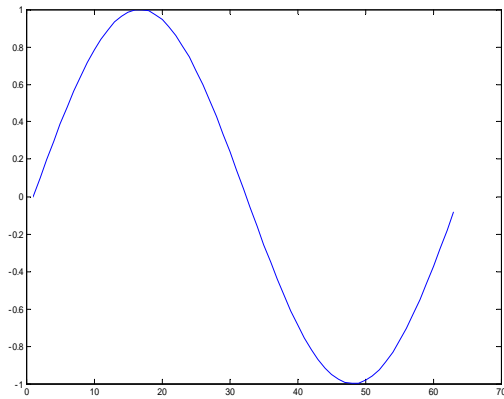


axis 関数による設定変更

axis 関数を使うと細かい設定の制御ができる。ここでは、axis 関数を使った効果的な例をいくつか示す。

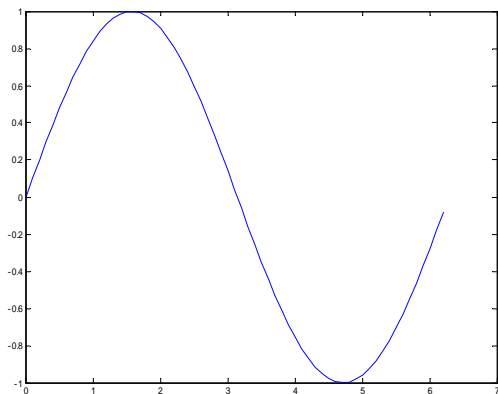
表示範囲の設定

```
>> plot(sin(0:0.1:2*pi))
```



これは、1 周期の sin 波を plot 関数で表示した例であるが、デフォルトの表示では、x 軸が 60 以降が途切れてしまっている。これは、次のように x 軸を指定しても同じである。

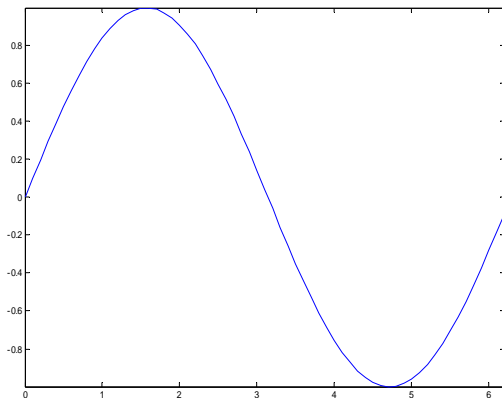
```
>> plot(0:0.1:2*pi,sin(0:0.1:2*pi))
```



このような場合、途切れをなくす有効な関数が axis 関数である。いくつかの axis 関数の使い方があがるが、もっとも簡単な使い方は、tight オプションである。

```
plot(0:0.1:2*pi,sin(0:0.1:2*pi));axis tight
```

また、axis tight の代わりに axis('tight')と記述してもよい。



また、設定は tight 設定で、例えば、y 軸の範囲のみ変更したい場合には、-inf,inf を使うこともできる。MATLAB での inf は無限大のことで

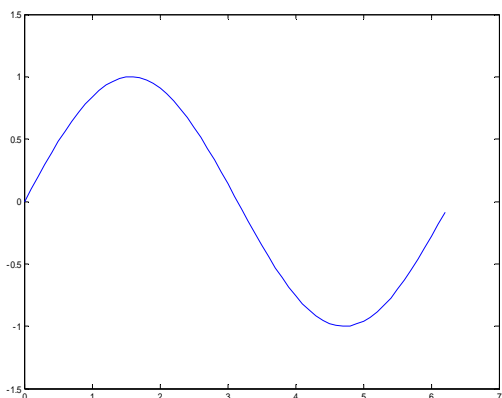
```
plot(0:0.1:2*pi,sin(0:0.1:2*pi));axis([-inf inf -1.5 1.5])
```

とすると、-inf,inf を設定したところはありませんいためデータの点数に合わせて表示する axis tight と同じ設定に、-1.5,1.5 の設定はマニュアル設定に変更することができる。

また、tight 設定ではなく標準の状態である一部のパラメータだけ変更したい場合にはどのようにすればよいのであろうか。

そのような場合には、設定にするとよい。plot 関数でグラフを表示した後、axis 関数により軸の設定をし直している。また、わざとスペースを空けたい場合にも axis 関数を使うと変更できる。例えば、x 軸の設定はそのまま、表示する y 軸のレンジを ±1 から ±1.5 に変更する例を示す。

```
>> plot(0:0.1:2*pi,sin(0:0.1:2*pi));tmpaxis=axis;axis([tmpaxis(1:2) -1.5 1.5])
```



この設定では、まず tmpaxis=axis; で plot 関数の現在の設定を保存している。

axis 関数では、入力を入れた場合の設定は、axis([xmin xmax ymin ymax])

であるので、再度 axis 関数により xmin,xmax の設定は、tmpaxis(1:2)により標準設定を使い、ymin,ymax のみ設定を変更している。

実は、axis 関数を使わず、xlim,ylim 関数を使うとそれぞれの設定も可能である。

```
>> plot(0:0.1:2*pi,sin(0:0.1:2*pi));ylim([-1.5 1.5])
```

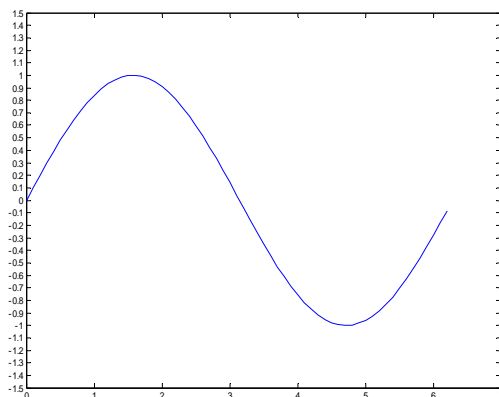
同様のことが set 関数によりさらに細かい設定をすることもできる。

axis 関数での設定と同様の設定を set 関数で実現するには次のようにする。

```
plot(0:0.1:2*pi,sin(0:0.1:2*pi));set(gca,'ylim',[-1.5 1.5]);
```

gca は GetCurrentAxis の略称で、表示された直後のグラフの軸に関する情報が格納されている。この中の 'ylim' プロパティを [-1 1] から [-1.5 1.5] に変更している。この方法は、それ以外の細かいパラメータ、例えば、y 軸の tick の間隔を 0.1 に変更することもできる。

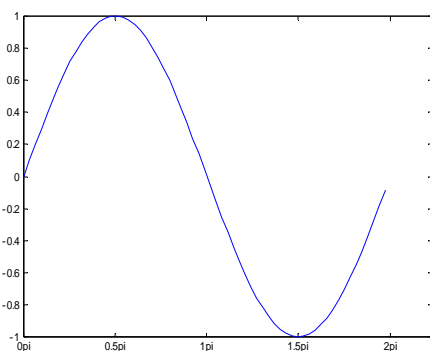
```
>> plot(0:0.1:2*pi,sin(0:0.1:2*pi));set(gca,'ytick',-1.5:0.1:1.5,'ylim',[-1.5 1.5])
```



数字以外の文字を軸に表示するには？

軸の属性の中で XTickLabel, Xtick を変更すると x 軸に数字の代わりに文字を表示することもできる。

```
xx=[];xlabel=0:0.5:2;
% xx=['0pi | 0.5pi | 1pi | 1.5pi | 2pi'];と言う文字列を作りたいのでそれを for 分て書くと
%%%%%%%%%%%%%%
for i=1:length(xlabel);xx=[xx num2str(xlabel(i),'pi | ');end;
xx=xx(1:end-1);% remove last |
%%%%%%%%%%%%%%
plot(0:0.1:2*pi,sin(0:0.1:2*pi));set(gca,'xtick',xlabel*pi,'xticklabel',xx)
```



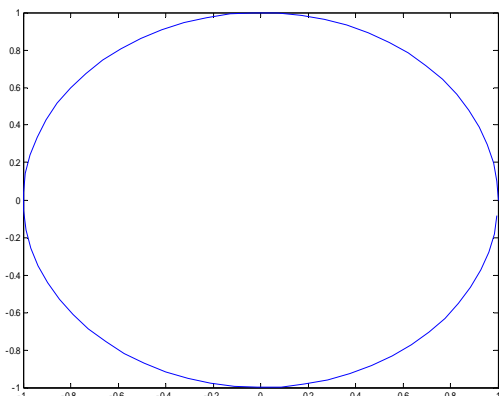
ここでは、0.5pi 間隔で描いているが、このプロパティを使うと、文字、数字を表示する間隔を自由に変更することもできる。

また、同様に YtickLabel, Ytick, ZtickLabel, Ztick を変更しても同様に可能である。これ以外にも軸のスケール、linear, log への変更、軸の数値の順序、昇順、降順への変更、表示する数字のフォント設定など細かい設定を変更することが可能である。詳しい設定などは、set, get 関数によりチェックしてみるとよい。

表示スケールの変更

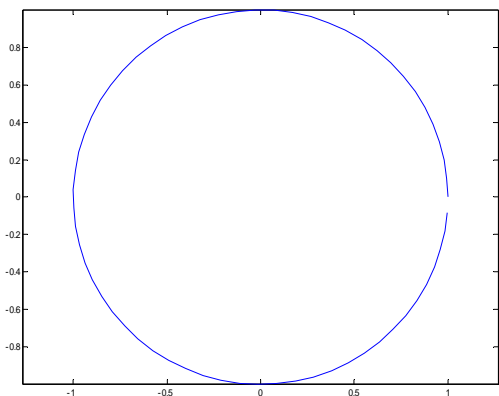
plot 関数のデフォルトの設定では、縦横比のバランスが異なる。例えば、

```
>> plot(cos(0:0.1:2*pi),sin(0:0.1:2*pi));
```



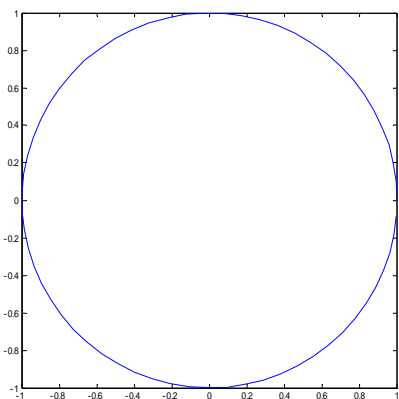
となり、円の表示が横長の楕円になってしまう。そこで縦横比を同一にするために、axis関数の equal オプションがある。

```
>> plot(cos(0:0.1:2*pi),sin(0:0.1:2*pi));axis equal
```



このコマンドを使うことで縦横比が同一になる。また、似たようなオプションで、軸の縦横比を同じにするオプションで axis square がある。

```
>> plot(cos(0:0.1:2*pi),sin(0:0.1:2*pi));axis square
```



このコマンドオプションを使うと、軸の縦横比を同じに保つことができる。