CHAPTER 3

リストと数列

Mathematica の特徴のひとつは「リスト」というベクトルのようなデータ形式をもっていることである．ベクトルも行列も数列も集合も，すべてリストにできる．「リストを制すものは *Mathematica* を制す」と言っても過言ではない．

数学ではしばしば数列の収束・発散が問題となる．とくに収束数列の場合，数値計算への応用という観点からその「収束速度」が重要な意味をもつ．たとえば，次のような問題を考えよう：

収束速度の比較問題．自然対数の底 e への収束列として，

$$a_n = \left(1+\frac{1}{n}\right)^n \qquad b_n = 1+\frac{1}{1!}+\frac{1}{2!}+\cdots+\frac{1}{n!}$$

のふたつがよく知られているが，理論的には b_n のほうが収束が速いとされている．それを数値的に確かめよ．

実際，誤差 $|a_n - e|$ は $3/n$ 以下，$|b_n - e|$ は $2/n!$ 以下であることが理論的にわかるので，b_n のほうがはるかに収束が速いのである．これを数値的に実感するには，a_n と b_n を *Mathematica* に 10 項目ぐらいまで計算させてみて，結果を並べてみるのがよい（問題 3.7）．こうした操作に必要な知識を学ぶのがこの章の目標である．

3.1　リストとは？

リストとはベクトル，行列，テンソル，数列，集合などなど，さまざまなものを表現できる *Mathematica* 独自のデータ構造（データのいれもの）であり，中括弧 `{ }` を

用いて

```
{a, b, c}   {{a, b}, {c, d}}   {Sin[x], Cos[x], Tan[x]}
```

のように表される．リストはベクトルによく似ているが，ベクトルよりもっと多くの演算が大胆に適用できる．具体例をみていこう．

[1] リスト v1 と v2 を定義：

```
In[1]:= v1 = {a, b, c};
        v2 = {p, q, r};
```

[2] v1 と v2 のベクトル的な和：

```
In[2]:= v1 + v2
Out[2]= {a + p, b + q, c + r}
```

[3] ベクトル的な定数倍：

```
In[3]:= 100*v1
Out[3]= {100 a, 100 b, 100 c}
```

[4] 各要素に 1 を足す:

```
In[4]:= v1 + 1
Out[4]= {1 + a, 1 + b, 1 + c}
```

[5] 文字 x から各要素を引く:

```
In[5]:= x - v1
Out[5]= {-a + x, -b + x, -c + x}
```

[6] 各要素を 3 乗：

```
In[6]:= v1^3
Out[6]= {a^3, b^3, c^3}
```

[7] 5 を「リスト乗」：

```
In[7]:= 5^v1
Out[7]= {5^a, 5^b, 5^c}
```

[8] 指数関数を各要素に作用：

```
In[8]:= Exp[v1]
Out[8]= {e^a, e^b, e^c}
```

[9] リストの「リスト乗」：

```
In[9]:= v1^v2
Out[9]= {a^p, b^q, c^r}
```

[10] 要素ごとの積：

```
In[10]:= v1*v2
Out[10]= {a p, b q, c r}
```

[11] 要素ごとの商：

```
In[11]:= v1/v2
```

$$\mathrm{Out[11]=}\ \left\{\frac{a}{p}, \frac{b}{q}, \frac{c}{r}\right\}$$

[12] ベクトルとしての**内積**：

```
In[12]:= v1.v2
Out[12]= a p + b q + c r
```

問題 3.1 `u = {1, 2, 3, 4, 5}` と定義する．文字 `x` と上で紹介したリストの演算をうまく組み合わせて，

$$\{1^3-1,\ 2^3-2,\ 3^3-3,\ 4^3-4,\ 5^3-5\},\quad \left\{x,\ \frac{x^2}{2!},\ \frac{x^3}{3!},\ \frac{x^4}{4!},\ \frac{x^5}{5!}\right\}$$

に等しいリストをそれぞれ作成せよ．ただし，n の階乗は `n!` もしくは `Factorial[n]` で計算できる．

【解答】　ほとんどパズルのような問題である．まず

```
In[ ]:= u = {1, 2, 3, 4, 5};
```

と定義する．たとえば $\{1^3-1,\ 2^3-2,\ 3^3-3,\ 4^3-4,\ 5^3-5\}$ の場合，

```
In[ ]:= u^3 - u
Out[ ]= {0, 6, 24, 60, 120}
```

次に $\{x,\ x^2/2,\ x^3/3!,\ x^4/4!,\ x^5/5!\}$ の場合，階乗のリストが

```
In[ ]:= u!
Out[ ]= {1, 2, 6, 24, 120}
```

と得られることを確かめた上で，

```
In[ ]:= x^u/u!
```

$$\text{Out[ ]}= \left\{x,\ \frac{x^2}{2},\ \frac{x^3}{6},\ \frac{x^4}{24},\ \frac{x^5}{120}\right\}$$

■

3.2　リストの生成

リストを効率的に作る組込み関数を紹介しておこう．まず（有限項からなる）等差数列を作るには，`Range` が良い．

[13] `Range` を用いて，リスト $\{1, 2, 3, 4, 5\}$ を作る：

```
In[13]:= Range[5]
Out[13]= {1, 2, 3, 4, 5}
```

[14] `Range` を用いて，リスト $\{4, 5, \ldots, 10\}$ を作る：

```
In[14]:= Range[4, 10]
Out[14]= {4, 5, 6, 7, 8, 9, 10}
```

[15] Range を用いて，$4 \leq x \leq 10$ で 0.7 刻みのリストを作る：

```
In[15]:= Range[4, 10, 0.7]
Out[15]= {4., 4.7, 5.4, 6.1, 6.8, 7.5, 8.2, 8.9, 9.6}
```

これは「初項 4，公差 0.7 の等差数列の 10 以下の部分からなるリスト」である．

一般項を与えてリストを生成するには，関数 Table を用いる．

[16] Table を用いて，平方数 (square numbers) からなるリスト sq を作る：

```
In[16]:= sq = Table[n^2, {n, 1, 10}]
Out[16]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

Table[..] の中身は「n を 1 から 10 まで（1 刻みで）動かして n^2 のリストを作れ」という意味である．Table はたいへん使い勝手が良いので，以後も頻繁に登場することになるだろう．

[17] うしろに**二重括弧 [[ ]]** をつけてリスト sq の 7 番目の**要素を取り出す**：

```
In[17]:= sq[[7]]
Out[17]= 49
```

[18] Length でリスト sq の**長さ**（要素の数）を求める．意外とよく使う関数である：

```
In[18]:= Length[sq]
Out[18]= 10
```

問題 3.2 (e への収束列) Table を用いて，自然対数の底 e への収束列 $a_n = \left(1 + \frac{1}{n}\right)^n$ の第 10 項目まで近似値（N 関数を施した値）からなるリスト an を作れ．

【解答】 [16] と同様に Table を用いる：

```
In[ ]:= an = Table[N[(1 + 1/n)^n], {n, 1, 10}]
Out[ ]= {2., 2.25, 2.37037, 2.44141, 2.48832,
  2.52163, 2.5465, 2.56578, 2.58117, 2.59374}
```

■

問題 3.3 (まとめて因数分解) Table を用いて，$x-1, x^2-1, \cdots, x^{10}-1$ からなるリストと，その各要素を因数分解した結果からなるリストを作れ．

【解答】 [16] と同様に Table 関数を用いて $x^n - 1$ のリストを作る：

```
In[ ]:= list = Table[x^n - 1, {n, 1, 10}]
```

$$\text{Out[ ]=}\quad \{-1+x,\ -1+x^2,\ -1+x^3,\ -1+x^4,\ -1+x^5,\ -1+x^6,\ -1+x^7,\ -1+x^8,\ -1+x^9,\ -1+x^{10}\}$$

次に Factor を list の各要素に適用する[*1]：

```
In[ ]:= Factor[list]
```

$$\text{Out[ ]=}\quad \{-1+x,\ (-1+x)(1+x),\ (-1+x)(1+x+x^2),\ (-1+x)(1+x)(1+x^2),\ (-1+x)(1+x+x^2+x^3+x^4),\ (-1+x)(1+x)(1-x+x^2)(1+x+x^2),\ (-1+x)(1+x+x^2+x^3+x^4+x^5+x^6),\ (-1+x)(1+x)(1+x^2)(1+x^4),\ (-1+x)(1+x+x^2)(1+x^3+x^6),\ (-1+x)(1+x)(1-x+x^2-x^3+x^4)(1+x+x^2+x^3+x^4)\}$$ ■

問題 3.4 (要素の取り出し・並べ替え) mat = {m,a,t,h,e,m,a,t,i,c,a} と定義する．このリストの文字（要素）の順序を逆にしたリスト tam を Table 関数を用いて作れ．(HINT：tam の n 項目は mat の何項目？)

【解答】 じつは組込み関数 Reverse を用いて tam = Reverse[mat] とすれば一発なのだが，ここはあくまで練習ということで Table を用いてやってみよう．

リスト mat を定義し，その長さを len とおく[*2]：

```
In[ ]:= mat = {m,a,t,h,e,m,a,t,i,c,a};
        len = Length[mat];
```

tam の n 番目の要素は mat の len - (n - 1) 番目の要素だから，

```
In[ ]:= tam = Table[mat[[len - n + 1]], {n, 1, len}]
```

```
Out[ ]= {a, c, i, t, a, m, e, h, t, a, m}
```
■

3.3　「リストのリスト」の行列形式・表形式

問題 3.3 の結果を見ると，長いリストが複数行にわたって出力されていて，あまり見やすいとはいえない．ここではリストを一覧表の形で並べたり，行列を「リストのリスト」として表現する方法を紹介しよう．

[*1] Factor も [8] の Exp のように，リストの要素それぞれに関数の作用を「分配」することができる．

[*2] もちろん数え上げれば len が 11 だとわかるが，今後の応用を考えてあえて「*Mathematica* にも意味がわかるように」定義したのである．

[19] 「リストのリスト」 m1 を定義[*3] :

```
In[19]:= m1 = {{a, b, c}, {p, q, r}};
```

[20] MatrixForm を用いて m1 を**行列** (matrix) として表示 :

```
In[20]:= m1 //MatrixForm
```

Out[20]//MatrixForm=

$$\begin{pmatrix} a & b & c \\ p & q & r \end{pmatrix}$$

この入力式は MatrixForm[m1] と書いてもよい.[*4][*5][19] で定義したリスト m1 の中身と，行と列の対応に注意しよう.

[21] TableForm を用いて m1 を**表** (table) として表示 :

```
In[21]:= m1 //TableForm
```

```
Out[21]//TableForm=
a   b   c
p   q   r
```

これも TableForm[m1] と書いてよい.

[22] Table で行列（リストのリスト）を生成することもできる :

```
In[22]:= m2 = Table[i + j, {i, 1, 4}, {j, 1, 5}]

Out[22]= {{2, 3, 4, 5, 6}, {3, 4, 5, 6, 7},
          {4, 5, 6, 7, 8}, {5, 6, 7, 8, 9}}
```

[23] TableForm で見てみよう :

```
In[23]:= m2 //TableForm
```

```
Out[23]//TableForm=
2   3   4   5   6
3   4   5   6   7
4   5   6   7   8
5   6   7   8   9
```

m2 の定義式（[22]）と比べると，{i, 1, 4} が行の方向，{j, 1, 5} が列の方向に対応することがわかる.

[24] 二重括弧 [[ ]] で 2 行 5 列目に対応する要素を取り出す :

```
In[24]:= m2[[2, 5]]
```

---

*3 [1] で定義した v1 = {a, b, c} と v2 = {p, q, r} を用いて，m1 = {v1, v2} としてもよい.

*4 逆の発想で，Sin[x] を x//Sin と書いてもよいのである．しかし本書では混乱をさけるために TableForm と MatrixForm 以外でこの記法は用いない.

*5 MatrixForm はリストをいわば「文字を行列風に配置した画像のようなもの」に置き換える関数である．たとえば 2*m1 は行列（リスト）として各要素を 2 倍したものになるが，2*MatrixForm[m1] はもはや行列ではなく，これ以上の計算ができない．次の TableForm も同様である.

```
Out[24]=  7
```

問題 3.5 (九九の表) リストの TableForm を用いて，1 から 9 までの積の表（九九の表）を作れ．また，1 から 19 までの奇数同士の積を表にせよ．

【解答】 まずは Table を用いて，

```
In[ ]:= kuku = Table[m * n, {m, 1, 9}, {n, 1, 9}];
        kuku //TableForm
```

```
Out[ ]//TableForm=
1   2   3   4   5   6   7   8   9
2   4   6   8   10  12  14  16  18
3   6   9   12  15  18  21  24  27
4   8   12  16  20  24  28  32  36
5   10  15  20  25  30  35  40  45
6   12  18  24  30  36  42  48  54
7   14  21  28  35  42  49  56  63
8   16  24  32  40  48  56  64  72
9   18  27  36  45  54  63  72  81
```

を得る．同様に，

```
In[ ]:= kisuu = Table[(2 m - 1)*(2 n - 1), {m, 1, 10}, {n, 1, 10}];
        kisuu //TableForm
```

```
Out[ ]//TableForm=
1   3   5   7   9   11  13  15  17  19
3   9   15  21  27  33  39  45  51  57
5   15  25  35  45  55  65  75  85  95
7   21  35  49  63  77  91  105 119 133
9   27  45  63  81  99  117 135 153 171
11  33  55  77  99  121 143 165 187 209
13  39  65  91  117 143 169 195 221 247
15  45  75  105 135 165 195 225 255 285
17  51  85  119 153 187 221 255 289 323
19  57  95  133 171 209 247 285 323 361
```

■

問題 3.6 (一覧表の作成) 1 から 10 までの自然数とその 2 乗，3 乗の一覧表を作成せよ．

【解答】 「{n, n^2, n^3} の形のリスト」を並べたリスト power を作成し，TableForm で表現する：

```
In[ ]:= power = Table[{n, n^2, n^3}, {n, 1, 10}];
        power //TableForm
```

Out[ ]//TableForm=

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

■

3.4 数列の和

数列の和を *Mathematica* に計算させたいときは，組込み関数 Sum を用いるとよい．

[25] Sum で 1 から 10 までの 2 乗の和 $\sum_{n=1}^{10} n^2$ を求める：

```
In[25]:= Sum[n^2, {n, 1, 10}]
```

Out[25]= 385

入力式はちょうど，[16] で定義したリスト sq = Table[n^2, {n, 1, 10}] の Table を Sum に変えた形になっている．

[26] *Mathematica* は 2 乗の和の一般項も知っている：

```
In[26]:= Sum[k^2, {k, 1, n}]
```

Out[26]= $\frac{1}{6}$ n (1 + n) (1 + 2 n)

[27] オイラーの無限級数 $1 + 1/2^2 + 1/3^2 + \cdots$ の値も知っている：

```
In[27]:= Sum[1/n^2, {n, 1, Infinity}]
```

Out[27]= $\frac{\pi^2}{6}$

入力式の Infinity は無限大 ∞ を表す組込み定数である．

[28] 8 次の多項式を作る[*6]：

```
In[28]:= Sum[x^n/n!, {n, 0, 8}]
```

Out[28]= $1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40\,320}$

[*6] ちなみに無限級数 Sum[x^n/n!, {n, 0, Infinity}] を計算するとちゃんと指数関数 Exp[x] になる．たいしたものである．

[29] Product で 1 から 19 までの奇数の積 $\prod_{n=1}^{10}(2n-1)$ を求める：

```
In[29]:= Product[2*n - 1, {n, 1, 10}]
Out[29]= 654 729 075
```

[30] 問題 3.5 で求めた「九九の表」にある数を全部足し上げる：

```
In[30]:= Sum[m * n, {m, 1, 9}, {n, 1, 9}]
Out[30]= 2025
```

では，冒頭の**収束速度の比較問題**に解答を与えよう：

問題 3.7 (収束速度の比較) 自然対数の底 e への典型的な収束列

$$a_n = \left(1+\frac{1}{n}\right)^n \qquad b_n = 1+\frac{1}{1!}+\frac{1}{2!}+\cdots+\frac{1}{n!}$$

を考える．これらの収束速度を比較しよう．

(1) Sum を用いて b_5 の近似値を求めよ．
(2) さらに Table を組み合わせて，b_n の第 10 項目までの近似値からなるリスト bn を作成せよ．
(3) 上の bn と問題 3.2 で作成した an を用いて，a_n, $|a_n - e|$, b_n, $|b_n - e|$ を比較する一覧表を作成せよ．ただし，実数もしくは複素数 x の絶対値は組込み関数 Abs[x] で与えられる．

【解答】 (1) 近似値であるから N を忘れずに：

```
In[ ]:= N[Sum[1/k!, {k, 0, 5}]]
Out[ ]= 2.71667
```

(2) Table の中に N，Sum を入れ子状に用いる：

```
In[ ]:= bn = Table[ N[Sum[1/k!, {k, 0, n}]], {n, 1, 10}]
Out[ ]= {2., 2.5, 2.66667, 2.70833, 2.71667,
         2.71806, 2.71825, 2.71828, 2.71828, 2.71828}
```

(3) 少し工夫が必要である．たとえばリスト an を使うと，a_5 と $|a_5 - e|$ はそれぞれ an[[5]] と Abs[an[[5]]-E] で表されることに注意しよう：

```
In[ ]:= hikaku = Table[
```

```
        {an[[n]], Abs[an[[n]]-E], bn[[n]], Abs[bn[[n]]-E]},
        {n, 1, 10}];
hikaku //TableForm
```

Out[ ]//TableForm=

2.	0.718282	2.	0.718282
2.25	0.468282	2.5	0.218282
2.37037	0.347911	2.66667	0.0516152
2.44141	0.276876	2.70833	0.0099485
2.48832	0.229962	2.71667	0.00161516
2.52163	0.196655	2.71806	0.000226273
2.5465	0.171782	2.71825	0.0000278602
2.56578	0.152497	2.71828	3.05862×10^{-6}
2.58117	0.137107	2.71828	3.02886×10^{-7}
2.59374	0.124539	2.71828	2.73127×10^{-8}

見てのとおり，b_n のほうが圧倒的な速さで収束することがわかる．e の近似値が必要になったとき，実用上は b_n を使うほうが効率がよい，ということである． ■

最後に，数列の和に関連する大学入試問題を Sum を使って解いてみよう：

問題 3.8 (大学入試問題から) *Mathematica* を用いて計算せよ．

(1) 次の和を数値的に（近似値で）求めよ：$\displaystyle\sum_{k=2}^{100} \frac{3}{\sqrt{k}+\sqrt{k-1}}$. **(99 兵庫大)**

(2) 数列 $\dfrac{2}{2^2-1}, \dfrac{2}{3^2-1}, \dfrac{2}{4^2-1}, \ldots$ の第 n 項までの和を求めよ．

(00 日本福祉大)

(3) 連続する m 個の奇数 $1, 3, 5, \ldots, 2m-1$ の中から，異なる 2 つの数をとって積を作る．こうして得られる ${}_mC_2$ 通りの積すべての和を求めよ．

(99 浜松医大)

【解答】 (1) 普通われわれは $\dfrac{1}{\sqrt{k}+\sqrt{k-1}} = \sqrt{k}-\sqrt{k-1}$ のような変形を期待するが，与えられた式のとおり *Mathematica* に Sum[3/(Sqrt[k] + Sqrt[k - 1]), {k, 2, 100}] と入力しても，ただ長い式が書き並べられるだけである．しかし問題では「数値的に」といっているから，上の入力式にさらに N 関数を施すか，もしくははじめから各項に N 関数を施して次のように計算させる[*7]：

```
In[ ]:= Sum[N[3/(Sqrt[k] + Sqrt[k - 1])], {k, 2, 100}]
Out[ ]= 27.
```

(2) 少し時間はかかるがそのまま入力すれば一般項を求めてくれる（一般項と i の範囲に注

[*7] Sum[3.0/(Sqrt[k] + Sqrt[k - 1]), {k, 2, 100}]; としても同じ結果が得られる．分子の 3 を 3.0 に変えることで，数値計算だと認識されるからである．

意.)：

```
In[ ]:=  Sum[2/(i^2 - 1), {i, 2, n + 1}]
Out[ ]=  (5 n + 3 n^2)/(2 (1 + n) (2 + n))
```

(3) これは若干難しいかもしれないが，次の恒等式

$$(x_1 + \cdots + x_m)^2 = (x_1^2 + \cdots + x_m^2) + 2 \underline{\sum_{1 \le i < j \le m} x_i\, x_j}$$

において $x_i = 2i - 1$ $(1 \le i \le m)$ とおけばよい（求めたいのは下線部）．あとは *Mathematica* にやらせよう：

```
In[ ]:=  a = Sum[2 i - 1, {i, 1, m}];
         b = Sum[(2 i - 1)^2, {i, 1, m}];
         c = (a^2 - b)/2
Out[ ]=  1/2 (m^4 + 1/3 (m - 4 m^3))
In[ ]:=  Factor[c]
Out[ ]=  1/6 (-1 + m) m (-1 - m + 3 m^2)
```

■

3.5　研究

以下を *Mathematica* を用いて実行してみよ．

(1) i が 1 から m まで，j が 1 から n までの値をすべて動くとき，$i + j$ の総和を求めよ．すなわち，$\sum_{1 \le i \le m, 1 \le j \le n} (i + j)$ を計算せよ．

(2) i, j, k が 1 から n までの値をすべて動くとき，$i + j + k$ の総和を求めよ．すなわち，$\sum_{1 \le i,j,k \le n} (i + j + k)$ を計算せよ．

(3) 初項 4，公比 8 の等比数列の第 n 項までの和を S_n で表す．このとき，S_{99} は カキ けたの整数，S_{100} は クケ けたの整数である．さらに，S_1 から S_{100} の間で，クケ 以下のすべてのけた数の S_n が存在するか確認せよ．

（92 センター本試・改）

(HINT: S_n の桁数は $1 + \log_{10} S_n$ の整数部分で与えられる．その一覧表を作れ．正の実数 x の整数部分は `Floor[x]` で与えられる．)

(4) ドキュメントセンターを開き，リストに関連してどのような組込み関数があるか調べよ．