

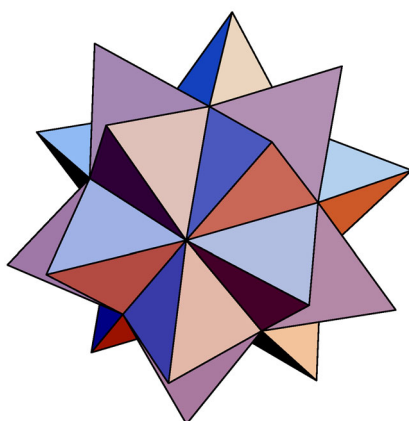
地域連携事業

夢大学 in TOYAMA '04

体験入学

「数学ワンダーランド 2004」

～ Mathematica を使って図形を描こう ～



講師： 吉田 範夫 (よしだ のりお)
幸山 直人 (こうやま なおと)

インターネット公開版

富山大学 理学部 数学教室

商標

- Mathematica は Wolfram Reserch 社の登録商標です。
- Windows は Microsoft 社の登録商標です。

本テキストの著作権について

本テキストの一部または全部を著作権法の定める範囲を超え、許可なく、複写・転載・複製・テープ化・ファイルに落とすことを禁じます。ただし、本テキストの表紙に「インターネット公開版」¹と記載してあるものについては、個人的な利用に限り、一部または全部の内容を改変することなく、再配布・複写・複製・ファイルに落とすことを許可します。

¹本テキストの「インターネット公開版」は、ホームページ等で公開すると著作権法に違反する可能性のある情報を取り除いたものです。

はじめに

近年は、コンピュータの進歩と共に「IT²」という言葉に後押しされ、ありとあらゆるところにコンピュータとインターネットが普及するようになりました。教育現場も例外ではなく、高等学校では「情報」という科目が必修となり、文書作成やプログラミングだけでなく情報収集・プレゼンテーション・モラルとマナーなど様々な情報(コンピュータとインターネット)に関する知識を学ぶことが義務付けられるようになりました。富山大学理学部数学科でもコンピュータを利用した数学の教育や学習が盛んに行われており、特に、数学に適した数式処理ソフト「Mathematica」が活用されています³。

本テキストでは、数式処理ソフト「Mathematica」のグラフィック機能やアニメーション機能を利用して様々な関数のグラフィックやアニメーションを描きます。第1章は、「Mathematica」を使用する上で最低限必要だと思われる使い方・仕組み・記号の意味などの内容について記述してあります。第2章では、「Mathematica」のグラフィック機能とそのコマンドについての解説し、様々な関数のグラフィックの描きます。また、「Mathematica」のアニメーション機能を利用し、作成したグラフィックをアニメーションとして眺めます。なお、「Mathematica」で作成したグラフィックやアニメーションをホームページ素材として利用できるようにする方法を付録として掲載しました。

本テキストは、中・高校生を対象に作成したのですが、それ以外の方にも、本テキストで作成したグラフィックやアニメーションを通して、視覚的な刺激を受け、数学に興味を持っていたければ幸いです。また、本テキストを足がかりに更に進んだ内容にも挑戦していただきたいと思います。

2004年9月10日

幸山 直人

²インフォメーションテクノロジー

³富山大学は「Mathematica」のサイトライセンスを購入しており、学生が自由に利用できる環境を整えています。

「Mathematica」について：

「Mathematica」は Wolfram Research 社が開発した数式処理ソフトウェアです。数学の計算は数値的・記号的・グラフィック的の3つに大きく別けることができますが、「Mathematica」は、これら全てを統一的な方法で取り扱うことができます。主な特徴としては、

- 記号を含む数値計算
- 関数やデータの可視化システム (グラフィックとアニメーション)
- モデリングやデータ解析
- 数学を含む多くの科学分野での知識を表現するシステム
- テキスト・グラフィック・アニメーション・サウンドを活かした対話的ドキュメントの作成
- 外部プログラムや内部 (組み込み) プログラムのコントロール

などがあります。従って、本テキストで利用する「Mathematica」の機能は、ほんの一部でしかありません。

キーボード入力について：

本テキストで用いる、キーボード入力に関する表現方法について説明します。一般的な文字1つを入力する場合は「a」のように記述し、文字列を順次入力するような場合は「abcdef」と記述しました。文字以外の特殊キーの入力については角カッコを使って [特殊キー名] という形で表記しました。

読み方	キーボード印刷文字	本テキストの表記方法
オルトキー	Alt	[Alt]
バックスペースキー	Backspace	[Backspace]
コントロールキー	Ctrl	[Ctrl]
デリートキー	Delete	[Delete]
改行キー	Enter	[Enter]
エスケイプキー	Esc	[Esc]
インサートキー	Insert	[Insert]
シフトキー	Shift	[Shift]
スペースキー	Space	[Space]
タブキー	Tab	[Tab]
カーソルキー	←, →, ↑, ↓	[←], [→], [↑], [↓]

また、ショートカットキーなど複数のキーを組み合わせるような場合は、個々のキーを文字「+」で連結した形で表記しました。例えば、「[Shift]+[Ctrl]+g」と表記してある場合、[Shift] を押しながら [Ctrl] を押しながら g を押してください。

目次

第 1 章 基礎編	1
1.1 Mathematica の使い方と仕組み	1
1.2 Mathematica でいろいろな計算を試みよう (中高校生向け)	11
1.3 Mathematica を使用する上で覚えておくこと	13
第 2 章 グラフィック編	17
2.1 グラフィック関数の紹介	17
2.2 関数のグラフィック	22
2.3 媒介変数表示の関数のグラフィック	25
2.4 多面体	28
付録 A 入力パレットの使い方	33
付録 B アニメーションの作り方	37

* 本テキストには CD-ROM が付属しています。同じ内容を筆者ホームページで公開しています。
<http://kouyama.math.toyama-u.ac.jp/main/etc/2004/yume/cdrom/index.htm>

第1章 基礎編

1.1 Mathematicaの使い方と仕組み

数式処理ソフト Mathematica を使ってみましょう。起動するには、Mathematica ノートブックのアイコン (図 1.1) をダブルクリックします。

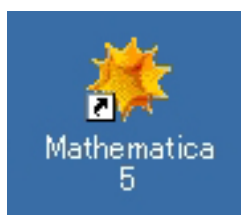


図 1.1: ノートブックのアイコン



図 1.2: カーネルのアイコン

起動すると、「ノートブック」・「メニューバー」・「(基礎的な) 入力パレット」などのウィンドウがデスクトップに表示されます (図 1.3)。

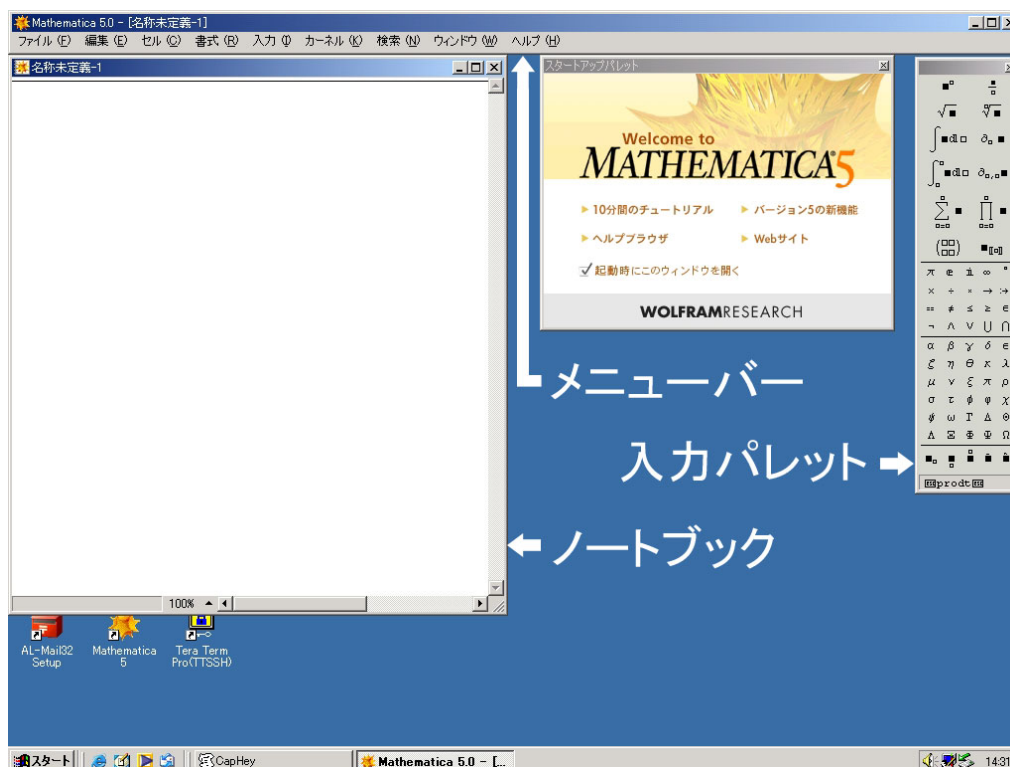


図 1.3: Mathematica 起動後のデスクトップ

まずは、シンプルな数式 $1 + 1$ を Mathematica で計算してみましょう。ノートブックに「 $1+1$ 」と入力します。同時に、セル¹の範囲を示すセルブラケット(`]`)が表示されます。

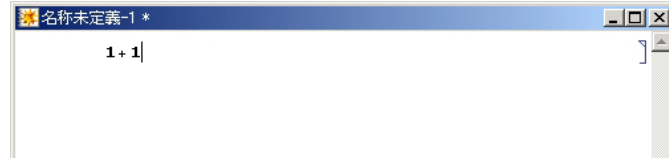


図 1.4: Mathematica ノートブック入力画面

入力が終わったら、このセルを選択し、**[Shift] + [Enter]**²を押します。Mathematica は、入力「 $1+1$ 」の前に「`In[1]:=`」を付加したのち、数式の評価³を始めます(図 1.5)。



図 1.5: Mathematica ノートブック計算中画面

評価が終わると、結果として「`Out[1]= 2`」が出力されます(図 1.6)。



図 1.6: Mathematica ノートブック出力画面

簡単な例を挙げたところで、Mathematica の仕組みについてもう少し詳しく解説しておきます。Mathematica は、構文解析⁴や数式の評価を実際に行うカーネルと、ノートブック・メニューバー・入力パレットなどコンピュータ(カーネル)とのやり取りを柔軟に行い、人間に扱いやすい GUI⁵環境を提供するフロントエンドで構成されています。特に、ノートブックの特徴である対話型インターフェースによって、図 1.7 のように、人間が **[Shift] + [Enter]** を押すことで、あらかじめノートブックに入力された数式の評価をカーネルに依頼し、カーネルは数式を評価した後、その結果をノートブックに返すという手順を繰り返し行います。ノートブックとカーネルのやり取りにおいて、ノートブックからカーネルに読み込まれた各入力行には、ラベル「`In[n]:=`」が付加され、

¹ノートブックは、入力・出力・グラフィック・テキストなどを混在した形で表示することができ、ノートブックが個々に扱うこれらオブジェクトの最小単位をセルと呼びます。また、図 1.6 のように、複数のセルが自動的にグループ化されたり、複数のセルをまとめてグループ化することもできます。

²[Shift] キーを押しながら [Enter] キーを押します。

³「数式を計算する」や「プログラムを実行する」と同じ意味合いを持ちます。

⁴数式を評価する前に、数式が正しく記述されているかチェックすること。

⁵GUI(グラフィカルユーザーインターフェース)。 対 CUI

カーネルが各入力行を評価した結果には、ラベル「 $Out[n]=$ 」が付加され、ノートブックに出力されます (n は自然数で、入力行が読み込まれる度に 1 足される)。これら「 $In[n]:=$ 」や「 $Out[n]=$ 」はカーネルによって自動的に付加されるもので入力する必要はありません。「 $In[n]:=$ 」が付加された入力行や「 $Out[n]=$ 」が付加された結果はカーネルに記憶され、再利用することができます。詳しくは 1.3 節に譲ります。

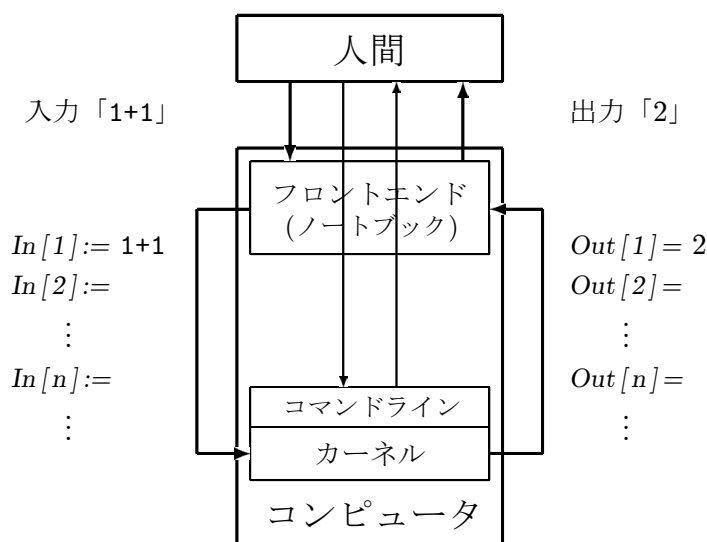


図 1.7: Mathematica の仕組み

* カーネルのアイコン (図 1.2) をダブルクリックすると CUI⁶付きのカーネルが起動し、図 1.8 のように直接カーネルに数式を評価させることができます。

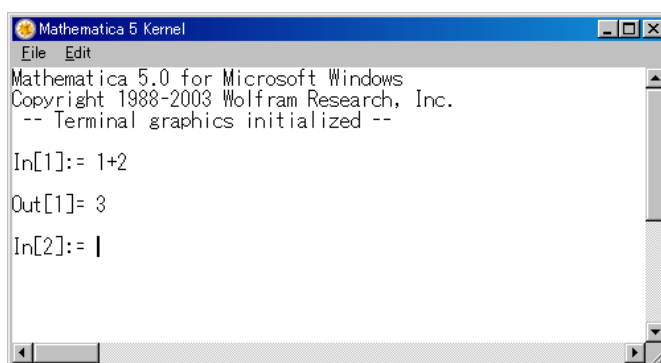


図 1.8: Mathematica カーネルでの実行例

Mathematica の仕組みをざっと説明したところで、四則演算や乗ベキなどの基本計算・ π や e などよく使う数学定数・ \sin や \log など基本的な数学関数の使い方について紹介し、それらが Mathematica でどのように扱われるか解説します。

⁶CUI(キャラクターユーザーインターフェース)。 GUI


```
In[7]:= 2 x
Out[7]= 2 x

In[8]:= 2 * x
Out[8]= 2 x

In[9]:= 2 x x
Out[9]= 2 x
```

Mathematica は「 $2x$ 」と入力しても「 $2[\text{Space}]x$ 」と理解しますが、「 x^2 」¹⁰や「 xy 」などのように「 $[\text{Space}]$ 」を入力し忘れると、「 x^2 」や「 xy 」という**変数名**として理解されてしまいます。従って、掛け算であることを明記したい場合には「 $*$ 」を使うと良いでしょう。

```
In[10]:= x2
Out[10]= x2

In[11]:= xy
Out[11]= xy
```

割り算には、演算子「 $/$ 」(**スラッシュ**)を使います。入力パレットから、「 \div 」ボタンを使用することもできますし、「 $\frac{\quad}{\quad}$ 」ボタンを使って分数の形で入力することもできます。なお、例のように数式の評価が終わると、約分され最もシンプルな形で結果が出力されます。

```
In[12]:= 7 / 21
Out[12]=  $\frac{1}{3}$ 

In[13]:=  $\frac{7}{21}$ 
Out[13]=  $\frac{1}{3}$ 

In[14]:= 7 ÷ 21
Out[14]=  $\frac{1}{3}$ 
```

Mathematica では小数点を含む数値は近似値として扱われますが、割り算を利用することで、小数点を含む小さな数についても厳密な値を持つ数値として扱うことができます。

```
In[15]:= 0.7 / 21
Out[15]= 0.0333333
```

例えば、上記のような小数点を含む小さな数に対して厳密な値として扱うには下記のように入力します。

¹⁰私たちが紙などに数式を記述するとき、 $x \times 2$ の意味で x^2 とは書かない。 $2x$ と書くほうが自然です。

```
In[16]:= (7 / 10) / 21
```

```
Out[16]=  $\frac{1}{30}$ 
```

```
In[17]:= 7 / 210
```

```
Out[17]=  $\frac{1}{30}$ 
```

乗べきには、演算子「**^**」(山または**アクセントシルコンフлекс**)を使います。入力パレットを使って「 2^{100} 」といった入力も可能です。

```
In[18]:= 2^100
```

```
Out[18]= 1267650600228229401496703205376
```

```
In[19]:= 2100
```

```
Out[19]= 1267650600228229401496703205376
```

x^{-1} は x の逆数を与えます。

```
In[20]:= 2-1
```

```
Out[20]=  $\frac{1}{2}$ 
```

数学の演算子に演算順序の優先順位があるように、Mathematica の演算子にも演算順序の優先順位が決められています。

優先順位： 低 \implies 中 \implies 高
 演算子： +, - \rightarrow *, / \rightarrow ^ \rightarrow (,)

下記の例は「 $x^2 \times y$ 」を意味し、「 x^{2y} 」を意味しているわけではありません。「 x^{2y} 」の意味を持つ数式を入力するには「 $x^{(2*y)}$ 」と入力する必要があります。

```
In[21]:= x^2 y
```

```
Out[21]=  $x^2 y$ 
```

■ よく使う数学定数

数学定数としては、円周率 π や自然対数の底 e などがよく知られています。Mathematica にもこれら数学定数が**組み込み定数**として準備され、利用することができます。なお、Mathematica の**組み込み定数の定数名は必ず大文字から始まります**。円周率 π を使用するには「**Pi**」と入力するか、入力パレットの「 **π** 」ボタンをクリックします。

```
In[22]:= Pi
```

```
Out[22]=  $\pi$ 
```

```
In[23]:=  $\pi$ 
```

```
Out[23]=  $\pi$ 
```

Mathematica は常に厳密に数値を扱おうとするため、 π の近似値を知るには、近似値に変換するための関数 (**N[数値, 桁数]**) を使って必要な精度の近似値を求める必要があります。

```
In[24]:= N[ $\pi$ , 60]
Out[24]= 3.14159265358979323846264338327950288419716939937510582097494
```

π の場合と同様に、自然対数の底 e を使用するには「**E**」と入力するか、入力パレットの「**e**」ボタンをクリックします。

```
In[25]:= E
Out[25]= e

In[26]:= e
Out[26]= e
```

π の場合と同様に、 e の近似値を求めることができます。

```
In[27]:= N[e, 60]
Out[27]= 2.71828182845904523536028747135266249775724709369995957496697
```

Mathematica は実数だけでなく複素数を扱うことができます。虚数単位 i を使用するには「**I**」と入力するか、入力パレットの「**i**」ボタンをクリックします。

```
In[28]:= I
Out[28]= i

In[29]:= i
Out[29]= i
```

虚数単位の定義より、 $i^2 = -1$ を満たします。

```
In[30]:= i2
Out[30]= -1
```

無限大を表す文字 ∞ なども組み込み定数として扱うことができます。無限大 ∞ を使用するには「**Infinity**」と入力するか、入力パレットの「 ∞ 」ボタンをクリックします。

```
In[31]:= Infinity
Out[31]=  $\infty$ 

In[32]:=  $\infty$ 
Out[32]=  $\infty$ 
```

Mathematica は角度をラジアンで扱います。組み込み定数「Degree」には変換係数 $\frac{\pi}{180}$ が定義されているため、「Degree」を掛けるだけで度からラジアンに変換されます。したがって、角度を度で表す場合は、下記のように「Degree」を掛けてラジアンに変換するか、入力パレットの「°」ボタンをクリックすることでラジアンに変換する必要があります。

```
In[33]:= 30 Degree
Out[33]= 30 °

In[34]:= 30 °
Out[34]= 30 °
```

■ 基本的な数学関数

組み込み定数と同様に、Mathematica の組み込み関数の関数名は必ず大文字から始まります。また、関数の変数は複数指定することができ、関数名に続けて角カッコ「[」と「]」でくくります。一般には、「関数名[変数 1, 変数 2, …, 変数 n]」という形をしています。馴染み深い $\sin(x)$, $\cos(x)$, $\tan(x)$ などの三角関数は「Sin[x]」、「Cos[x]」、「Tan[x]」のように入力することができます。ただし、変数 x はラジアンで与えなければなりません。

```
In[35]:= Sin[ $\frac{\pi}{6}$ ]
Out[35]=  $\frac{1}{2}$ 

In[36]:= Cos[30 °]
Out[36]=  $\frac{\sqrt{3}}{2}$ 

In[37]:= Tan[ $\frac{\pi}{12}$ ]
Out[37]=  $2 - \sqrt{3}$ 
```

自然対数関数 $\log_e(x)$ (または e を省略して $\log(x)$) は「Log[x]」のように入力することで使用することができます。

```
In[38]:= Log[x]
Out[38]= Log[x]
```

常用対数関数 $\log_{10}(x)$ や底 2 の対数関数 $\log_2(x)$ も「Log[10,x]」や「Log[2,x]」のように入力することで使用することができます。Mathematica は b を底とする一般的な対数関数 $\log_b(x)$ を対数関数の性質にしたがって、

$$\log_b(x) = \frac{\log_e(x)}{\log_e(b)}$$

に変換します。

```
In[39]:= Log[10, x]
```

```
Out[39]=  $\frac{\text{Log}[x]}{\text{Log}[10]}$ 
```

```
In[40]:= Log[2, x]
```

```
Out[40]=  $\frac{\text{Log}[x]}{\text{Log}[2]}$ 
```

指数関数 e^x (または $\exp(x)$) は「**Exp**[x]」のように入力することで使用することができます。

```
In[41]:= Exp[x]
```

```
Out[41]=  $e^x$ 
```

平方根 \sqrt{x} は「**Sqrt**[x]」のように入力することで使用することができます。

```
In[42]:= Sqrt[2]
```

```
Out[42]=  $\sqrt{2}$ 
```

```
In[43]:=  $\sqrt{2}$ 
```

```
Out[43]=  $\sqrt{2}$ 
```

立方根 $\sqrt[3]{x}$ またはそれ以上の累乗根 ($\sqrt[n]{x}$) については、入力パレットで入力することができます。下記の出力結果からもわかるように

$$\sqrt[n]{x} = x^{\frac{1}{n}}$$

が成り立ち、乗ベキの形で表せることがわかります。

```
In[44]:=  $\sqrt[3]{2}$ 
```

```
Out[44]=  $2^{1/3}$ 
```

```
In[45]:=  $\sqrt[4]{2}$ 
```

```
Out[45]=  $2^{1/4}$ 
```

Mathematica カーネルが数式を評価するとき、数式は、直接カーネルが処理できる**完全形**という形式に変換されます。この形式は、私たちがフロントエンド(ノートブック)から覗き見る数式とは異なり、関数のみによって表現されています。この形式に変換するには関数「**FullForm**[変数(数式)]」を下記のように使用します。 $\sqrt{2}$ は「 $2^{(1/2)}$ 」と入力したものと同じですから、下記の例と比較すると、

乗ベキ「 x^y 」の完全形は「**Power**[x,y]」

割り算(有理数)「 m/n 」の完全形は「**Rational**[m,n]」

で表されていることがわかります。ただし、**m** と **n** には具体的な整数値が入ります。

```
In[46]:= FullForm[√2]

Out[46]//FullForm=
Power[2, Rational[1, 2]]

In[47]:= FullForm[∛2]

Out[47]//FullForm=
Power[2, Rational[1, 3]]
```

四則演算「+」、「-」、「*」、「/」の完全形も関数として表されます。

演算子	関数	使用例	使用例の完全形
+	Plus []	x+y+z	Plus[x,y,x]
*	Times []	x*y*z	Times[x,y,x]
^	Power []	x^y	Power[x,y]
-	Minus []	x-y	Plus[x,Times[-1,y]]
/	Divide []	x/y	Times[x,Power[y,-1]]

引き算と割り算は、それぞれ

$$x - y = x + (-y) = x + (-1) \times y,$$

$$\frac{x}{y} = x \times \frac{1}{y} = x \times y^{-1}$$

のように変形することができるので、引き算と割り算の関数を必要としません。すなわち、引き算は足し算と掛け算、割り算は掛け算と乗ベキに置き換えて表現することができます。このように Mathematica は数学の厳密な理論にしたがって数式の評価を行います。

```
In[48]:= FullForm[x + y]

Out[48]//FullForm=
Plus[x, y]

In[49]:= FullForm[x - y]

Out[49]//FullForm=
Plus[x, Times[-1, y]]

In[50]:= FullForm[x * y]

Out[50]//FullForm=
Times[x, y]

In[51]:= FullForm[x / y]

Out[51]//FullForm=
Times[x, Power[y, -1]]

In[52]:= FullForm[x ^ y]

Out[52]//FullForm=
Power[x, y]
```


ユーザーが独自の関数を定義することもできます。ユーザーが定義する関数は小文字から始まる文字列の関数名を使用することになっています。例えば、 $f(x) = x^2 + 2x + 1$ や $g(x) = ax^2 + bx + c$ などを関数として定義するには記号「:=」を用いて下記のように入力します。「x_」や「a_」は関数の変数であることを表しています。

```
In[53]:= f[x_] := x^2 + 2 x + 1
In[54]:= g[x_, a_, b_, c_] := a x^2 + b x + c
```

Mathematica では、記号「=」は「左辺に右辺を代入する」という意味に用いられ、「左辺は右辺に等しい」という意味ではありません。「左辺は右辺に等しい」という意味を表すには記号「==」を用います。例えば、「x=1」や「x=y」は「xに1を代入する」や「xにyを代入する」という意味になり、「x==1」や「x==y」は「xは1に等しい」や「xはyに等しい」という意味になります。また、関数名と同様に、ユーザーが独自に設定する変数(定数)の変数名は、小文字から始まる文字列を使用することになっています。

1.2 Mathematica でいろいろな計算をしてみよう (中高校生向け)

Mathematica は、中学校や高等学校で学習する初等的な数学の計算を容易に行うことができます。この節では、その一例を紹介します。

■ 整数に関する演算

関数「GCD[]」を用いると最大公約数を求めることができます。また、関数「LCM[]」を用いると最小公倍数を求めることができます。例は、123, 456, 789 の最大公約数と最小公倍数です。

```
In[1]:= GCD[123, 456, 789]
Out[1]= 3
In[2]:= LCM[123, 456, 789]
Out[2]= 4917048
```

メルセンヌ予想¹¹の中で $2^{67} - 1$ と $2^{257} - 1$ が素数とならないことを Mathematica で確かめてみましょう。関数「FactorInteger[]」を用いると整数を素因数分解することができます。例は、 $2^{67} - 1$ の素因数分解です。

```
In[3]:= FactorInteger[2^67 - 1]
Out[3]= {{193707721, 1}, {761838257287, 1}}
```

$2^{257} - 1$ のように非常に大きな素数で合成された数の素因数分解は、コンピュータを使っても非常に多くの時間を要します¹²。関数「PrimeQ[]」を用いると素数判別を行うことができます。例は、 $2^{257} - 1$ の素数判別です。

¹¹メルセンヌ(フランス; 1588年~1648年)は「257までの $n = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$ (すべて素数)のときに $2^n - 1$ も素数である」と予想しましたが、実際には、 n が 67 と 257 のとき合成数となり、61, 89, 107 のとき素数となることが知られています。

¹²公開鍵暗号の安全性は、素因数分解の難しさに依存しています。

```
In[4]:= PrimeQ[2^257 - 1]
Out[4]= False
```

■ 組み合わせに関する演算

記号「!」を用いると階乗の値を求めることができます。例は、40!の値を求めています。

```
In[5]:= 40!
Out[5]= 815915283247897734345611269596115894272000000000
```

関数「Binomial[n,m]」を用いると二項係数 $\frac{n!}{m!(n-m)!}$ の値を求めることができます。例は、 $\frac{10!}{3!(10-3)!}$ の値を求めています。

```
In[6]:= Binomial[10, 3]
Out[6]= 120
```

■ 数式の操作

関数「Expand[]」を用いると多項式の積を展開することができます。例は、 $(x+2y+1)(3x-y+5)$ を展開しています。

```
In[7]:= Expand[(x + 2 y + 1) (3 x - y + 5)]
Out[7]= 5 + 8 x + 3 x^2 + 9 y + 5 x y - 2 y^2
```

関数「Factor[]」を用いると多項式を因数分解することができます。例は、 $5 + 8x + 3x^2 + 9y + 5xy + 2y^2$ を因数分解しています。

```
In[8]:= Factor[5 + 8 x + 3 x^2 + 9 y + 5 x y - 2 y^2]
Out[8]= (5 + 3 x - y) (1 + x + 2 y)
```

三角関数についても、関数「Expand[]」を用いると合成された三角関数に展開することができます。例は、 $\sin(x+y)$ と $\cos(x+y)$ を展開しています。

```
In[9]:= Expand[Sin[x + y], Trig -> True]
Expand[Cos[x + y], Trig -> True]
Out[9]= Cos[y] Sin[x] + Cos[x] Sin[y]
Out[10]= Cos[x] Cos[y] - Sin[x] Sin[y]
```

同様に、関数「Factor[]」を用いると簡素な引数を持つ三角関数の積に分解することができます。例は、 $\sin(x)\cos(y) + \cos(x)\sin(y)$ と $\cos(x)\cos(y) - \sin(x)\sin(y)$ を分解しています。

```
In[11]:= Factor[Sin[x] Cos[y] + Cos[x] Sin[y], Trig -> True]
Factor[Cos[x] Cos[y] - Sin[x] Sin[y], Trig -> True]
Out[11]= Sin[x + y]
Out[12]= Cos[x + y]
```

■ 連立方程式

関数「`Solve[]`」を用いると連立方程式の解を求めることができます。例は、連立方程式

$$\begin{cases} 2x + y = 3 \\ 4x - y = 1 \end{cases}$$

の解を求めています。

```
In[13]:= Solve[{2 x + y == 3, 4 x - y == 1}, {x, y}]
```

```
Out[13]= {{x -> 2/3, y -> 5/3}}
```

■ 微分

関数「`D[]`」を用いると関数を偏微分することができます。例は、 x^n の偏微分を行っています。

```
In[14]:= D[x^n, x]
```

```
Out[14]= n x^{-1+n}
```

関数「`Dt[]`」を用いると関数を全微分することができます。例は、 $x^2 + y^2$ の全微分を行っています。結果は $2x + 2yy'$ です。

```
In[15]:= Dt[x^2 + y^2, x]
```

```
Out[15]= 2 x + 2 y Dt[y, x]
```

■ 積分

関数「`Integrate[]`」を用いると関数の不定積分を求めることができます。例は、 $\log(x)$ の不定積分を行っています。

```
In[16]:= Integrate[Log[x], x]
```

```
Out[16]= -x + x Log[x]
```

関数「`Integrate[]`」を用いると関数の定積分を求めることができます。例は、0 から π まで $\sin(x)$ を定積分しています。

```
In[17]:= Integrate[Sin[x], {x, 0, Pi}]
```

```
Out[17]= 2
```

1.3 Mathematica を使用する上で覚えておくこと

Mathematica を使用する上で、いくつか覚えておかなければならないことがあります。

■ 基本事項

ある問題を解決しようとするとき、計算途中で得られる結果を出力する必要がない場合があります。この様な時には、関数や定数の後ろに**セミコロン**(;)をつけることで、数式は通常どおり評価されますが、結果は出力されなくなります。

```
In[1]:= 1 + 1;
```

また、**コメント**を数式(プログラム)の中に記述することで数式を見やすくする働きがあります。コメントは「(*)と「*)」でくくります。

```
In[2]:= (* コメントです *)
```

■ 数式と結果の再利用

下記のように、数式「 $x=1$ 」を評価すると、Mathematicaは x を**大域的**なオブジェクト(変数)として扱います。そのため、「 $2x$ 」のように x を含む数式を評価すると、数式に含まれる x に1が代入され、その値が結果として返されます。すなわち、 x は定数として扱われることとなります。

```
In[3]:= x = 1
Out[3]= 1

In[4]:= 2 x
Out[4]= 2
```

また、下記のように x を関数の変数として扱う場合は、 x は**局所的**なオブジェクト(変数)として扱われるため、大域的なオブジェクトと区別されます。

```
In[5]:= h[x_] := 2 x + 1

In[6]:= h[1]
Out[6]= 3

In[7]:= h[2]
Out[7]= 5

In[8]:= h[100]
Out[8]= 201
```

入力行に付加される「 $In[n]:=$ 」を注意深く見ると、関数を定義する「 $:=$ 」が含まれていることがわかります。このことから、Mathematicaは $In(n)$ という統一された関数名で全ての入力行を管理していることがわかります。数学で表現すれば

$$\begin{aligned} In(1) &= f(x) = x + 1 \\ In(2) &= g(x) = x^2 + 2x + 3 \\ &\vdots \quad \quad \quad \vdots \\ In(n) &= h(x) = 2x + 1 \end{aligned}$$

のように定義されているのと同じです。従って、すでに評価した入力行を再利用することができます。 **$In[-n]$** のように負の整数を指定すると、現時点から n 行前の入力行を再評価します。また、 **$In[n]$** のように正の整数を指定すると、第 n 行の入力行を再評価します。例のように「 $In[-1]$ 」を評価すると直前の数式「 $h[100]$ 」が再評価されます。2つ目の例は、 x を1から3に代入し直したのち、「 $In[4](:=2x)$ 」を再評価しています。

```
In[9]:= In[-1]
Out[9]= 201

In[10]:= x = 3;
Out[10]= 3

In[11]:= In[4]
Out[11]= 6
```

出力結果に付加される「Out[n]=」も「In[n]:=」と同様で、MathematicaはOut(n)という統一された変数名で結果を管理しており、出力結果を再利用することができます。Out[-n]のように負の整数を指定すると、現時点からn行前の出力結果を再利用することができます。また、Out[n]のように正の整数を指定すると、第n行の出力結果を再利用することができます。下記の例のように「%」を使って、1行前の出力結果「%」、4行前の出力結果「%%%」¹³、第8行の出力結果「%8」を再利用することもできます。

```
In[12]:= %
Out[12]= 6

In[13]:= %%%
Out[13]= 201

In[14]:= Out[-3]
Out[14]= 6
```

```
In[15]:= %8
Out[15]= 201

In[16]:= Out[8]
Out[16]= 201
```

■ ヘルプを利用する

Mathematicaには、非常に優れたヘルプ機能が備わっています。上手に利用してMathematicaを使いこなしましょう。メニューバーの「ヘルプ」→「ヘルプ」を選択することで、図1.9のようなノーブックで作られた「ヘルプ」が利用できます。こちらのほうが、より詳しい情報を得ることができます。「ヘルプ」ウインドウ内にある「Mathematica ツアー」や「デモ」を閲覧することで、Mathematicaの様々な機能を垣間見ることもできます。

¹³n行前の出力結果を再利用したい場合は、%をn個入力します。

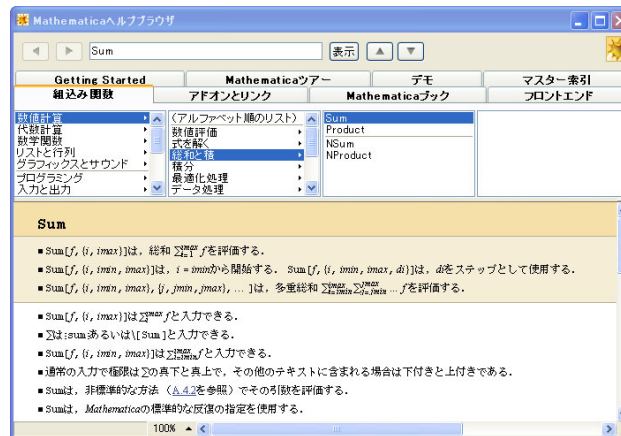


図 1.9: 「ヘルプ」 ウィンドウ

ノートブックからヘルプ機能を利用する方法について解説します。「?**関数名 (変数名)**」と入力することで、関数 (変数) についての情報を得ることができます。例えば、関数「Sum」を調べるには「?Sum」と入力します。

```
In[17]:= ? Sum

Sum[f, {i, imax}]は、総和  $\sum_{i=1}^{imax} f$  を評価する。
Sum[f, {i, imin, imax}]は、 $i = imin$  から開始する。
Sum[f, {i, imin, imax, di}]は、 $di$  をステップとして使用する。
Sum[f, {i, imin, imax}, {j, jmin, jmax}, ... ]は、多重総和  $\sum_{i=imin}^{imax} \sum_{j=jmin}^{jmax} \dots f$  を評価する。 詳細
```

ユーザーが定義した関数や定数についても調べることができます。

```
In[18]:= ? h

Global`h

h[x_] := 2 x + 1
```

関数名が曖昧なときには、「*」¹⁴を使ってパターンにマッチする関数の一覧を得ることもできます。

```
In[19]:= ? Su*

System`
Subresultants          SubValues
Subscript              SugarCube
SubscriptBox          Sum
SubscriptBoxOptions   Superscript
Subscripted           SuperscriptBox
Subsuperscript        SuperscriptBoxOptions
SubsuperscriptBox    SurfaceColor
SubsuperscriptBoxOptions SurfaceGraphics
Subtract              SuspendPacket
SubtractFrom
```

¹⁴*は「任意の長さを持つ文字列にマッチする」を意味します。関連項目として「正規表現」。

第2章 グラフィック編

2.1 グラフィック関数の紹介

Mathematica の主なグラフィック関数には、表 2.1 のようなものがあります。

グラフィックの種類	2次元	3次元
基本図形	<code>Graphics[]</code>	<code>Graphics3D[]</code>
関数	<code>Plot[]</code>	<code>Plot3D[]</code>
媒介変数表示の関数	<code>ParametricPlot[]</code>	<code>ParametricPlot3D[]</code>

表 2.1: Mathematica の主なグラフィック関数

点「`Point[]`」、直線「`Line[]`」、円「`Circle[]`」、多角形(ポリゴン)「`Polygon[]`」、立方体「`Cuboid[]`」などの基本的な図形を描くには、関数「`Show[]`」と共に「`Graphics[]`」(2次元)または「`Graphics3D[]`」(3次元)を使用します。

`Show[obj, option]`

`Show[{obj1, obj2, ..., objn}, option]` (複数のオブジェクト $obj_1, obj_2, \dots, obj_n$)

`Graphics[obj, option]` (2次元)

`Graphics[{obj1, obj2, ..., objn}, option]` (2次元, 複数のオブジェクト $obj_1, obj_2, \dots, obj_n$)

`Graphics3D[obj, option]` (3次元)

`Graphics3D[{obj1, obj2, ..., objn}, option]` (3次元, 複数のオブジェクト $obj_1, obj_2, \dots, obj_n$)

例を下記に挙げておきます。例に使用されているオプション「`AspectRatio -> Automatic`」は2次元のグラフィックの縦横比を1に設定します¹。また、オプション「`PlotRange`」はグラフィックの描画範囲の設定を行います。

`PlotRange -> {{xmin, xmax}, {ymin, ymax}}` (2次元)

`PlotRange -> {{xmin, xmax}, {ymin, ymax}, {zmin, zmax}}` (3次元)

```
Show[Graphics[{RGBColor[0, 0, 1], Circle[{0, 0}, 1]}]]
Show[Graphics[Circle[{0, 0}, 1], AspectRatio -> Automatic]
Show[Graphics[{Circle[{-1, -1}, 1], Circle[{1, 1}, 1]}],
{AspectRatio -> Automatic, PlotRange -> {{-5, 5}, {-5, 5}}}]
```

アニメーションを作成してみましょう。まず、関数「`Do[]`」などを使って、オブジェクトを連続的に少しずつ動かしたグラフィックを作成します。作成した全てのグラフィックセルを選択し、メニューバーから「セル」→「グラフィックスのアニメーション化」(またはショートカットキー「`[Ctrl]+y`」)を選択します。

¹3次元のグラフィックはデフォルトで `AspectRatio -> Automatic` が設定されています。

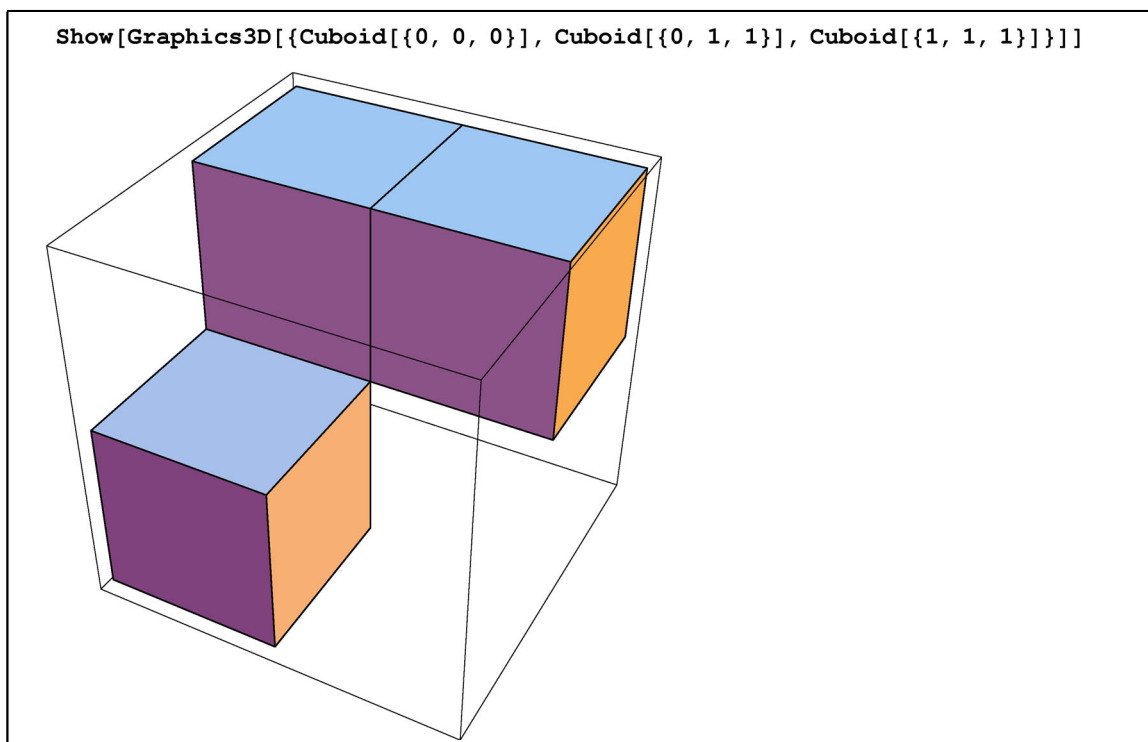
Do[obj(k), {k, k_{min}, k_{max}, d_k}] (d_k は増分を表す)
 Do[obj(k), {k, k_{min}, k_{max}}] (d_k 省略時の増分は1です)

下記は、アニメーションを作成するプログラムの例です。

```
Do[Show[Graphics[
  {Circle[{Sin[k], Cos[k]}, 1], Circle[{Sin[-k + Pi], Cos[-k + Pi]}, 0.5]}],
  {AspectRatio -> Automatic, PlotRange -> {{-5, 5}, {-5, 5}}}],
  {k, 0, 2 Pi - Pi / 12, Pi / 12}]
```

下記の例は、関数「Graphics3D[]」を用いて複数のオブジェクトを描画したものです。

```
Show[Graphics3D[{Cuboid[{0, 0, 0}], Cuboid[{0, 1, 1}], Cuboid[{1, 1, 1}]}]]
```



関数「Plot[]」と「Plot3D[]」は、

$$y = f(x) \quad (2 \text{次元})$$

$$z = g(x, y) \quad (3 \text{次元})$$

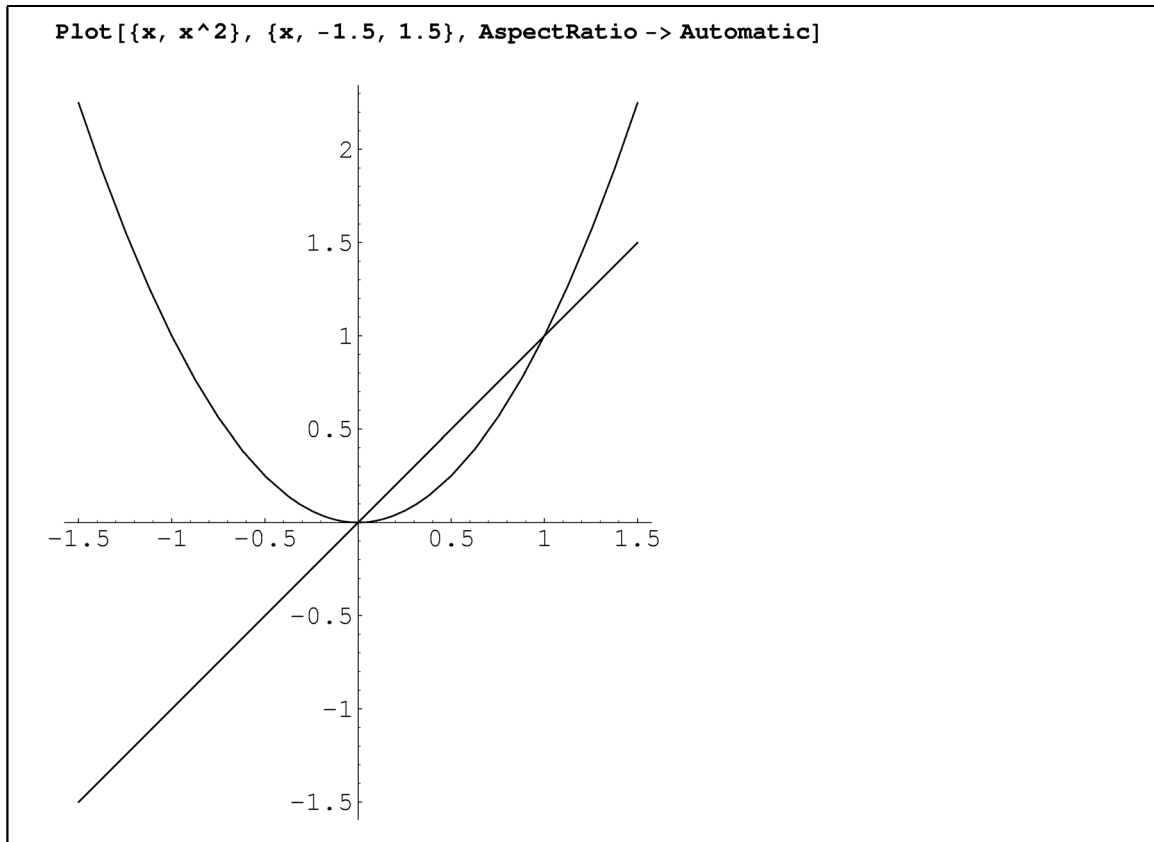
の形をした関数のグラフィックを描画します。

Plot[f(x), {x, x_{min}, x_{max}}] (2次元)

Plot[{f₁(x), f₂(x), ..., f_n(x)}, {x, x_{min}, x_{max}}] (2次元, 複数の関数 f₁(x), f₂(x), ..., f_n(x))

Plot3D[g(x, y), {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] (3次元)

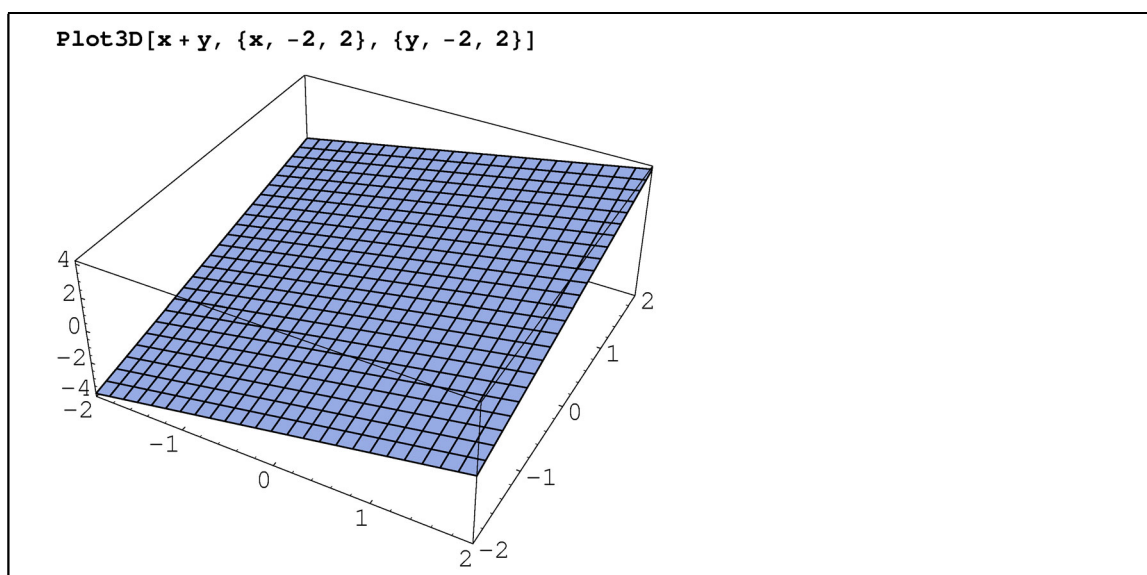
下記の例は、関数 $y = x$ と $y = x^2$ を描画したものです。



下記のプログラムは、関数 $y = f(x) = x^2$ 上の点 $(x, y) = (k, f(k))$ の接線 $y = f'(k)(x - k) + f(k)$ を次々に変化させるアニメーションです。

```
f[x_] := x^2
Do[Plot[{f[x], f'[k] (x - k) + f[k]}, {x, -2, 2},
  {AspectRatio -> Automatic, PlotRange -> {{-2, 2}, {-1, 4}}}], {k, -2, 2, 0.1}]
```

下記の例は、関数 $z = x + y$ を描画したものです²。



関数「ParametricPlot[]」と「ParametricPlot3D[]」は、 t_1, t_2, \dots, t_m を媒介変数とする

$$\begin{cases} x = f_x(t_1, t_2, \dots, t_m) \\ y = f_y(t_1, t_2, \dots, t_m) \end{cases} \quad (2 \text{次元})$$

$$\begin{cases} x = g_x(t_1, t_2, \dots, t_m) \\ y = g_y(t_1, t_2, \dots, t_m) \\ z = g_z(t_1, t_2, \dots, t_m) \end{cases} \quad (3 \text{次元})$$

の形をした関数のグラフィックを描画します。

ParametricPlot[$\{f_x, f_y\}, \{t_1, t_{1_{min}}, t_{1_{max}}\}, \dots, \{t_m, t_{m_{min}}, t_{m_{max}}\}, option]$ (2次元)

ParametricPlot[$\{\{f_{x1}, f_{y1}\}, \dots, \{f_{xn}, f_{yn}\}\}, \{t_1, t_{1_{min}}, t_{1_{max}}\}, \dots, \{t_m, t_{m_{min}}, t_{m_{max}}\}, option]$ (2次元, 複数の関数)

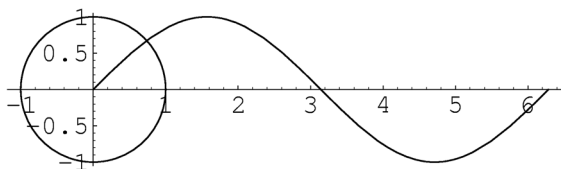
ParametricPlot3D[$\{g_x, g_y, g_z\}, \{t_1, t_{1_{min}}, t_{1_{max}}\}, \dots, \{t_m, t_{m_{min}}, t_{m_{max}}\}, option]$ (3次元)

ParametricPlot3D[$\{\{g_{x1}, g_{y1}, g_{z1}\}, \dots, \{g_{xn}, g_{yn}, g_{zn}\}\}, \{t_1, t_{1_{min}}, t_{1_{max}}\}, \dots, \{t_m, t_{m_{min}}, t_{m_{max}}\}, option]$ (3次元, 複数の関数)

² $z = x + y$ は $x + y - z = 0$ と変形すれば、平面を表す関数であることがわかります。

下記に例を挙げておきます。

```
ParametricPlot[{{Cos[t], Sin[t]}, {t, Sin[t]}}, {t, 0, 2 Pi},
  AspectRatio -> Automatic]
```

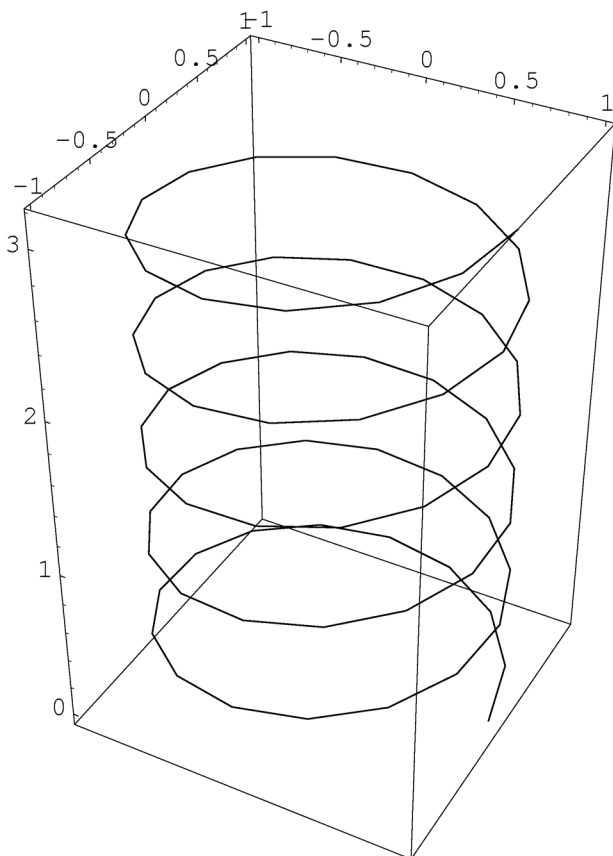


下記のプログラムは、関数 $\begin{cases} x = t \\ y = \sin(t) \end{cases}$ の位相を次々に変化させるアニメーションです。

```
Do[ParametricPlot[{t, Sin[t + k]}, {t, 0, 2 Pi}, AspectRatio -> Automatic],
  {k, 0, 2 Pi - Pi / 12, Pi / 12}]
```

下記の例は、関数 $\begin{cases} x = \cos(t) \\ y = \sin(t) \\ z = t/10 \end{cases}$ を描画したもので、螺線を描きます。

```
ParametricPlot3D[{Cos[t], Sin[t], t / 10}, {t, 0, 10 Pi},
  AspectRatio -> Automatic]
```

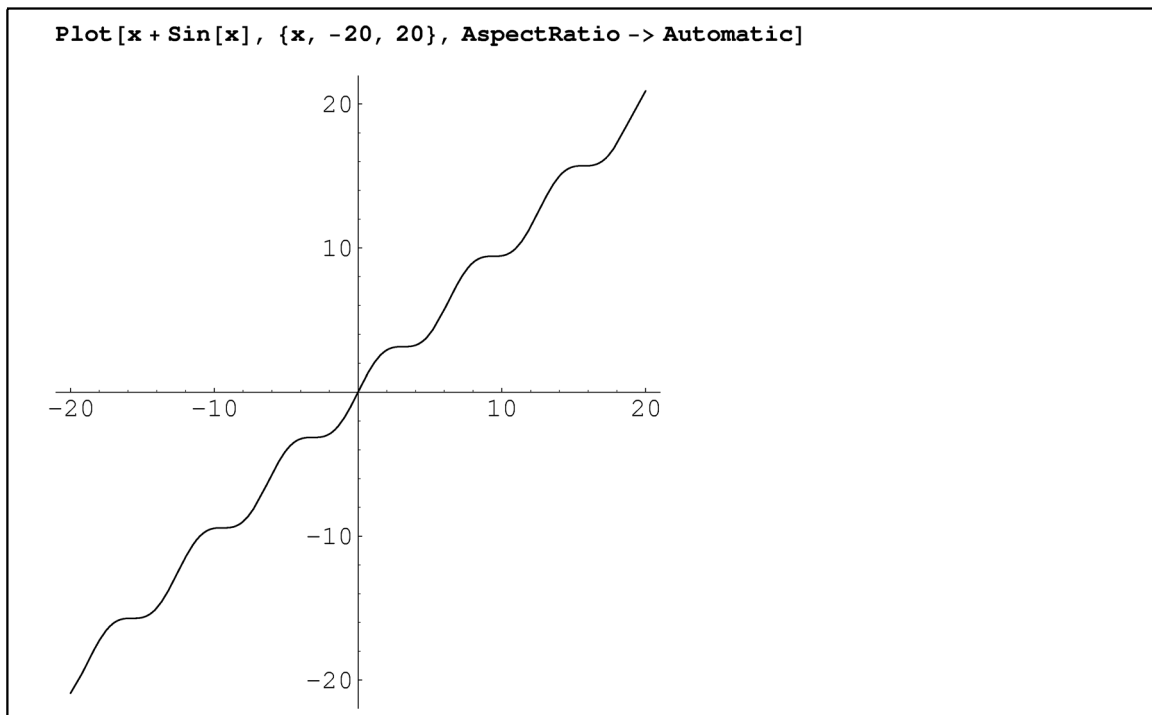


下記の例は、螺旋線をバネのように弾ませるアニメーションです。

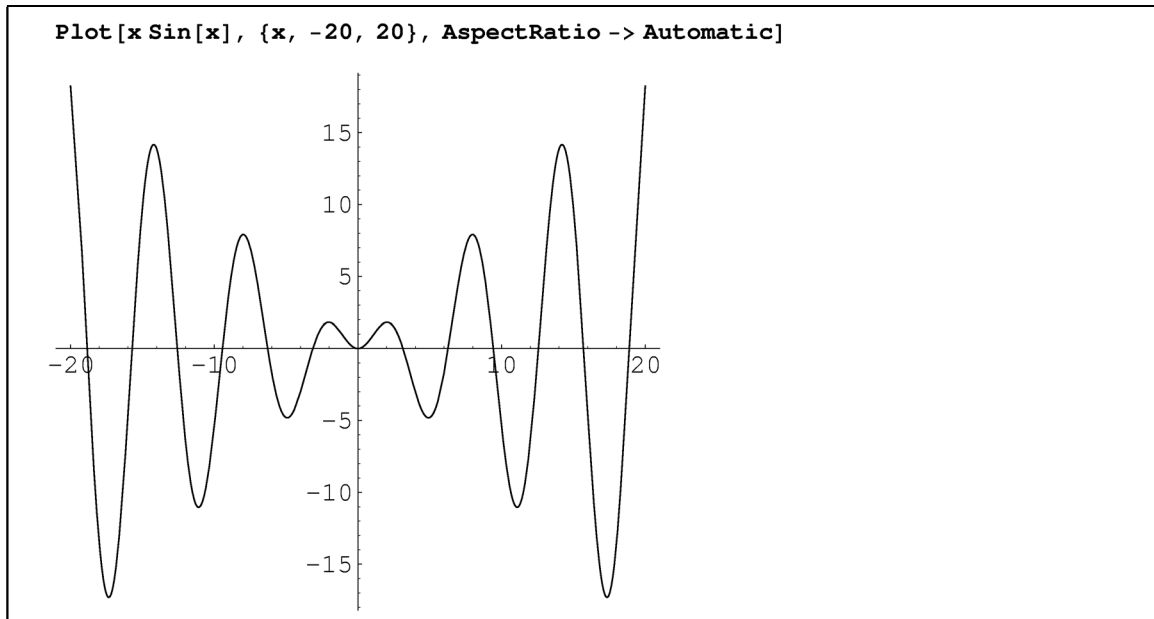
```
Do[ParametricPlot3D[{Cos[t], Sin[t], t/k}, {t, 0, 10 Pi},  
  {AspectRatio -> Automatic, PlotRange -> {{-1, 1}, {-1, 1}, {0, 4}}}],  
  {k, 10, 20}]  
Do[ParametricPlot3D[{Cos[t], Sin[t], t/k}, {t, 0, 10 Pi},  
  {AspectRatio -> Automatic, PlotRange -> {{-1, 1}, {-1, 1}, {0, 4}}}],  
  {k, 19, 9, -1}]
```

2.2 関数のグラフィック

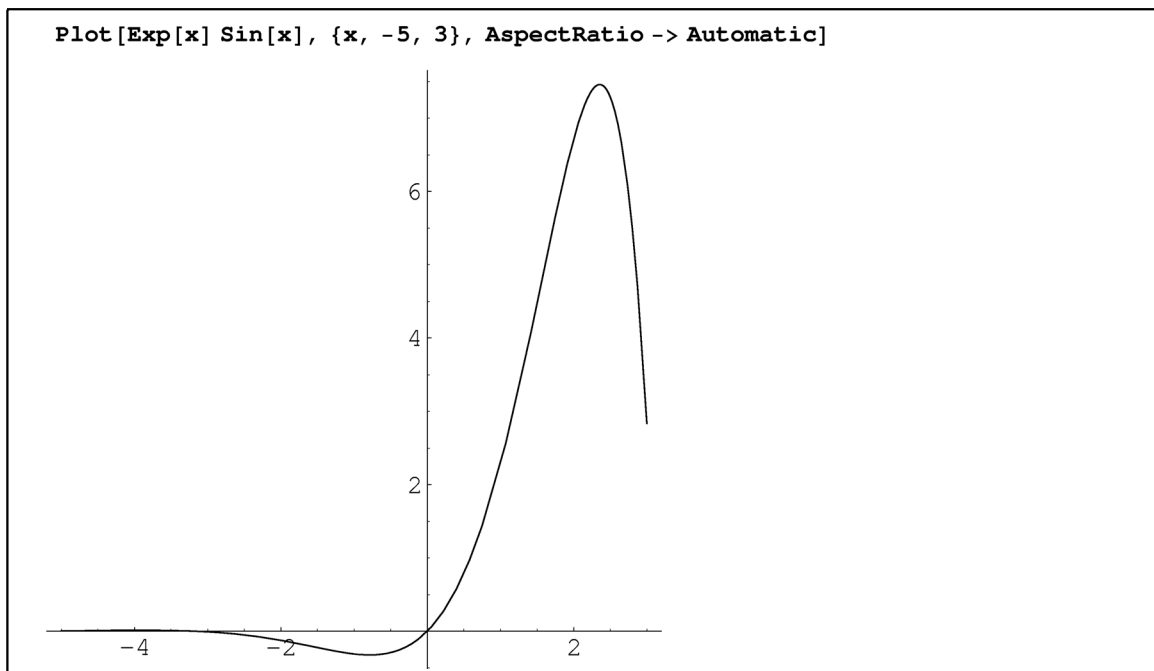
下記の例は、関数 $y = x + \sin(x)$ のグラフィックです。



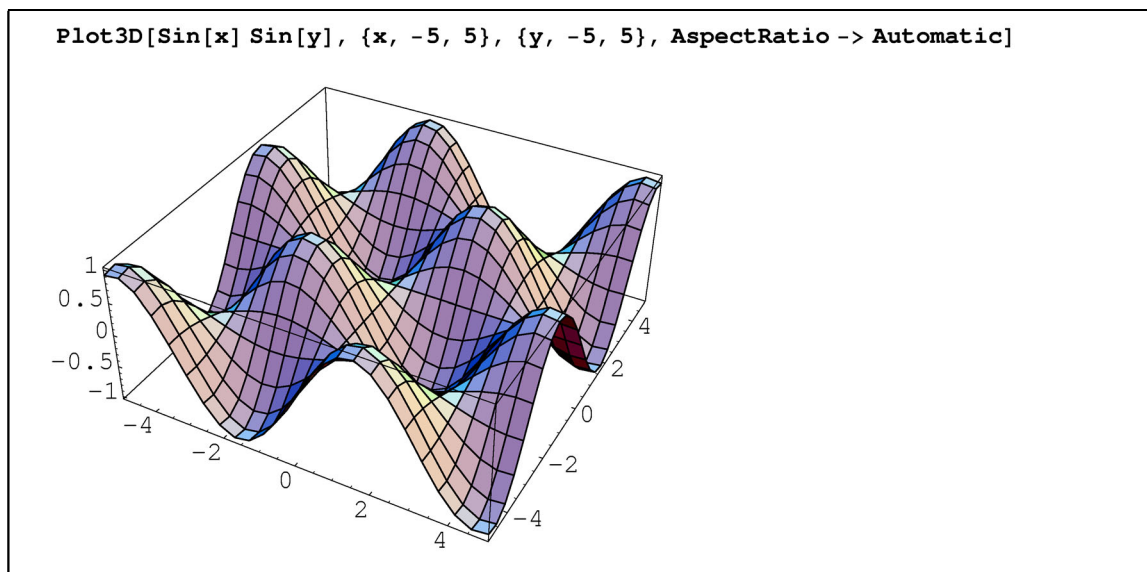
下記の例は、関数 $y = x \sin(x)$ のグラフィックです。



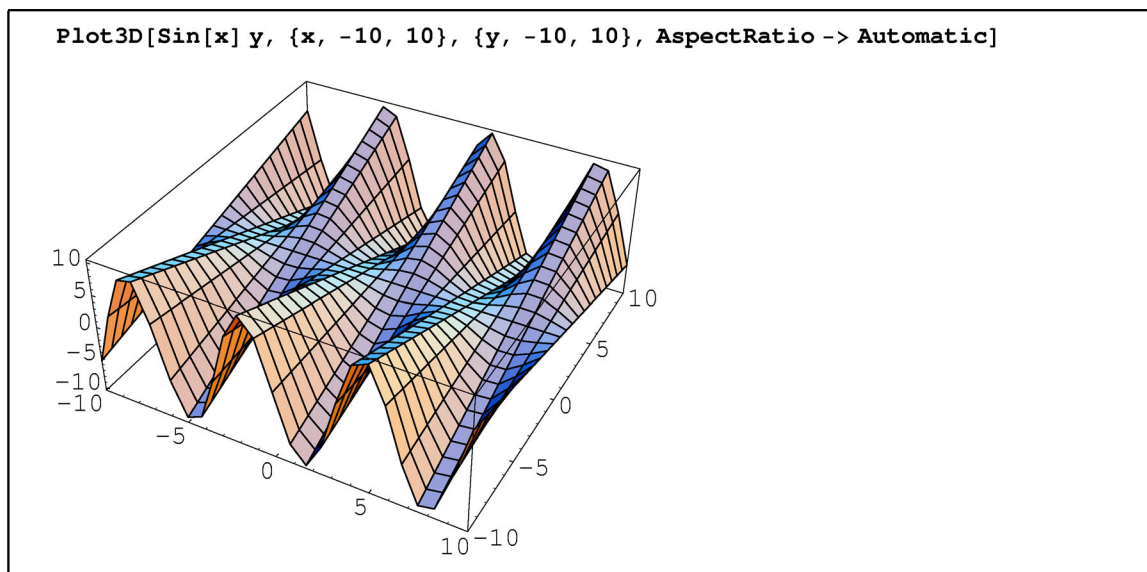
下記の例は、関数 $y = e^x \sin(x)$ のグラフィックです。



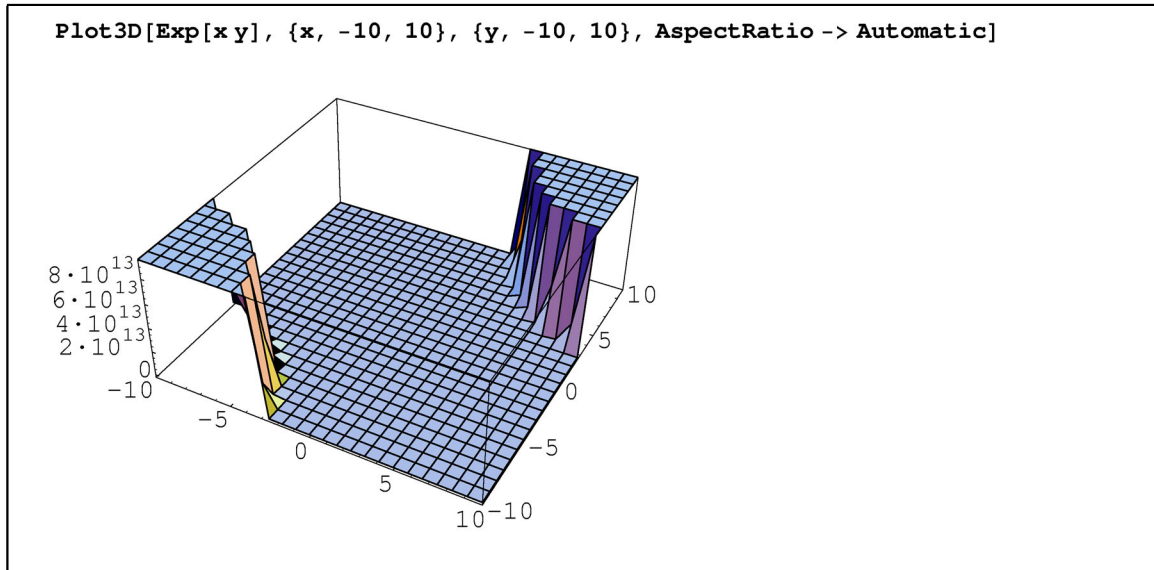
下記の例は、関数 $z = \sin(x) \sin(y)$ のグラフィックです。



下記の例は、関数 $z = y \sin(x)$ のグラフィックです。



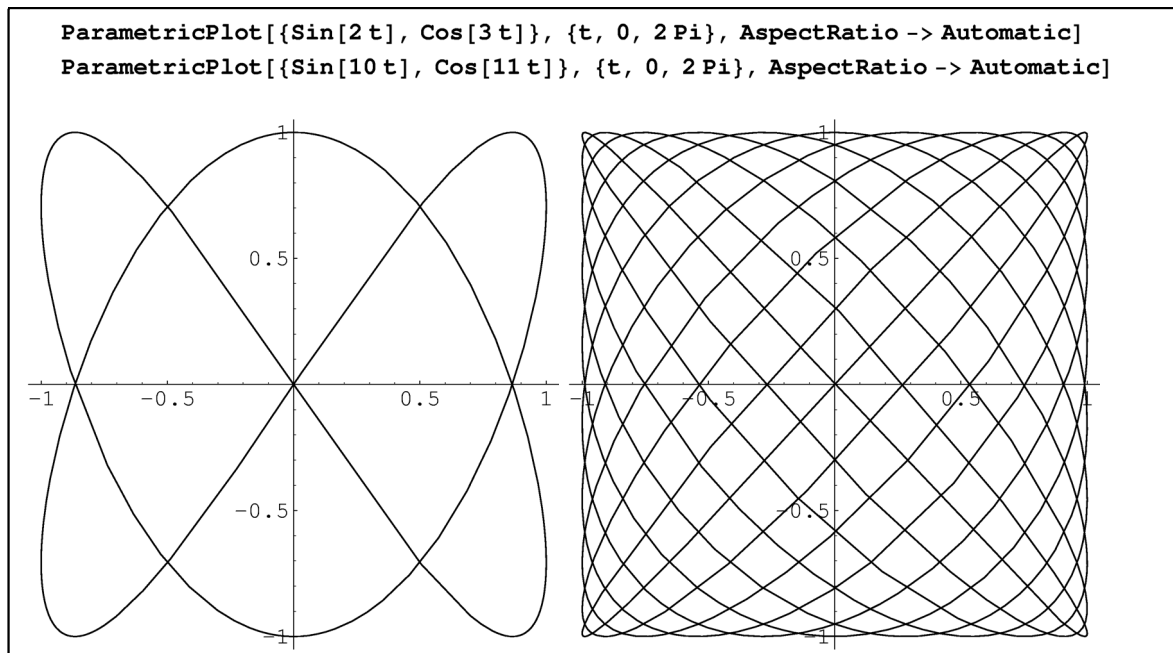
下記の例は、関数 $z = e^{xy}$ のグラフィックです。



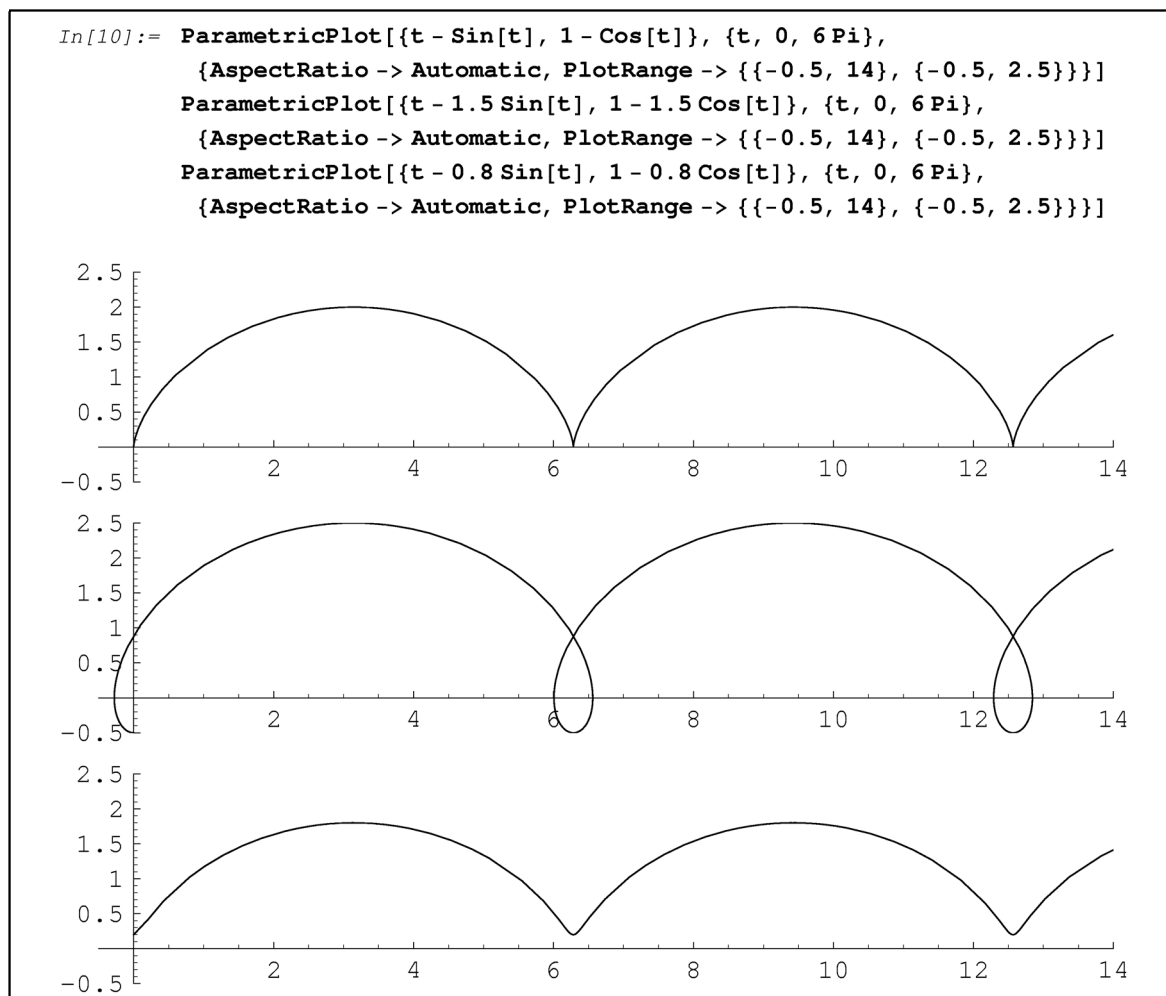
2.3 媒介変数表示の関数のグラフィック

関数 $\begin{cases} x = a \sin(mt) \\ y = a \sin(nt) \end{cases}$ のように表される曲線を **リサージュ曲線** と呼びます (a, m, n は定数)。下

記の例の左側は、 $a = 1, m = 2, n = 3$ のリサージュ曲線で、右側は、 $a = 1, m = 10, n = 11$ のリサージュ曲線です。



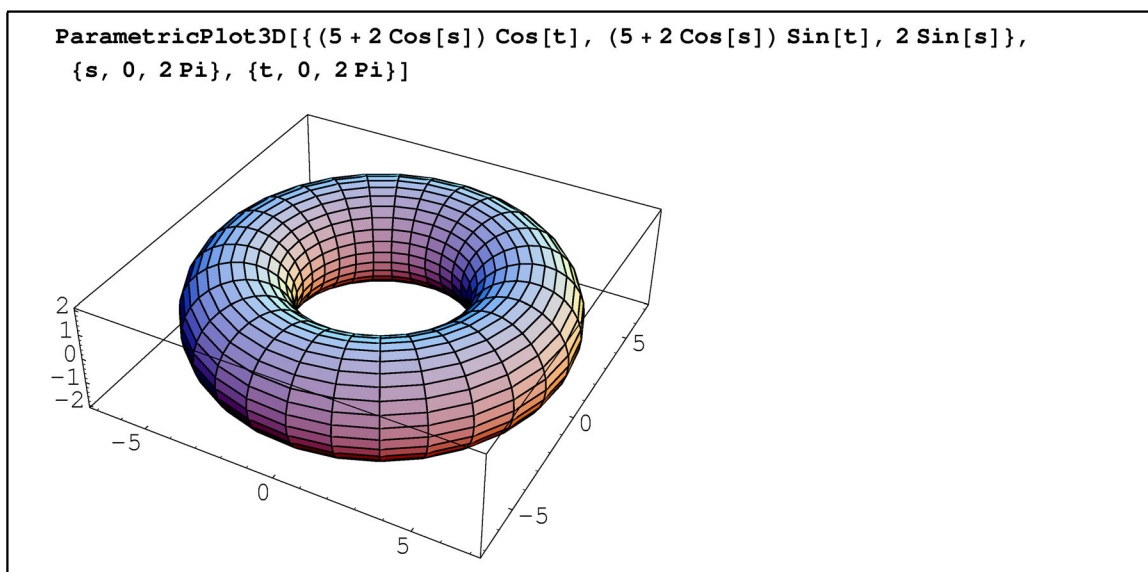
直線上を円が滑らないで回転していくとき、円周上のある1点が描く曲線を**サイクロイド曲線**と
いいます。この曲線は、関数 $\begin{cases} x = t - \sin(t) \\ y = 1 - \cos(t) \end{cases}$ によって表されます。また、直線上を円が滑らな
いで回転していくとき、円周上のある1点ではなく、円周上の内部または外部にある点が描く曲
線を**トロコイド曲線**と呼びます。この曲線は、関数 $\begin{cases} x = t - b \sin(t) \\ y = 1 - b \cos(t) \end{cases}$ によって表されます (b は
定数)。下記にサイクロイド曲線とトロコイド曲線の例を挙げておきます。



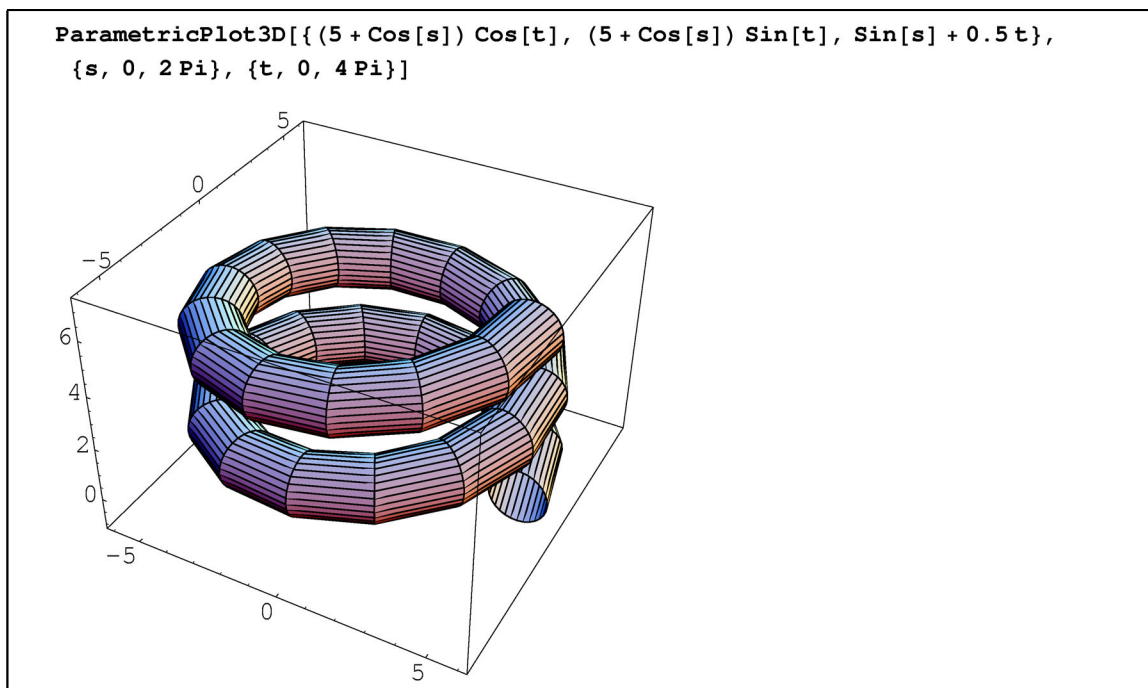
ドーナツ型をした**トーラス**は、関数

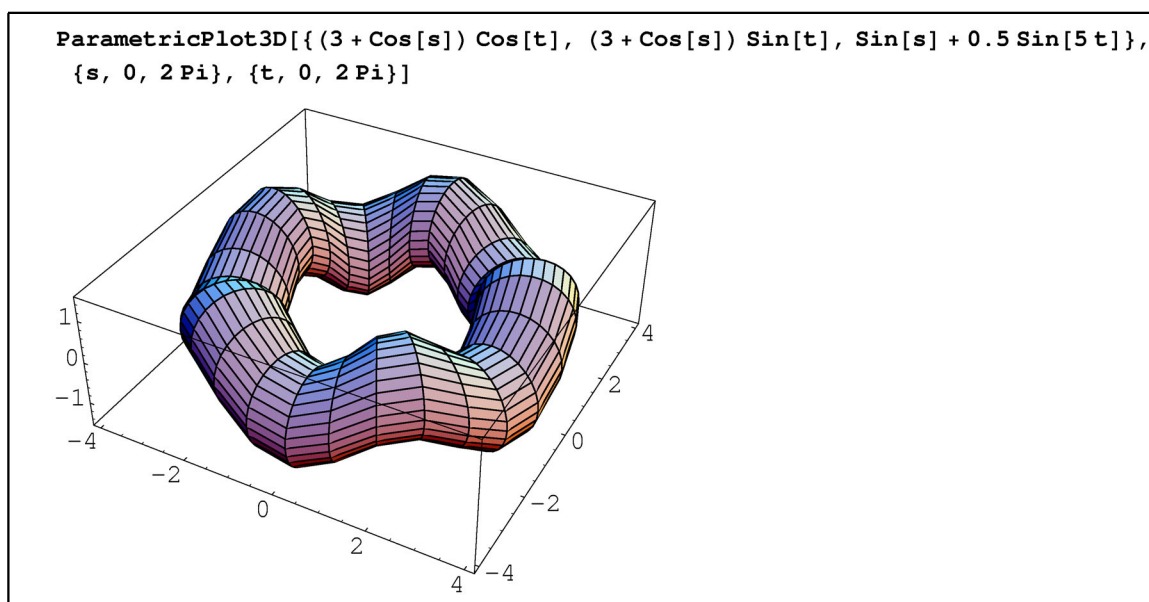
$$\begin{cases} x = (a + b \cos(s)) \cos(t) \\ y = (a + b \cos(s)) \sin(t) \\ z = c \cos(s) \end{cases}$$

で表すことができます。なお、 a, b, c は定数で、 a はトーラスの半径の値を、 b と c は管の縦と横の半径の値をそれぞれ設定します。下記の例は、 $a = 5, b = 2, c = 1$ のトーラスです。



トーラスを表す関数を少し変えると、下記の2つの例のようなグラフィックを描くことができます。





また、下記の2つの例のようなアニメーションを作成することができます。

```
Do[ParametricPlot3D[{(5 + Cos[s]) Cos[t], (5 + Cos[s]) Sin[t], Sin[s] + 0.5 t},
  {s, 0, 2 Pi}, {t, 0, k}, PlotRange -> {{-6, 6}, {-6, 6}, {-1, 8}},
  PlotPoints -> {10, 40}], {k, Pi / 6, 4 Pi, Pi / 6}]
```

```
Do[ParametricPlot3D[
  {(3 + Cos[s]) Cos[t], (3 + Cos[s]) Sin[t], Sin[s] + 0.5 Sin[5 t + k]},
  {s, 0, 2 Pi}, {t, 0, 2 Pi}, PlotRange -> {{-6, 6}, {-6, 6}, {-2, 2}},
  {k, 0, 2 Pi - Pi / 6, Pi / 6}]
```

2.4 多面体

全ての面が合同な正多角形で構成される**正多面体**は、四面体、六面体、八面体、12面体、20面体の5種類です。これら多面体の頂点の座標を全部計算するのは大変ですが、Mathematicaは、パッケージ「Graphics`Polyhedra`」を読み込むことで、多面体を容易に扱うことができます。Mathematicaで用いられる各多面体のオブジェクト名を表2.2に挙げておきます。

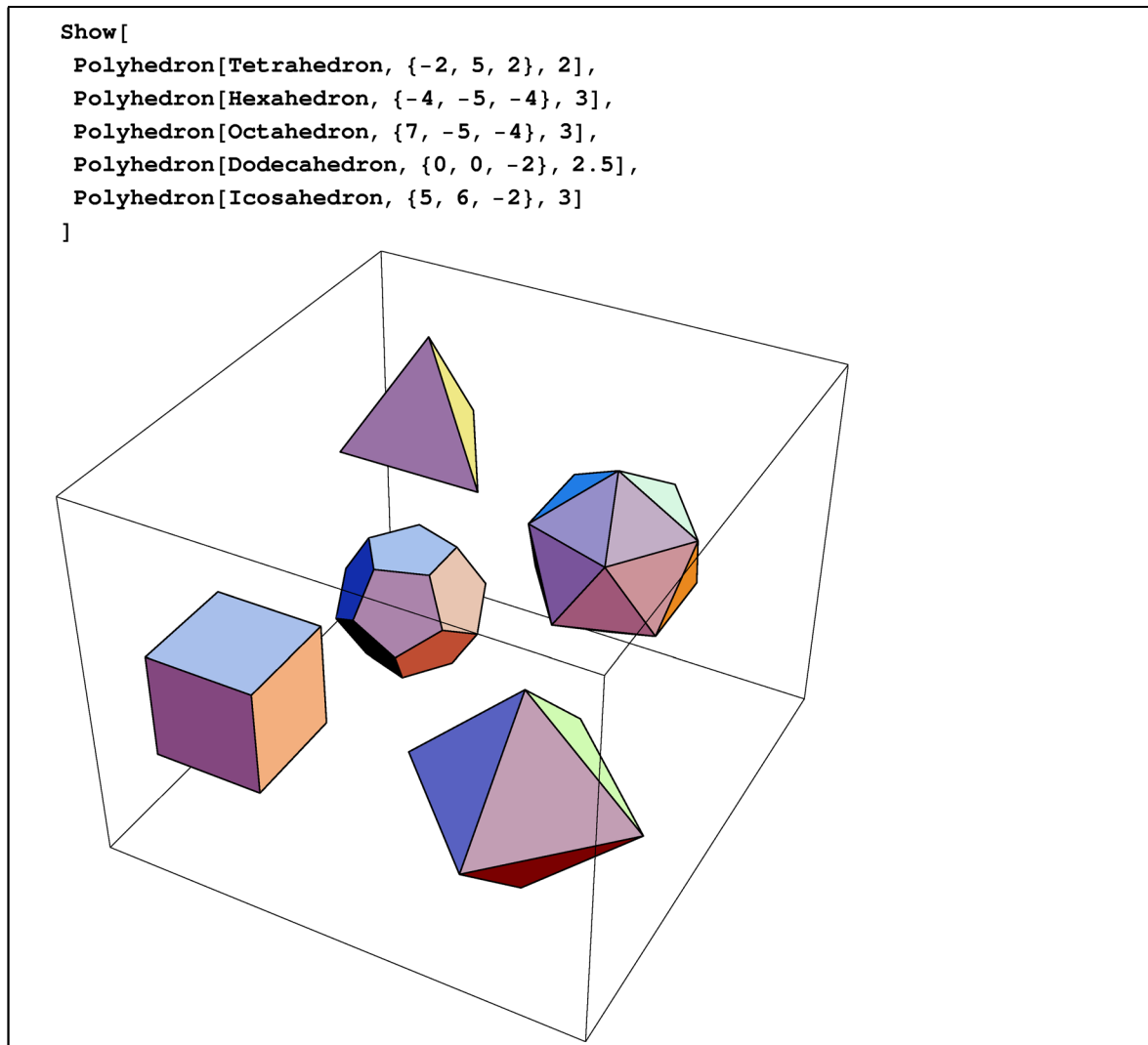
多面体名	オブジェクト名
四面体	Tetrahedron
六面体	Hexahedron
八面体	Octahedron
12面体	Dodecahedron
20面体	Icosahedron

表 2.2: 多面体名と Mathematica で用いられるオブジェクト名

多面体を描画するには

```
Show[Polyhedron[多面体のオブジェクト名,{ $x,y,z$ }, $r$ ]
```

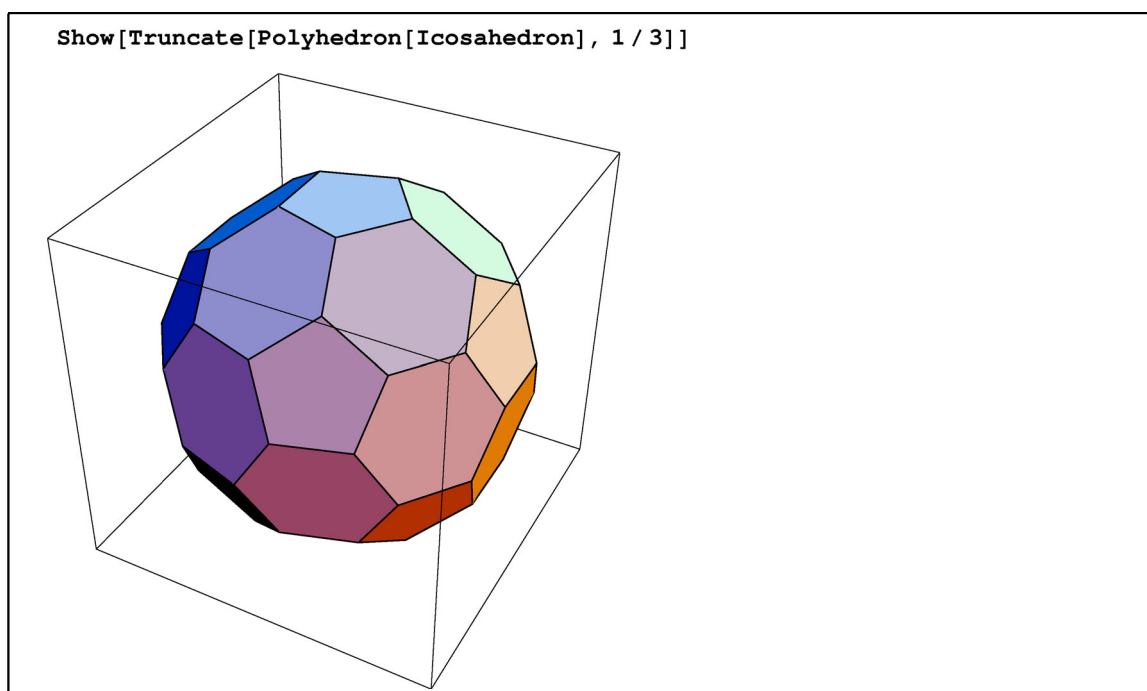
のように入力します。 (x,y,z) は多面体の中心を設定し、 r は拡大縮小率を設定します。下記に使用例を挙げておきます。



また、多面体の頂点を切り取る関数「`Truncate[]`」を使用すると、全ての面が(合同ではない)正多角形で構成される**準正多面体**を作成することが出来ます。使い方は、

```
Show[Truncate[Polyhedron[多面体のオブジェクト名,{x,y,z},r],s]]
```

のように入力します。 r は切り取る点の位置を示す割合を設定します。0から0.5の数値を設定することが出来ます。下記に例を挙げておきます。また、合わせて関数「`Truncate[]`」を利用したアニメーションの例を挙げておきます。

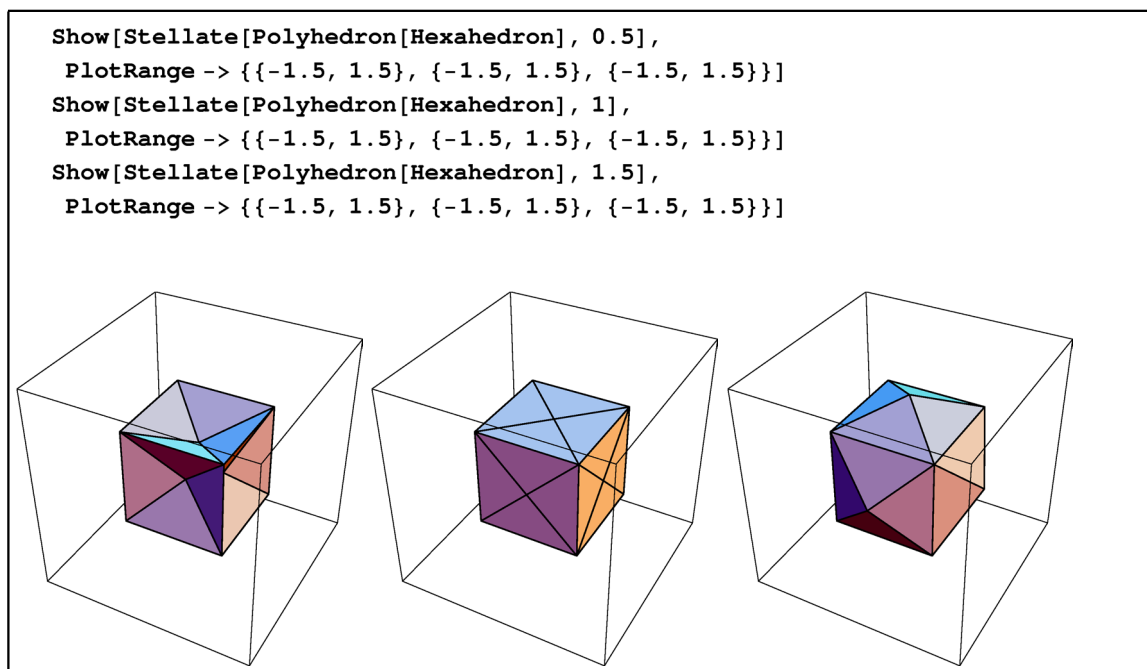


```
Do[Show[Truncate[Polyhedron[Icosahedron], k],
  PlotRange -> {{-1.5, 1.5}, {-1.5, 1.5}, {-1.5, 1.5}}, {k, 0, 0.5, 0.05}]
Do[Show[Truncate[Polyhedron[Icosahedron], k],
  PlotRange -> {{-1.5, 1.5}, {-1.5, 1.5}, {-1.5, 1.5}}, {k, 0.45, 0.05, -0.05}]
```

最後に、多面体を星状化する関数「`Stellate[]`」を紹介しておきます。使い方は、

```
Show[Stellate[Polyhedron[多面体のオブジェクト名,{x,y,z},r],t]]
```

のように入力します。 t は星の伸び具合を設定します。下記に例を挙げておきます。



関数「`Stellate[]`」を利用したアニメーションの例を2つほど挙げておきます。

```
Do[Show[Stellate[Polyhedron[Hexahedron], k],
PlotRange -> {{-5, 5}, {-5, 5}, {-5, 5}}, {k, 0, 5, 0.5}]
Do[Show[Stellate[Polyhedron[Hexahedron], k],
PlotRange -> {{-5, 5}, {-5, 5}, {-5, 5}}, {k, 4.5, 0.5, -0.5}]
```

```
Do[Show[Stellate[Polyhedron[Dodecahedron], k],
PlotRange -> {{-5, 5}, {-5, 5}, {-5, 5}}, {k, 1, 5, 0.5}]
Do[Show[Stellate[Polyhedron[Dodecahedron], k],
PlotRange -> {{-5, 5}, {-5, 5}, {-5, 5}}, {k, 4.5, 0.5, -0.5}]
```


付録 A 入力パレットの使い方

Mathematica は、バージョン 3 までは文字列 (コマンドライン形式¹) で数式を表現していましたが、バージョン 4 から私たちが日常使用しているような形で数式を表現できるようになりました。前者のような文字主体の形式を「**インプットフォーム (InputForm)**」と呼び、後者の日常的でグラフィカルな形式を「**スタンダードフォーム (StandardForm)**」と呼びます。

Mathematica にはスタンダードフォームによる入力を補助するために、「**入力パレット**²」と呼ばれるフロントエンドが付属しています。ここでは、もっとも基本的な入力を補助する入力パレット「**基本的な入力 (BasicInput)**」パレット (図 A.1) の使い方について説明します。「基本的な入力」パレット以外にも数多くの入力パレット (図 A.2) が付属し、メニューバーの「ファイル」→「パレット」から使用したいパレットをデスクトップに表示することができます。また、上級者向けにショートカットキー (エイリアス) による入力方法やインプットフォームからスタンダードフォームへ一括変換する方法について解説します。

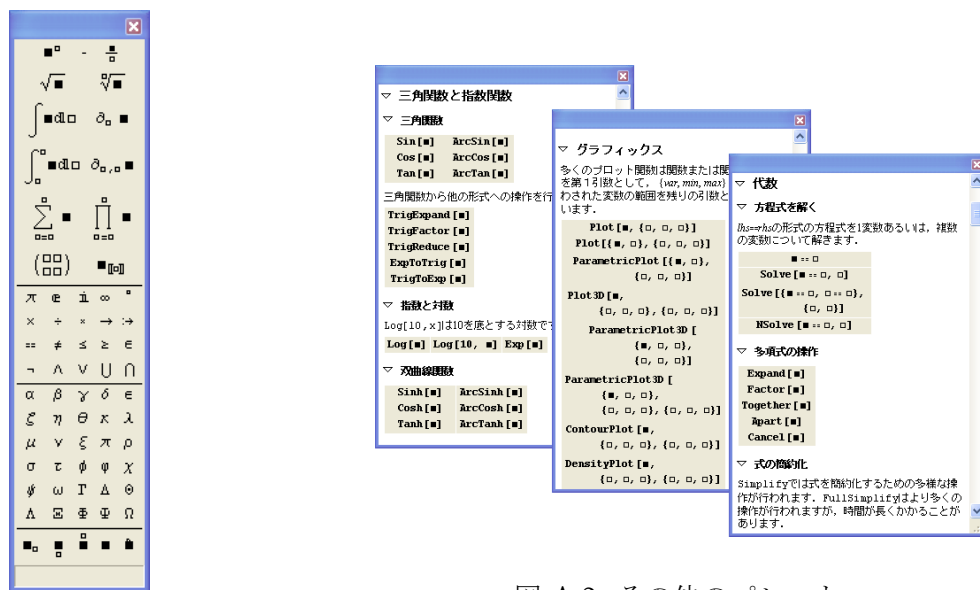


図 A.2: その他のパレット

図 A.1: 「基本的な入力」パレット

* 詳しい内容については、Mathematica に付属の「ヘルプ」をご覧ください。

¹ キーボードから入力を入力すると横一線に文字が出力されるような入出力形式。

² 入力パレットは自由にカスタマイズでき、自分だけのパレットを作成することができます。

■ 入力パレットによる入力

基本的には、入力パレットから「 $\sqrt{\square}$ 」や「 $\int_{\square}^{\square} \blacksquare d\square$ 」など入力したい数式のボタンをクリックし、ボックス「□」に必要な数式を入力していきます。□から□へカーソルを移動するには「[Tab]」(または「[→]」)キーを押します。もし、間違えて入力したボックスがあれば、何度か「[←]」キーを押して訂正位置までカーソルを動かし、間違えたボックスを訂正します。例として、数式

$$\int_0^{2\pi} \sin(x) dx$$

を入力してみましょう。まず、入力パレットからボタン「 $\int_{\square}^{\square} \blacksquare d\square$ 」をクリックします。

$$\int_{\blacksquare}^{\square} \square d\square$$

カーソルがある黒いボックス「■」に「0」を入力します。次の□にカーソルを移動するため、「[Tab]」キーを押します。

$$\int_0^{\square} \square d\square$$

カーソルが次の□に移動するので、■に「2」を入力し、さらにボタン「 π 」をクリックしてπを入力します。次の□にカーソルを移動するため、「[Tab]」キーを押します。

$$\int_0^{2\pi} \blacksquare d\square$$

カーソルが次の□に移動するので、■に「Sin[x]」を入力します。次の□にカーソルを移動するため、「[Tab]」キーを押します。

$$\int_0^{2\pi} \text{Sin}[\mathbf{x}] d\blacksquare$$

カーソルが最後の□に移動するので、■に「x」を入力します。

$$\int_0^{2\pi} \text{Sin}[\mathbf{x}] d\mathbf{x}$$

数式 $\int_0^{2\pi} \sin(x) dx$ の入力が完了しました。

また、入力パレットの中で「 $\sqrt{\square}$ 」や「 $\int_{\square}^{\square} \blacksquare d\square$ 」のように■を含むボタンは下記のような入力方法が可能です。

$$\text{Sin}[\mathbf{x}]$$

先に、■に入力する予定の数式「Sin[x]」を入力しておきます。

$$\text{Sin}[\mathbf{x}]$$

入力した数式を選択し、入力パレットからボタン「 $\int_{\square}^{\square} \blacksquare d\square$ 」をクリックします。

$$\int_{\blacksquare}^{\square} \text{Sin}[\mathbf{x}] d\square$$

後は、前例のように「[Tab]」キーを使って□に数式を入力して行きます。

■ InputForm から StandardForm へ形式変換

Mathematica に慣れてくると、入力パレットを使って入力するより、キーボードからインプットフォームで入力するほうが効率よく入力できます。従って、先にインプットフォームで入力しておいて、後でまとめてスタンダードフォームに変換する方法もあります。変換するには、メニューバーから「セル」→「形式変換」→「StandardForm」(ショートカットキー「[Shift]+[Ctrl]+n」)を選択します。

変換前
<code>Sum[x^2, {x, 1, Infinity}]</code>
変換後
$\sum_{x=1}^{\infty} x^2$

変換前
<code>(-b + Sqrt[b^2 - 4*a*c])/(2*a)</code>
変換後
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

逆に、StandardForm から InputForm に変換することもできます。変換するには、メニューバーから「セル」→「形式変換」→「InputForm」(ショートカットキー「[Shift]+[Ctrl]+i」)を選択します。

変換前
$\sum_{x=1}^{\infty} x^2$
変換後
<code>Sum[x^2, {x, 1, Infinity}]</code>

変換前
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$
変換後
<code>(-b + Sqrt[b^2 - 4*a*c])/(2*a)</code>

■ ショートカットキー (エイリアス) による入力

ショートカットキー：

パレット	ショートカットキー	例	例の入力方法
\blacksquare^{\square}	[Ctrl]+^	x^y	x [Ctrl]+^ y
$\frac{\blacksquare}{\square}$	[Ctrl]+/	$\frac{x}{y}$	x [Ctrl]+/ y
$\sqrt{\blacksquare}$	[Ctrl]+@	\sqrt{x}	[Ctrl]+@ x
$\sqrt[\square]{\blacksquare}$	[Ctrl]+@ と [Ctrl]+%	$\sqrt[\square]{x}$	[Ctrl]+@ x [Ctrl]+% y

エイリアス：

一般数学記号

パレット	エイリアス	完全形
×	[Esc]*[Esc]	\[Times]
÷	[Esc]div[Esc]	\[Divide]
π	[Esc]p[Esc]	\[Pi]
e	[Esc]ee[Esc]	\[ExponentialE]
i	[Esc]ii[Esc]	\[ImaginaryI]
∞	[Esc]inf[Esc]	\[Infinity]
°	[Esc]deg[Esc]	\[Degree]

ギリシャ文字

パレット	エイリアス	完全形
Γ	[Esc]G[Esc]	\[CapitalGamma]
Δ	[Esc]D[Esc]	\[CapitalDelta]
Θ	[Esc]Q[Esc]	\[CapitalTheta]
Λ	[Esc]L[Esc]	\[CapitalLambda]
Π	[Esc]P[Esc]	\[CapitalPi]
Σ	[Esc]S[Esc]	\[CapitalSigma]
Υ	[Esc]U[Esc]	\[CapitalUpsilon]
Φ	[Esc]F[Esc]	\[CapitalPhi]
Ψ	[Esc]Y[Esc]	\[CapitalPsi]
Ω	[Esc]O[Esc]	\[CapitalOmega]

パレット	エイリアス	完全形
α	[Esc]a[Esc]	\[Alpha]
β	[Esc]b[Esc]	\[Beta]
γ	[Esc]g[Esc]	\[Gamma]
δ	[Esc]d[Esc]	\[Delta]
ϵ	[Esc]e[Esc]	\[Epsilon]
ζ	[Esc]z[Esc]	\[Zeta]
η	[Esc]h[Esc]	\[Eta]
θ	[Esc]q[Esc]	\[Theta]
κ	[Esc]k[Esc]	\[Kappa]
λ	[Esc]l[Esc]	\[Lambda]
μ	[Esc]m[Esc]	\[Mu]
ν	[Esc]n[Esc]	\[Nu]
ξ	[Esc]x[Esc]	\[Xi]
π	[Esc]p[Esc]	\[Pi]
ρ	[Esc]r[Esc]	\[Rho]
σ	[Esc]s[Esc]	\[Sigma]
τ	[Esc]t[Esc]	\[Tau]
ϕ	[Esc]f[Esc]	\[Phi]
φ	[Esc]j[Esc]	\[CurlyPhi]
χ	[Esc]c[Esc]	\[Chi]
ψ	[Esc]y[Esc]	\[Psi]
ω	[Esc]o[Esc]	\[Omega]

* π は組み込み定数として定義されているため、変数として利用することができません。それ以外のギリシャ文字は、変数として利用することができます。

付録B アニメーションの作り方

皆さんは、かわいらしいマスコットが動いているホームページを見たことがありますか？自分でホームページを作っている人の中には『私のホームページにもこんなあったらなあ！』とっている人も多いことと思います。今回は、Mathematica で作成したアニメーション(グラフィック)をホームページ素材として利用できるようにする方法を紹介합니다。ホームページ素材として利用できる標準的な画像形式には JPEG¹と GIF²の2種類があって、特に、GIF は透過色を指定したり、パラパラマンガの要領でアニメーションにすることができる仕様となっています。

作成手順としては、まず、Mathematica でパラパラマンガの各コマとなるグラフィックを描き、さらに各グラフィックを GIF ファイルに変換します。この時点で、個々の GIF ファイルをホームページ素材として利用することができます。次に、古溝 剛(ふるみぞ つよし)氏が作成した Giam というフリーウェアを使ってアニメーション GIF(アニメーションになった GIF ファイル)を作成します。星状化 20 面体(本テキストに付属の CD-ROM に収録してあるノートブックファイル「appendix_b.nb」)のアニメーションを例に話を進めて行きます。

■ Mathematica を使って GIF ファイルを作成

まず、Mathematica でノートブックファイル「appendix_b.nb」を開き、各数式を評価してください。パッケージ「Polyhedra」を読み込ませるのを忘れないでください(一度で良い)。図 B.1 のようにアニメーションの各コマとなるグラフィック画像が作成されます。もし、画像サイズを大きく(小さく)したいのであれば、図 B.2 のように最初のグラフィックセルの画像を大きく(小さく)してから数式を再評価してください。

次に、ノートブックの各グラフィックセルを GIF ファイルに変換するために、図 B.3 のようにグラフィックセル以外の全てのセルを削除し、メニューバーから「ファイル」→「特別な形式で保存」→「HTML」を選択³します。保存場所を指定するためのウインドウ(図 B.4)が現れますので、今回はデスクトップに「anime」というディレクトリ名で保存しておきます。ディレクトリ「anime」を開き、さらに配下にあるディレクトリ「HTMLFiles」を開くと、図 B.5 のようにノートブックの各グラフィックセルが GIF ファイルに変換されていることを確認することができます。

¹JPEG(Joint Photograph Experts Group)は24ビット(1670万色)まで扱うことができ、多くの色数を必要とする写真などの表現に向いています。このフォーマットのファイル圧縮の仕組みは、「明るさの変化に比べ、色調の変化には比較的鈍感」という人間の目の性質を利用して、色調変化の部分のデータを捨てることでファイル容量を小さくしています(不可逆圧縮)。

²GIF(Graphics Interchange Format)は最大8ビット(256色)まで扱うことができ、色数をあまり必要としないロゴ・リンクボタン・アイコン・アニメ調のイラストなど、特に単色ベタ面を多く含む平坦な画像に向いています。このフォーマットのファイル圧縮の仕組みは、たとえば「0101010101」を「01×5」などのように圧縮するLZW(Lempel Ziv Wilch)圧縮法が用いられています(可逆圧縮)。ただし、この圧縮方法には米国 Unisys 社の特許技術が含まれていたため、GIFの利用にあたって特許料の問題を抱えていました。しかしながら、米国で2003年6月20日、日本でも2004年6月20日に特許の有効期限が切れたため、GIFを自由に利用できるようになりました。次に紹介するGiamもシェアウェアからフリーウェアとして復活しています。

³今回はグラフィックセル以外には必要なかったので削除しましたが、この方法は、テキストセルを含む全てのセルをセルごとにGIFファイルに変換してくれるので、作成したノートブックそのものをホームページで公開したい場合などにも使えます。

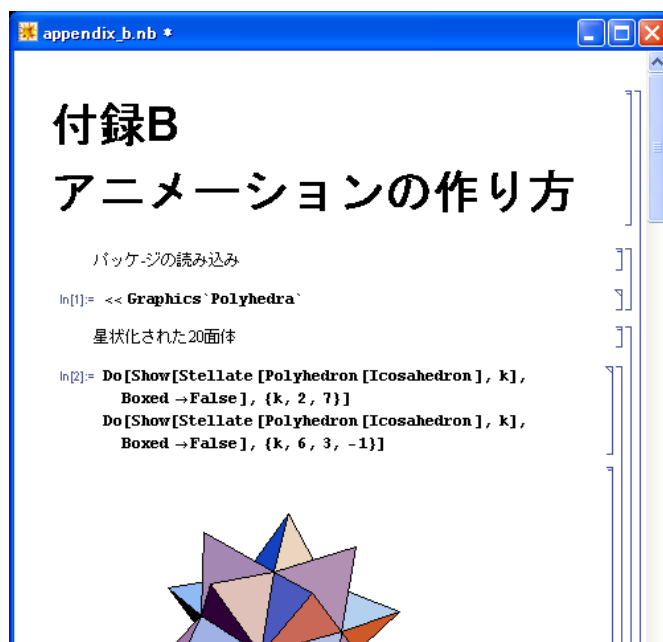


図 B.1: コマの作成

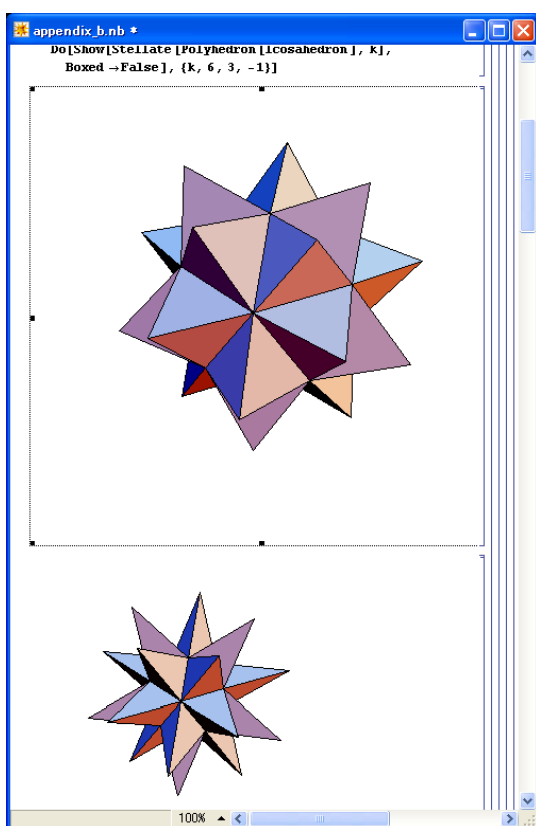


図 B.2: 画像サイズの変更

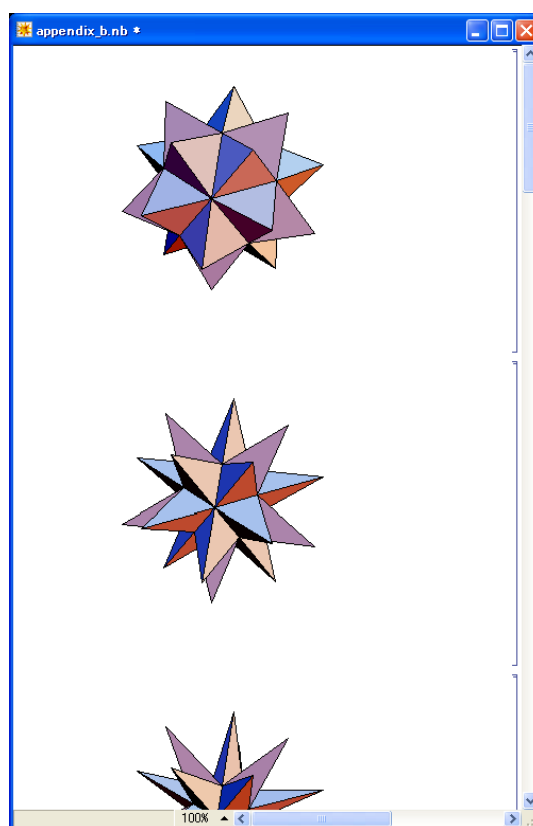


図 B.3: 不要部分の削除



図 B.4: 「HTML」で保存

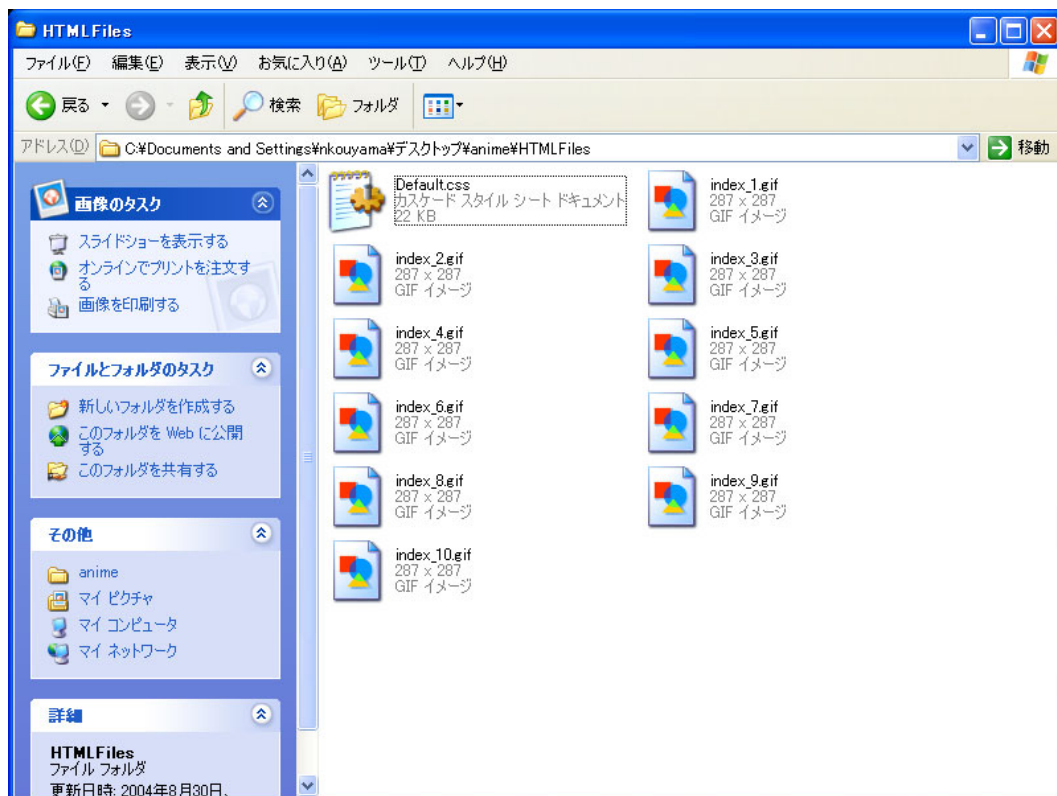


図 B.5: 変換された GIF ファイル

- * 作成された GIF ファイルは、そのままホームページ素材として利用することができます。
- * ファイル「Default.css」は必要ありません。削除してください。

■ Giam を使ってアニメーション GIF を作成

アニメーション GIF を作成するにあたって、古溝 剛氏の Giam というアニメーション GIF を作成するフリーソフトを使用します。Giam のインストールからはじめましょう。

まず、古溝 剛氏のホームページ (<http://homepage3.nifty.com/furumizo/>) から圧縮ファイル「giam201.lzh」(図 B.6) をダウンロードし、適当な場所に保存します。今回はデスクトップに保存しました。ダウンロードした圧縮ファイルをデスクトップに展開します。デスクトップにディレクトリ「giam201」が現れるので、ディレクトリを開いて中にある Giam のアイコン(図 B.7) をダブルクリックします。Giam が起動し、図 B.8 のようなウインドウが開きます。



図 B.6: 圧縮ファイル「giam201.lzh」



図 B.7: Giam のアイコン

それでは、Giam を使ってアニメーション GIF を作成して行きましょう。Giam を起動したら、まずはパラパラマンガのコマとなる GIF ファイル達を図 B.3 の①の部分にドラッグ&ドロップし、コマを登録します。コマが登録されると図 B.8 の②の部分にコマの一覧が表示されます(図 B.9)。

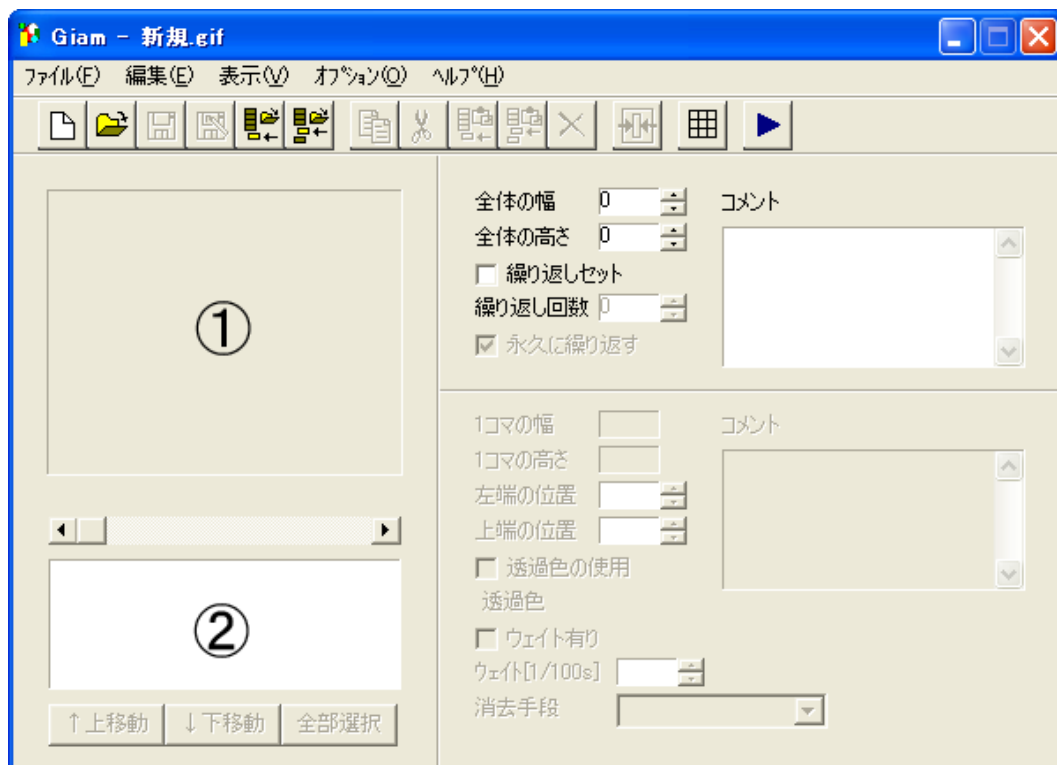


図 B.8: Giam の初期ウインドウ

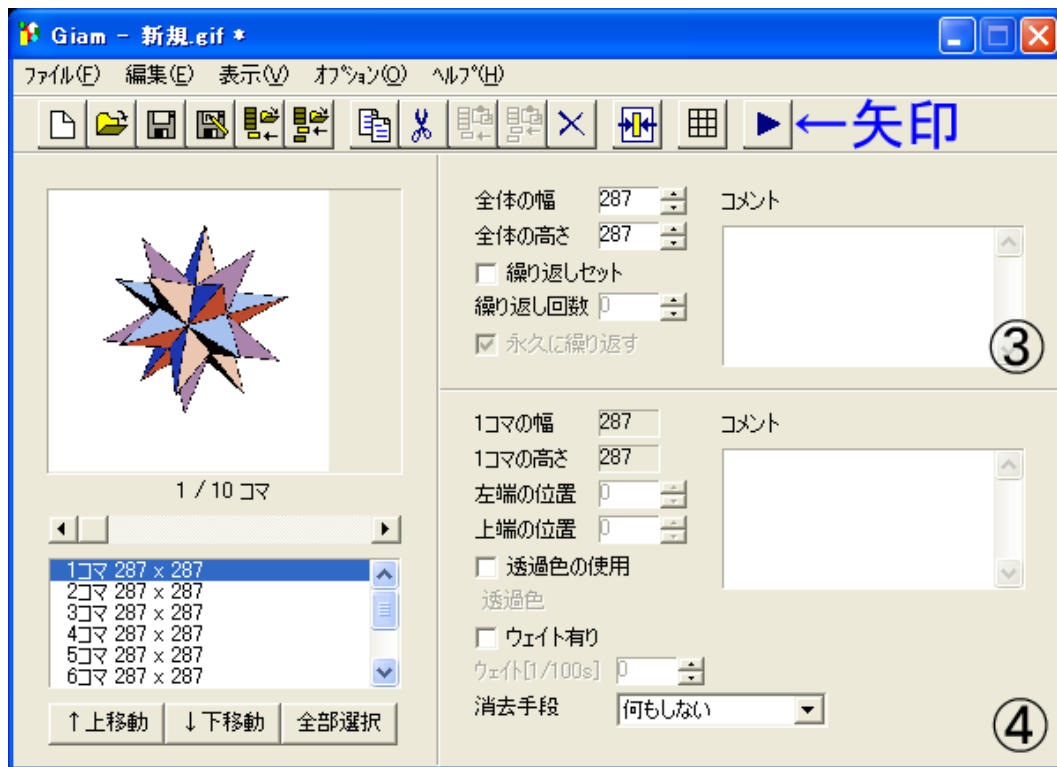


図 B.9: GIF ファイルの登録

次に、②の上にあるスクロールバーを動かして正しい順番に登録されているかチェックし、間違っている場合は、「↑上移動」と「↓下移動」のボタンを使って修正してください。正しいことが確認できたら右上の再生ボタン「▶」をクリックしてみてください。図 B.10 のような「View」ウインドウが現れ、アニメーション GIF の出来栄を確認することができます。「背景色」ボタンを使うと背景の色が換えられ、特に、透過色を指定した場合に透けて見える背景の色とのバランスを確認することができます。もし、アニメーションの出来が気に入らない場合は、図 B.9 の③と④の領域にあるアニメーションのオプションを設定しなおしてください。次のページにオプションの詳細な説明を載せておくので、参考にしてください。

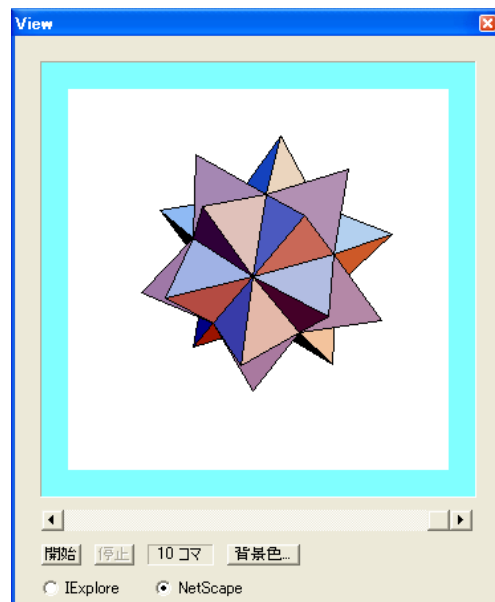


図 B.10: 確認のための「View」ウインドウ

気に入った作品ができたなら、メニューから「上書き保存 (または 名前を付けて保存)」を選択し、保存すれば完成です。保存する前に「GIF の書き込みオプション」ウインドウ (図 B.11) が現れますが、今回はそのまま「OK」ボタンをクリックしてください。

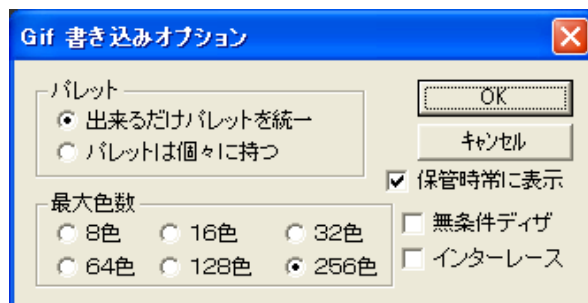


図 B.11: GIF の書き込みオプション

最後に、図 B.9 の右側にあるアニメーションのオプションについて簡単な説明をつけておきます。読者のみなさんもいろいろ試してみてください。なお、筆者が作成したアニメーション GIF を付属の CD-ROM の「アニメーション作品例」に収録しておいたので、そちらも参考にしてください。

■ ③領域 (全体に関わるオプション) :

全体の幅・全体の高さ アニメーションを表示するときの背景の大きさを設定します。

繰り返しセット アニメーションを繰り返し表示する場合に設定します。「繰り返し回数」または「永久に繰り返す」を設定することができます。チェックが外れている場合は 1 順のアニメーションとなります。

コメント 全体に関するコメントを設定します。

■ ④領域 (個々のコマに関わるオプション) :

基本的には②から 1 コマ選択し、そのコマに対してオプションを設定します。

1 コマの幅・1 コマの高さ 各コマの大きさを表示しています。大きさを変更することはできません。

左端の位置・上端の位置 全体の幅・全体の高さを基準に、各コマの表示位置を設定します。

透過色の設定 各コマの透過色を設定します。この項目のみコマの複数選択ができません。1 コマずつ設定してください。

ウェイト有り アニメーションの進行速度を設定します。1/100 秒から設定できます。

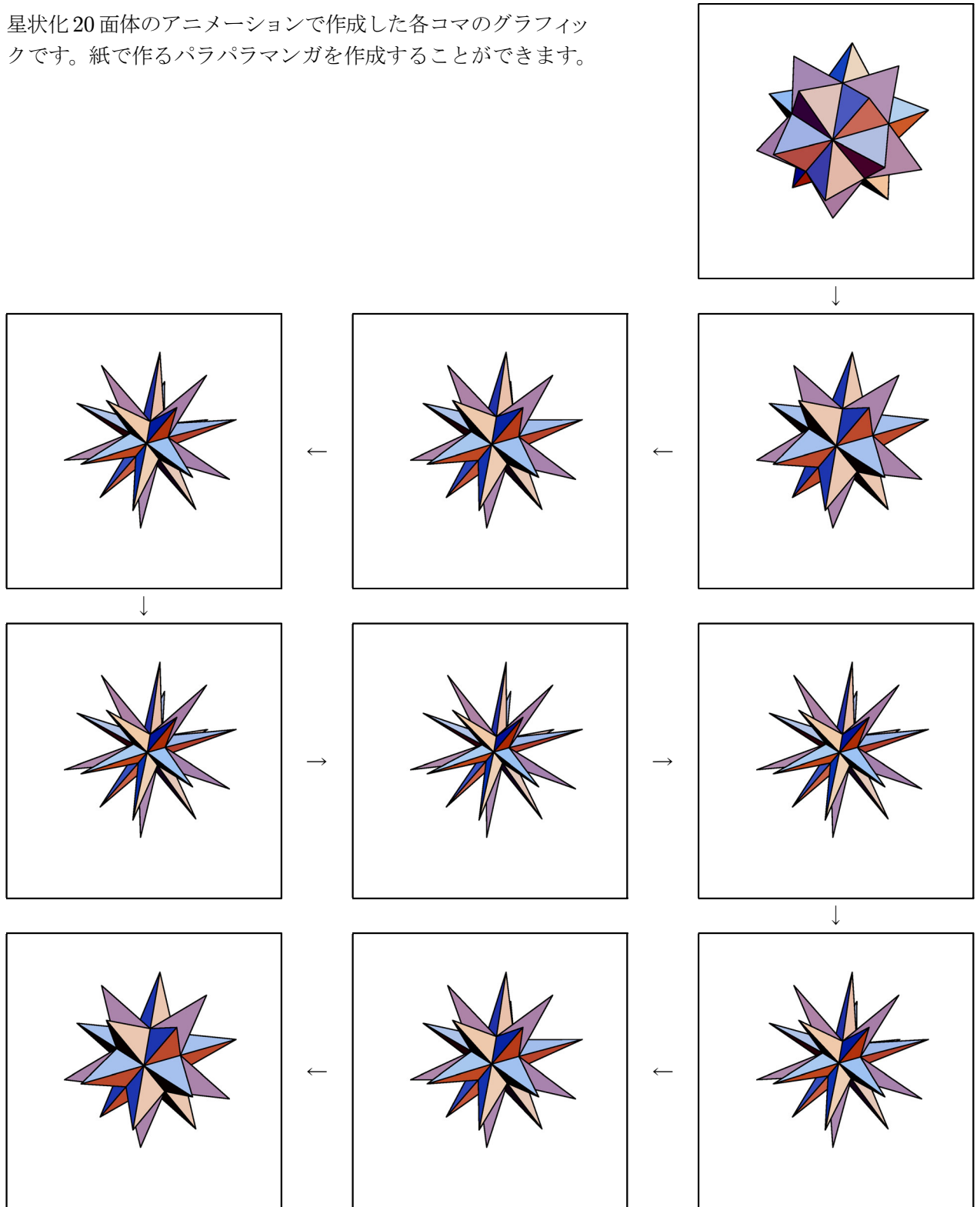
消去手段 次のコマを表示する際に前のコマをどのように削除するかを設定します。「何もしない」・「そのまま残す」・「背景色でつぶす」・「直前の画像に戻す」が設定できます。透過色を設定した場合は「背景色でつぶす」に設定しておかないと、前のコマが見えてしまいます。

コメント 各コマに関するコメントを設定します。

* 詳しくは、Giam のメニューの「ヘルプ」を参照してください。

■ 紙で作るアニメーション (パラパラマンガ)

星状化20面体のアニメーションで作成した各コマのグラフィックです。紙で作るパラパラマンガを作成することができます。



参考文献

第1章の基礎編と付録Aについては、主に

- スティーブン・ウルフラム 著, *Mathematica A System for Doing Mathematics by Computer Second Edition*(日本語版), アジソン・ウェスレイ・パブリッシャーズ・ジャパン 出版
- Mathematica に付属のヘルプ

を参考にしました。

第2章のグラフィック編については、主に

- 小林 道正 著, *Mathematica による関数グラフィックス*, 森北出版株式会社 出版
- Mathematica に付属のヘルプ

を参考にしました。

付録Bについては、主に

- Giam に付属のヘルプ
- Mathematica に付属のヘルプ

を参考にしました。

本テキストに関するご意見・ご質問・ご指摘等は、返信用の電子メールアドレスを明記の上、下記の著者電子メールアドレスへお願いします。なお、本テキストの内容を逸脱と思われるお問い合わせや、筆者のスキルの及ばない範囲のお問い合わせについては、お答えいたしかねますのでご了承ください。

「数学ワンダーランド 2004」

～ Mathematica を使って図形を描こう ～

2004年9月10日 第1版

著者：幸山直人(こうやま なおと)

電子メールアドレス：nkouyama@sci.toyama-u.ac.jp

ホームページアドレス：<http://kouyama.math.toyama-u.ac.jp/main/>

発行：富山大学 理学部 数学教室

ホームページアドレス：<http://www.sci.toyama-u.ac.jp/math/indexJ.html>

© 2004 幸山直人