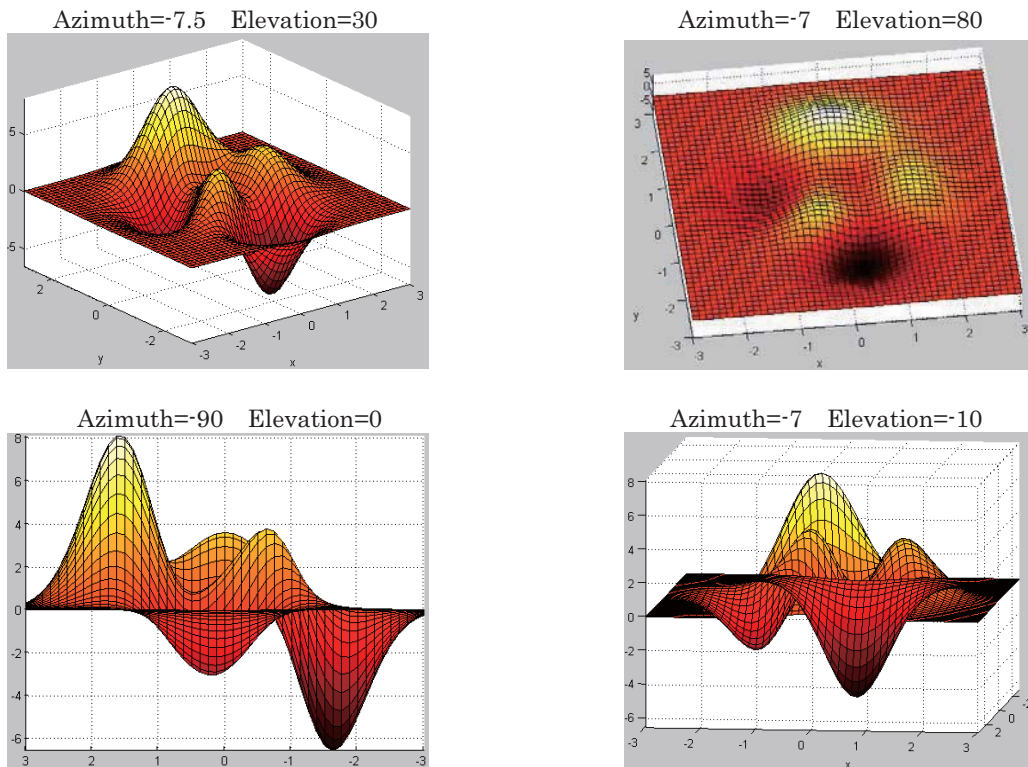


図 1 3 : peaks 関数による 3次元グラフィックスを異なる 4つの視点から表示した例

7 まとめ

本稿では、高性能な数値計算機能と多彩な可視化機能を備えた技術計算ソフト **MATLAB** の入門として、その特徴、数値計算の指示方法、グラフィックスの作成方法などを紹介しました。今回紹介したものは、**MATLAB** の利用法のほんの一部分にすぎません。是非一度、実際に自分で利用して、その豊富な機能と高い生産性を体験していただきたいと思います。なお、マニュアルがサイバーサイエンスセンター 1 階のプログラム相談室に設置してありますので、使い方ももっと詳しく知りたい方はそちらをご覧ください。

参考文献

- [1] MATLAB による制御理論の基礎：野波健蔵，東京電機大学出版局
- [2] MATLAB による制御のためのシステム同定：足立修一，東京電機大学出版局
- [3] だれでもわかる MATLAB：池原雅章，培風館
- [4] はやわかり MATLAB 第 2 版：芦野隆一，共立出版
- [5] 最新 MATLAB ハンドブック第 3 版：小林一行，秀和システム
- [6] MATLAB グラフィックス集：小国 力，朝倉書店
- [7] MATLAB と利用の実際：小国 力，サイエンス社
- [8] MATLAB の総合応用：高谷邦夫，森北出版
- [9] 最新使える！MATLAB：青山貴伸，講談社
- [10] 使える！MATLAB/Simulink プログラミング：青山貴伸，講談社
- [11] MATLAB による画像&映像信号処理：村松正吾，CQ 出版
- [12] <http://www.mathworks.co.jp>