Chapter 10

ベクトル場と微分方程式

ベクトル場は物理学や数学において最も基本的な概念のひとつである．とくに現代の幾何学ではさまざまな形でベクトル場が登場し，重要な研究対象となっている．

微分方程式の重要性はいうまでもないだろう．いわゆるカオス理論（力学系理論）の源流も，ポアンカレによる微分方程式の研究にまでさかのぼることができる．

Mathematica にたくさんの例を計算させて，これらの対象にできるだけ具体的なイメージをもてるようになろう．

10.1 勾配ベクトル場

「地形図」とよばれるタイプの地図には海抜高度を表す等高線が描かれていて，そこから地形の凹凸に関する情報を得ることができる．

xy 平面上に関数 $z = f(x, y)$ があって，それがどこかの地域の海抜高度を表しているとしよう．もし地形図がほしければ，*Mathematica* の組込み関数 `ContourPlot` を用いて等高線グラフを描けばよい（5 章 5.4 節）．

いま，点 (p, q) にあたる地点にボールを置いたとしよう．このとき，ボールはどの方向に転がり始めるだろうか．

経験的に（もしくは物理学の諸原理によって）ボールは等高線に垂直な方向に，「高度をもっとも下げる」方向に転がる．関数 f を用いると，その方向は**勾配ベクトル** (gradient vector) [*1]

$$\operatorname{grad} f(p, q) = (f_x(p, q), f_y(p, q))$$

に「マイナスをつけた」方向として表されることが知られている．すなわち，平面の

[*1] $\nabla f(p, q)$ とも表される．

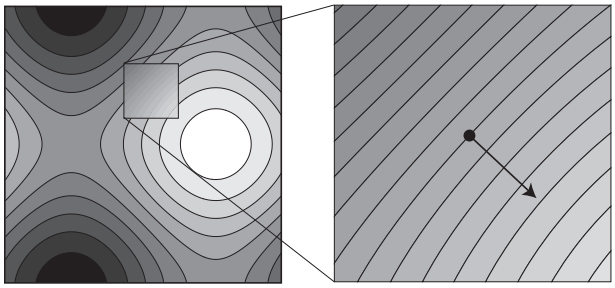


図 10.1　勾配ベクトルは「関数がもっとも増加する」方向を指し示す．したがって等高線に垂直なベクトルとなる．

各点 (p, q) にはボールが転がり始める方向を示唆する 2 次元ベクトル $\operatorname{grad} f(p, q)$ がひとつずつ対応している．[*2] これは「ベクトル場」とよばるものの典型例で，「勾配ベクトル場」ともよばれている．

さて *Mathematica* は，（勾配とは限らない，一般の）ベクトル場やそれが生成する「流れ」がとる経路を描くための組込み関数が準備されている．まずはその具体例を見ていこう．

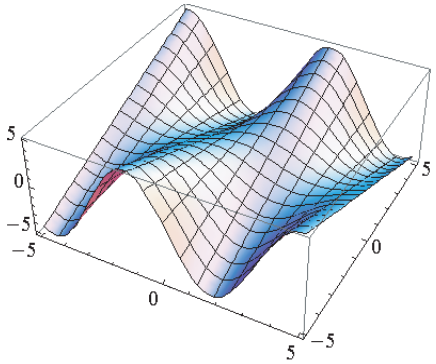
[1] 平面上の関数 `f` を即時的に定義．ここでは $f(x, y) = \cos x + y \sin x$ とした：

```
In[1]:= f[x_, y_] = Cos[x] + y*Sin[x];
```

[2] `Plot3D` で 3 次元グラフを描かせる．だいたいこんな「地形」である：

```
In[2]:= Plot3D[f[x, y], {x, -5, 5}, {y, -5, 5}]
```

Out[2]=



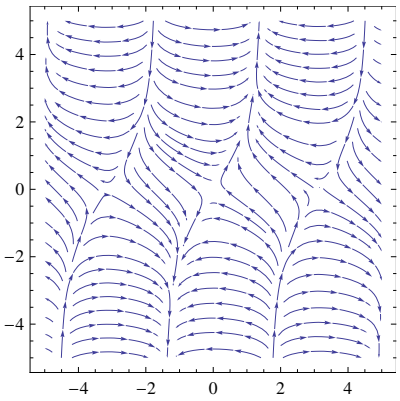
[3] `ContourPlot` で凡例付き「地形図」を描かせる．等高線の数も指定：

```
In[3]:= c = ContourPlot[f[x, y], {x, -5, 5}, {y, -5, 5},
                        Contours -> 9, PlotLegends -> Automatic]
```

[*2] あくまで「その点に置いたボール（初速度 0）が転がり始める方向」であることに注意しよう．実際に転がり始めたらボールは重力により加速されるから，一般にボールの速度ベクトルとそこでの勾配ベクトルは一致しない．勾配ベクトルはむしろ，その「加速度ベクトル」と関係している．

```
In[7]:= s = StreamPlot[gradf[x, y], {x, -5, 5}, {y, -5, 5}]
```

Out[7]=

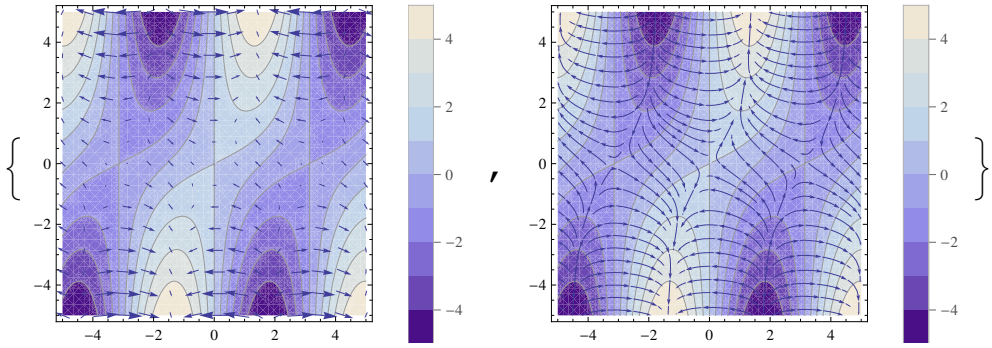


ここで描かれた曲線は**流線**とよばれる．（次節を参照.）

[8] Show を用いて v もしくは s を等高線グラフ c に重ねて表示させる：

```
In[8]:= { Show[c, v], Show[c, s] }
```

Out[8]=



このようにリストにすればグラフを並べて表示でき，比較もしやすくなる．

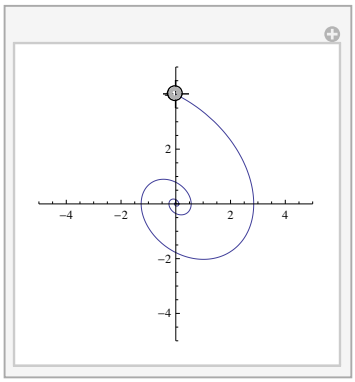
ここで注意しないといけないことは，勾配ベクトルやそれから生成される「流れ」は，関数が「増加する」方向を向いているということである．等高線グラフでいうと，青色の濃いほうから薄いほうへと進む．ボールが転がる向きとはちょうど逆方向なのである．

10.2　一般の 2 次元ベクトル場

一般に，平面上の各点に何らかのベクトルを対応付けたもの（一種の写像）を**ベクトル場** (vector field) とよぶ．たとえば点 (x, y) に 2 次元ベクトル $(u(x, y), v(x, y))$ を対応付ける写像 V はベクトル場である．具体的には，勾配ベクトル場をイメージするのが一番わかりやすいが，一般に与えられたベクトル場 V がある関数の勾配ベクトル場として表現できるとは限らないことに注意しておこう．

このベクトル場 $V : (x, y) \mapsto (u(x, y), v(x, y))$ について，時刻 $t \in \mathbb{R}$ でパラメーター付けされた曲線 $C : t \mapsto (x(t), y(t))$ が速度ベクトルに関する等式

Out[ ]=



さらに m の定義式の部分を編集することで行列が（ア）～（ウ）の場合も試すことができるだろう．（次の図 10.2 参照．） ■

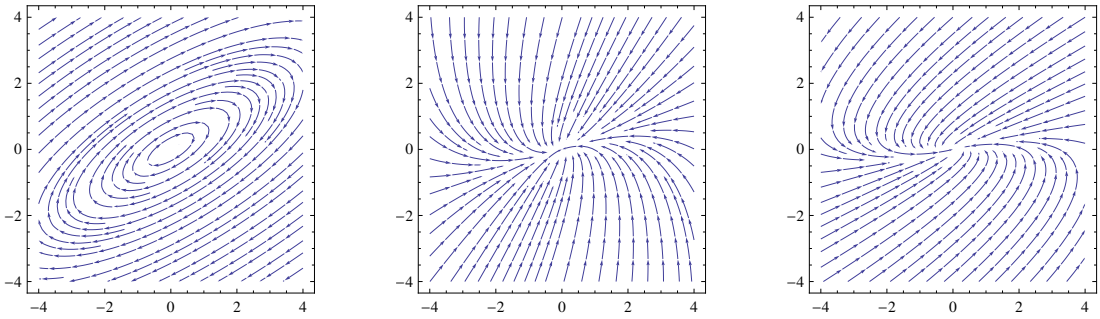


図 10.2 （ア）～（ウ）の行列に対応する線形微分方程式の解曲線．

10.5 研究：ロトカ-ヴォルテラ方程式

ある地域における時間 t でのウサギ（被食者）とキツネ（捕食者）の個体数をそれぞれ $x = x(t)$, $y = y(t)$ としたとき，これらの関数は次の**ロトカ-ヴォルテラ方程式** (Lotka-Volterra equations) でモデル化できる．

$$x' = x(\alpha - \beta y), \qquad y' = -y(\gamma - \delta x).$$

ただし，$\alpha, \beta, \gamma, \delta$ は正の定数である．たとえば餌（草）が無限にありかつ天敵のキツネがいなければ，ウサギは指数関数的に増えるのでウサギの個体数は微分方程式 $x' = \alpha x$ で表されるだろう．[*6] しかし実際には，捕食者であるキツネの個体数 y に比例してその増加係数 α は押さえられ，$\alpha - \beta y$ と修正すべきである．この値が正であればウサギは増加し，負であればウサギは減少する．これが方程式 $x' = x(\alpha - \beta y)$ の解釈である．

[*6] 一定期間 Δt に増える個体数 Δx が Δt とその時点での個体数 x に比例するから，$\Delta x = \alpha x \Delta t$.

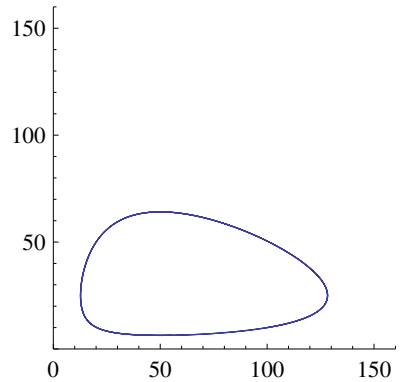
キツネのほうは $y' = \delta xy - \gamma y$ と見たほうがわかりやすい．キツネは主食であるウサギの個体数 x とキツネ自身の個体数 y に比例して増加するが（δxy の項），一方で自然死や移動にともなう自然減少も個体数に比例して存在する（$-\gamma y$ の項）．

先の 10.4 節を応用して，この方程式の解の挙動をグラフ化してみよう．

[23] まずはロトカ-ヴォルテラ方程式を定数 $\alpha = \gamma = 1$, $\beta = 0.04$, $\delta = 0.02$，初期値 $x(0) = 100$, $y(0) = 10$, 変域 $0 \leq t \leq 20$ として解き，ParametricPlot で解軌道を xy 平面に図示してみよう．ただし，汎用性を高めるために（すなわち，あとでパラメーターなどの変更がやりやすいように）「定数 α, β, γ, δ の定義」，「微分方程式の条件式の定義」，「方程式の解を求めさせる命令」はそれぞれ分割して定義してみる[*7]：

In[23]:=
```
{al, be, ga, de} = {1.0, 0.04, 1.0, 0.02};
eq := {x'[t] == x[t]*(al - be*y[t]),
       y'[t] == - y[t]*(ga - de*x[t])
        x[0] == 100, y[0] == 10};
sol = NDSolve[ eq, {x[t], y[t]}, {t, 0, 20}];
{xx[t_], yy[t_]} = {x[t], y[t]} /. sol[[1]];
ParametricPlot[{xx[t], yy[t]}, {t, 0, 20},
                                PlotRange -> {0, 160}]
```

Out[23]=



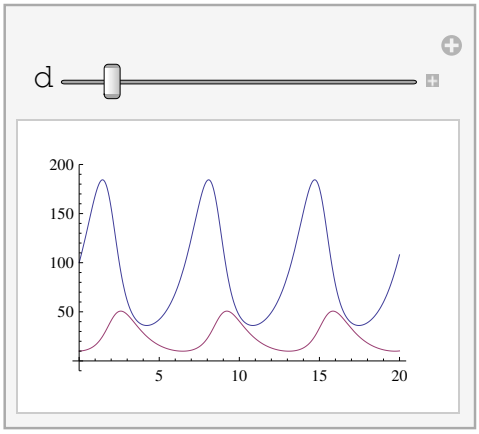
このように，閉曲線（ループ）が得られた．じつは初期値が適当な条件をみたすとき，ロトカ-ヴォルテラ方程式は周期解をもつことが知られている．すなわち，ウサギとキツネの個体数は一定の周期で増減を繰り返す．

[24] ParametricPlot を Plot に変えると，xx[t] と yy[t] の t に関するグラフをまとめて描いてくれる．

---

*7 前章でも述べたが，*Mathematica* は**条件式**（== や < で結ばれた式，11 章 11.5 節参照）そのものを文字のように扱うことができる．この eq は条件式からなるリストである．

Out[ ]=



■

10.6　研究：ローレンツ・アトラクター

次の連立微分方程式は**ローレンツ方程式** (Lorenz equation) とよばれ，いわゆる「カオス」を生成する例として，最初に発見されたもののひとつである：

$$\begin{cases} x' &= \sigma(y-x) \\ y' &= x(\rho-z)-y \\ z' &= xy-\beta z \end{cases}$$

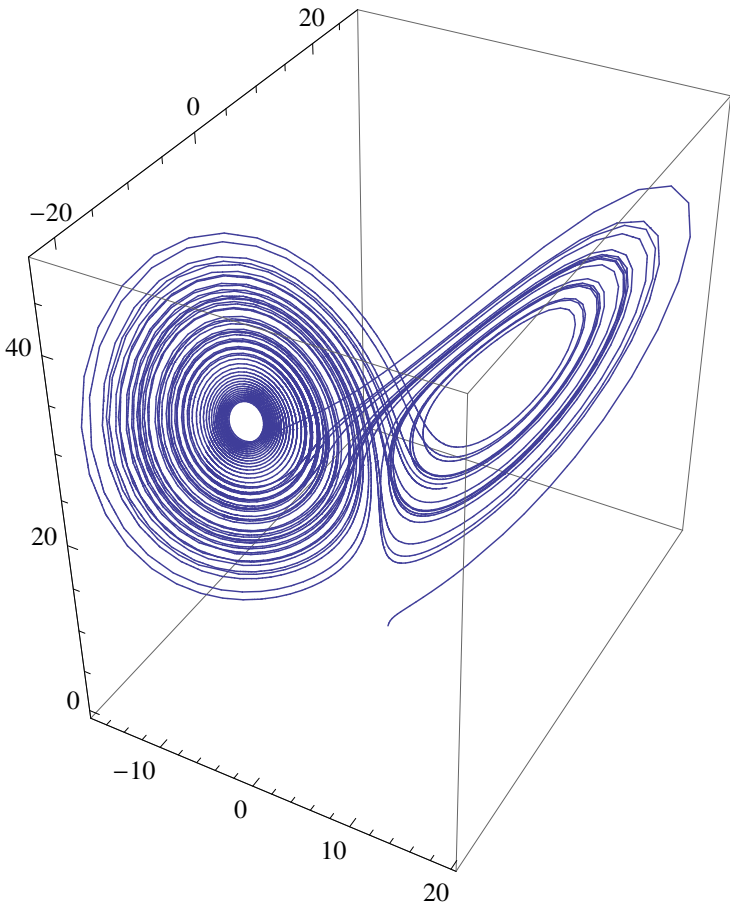
ただし，σ, ρ, β は定数とする．解曲線が集積する集合は極めて複雑な図形となることが知られており，この方程式を発見した気象学者ローレンツ（Edward N. Lorenz）にちなんで**ローレンツ・アトラクター** (Lorenz attractor) とよばれている．

[25] ここでは，このローレンツの原論文[*8] にあるパラメーター $\sigma = 10, \rho = 28, \beta = 8/3$ を選んで，解曲線（アトラクターを近似していると考えられる）を描かせてみよう：

In[25]:=
```
sol = NDSolve[ {x'[t] == 10*(y[t] - x[t]),
                y'[t] == x[t]*(28 - z[t]) - y[t],
                z'[t] == x[t]*y[t] - (8/3)*z[t],
                 x[0] == 1, y[0] == 0, z[0] == 0},
                      {x[t], y[t], z[t]}, {t, 0, 50}];
p[t_] = {x[t], y[t], z[t]} /. sol[[1]];
lorenz = ParametricPlot3D[p[t], {t, 0, 50},
                                    PlotRange -> All]
```

[*8] E.N.Lorenz. Determinastic nonperiodic flow. *Journal of the Atomospheric Sciences*, **20**(1963), 130 – 141.

Out[25]=



「アトラクター」というのは「ひきつける」を意味する動詞 attract から来ている．初期値を変化させても解曲線は最終的に同じ集合に集積するのでこの名がある．イメージとしては，アトラクターに解曲線という「糸」が「巻きついていく」ような感じである．（試しに初期値の座標を 100 や 1000 ぐらいに大きくしてみよ．）

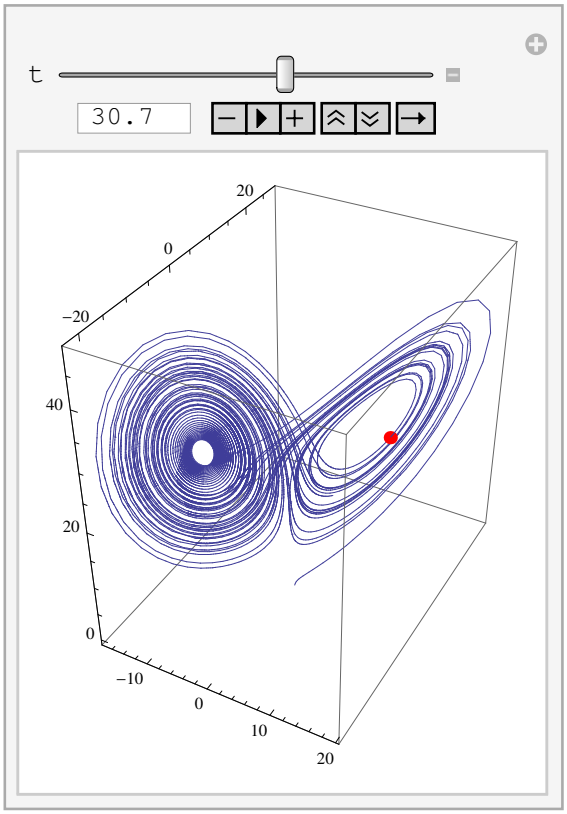
一方で，解曲線の「巻きつき方」は予想外に複雑なものとなる．規則性が見えないのである．次の問題でその様子を視覚化してみよう：

問題 10.4 (アトラクター上の軌道) [25] のグラフ `lorenz` に加えて解曲線 `p[t]` 上を動く赤い点が表示されるようなアニメーションを `Manipulate` を用いて作成せよ．

【解答】 [25] に加えて，次のように入力すればよい：

In[ ]:=
```
pt[t_] = Graphics3D[{PointSize[Large], Red, Point[p[t]]}];
Manipulate[Show[lorenz, pt[t]], {t, 0, 50}]
```

Out[ ]=



`pt[t_]` の定義では「3 次元空間に大きな赤い点を `p[t]` の位置に描く」ために組込み関数 `Graphics3D` を用いた．（詳細はドキュメントセンターを参照のこと．）■

アニメーションコントロールを開いて，再生速度を適度に遅くすればアトラクター上の極めて複雑な挙動がわかる．[*9]ローレンツ・アトラクターは蝶の羽にたとえられることが多いが，解曲線は右の羽，左の羽と不規則に移動しながら「巻きついて」いく．じつはその「不規則性」すら，安定していないのである．ためしに初期値を $(1, 0, 0)$ からわずかに変えて $(1.01, 0, 0)$ としてみよう．解曲線は同じようにローレンツ・アトラクターに「巻きついて」いるように見えるだろうが，実際の挙動はかなり変化している：

問題 10.5 (初期値鋭敏性) 初期値をわずかに変えた解曲線を `p2[t]` とおき，その上を動く緑の点を問題 10.4 で作成したアニメーションに追加せよ．

解答は省略するが，ぜひ挑戦していただきたい．赤と緑の点は最初一緒に動いているが，ある時点からまったく違った動きを始めることがわかるだろう．初期値のわずかな差が思わぬ変化を生む．これがローレンツ方程式「初期値鋭敏性」もしくは「カオス性」といわれる性質である．

[*9] こんなアニメがたった 10 行程度の命令で作れてしまうのは衝撃的ですらある．同程度のプログラムを他の言語で自作するのはかなり面倒なのである．