

中高数学における Mathematica を使った 指導内容と指導事例

沖山 義光

1. はじめに

私は中高生を教えていた頃に、日々実践の結果から指導法の工夫の結果として11個の観点をまとめた。¹⁾

- ①数学史や数学用語の由来などを知らせる。
- ②数学の実用性，社会性を話題にする。
- ③数学の体系などを教養としての知識と考えて教える。
- ④数学の他の分野との関係を意識して指導する。
- ⑤小，中学での既習事項との関連を知らせる。
- ⑥一般的な知識への欲求，美的，合理的，批判的精神を喚起する。
- ⑦数学の厳密性，普遍性を意識して指導する。
- ⑧生徒に問題作りをさせる。
- ⑨作業や実験を通した指導をする。
- ⑩わかりやすい授業や難しいが面白い内容を工夫して達成感を与える。
- ⑪数学的な考え方の良さを強調する。

この観点をもとに前回は、

- ⑧問題作りをさせる。
の実践例として中3生対象にした因数分解の指導実践例を試みた。

今回は、

- ⑨作業や実験を通した指導をする。
を中心に③，⑥，⑩を包含した指導事例を提案する。

2. 授業内容

Mathematicaの命令を各自で入力・実行して、その結果やエラーメッセージに対応して、その意味を探り、自分なりに命令を変えることで命令の意味を発展的に理解を深める。命令をマニュアルから学んだり、教師の指導で教えてもらうのではなく、各自がとにかく命令を実行しその結果からその命令が何を意味しているかを理解していく自学自習による学習をする。これらの作業を通して生徒各自の数学的な考えの深さや多様性が見えてくる。Mathematicaは理工学部の各専門分野でも活用されていて高度な数式処理ソフトであり、その基本的な命令を生徒に紹介しそれを自分なりに進化させていけば各生徒の興味・関心や能力に応じていくらでも研究できる。

指導内容

目標：Mathematica を使って数学の問題を解き明かしてみよう。

1) Mathematica の立ち上げ

① Mathematica 入門を開く。

デスクトップ上の Mathematica 入門テキストをクリックして開く。読んで、実行してみてください。Mathematica が何であるかがわかればよいでしょう。

② Mathematica の立ち上げ。

スタート→プログラム→ Mathematica を

クリックして起動する。

2) プログラムの入力方法と実行

Mathematica の入力画面で名称未定義-1 の枠内にカーソルを移動し、以下にある命令を1つ入力→実行 (**Shift キー + リターンキー** **[SHIFT] [ENT]**) して下さい。Error メッセージが出たらコメントを読みミスを見つけて、訂正しやり直す。

- ・1つ1つの命令がどんなことなのか実際に実行してみて考えて下さい。そして命令の意味が分かったらそのことを自分なりに数値を変えるなど、命令を書き換えて実行してみる。そうするとその命令がどんなことをしてくれるか学べます。失敗しても気にすることはありません。単にエラーが出るだけですからそのメッセージを読めばよい学習になるのです。

3) 保存の方法

最後に名称未定義-1.nb となっているのを自分のファイル名にして保存する。

ファイル→保存→デスクトップにしファイル名を各自書き換え保存する。

注) 拡張子 .nb は自動的に付きます。

プログラム命令の例

①数値計算

5+7

2^100

2 3 4

2/3

N[2/3]

1/3+2/7

N[1/3+2/7]

Sqrt[2]

N[Sqrt[2]]

N[Sqrt[2],20]

N[Pi,1000]

②文字計算

3x-x+2

(a + b)^2/(a + b)

Expand[(x+2y) ^2]

Factor[x^2-5x+6]

Solve[x^2+2x-7= =0,x]

Solve[a x^2+b x + c= =0,x]

代入の仕組み① a=3;a+2;Clear[a];a+2

代入の仕組み② b=x+y;5b+1;Expand[b^3]

真偽判定の仕方 1+2==3;1+2==5;x+y==z

関数定義の仕方 f[x_]:=x^2

f[3];f[a];f[z+1]; (f[b]-f[a]) / (b-a);Simplify[%]

注) セミコロン ; で 1 行に命令を複数書ける。

1 行に分けて入力・実行してみよう。

関数例

Sqrt[x] Exp[x] Log[x] Sin[x] Cos[x]

Tan[x] Abs[x] FactorInteger[x]

③グラフィックス

●プロット (グラフを書く。)

Plot [x^2, {x,-3, 3}]

Plot [2x-1, {x,-4, 4}]

Plot [x^2-4x, {x,-1, 5}]

Plot [{x^2,-x^2+4}, {x,-3, 3}]

オプション

縦横の比

Plot[x^2,{x,-3,3},AspectRatio

→ Automatic]

座標の格子 (グリッド)

Plot[x^2-3x+1,{x,-1,3},GridLines

→ Automatic]

●アニメーション

反復実行

Do [Print [i^2], {i,1,5}]

Do [Print [2k-1], {k,1,5}]

Do [Print [1/m], {m,3,6}]

Do [Print [n!],{n,5}]

Do [Print [x],{x,-1,3,0.5}]

```

Manipulate[Plot[a x^2,{x,-3,3}],{a,-3,3}]
Manipulate[Plot[x^2+b x,{x,-3,3}],{b,-3,3}]
Manipulate[Plot[x^n,{x,-2,2}],{n,1,10}]
●グラフィックスプリミティブ
点、線分、円、多角形
Show[Graphics[Point[{3,2}]]]
Show[Graphics[Point[{3,2}],Axes->True]]
Show[Graphics[Line[{{-1,5},{3,-4}}],
Axes->True]]
Show[Graphics[Circle[{0,0},1]]]
Show[Graphics[Circle[{0,0},1],
AspectRatio->Automatic]]
Show[Graphics[Circle[{0,0},3],
AspectRatio->Automatic, Axes->True]]
Show[Graphics[Polygon[{{0,0},{3,1},{2,4}}]]]
Show[Graphics[{Circle[{1,0},2],Circle[{2,0},3]}]]]
直線の方程式、陰関数
ContourPlot[x+2y-2==0,{x,-4,4},{y,-4,4},
Axes->True]
ContourPlot[y-3==2(x-1),{x,-2,2},{y,-2,2},
Axes->True]
ContourPlot[(x-1)^2+(y-1)^2==2,{x,-1,3},
{y,-1,3},Axes->True]

```

この指導内容は、お茶の水女子大学と附属高等学校との高大連携プログラム「虹の数学」の授業で開発したもので、当時の Mathematica5 によるプログラム命令を最新の Mathematica 9 での命令に対応するように直したものである。²⁾

グラフの定数を変化させて動くように表示することができるようになったり (Manipulate)、陰関数のグラフが表示できるようになった。

生徒に、このプリントを渡しとにかく命令を入力して実行してみなさいというのでいいが、パソコンもプログラムも初めての生徒もいることを考えて、初めは、①数値計算の命令をプレゼンテーションしてもいい。

文字はすべて半角文字を使い、生徒も一緒に 5+7 と入力し、実行 (**Shift キー+リターンキー** (**SHIFT** **ENT**)) すると

12 と画面に表示される。これで何が行われたかは、説明しなくてもわかる。

次に、
2 3 4 と入力する。(2,3,4の間は半角スペースにする。そのことに意味がある。)

実行 (**Shift キー+リターンキー** (**SHIFT** **ENT**)) すると

24 と画面に表示される。ここで、生徒にこれは何を意味するか聞いてみる。2×3×4であることは推測がついてくる。他の数字で各自確かめてみるように指示し確信を得させることができる。

以下同じように、例にある命令を入力・実行し、その意味を推測・確認することで学習を進めていく。

ここで多くの生徒たちは、与えられた命令を次々に入力し実行していくことが多い。昨今は携帯電話や 아이폰、パソコンなどに慣れているので文字入力はとても速い。数値計算からグラフィックス命令まで様々なバリエーションや推測の確認をすると 10 時間は十分にかかる内容であるのに 1 時間で終了という生徒も現れる。これは決していいことではない。単純に文字入力して実行していただくだけでは学びはない。

例えば、初めの
5+7
でも、5+8を入力して確かめる生徒は多くない。さらに 5+6+2などの 2 項演算から 3 項演算への発展も考えてほしい。さらに、引き算はどうするのか。割り算は？かっこを付けるとどうなるか。それらを自分で具体的な数字で確かめる。かっこにも種類がある。これらのことで数

値計算の仕組みや計算能力も上がる。

掛け算は半角スペースという Mathematica 独特の命令なのでこれを2番目の例に出している。さらに、累乗は半角山形[^]なのでこれも例に入れ推測するように工夫してある。

以下の命令も推測はだんだん難しくなるので、例として出された命令をいかに変化させて実行してみるか。そこが数学的な学習になる。また、自分では間違いなく入力したつもりでも、実行してみるとエラーメッセージが出ることも多い。これは Mathematica の命令がとても厳密にできているため、数学でいえば計算ミスをした後はすべておかしくなる厳しさと相通じるものがあると思う。簡単な命令の一文字を変化させてもエラーメッセージになることもあり、それがどうしてかを考えることで理解は深まっていくので、むしろ自分の考えたことがすぐに結果として出てくることを利用して即座に訂正するなどの試行錯誤ができる。その意味ではエラーメッセージを恐れずにむしろエラーメッセージを出すことによって学んでいくように指導すると効果的である。

このようにして、Mathematica の命令を学ぶこと自体でも十分に数学の学習になっている。

さらに、命令がある程度理解できたら、教科書や複雑な問題を解いた後の確認や手計算では難しいグラフの全体像などを描いて解決の糸口を Mathematica で見つけるなどに利用するとい

主な教材例

連立方程式, 方程式

```
Solve[{x+y==1,x+2 y==2},{x,y}]
```

```
Solve[x^3-1==0,x]
```

極限值

```
Limit[Sin[x]/x, x -> 0]
```

微分計算

```
D[Sin[x]*Cos[x],x]
```

積分計算

```
Integrate[Cos[x],[x,0,1]]
```

微分方程式

```
DSolve[y'[x]+y[x]==0,y[x],x]
```

グラフ表示

一変数のグラフ

```
Plot[x+1/x,{x,-2Pi,2Pi}]
```

```
Plot[Log[x^2+1],[x,-2Pi,2Pi]]
```

重ね書き

```
g1=Plot[Sin[x],[x,0,2Pi]]
```

```
g2=Plot[Cos[x],[x,0,2Pi]]
```

```
Show[g1,g2]
```

媒介変数表示

```
ParametricPlot[{Sin[2t],Cos[3t]},{t,0,2Pi}]
```

漸化式

```
fib[0]=fib[1]=1;
```

```
fib[n_]:=fib[n-2]+fib[n-1];
```

```
fib[10]
```

```
Table[fib[i],{i,10}]
```

繰り返し文 1

```
i=1;
```

```
While[{i<5,i++};
```

```
Print[Factor[x^i-1]];]
```

繰り返し文 2

```
i=2;
```

```
Do[
```

```
Print[i];
```

```
Print[Factor[x^i-1]];
```

```
i++;
```

```
If[i>5,Break[ ],{100}]
```

その他

周期関数等の振幅を音に変換する Play 命令、3次元のグラフィックス ContourPlot3D など十分に教材として面白い。

3. 指導事例

前記「指導計画」の最後には While 文や Do 文を紹介した。このように、プログラミング言語的な指導も可能ではある。しかしそのことを

あえて強調するのは Mathematica を使う私の趣旨ではない。Mathematica に計算させるのは基本的に生徒が考え計算するときの補助として使い、できるだけ作業的な操作をすることで主体的に利用することにしたい。そのような趣旨で指導した中学生にもできる指導事例を紹介する。^{3) 4)}

目標：完全数を自分で見つけよう。

6 の約数のうちそれ自身 6 を除いたものを加えると $1+2+3=6$ とそれ自身になる。このような自然数は他にもあるのだろうか。実はその次に大きい数は 28 である。

さらに大きな完全数を求めるには、 k を自然数として、メルセンヌ数 2^k-1 が素数になると $2^{k-1}(2^k-1)$ は完全数になることを使えばいい。このことは高校生であれば証明も可能であるが中学生には定理として伝えることにする。

メルセンヌ数が素数になるかどうかを $k=2, 3, 4, \dots$ と代入して調べていけば完全数は見つけられる。しかし 2 の累乗は k の値で極端に大きくなり手計算では大変である。そこで、Mathematica を使って求めることにする。

命令言語は

PrimeQ[x] : x が素数かどうかを判定する

Divisors[x] : x の約数すべてを出力する
 があれば後は数値計算の簡単な命令で可能である。

以下 Mathematica の Notebook 画面で説明する。

```

数の計算 (1)
(4+2) 3 [SHIFT] [ENT] 18
1/2+1/3 [SHIFT] [ENT]
5/6
2^10 [SHIFT] [ENT]
1024
12! [SHIFT] [ENT]
    
```

```

479001600
数の計算 (2)
GCD[12,16] [SHIFT] [ENT]
4
LCM[12,16] [SHIFT] [ENT]
48
FactorInteger[360] [SHIFT] [ENT]
{{2,3},{3,2},{5,1}}
Divisors[60] [SHIFT] [ENT]
{1,2,3,4,5,6,10,12,15,20,30,60}
Prime[10] [SHIFT] [ENT]
29
PrimeQ[2^13-1] [SHIFT] [ENT]
True
    
```

注) 画面に [SHIFT] [ENT] は出ません。

数の計算(1)では数の四則計算、累乗、階乗計算の演習。数の計算(2)では整数の計算で最大公約数、最小公倍数、素因数分解、ある数のすべての約数の列挙、2から順に並べたとき任意の順番の素数の明示、ある数が素数かどうかの判定などを学ぶ。

```

完全数を見つける . 2^(k-1)(2^k-1)
2^2-1 [SHIFT] [ENT]
3
PrimeQ[2^2-1] [SHIFT] [ENT]
True
Divisors[2^(2-1)(2^2-1)] [SHIFT] [ENT]
{1,2,3,6}
1,2,3,6
1+2+3==6 [SHIFT] [ENT]
True
True となれば完全数!!
2^(2-1)(2^2-1) [SHIFT] [ENT]
6
2^30-1 [SHIFT] [ENT]
1073741823
PrimeQ[2^30-1] [SHIFT] [ENT]
    
```

```

False
Divisors[2^31-1] [SHIFT] [ENT]
{1,2147483647}
PrimeQ[2^31-1] [SHIFT] [ENT]
True [SHIFT] [ENT]
2^30 (2^31-1) [SHIFT] [ENT]
2305843008139952128
Divisors[2^30 (2^31-1) ] [SHIFT] [ENT]
{1,2,4,8,16,32,64,128,256,512,1024,2048,
...途中略...
1152921504069976064,2305843008139952128

1+2+4+8+16+32+64+128+256+512+1024+
2048+4096+8192+16384+32768+65536+
131072+262144+524288+1048576+2097152+
4194304+8388608+16777216+33554432+
67108864+134217728+268435456+536870912
1073741824+2147483647+4294967294+
8589934588+17179869176+34359738352+
68719476704+137438953408+274877906816
+549755813632+1099511627264+
2199023254528+4398046509056+
8796093018112+17592186036224+
35184372072448+70368744144896+
140737488289792+281474976579584+
562949953159168+1125899906318336+
2251799812636672+4503599625273344+
9007199250546688+18014398501093376+
36028797002186752+72057594004373504+
144115188008747008+288230376017494016+
576460752034988032+1152921504069976064
==2305843008139952128 [SHIFT] [ENT]
True
    
```

初めにメルセンヌ数の $k=2$ のとき、それが素数であることをPrimeQ[]で確認し、 $2^{(2-1)}$ (2^2-1)をDivisors[]を用いてすべての約数を列挙し、その数自身以外の約数の和とその数自身を==で結んで [SHIFT] [ENT] とするとTrueとなる。

これで、6が完全数であることが確認できる。次に $k=30$ のときはメルセンヌ数が素数でないこと、 $k=31$ のときには、メルセンヌ数が素数となり確かに 2^{30} ($2^{31}-1$)はその数自身以外の約数の総和とその数自身が等しくなることを確認する。

ここまでを、画面を通してプレゼンテーションしながら生徒は各自のパソコンで実際に操作する。あとは、 $k=3, 4, 5, \dots$ のときメルセンヌ数が素数になるかを調べて完全数を見つける作業を各自で行う。生徒は考え方が理解できるとかなりの速さで発見していきその場で発表し合い楽しくなる。

なお、StringReplace[]を使えば、それ数自身以外の約数の総和とその数自身が等しいことを確かめるための入力は簡単にできる。だが、その前に1つ1つ入力するという作業をすることが理解を深める大事なことである。

また、累乗の計算は意外なほどその数が大きくなり、 $k=100$ などとすると、コンピュータの処理速度から時間がかかり時にはフリーズしたようにみえることもある。このことは、累乗の数の大きさの感覚と逆にコンピュータの処理速度は万能ではないことを知る良い経験となる。

4. 終わりに

指導計画では、Mathematicaの使い方を学ぶということが中心であるがそのこと自体が数学的であることとそれを用いて中高の数学の問題を解決する手助けとして役立つことがねらいである。

例えば、因数定理が使えないような一般の3次方程式はタルタリア・カルダノの解法を用いて解くことは高校生でも可能である。そのような3次方程式を実際に解いたり、または比較的簡単な解になるような3次方程式を作るときなどにMathematicaを手助けに用いると有効である。

そのような指導事例として「完全数を求めてみる」を提案した。

生徒たちが、日々の学習で自分で疑問に思うことを Mathematica を使ってさらに理解を深めより高度で正しい数学に一步でも近づければいいと思う。

参考文献

- 1) 沖山義光：“2006年度公開研究会「数学I」”
お茶の水女子大学附属高等学校研究紀要
(2006)
- 2) 沖山義光・茶園幸子・阿部真由美：“特別
教育プログラム「虹の数学」” お茶の水女
子大学附属高等学校研究紀要 (2005)
- 3) 沖山義光：“第11回中学生向け理数講習会
報告「数学①コース：Mathematicaを使っ
て完全数を見つけよう」” お茶の水女子大学
附属高等学校研究紀要 (2007)
- 4) 沖山義光：“第12回中学生向け理数講習会
報告「数学①コース：Mathematicaを使っ
て完全数を見つけよう」” お茶の水女子大学
附属高等学校研究紀要 (2008)