

# 第6章 MATLAB のグラフィックスおよびプログラミング I

## 6.1 実験目的

前章では MATLAB 対話形式の環境における MATLAB の操作やファイルの管理、コマンド・関数の利用、データ行列の演算を習得した。本章では、MATLAB のグラフィックス機能および MATLAB のコマンドを記述したスクリプトファイルの作成(スクリプト M-ファイルプログラミング)を習得することを目的としている。

## 6.2 実験内容

### 6.2.1 MATLAB のグラフィックス

Matlab のグラフィックス機能は非常に豊富で、簡単なコマンド・関数のみで 2、3 次元の図形を作成することができる。グラフィックスの関数は通常デフォルト値が設定されており、関数に適切なデータ行列を与えれば、自動的に図形を作成できる。また、図形に注釈を付けたり、図形を印刷したりする関数も用意されている。

### 1.2 次元のグラフィックス

1) **プロットの関数とオプション(グラフの色、線のスタイルとマーカ)** 2次元のグラフィックス関数には、基本タイプの関数と特殊タイプの関数がある。明確的に区別されていないが、基本タイプの関数は曲線的な図形を生成するための関数である。一方、特殊タイプの関数は棒グラフ、面積図、散布図、アニメーションなどより効果的な表現力がある図形を生成するための関数である。代表的な関数の例を次表に示す。

基本的な 2 次元プロット関数		2 次元グラフ関数	
plot	線形プロット	Stem	離散データの表示
semilogx	X 片対数プロット	Quiver	矢印図
semilogy	Y 片対数プロット	Scatter	散布図
loglog	両対数プロット	Image	イメージの表示
plotyy	左右両軸プロット	Hist	ヒストグラム

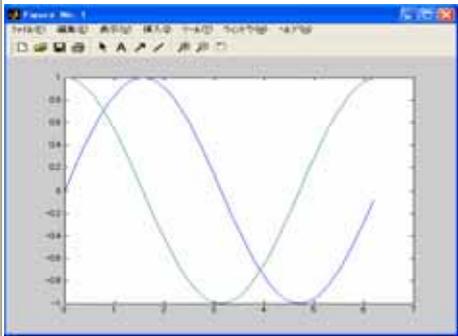
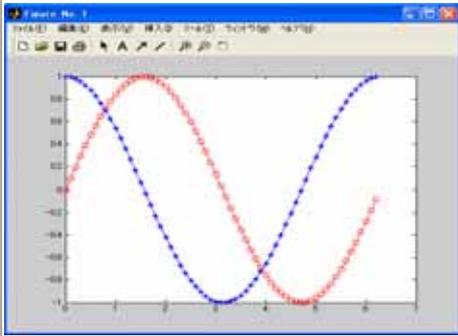
polar	極座標プロット	Contour	等高図
-------	---------	---------	-----

次に代表的な2次元ラインプロット関数 plot について説明する。

plot(x1,y1,s1,x2,y2,s2,x3,y3,s3,...)関数は3種類の引数(x,y,s)がある。ここでは簡単に、x1,y1,s1 のみの場合について説明する。x1,y1 はそれぞれ横軸と縦軸に対するデータ組で、s1 はx1,y1 データ組に対して、プロットされている波形のカラー、タイプ、マーカの3つのオプションを指定するための引数である。3つめの引数には、次表に示すような文字列 Color LineStyle Marker の識別子を用いる。

色とその識別子		マーカとその識別子		ラインスタイルとその識別子	
B	青	.	点	-	実線
G	緑	o	円	:	点線
R	赤	x	x印	-.	鎖線
C	シアン	+	プラス印	--	破線
M	マゼンタ	*	星印		
Y	黄	s	正方形		
W	白	d	ダイヤモンド		
K	黒	v	三角形(上向き)		
		^	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		

**例題 6\_1:** 一周期分の正弦波と余弦波を同一ウィンドウにプロットしてみよう。

プロットコードと説明文	プロットの結果
<pre>&gt;&gt; clear          %Workspace 中の全ての変数の削除 &gt;&gt; x=0:pi/31.4:2*pi;%入力引数 x にデータを与える &gt;&gt; y1=sin(x);     %正弦波の計算結果を y1 に &gt;&gt; y2=cos(x);     %余弦波の計算結果を y2 に &gt;&gt; plot(x,y1,x,y2) %x,y1 と x,y2 のプロット</pre>	
<p><b>オプションの設定例：</b></p> <ol style="list-style-type: none"> <li>1 . 線の色： y1 が赤 ( r )、y2 が青 ( b )</li> <li>2 . 線のタイプ： y1 が実線 ( - )、y2 も実線 ( - )</li> <li>3 . マーカのタイプ： y1 が円 ( o )、y2 が星形 ( * )</li> </ol>	
<pre>&gt;&gt; plot(x,y1, 'r-o',x,y2, 'b-*')</pre>	<pre>r-o : 赤い_実線_円 b-* : 青い_実線_星印</pre>

## 2) グラフの軸制御と注釈

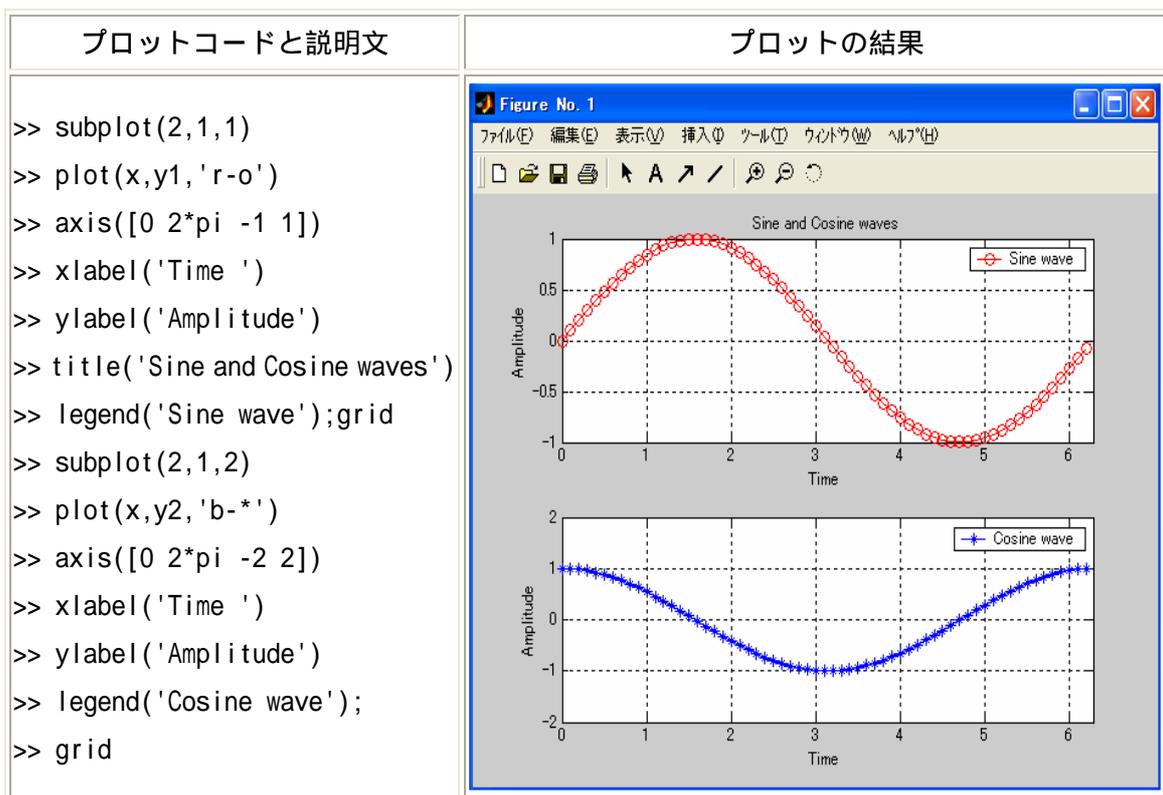
プレゼンテーション用などの見栄えのよいグラフを作成する場合には、波形のスタイルやカラーだけでなく、グラフに注釈を加えたり、座標軸のスケール変更と装飾を付けたりすることが必要となる。このために、次表で示すような装飾関数を利用することができる。

基本的なグラフ装飾関数

grid	グリッド	title	タイトル
axis	軸スケールの変更	xlabel	x 軸ラベル
subplot	複数の座標軸の作成	ylabel	y 軸ラベル
legend	凡例	text	テキストを表示

MATLAB は、上述の基本的な関数を利用してグラフを装飾するほかに、グラフのウィンドウにあるオブジェクトのプロパティを利用して同様な装飾の操作が可能である。これについての詳しい説明は、MATLAB の help を用いて参照すること。

**例題 6\_2.** 上記の表に示している軸制御と装飾のコマンドを使って、例題 4\_8 の正弦波と余弦波を同一ウィンドウの異なる座標系にプロットしてみよう。なお、コマンド `subplot(m,n,i)` は、Figure ウィンドウの  $m$  行  $n$  列  $i$  番目の座標系を表している。`axis([ xmin xmax ymin ymax ])` は各座標系の最小値と最大値を設定するために用いられる。



### 3) グラフの印刷とファイルへの保存

MATLAB では、グラフの印刷とファイルへ出力の方法は多種多様である。例えば、Matlab のメニューから「印刷」を選択して図形を印刷することができる。また、ワードプロセッサのようなアプリケーションで使えるグラフィックスフォーマットのファイルに出力することもできる。ここで、コマンド `print` を使用する場合のプリンタへの印刷、ファイルへの出力について説明する。

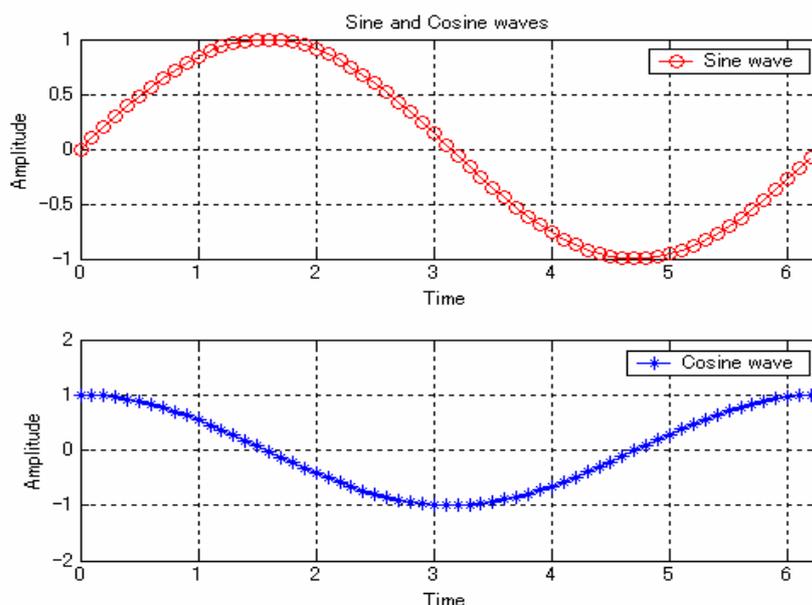
`print` の構文は `print -device -options filename` である。構文中にオプション `-device` と `-options` が省略された場合には、Matlab のデフォルト値が用いられる。例えば、次のコマンドは設定されているデフォルト値で動作される。

>> print	現時点のグラフウィンドウをプリンタで印刷する。
>> print myfile	現時点のグラフウィンドウをファイル myfile に出力する。

プリンタのデバイス(tiff や PostScript)や、プリントされるファイルの特性(解像度)を特に指定したい場合には、オプション-device と-options を利用することができる。例えば、オプション-device を -dbitmap に指定すると、Windows のビットマップフォーマットのファイルが生成される。オプション-device と-options の指定について詳しい説明は `>> help print` で参照すること。

**例題 6\_3:** 例題 6\_2 でプロットされた図形をビットマップフォーマットのファイルで保存し、Windows のワードプロセッサに読み込む手順を示してみよう。

- 1) Matlab の Command Window から、`>> print -dbitmap Exam10` を実行すれば、ビットマップフォーマットの新規ファイル Exam10.btm が作成される。
- 2) Microsoft の Word を開き、メニューの挿入(I)\_図(P)\_ファイル(F)から Exam10 を選ぶと、次の図形が文章に挿入される。



### 2.3 次元のグラフィックス

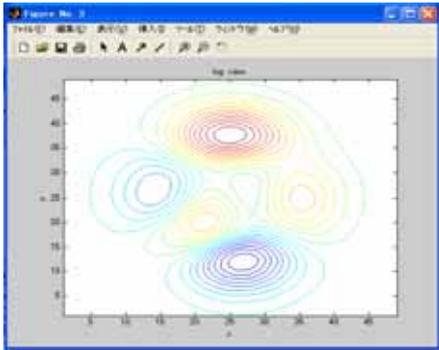
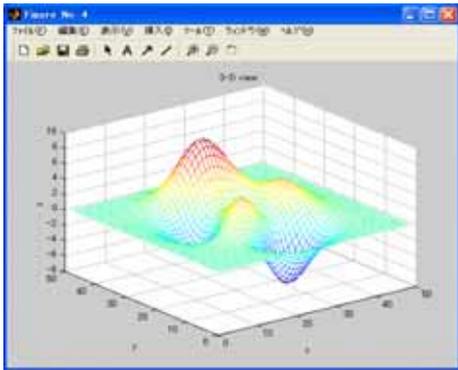
3次元のグラフは2次元のライン図や平面図よりも、立体的な情報の拡大と縮小、図形の透明化、視点の移動などが行えるので、目的に応じてグラフの特徴や外観などを表現することには、非常に便利である。3次元グラフを作成するための基本的なプロット関数を次表に示す。

代表的な 3 次元プロット関数と色指定コマンド

plot3	立体プロット	surf	等高平面図
mesh	立体網図	colormap	カラーマップ
contour3	立体等高図	caxis	カラー軸のスケーリング
quiver3	立体矢印図	shading	影つけ図

**例題 6\_4:** 関数 peaks を利用した 2 次元と 3 次元のグラフを作成してみよう。

関数 peaks を実行すると、大きさが 49 行 49 列の行列が作成される。ここで、配列 z を高さに見なせば、線、平面、立体グラフのプロットは次のようになる。

プロットコードと説明文	プロットの結果
<p>1 ) peaks 関数の読み込み :</p> <pre>&gt;&gt; [x y z] = peaks;</pre>	
<p>2 ) z 平面の等高線図で出力 :</p> <p>等高線数は 20 とする。</p> <pre>&gt;&gt; figure(3); &gt;&gt; contour(z,20) &gt;&gt; xlabel('x');ylabel('y'); &gt;&gt; title('top view')</pre>	
<p>3 ) 3 次元の立体図で出力 :</p> <pre>&gt;&gt; figure(4);mesh(z) &gt;&gt; xlabel('x'); ylabel('y'); &gt;&gt; zlabel('z'); &gt;&gt; title('3-D view')</pre>	

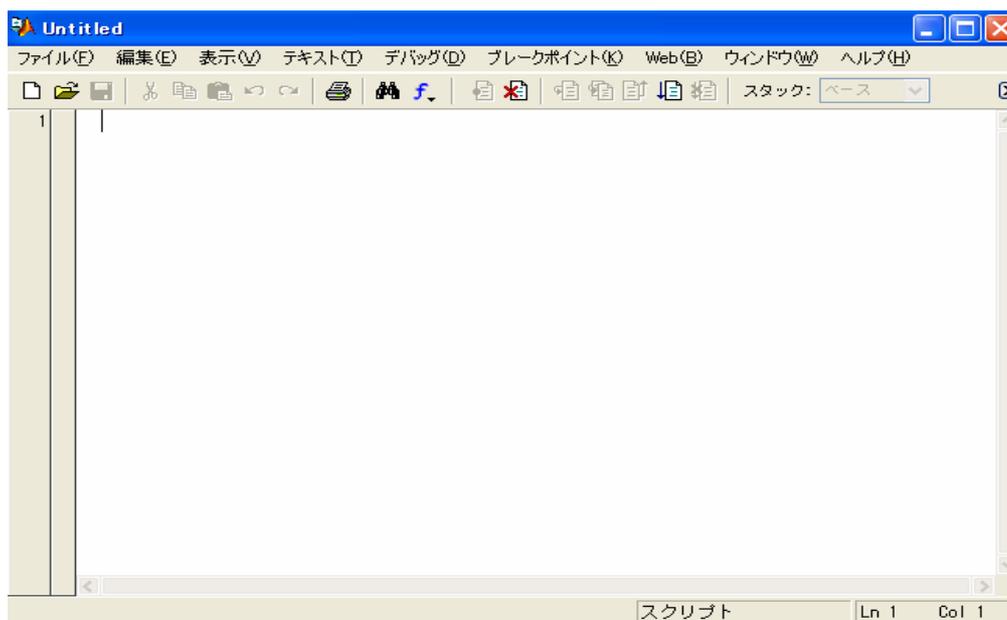
## 6.2.2 MATLAB プログラミングの基礎

### 1. M-ファイル

M-ファイルとは拡張子「.m」が付いている MATLAB 言語のスク립トファイルのことである。Matlab はインタプリタ言語であるため、作成した M-ファイルは Command Window で直ちに実行されることが可能である。M-ファイルの内容は一連実行したいコマンドや関数で構成される。第 5 章で述べたような個々のコマンド・関数を Command Window で逐次に行う代わりに、複数のコマンドを M-ファイルに記述し、一括処理することができる。このことから、Matlab プログラムは複雑な処理や、大規模な演算などが必要とする場合に非常に便利である。

## 2. M-ファイルの作成、編集、保存と実行

新規の M-ファイルを作成するには、Matlab に付随されているエディタツール（デバッグウィンドウ）を利用することができる。エディタを起動する方法は幾つかあるが、例えば、Command Window で `>> edit` とタイプインすれば、次のようなエディタツールの画面が表示される。



上のエディタツールに実行したいコマンドや関数を順に記述し、スクリプトを作成することが出来る。次の図では、スクリプトの作成の一例を示している。

```

1 - clear all
2 - t=linspace(0,2,100);
3 - y1=sin(2*pi*50*t);
4 - y2=cos(2*pi*50*t);
5 - plot(t,y1,t,y2,t,y1+y2,'r-*');
6 - legend('y1 = sin(100*pi*t)', 'y2 = cos(100*pi*t)', 'y1 + y2')
7 - grid
8 - xlabel('Time [s]')
9 - ylabel('Magnitude')
10 - title('Combination of Sine and Cosine Waves')

```

作成した M-ファイルを保存する場合には、メニューからファイル(F)\_別名で保存(A)を選択し、名前を付けてデスクトップにある **MYWORK** フォルダに保存する。

フォルダ **MYWORK** にあるプログラムを実行させる前に、まず Mat lab のパスが **MYWORK** とリンクさせる必要がある。リンクの作成方法は、第 4 章で述べたように Command Window にあるパスの参照ボタン<...>をクリックし、デスクトップにある **MYWORK** フォルダを選択する。

プログラムの実行方法は幾つかあるが、例えば、エディタツールのメニューからデバッグ(D)\_実行(R)を選択して実行する。

プログラムの実行途中でエラーが生じてしまう場合には、エラーメッセージの指示により、エラーが生じた箇所を修正することができる。また、メニューのブレークポイント(K)\_エラーでストップを利用すれば、エラーの修正が効率的に行えるだろう。

既存のプログラムをエディタツールで開きたい場合には、Command Window で `>> edit filename` とタイプインする。この時、filename に正しいパスが入力されないと、動作しない。

**例題6\_5**： 一般化された正規分布族の確率密度関数は次式のように定義される。

$$p(x) = \frac{\alpha\lambda}{2\Gamma(1/\alpha)} \exp(-|\lambda x|^\alpha) \quad (6-1)$$

上式において、xは変数で、λは分散を正規化する係数である。また、Γ(·)はガンマ関数と

呼ばれ、次のように定義される。

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} \exp(-t) dt \quad (6-2)$$

このガンマ関数 $\Gamma(\cdot)$ を計算するため、Matlabでは`gamma()`関数が用意されている。また、 $\alpha$ は分布の形を制御する係数である。例えば、 $\alpha = 2$ の時の分布はよく知られている正規分布(Gaussian)である。また、 $\alpha < 2$ の分布族は裾が重い分布の一族(Super-Gaussian)で、逆に $\alpha > 2$ のとき、裾が軽い分布の一族(Sub-Gaussian)になる。

ここで、確率密度関数 $p(x)$ と変数 $x$ の関係をグラフ化するプログラムを作成し、実行してみよう。但し、正規化係数は $\lambda = 1/\sqrt{2}$ とする。分布の形を制御する係数 $\alpha = 1, 2, 3$ とする。  
 $x$ の範囲は $x=-5:0.1:5$ とする。

```
プログラム Samp6_5.m

% The generalized Gaussian distributions
% filename: Samp6_5.m
% Jianting Cao, Feb. 10, 2003

% Workspace 変数をすべて削除
clear all

% 変数 lambda, x の定義
lambda = 1/sqrt(2);
x = -5:0.1:5;

% alpha=1, super-Gaussian 分布の計算
alpha = 1;
p1 = (alpha*lambda)/(2*gamma(1/alpha))*exp(-abs(lambda*x).^alpha);

% alpha=2, Gaussian 分布の計算
alpha = 2;
p2 = (alpha*lambda)/(2*gamma(1/alpha))*exp(-abs(lambda*x).^alpha);

% alpha=3, sub-Gaussian 分布の計算
alpha = 3;
p3 = (alpha*lambda)/(2*gamma(1/alpha))*exp(-abs(lambda*x).^alpha);
```

```

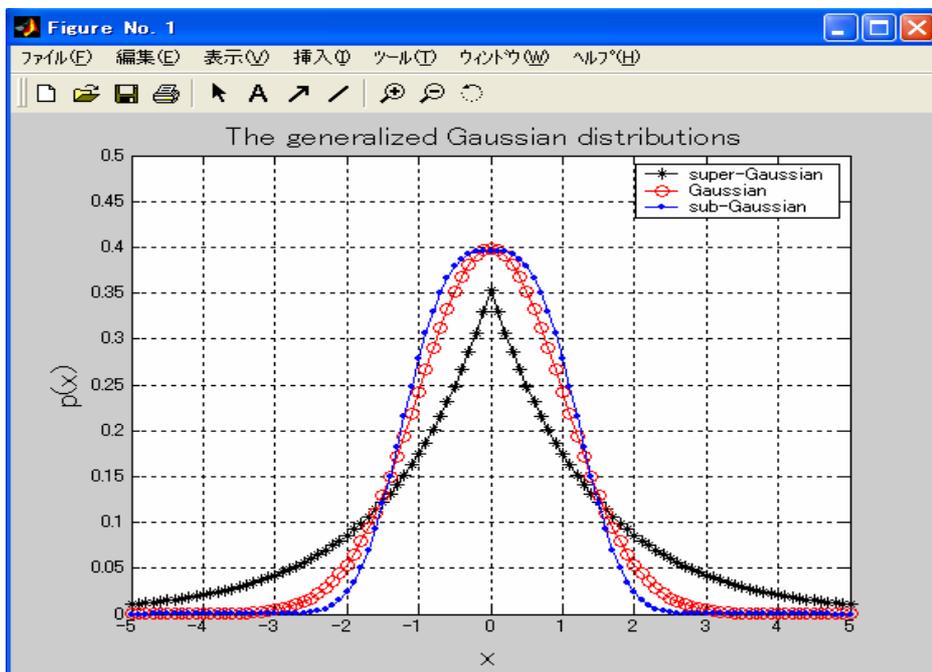
% プロットと修飾
plot(x,p1,'k*-',x,p2,'ro-',x,p3,'b.-');
title('The generalized Gaussian distributions ','FontSize',14);
ylabel('p(x)','FontSize',16);
xlabel('x','FontSize',16);
legend('super-Gaussian', 'Gaussian', 'sub-Gaussian');
axis([-5 5 0 0.5]);
grid;

```

作成したプログラムを実行するには、Command Window で次のファイル名を入力するだけである。

```
>> Samp6_5
```

これを実行すると、次の処理結果が得られる。

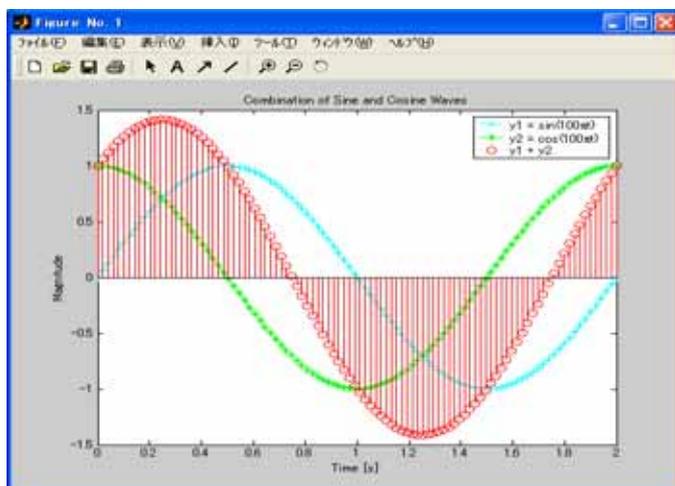


### 6.3 実験課題

1. 次表に示すプロット関数から任意の関数を選んで二つのプロット例を作成し（データが各自で作ること）、結果を示せ。（なお、必要に応じて関数の説明を `>> help 関数名` で参照しその機能を把握すること）。

基本的な2次元プロット関数		2次元グラフ関数	
<code>semilogx</code>	X片対数プロット	<code>quiver</code>	矢印図
<code>semilogy</code>	Y片対数プロット	<code>scatter</code>	散布図
<code>loglog</code>	両対数プロット	<code>Image</code>	イメージの表示
<code>plotyy</code>	左右両軸プロット	<code>Hist</code>	ヒストグラム
<code>polar</code>	極座標プロット	<code>contour</code>	等高図

2. 正弦波  $y_1 = \sin(2\pi \cdot 50 \cdot t)$ 、余弦波  $y_2 = \cos(2\pi \cdot 50 \cdot t)$  とそれらの合成波  $y_1 + y_2$  を下に示すような図を参考に各自で `plot()` と `stem()` 関数を用いて作成せよ。作成した図形をビットマップフォーマットで保存し、レポートに含めよ。



(ヒント:)

`t=linspace(0,2,100)` とする。  
また、異なる関数で複数の図形を同時に表示したい場合には、2 個目以後の関数を実行する前に、必ずコマンド `hold on` を実行する必要がある。また、一番最後の `plot` 関数を実行し後に、`hold off` で元の設定に戻す。

3. 自由度  $\beta$  である t-分布の確率密度関数は

$$p(x) = \frac{\lambda \Gamma(\frac{\beta+1}{2})}{\sqrt{\pi\beta} \Gamma(\frac{\beta}{2})} \left(1 + \frac{(\lambda x)^2}{\beta}\right)^{-\frac{\beta+1}{2}}$$

で定義される。 $\lambda$  は分散を正規化する係数である。また、 $\Gamma(\cdot)$  はガンマ関数である。t-分布族 ( $0 < \beta < \infty$ ) は裾が重い形をしており、 $\beta$  が無限大につれ正規分布に近づく。これ

は一般化された正規分布族  $\alpha < 2$  の場合に類似する。即ち、t-分布族は正規分布(Gaussian)と裾が重い分布族(Super-Gaussian)だけを含んでいるが、裾が軽い分布族(Sub-Gaussian)を含んでいない。

ここで、係数は  $\lambda = 1$  ,  $\beta = 1, 3, 100$  ,  $x = -5:0.1:5$  とする。  $x \sim p(x)$  の関係を表すグラフのプログラムコードを作成し、プログラムコードと実行した結果を示せ。

参考のために実行結果の一例を下図に示す。時間的に余裕がある場合は更に  $(x, \beta) \sim p(x, \beta)$  の関係を表す3次元のグラフを作成せよ。

