

GeoGebraによる説明と例題の交互の提示による スライド教材とその作成の簡易化

明治大学・総合数理学部 原 知己 (Tomoki Hara)
明治大学・総合数理学部 阿原 一志 (Kazushi Ahara)
School of Interdisciplinary Mathematical Sciences,
Meiji University

1 まえがき

GeoGebraは教育用のソフトウェアであり、平面・空間の作図のほか、代数計算(CAS)・表計算の機能を備えている。一般的には動的幾何学ソフトウェア(DGS)に分類される。GeoGebraは教育目的であれば無料で使えるオープンソースソフトウェアである。GeoGebraを用いて作成した教材は、GeoGebraTubeを通してインターネット上に公開することができる。GeoGebraの機能を拡張するために、オリジナルのスクリプト言語のほかにJavaScriptを用いることができるようになってきている。本研究ではGeoGebraの機能を拡張して、二画面を用いた自習教材を提案するとともに、この自習教材を(GeoGebraにあまり慣れていない)教員でも作成できるような支援システムについて提案する。

2 既存の方法・システムとの比較

先述したGeoGebraTubeには、主に海外のユーザが自作したGeoGebraによる数学教材が60万件以上アップロードされている。これらの教材はグラフや図形をアニメーションで表現したものが大半を占め、授業中に教師が生徒に対して提示することを前提として作成されたものであるといえる。しかしながら、これらの教材を生徒が後から自学自習をすることを想起してみると、生徒に向けた説明文は多くの場合十分ではなく、その教材が何を説明をしているかが分からなくなる可能性があると考えられる。これでは教師が生徒に対して一方的に提示するためだけの教材になってしまう。またGeoGebraTubeの教材の多くは単発で存在しており、教材同士の連携は考えられていない。(GeoGebraBookという形で教材をつなげてストーリーを形成することができる枠組みは存在している。)本研究では、1つひとつの教材の中に単元のストーリーがあるように構成でき、生徒が復習するときの理解の助けになるようなものをGeoGebraの単体の教材で提示することができるかどうかについて考察した。

また、GeoGebraの教材はプログラムを組まずに、相当のレベルの教材を作成できるように作られている。実際に、マウス操作だけで点や直線を生成することは可能である。マウスだけで教材を作る時には、作図手順を整理したり編集したりすることは想定していない。図を作成した手順をプログラムのように一覧で表示することができれば、教材の中のストーリーの流れについても一覧したり編集したりしやすくなるのではないかと

考えられる。本研究では、簡素な操作で、作図手順や教材の流れを一覧したり編集できるようなシステムを提案する。

3 システムの概要

3.1 システムの目的

現在の学校の数学の授業では、定義や公式などの説明を行い、それに準ずる例題を解き、答え合わせを行い、そして次の説明に移るといったサイクルを繰り返していることが大半である。授業が教科書に従って行われていて、生徒がその教科書を使って予習したり復習したりするときに、次の3つの問題点が挙げられる。1つ目は、数学を苦手な生徒にとって多くの文章を読み進めることが難しいという点である。このような生徒にとって教科書の長い文章を咀嚼することは難しい。そのことが予習・復習をするときの弊害になりうると考えられる。2つ目は、生徒が理解していなくても予習・復習が先に進めてしまうという点である。教科書を読み進めたとしても、それは理解しているかの確認にはならないし、問題が解けることの確認にもならない。3つ目は、教科書だけでは答え合わせが出来ないという点である。教科書は答えが書かれていないのが普通であり、そのため生徒が予習・復習のために教科書の問題を解いたとしても正しく求められているかどうかを確かめることが出来ない。自分の解答に対するフィードバックがすぐに得られるかどうかで学習効果が変わってしまう。一方で、解答の一覧があったとしても、答えを先に見てしまうと教育効果が薄れてしまうという問題点もある。第1点を解決する方法として、電子教材においては、文章による情報のある程度制限することが必要であると考えた。第2点を解決する方法として、生徒が理解して解けたことを確認しながら先に進めるシステムが必要であると考えた。第3点を解決する方法として、生徒が答えた解答に対して適宜正解しているかを判定して生徒に示してくれるようなシステムが必要であると考えた。

3.2 システムの流れ

基本的には学校の授業形態と同様に説明と例題を交互に提示するのであるが、このシステムでは画面を左右の二つに分け、左側では単元の説明を行い、右側では例題を解くように役割を分けて提示する。そのうえで、左右の窓を順番に注目させる仕組みを提案する。最初に、左側半分に定義や公式に関する説明を提示する。提示する際はスライド再生方式にして情報の制限を行う。生徒は説明を読み進めていき、自分が理解できたと思ったら、例題へ進むボタンを押す。このボタンを押した後は左側半分の表示は固定され今度は右側に例題が提示される。生徒は左側の単元説明を見ながら右側の問題に取り組むことができる。例題は答え合わせができるようになっており、正解できたら、次の例題へ進むようにも設定できるし、次の単元へ進んで左側の説明が更新されるようにも設定できる。このサイクルを繰り返して理解と定着につなげる。

ここでポイントとなるのは、まず、理解を促進するための左側の説明は、例題を解くために必要な内容に特化することができることである。また、例題を解けないうちは次の説明に進めないようにストーリーの流れを制御できるので、生徒に与える情報を制限し安易に先に進めないようにすることで生徒が理解をしなければならない状況を作り出すことができる。

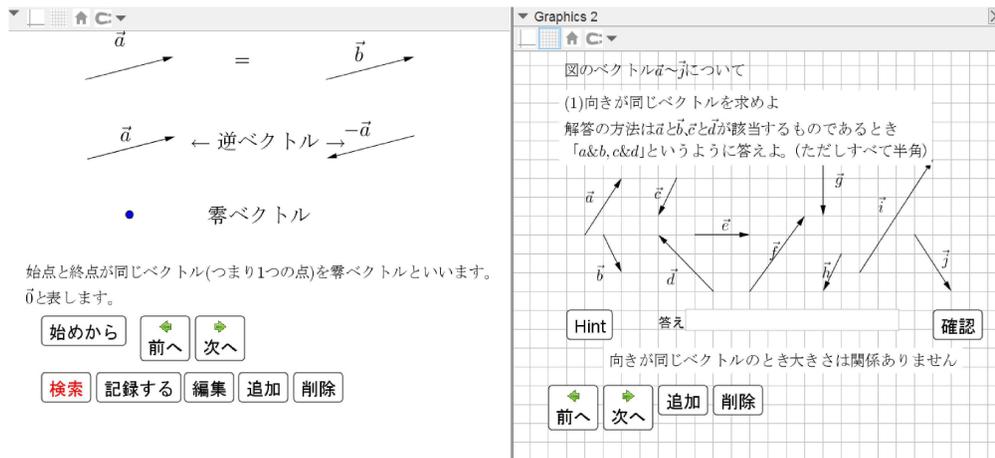


図 1: 画面の様子

4 教材作成を容易にするためのシステム構成

4.1 目的

コンピュータを用いたスライド教材自体は目新しいものではない。生徒が理解しないうちは先に進めないように制御したり、例題の正誤判定ができることを除けば PowerPoint や Pdf を用いても同じことができよう。また、Moodle のように、解答の正誤判定に重点をおくシステムも存在している。しかし、本システムでは、スライドの制御、例題の正誤判定が総合的に行えるものを目指し、その実現のために GeoGebra を採用した。しかし GeoGebra には元々スライドの機能があるわけではなく、GeoGebra のコマンドを用いてひとつの教材の中でスライド表示をしたいときは、スライドの番号を制御する変数を用意し、その値に応じてコンテンツの表示・非表示の条件分岐を設定しなければならない。

しかしこの方法ではスライドシートを追加したり削除したりしたいときに、スライドの番号を1つひとつ訂正しなくてはならず非常に面倒な作業が必要である。表示されるオブジェクトがどのスライドで表示されるのかという管理も容易ではない。

こういったスライド表示の制御は GeoGebra のスクリプトを用いても作成可能であるだろうがスクリプト関数が限られているため柔軟な作りにならない。一方 JavaScript を用いれば比較的容易に作成することは可能だが、JavaScript の高度な知識が必要となり、

限られた人しか作成できない恐れがある。このような状況を踏まえ、スライドシートの追加や削除が簡単に行えて、教材全体の管理がしやすく、プログラムの知識のない教員でも一定の成果が得られる教材を作成できるようなシステムの設計を目指した。

4.2 方法

教材の作成を簡易化するため、極力作成者が入力する手間を省けるような設計にする。PowerPointのように部品を並べて直感的に画面構成ができるようにすると同時に、教材としての流れが可視化されるような作りをすることが重要である。ここで、教材を作成する人が行うべき作業を以下の3つにまとめてみた。

スライド時に提示する文章の作成

教材の作成者は左半分の画面で行う説明のための文章を作る。作成者はテキストボックスを1つひとつ追加するのではなく、システムによって準備されたテキストボックスに説明文を書き込めばよい仕組みにしてあるので、PowerPointのテキストボックスに似た感覚で文章の作成・編集ができるようになっている。

スライド時に提示するオブジェクトの作成

作成者は説明や例題のスライドで表示する点や関数、ベクトルなどのオブジェクトを作成する。作成したオブジェクトは自動で表示・非表示の設定がされるため、条件分岐の設定は必要がない。作成者は左側の画面の説明に必要な挿絵や図、また右側の画面の説明に必要な挿絵や図を、GeoGebra本来の機能（点・直線・ベクトル・関数のグラフなど）を用いて作成する。

例題の正誤判定に用いるための関数の設定と答えの入力

右側の画面に表示される例題のスライドでは生徒が回答を入力できるような枠が準備されている。回答が正しかった場合には次の例題や次の単元へと進めるようになっており、回答が誤っている場合には、正解するまで終了することができないようになっている。

生徒が入力した答えに対しての正誤判定のために、判定する方法と正解とを登録しておく必要がある。判定する方法はあらかじめGlobal JavaScriptと呼ばれる場所に関数としてプログラムを組んでおくことができるようにしてある。また正誤判定をするボタンにも3行程度のJavaScriptプログラムが組み込み済みである。このような仕組みのもとで、教材を作る教員はJavaScriptによるプログラムを組む必要がないように工夫されている。

教材作成者が入力したテキスト、点や直線などのオブジェクト、関数の名前、解答などは、GeoGebraに備わっている表計算画面を用いて一元的に管理を行うように作られている。図2がGeoGebraの表計算画面であり、教材に含まれている情報を格納している様子である。A列がテキスト、B列が正誤判定を行う関数の名前、C列が正誤判定をする際の答え、D列とE列が表示するオブジェクトを格納する列になっている。A列の「sentence」、「blank」、「question」はそれぞれ特別の意味を持っている。「sentence」か

ら下のテキストは説明の画面（左側）に表示される。「question」から下のテキストは例題の画面（右側）に表示される。「blank」は1枚以上のスライドに表示されるオブジェクトを非表示にして、新しいスライドを作るためのものである。A~E列の内A,D,E列はGraphicsの画面から編集が行える。ただしB,C列の内容は今のところ表機能の画面から編集を行う。

表計算の機能を使って教材の管理・編集をするように設計したが、いくつか問題点が残されている。最も顕著な問題は、表計算画面における文字列の取り扱い方である。表計算においてセルをダブルクリックすれば編集可能になる。数値の場合はそのまま数字を書き込めばよいが、文字列への編集を行う場合にはダブルクォーテーションで囲まなければならない。ダブルクォーテーションなしの文字列は変数やオブジェクトの名前であると認識されてしまう。GeoGebraを詳しく知らない人にとってこの点は非常に分かりづらい。今後の課題として、教材作成者になるべく表の画面を編集しなくてもよいような仕組みに改良を加えていくつもりである。

	A	B	C	D	E
1	sentence	function	Answer	以降継続	このスライドのみ
2	ベクトルとは向きと大きさを持...			u	
3	ベクトルの根元を始点、先を終...			text2,text3	
4	ベクトルは基本的に点と点を結...			text1,text4	
5	このときこのベクトルを \vec{AB} (A			text5	
6	また1つの文字で \vec{a} と表したり			text6	
7	\vec{a} の大きさは $ \vec{a} $ と表します。			text7,c	
8	さらに大きさが1のベクトルを...			text8,v,text30,p	
9	blank				
10	2つのベクトルが等しいという 向きが同じで大きさが等しいと				
11	\vec{a} と \vec{b} は向きと大きさが同じなの このとき $\vec{a} = \vec{b}$ と表します。			$w, w_1, \text{text10}, \text{text11}, \text{text12}$	
12	\vec{a} と大きさが同じで向きが反対の $-\vec{a}$ と表します。			$w_3, \text{text15}, \text{text13}, \text{text14}, w_2$	
13	始点と終点が同じベクトル(つま...			H,text16	
14	question				
15	向きが同じベクトルのとき大き...	Set	a&i,c&h	text17,a,b,d,e,f,g,h,i,j,k,text19,t...	
16	等しいベクトルとは向きも大き...	Set	a&j,e&g,b&c&h		
17	等しいベクトルとは向きも大き...				

図 2: 表計算画面

4.2.1 作成済みの正誤判定を行う関数

現在、作成済みの関数は下の2つある。

1. Set 関数

これはあるものとあるものが対応するものであるとき正解と判定するための関数である。例えば \vec{a} 、 \vec{b} 、 \vec{c} 、 \vec{d} という4つのベクトルがあるとする。ただし \vec{a} と \vec{b} 、 \vec{c} と \vec{d} は向きが同じとする。ここで向きが同じものを答えるという問題を与えたとする。このとき \vec{a} と \vec{b} 、 \vec{c} と \vec{d} がそれぞれ対応するものなので「a&b,c&d」と答え

させる。このように対応するもの同士は「&」でつなぎ、異なる対応するものは「,」で区切る。ただし対応するものだから順番が入れ替わっても正解にならなければならない。このルールに則っていれば「b&a,c&d」や「c&d,a&b」でも正解になる。現状では2個の対応に関しては正しい正誤判定が行える。3個以上の正誤判定に関しては正しい正誤判定が行えていないので今後、改善していく予定である。

2. ObjectEqual 関数

これは2つのオブジェクトが同一であるかを判定するための関数である。2つのオブジェクトが等しい状態にあるとき正解と判定する。その具体例を下に示す。

- (a) 二次関数：2つの関数が重なれば正解
- (b) 点、円、直線：2つが重なれば正解
- (c) ベクトル：2つのベクトルの成分が同じであれば正解。従って2つのベクトルが重なる必要はない。
- (d) 線分：長さが等しければ正解
- (e) 三角形：面積が等しければ正解。従って合同でなくても正解となる。
- (f) 角度：大きさが同じであれば正解

このように文字列だけでなくオブジェクトによる正誤判定も行える。これが GeoGebra の得意とするところであることに加え、生徒にコンピュータを用いて新しい体験を与えられる可能性がある。1つ問題なことが三角形の同一であるかの判定が面積によって行われているという点である。これでは合同な三角形でなくても正解と判定されてしまう。作成者が求める正解を生徒が正確に答えられるように改善が必要な点はあるが GeoGebra としての利点はあると考えている。

今現在、2つの関数を作成している。これらは Global JavaScript という画面に関数としてまとめて記述する。仮に人が作った関数を使いたい場合は Global JavaScript にコピー&ペーストすればよい。作った関数を共有するためのコミュニティを作るとより良い教材作成が行えると考えている。

5 システムの利点

1. スライドにより生徒に与える情報を制限することが出来る
スライド方式を採ることにより生徒が大量の情報を分割して読むことが出来る。またスライドが理解しないと先に進めないように制限することにより生徒が意識して理解に努めようとすると考えている。
2. 説明と例題を同時に見ることが出来る
説明と例題を1つの画面で表示する場合、例題に取り組んでいるとき説明を見たい場合、前に戻ると問題は非表示になってしまう。それは取り組みにくい原因になる。そのため説明と例題を互いに見ながら行えるよう2画面にしている。

3. 単発である GeoGebra 教材を 1 つにまとめることが出来る

先述した GeoGebraTube には多くの教材がアップロードされている。それらは学校の授業内において提示することが前提として作られているであろうものが大半を占める。授業というストーリーの中で提示される分には生徒に教材の意味が伝わると思うが改めて生徒が復習をしようとするときはその教材が一体何を表すものなのかが分からなくなってしまう可能性がある。そのためスライドの中でストーリーを作りながら提示することが復習をする上では効果的だと考えた。また教材を 1 つにまとめることによりファイルの数を減らすことが出来るという利点もある。

4. 追加、削除が容易

条件分岐により表示・非表示を設定するのではなく、表によって表示・非表示を設定するので追加や削除が非常に容易になる。条件分岐の場合は追加や削除をするたび数値を変えなければならないが表は変える必要がない。そのため作る前に計画を立てる必要性が減り、自由な教材作成が行える。

5. 管理が単純

表により視覚的にテキストやオブジェクトの管理が行える。どのオブジェクトがどのスライドで表示されるかが一目で分かり、表という場所に情報を制限することにより時間が経って改めて編集したい場合も分かりやすく管理が単純である。

6. プログラムを組む必要がない

本システムは GeoGebra をもとに JavaScript を用いてプログラムを組んでいる。同じものを作ろうとした場合、JavaScript を使わなければならないが GeoGebra ユーザーは JavaScript を使うことはあまりない。むしろ使い方が分からないというユーザーの方が多いと感じている。また JavaScript 自体の文法やプログラミングの技術がなければならない。それを現場の教師が一から学ぶというのは時間的に厳しいものがある。しかしながら本システムではほとんどの場面でプログラムを組む必要がない。先述したように作成者が行うことはテキストやオブジェクトを作るだけである。正誤判定を行うための関数で新しいものが必要な場合は自らプログラムを組まなければならないが、他の人が作ったコードをそのまま持ってくるのが可能である。従ってオブジェクトの表示・非表示やスライドの制御のプログラムを組む必要がなく作ることが可能である。

6 システムの欠点

GeoGebra のアニメーションに関してはある程度の知識を必要とする点が欠点である。本システムは全く知識がなくても出来るというのではなくある程度は GeoGebra に関して知っていることを前提として作っている。また元々定義されている変数は使うことが出来ない。作成者が同じ名前で作ると上書きされて前の情報は消えてしまう。そうなるプログラム自体が壊れてしまう可能性がある。これと同様に生徒がプログラムを壊すことが出来てしまう。スライドを制御する変数を書き換えられれば解けなくて

も先に進むことが出来てしまう。現状では生徒の善意に任せておくことしかできない。こちらの意図しない変更に関しては何らかのフィードバックを返さなくてはならないと考えている。

7 今後の展望

教材作成者にとっての操作をより簡略化していこうと考えている。教材作成のための操作がまだまだ複雑なところが散見され、教材を幅広く作成してもらう障害になってしまう可能性がある。スライド教材を作る上で便利なものという認識に作成者になるようユーザーインターフェースに改良を加えていく予定である。また自分以外の人に使用してもらい具体的な改善点も把握していきたいと考えている。

8 研究で得られた知見・結論

GeoGebra はプログラミング言語としてあるわけではなく簡単に数学の教材を作るためのものである。その上で補助のための JavaScript であった。確かに GeoGebra は簡単にアニメーションなどを作ることが出来る。加えて JavaScript を使えばもっと充実した機能を持つ教材を作ることが可能である。しかし JavaScript を用いることはより難易度を高め、敬遠される要因になってしまう。本研究で JavaScript を用いながらそれを意識せず誰もが同じように教材が作れることが提案できたと考えている。様々な教師が同じように簡単に教材を作れることが教材としての質も上がっていくことにつながっていくと思っている。

参考文献

- [1] GeoGebra の JavaScript に関するリファレンス
<https://wiki.geogebra.org/en/Reference:JavaScript>
- [2] GeoGebratube <https://tube.geogebra.org/>
- [3] 教材の掲載サイト
<http://www.aharalab.sakura.ne.jp/geogebra/index.php>