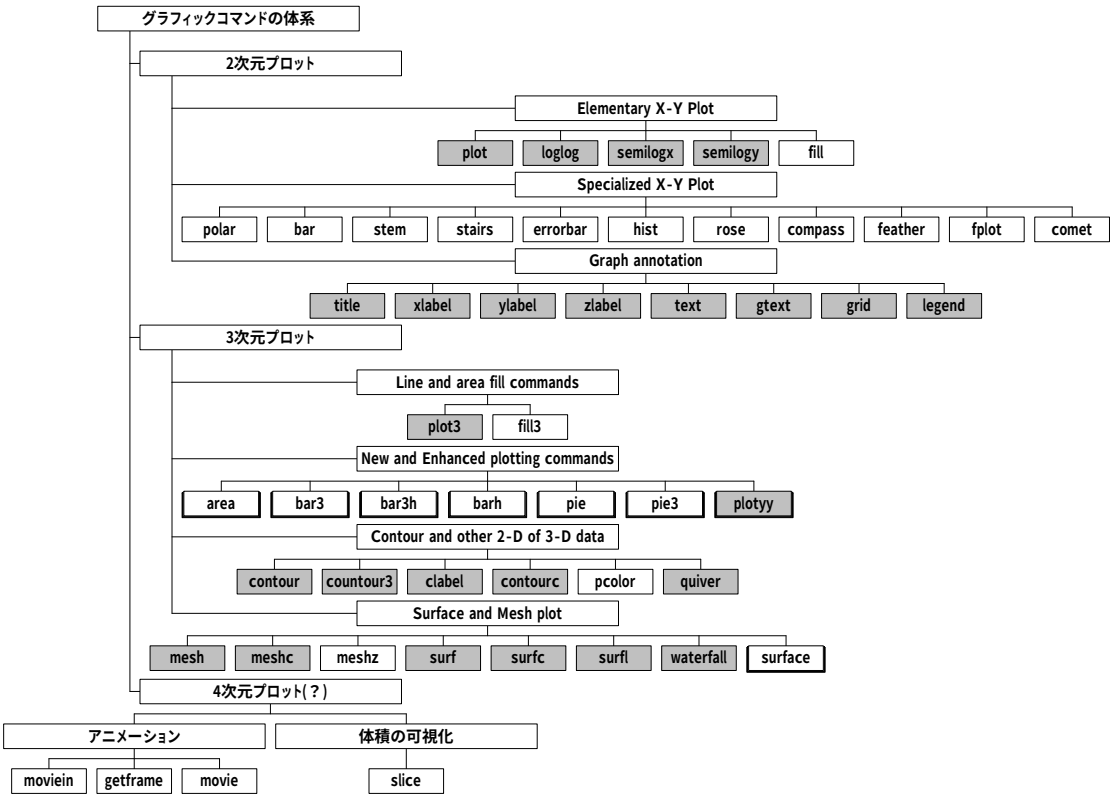
MATLAB(2)

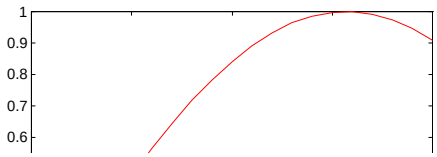
MATLAB グラフィックコマンド体系

MATLABでは、データをビジュアライズするための多様なグラフィックス関数コマンドを用意している。ここでは、代表的ないくつかのグラフィックス関数コマンドの使い方について勉強していく.



2 次元グラフィックス

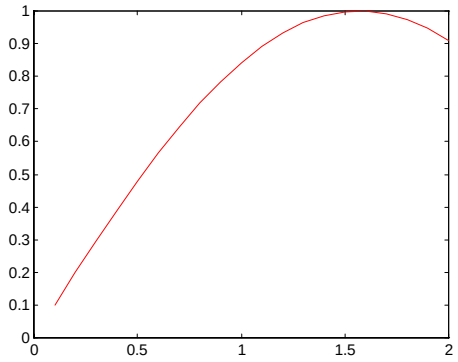
plot 関数



plot関数はMATLABで比較的よく使う数値データをビジュアライズするための関数コマンドの1つである。MATLABの関数コマンド全般に言えることであるが、入出力引数の数により、出力される表示、または数値が変わる. plot関数も例外ではない. ここでは具体的な例を示しながら解説していく.

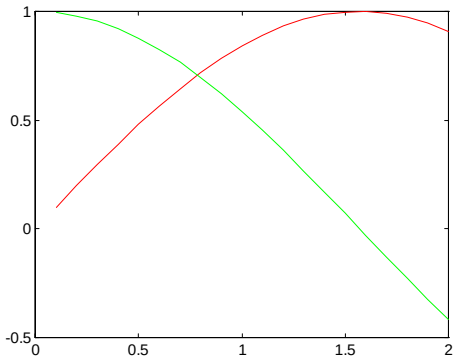
例えば最も簡単な使い方では入力引数が__の場合、

```
>>x=(1:20)*0.1;
>>y=sin(x);
>>plot(y)
```



入力引数（ここでは変数y）が一つの場合には、x軸は変数yの長さに自動的に割り当てられる。もしx軸も1から20までではなく実際と合わせた形0.1から2という単位にしたい場合には以下のように書く。

```
>>_____
```



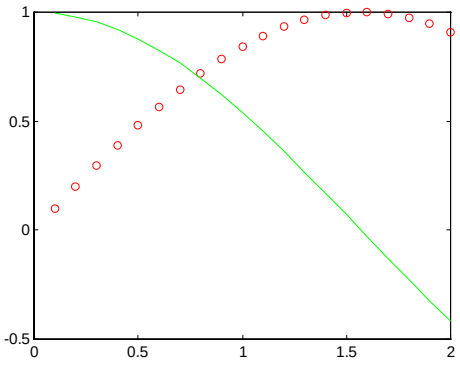
また、__つ以上のグラフをオーバーラップして書きたい場合には、

plot(x1,y1,x1,y2,x1,y3,...)

といった具合に書く、2つの場合には

```
>>plot(x,y,x,cos(x))
```

ラインスタイル、マーカや色について



MATLABのplot関数では、_____や、色やマーカを変更することができる。例えば、

yの線を線でなく__にそして、cos(x)を破線にしてみよう。

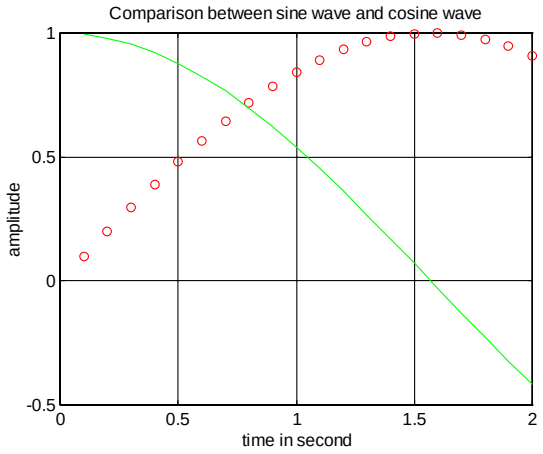
このような場合には

```
>>plot(x,y,'o',_____)
```

このように Plot ラインのタイプには以下のものが定義されている。

y	黄	m	紫(magenta)	c	水色（cyan）
r	赤	g	緑	b	青
w	白	k	黒(black も OK)	.	ポイント
o	丸	x	×マーク	+	＋マーク
*	＊マーク	-	線	:	点線
-.	一点鎖線	--	破線		
s	□マークd	◇	マーク	v	▽（下向き）
^	△（上向き）	>	三角（右向き）	<	三角（左向き）
p	☆マーク	h	△マーク		

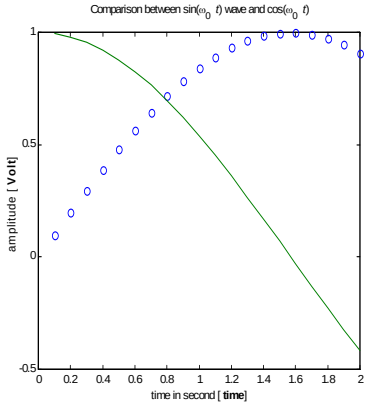
グリッドやラベルの追加



_____コマンドは、plot図に格子を引く関数コマンドである。また、plot図のx軸、y軸にラベルをつけたい場合には、_____、_____、それに_____関数がそれぞれ用意されている。
例えば上の例にラベルとタイトルをつけて表にしてみよう。

```
>> plot(x,y,'o',x,cos(x),':')
>>grid
>>xlabel('time in second')
>> ylabel('amplitude')
>> title('Comparison between sine wave and cosine wave')
```

となる。



文字列の修飾にはTeXライクなコマンドを使うことができる．例えば、タイトルやラベルの中でギリシア文字や、上付き文字下付き文字をあらわすと

```
》   plot(x,y,'o',x,cos(x),':')
》  title('Comparison between sin(\omega_0 {\it t}) wave and cos(\omega_0 {\it t})')
》  xlabel('time in second [{\bf time}]')
》  ylabel('amplitude [{\bf Volt}] ')
```

また、legend コマンドを使うとプロットした線に対してのコメントが入れることができる。

```
>>legend('sin(x)','cos(x)')
```

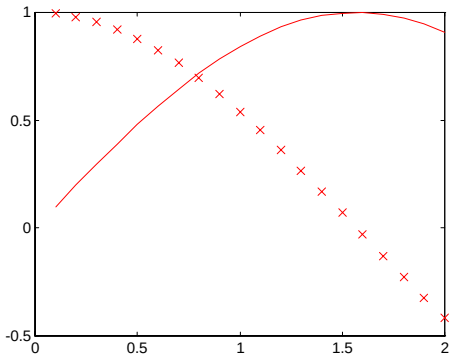
プロットのコメントの位置は、自動的に配置される。都合の悪い位置で表示された場合、マウスを使いドラックし移動もできる。

軸のカスタマイズ

axis([xmin xmax ymin ymax])	軸の最大値最小値を指定する。
axis auto, axis('auto')	データの最小値、最大値を基準にする（default）
axis square,axis('square')	軸の形を正方形にする。
axis equal, exis('equal')	軸のスケーリングファクタを同じにする。
axis off,axis('off')	軸の表示をなくす。
axis on, axis('on')	軸の表示をする。

もし、標準で使用される軸の設定が気に入らない場合には、カスタマイズも可能である。なおaxisの指定は、プロットコマンドの後にすると有効になる。

HOLD コマンド



新しい線を現在のプロットに書き足したい場合には_____コマンドがある。通常、plotコマンドを実行するとカレントのfigureは消されそれに新しいplotが描かれるが、_____コマンドを使った後でplotコマンドで線を表示すると、カレントのfigureは消されずその上にオーバーラップして表示される。

```
>>plot(x,y)
>>hold on
>>plot(x,cos(x),'x')
>>hold off
```

subplot コマンド

信号処理など時折、信号を比較したい場合などがある。線種の違いやカラーによる区別もよいが、例えば周波数プロットと時間軸のプロットなどそれぞれ軸の定義の異なるものを、オーバーラップして表示しても見づらくなるだけである．このような場合に便利な関数コマンドがMATLABの_____である。

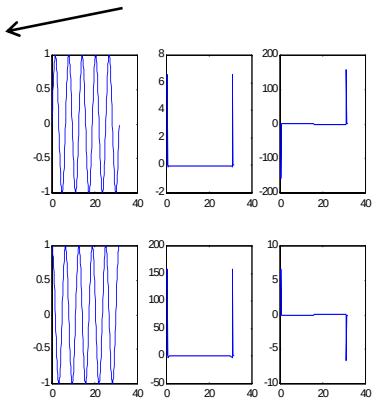
```
>>x=0:0.1:10*pi;
y=sin(x);z=cos(x);ffty=fft(y);fftz=fft(z);
subplot(2,3,1);
plot(x,y);
subplot(2,3,2);
plot(x,real(ffty));
subplot(2,3,3);
```

時間軸

周波数軸

(2,3,1)	(2,3,2)	(2,3,3)
(2,3,4)	(2,3,5)	(2,3,6)

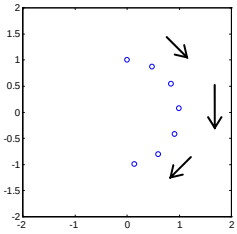
```
plot(x,imag(ffty));
subplot(2,3,4);
plot(x,z);
subplot(2,3,5);
plot(x,real(fftz));
subplot(2,3,6);
plot(x,imag(fftz));
```



サブプロットのレイアウト
まず、列を数えそれから行を数え

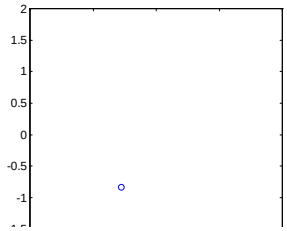
plot コマンドを使ったアニメーション

plot コマンドをうまく利用すると簡単なアニメーションを作ることができる。
ここでは、丸いドットが円運動するようにしてみよう。



```
close all;figure(1);for x=0:0.01:10;
plot(sin(x),cos(x),'ob');
axis([-2 2 -2 2]);axis('square');
drawnow;
end
```

ここではaxisコマンドによって座標軸設定を固定、軸表示を標準の横長から正方に再設定している。_____コマンドは、グラフィックバッファに入った情報をすべて画面表示しなさいといった命令である。この命令がない場合、グラフィックバッファに書き換えた情報が入るが、その状態変化はend文が実行されるまで表示されない。つまりアニメーションされず結果のみ表示される。____コマンドを入れることによりループ毎にグラフが表示されるが、毎回、座標軸がフラッシュされ、ちらつき現象がおきる。スムーズなアニメーションを行うためには、以下のようなテクニックを使用すると良い。



```
close all;figure(1);set(1,'doublebuffer','on');
for x=0:0.01:10;
plot(sin(x),cos(x),'ob');
axis([-2 2 -2 2]);axis('square');drawnow;
end
```

このスクリプトを前述のスクリプトとの違いは、set(1,'doublebuffer','on');の所のみの違いである。この命令をいれることで、ちらつき現象を防ぎスムーズなアニメーションを実現できる。この機能を使うことで、データの動きのビジュアライゼーションなどが容易にできる。

3 次元グラフィックス

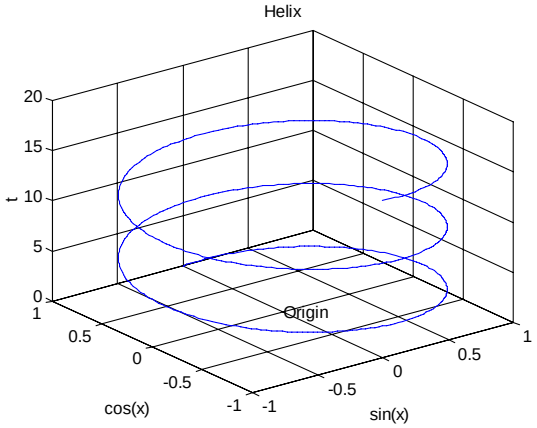
MATLAB には様々な種類の 3 次元グラフィックスをサポートしている。
3 次元で直線を表現するコマンド、面を表現するコマンド、ワイヤーフレームを表現するコマンドなどがある。

plot3 関数

plot3関数は、plot関数の3次元拡張版である。書き方は、z軸方向の情報を追加する以外はほぼplot関数と同じである。一般的なplot3の文法は、

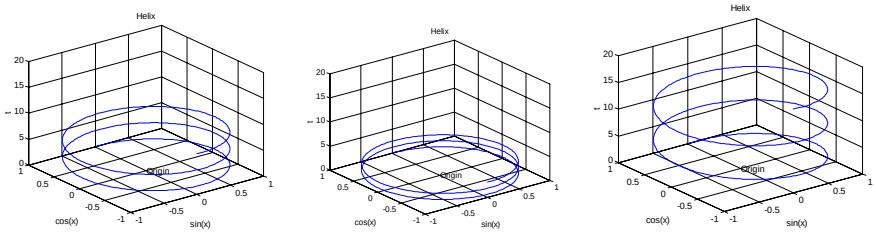
plot3(x1,y1,z1,s1,x2,y2,z2,s2,....)となっている。ここでxn,yn,znは、座標を示すベクトルか行列である。snは、通常省略可能である。plotと同様にラインスタイルの設定などを行うことができる。このplot3は、一般に3次元の1つの変数を表示するのに適している。以下に簡単な3次元プロットの例を示す。

```
>> t=0:pi/100:5*pi;abc=plot3(sin(t),cos(t),t,'b');title('Helix');xlabel('sin(x)');ylabel('cos(x)');zlabel('t')
>> grid;text(0,0,0,'Origin');
```

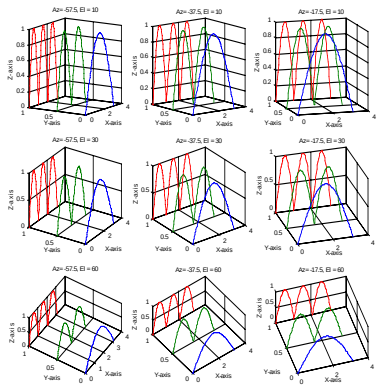


このように簡単に 3 次元のプロットが描く事ができる。
先ほどのテクニックを使うと 3 次元プロットでも同様にスプリングが大きくなっていくようなアニメーションも作れる。

```
close all;figure(1);set(1,'doublebuffer','on');
t=0:pi/100:5*pi;
for k=0.1:0.01:1
plot3(sin(t),cos(t),k*t,'b');axis([-1 1 -1 1 0 20]);
title('Helix');xlabel('sin(x)');
ylabel('cos(x)');zlabel('t');grid;text(0,0,0,'Origin')
drawnow;end
```



ビューポイントの移動
plot3 では、デフォルトのビューポイントを Azimuth=_____度 Elevation=30 度に固定してある。3 次元のグラフは、見るポイントによってかなりイメージが変わる。以下に view を使用し様々な角度から見た例を示す



```
X=linspace(0,pi).'; % generate 100 points of data from 0
to pi
Z=[abs(sin(X)) abs(sin(2*X)) abs(sin(3*X))];
Y=[zeros(size(X)) ones(size(X))/2
ones(size(X))];count=1;
for xx=[10 30 60]
   for yy=[-57.5 -37.5 -17.5]
   subplot(3,3,count);count=count+1;
   plot3(X,Y,Z);
   grid;xlabel('X-axis','fontsize',5);
   ylabel('Y-axis','fontsize',5);
   zlabel('Z-axis','fontsize',5)
   strtitle=['Az= ', num2str(yy), ', El = ' ,num2str(xx)];
   title(strtitle,'fontsize',5);view(yy,xx);
   end;end
set(findobj('type','axes'),'fontsize',5);
```

2 変数の 1 出力関数のプロット

plot3 が線のプロットを描くのとは異なりしばしば、2 変数 1 出力の関数のプロットを可視化したい場合がある。つまり

$$z = f(x, y)$$

というような関数のプロットである。MATLAB では、このような関数をプロットする場合、x、y はアレイの情報、z は行列の情報を必要とする。それでは、関数を使いそのような変数を作りプロットしてみる。まず、

```
>>x=-3:3
>>y=1:5
[X Y]=meshgrid(x,y)
X =
     -3     -2     -1      0      1      2      3
     -3     -2     -1      0      1      2      3
     -3     -2     -1      0      1      2      3
     -3     -2     -1      0      1      2      3
     -3     -2     -1      0      1      2      3
Y =
      1      1      1      1      1      1      1
      2      2      2      2      2      2      2
      3      3      3      3      3      3      3
      4      4      4      4      4      4      4
      5      5      5      5      5      5      5
```

_____コマンドは、3D プロットの為の X と Y 軸のアレイをつくる役割を果たす。

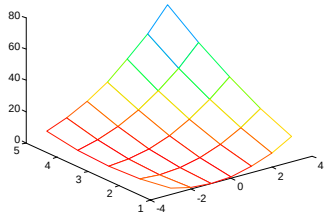
例えば、Z=(X+Y)^2 のグラフを描くためには、

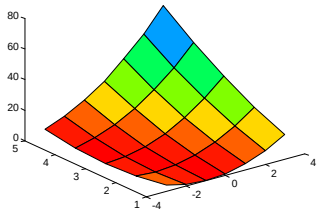
```
>>Z=_________
Z =

      4      1      0      1      4      9     16
      1      0      1      4      9     16     25
      0      1      4      9     16     25     36
      1      4      9     16     25     36     49
      4      9     16     25     36     49     64
```

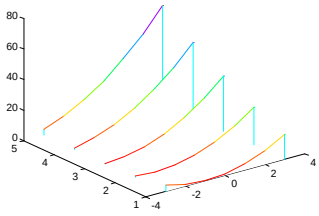
ここでは、二乗を計算するときは_____オペレータを用いる。なぜならばここで行いたい演算は行列演算ではなく要素同士の演算を行いたいからである。

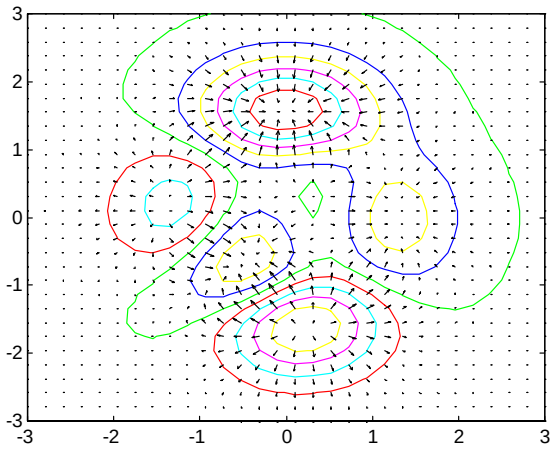
それでは、これらをもとにプロットしてみる。このようなプロット用のコマンドは、_____コマンド、_____コマンド、_____コマンドなどがある。

mesh(x,y,Z)　　surf(x,y,Z)　　waterfall(x,y,Z)









quiver 関数

2 次元の流体解析などに便利な関数として_____関数がある。この関数を使った例は以下の通りである。1 行目では、まず、2 次元データの作成、_____関数で、勾配の計算をしている。つぎに、contour 関数でまず外形を描きグラフをホールドし次に_____関数で個々のベクトルを描いている。

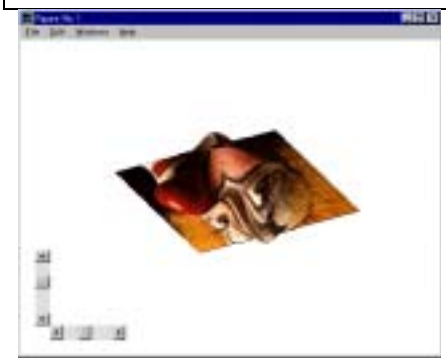
```
>>[X Y Z]=peaks(30);
[DX DY]=gradient(Z,0.5,0.5);
contour(X,Y,Z,10);
hold on
quiver(X,Y,DX,DY)
hold off
```

テクスチャマッピング

MATLABでは、一応、ゲームなどでよく知られている_____もサポートしている。この_____機能を使うには、Handle Graphicsを使いこなす必要がある。（Handle Graphicsについては、次のセクション）ここで示したサンプルプログラムではスライダーを用い表示している3次元グラフのビューポイントを移動することができる。表示速度は、若干遅くなるが非常にパワフルなMALTABのビジュアライゼーションツールとしての能力を示すものである。なおこの中のuicontrolで使用している''は、ダブルクオォーテーションではなく、シングルクォーテーション2つなので入力するときには注意すること。MATLABの文法では、ダブルクォーテーションは使わない．また、引数が多くて一行で書ききれないような場合、適当な個所に”．．．”を入れると複数行に渡り書くことができるようになる．

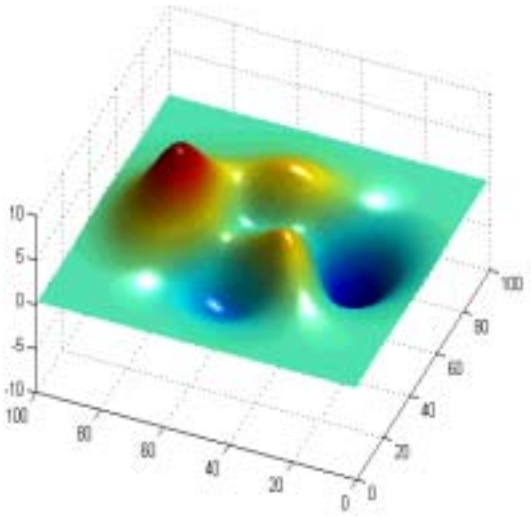
```
load clown
[x,y,z]=peaks(30);
figure('colormap',map)
surfhandle=surf(x,y,z)
set(surfhandle,'Facecolor','texture','cdata',X)
set(surfhandle,'erasemode','background');
set(gca,'drawmode','fast')
```

```
shading flat
u1=uicontrol('style','slider','pos',[40 20 100 20],...
                'max',180,'min',-180,'value',-37.5,...
'callback','AZ=get(u1,''value'');EL=get(u2,''value'');view(AZ,EL)');
u2=uicontrol('style','slider','pos',[20 40 20 100],...
                 'max',180,'min',-180,'value',30,...
'callback','AZ=get(u1,''value'');EL=get(u2,''value'');view(AZ,EL)');
```



light コマンド

Version 5 では、surface や patch に対して、光源の方向の設定ができるようになった。以下の例では、光源の位置が無限遠にあると仮定し、その光源に対するベクトルが[1 -2 1]の方向にあった場合の3D の図形を描いている。また_____コマンドを用いると、標準で3次元の図形をマウスを用いて回転し任意の方向から図形を見ること可能である。



```
》 surf(peaks(100));shading interp;light('Position',[1 -2 1]);rotate3d
```

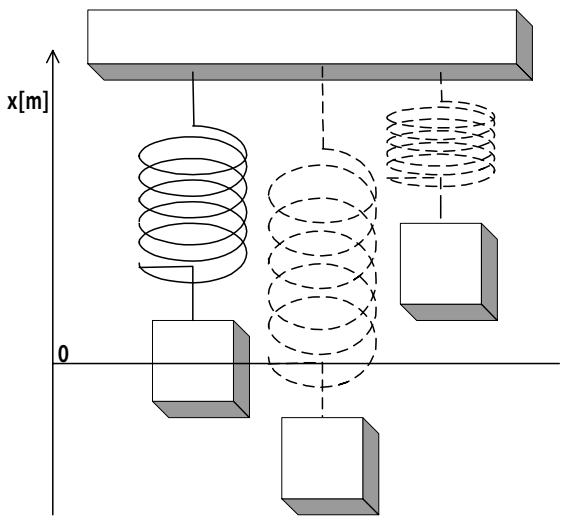
演習問題

課題1

スプリングのアニメーションプログラムを改良し、スプリングが上下に伸び縮みするアニメーションプログラムを実現せよ。その際、スプリングが伸びで縮む動作を1回としてn回伸び縮みするアニメーションを実現せよ。なお回数n回は、学生証番号下1桁の数とし、下1桁が 0 の場合には、10 とする。

課題 2

右図のように原点 O を中心として x 軸上を単振動する重りがある。振幅 0.1[m],振動数 f[Hz]、物体が原点を正の向きで通過する時刻をt＝0とする。空気抵抗は無視するとした時、



1．時刻 t[s]と重りの位置 y[m]
2．時刻 t[s]と重りの速度 v[m/s]
3．時刻 t[s]と重りの加速度 a[m/s^2]
の関係を時刻 0[s]からについてグラフ表示せよ。
ただし、波形は、少なくとも 2 周期以上多くても 10 周期未満で表示せよ。
振動数 f[Hz]の値は、（各自の学生証番号下 2 桁）×0.1＋1とする。
例
00D5001 ならば f=1.1[Hz]
00D5023 ならば f=3.3[Hz]
00D5200 ならば f=1.0[Hz]
MATLAB でシミュレーション表示する場合の注意点
グラフには、縦軸、横軸の物理量、単位を記入することそのほか必要であれば、グラフの線の種類（太、細）や目盛りなどグラフオプションを設定しよりわかり易いグラフになるように工夫すること。