

数式処理ソフト「MapleV」の紹介

工学系研究科生体機能システム制御工学専攻
中國 真教*

1 はじめに

MapleV (メイプル ファイブ) はコンピュータ代数の最先端の研究を行なっているカナダの Waterloo 大学の研究成果を基に Waterloo Maple Software 社が開発した会話型ソフトウェアの数式処理システムで、数式を解析的に扱うことや数値計算の出来る高度なソフトウェアです。MapleV には約 3000 種類の数学関数が用意されており、数式処理、数値計算、グラフィックスなどの数学処理において強力な機能を提供します。本稿では前半で MapleV の概要を説明します。後半では、まず基本的な MapleV の文法から説明し、その後、簡単な計算例をいくつか挙げて MapleV の操作法を説明します。計算例は皆さんにも手軽に再現できるように 2, 3 行程度のコマンドを入力するだけで結果を表示できるような計算例のみ挙げています。

2 学内での MapleV の利用 (サイトライセンスについて)

MapleV (Release 5) は情報処理センターの研究用および教育用システムで利用可能ですが、佐賀大学では MapleV のサイトライセンスの取得により、佐賀大学内に設置されているコンピュータで、教職員が管理・研究および授業諸活動を支援する目的で使用しているコンピュータならば、情報処理センターが所有する MapleV をそのコンピュータへインストールすることが許可されています¹。また、佐賀大学の教職員及び学生は定価よりも安い価格で MapleV を購入することができます²。

3 MapleV (Release 5) が動作する OS

MapleV が動作するプラットフォームと OS は以下の通りです³。

プラットフォーム	OS
Intel	Windows 95, NT, Linux
Macintosh	Mac OS
Sun SPARC	Solaris
HP9000	HP-UX
DEC Alpha	OSF/1, Digital Unix
IBM RS/6000	AIX
SGI R3000	IRIX

* masanori@ms.saga-u.ac.jp

¹ 情報処理センターではインストールの手引と一緒に MapleV の貸し出しを行なっています。

² 価格は生協にお問い合わせ下さい。

³ 詳細は <http://www.cybernet.co.jp/products/maple/> を御覧下さい。

4 MapleV の特徴

MapleV には次のような特徴があります。

豊富な数学関数ライブラリ

因数分解，方程式や不等式，微分，積分，微分方程式などの一般的な数学関数を始めとし，数値計算はもちろん，各関数パッケージにより，専門的な数学の問題にも対応します。

柔軟性の高いプログラミング機能

MapleV のプログラミング言語⁴ を用いて，ユーザ独自に関数を作成することが可能です。また，その言語は完全な手続き型のプログラミング言語で標準的な言語と類似しています。また，Maple の言語で書かれたプログラムをデバッグするためのツールやファイルの入出力関数も用意されています。

高度なグラフィックス機能

MapleV での計算結果は，2 次元，3 次元のカラーグラフィックスによって視覚化することが可能です。3 次元グラフィックスについては視点の変更や，隠線処理，軸の設定などが簡単に行えます。アニメーション機能では，時間の関数としてモデルがどのように変化するかを表示し，そのプロセスと現象をシミュレーションすることが可能です。また，それらのシミュレーションの再生方向の変更，再生スピードの制御も可能です。

コメントの書き込みやハイパーリンク機能

数式と数式の間ユーザが自由にコメントを書き込むことが可能です。そしてそのコメント文は文字の大きさ，書体などを自由自在に変更できるのでワープロとしての機能も果たします。さらにハイパーリンクを付加することができるので，そのリンクをクリックすると同一のワークシート内の指定された位置や他のワークシート，あるいはヘルプのページへ瞬時に移動できます。

他のアプリケーションとの連携

MapleV は MATLAB とのリンクが可能であったり，Maple 言語を C 言語や FORTRAN のコードに変換することが可能です。また，ワークシートの内容を他のファイル形式へ変換することも可能で，Release 4 では Postscript や L^AT_EX への変換がサポートされていましたが，Release 5 では HTML 形式への変換機能が新たに追加されました。HTML 形式への変換のサポートにより，数式やグラフィックスは GIF ファイルへ，アニメーションは GIF アニメーションへ変換されるので，MapleV のワークシート上で実行した結果をそのまま WEB ブラウザで再現することが可能です。

5 MapleV のユーザインターフェース

ここでは，MapleV のユーザインターフェースの説明をします。Windows 版，Macintosh 版，UNIX 版でそれぞれのインターフェースはほぼ同じ⁵ なので例として Windows 版を使って紹介します。

MapleV は起動後に図 1 のようなウィンドウが現われます。MapleV の起動は Windows 版，Macintosh 版では MapleV のアイコンをクリックして起動します。UNIX 版ではシェルのコマンドラインから `xmaple` と入力してください。

⁴ これを Maple 言語と呼ぶことにします。

⁵ Windows 版と UNIX 版では一部のメニューが異なります。

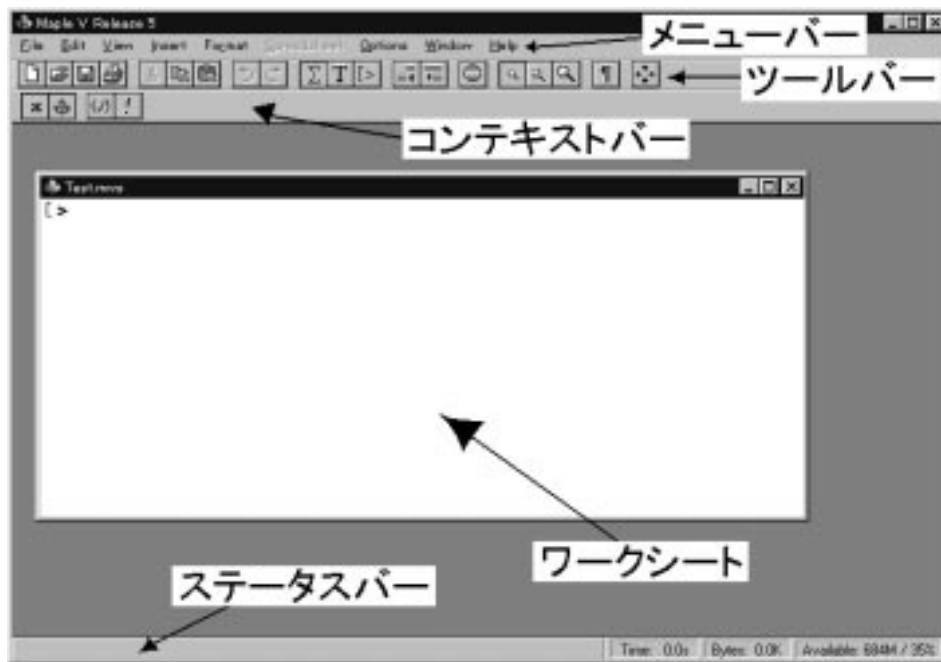



図1 MapleV を操作するためのウィンドウ


図1において、上から、メニューバー、ツールバー、コンテキストバー、ワークシート、ステータスバーと並んでいます。メニューバーには Windows 上で動作する一般的なアプリケーションと同様に File, Edit, ..., Help などのメニューが並んでいます。Macintosh 版の場合は、Macintosh 上で動作する一般的なアプリケーションと同様に、これらのメニューはディスプレイの一番上に配置されます。

ツールバーには、一般的に使用頻度が高い機能をボタンにして並べています。また、コンテキストバーには、図1では表示されていませんが、ボタンの右側に横に長い窓が現われます⁶。その横に長いウィンドウが現われているときに、コンテキストバーの下にあるワークシートに文字を書き込むと、横に長いウィンドウの中にもワークシートで入力した文字が表示されます。

そして、ワークシートには数式やコマンドを MapleV と対話をしながら入力します。ここでは、当然 Maple 言語を用いたコマンドを入力するのですが、Maple 言語を使って数式の入力をしたくないという方のために、簡単な操作で数式の入力が可能な数式エディタの GUI⁷ が用意されています。これは、メニューバーから [**V**iew] [**P**alettes] [**E**xpression Palettes] を選択していくと図2のような積分 (\int)、総和 (\sum)、極限 (\lim)、対数 (\log) などの記号のボタンが集まった Expression というパネルが表示され、入力したい記号のボタンをマウスでクリックすると、数式の雛型がワークシート上に表示され、例えば積分の計算をするために積分のボタンをマウスでクリックするとワークシート中には

$$\int_{?}^{?} ? d?$$

と表示され、MapleV は ? 部分へのユーザの入力を待っています。全ての ? 部分に数字や文字を入力し、その後 Enter キー (Return キー) を押すか、コンテキストバーの  ボタンをマウスでクリックすると、その積分が計算され解答が次の行に出力されます。また、? 部分に分数などを入力したい場合は、さらに Expression パネルから、入力したい記号の雛型のボタンをクリックすると、その ? 部分が分数などの指定した数式の型に変化します。このようにして複雑な式を容易に作る事が可能です。

⁶ コンテキストバーの左端の  ボタンを押すと現われます。

⁷ Graphical User Interface の略。



図2 Expression パネル

ここで紹介したインターフェースは GUI ですが，Windows 版や UNIX 版ではキャラクタベースのインターフェースも利用可能です．シェルのコマンドラインから Windows (MS-DOS) 版では `cmaple.exe` ，UNIX 版では `maple` と入力すればキャラクタベースの MapleV が起動します⁸ ．キャラクタベースの MapleV を終了させる場合は，いずれの OS の場合も `quit` と入力してください．勿論，キャラクタベースで MapleV を動かした方がコンピュータにかかる負荷は小さくなりますので，グラフィックスを表示する必要がない場合はキャラクタベースの MapleV の利用をおすすめします．

6 Maple 言語の基礎

ここでは，次節で紹介する Maple 言語での計算例を説明する前に Maple 言語のコマンドや文法，操作などの基本的な説明を行います．

6.1 この解説中での表記について

- ユーザとは MapleV を操作する人を指します
- `[>` は MapleV のワークシート中に現れるプロンプトを示します
- 下線が引いてある文字列はユーザが入力する文字列です

6.2 Maple 言語のコマンド

Maple 言語にはコマンドというものが存在します．本稿の解説で使われるコマンドを以下に示します．

コマンド	機能
<code>evalf</code>	小数点表示を用い正確な有理数を強制的に求める
<code>factor</code>	因数分解をする
<code>expand</code>	結果の検証を行なう (<code>factor</code> の逆の操作)
<code>solve</code>	方程式を解く
<code>plot</code>	式のプロットまたはグラフ表示を行う

⁸ 起動しない場合は，実行ファイルへのパスが通っているかどうか確認して下さい．

6.3 コマンドや数式などの入力時の注意

Maple 言語ではコマンドや数式を書き、その計算を始める場合は、数式の最後に ; (セミコロン) を付けて Enter キー (Return キー) を押します。するとコンピュータが計算を開始し、計算が終了すればその結果をワークシートに出力します。計算の途中で、入力ミスなどにより計算が間違っている行があればその行を修正し、カーソルを他の行へ移動する前に、必ず Enter キーを押してください。これはコンピュータへ情報の更新を知らせる操作です。単に文字を書き換えただけで Enter キーを押さなければ、コンピュータは文字を修正したことに気付いてくれません。複数行にわたって計算する場合などは特に気を付けて下さい。間違った行を修正した後、その修正結果が影響すると思われる全ての行で Enter キーを押して、上から順々に情報更新を行わなければ最終的に正しい結果が得られません。

また、MapleV はワークシートに入力されたアルファベットの大文字、小文字を区別しますので注意して下さい。

6.4 基本操作と文法 (まずは簡単な計算から)

まず、MapleV を起動して、ワークシートの中に [> (プロンプト)] が表示されているとき、そのプロンプトの右側に式を入力します。'1 + 1' という簡単な計算から始めてみましょう。コマンドラインには次のように下線部の文字列のみ入力してください。そして、通常は入力した式の最後に ; (セミコロン) を必ず付けてください⁹。'1 + 1;' と入力すると、ユーザが入力した次の行には、その計算の結果が出力されます。

————— 1+1 を計算する入力例とその出力結果 —————

```
[ > 1+1;
                                     2
```

MapleV では、このようにして四則演算を行ないます。掛け算の演算子は * (アスタリスク)、割り算の演算子は / (スラッシュ) で表します。また、累乗の演算子は ^ (ハット) で表します。そして、計算の順序を決める括弧 () も使うことができます。これらの演算子や括弧を用いてもっと複雑な計算をしてみましょう。それでは、

$$2^{10} + 3\{9 - 5(33 - 8/7)\}$$

を計算してみます。入力例とその出力結果は次のとおりです。

————— $2^{10} + 3\{9 - 5(33 - 8/7)\}$ を計算する入力例とその出力結果 —————

```
[ > 2^10+3*(9-5*(33-8/7));
                                     4012
                                     /
                                     7
```

ここでは計算結果が分数で出力されてしまいました。なぜ計算結果を小数点を用いて出力しないのでしょうか？ これは分数を用いることにより正確な答えを表示するためです。小数を用いて出力するためには evalf コマンドを使います。入力例とその出力結果は次のとおりです。

————— $2^{10} + 3\{9 - 5(33 - 8/7)\}$ の計算結果を小数点表示する入力例とその出力結果 —————

```
[ > evalf( 2^10+3*(9-5*(33-8/7)) );
                                     573.1428571
```

⁹ ; (セミコロン) の代わりに : (コロン) を付けて、計算結果を出力させないようにすることも可能です。

このように小数点を用いて出力されましたが、この答えは正確ではありません。正確な値に近い値を表示したい、つまり、表示する桁数をもっと増やしたいという場合は、小数点表示の桁 (digits) 数を指定することができます。今度は先程の答えを 30 桁で表示されるように指定しましょう。入力例とその出力結果は次のとおりです。

——— 小数点表示の桁数を指定した入力例とその出力結果 ———

```
[ > evalf( 2^10+3*(9-5*(33-8/7)), 30 );  
573.142857142857142857142857143
```

小数点以下は 142857 が続く循環小数になっているようです¹⁰。そこで、evalf の書式は次のようになります。

——— evalf の書式 ———

```
evalf( 数式, 桁数 )      ( 桁数はカンマと共に省略可 )
```

また、小数点表示の結果を出力させるたびに evalf で桁数を指定するのは面倒だという場合は、Digits という変数に桁数の値を代入することにより、それ以降の計算では evalf で桁数を指定する手間を省くことができます。桁数を設定する場合は次のように入力してください。

——— Digits を使って桁数を 30 桁にする設定 ———

```
[ > Digits := 30;
```

つまり、Digits は次のように設定します。

——— Digits を使った桁数の設定 ———

```
Digits := 桁数
```

ここで := という記号が出てきましたが、これは右辺の式を左辺に代入するという意味です。この記号を用いて、式に名前を付けることができます。例えば、 $x^2 + 2x + 3$ という式に y という名前を付ける場合は

——— $x^2 + 2x + 3$ に y という名前を付ける操作 ———

```
[ > y:=x^2+2*x+3;
```

とします。このようにしておくと、 $x^2 + 2x + 3$ が必要なときに y という名前で式を呼び出すことができます。これは、入力が面倒な長い式を何度も繰り返して使用する必要がある場合に非常に便利です。

6.5 長い式の入力

計算によっては複雑な文法で、しかも一つの式がとて長くなることもあるでしょう。このような場合は式の途中から適当な場所で自由に折り返すことができます。例えば、 $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$ を入力してみましょう。この式を +6 から折り返すことにします。その場合は、 $1 + 2 + 3 + 4 + 5$ まで入力

¹⁰ 出力結果の一番右端の桁は 2 になる筈ですが、さらにその桁の右側の桁の数字は 8 になるので四捨五入して 3 になっています。

した後に、単に Enter キーを押してしまうと文法の間違いのためにエラーになってしまいますので、シフトキーを押しながら Enter キーを押します。すると、エラーを起こさずに式を折り返すことが出来ます¹¹。長い式を折り返した様子を以下に示します。

長い式を折り返した様子

```
[ > 1+2+3+4+5
    +6+7+8+9+10;
```

6.6 直前に出力した結果の再利用

計算の過程で出力した結果を利用し、次の計算で利用したい場合があります。その場合は、その結果は記号 := を使ってその結果に対して名前を付けておけば良いのですが、その他にも方法があります。これは、計算の過程で直前に出力された結果を呼び出すときに有効な方法です。直前に出力された結果は % という名前¹² が付けられています。例えば、「1+2+3」を計算した場合は % は次のようになります。

% に入っている値の例

```
[ > 1+2+3;
                                     6
[ > %;
                                     6
[ > %+4;
                                     10
[ > %;
                                     10
```

このように % には直前で出力された結果が含まれています。

7 計算例

Maple 言語の基本を説明したところで、簡単な計算例を挙げて説明します。ここでは、以下のような計算の説明をします。

- 因数分解と式の展開
- 微積分（極限、微分）
- 行列の計算
- 2次元、3次元グラフィックスとそのアニメーション

以上のような例を用いて、MapleV の操作方法と数式の入力の説明を行ないます。

¹¹ 行列を入力する場合は、行と列を揃えて入力することが出来るので非常に見やすくなります。

¹² MapleV Release4 では % ではなく ' ' (ダブルクォーテーション)を使います。

7.1 因数分解

因数分解では factor コマンドを使います。ここでは、

$$x^6 - x^5 - 9x^4 + x^3 + 20x^2 + 12x$$

を因数分解してみましょう。コマンドの使い方は次のとおりです。

factor コマンドの使い方

factor (多項式)

入力例とその出力結果は次のとおりです。

多項式を因数分解する操作の入力例とその出力結果

```
> factor( x^6 - x^5 - 9*x^4 + x^3 + 20*x^2 + 12*x );  
x(x-2)(x-3)(x+2)(x+1)^2
```

7.2 式の展開

式の展開には expand コマンドを使います。コマンドの使い方は次のとおりです。

expand コマンドの使い方

expand (展開する式)

expand コマンドは factor の逆の操作を行います。先程の例で因数分解したときに出力した結果である $x(x-2)(x-3)(x+2)(x+1)^2$ を展開してみましょう。入力例とその出力結果は次のとおりです。

式を展開する操作の入力例とその出力結果

```
> expand( x*(x-2)*(x-3)*(x+2)*(x+1)^2 );  
x^6 - x^5 - 9x^4 + x^3 + 20x^2 + 12x
```

7.3 極限

極限を求めるには limit コマンドを使います。コマンドの使い方は次のとおりです。

limit コマンドの使い方

limit (関数, 極限をとる座標の値)

ここでは、 x が 2 に近づくときの $\frac{x^2-4}{x-2}$ の極限を求めます。入力例とその出力結果は次のとおりです。

————— $\frac{x^2-4}{x-2}$ の極限を求める操作の入力例とその出力結果 —————

```
[ > limit( (x^2-4)/(x-2), x=2 );  
4
```

次に, $\tan x$ の極限について考えてみましょう. x が $\pi/2$ に近づいたときの極限は左側から近づいた時と右側から近づいたときの $\tan x$ の極限を考えます. ここでは `limit` コマンドにオプションを付けます.

————— 左右から極限を求めるオプション —————

`left, right`

入力例とその出力結果は次のとおりです.

————— $x = \pi/2$ の左右から $\tan x$ の極限を求める操作の入力例とその出力結果 —————

```
[ > limit( tan(x), x=Pi/2, left );  
  
[ > limit( tan(x), x=Pi/2, right );  
-
```

7.4 関数の微分

関数を微分するのは `diff` コマンドを使います. コマンドの使い方は次のとおりです.

————— `diff` コマンドの使い方 —————

`diff(関数, 微分する変数)`

ここでは $5x^3 + 2x^2 - 3x + 8$ を x について微分します. 入力例とその出力結果は次のとおりです.

————— 関数を微分する操作の入力例とその出力結果 —————

```
[ > diff( 5*x^3 + 2*x^2 - 3*x + 8, x );  
15x^2+4x-3
```

7.5 行列の計算

ここでは次のような行列 A, B についての計算を行ないます. 行列を計算するには `evalm` コマンドを使います.

$$A = \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix}$$

まず, 最初に次のように行列を A, B それぞれに読み込みます. 入力例とその出力結果は次のとおりです.

行列の値を変数に代入する操作の入力例とその出力結果

```
> A := matrix( [ [1,a,b], [0,1,c], [0,0,1] ] );
```

$$A := \begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$$

```
> B := matrix( [ [1,2,3], [2,3,4], [5,6,7] ] );
```

$$B := \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix}$$

そして evalm コマンドを使って行列 A , B の和を計算します。入力例とその出力結果は次のとおりです。

行列の和を計算する操作の入力例とその出力結果

```
> evalm( A + B );
```

$$\begin{bmatrix} 2 & a+2 & b+3 \\ 2 & 4 & c+4 \\ 5 & 6 & 8 \end{bmatrix}$$

行列 A と行列 B の行列どうしの掛け算では、演算子 $\&*$ を使います¹³。入力例とその出力結果は次のとおりです。

行列どうしの積を計算する操作の入力例とその出力結果

```
> evalm( A &* B );
```

$$\begin{bmatrix} 1+2a+5b & 2+3a+6b & 3+4a+7b \\ 2+5c & 3+6c & 4+7c \\ 5 & 6 & 7 \end{bmatrix}$$

7.6 2次元グラフィックス

ここでは2次元のグラフィックスを表示します。2次曲線 $y = 2x^3 - 3x^2 - 12x + 10$ のグラフを表示してみましょう。グラフを描く前にはグラフィックスパッケージである plots パッケージを呼び出します。そして, MapleV で2次元のグラフを描く場合は plot コマンドを使います。plot コマンドでは x の範囲, つまり定義域を設定しなければなりません。そこで x の範囲を適当に $-3 \leq x \leq 3$ とします。コマンドの使い方は次のとおりです。

plot コマンドの使い方

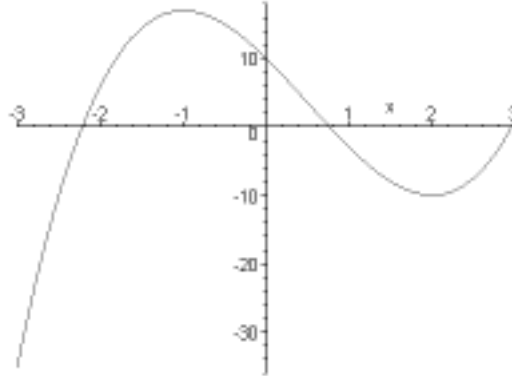
plot (グラフを表す式, x の範囲)

入力例とその出力結果は次のとおりです。

¹³ 行列をスカラー倍するときに使う演算子は * でなければなりません。

2次元グラフを描く操作の入力例とその出力結果

```
> with(plots);  
> plot( 2*x^3-3*x^2-12*x+10, x=-3..3 );
```



ここで出力したグラフはコピー&ペーストで画像ファイルとして保存することが可能です。

次に、螺線を表示してみましょう。螺線を描くコマンドは polarplot です。polarplot コマンドは極座標形式でグラフを描くコマンドです。コマンドの使い方は次のとおりです。

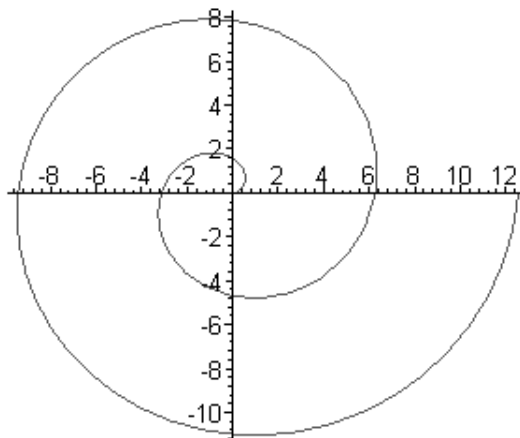
polarplot コマンドの使い方

polarplot (半径, 角度の範囲)

ここで、グラフを描く前には plots パッケージを呼び出す必要があるのですが、そのパッケージは2次元曲線のグラフを出力するときに呼び出しているため、再度呼び出す必要はありません。また、先程の例と同様に定義域である角度の範囲を決めなければなりません。ここでは角度の範囲を適当に $0 \leq \theta \leq 4\pi$ とします。定義域の指定で、 π の値を 3.14 としても良いのですが、Maple 言語では π を Pi として表すことができるので Pi を使うことにします。また、これは螺線なので“半径=角度”として考えます。入力例とその出力結果は次のとおりです。

螺線を描く操作の入力例とその出力結果

```
> polarplot( theta, theta=0..4*Pi );
```



7.7 3次元グラフィックス

ここでは先程示した螺線の3次元バージョンを表示してみましょう。MapleVで3次元のグラフを描く場合は、`cylinderplot` コマンドを使います。`cylinderplot` コマンドは極座標形式で円柱グラフを描くコマンドです。コマンドの使い方は次のとおりです。

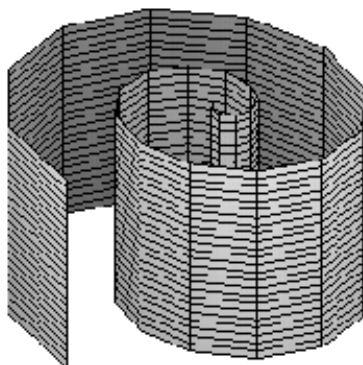
— `cylinderplot` コマンドの使い方 —

`cylinderplot` (半径, 角度の範囲, z の範囲)

そして、角度と z の範囲をそれぞれ適当に $0 \leq \theta \leq 2\pi$, $-1 \leq z \leq 1$ とします。入力例とその出力結果は次のとおりです。

— 螺線 (3次元バージョン) を描く操作の入力例とその出力結果 —

```
> cylinderplot( z, theta=0..2*Pi, z=0..1 );
```



そこで表示されたグラフをクリックしてみてください。コンテキストバーのところに新しいボタンが現われたはずですが、そのコンテキストバーの左端にある、 θ や φ の値を変更してみましょう。すると少しずつ螺線が動いていくのがわかります。コンテキストバーには他にもいくつかボタンがあるので、適当に押してみてください。螺線が様々な表示に変化します。

ところで、この3次元の螺線は2次元の螺線に比べて滑らかではではありません。そこで、もっときめの細かい滑らかな曲面にするためにオプションを付け加えます。

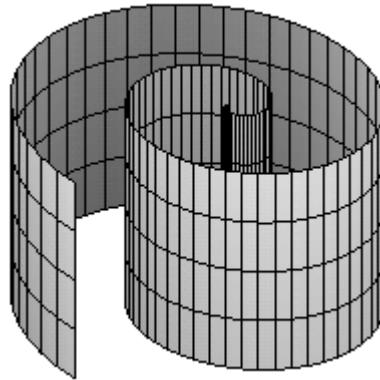
— グラフの曲面を滑らかにするオプション —

`grid`=[角度の解像度, z 軸の解像度]

それぞれの解像度は数字が大きくなると滑らかに、数字が小さくなると粗いグラフになります。 z 軸の解像度に関しては、関数が z 軸の方向には何も変化しないので z 軸の解像度には小さな値を代入することになります。入力例とその出力結果は次のとおりです。

滑らかな螺旋線 (3次元バージョン) を描く操作の入力例とその出力結果

```
> cylinderplot( z, theta=0..2*Pi, z=0..1, grid=[100,5] );
```



7.8 アニメーション

関数群 $f(x) = ax^2$ は各種の放物線を表します . 定数 a の違いによってどう変わるかをアニメーションで表示します . ここでは `animate` コマンドを使います . `animate` コマンドを実行する前には , やはり `plots` パッケージを呼び出す必要があります . コマンドの使い方は次のとおりです .

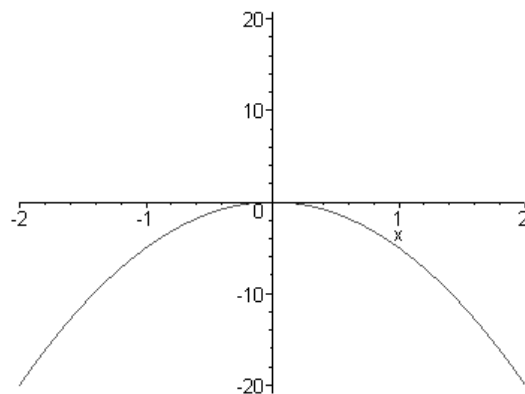
animate コマンドの使い方

`animate (グラフを表す式 , x の範囲 , a の範囲)`

入力例とその出力結果は次のとおりです .

アニメーションのを行う操作の入力例とその出力結果

```
> animate( a*x^2, x=-2..2, a=-5..5 );
```



そこで表示されたグラフをクリックしてみてください . すると , 3次元グラフィックスのときと同様にコンテキストバーのところ新しいボタンが現われました . そのボタンはビデオデッキという「停止」, 「再生」, 「巻戻し」などのボタンと同じです . 再生ボタンを押すことによってアニメーションを開始します . 左

から 2 番目の再生ボタンを押してみましょう。すると、 a の値が変化するにつれて、上に凸だった放物線が下に凸の放物線に変化していきます。

8 他のアプリケーションとの連携

MapleV では Maple 言語で書かれたプログラムを C 言語や FORTRAN のソースコードへ変換したり、ワークシート上で行った計算の結果をそのまま Plain Text 形式や、 \LaTeX 形式、HTML 形式に出力することが可能です。ここでは、文章作成に関連する \LaTeX や HTML ファイルなどへの変換について説明します。

ワークシートに入力した内容をこれらのファイル形式へ変換する場合は、MapleV のメニューバーから [File] [Export As] と辿っていくとそれぞれのファイルに変換するためのメニューが現れます。Plain Text、Maple Text、 \LaTeX 、HTML のそれぞれの形式に変換した場合の数式やグラフは次のようなファイル形式で保存されます。

出力形式	数式	グラフ
Plain Text	Maple 言語形式	なし
Maple Text	Maple 言語形式	なし
\LaTeX	\LaTeX 形式	EPS 形式
HTML	イメージとして GIF ファイルに保存	GIF 形式

Plain Text 形式や Maple Text 形式に変換した場合は他の変換形式と違い、グラフは保存されず Maple 言語形式で書かれた Text ファイルだけが出力されます。また、Plain Text 形式と Maple Text 形式の二つは、ほぼ同じ内容に変換されます¹⁴。そして、 \LaTeX 形式に保存したファイルを DVI 形式のファイルに変換するためには `maple2e.sty` という \LaTeX のスタイルファイルが必要です。これは MapleV がインストールされているフォルダの中の `etc` フォルダに入っています。

9 おわりに

簡単ではありますが、MapleV の紹介をしました。MapleV では便利な機能を広い分野にわたって豊富にとりそろえてあります。これらの便利な機能はさまざまな研究分野で多大な支援が期待できるでしょう。学生の方には、自分の手で計算した問題の解答を MapleV を利用して検算したり、想像が困難な 3 次元のグラフィックスなどを表示させてグラフの概形を確認するような利用法を期待します。

ページ数の制限により、詳しく解説することはできませんでしたが、ここで紹介出来なかった計算例などを筆者のホームページ¹⁵で紹介していますのでそちらの方も御覧下さい。最後に、ここで紹介できなかった MapleV ライブラリ関数（標準ライブラリ以外）の一覧を示しておきます。これらは専門分野別にまとめられています。

¹⁴ あるファイルを Plain Text と Maple Text の二つの形式に変換したところ、出力された 2 つのファイルの内容は全く同じものでした。

¹⁵ URL は <http://www.ms.saga-u.ac.jp/~masanori/maple/> です。

表 1 MapleV で利用可能な専門分野別パッケージ

パッケージ名	内容
Detools	微分方程式の解析ツール
Domains	計算のドメインを作成
GF	ガロア体
Gaussint	ガウス積分
LRtools	線形回帰方程式
combinat	組み合わせ関数
combstruct	組み合わせ構造のランダム生成や操作
diforms	微分形式
finance	金融計算
genfunc	有利生成関数
geometry	ユークリッド幾何学
grobner	グレーブナー基底
group	置換群と有限表現群
inttrans	積分変換
liesymm	Lie 対称
linalg	線形代数
logic	ブール論理学
networks	グラフネットワーク
numapprox	数値近似
numtheory	数論
orthopoly	直交多項式
plottools	基本的なグラフィックオブジェクトの操作
powseries	形式的べき級数
process	(UNIX) マルチプロセス
simplex	線形最適化
stats	統計学
student	学生用計算パッケージ
tensor	テンソルの計算, および一般相対性理論

謝辞

本稿の執筆にあたってアドバイスを頂いた本学理工学部数理科学科小倉幸雄氏, 亀谷幸生氏に深謝致します。

参考文献

- [1] Wade Ellis, Jr., Eugene Johnson, Ed Lodi, Daniel Schwalbe 著, 臼田 昭司, 藤田 寿之, 吉田 丈夫 訳, 実践数式処理シリーズ MAPLE V 数式処理入門 MAPLE V FLIGHT MANUAL, インターナショナル トムソン パブリッシング ジャパン, 1997.
- [2] K.M. Heal, M.L. Hansen, K.M. Rickard 著, 笠嶋友美 訳, はじめての Maple V リリース 4, シュプリンガー・フェアラー東京, 1997.