
* *
* 論 文 *
* *

教材 Web ページの研究開発

墨田区立本所中学校 永井信一

1 はじめに

カリキュラム・学習指導法分科会では、ここ数年コンピュータをはじめとする新しいテクノロジーの活用を前提としたカリキュラム改善の研究¹⁾を進めてきた。また、1999年には政府のミレニアムプロジェクトの1つの柱として「教育の情報化」が発表されている。その内容は2001年度までに、すべての公立小中高等学校でインターネットを利用することができるようにし、すべての公立学校教員がコンピュータの活用の力を身につけられるようにする。さらに2005年度までに、すべての学級のあらゆる授業において教員および生徒がコンピュータを活用できる環境を整備するというものである。教育ソフトの整備として、教科書準拠の動画コンテンツの開発を最重要課題として取り上げ、すべての子どもたちに分かる授業を実現するという目標を掲げている。

このような状況を考えたとき、中学校数学科におけるインターネットの活用を進めるためには教師自身による教材 Web ページの開発が必要であると思われる。本稿では、Mathematica を教材作成ツールとして活用し、新しいスクリプト技術を使ったインタラクティブな教材の開発方法について述べる。

2 数学教育とインターネット

現在行われている中学校の数学教育におけるインターネットの活用では、次のようなものがある。

- ①教師による実践事例，教材，指導案などの公開および情報交換
- ②開発者による教材ソフトの公開と配布
- ③コンクール形式による生徒を対象とした問題の公開，解答の募集
- ④電子メールを使った生徒の意見交換，学校間の交流
- ⑤教材として利用できる Web ページ

教師が教材研究のためにインターネットを利用することは、ここ数年で大きく広がり、数学教育に

関する多くの Web ページが存在し、そのリンク集も充実してきている。教員が作成した教材ソフトもインターネットの利用で以前より入手しやすくなってきている。しかし、授業の中で生徒がインターネットを利用する場面はまだ少ない。特に中学校においては、いわゆる「調べ学習」といった情報収集ツールとしての使われ方がまだ主流である。

数学の授業でコンピュータを利用することは、程度の差はあるがどこの中学校でも可能になった。市販の教材ソフトも Windows が普及してからは選択の幅が広がったが、画面がきれいになったり、キャラクターがしゃべったりするだけで、内容的にはあまり大きな進歩はないように思われる。

そのような中でここ数年、カブリ (Cabri Geometry) などの優れた市販ソフトや愛知教育大の飯島助教授が開発した作図ツール GC (Geometric Constructor)²⁾ など作図ツールを利用した実践に注目が集まっている。Web 上にはカブリや GC で作成された教材や実践事例も数多く存在する。教材 Web ページの開発でもこのような生徒自身の操作が中心となるものも含めていく必要があるだろう。

3 Web ページとして教材開発を行う利点

Web ページとして教材開発を行う利点は次のようなものである。

- ① 比較的簡単に作成することができる。
- ② そのままインターネットで公開できる。
- ③ 校内 LAN のサーバに置けば、ブラウザ上で実行できる。
- ④ テキストファイルなので改変がしやすい。

以前は、教材ソフトの自作というと Basic や C といったプログラミング言語を使ったものが中心であり、MS-DOS から Windows に移行した現在でも Visual Basic などで作成することは可能である。しかし、一般の教師がプログラミング技術を習得することはコンピュータの機能向上と共にかえって困難となっている。一方、インターネットの普及に伴って、Web ページ作成ソフトも使いやすくなり自分のホームページを持っている教師や生徒も少しずつ増えてきている。ホームページを作るような手軽さで教材が作成でき、そのままインターネットで公開することもできるので、遠隔地や通信教育にも利用できる。普通の教材ソフトと違い、コンピュータの台数分だけ用意したりする必要はなく、校内 LAN のサーバに置くだけで実行できるという利点もある。市販の教材や自作の教材でもコンパイルされて実行形式になっているものは改変できないが、Web ページは基本的にはテキストファイルなので授業に合わせて改変することも可能である。

4 教材作成の流れ

① 基本的な部分

教科書の内容をそのまま Web ページとして作成するためには、文章だけでなく、数式、関数のグ

ラフ、平面図形、空間図形などをどう作成するかが重要なポイントとなる。目的に応じて表計算ソフトのグラフ作成機能や図形作成ソフトを使い分けてもよいが、**Mathematica**（図1）を利用することで、必要なものはほとんど作ることができる。さらに、**Mathematica** ではファイルを保存する際に、**HTML** (Hyper Text Markup Language) 形式を選択することができるので、画面のイメージをそのまま **Web** ページにできると考えてよい。ただし、現在は数式部分もグラフィック部分もブロックごとに **GIF**(Graphics Interchange Format) 形式の画像として保存される。

このようにして作成された **GIF** 形式の画像を **Web** ページ作成ソフトを利用して組み合わせることで教材 **Web** ページを作成する。**Windows** に付属する **Front Page Express** でもよいし、最近の文書作成ソフト（ワープロ）は **Web** ページの形式（**HTML**）で保存できるものが多いので、それを利用するのもよい。教科書をただ画面上に表示するだけならこれだけで作成可能である。

```
gr1=Plot[x^2,{x,-3,3},DisplayFunction->Identity];
gr2=Plot[2x+2,{x,-3,3},DisplayFunction->Identity];
gr3=Graphics[Line[{{0,0},{2,4}}]];
Show[{gr1,gr2,gr3},AxesLabel->{"x","y"},
DisplayFunction->$DisplayFunction]
```

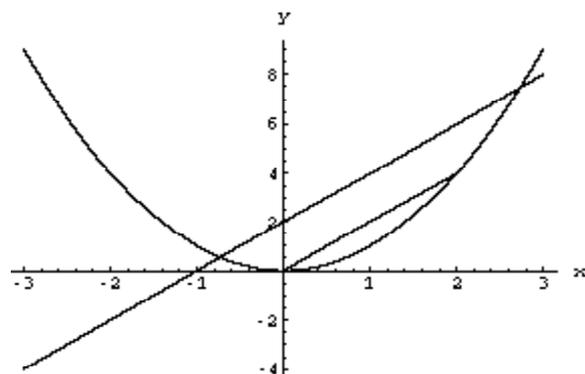


図1 Mathematica のプログラムと出力結果

② リンク機能の利用

HTML のリンク機能（図2）を利用することで、問題部分にヒントや解説機能をつけることができる。ボタンを作成し、リンク先にヒントや解説のページを用意するだけなので、**Web** ページ作成ソフト上で簡単に設定できる。

<p>問題 次の計算をなさい。</p> <p>① $3\sqrt{2} + \sqrt{50}$</p> <p style="text-align: center;"> <input type="button" value="ヒント"/> <input type="button" value="解説"/> </p>	<p><input type="button" value="ヒント"/> をクリックすると、 $\sqrt{50} = \sqrt{25 \times 2}$ を表示するページへジャンプ。</p> <p><input type="button" value="解説"/> をクリックすると $3\sqrt{2} + \sqrt{50} = 3\sqrt{2} + 5\sqrt{2} = 8\sqrt{2}$ を表示するページへジャンプ。</p>
---	---

図2 簡単なリンク機能

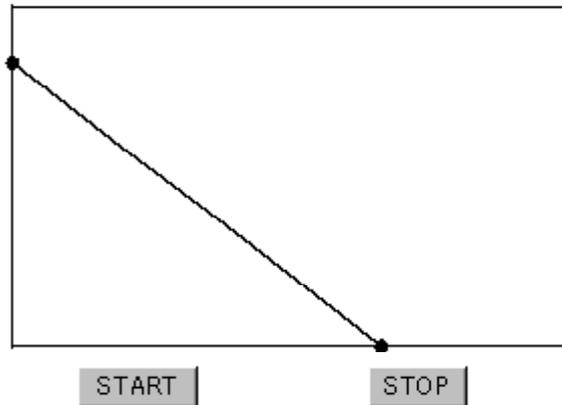
③ JavaScript の利用

動きのあるページを作るには専用のソフトが必要に思えるが、Mathematica で作成した GIF 画像を紙芝居のように表示するだけで十分可能である。GIF 形式の画像には GIF アニメーション (GIF89a-based Animation) という種類があり、これを利用すれば簡単に作成できる。ただし、完成したアニメーションは動きっぱなしで制御することはできない。教材として利用するためには、表示を途中で止めたり、速さを変えたりできる方がよい。そこで、表示をコントロールするためのスクリプト言語が必要となる。現在もっとも利用されているものが JavaScript³⁾ であり、スクリプトは HTML ドキュメントの中にテキストファイルとして埋め込まれ (図3・図4)、ブラウザが解釈して実行する。ほとんどのブラウザで利用でき、インターネットに接続していなくても実行できる。なお、Java と JavaScript は別物であり Java は C++ に似たオブジェクト指向言語で、コンパイルを行い実行ファイル (JavaApplet) を作成して利用するものである。Web プログラミングの主流ではあるが習得は簡単ではない。

JavaApplet の利用については後述する。JavaScript はエディタがあれば作成することができ、ブラウザで実際に動作を確認しながら作業を進めることができるので、初心者向きと言える。

<pre> スクリプト開始 <script language="JavaScript"><!-- var TimeSet=200; ←速さ (ミリ秒) var ImageStart=3; ←最初の画像番号 var ImageMax=53; ←最後の画像番号 var timerID=null; ImageSet=ImageStart; ANIMA=new Array(); ←画像を入れる配列 for(i=ImageStart;i<ImageMax+1;i++){ ANIMA[i]=new Image(); ANIMA[i].src="kadai_1.txtgr"+i+".gif"; } </pre> <p style="text-align: center;">↑ Mathematica で作成した 画像ファイルの名前</p>	<pre> 画像を切り替えて表示する関数 function anime() { clearTimeout(timerID); document.animation.src=ANIMA[ImageSet].src; ImageSet++; if(ImageSet>ImageMax){ ImageSet=ImageStart; } timerID=setTimeout("anime()",TimeSet); } アニメーションを止める関数 function stopanime(){ clearTimeout(timerID); } // --></script>> ←スクリプト終了 </pre>
---	---

図3 GIF 画像を使ってアニメーションをするスクリプト



```

<src="kadai_1.txtgr3.gif" alt="A n i m a t i o n"
  width="288" height="288" name="animation">
</form>
<input type="button" value="start"
  onclick="anime()"> ←クリックで開始
<input type="button" value="stop"
  onclick="stopanime()"> ←クリックで停止
</form>

```

図4 動点の問題 画像と HTML

④ VBScript の利用

VBScript(Visual Basic Scripting Edition)⁴⁾ は基本的に IE (Internet Explorer) のみで使えるスクリプト言語であり、現在のところ他のブラウザでは動作しない。汎用性がないにもかかわらず VBScript を利用するメリットの1つは、作成者がすでに Basic 言語を知っていたり Excel などでもマクロ (VBA) を作成した経験があれば、習得がしやすいことである。このことは、過去に作成した Basic による教材があれば VBScript に書き直すことによってもう一度 Web 教材として甦らせることができるということでもある。実際、グラフィックを多用していなければ比較的簡単に移植することができる。

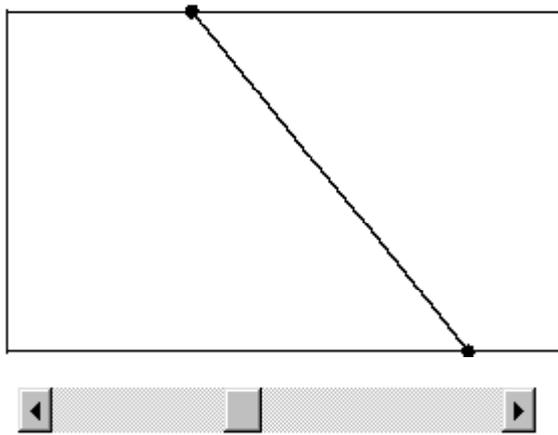
もう1つは ActiveX コントロール (図5・図6) が簡単に利用できることである。ActiveX コントロールとは、アプリケーションにいろいろな機能を追加することができる部品のようなプログラムであり、これを利用することで JavaScript 以上に表現が豊かになる。

```

<object id="Scrollbar1" name="Scrollbar1"
  classid="clsid:DFD181E0-5E2F-11CE-A449-00AA004A803D" align="baseline" border="0"
  width="257" height="22">
<param name="Size" value="5820;582"> <param name="Min" value="3">
<param name="Max" value="53"> <param name="Position" value="3">
<param name="SmallChange" value="1"> <param name="LargeChange" value="5">
</object>

```

図5 ActiveX コントロールのスクロールバーを利用する HTML



```

<script language="VBScript"><!--
Sub Scrollbar1_Change
    Label1.Caption=Scrollbar1.value
    b=Int(Scrollbar1.value)
    Image1.PicturePath="kadai_1.txtgr"&b&".gif"
End Sub
--></script>

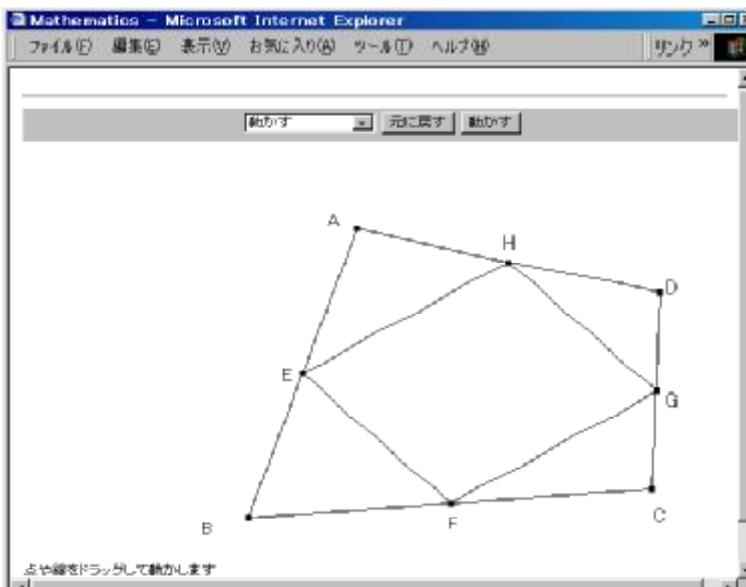
```

スクロールバーと画像ファイルの番号を関連づけることで、自由に動かすことができる。

図6 ActiveX コントロールを利用したスクリプトの例

⑤ 平面図形の教材

平面図形を作成しアニメーションで動きを表現するだけでなく、カブリやGCのような図形の変形をWeb上で実現することも可能となっている。これまではJavaによるプログラミングが必要だったが、JavaApplet にパラメータを渡すことで、簡単に実現するツールが存在する。例えば、平成10年度文部省学習用ソフトウェア研究開発委託事業による OperateMath⁵⁾がある。



```

<applet
    code=classes.OpeMath.class
    name=OpeMath
    width=600
    height=400 >
<param name=fname value="chuu.mth">
</applet>

```

chuu.mth にパラメータが記述されている。

図7 OperateMath で作成したデータをブラウザに読み込んだ状態と HTML

一般の図形作成ツールのように図形を描いてデータを作成し、JavaApplet で利用できる形式で保存する。それを Web ページに組み込む（図7）ことで、ブラウザ上から変形したり、図形を描き加えたりする操作が可能となる。カブリのデータを JavaApplet に変換する Cabri Java Project でもこのようなことが可能である。中学校の図形教育ではこのような自分で操作できる教材が特に有効であると思われる。

⑥ 空間図形の教材

Mathematica ではさまざまな立体を作成することができるが、一度 GIF 形式の画像に出力してしまうと、後から視点を変えて動かしてみることはできなくなる。ブラウザ上で動く 3D グラフィックスを実現する方法の1つは、Mathematica のデータを VRML (Virtual Reality Modeling Language) に変換することである。VRML に対応したプラグイン・アプリケーションを組み込むことでブラウザ上で自由に視点を変えて立体を眺めることができる。

もう1つは、平面図形のとくと同じように JavaApplet にパラメータを渡すことで実現する方法である。ここでは、Live Graphics 3D (Martin Kraus) の例（図8）を紹介する。

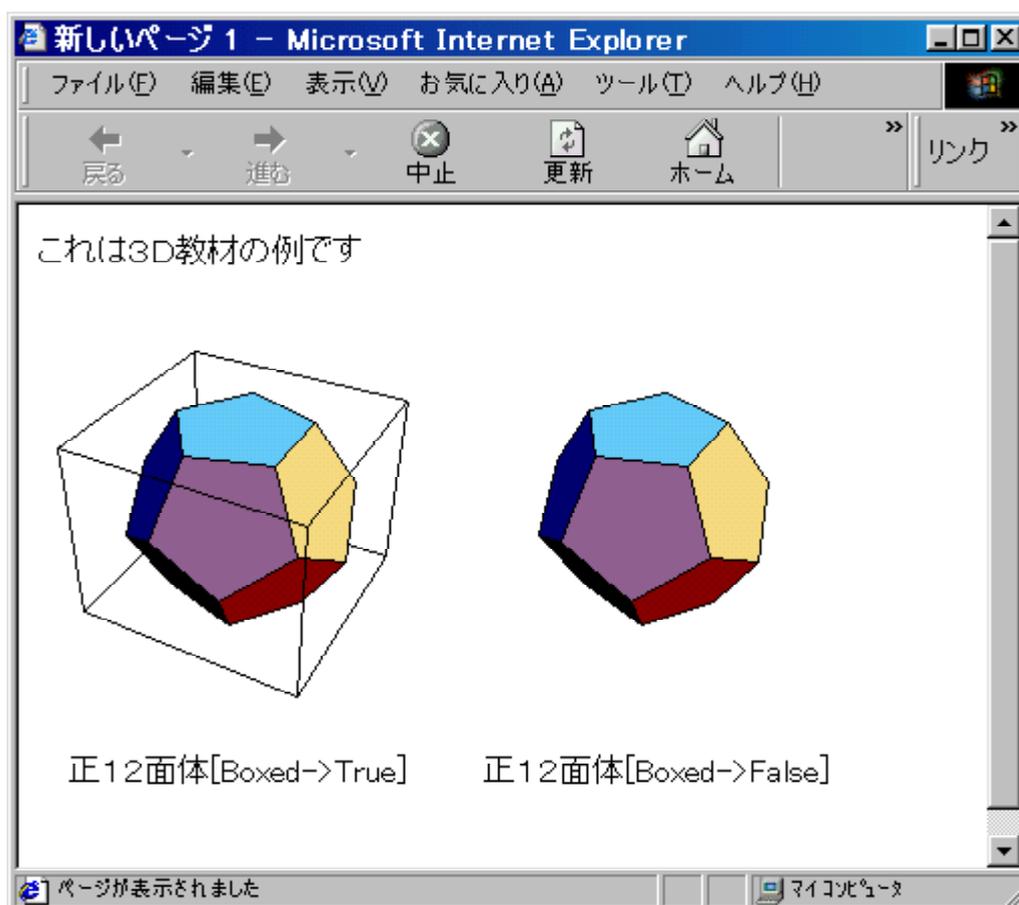


図8 ブラウザに読み込んだ状態

まず Mathematica で立体のデータを作成し (図 9), Input Form で変換する. 次にその出力データ (図 10) を HTML のアプレットタグに埋め込む (図 11) だけである. ファイルとして埋め込んでもよい.

Needs["Graphics`Polyhedra`"]	パッケージのロード
LiveForm[g_]:=NumberForm[InputForm[N[g]],4];	変換する関数の定義
gr=Polyhedron[Dodecahedron];	正 12 面体の作成
LiveForm[gr]	変換
Show[gr,Boxed->False];	画像の表示 (なくてもよい)

図 9 Mathematica で正 12 面体のデータを作成し変換

```
Graphics3D[{
  Polygon[{{0.5257, 0.382, 0.8507}, {-0.2008, 0.618, 0.8507}, {-0.6498, 0., 0.8507},
    {-0.2008, -0.618, 0.8507}, {0.5257, -0.382, 0.8507}}],
  Polygon[{{0.5257, 0.382, 0.8507}, {0.5257, -0.382, 0.8507}, {0.8507, -0.618, 0.2008},
    {1.051, 0., -0.2008}, {0.8507, 0.618, 0.2008}}],
  Polygon[{{0.5257, 0.382, 0.8507}, {0.8507, 0.618, 0.2008}, {0.3249, 1., -0.2008},
    {-0.3249, 1., 0.2008}, {-0.2008, 0.618, 0.8507}}],
  Polygon[{{-0.2008, 0.618, 0.8507}, {-0.3249, 1., 0.2008}, {-0.8507, 0.618, -0.2008},
    {-1.051, 0., 0.2008}, {-0.6498, 0., 0.8507}}],
  Polygon[{{-0.6498, 0., 0.8507}, {-1.051, 0., 0.2008}, {-0.8507, -0.618, -0.2008},
    {-0.3249, -1., 0.2008}, {-0.2008, -0.618, 0.8507}}],

    途中省略

  RenderAll -> True, Shading -> True, SphericalRegion -> False,
  Ticks -> Automatic, ViewCenter -> Automatic,
  ViewPoint -> {1.3, -2.4, 2.}, ViewVertical -> {0., 0., 1.},
  DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction,
  FormatType -> $FormatType, TextStyle -> $TextStyle} ]
```

図 10 作成された Number Form

```

<APPLET CODE="Live.class" WIDTH=200 HEIGHT=200>
<PARAM NAME=INPUT VALUE="
Graphics3D[{
    Polygon[{{0.5257, 0.382, 0.8507}, {-0.2008, 0.618, 0.8507}, {-0.6498, 0.,
        途中省略
    FormatType :- $FormatType, TextStyle :- $TextStyle}]
">
</APPLET>

```

ここに
データを
書き込む

図 11 Form を HTML に埋め込む

5 おわりに

ミレニアムプロジェクトでは、数年後にはすべての教員がコンピュータを使って文書作成や表計算ができるようになることを目指している。近い将来、すべての教員が自ら開発した教材をインターネットで発信することが可能になり、教材のデータベースも整備されるだろう。今回、教材 Web ページの開発についていくつかの手法を述べてきたが、上記以外にも数式表示のための MathML, HTML を拡張し動的表現を可能にする Dynamic HTML などの利用も考えられる。スクリプト技術を利用したインタラクティブな教材 Web ページの作成は始まったばかりで、まだ誰にでも簡単に作成できるというレベルではないが、カリキュラム・学習指導法分科会を中心に開発を進め早稲田大学数学教育学会のホームページで順次公開していきたい。

最後に、この研究を支援していただいた早稲田大学数学教育学会の諸先生方に深く感謝する。また、本研究は早稲田大学数学教育学会課題研究費の助成をうけたものである。

参考文献

- [1] 青木, 井上, 越前, 佐藤, 武沢, 永井, 南, 渡辺: 中学校, 高等学校数学科におけるカリキュラムの改善, 早稲田大学数学教育学会誌, 第17巻, 1999.
- [2] 飯島康之, 磯田正美, 大久保和義: コンピュータで数学授業を変えよう, 明治図書, 1995.
- [3] 河西朝雄: JavaScript入門, 技術評論社, 1996
- [4] 河西朝雄: VBScript入門, 技術評論社, 1997
- [5] 日本教育ソフト研究所: 操作実験ネットワーク OperateMath 平面の図形, 大日本図書, 2000.