高専数学の学習と数式処理

八木善彦*

Learning with Computer Algebra Systems in the Technical College Mathematics

Yoshihiko YAGI*

ABSTRACT

In order to promote the use of the computer algebra system in learning mathematics, a survey was conducted on the assumption that student using the computer algebra software“ Mathematica ” to assist them in solving problems from the technical college mathematics textbook will achieve better results.

The results show that Mathematica contributes as a learning aid in solving the technical college mathematics textbook problems for the lower grades with few difficulties. Moreover, the majority of the textbook problems can be solved with the use of basic function from Mathematica.

Keywords: Mathematica, mathematics education, computer algebra system

1. はじめに

近年, コンピュータやインターネットなどのインフラ面, 学校教育用コンテンツの開発やソフト面の整備が急速に進展し, 数学教育の情報化に必要な環境が整いつつある.

このようなテクノロジーを利用した教育や学習を促進するために, 本校では, Mathematica を総合情報センターに導入して学生の利用を図っている. 今回, 数学の学習に数式処理システムが, どのような役割をになっていけるかを調査する一環として, 数式処理ソフト Mathematica で, 高専数学教科書 (1)(2)(3) の問題を解く場面を想定した調査を行った.

2. 高専の数学

高専や高校で習う教科書の複雑な微分や積分の例題を Mathematica は解くことができる. 大学や大学院入試の問題も Mathematica は解くことができる (4). 高専の学生が, 高専の数学の教科書や問題集を解答する場合 Mathematica で何ができ, 何が問題となるであろうか. 1年から3年までの教科書を利用して調査する.

Mathematica を使用するためには基本的なチュートリアルが必要である. しかし Mathematica のチュートリアル を解説 (5) した後は, ソフト自体が解説やヘルプをもっているため, 使用関数のヒントを与えるだけで使えるようになる. 積極的な利用を促す為には日本語版が必須である. 古いバージョンの Mathematica では不等式の解法はできないが, 現在のバージョンでは, 問題なく動作する.

2.1 1年の数学

1年の数学の数学は, 数と式, 2次の関数・方程式・不等式, 命題・等式・関数, 指数関数・対数関数, 三角関数, 平面上の図形, 個数の処理にわかれ, 学習の補助として Mathematica を利用してもなんら問題はなかった.

表 1: 数と式の分野

分野	コマンド	備考
式の計算	$+, -, *, /, Expand$	問題なし
因数分解	$Factor$	問題なし
除法	$PolynomialQuotient$, $PolynomialRemainder$	問題なし
約数, 倍数	$PolynomialLCM$, $PolynomialGCD$	文章題立式必要
数の種類	$IntegerQ$, $Simplify$, $Head$	集合, 複素数の知識
不等式証明	$Reduce, Reals$	ver.4 × ver.5 ○ 必要十分条件なので厳密
√の計算	$Sqrt$	小数は分数に直す必要あり
有理化	$ExpandNumerator$, $ExpandDenominator$	有理化の操作
繁分数	$Simplify$	問題なし

*一般科　教授

2.1.1 数と式

数と式の分野には加減乗除のほか, 展開 (Expand), 因数分解 (Factor), 部分分数分解 (Apart), かわったところでは整式の割り算での商と余りを求める関数, 整式の最大公約数, 最小公倍数を求める関数が使用できる. 数の種類の判定には, 内容に応じた知識と計算方法の理解がないと求める結果は得られない. 複素数の虚数部が厳密なゼロか有限精度のゼロかを区別することは理屈っぽいだけで実用的な意味がないように思われる. しかし, 区別することは重要で複素数を使ったベキと根の解釈でこの違いは大きな意味をもつ [6, p.705]. 分数の有理化では, 計算方法を指示し解答を得ることになる (表１).

2.1.2 ２次の関数・方程式・不等式

２次の関数・方程式・不等式の分野ではグラフを描く (Plot), 解く (Solve) を中心として使用することになる (表 2). 解と係数の関係や判別式の計算では, 公式を理解したうえで代入計算をすることになる. 最大値や最小値を求める問題では, グラフを見ながら考えることが必要になる. 共有点や共有点の個数では, 実数のオプションを使う等, Mathematica の深い部分が必要となるが, 問題解答の補助として使用するのであればそこまでこだわる必要はない.

表 2: ２次の関数・方程式・不等式の分野

分野	コマンド	備考
２次関数のグラフ	$Plot$	問題なし
頂点	$PolynomialQuotient,$ $PolymomialRemainder$	式組み合わせ必要
最大値, 最小値	$Plot$	思考と操作
２次方程式	$Solve$	問題なし
複素数の計算	$+,-,*,/,$ $Expand$	有理化指示要
解と係数の関係	$/.x->a$	代入計算のみ
判別式	$/.x->a$	代入計算のみ
共有点	$Solve$	問題なし
共有点の個数	$Solve, Reals$	問題なし
接点	公式代入	判別式代入計算
不等式	$Solve, Reals$	ver.4 × ver.5 ◯
絶対値不等式	$Solve, Reals$	ver.4 × ver.5 ◯

2.1.3 命題・等式・関数

命題・等式・関数の分野では, 証明問題を解答させる部分では Mathematica には無理がある (表 3). 命題計算などは, Mathematica で可能であるが, わざわざそのための定義と計算規則を遂行するのは, 教科書の問題を解くには大げさである. 証明過程での式変形や結果の確認がおもな利用法となる. 不等式の解を求めるには, Ver.5 以降を使用する.

表 3: 命題・等式・関数の分野

分野	コマンド	備考
集合	$\cap, \cup, \subset, \supset$	無限集合, リスト
命題, 必要条件, 十分条件	証明不可	命題計算可能
背理法, 数学的帰納法	証明不可	式変形可能
恒等式	$Solve$, ほか	式変形
剰余の定理, 因数整理	$/.x->a$	代入計算
高次不等式	$Solve, Reduce$	ver.4 × ver.5 ◯
等式・不等式の証明	$Solve, Reduce$	ver.4 × ver.5 ◯
関数とグラフ	$Plot$	問題なし
偶感数・奇関数	$/.x->-x$	代入計算
分数関数	$Plot$	問題なし
無理関数	$Plot$	問題なし
無理方程式	$Solve$	問題なし
逆関数	$Solve$	問題なし

2.1.4 指数関数・対数関数

指数関数・対数関数の分野では, 計算が中心であり, Mathematica で十分対応できる (表 4).

表 4: 指数関数・対数関数の分野

分野	コマンド	備考
累乗根	$+, -, *, /,$	問題なし
指数関数	$Plot, Solve$	問題なし
対数関数	書き換え	意味を理解
対数不等式	$Solve$	ver.4 × ver.5 ◯
大小比較	−	差を調べる

2.1.5 三角関数

三角関数の分野では, 計算が中心であり, Mathematica で十分対応できる (表 5). 証明は (左辺)−(右辺)= 0 で計算できるが, 式変形の経過は確認できない. °(Degree) は定数であり, 積として弧度法へと変換され, あたかも度の単位で計算したような効果がある.

2.1.6 平面上の図形

平面上の図形の計算では, 基本的に公式に代入する形が多い. 状況に応じて式変形をする. 領域に色をつけたい場合は FilledPlot(標準ライブラリ [7] の読み込みが必要) など特殊な機能が必要になる (表 6). 平方完成の関数は Mathematica の組み込み関数にはない. 式をみながら手作業で変形をする必要がある.

表 5: 三角関数の分野

分野	コマンド	備考
三角関数の値	$Sin, Cos, Tan, Degree$	一般角, 弧度法
等式の証明	$Simplify, Plot, TrigReduce$	問題なし
三角関数の合成	$/.x->a$	代入計算
三角方程式	$Solve$	問題なし
三角不等式	$Solve$	ver.4 × ver.5 ○
三角形の解法	$Factor, Simplify$	代入
ヘロンの公式	$/.x->a$	代入計算

表 6: 平面上の図形の分野

分野	コマンド	備考
内分点外分点	$/.x->a$	代入計算
距離	$/.x->a$	代入計算
直線の方程式	$/.x->a$	代入計算
2直線の関係	$/.x->a$	代入計算
軌跡	$/.x->a$	代入, 式変形
円の方程式	$/.x->a$	平方完成
アポロニウスの円	$/.x->a$	代入, 式変形
2次曲線	$/.x->a$	代入計算
不等式と領域	$Reduce, FilledPlot$	ライブラリ
領域における最大最小	$Plot$	解の参考程度

2.1.7 個数の処理

個数の処理の分野では, 順列の組み込み関数はもたない. 階乗 (!) を使って計算することになる (表 7). 組合わせは Binomial が使用できる. 規則的な書き出しと数え上げは Mathematica の得意な計算であるが効果的な式を作り上げるには Mathematica を深く知っておく必要があり, 学生が答え合わせに使用するには大がかりすぎる. 計算を Mathematica に指示し電卓のかわりに使う.

表 7: 個数の処理の分野

分野	コマンド	備考
場合の数	!	問題なし
二項定理	$!, Binomial$	問題なし

2.2 2年の数学

2年の数学は大きくわけて, 微分積分と線形代数である. 分野は数列, 微分法, 積分法, ベクトルと図形, 行列と行列式からなる.

2.2.1 数列

等差数列, 等比数列では, 公式に代入することにより一般項や和を求める. 初項や公差などの数を求めるには連立方程式を解けばよい (表 8). Mathematica は, 総和 (Sum) や極限 (Limit) も組み込み関数としてもち, 計算には困らない.

表 8: 数列の分野

分野	コマンド	備考
等差数列, 等比数列	$/.x->a$ $Solve$	公式に代入
和, Σ	Sum	問題なし
数学的帰納法	$/.x->a$	代入, 式変形
無限数列, 極限	$Limit$	問題なし

2.2.2 微分法

ほとんどの問題が Mathematica の組み込み関数で処理できる (表 9). 増減表をかく機能はないが, グラフ描画 (Plot) を使用することで代用できる. 不等式の証明においても増減表の代わりはグラフ描画で対応できる.

表 9: 微分法の分野

分野	コマンド	備考
導関数	$Diff$	問題なし
接線, 速度	$/.x->a$	公式に代入
関数の増減	$Plot$	グラフで確認
極値	$Diff, Solve$	問題なし
関数の最大値最小値	$Plot$	問題なし
関数の極限	$Limit$	問題なし
関数の連続性, 中間値の定理	$Limit, /.x->a$	問題なし
対数関数・指数関数・三角関数の導関数	$Diff$	問題なし
導関数の応用	$Diff, Plot$	問題なし
実数解の個数	$Solve, Plot$	問題なし
不等式の証明	$Plot$	問題なし
接線・法線と近似式, 速度加速度	$/.x->a$	公式に代入

2.2.3 積分法

積分に対しては組み込み関数の積分 (Integrate) で対応できる (表 10). 三角関数や対数関数などで結果が複雑な形になるものはさらに式変形が必要となる.

表 10: 積分法の分野

分野	コマンド	備考
不定積分	$Integrate, Simplify$	三角対数関数は式変形必要
定積分	$Integrate$	問題なし
囲まれた面積, 体積, 回転体の体積	$Integrate$	公式代入

2.2.4 ベクトルと図形

ベクトルはそのまま Mathmatica で計算でき代入なども問題ない (表 11). 図形に関しては公式への代入計算やグラフ描画など補助的な役割を果たす. 問題によっては代入計算を実行するまえに立式が必要なものもある.

表 11: ベクトルと図形の分野

分野	コマンド	備考
ベクトル	$/.x->a$	立式必要
直線, 法線, 距離, 円	$/.x->a$	立式必要, 公式代入
空間ベクトル	$+,-,*,/$	問題なし
直線, 平面, 球	$+,-,*,/$	公式代入

2.2.5 行列と行列式

行列と行列式はそのまま Mathmatica で計算でき代入なども問題ない (表 12). n 乗に関しては M^n ではなく行列の n 乗 (MatrixPower) を使用する. 行列の積はピリオド (.) を使用する. 教科書などの記号と異なるので注意が必要である. 逆行列 (Inverse), 転置 (Transpose), 行列式 (Det), 固有値 (Eigenvalues), 固有ベクトル (Eigenvectors), ガウスの消去法 (RowReduce) などの関数名は覚えておく必要がある. 行列式の展開は組み込み関数がなく手操作またはプログラムの必要があり, 補助的な使用にとどまる. 掃き出し法も途中の変形を確認したければ手操作またはプログラムの必要がある. 階数 (MatrixRank) はバージョンを確認し使用する.

表 12: 行列と行列式の分野

分野	コマンド	備考
行列	$+,-,*,/,.,$ $MatrixPower$	M^n は意味が異なる
逆行列	$Inverse$	問題なし
1次変換	$Transpose, .$	問題なし
行列式, 展開	Det	手操作
掃き出し法	$RowReduce, [[]]$	手操作
固有値と対角化	$Det, Solve,$ $EigenSystem$	問題なし
行列の階数	$MatrixRank$	ver.4 × ver.5 ○

2.3 3年の数学

3年の数学は微分法, 積分法, 偏微分と重積分, 微分方程式と複素数の5分野からなる.

2.3.1 微分法

3年の微分法は高次導関数や逆関数の微分法, ロピタルの定理, べき級数などである. 計算は Mathematica の組み込み関数で処理でき, 学生の問題演習の補助として使用して問題ない (表 13). 曲線の凹凸や変曲点については, 増減表の代わりにグラフで対応する. 収束半径などは公式に代入し極限計算を行う. 収束を判定するための基礎的な知識は理解していなければならない.

表 13: 微分法の分野

分野	コマンド	備考
第2次導関数と凹凸	$Diff, Plot$	問題なし
逆関数	$Solve$	問題なし
媒介変数	$Plot$	問題なし
速度加速度	$Diff$	公式に代入
極座標と曲線	$PolarPlot$	問題なし
平均値の定理, ロピタルの定理	$Limit$	問題なし
テイラーの定理	$Series, Diff$	公式に代入

2.3.2 積分法

積分は組み込み関数で処理できる (表14). 誤差関数については基本的な知識が必要である. 積分 $\int_0^{\infty} e^{-x^2}dx$ の値は警告がでるが Integrate 関数で計算結果は問題なく得られる. 三角関数や対数関数などで結果が複雑な形になるものはさらに式変形が必要となる. Mathematica の組み込み関数で区分求積法を自動的に積分に変換してくれる関数はない. 学生が自力で変換しなければならない.

表 14: 積分法の分野

分野	コマンド	備考
積分法	$Integrate$	三角対数関数は式変形必要
定積分	$Integrate$	問題なし
区分求積法	$Limit, Sum$	積分置換え不可
極座標	$Integrate$	問題なし
曲線の長さ	$Integrate$	公式に代入
広義積分	Erf	erf 関数

2.3.3 偏微分と重積分

偏微分と重積分の分野でも微分 (Diff) と積分 (Integate) で対応できる (表 15). Mathematica は多変数関数の極値を求めるための連立方程式も解ける. 全ての解をもれなく見つけてくれるので, 学生が自分の計算した答えを検証するのに適する. ラグランジュの乗数法を計算する場合なども便利に使える. 2変数関数であればグラフ描画 (Plot3D) で極値の様子などを見ることもできる. グラフでは, 連続かどうかの確認などもでき, 重宝する.

表 15: 偏微分と重積分の分野

分野	コマンド	備考
偏微分, 連続, 平均値の定理, 全微分	$Diff$	問題なし
極値, 陰関数定理, 接線, 法線, 解異点	$Diff, Solve$	問題なし
ラグランジュの乗数法	$Diff, Solve$, 代入	問題なし
重積分	$Integate$	順序変更は別 自動変更は不可
極座標による重積分	$Integate$	問題なし

2.3.4　微分方程式

微分方程式は微分方程式の解析解 (Dsolve) で求めることができる (表 16). 初期値問題や境界値問題も Dsolve で解ける. Dsolve は学生が微分方程式を解けなくても解を求めてしまうため, 解法の練習としては役立たない. 検算として利用することになる. 全微分方程式や特殊な形の微分方程式では, あらかじめ行っておくべき式変形などの知識も必要になる. Mathematica の組み込み関数では包絡線や特異解などは自動的に求めてくれない. 手順に従って計算する必要がある.

表 16: 微分方程式の分野

分野	コマンド	備考
微分方程式	$Dsolve$	問題なし
全微分方程式	$Dsolve$	基礎知識必要

2.3.5　複素数

３年の複素数は加減乗除とド・モアブルの定理で本校では１年または２年で講義される. 計算に関しては Mathematica の組み込み関数で問題なく計算できる (表 17). 複素方程式で与えられた図形がどんな図形かは学生が式変形を行い自ら判断する必要がある.

表 17: 複素数の分野

分野	コマンド	備考
複素数と演算	$+, -, *, /$	問題なし
複素平面	$ListPlot$	問題なし
ド・モアブルの定理	公式代入	問題なし
n 乗根	$Solve$	問題なし
図形への応用	Abs, Im, Re	図形判断不可

3.　まとめ

高専の１年から３年の教科書を通して問題を解いてみることで, ほとんどの問題が Mathematica の基本関数で解けることがわかった. 計算途中の論理的過不足も判明し, 問題の途中で漠然としていた点も違った視点から眺め直すことによりはっきりする. そういった意味で, 学生に早くから数式処理ソフトのリテラシー教育を行うことは重要である. 学生は数式処理ソフトの使用法を体得すると同時に, いろいろな使い方を知ることになる. 数式処理ソフトは本来, 論文作成時の計算の検証に使用されてきたものであり, 技術者の基礎的な能力を向上させるために有用である. また, 情報をわかりやすく計画提案する能力も数式処理ソフトのリテラシーで向上するものと期待できる.

4.　おわりに

最も商業的に成功しているといわれている数式処理ソフトは Mathematica であり, 現在の最新 Mathematica のバージョンは７である. 総合情報センターに導入されている Mathematica はバージョンが 4.2 であり不等式を解いたり, 行列の階数を関数１つで計算するにはいたらない, 今回の調査で判明したように Mathematica は学生が問題を解くに当たっての補助機器として学生を助けることができる. バージョンがあがるにつれ使い易くなっており, web やネットワークの機能も取り込みつつある. また工学分野では, 設計, 開発, 生産の標準的ツールになりつつある.

学生が自ら問題をみつけ探求していくためにコンピュータテクノロジーの効果は圧倒的であり, とくに Mathematica の計算能力とグラフィックの機能だけでも学生の勉学に利益をもたらす.

学生が問題を解いている傍らに数式処理システムがあり, 必要なときに計算を試したりアルゴリズムを確認したりということがすぐにできる環境で勉強できることは技術者の基礎的な能力を向上させ, みずから考えるという能力をはぐくむものと考えられる.

そういった意味で, 数式処理システムをわかりやすく使いこなす数学の授業計画を今後とも提案していきたい.

参考文献

(1) 田代嘉宏, 難波完爾:「新編　高専の数学 1 (第 2 版)」, 森北出版,2000.
(2) 田代嘉宏, 難波完爾:「新編　高専の数学 2 (第 2 版)」, 森北出版,2000.
(3) 田代嘉宏, 難波完爾:「新編　高専の数学 3 (第 2 版)」, 森北出版,2000.
(4) 梶原壌二:「Mathematica と Theorist での大学院入試への調査園」, 現代数学社,1994.
(5) 田村直行:「神戸大学 Mathematica 入門ページ」, http://bach.istc.kobe-u.ac.jp/mma/nyumon/.
(6) Stephen Wolfram.:「The Mathematica Book, 5th ed.」Wolfram Media , 2003.
(7) Wolfram Research:「Guide to Standard Mathematica Packages」,Wolfram Research, Inc.