

数式処理システムの教育現場での利用 I —教材配付サーバの構築と教育実践報告—

神戸大学大学院

総合人間科学研究科 佐伯 昭洋(Akihiro SAEKI)*

神戸大学発達科学部 高橋 正(Tadashi TAKAHASHI)†

概 要

数式処理システムの教育現場での利用は、未だに、一部の教師の試みとしての報告が多い。その理由としては、教育現場において数式処理システムを授業のなかで使うには、生徒にその操作をさせなければならないこと、さらには、適切な教材の選定等様々なことを留意しなければならないからである。したがって、数式処理システムが安価なパソコンで使えるようになって、数学教育現場での利用は進んでいない。また、数式処理システムを、現場の数学教師が授業で使うには、数学の授業で使えるサンプルプログラムを豊富にしなければならない。そこで、本研究においては、日本の中学校及び高等学校の数学の授業において使える Mathematica のサンプルを教材配付サーバに置き、それを用いて、数学の授業実践を行うことにより、どのような数学の授業が可能になるかを明らかにする。

1 はじめに

近年、数学教育ではコンピュータを利用した授業が行われており、またそれに関する教育実践報告が活発である。通常の授業では、教師主導型の一斉授業であるのに対し、コンピュータを利用した授業では、生徒自身が考え、考察を進めて行くことが可能となる。このような発見学習的な授業においては、生徒が数学に興味を抱いたり、数学と他の学問の関連性、数学を学ぶ意義を見出すことが期待されている。しかし、コンピュータを利用した授業で用いられるソフトウェアは、教師・生徒双方にとって、操作や入力知識を新たに必要とする。そのため、実際にコンピュータを利用した授業が少ない。また、発見学習を行うには誤作動や操作ミスなどにも対処しなければならないので、教師の負担は今まで以上に増大する。

本研究では、中学校及び高等学校の数学の授業において使える Mathematica の教材サンプルを教材配付サーバに置き、それを現場の数学教師が必要に応じてダウンロードする仕

*saeki@main.h.kobe-u.ac.jp

†takahasi@kobe-u.ac.jp

組みを構築する方法を示し、その教材サンプルを用いて授業を行った際の実践結果を報告する。授業内容としては、高等学校の数学Ⅱ「三角関数」で扱われる $y = a \sin r (\theta - \alpha)$ のグラフに関する性質について、Mathematica を用いて様々な具体例を提示することで視覚的な理解を生徒に支援する授業を行った。

そして、実践授業を基にして、数式処理システムを利用した授業の特徴及び問題点について考察し、どのような授業を行うべきかについて考察を行う。

2 Mathematica 教材配付サーバの構築

2.1 教材データサイト

数学教師は、教材の一部として特定のソフトウェアを用いるのであって、特定のソフトウェアを利用する素材として数学という例を挙げているのではない。したがって、ソフトウェアを用いてプログラムを作成できることのできない数学教師でも利用できるように、その負担を軽減することが必要である。

現場の数学教師は、下記の問題を抱えている。

- ソフトウェアの操作方法や利用する利点分からない。
- プログラムの作成方法が分からない。

特に、授業で使用するプログラムの作成は困難である。自らプログラムを作成するには、ソフトウェアに関する文献を調べるか、ソフトウェアに詳しい人に手伝ってもらふ事例が多いようである。しかし、授業で用いるためには、必要なプログラムを素早く入手しなければならない。このことを可能にするものとして、「数学の教材として活用できるプログラムのデータベース」を構築することが必要である。さらに、個人レベルで収集・蓄積できるリソースの数には限りがあるため、教材を広く求めることも考慮しなければならない。

Mathematica は、MathReader という読み込み専用の無料のソフトウェアがあり、教師自身がプログラムを実行することで、授業で有用かどうかの判断基準になる。そのため、本研究では、Mathematica の教材用プログラムをサーバ蓄積することを行った。

Mathematica を開発した Wolfram Research, Inc. では MathSource という Mathematica のサンプルプログラムやパッケージを蓄積しているデータサイトを公開している ([2])。蓄積されているプログラムは膨大であり、その内容は多くの分野にわたる。また、蓄積されているプログラムはテーマ別に整然と分類されており、その内容は多くの分野にわたる。しかし、MathSource は、日本における中学校・高等学校の数学教育に重点を置いたものではないため、蓄積されているプログラムは日本のカリキュラムには適応していないものが多い。また、プログラムを実行すると、どのようなことが行えるかはダウンロードする前

に書かれているが、それらは英語で書かれているため、英語を不得手とする現場の数学教師はプログラムの内容を理解するだけで一苦勞である。

以上のことから、Math Source を利用できる現場の数学教師は、「ある程度英語が分かり、Mathematica を使いこなしている者」に限定される。日本の中学校・高等学校の数学教育を対象にするならば、プログラムの内容を「中学校・高等学校の数学教育」に絞り、プログラムの説明も日本語で明確に行う必要がある。本研究では日本の中学校・高等学校における数学教育に活用できる教育用 Mathematica プログラムを蓄積するデータサイトの構築を行った。そしてさらに、データサイトにはどのようなプログラムを蓄積すればよいかを考察した。

2.2 Mathematica プログラムの種類

Mathematica には.nb という拡張子を持つ「ノートブック」と、.m という拡張子を持つ「パッケージ」に分類される。

ノートブックはプログラムを作成、実行することができる。全くの白紙から作業を始めるため、自分が考えた方法でコマンド（関数）を入力し、実行を繰り返して上手くいかなかったら修正を加えるということを繰り返す。これは試行錯誤を伴う発見学習を行う際には有用である。特に、長期間にわたって Mathematica を授業で利用する場合、操作する側（生徒）の操作能力が上がるため、理想的な発見学習が行えることが期待される。その反面、覚えなければならないコマンドが必然的に多くなる。そのため、コマンドを覚えさせるのに時間を費やすことになるので、数回だけの授業で利用する場合や、教師が提示用ツールとして用いる場合には適切ではない。

また、ノートブックはチュートリアルとしての役割も果たす。ノートブックはプログラム入力だけではなく、文字のフォントやサイズを変更したテキスト入力や、背景の色を変化させることが可能である。この機能によって、プログラムの詳細をテキストで分かりやすく述べることが可能になる（図1）。

一方、パッケージは新しいコマンドを定義したプログラムのことである。ノートブックと異なり、パッケージはファイル自体を開く必要がない。パッケージに収録されている新しく定義したコマンドを利用する場合は通常のノートブックでそのパッケージをロードすればよい。図3で<<TriOptions.m という箇所がそれに該当する部分である。パッケージをロードしたら、そのパッケージで定義されているすべてのコマンドを利用することができる。教育現場では数学教師があらかじめ作成しておいたパッケージを、生徒が利用するコンピュータに保存しておき、授業になったらそのコマンドを利用させるという形で利用することができる。

一度に多くのサイコロを振る

サイコロを何回も振ることで確率の概念（もしくは大数の法則）を実際に体感させることは、よく行われることです。以下に記すコマンドによって、一度に擬似的に多くのサイコロをいっぺんに振り、それを度数分布表（略式ですが）およびヒストグラムで表すことができます。

```
BeginPackage["Dice`", {"Graphics`Graphics`"}];
Begin["`Private`"];
{Dice[n_] := Table[Random[Integer, {1, 6}], {n}];
DiceCount := Table[Count[%, i], {i, 6}];
Graph := BarChart[Table[Count[%%, i], {i, 6}]];
End[];
EndPackage[];
```

Dice[変数]・・・1から6までの整数をランダムに[変数]個分表示することで、擬似的にサイコロを振る。
DiceCount・・・Diceの結果を{(1の目が出た回数)、(2の目が出た回数)、.....、(6の目が出た回数)}の形で表示する。
Graph・・・上のDiceの結果をヒストグラム（棒グラフ）で表す

注意：Dice->DiceCount->Graphの順番で実行して下さい。
 もしDiceCountはいらぬ場合は、Dice.mの、上の赤い部分を削除した上で、その下の紫色の部分を変えて下さい。

■ 利用方法

まず、以下のように入力してEnterキーを押します。こうすることでDiceというコマンドを使えるようになります。

```
<<Dice.m
```

これでDice,DiceCount,Graphという一連のコマンドが使えるようになりました。
 では、実際に実行してみましょう。手始めに50回ほど振って（振ったつもりになって）みましょう。

```
Dice[50]
```

```
{4, 3, 6, 4, 1, 6, 3, 2, 1, 3, 5, 6, 1, 5, 1, 2, 5, 1, 5, 3, 4, 3, 6, 1, 5, 1, 6, 5, 1, 5, 1, 5, 4, 2, 4, 6, 3, 3, 6, 4, 3, 4, 5, 4, 1, 5, 1, 5, 6, 1}
```

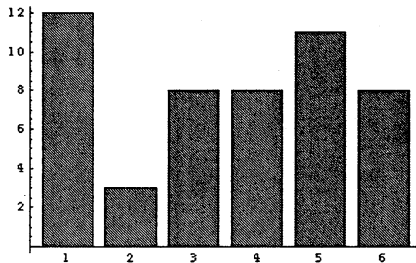
それから、**DiceCount**でどのように分布されているか、調べてみましょう。

```
DiceCount
```

```
{12, 3, 8, 8, 11, 8}
```

さらに、それをヒストグラムで表します。

```
Graph;
```



ただ、Graph;の「;」を抜かしてもグラフは表示されますが下に長い詳細が出力されます。

図 1: ノートブックの例

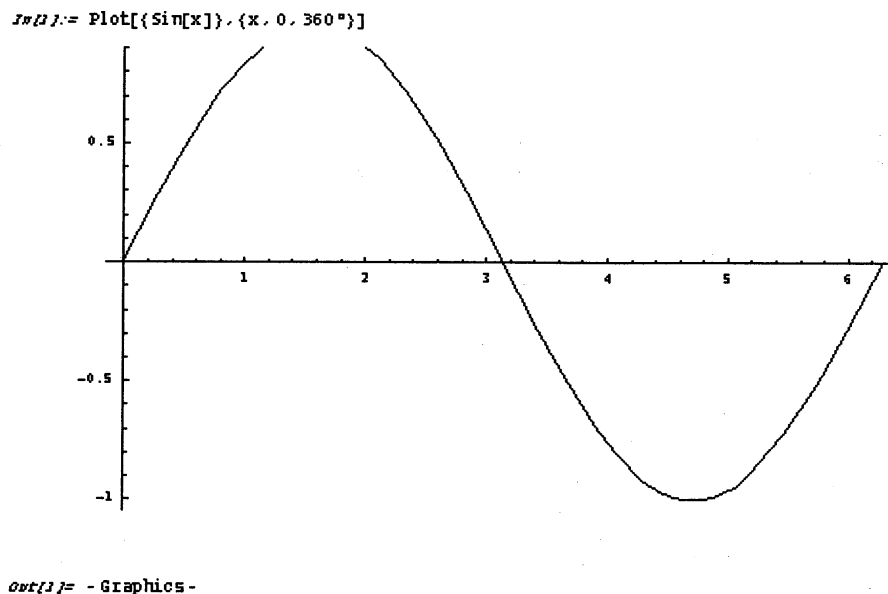


図 2: パッケージを使わなかった場合

また、パッケージを用いることで入力すべきプログラムの簡略化を行うこともできるため、授業内容の理解に集中することができる。また、図 2 及び図 3 のように、関数の設定も授業を円滑に進めることが出来るように設定し直すことも可能である。図 2 はパッケージを使わなかった場合で、図 3 はパッケージを使った場合である。Mathematica は数学教育に有効であるが、必ずしも日本の数学教育現場に対応しているわけではない。このようにパッケージを有効に利用することで、はじめて授業を円滑に行うことができる。

2.3 教育用 Mathematica プログラムの蓄積方法

Mathematica を授業で用いる場合、パッケージのみを用いるのでは詳しい説明や具体例を挙げるができない。また、ノートブックだけでは授業で用いる場合、プログラム全体を入力しなければならない。パッケージとノートブックの両方を用いると、数学的な考察を行うのに不可欠な入力とはノートブックに直接入力を行うことができる上、TriOptions のように、不必要な入力をパッケージによって簡略化するという使用方法をとるのが効果的である。従って、プログラムは、パッケージとノートブックを併用した形で蓄積している。具体的には以下のようにプログラムの蓄積を行っている。

座標を明記する Ticks や、関数の表示範囲を示す PlotRange など、Plot という関数をグラフ化するコマンドだけでも多くの設定コマンドがある。これらの設定コマンドは、グラフや出力結果を見やすいようにグラフの色を変えるなど、設定を変更を行うものである。しかし、設定の変更によりプログラムが複雑になりすぎると、操作する側が「何をやっているのか」分からなくなる。そのためにパッケージを用いるが、生徒自身が考えることまで、

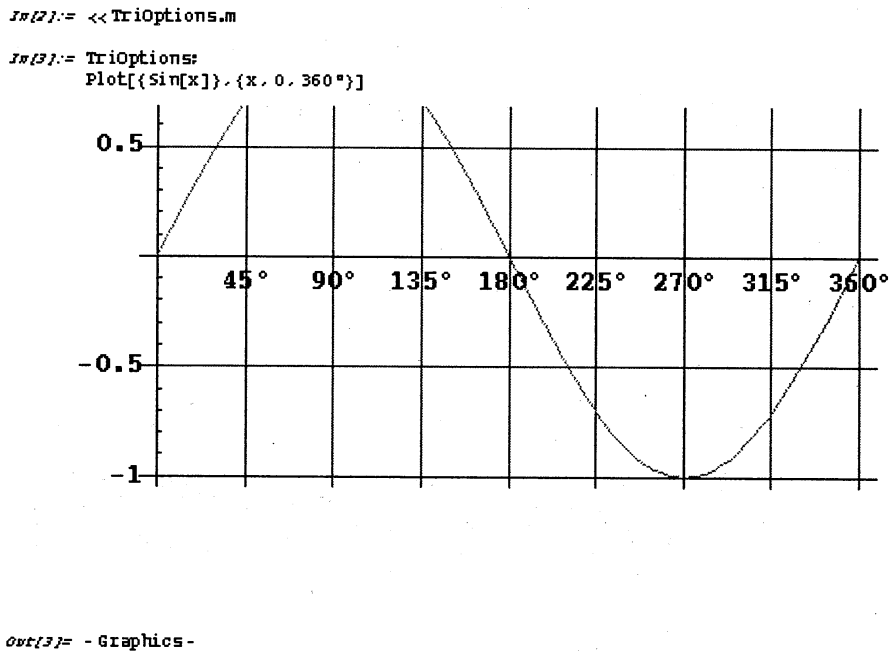


図 3: パッケージを使った場合

パッケージの内容で省略してはならない。

また、教師はそれぞれの授業の方法を持っているため、パッケージがその教師の授業進行と異なる場合も考えられる。したがって、ある程度の編集を行うことができるようにしなければならない。通常ではパッケージの説明はコメントアウトでパッケージプログラムに掲載されているが、「パッケージのどの数字を変化したら、プログラムはどのように変わるか」などの具体例を含めた説明は行われていない。

そのため、ノートブックの利点であるワープロソフトに準じたテキスト入力機能を活かした説明によって行うものとする。例えば、図 1 では赤や紫の箇所を変更もしくは削除するとどのようなことが起きるかということを記述している。その際、重要な箇所は文字の拡大を行うことや、背景の色を目立つ色に変える等の配慮を行っている。

ノートブックの形で「空間図形の図示」や「軌跡の移動」といったプログラムを保存し蓄積することで、それらのプログラムをダウンロードした者は Mathematica の機能をその目で確かめることができる。また、これらのプログラムを教師が Mathematica を提示用ツールとしての利用することも考えられるため、パッケージが作成されていないプログラムも、ノートブックの形で蓄積することにする。この時も、同一ファイル内に詳細な説明を述べたテキストを記述している。

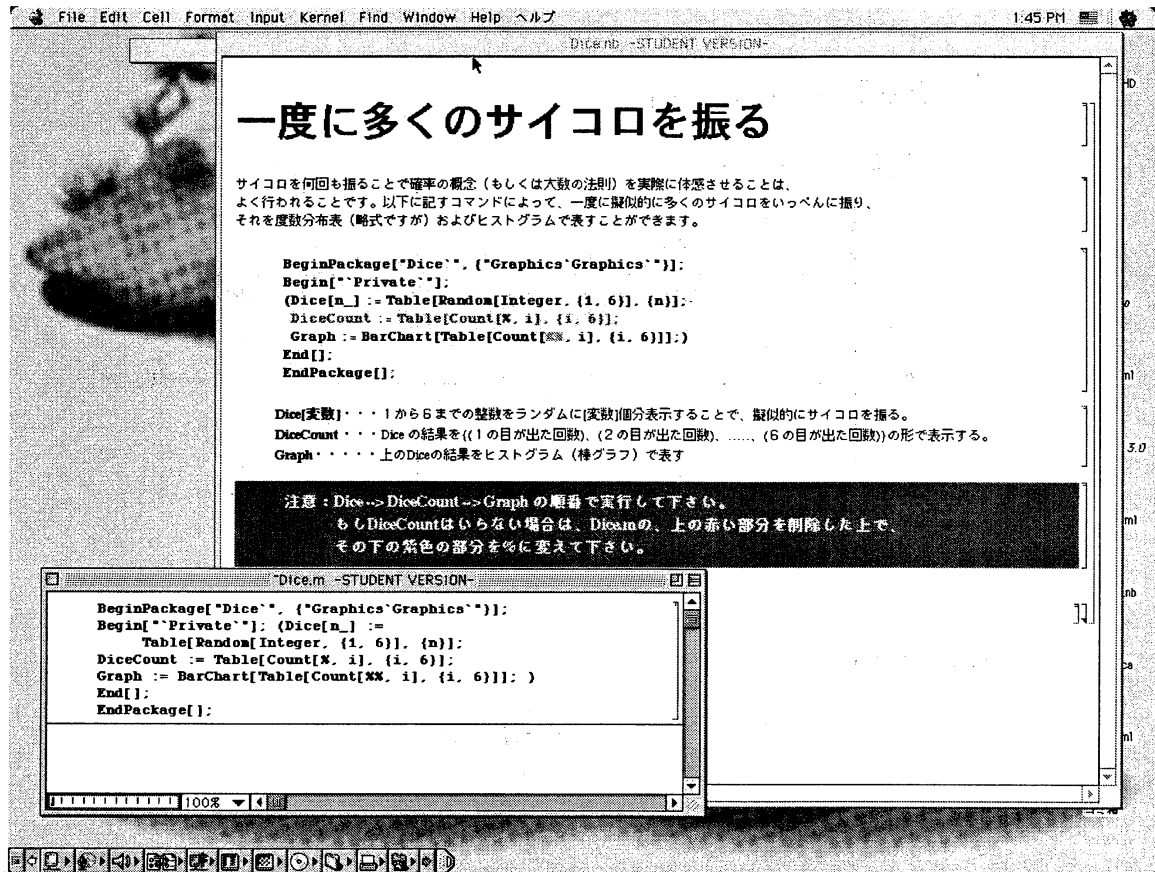


図 4: ダウンロード画面

2.3.1 ダウンロードの方法

プログラムのダウンロードは簡単でなければならない。しかし、プログラムをそのままの形でネットワーク上に提示していると、ダウンロードの際にうまくダウンロードできない。また、.m、.nb が Mathematica のファイルであることを認識させなければ、ファイルは「編集不可なテキストファイル」としてダウンロードされてしまう。このことを行うには OS の設定自体を変更させなくてはならない。しかし、圧縮・解凍ソフトでプログラムを圧縮することでこの問題は解決できる。教育用 Mathematica プログラム配付サーバでは、Windows 及び MacOS で用いられている Zip 形式のファイルに変換して図 4 のようにダウンロードを行うことができるようにしている。プログラムは Macintosh 用の Zip 形式ファイル圧縮・解凍ソフトである ZipIt を使用した ([3])。

2.3.2 プログラムの検索方法

プログラムは「科目別」と「単元別」に分類し、ホームページからは、いずれの分類からでも検索が行うことができるようにした。

「科目別」ではプログラムを使用する学年もしくは科目で分類した。例えば、実践授業で利用したプログラムは「数学Ⅱ」に蓄積されている。また「因数分解」のように復数年にわたって履修する単元がいくつか存在する。そのため、混乱を避けるため「単元別」による分類も行うことにした。

2.3.3 一般公開

現時点では、プログラム数の不足により、一般公開は行っていない。しかし、一般公開を行うことによって、プログラムを広く求めることも可能になる上、プログラムをダウンロードした利用者からの評価や改善案等のフィードバックが期待できるため、プログラムの数が整い次第、早急にその公開を行う予定である。

3 実践授業

Mathematica を用いた実践授業の経過・結果及び生徒の感想・意見をまとめる。実践授業では数学Ⅱの「三角関数」で学ぶ多くの実例を見せることで、生徒の理解を支援することを試みた。実践授業は2000年11月30日及び12月6日の2日間、私立甲南女子中学・高等学校で行ったものである。

3.1 実践授業にあたって

三角関数のグラフを Mathematica で図示すると、 $y = a \sin r(\theta - \alpha)$ のグラフの a および r の値を変型させるだけで簡単に図示できる。すなわち、以下のグラフが数字入力のみで作成することが可能である。

- a の値を変型すると、 y 軸方向に拡大したグラフ。
- r の値を変型すると、 θ 軸方向に拡大したグラフ。
- α の値を変型すると、 θ 軸に対して平行移動したグラフ。

設備上の関係で、生徒にコンピュータを実際に操作させることができなかった。また、コンピュータの図だけに目を奪われて実際に鉛筆でグラフが作成できないことも考えられる。そこで、この実践授業の前の授業に「鉛筆による三角関数のグラフ作成」を行うことで、この弱点を補った。また、実践授業の詳細は以下の通りである。

実践授業案

目標：標準的な三角関数が θ 軸及び y 軸方向に拡大・縮小したグラフ及び θ 軸方向に平行移動したグラフの特徴を理解させる。

前授業の内容 (準備段階)

0° から 360° までの三角比を求めさせ、生徒にグラフに書かせた上で、「周期」という概念に触れた。このとき、以下の点に留意して板書及び指導を行った。

- $\sin\theta$ のグラフは 180° を境に左右対称であること。
- $\cos\theta$ のグラフは $\sin\theta$ がずれたものであること (次の授業の為、具体的に触れるのは避けた)。
- 正接のグラフは周期が正弦・余弦とは異なり、 180° であること。

授業計画 (2 時間)

1 時間目: 関数を実数倍したグラフと標準的なグラフの違いを調べる。

2 時間目: 一般角を実数倍したグラフの特徴と、 x 軸に平行移動したグラフの特徴をしらべる。

教材: 授業プリント及び、教科書 (啓林館・数学 II)

備考: プロジェクターを通してモニタに映し出された結果を生徒に示す。また、別に板書を行う。

1 時間目

目標:

- $y = a \sin\theta$ のグラフの特徴を理解する。
- 「周期」の概念および、標準的な三角関数のグラフの特徴に対する認識を確かなものにする。

項目	指導内容	留意点
導入	$2\sin\theta$ と $\sin 2\theta$ の相違点について	板書及び口頭による説明。理解度が低いようならば実例を挙げる。
展開	$y = 2\sin\theta$ と $y = \sin\theta$ の違いを調べる	Mathematica による図示。巡回・指導を行ったあとに説明を Mathematica のグラフを用いて説明する。板書による指導。
	$y = a \sin\theta$ の性質を具体例を用いて推測させる	Mathematica による図示。いくつかの具体例を挙げる。

2 時間目

目標: $y = \sin r\theta$ 及び $y = \sin(\theta - \alpha)$ のグラフの特徴を理解する。

項目	指導内容	留意点
復習	$\sin\theta$ と $\sin 2\theta$ の違いを調べる	Mathematicaによる図示。巡回・指導を行ったあとに説明を Mathematica のグラフを用いて説明する。
展開	$y=\sin r\theta$ の性質を具体例を用いて推測させる	Mathematicaによる図示。巡回指導を行い、具体例を示す。他の三角関数についても同様のことを行う。時間に余裕があれば、以前学んだ公式の復習を行う。
	$y=\sin\theta$ の性質を具体例を用いて推測させる	Mathematicaによる図示。一般数について認識があまりないようなので、いくつかの具体例を挙げる。

3.2 実践授業にあたっての留意点

数式処理システムを利用する授業は生徒にとって初めてであり、Mathematicaにも触れたことがない。そのため、実践授業では以下の事に留意した。

- プログラムを実行する際、「どのようなことを行ったのか」を明確にする。
- 生徒の中には、コンピュータを利用するのだから難しい事を学ぶに違いない、と警戒する者もいた。そのため、「周期関数」などの重要性の低い用語に関しては極力平易な言葉を用いた。
- Mathematica ではグラフの θ 軸が弧度法で表されるため、度数法に直すコマンド `TriOptions` を新たに作成した (図 5 参照)。

```

BeginPackage["TriOptions`"]
Begin["Private`"]
TriOptions := SetOptions[Plot, Ticks -> {{45°, "45°"}, {90°, "90°"}, {135°, "135°"},
  {180°, "180°"}, {225°, "225°"}, {270°, "270°"}, {315°, "315°"}, {360°, "360°"},
  {405°, "405°"}, {450°, "450°"}, {495°, "495°"}, {540°, "540°"},
  {585°, "585°"}, {630°, "630°"}, {675°, "675°"}, {720°, "720°"},
  {765°, "765°"}, {810°, "810°"}, {855°, "855°"}, {900°, "900°"}}, Automatic],
  GridLines -> {Table[ $\frac{\pi n}{4}$ , {n, -2, 20}], Automatic}, PlotStyle -> {RGBColor[1, 0, 0],
  RGBColor[0, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1], RGBColor[1, 1, 1]},
  TextStyle -> {FontSize -> 18, FontWeight -> SemiBold}]
RESET := SetOptions[Plot, Ticks -> Automatic,
  GridLines -> None, PlotStyle -> Automatic, TextStyle -> Automatic]
End[]
EndPackage[]

```

図 5: TriOptions.m

3.3 授業の反省

準備のための授業を含めると3時間の授業を実施したが、生徒個人の理解力の差を埋めるには至らなかった。また、時間の都合で問題演習をほとんど行えなかったが、時間を割いてでも問題演習の時間を作るべきであった。

授業後、生徒にアンケートを行い、2クラス81名のうち59名が回答した。その集計結果の一部は以下に示す。

今回の授業はいつもの授業と比べてどうでしたか？

よかった 31人

悪かった 10人

たいして変わらなかった 18人

またもう一度数式処理システムを使った授業を受けてみたいですか？

受けてみたい 13人

今度は自分が操作したい 26人

二度と受けたくない 3人

どちらでもない 17人

重要な箇所のみならず不必要な箇所までノートをとる生徒が多数見られた。また、図示によるグラフの特徴はつかめたものだと確信しているが、視覚的な事を強調する余り数式による計算がおろそかになってしまった感が否めない。さらに、グラフは見えたがプログラムの文字が見えない、すなわちどのようなグラフを描いているのか分からないという批判的な意見がいくつかあった。確かに部屋の構造上、文字がみやすいとは言えなかった。その対策として描くグラフやこれから実行することについて説明を行ったが、配付したプリントにもどのようなグラフを描くか明言しておくべきであった。

今回の実践授業では視覚的な理解から「数学的な思考」へと生徒を導くのが目的であった。しかし、視覚的な理解を求めるのに時間を費やしてしまい、「数学的な思考」へのつながりが行えたのか疑問が残る。生徒の意見に「グラフとかは自分で書くことで身に付くもの」とあるように、事前にもう少しグラフを手書きで作成させておくべきであった。

また、数式処理システムを使った授業に関しておおむね肯定的な意見や、「将来性がある」と評価する生徒がいたのに対し、同アンケートで「いつもの授業がいい」と答えた生徒は全体の半分に満たなかった。これは、数式処理システムは、数学の理解を支援するものであるが、生徒が興味を持つ要素ではない、興味を抱かせるにはやはり授業の内容を充実させなければならない。

今回の実践授業でのポイントは「いかにして『実質的淘知』を数学的思考抜きで理解させ

られるか」であった。その点だけを考えれば目的を達したということになるが、「生徒の興味を引く」「数学的思考への橋渡しをする」という点に関しては目立った効果はなかったように思える。いかにして生徒の興味を引き、数学的思考へとつなげるような授業を行うことが出来るかが、数式処理システムを利用した授業を活性化させる鍵になる。

4 考察

4.1 テクノロジーを用いた授業方法

テクノロジーを用いることで「数学を理解する」という目的が達成されるだけではなく、「試行錯誤を主体とした学習」等、数学そのものの本質を変容させることもある。このような「道具」は「シンボル装置」とみなすことができる ([1])。シンボル装置は人間と学習対象の間に入るもののことで、そのため人間とシンボル装置・シンボル装置と学習対象の間に何らかの関係がある。この関係のことを界面と呼ぶ。数学教育においては生徒（人間）とテクノロジー（シンボル装置）の間に「テクノロジーにいかに関わり親しむか」という界面（第1界面）とテクノロジーと数学（学習対象）の間に「テクノロジーを使ってどのように数学を学ぶか」という界面（第2界面）が存在する。

そのため、数学教育の現場でテクノロジーを活用させるためには、2つの壁を乗り越えなければならない。まず、第1界面を超える必要があるが、その壁はできるだけ低いものでなければならない。壁が高いとその分、テクノロジーの活用には費やす時間は多くなり、学習の「道具」としての活用は望めなくなる。従って、第1界面の壁、すなわちテクノロジーの操作は極力簡易なものでなければならない。

第2界面ではテクノロジーを使って数学教育がどのように変わるかが焦点となる。手計算や手書きのほうが効果的である場合は、テクノロジーを使う必要は全くない。テクノロジーを用いた方が効果的であることや、テクノロジーでなければできないことを見い出さなければならない。

また、上記の界面に加えて生徒と数学の間にも壁が存在する。この壁が高すぎると生徒の多くが数学への興味を失い、低すぎると学習の意味をなさない。生徒のレベルに応じた授業を行うことは自明の理であるが、これはテクノロジーを用いた場合でもあてはまる。効果的にテクノロジーを用いることのできる授業や、テクノロジーを用いることでしか行えない授業は概して高度な内容であることが多い。しかし、テクノロジーはあくまでも試行錯誤による仮説設定を支援するための道具であって、演繹や検証は生徒自身が行わなければならない。

生徒はテクノロジーを学ぶわけではなく「数学」を学ぶのである。そして、テクノロジーを用いることで「数学」自体が簡単になるわけではない。従って、第1界面は生徒の負担にならない様に出来るだけその壁を低くし、第2界面は生徒の理解度、学習の度合いに応じ

た高さでなければならない。それでは、どのように数学教育の現場でテクノロジーを用いればよいのか。原則としてはテクノロジーの特性を活かし、試行錯誤を中心とする授業形態が望ましい。しかし、試行錯誤による発見学習のみでは、一部の数学的思考を既に身に付けている生徒のみが授業を理解し、そうでない生徒はどうすればいいのか悩んだ末に、「結果を丸暗記するだけ」という状況に陥ってしまう。基礎的な概念の理解という目的でテクノロジーを利用する時には大きな問題となる。したがって、大多数の「数学的思考を身に付けていない生徒」に理解させるには指導面による配慮を行わなければならない。

- 「自分で解いた」という達成感や満足感が生徒は味わえないのではないか。
- 途中経過がブラックボックス化してしまい、ますます過程を無視する傾向になるのではないか。
- テクノロジー=答えを出してくれるものと生徒は解釈してしまい、数学の授業を受ける必要性（特に計算分野）を認識しないのではないか。

テクノロジーを利用して基礎的な概念を理解することを目標としている場合、通常の授業を意識して指導を行ったほうがよい。実践授業では時間上の都合で $y = a \sin r(\theta - \alpha)$ のグラフ作成は次の授業に持ち越しとなったが、グラフをイメージさせ、実際に表を作成して作図させてから Mathematica でグラフを表示する、という方法も考えられる。この場合、逆に表を作成するよりも格子点を求めたほうが速く描けることに気付くことが多い。表を作成できない生徒はどのようなグラフになるのか、見当もつかないであろう。しかし、Mathematica によって作成された「完成されたグラフ」を見ることで、生徒はこのグラフの性質を視覚的に理解することができる。

板書とは異なり、Mathematica では生徒が自分自身の理解度を確かめる為、変数に他の数字を代入した、生徒オリジナルのグラフを作成することができるため、生徒が授業内容を理解した「つもり」になって進んでしまうことは少なくなる。

教師が生徒に Mathematica を用いて表示されたグラフや表を提示するという形の授業を行う場合、独りよがりな授業になってしまう恐れがある。適切な教材を作るのは教師自身であって、テクノロジーではない。したがって、入念な授業研究を行わなければ、かえって逆効果になる恐れがあることにも留意しなければならない。

生徒の理解を支援する手段は、生徒が実際にテクノロジーを操作する場合における「新規コマンドの作成」「サンプルプログラムの提示」といった第1接面を低くするということも含まれる。具体例としては「二次方程式で虚数解が出たら『解なし』と出力させるプログラム」や、実践授業で用いた TriOptions が挙げられる。

しかし、新しいコマンドの作成やサンプルプログラムを提示する際に注意しなければならないのが「ブラックボックス」である。ブラックボックスとはプログラムが実行して結果を出す過程が使用者には見えなくなることであり、テクノロジーを利用する上での問題点

の一つとされている。新しいコマンドを作成する目的は「煩雑な入力を省略する」ことにあるが、その中で、「必要な入力」までもが簡略化されないようにしなければならない。

「放物線と直線に囲まれた部分の面積を求める」という問題を考えてみる。この問題を解くには、「連立方程式を解いて交点の座標を求める」「2交点の定義域内で、どちらが上に位置しているかを確認する」という過程を経て、初めて「交点間の範囲内での定積分により面積を求める」ことが出来る。授業においては、これらの過程を省略してはならない。この場合で、「連立方程式を解くコマンド」「定積分の計算を行うコマンド」は、Mathematicaでは、SolveとIntegrateのコマンドが使えるので、パッケージを用いて新たにコマンドを作成する必要はない。

4.2 視覚的な支援や具体例から理解につながる授業

基礎的な概念の理解は、教師主導型の一斉授業によって、「教師から生徒に教えられる」ことが多く、それらを理解していることを前提として授業が進められる。生徒の理解度や学習達成度によっては、視覚的な支援や具体例を用いた理解、つまり数学的な論理が最小限で済むような指導を行うことも必要である。授業内容によっては、従来通りの板書及び、紙と鉛筆を使った授業の方が効果的である。特に、板書には「手軽に板書の一部を消去もしくは編集できる」というテクノロジーにはない利点がある。また、生徒もノートに写すべき事柄と念頭で理解すべき事柄の区別が容易になる。Mathematicaを数学教育を行う際の「道具」として用いる時には、実演だけでは確かな理解を得られるとは限らない。Mathematicaを用いた場合でも、黒板やノートをフルに活用する必要がある。

現在、ほとんどの中学校・高等学校ではコンピュータ室が設置されているが、その多くが板書やノート取りには適さない環境にある。「Mathematicaを使った授業」「Mathematicaを使わない授業」にこだわるのではなく、「黒板、チョーク、ノート、鉛筆、教科書、Mathematica」のすべてを授業で利用し、効果的な授業を行う柔軟さが必要である。

参 考 文 献

- [1] 佐伯 胖ほか「コンピュータのある教室」岩波書店,1996
- [2] MathSource <http://www.mathsource.com>
- [3] ZipIt 入手先 <http://www.maczipit.com>