CHAPTER 5

微分・積分・いろんなグラフ

本章の目標は Mathematica で

- 関数を微分・積分する,
- さらに生成された導関数や原始関数のグラフを描かせる,
- といった操作を学ぶことである.

Mathematica にはグラフを描かせるための組込み関数が4章で学んだ もの以外にもいろいろと準備されている.とくに描画オプションを上手に 活用して,見やすい図を描けるようになろう.

5.1 自分で関数を定義する(即時割り当て)

Mathematica では好きな文字列に好きな関数を自分で割り当てて,三角関数 Sin や指数関数 Exp のように扱うことができる(1章1.5節,2章2.3節も参照).具体例 をみていこう.

[1] 関数 f を $f(x) = x^2 + 1$ で定義^{*1}:

$$\ln[1] = f[x_] = x^2 + 1$$

 $Out[1] = 1 + x^2$

左辺の変数にはアンダースコア(_)をつけることに注意しよう.このとき, x_0 部分は *Mathematica* のシンタックス・カラーリングにより x_0 のように表示される.

[2] たとえば f(x) に複素数 x = 1 + i を代入:

ln[2]:= f[1 + I]

^{*1} このイコール(=)を用いた定義の仕方は「即時割り当て」とよばれ,7章で扱うコロン・イコー ル(:=)を用いた「遅延割り当て」と区別される.違いは7章7.1節で解説する.

Out[2]= 1 + 2 i

[3] リスト {2, 3} を代入すると, {f[2], f[3]} に対応するリストが得られる:

 $\ln[3] = f[\{2, 3\}]$

 $Out[3] = \{5, 10\}$

[4] 2 変数関数 g(x, y) = xy + 1 を定義:

 $\ln[4]:= g[x_, y_] = x*y + 1$

Out[4] = 1 + X Y

- [5] g に $x = 3, y = \pi$ を代入:
 - ln[5]:= g[3, Pi]
 - Out[5]= $1 + 3 \pi$

[6] 定数(文字式)として使用中の文字列を関数として再利用(上書き)すると,エラーが出ることがあるので注意しよう.たとえば一度, a の値を文字式 a = 3 + 2 b と 定義:

 $\ln[6]:= a = 3 + 2 b;$

[7] そのあと文字 a を関数 x² として上書きし, a[2] を計算させてみる:

 $\ln[7]:= a[x_] = x^2; a[2]$

Set::write: (3+2 b)[x_] のタグ Plus は Protected です. ≫

Out[7] = (3 + 2 b) [2]

このようにエラーが出て、期待通りの結果 $(2^2 = 4)$ とはならない.^{*2}

[8] 関数を定義するときには念のためにクリアする習慣をつけるとよい:

ln[8]:= Clear[a];

$a[x_] = x^2; a[2]$

Out[8]= 4

[9] 一方, 関数から関数への上書きは問題ない:

^{*&}lt;sup>2</sup> このエラー・メッセージはだいたい「関数の名前にプラスの記号は使えません」といったことを伝 えている.(関数名は数値で始まったり,演算が含まれてはいけない.)[7]の入力式の a がその まま 3 + 2 b に置き換えられてしまったからである. もし [6] が a = b; であれば, ここは関数 b[x_] = x² の定義とみなされて問題なく機能する.[9]で関数から関数への上書きがうまくいく のは, すでに関数名として問題のなかった文字列がそのまま利用されるからである.

 $\ln[9] = f[x_] = x^2; f[2]$

Out[9]= 4

以下では文字 f を関数としてたびたび上書きして再利用するが,必要に応じて Clear[f] と入力し定義をクリアするようにしよう.

5.2 1 変数関数の微分

関数を微分して導関数を求め、それをまた自作の関数として定義し活用することが できる.

[10] 微分には組込み関数 D を用いる. たとえば Log[x] を変数 x で微分する:

```
In[10] = D[Log[x], x]

Out[10] = \frac{1}{x}
```

[11] D で Log[x] を変数 x について 16 階微分する:

 $In[11] = D[Log[x], \{x, 16\}]$ $Out[11] = -\frac{1307674368000}{x^{16}}$

一般に関数 f[x] を変数 x について k 階微分するには D[f[x], {x, k}] とすれば よい.

[12] 関数 f を $f(x) = x^3 - 3x$ で定義 ([1] の f を上書き):

 $\ln[12] = f[x_] = x^3 - 3*x;$

[13] 関数 f の x による導関数を求め、それを自作の関数 df として定義:

```
\ln[13] = df[x_] = D[f[x], x]
```

 $Out[13] = -3 + 3 x^2$

[14] 関数 df の x = 2 での値を求める. すなわち f'(2) の値:

ln[14] = df[2]

Out[14]= 9

[15] 関数 f の 2 階導関数を ddf と定義:

 $\ln[15]:= ddf[x_] = D[f[x], \{x, 2\}]$

Out[15]= 6 X

もちろん ddf [x_] = D[df [x], x] と定義してもよい.

[16] これらの関数をリストにして、グラフをまとめて描く:



関数のグラフはリストした順に青,赤,黄土色に色付けされる.

問題 5.1 (極値を求める) [12] で定義した関数 f [x] の極大値・極小値を求めよ.

【**解答**】 まずは導関数 df [x] が 0 となる x を Solve で求めよう:

 $\ln[] = \text{sol} = \text{Solve}[df[x] == 0, x]$

 $\mathsf{Out}[] = \{ \{ x \rightarrow -1 \}, \{ x \rightarrow 1 \} \}$

これが $f(x) = x^3 - 3x$ の極値を与える x の「候補」である. そこでの f(x) の値は

ln[]:= f[x] /. sol

 $Out[] = \{2, -2\}$

と求まるが、本当に極値かどうかを判定するにはもう少しグラフの形状(凹凸)を調べないといけない.これらの点での2階導関数の値は

$\ln[] := ddf[x] /. sol$

 $Out[] = \{-6, 6\}$

であるから、符号をみることで x = -1 のまわりでは上に凸、x = 1 のまわりでは下に凸となることがわかる.よって x = -1 で極大値 2, x = 1 で極小値 -2.

問題 5.2 (*n* 階導関数の一覧) (1) Table と TableForm を用いて, $f(x) = x^2 \sin x$ の 0 階から 5 階微分までの一覧表を作成せよ.

(2) [16] をまねて、上で得られた関数のグラフをまとめて描け.

【**解答**】 まず Table を用いて, x² Sin[x] の導関数 (derivative) からなるリスト deriv を作る (その出力はセミコロンで省略). さらに TableForm で一覧にする:

```
ln[]:= deriv = Table[D[x^2 Sin[x], {x, n}], {n, 0, 5}];
```

deriv //TableForm

Out[]//TableForm=	x ² Sin[x]
	$x^2 Cos[x] + 2 x Sin[x]$
	$4 \times Cos[x] + 2 Sin[x] - x^2 Sin[x]$
	6 Cos[x] - x ² Cos[x] - 6 x Sin[x]
	$-8 \times \cos[x] - 12 \sin[x] + x^2 \sin[x]$
	$-20 \cos[x] + x^2 \cos[x] + 10 x \sin[x]$

これらのグラフを一斉に描かせる(描画範囲は $|x| \leq 10$ とした.):



5.3 多変数関数の偏微分

[4] の g のように,2 変数以上の自作関数も定義することができる.これらを偏微 分して活用する方法を紹介しよう.

[17] 2 変数関数 $f(x,y) = x^2 + y^2$ を f で定義 ([12] をさらに上書き):

 $\ln[17] = f[x_, y_] = x^2 + y^2;$

[18]/[19] f の x 偏導関数 f_x , y 偏導関数 f_y をそれぞれ fx, fy と定義:

```
In[18]:= fx[x_, y_] = D[f[x, y], x]
Out[18]= 2 x
In[19]:= fy[x_, y_] = D[f[x, y], y]
Out[19]= 2 y
```

[20] もちろん関数として普通に値を計算させることができる:

In[20]:= fy[1, Pi]
Out[20]= 2 π



 $\ln[27]:=$ ContourPlot[f[x, y], {x, -2, 2}, {y, -2, 2}]

デフォルトの設定では紺に近いブルーが基調で,値が高いところほど明るい(白みが かった)色になる.

[28] バージョン 9 からの新機能として、オプション PlotLegends -> Automatic を 加えることで等高線グラフのとなりに色と高さの凡例が入れられるようになった:

 $\ln[28]:=$ ContourPlot[f[x, y], {x, -2, 2}, {y, -2, 2}, {y, -2, 2},

PlotLegends -> Automatic]



[29] ContourPlot で「高さ 1」での等高線 (contour) を描かせる. これは陰関数 f(x,y) = 1 のグラフを描かせることと同値である:

```
\ln[29]:= ContourPlot[f[x, y] == 1, {x, -2, 2}, {y, -2, 2}]
```



前章 4.3 節でも注意したが, == が使われていることに注意しよう.

[30] DensityPlot は ContourPlot と動作が似ているが, 描画領域を関数の値(高低) に応じて「滑らかに」色づけする密度グラフ(密度プロット)とよばれるものを描く. こちらもバージョン9から凡例をつけられるようになった*6:

 $\ln[30]:=$ DensityPlot[f[x, y], {x, -2, 2}, {y, -2, 2},



PlotLegends -> Automatic]

問題 5.5 (極値の可視化) 問題 5.4 の関数 $f(x,y) = x^3 - y^3 - 3x + 12y$ について, 以下の方法で f(x,y) の極大・極小点および鞍点がわかるようなグラフを描け. 描画 範囲は適当に選んでよい.

(1) Plot3D を用いる. ただしオプションとして、グラフの入る「箱」の縦・横・高さの長さを同じにする BoxRatios -> {1, 1, 1}, 描画範囲を |f(x,y)| ≤ 30 に制限する PlotRange -> {-30, 30} と, x 軸や y 軸にラベルをつける AxesLabel -> {x, y} をつけよ.

^{*6} バージョン 8 以前の場合,オプション PlotLegends -> Automatic とその直前のコンマは外して 入力しよう. 次の問題 5.5(2) も同様である.

(2) ContourPlot を用いる.ただし,色の凡例を添えるオプション
 PlotLegends -> Automtic と等高線の数を 30 に増やす(指定する)オプション Contours -> 30 を加えよ.

【解答】 次のように入力すれば、下のようなグラフが得られる:

In[]:= f[x_, y_] = x^3 - y^3 - 3 x + 12 y; r = 4; Plot3D[f[x, y], {x, -r, r}, {y, -r, r}, BoxRatios -> {1, 1, 1}, AxesLabel -> {x, y}, PlotRange -> {-30, 30}]

```
In[]:= ContourPlot[f[x, y], {x, -r, r}, {y, -r, r},
PlotLegends -> Automatic, Contours -> 30]
```



5.5 積分

Mathematica での積分は組込み関数 Integrate を用いる. Integrate は不定積分・定積分の両方に対応する. 数値積分が必要な場合は NIntegrate を用いる.