

関数のグラフィック表示

田中雅博

最適化プログラミング

最適化の問題を扱う際に、関数の形を視覚的に見たいと思うことはよくあることであり、それは問題を考える上で大いに参考になる情報である。そこで、今回は、変数が1つ、2つ、3つの問題の関数形を把握するためのグラフィックスについて述べる。

1 1変数の関数

MATLABで2次元プロットと呼んでいるのは、変数が1つの場合である。つまり、 $f = f(x)$ の形の関数である。

1.1 グラフのプロット

MATLABで関数表示をするときは、 x の値をあらかじめギザギザにならない程度の幅で離散的に作っておき、それに対して、 f の値をもとめ、それらを1回の命令でグラフを描くのがよい。

【例】

$$f = x^2 - 4x + 1$$

関数 reil.m

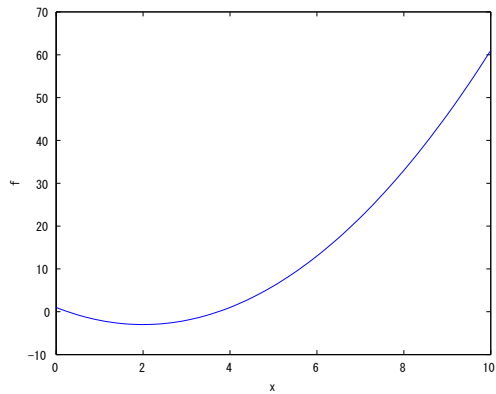
```
function [ f ] = reil( x )
%
%
f=x.^2-4*x+1;
```

end

スクリプト main1.m

```
x=0:0.5:10;
f=reil(x);
plot(x,f)
xlabel('x')
ylabel('f')
```

※データ点(0.50刻み)の部分に正方形のマーカー(□のこと。用語は教科書に準じる)にせよ。



1.2 グリッドラインの表示 (教科書 p.60)

plot したあとに

```
grid on
```

すると、グリッドラインが描かれる。

```
grid minor
```

も試してみよ。

1.3 タイトルと軸ラベル

次の 2 つの方法がある。詳細は教科書参照のこと。

- figure ウィンドウの「挿入」メニューで入力
- コマンドで入力

1.4 複数のグラフをプロット

これも、2 つの方法がある。

- plot コマンドの中に、複数個のデータを入れる。

```
x=0:0.5:10;
f=rei1(x);
f2=rei1(x)*2;
plot(x,f,x,f2)
xlabel('x')
ylabel('f,f2')
```

- plot コマンドを複数個実行する。最初の plot コマンドのあとに、hold on を入れ、最後の plot のあとに hold off を入れること。

```
x=0:0.5:10;
f=rei1(x);
f2=rei1(x)*2;
plot(x,f)
hold on
plot(x,f2)
xlabel('x')
ylabel('f,f2')
hold off
```

1.5 凡例の表示

これも、2つの方法がある。

- GUI で入れる
- legend 関数で入れる

1.6 複数個の座標軸を描く

```
subplot(m,n,k)
```

を実行する。m は縦の分割数、n は横の分割数、k は何番目に該当。

例えば、縦3つ、横2つの6つ描くことにし、その1つ目なら

```
subplot(3,2,1)
```

である。その後、plot コマンドなどを実行すれば、その直前の subplot に従う。

2 2変数の関数

MATLAB で3次元プロット（3次元グラフ）と呼んでいるのは、変数が2つの場合である。つまり、 $f = f(x, y)$ の形の関数である。

MATLAB で用意されているプロットは次のようなものがある。

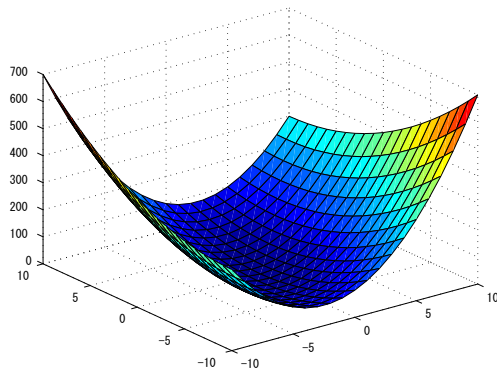
- メッシュプロット
- 表面プロット
- シェーディング付き表面プロット
- 等高線図
- Quiver プロット
- スライス プロット

では、ここで、 $f = 4x^2 - 2xy + y^2$ という2変数の関数をプロットしてみよう。まず、`rei2(x, y)` という関数を作成する。

```
function f = rei2(x,y)
f=4*x.^2 - 2*x.*y + 2*y.^2; % xやyがベクトルや行列で入ってきても計算できるように、.
(ドット)をつけて計算式を作る
end
```

そして、それを使うスクリプト `main2.m`

```
x=-10:10;
y=-10:10;
[xx,yy]=meshgrid(x,y); %グラフを書くために、メッシュをつくる。
z=rei2(xx,yy); % 各格子点ごとに関数値を求める (zも同じ点数を持つ)
surf(xx,yy,z) % 3次元グラフを描く
```



2.1 視点の設定

3次元プロットでは、視点が重要。次の2つの方法がある。

- GUI (「3次元回転」)
- `view` コマンド

`view`(方位角, 仰角)

あるいは、座標上のベクトルとして視点を与える。

```
view([25 25 5])
```

2.2 figure ウィンドウの追加

```
>> figure
```

あるいは

```
>> figure(2)
```

で、ウィンドウの追加ができる。追加するウィンドウの番号は任意の番号にできる。

2.3 図形やテキストの挿入

次の2つの方法がある。

- GUI (「挿入」)
- text コマンド

2.4 ヒストグラムのプロット

ベクトル x にデータがあるとき、簡単にヒストグラムの作成ができる。

例

```
>> x = rand(1000,1)
>> histogram(x)
```

ヒストグラムのビンの個数を2つ目の引数で指定できる。次の例は、5ずつの幅で集約。

```
>> histogram(x, 5)
```

2.5 散布図のプロット

ベクトル x と y にそれぞれ x 座標と y 座標があるとき、散布図が描ける。

例

```
>> x = randn(1000,1);
>> y = randn(1000,1);
>> scatter(x, y)
>> axis equal
```

2.6 円グラフ

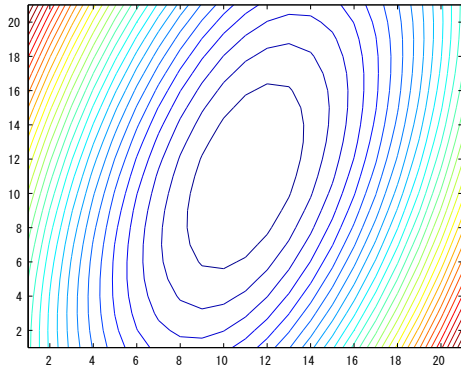
```
>> x = [1 2 3 4];
>> pie(x)
```

こんなこともできる。

```
>> x = [1 2 3 4];
>> pie(x,[0 1 0 0])
```

3次元円グラフ

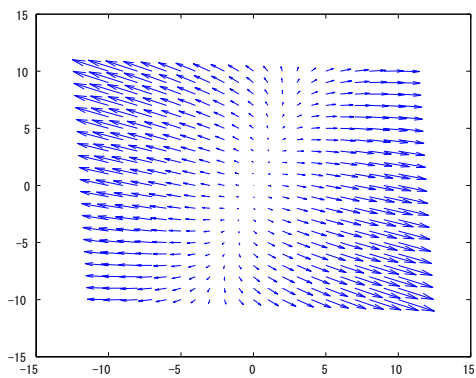
```
>> x = [1 2 3 4];
>> pie3(x)
```



2.7 等高線

2.8 Quiver

```
x=-10:10;
y=-10:10;
[xx,yy]=meshgrid(x,y);
z=rei2(xx,yy);
[px,py] = gradient(z,.2,.2);
quiver(x,y,px,py,2);
```



3 図形の描画

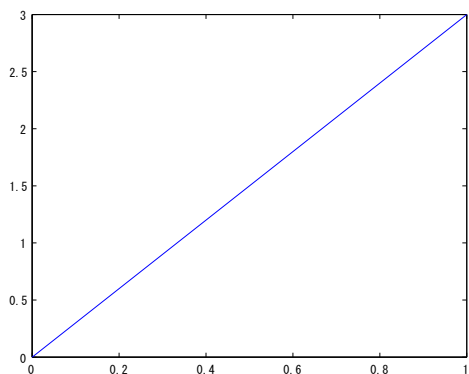
MATLAB によって、基本的な図形を描画してみよう。

3.1 直線

`plot(x,y)` によって、線分を順につないだ折れ線が描ける。x, y の要素の数が 2 つなら、線分となる。

【例】 (0,0), (1,3) を結ぶ線分を描く

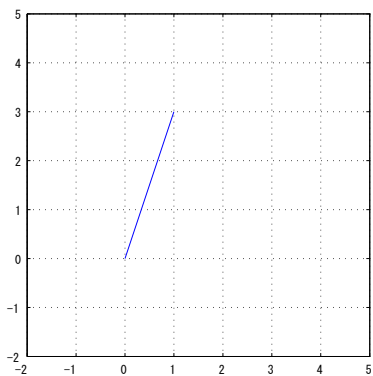
```
x=[0 1];  
y=[0 3];  
plot(x,y)
```



しかし、これではわかりにくいので、次のようにする。

- 縦と横を同じスケールで描く
- 外も少し描く
- grid を描く

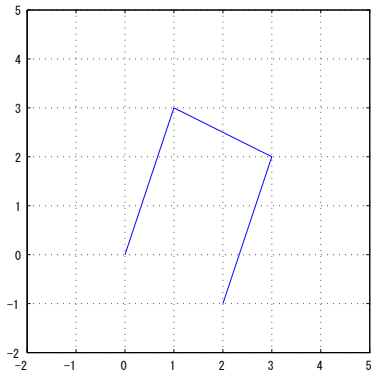
```
x=[0 1]; y=[0 3]; plot(x,y) axis equal axis([-2 5 -2 5]) grid on
```

 これでだいぶわかりやすくなった。

次に、x と y の要素を 4 つにしてみる。

```
x=[0 1 3 2];  
y=[0 3 2 -1];
```

多角形、円など、様々な図形を描くのも、この応用である。



3.2 円を描く

では、円 $x^2 + y^2 = 1$ を描くことを考えてみよう。円周上の点を、短い間隔で多数指定して x と y に入れば、`plot` で円が描けることに気づくであろう。

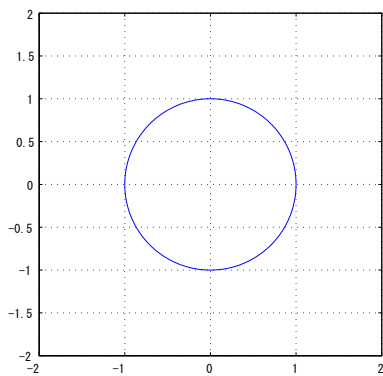
これに、円の媒介変数表示が役立つ。x 軸の方向を角度 0 度、y 軸の方向を 90 度とする、左回りの角度が用いられる。

いま、角度を 1 度刻みとしよう。0 度から 360 度で 1 周であるから、`t=0:1:360` と置こう。MATLAB の三角関数では、引数にラジアンが用いられるので、

```
t=0:360;
x=cos(t*pi/180);
y=sin(t*pi/180);
```

```
plot(x,y)
axis equal
axis([-2 2 -2 2])
grid on
```

で円が のように描ける。



楕円

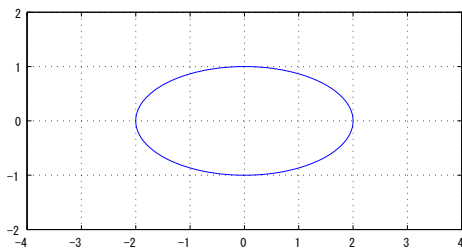
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

を描いてみよう。この場合、

```
x=a*cos(t*pi/180);
```

```
y=b*sin(t*pi/180);
```

となる（あとは描く範囲を変えた以外は同じ）。 $a = 2, b = 1$ とした場合



すでに図形が描かれている上に描くときには、注意が必要である。2番目の plot 命令が実行される際に、それ以前の図形は消えてしまう。

```
function f = rei3(x,y)
%UNTITLED2 この関数の概要をここに記述
%  詳細説明をここに記述
f=x.^2/4 + y.^2;
```

以下は、それを使うスクリプト。

```
x=-4:0.2:4;
y=-2:0.2:2;
[xx,yy]=meshgrid(x,y);
z=rei3(xx,yy);
[px,py] = gradient(z,.2,.2);
quiver(x,y,px,py,2);
```

```
hold on
```

```
t=0:360;
x=2*cos(t*pi/180);
y=sin(t*pi/180);
```

```
plot(x,y,'r')  
axis equal  
axis([-4 4 -2 2])  
grid on
```

