Chapter 7

テイラー・フーリエ・Manipulate

整数でない実数は適当な有効数字の小数で表すのが実用的である．同じことを関数でやるのがたとえばテイラー展開であり，フーリエ展開である．

この章では *Mathematica* を利用してテイラー・フーリエ展開を計算し，さらに組込み関数 `Manipulate` を使って「動的に」グラフ化する方法を学ぶ．

無理数 $\sqrt{2} = 1.4142\ldots$ は

$$\sqrt{2} = 1 + \frac{4}{10} + \frac{1}{10^2} + \frac{4}{10^3} + \frac{2}{10^4} + \cdots$$

という無限和で表現できる．たとえば $\sqrt{2} \approx 1.41$ と書くとき，それは上の無限和を 3 項目で打ち切って，残りは「誤差」として無視します，ということに他ならない．

同様に数直線 $\mathbb{R}$ 上の滑らかな関数（何度でも微分できる関数）$f : \mathbb{R} \to \mathbb{R}$, $y = f(x)$ に対し，$x = a$ での**テイラー展開** (Taylor expansion)

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_{n+1}(x)$$

が得られる．右辺の $R_{n+1}(x)$ はいわゆる**剰余項**で，関数の「無視したい」部分である．[*1]それ以外の部分が $f(x)$ の a における n 次の**テイラー多項式**とよばれるもので，$f(x)$ の「多項式による最良近似」にあたる．$x - a = 1/10$ としてみれば，上の $\sqrt{2}$ との類似が感じられるだろう．一般に $|x-a|$ がある程度小さければ n を増やすことで近似は良くなるが，[*2]実際のところテイラー多項式を求めるというのは重労働であっ

[*1] 実際 $x \to a$ と極限をとると $R_{n+1}(x)/(x-a)^n \to 0$ となる（すなわち $(x-a)^n$ より速く 0 に収束する）が，x が a から離れた場合 $R_{n+1}(x)$ は発散するのが普通である．

[*2] テイラー級数展開の収束半径より小さければよい．

て，見たこともない関数の 5 階微分を求める，なんて考えただけでも数学をやめたくなってしまう．こういうことはぜひ，*Mathematica* にやらせてしまおう．

この章では与えられた関数のテイラー展開を求め，そのグラフを描くための知識を学ぶ．さらに，組込み関数 Manipulate を用いて，テイラー多項式の次数 n が増えると近似が良くなっていく様子を実感できるような「動く」グラフ作りを学ぶ．

また関数を三角関数の和で近似するフーリエ展開についても，同様に「動く」グラフを作成してその近似の様子を観察してみよう．

7.1 関数の定義（即時割り当て vs 遅延割り当て）

変数や関数を定義する際には，5 章で学んだ**即時割り当て**（即時的定義）とこれから学ぶ**遅延割り当て**（遅延的定義）の二種類があり，うまく使い分ける必要がある．その違いを，**乱数**を発生させる組込み関数 RandomReal を用いて確認してみよう．

[1] RandomReal は 0 と 1 の間の実数をひとつランダムに選ぶ：

```
In[1]:= RandomReal[ ]

Out[1]= 0.674587
```

[ ] の中に何も書かないので違和感があるかもしれないが，気にしなくてよい．[*3]RandomeReal は何も代入しなくても，勝手に値を決めて出力する関数なのである．（詳しくは 9 章参照．）

[2] まずは**イコール** = で文字 x に RandomeReal[] を**即時割り当て**：

```
In[2]:= x = RandomReal[]

Out[2]= 0.276893
```

右辺が一旦評価された上で，その値が（定数として） x に割り当てられる．

[3] そのあと x を使うときは，毎回同じ結果となる：

```
In[3]:= {x, x, x}

Out[3]= {0.276893, 0.276893, 0.276893}
```

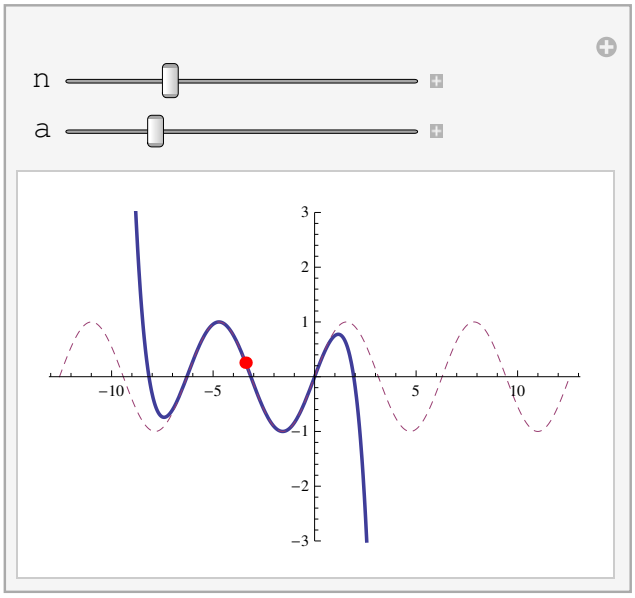
[4] 次に**コロン・イコール** := で文字 x に RandomeReal[] を**遅延割り当て**：

```
In[4]:= x := RandomReal[]
```

遅延割り当ての場合（セミコロン ; をつけなくても）何も出力されない．

[*3] ここではわざと半角の空白を入れたが，それすら必要ない．RandomeReal[] と書けばよい．

Out[21]=



`gr[a_, n_]` はアのグラフに相当するものを描くための自作関数であり，`pt[a_]` は「赤く大きな点」を `{a, f1[a]}` に描くための自作関数である．これらを同時に表示するのが `Show` の役割で，`Manipulate` で `n, a` の値を変化させる． ■

7.5　フーリエ展開

フーリエ展開もアイディアはテイラー展開によく似ている．どちらも与えられた関数を，「よくわかっている関数の和」で近似するのである．ただしテイラーが x のべき乗 x^k を用いるのに対して，フーリエは $\sin kx, \cos kx$ $(k = 0, \pm 1, \pm 2, \ldots)$ の形の三角関数を用いる．

定義（フーリエ展開）．区分的に連続な関数 $f : \mathbb{R} \to \mathbb{R}$ および $n = 0, 1, 2, \ldots$ に対し，

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nx\,dx, \quad b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nx\,dx,$$

と定義する．このとき，

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty}\{a_n \cos nx + b_n \sin nx\} \qquad (*)$$

と表し，$(*)$ の右辺の級数を $f(x)$ の**フーリエ級数** (Fourier series) もしくは**フーリエ展開** (Fourier expansion) とよぶ．

関数 f と x が適当な条件をみたせば，フーリエ級数は収束しかつ $(*)$ 式の $\sim$ を等号 $=$ に変えた式が成立する．

a_n, b_n の定義より，関数 f のフーリエ級数は区間 $[-\pi, \pi]$ での f の値だけで決まってしまう．また，フーリエ級数は（発散する場合もふくめて）周期 2π をもつことに注意しよう．

この節ではフーリエ級数が収束する様子を *Mathematica* で観察してみよう．

[22] 関数 `f2[x]` を適当に定義（$f_2(x) = x$）：

```
In[22]:= f2[x_] = x;
```

[23] `FourierTrigSeries` で関数 `f2[x]` の 5 次フーリエ展開を求める：

```
In[23]:= FourierTrigSeries[f2[x], x, 5]
```

```
Out[23]= 2 Sin[x] - Sin[2 x] +
          2          1          2
          - Sin[3 x] - - Sin[4 x] + - Sin[5 x]
          3          2          5
```

こちらはテイラー展開の `Series` のように「剰余項」にあたるものがつかないので，出力はそのまま関数として使える．

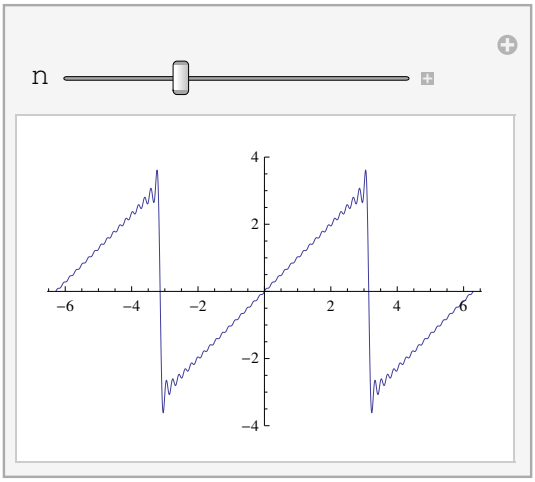
[24] `n` 次のフーリエ展開を関数 `g2` として「遅延的に」定義：

```
In[24]:= g2[x_, n_] := FourierTrigSeries[f2[x], x, n]
```

[25] `Manipulate` で近似の様子を確認：

```
In[25]:= Manipulate[Plot[Evaluate[g2[x, n]],
                         {x, -2 Pi, 2 Pi}, PlotRange -> {-4, 4}],
                  {n, 1, 100, 1}]
```

Out[25]=



もとの関数 $f_2(x) = x$ 自体は数直線 $\mathbb{R}$ 上で連続だが，得られたフーリエ展開は（不連続点をもつ）関数

$$\tilde{f}_2(x) = \begin{cases} x & (-\pi < x < \pi) \\ 0 & (x = \pm\pi) \end{cases}$$

を周期 2π で $\mathbb{R}$ 全体に拡張したものに各点収束するのである.[*8]また，不連続点のまわりに発生する「ギザギザ」は**ギブズ現象** (Gibbs phenomenon) とよばれる．`n` を増やしてもこの「ギザギザ」は消えないが，不連続点に寄っていくので級数の各点収束性には矛盾しない.

問題 7.4　$f(x) = x^2$ もしくは $f(x) = |x|$ について，フーリエ展開（級数）が収束していく様子を `Manipulate` で図示せよ．（ただし，`x` の絶対値は `Abs[x]` で与えられる.）また，極限として得られる関数はどのようなものか？

【解答】　[22]–[25] の関数を変えるだけであるから，*Mathematica* による入出力は省略する（下の図 7.2 参照）．これらの例では $f(\pi) = f(-\pi)$ が成り立つことに注意しよう．フーリエ級数

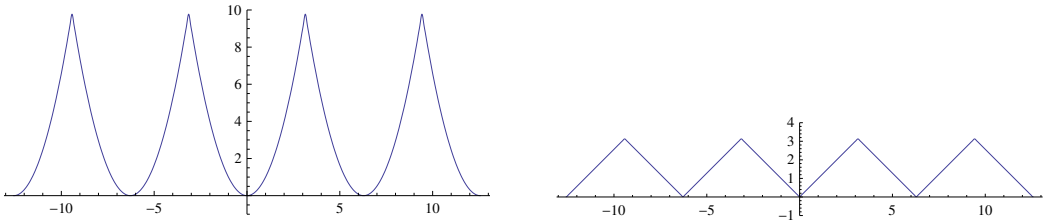


図 7.2　偶関数 x^2 および $|x|$ の 40 次のフーリエ展開.

の極限として得られる関数は $f(x)$ $(-\pi \le x \le \pi)$ を周期 2π で $\mathbb{R}$ 全体に拡張して得られる連続関数となり，収束する過程でギブズ現象はおきない.[*9] ■

問題 7.5 (矩形波 (Square wave) のフーリエ展開) 次の関数はある関数のフーリエ展開であり，$\mathbb{R}$ 全体で各点収束する．その収束の様子を `Manipulate` と `Plot` で視覚化せよ.

$$s_n(x) = \frac{4}{\pi}\left(\sin x + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \cdots + \frac{\sin(2n-1)x}{(2n-1)}\right)$$

これはどのような関数のフーリエ展開だと考えられるか？

【解答】　[24] のかわりに $s_n(x)$ にあたる関数を `Sum` を用いて（即時的に）定義し，[25] のように収束の様子を視覚化すればよい：

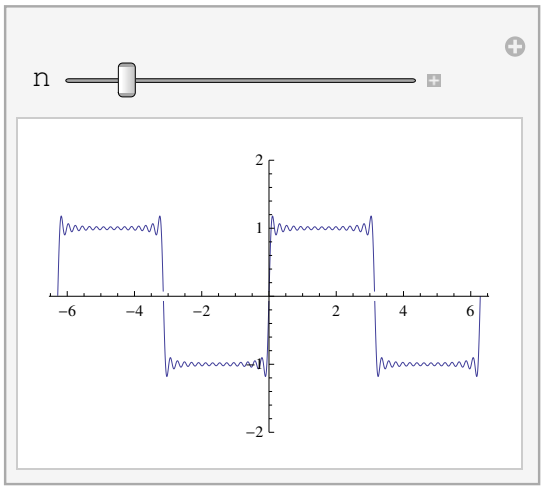
[*8] 関数の列 $\{g_n : \mathbb{R} \to \mathbb{R}\}_{n=1}^{\infty}$ が関数 $g : \mathbb{R} \to \mathbb{R}$ に**各点収束**するとは，すべての実数 x について数列 $\{g_n(x)\}_{n=1}^{\infty}$ が $g(x)$ に収束することをいう.

[*9] この場合フーリエ級数は極限関数に一様収束する．ここで周期 2π をもつ周期関数の列 $\{g_n : \mathbb{R} \to \mathbb{R}\}_{n=1}^{\infty}$ が関数 $g : \mathbb{R} \to \mathbb{R}$ に**一様収束**するとは，「誤差の最大値」$M_n = \max_{|x|\le\pi} |g_n(x) - g(x)|$ が $n \to \infty$ のとき 0 に収束することをいう.

その上で：

```
In[ ]:= s[x_, n_] = (4/Pi)*Sum[Sin[(2 k - 1) x]/(2 k - 1), {k, 1, n}];
        Manipulate[ Plot[Evaluate[s[x, n]], {x, -2 Pi, 2 Pi},
                                PlotRange -> {-3, 3}],
                    {n, 1, 100, 1}]
```

Out[ ]=



n を大きくすれば，$s_n(x)$ が $[-\pi, \pi]$ 上の不連続点をもつ関数

$$\tilde{f}(x) = \begin{cases} 1 & (0 < x < \pi) \\ -1 & (-\pi < x < 0) \\ 0 & (x = 0, \pm\pi) \end{cases}$$

を周期 2π で実数全体に拡張したものに各点収束することが観察できる． ■

7.6 研究

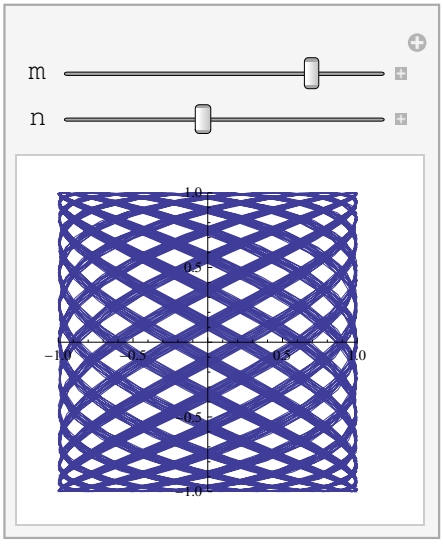
問題 7.6 (リサジュー曲線再訪) Manipulate と ParametricPlot を用いて，リサジュー曲線 $\begin{pmatrix} \cos mt \\ \sin nt \end{pmatrix}$ が n, m （整数とは限らない）に応じて変化する様子を確認せよ．

【解答】 たとえば次のように入力する[*10]：

```
In[ ]:= lissa[t_, m_, n_] = {Cos[m*t], Sin[n*t]};
        Manipulate[ParametricPlot[lissa[t, m, n],
                            {t, -2 Pi, 2 Pi}, PlotRange -> {-1, 1}],
                   {m, 1, 100}, {n, 1, 100}]
```

[*10] 6 章問題 6.3 のように m と n が整数であれば曲線 $(\cos mt, \sin nt)$ は閉曲線（ループ）を描くが，この問題の場合は閉曲線とは限らず，出力で描かれる $|t| \leq 2\pi$ での絵は曲線の一部ということになる．

Out[ ]=



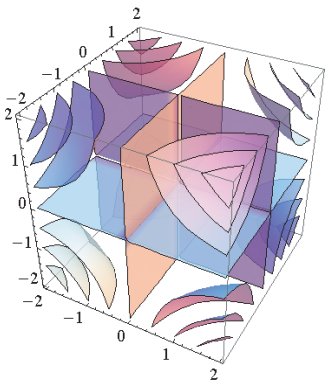
■

問題 7.7 (等位面の変化) Manipulate と ContourPlot3D を用いて，陰関数 $xyz = k$ で表される 3 次元空間内の曲面が定数 k に応じて変化する様子を確認せよ．

【解答】 ContourPlot3D は ContourPlot の 3 次元版で，関数の等高面を描く．たとえば：

In[ ]:=
```
ContourPlot3D[x y z, {x, -2, 2}, {y, -2, 2}, {z, -2, 2},
  Contours -> 7, ContourStyle -> Opacity[0.7],
  Mesh -> None]
```

Out[ ]=

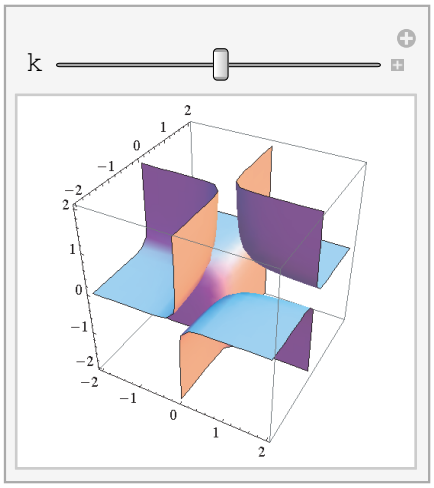


オプション Contours -> 7 で等高面の数を 7 枚に指定し，ContourStyle -> Opacity[0.7] で面の透明度を指定し，最後の Mesh -> None で面上のメッシュが描かれないようにする．

特定の等高面 $xyz = k$ を描かせたい場合，ContourPlot のときと同様に，方程式を x y z == k の形で書き入れればよい．（4 章 4.5 節も参照．）Manipulate と組み合わせると次のようになる：

In[ ]:=
```
Manipulate[ContourPlot3D[x y z == k,
  {x, -2, 2}, {y, -2, 2}, {z, -2, 2}, Mesh -> None],
 {k, -2, 2}]
```

Out[ ]=



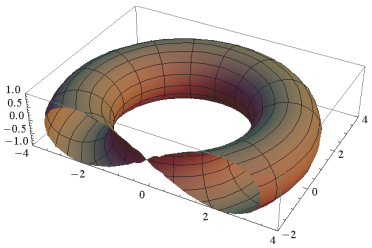
■

問題 7.8 (ドーナツを食べる) トーラスを

```
tor[u_, v_] = {(3 + Cos[v])Cos[u], (3 + Cos[v])Sin[u], Sin[v]};
```

で定義する．以下を実行し，トーラスの断面を確認せよ．

```
Manipulate[ParametricPlot3D[tor[u, v],
                    {u, 0, 2 Pi}, {v, 0, 2 Pi}, PlotStyle -> Brown,
                     PlotRange -> {{-4, 4}, {a, 4}, {-1, 1}}],
           {a, -4, 4}]
```



その他の研究課題

(1) 問題 7.3 を参考にして，問題 7.2 の「動く接線グラフ」をもっと見やすくせよ．

(2) $z = f(x, y)$ のグラフの点 $(p, q, f(p, q))$ における接平面は

$$z = f(p, q) + f_x(p, q)(x - p) + f_y(p, q)(y - q)$$

で与えられる．関数 $z = xy$ に対して，「動く接平面のグラフ」を作成せよ．

(3) *Mathematica* は高次元のテイラー展開・フーリエ展開にも対応している．ドキュメントセンターで使い方と具体例を調べよ．