

スマートフォンでの入力に配慮した反転学習用ランダム問題 の開発

吉富 賢太郎

大阪府立大学 高等教育推進機構

On development of randomized multichoice quizzes for using with mobile phones

Kentaro Yoshitomi

Faculty of Liberal Arts and Sciences, Osaka Prefecture University

大学における専門基礎数学の授業において、反転授業の取り組みはまだ始まったばかりである。これまで筆者は短時間の動画を開発してきたが、視聴確認テストが対面授業設計には重要である。旧来は Moodle 上 CAS ベースで動作する STACK を主に用いて行ってきたが、行列の成分入力等煩雑さがあり、学生の負担となっている。一方、多肢選択問題は知識習得を目的とした場合に多用されてきた問題形式で概念理解を中心とする数学においては有効性に疑問があった。しかし、スマートホンを常用する学生向けにとって、入力が平易であり、CAS を用いて大量のランダム化された数値データを利用した多肢選択問題を Moodle のランダム出題機能を用いて利用する方法の有効性に着目するようになった。開発のポイントと活用事例、今後の展望について報告する。

キーワード： 数学教育, LMS, 多肢選択問題, CAS

1. はじめに

1.1 動機と背景

線形代数や微積分学のような大学における専門基礎数学科目において反転授業・反転学習の事例はまだ極めて少ないと言ってよい。線形代数については、特に例えば本学では後期に展開されるような抽象的な内容に関しては、一層実施が困難なのが実情である。原因として、数学が知識習得よりも主に概念理解を中心とする科目であるという特異性が影響していると考えられる。

このような状況にあって、反転授業を実施するには、動画教材の視聴や対面授業の前後において学生がさまざまな概念をどの程度理解しているかを把握することが肝要である。数学理解の認知科学的な解釈は困難な課題であり、筆者は明るくないが、多角的にさまざまな表現で問い掛けを行うことにより理解の程度を把握することは可能と考える。

例えば、基底の定義を理解しているかはさまざまな部分空間の基底を解答させることにより把握できる。計算が必要な場合もあるが、数値的にランダム化されて

いても計算不要でかつ思考が必要とされる問題は開発可能である。また、次元と組み合わせれば次元の概念の理解を確認することができ、数ベクトル空間の次数と混同していないかを確認することもできる。

このような理解度を問うテストをさまざまなタイミングで実施するには1週間毎に実施する授業内の紙ベースの課題やテストでは間に合わない。特に動画視聴確認の場合、授業冒頭で回収するプリントで行っていたのでは授業設計上も間に合わないのである。したがって、授業時間外学習を ICT を活用して行う他にすべはない。

1.2 数学教育における ICT 利用

上述のように ICT の活用は動画視聴確認の他、木目細かい理解度確認が必要な場合には必須であるが、もちろん、授業内での実施も視野にいれた成績評価を目的とする小テスト、理解度促進のための自宅外学習、クラス規模により学力差が大きい場合の補習課題などさまざまな用途も考えられる。

このように活用範囲の広い ICT であるが、実際に使われている具体的事例としては Moodle 等の LMS や

また、CAS の 1 つである Maxima を利用する Moodle の問題プラグインである STACK⁽¹⁾, 筆者の所属する大阪府立大学で運用している MATH ON WEB⁽²⁾や開発中の Moodle 用の MATH ON WEB プラグイン、また、世界的には M'obius courseware(旧 Maple T.A.)⁽³⁾, WeBWork⁽⁴⁾, Numbas⁽⁵⁾などがある。これら CAS を用いたシステムでのテストや学習は数学教育において特に重要な役割を果たす。

CAS を用いたシステムが必要である 1 つの理由は、ランダムに与えられた数値に対する答は (数学はよく答が 1 つと言われるのに反して) 一般には 1 つではなく、CAS を含め何らかのプログラミング言語を用いたアルゴリズムによってしか採点できない問題が多数あるからである。

1.3 CAS 利用の場合の問題点

一方で上述のようなシステムでは解答入力に骨が折れるという問題点も発生している。これは、昨今学生が主にスマートフォンを利用してこのような ICT を利用する、という状況に起因する。

1 つには、PC や Mac を主として利用する時代に比べて半角・全角の区別ができない (区別があることを知らないと思われる) 学生が多い、ということである。殆どの場合、情報リテラシー教育によって、1 年後半から 2 年次以降には解消されることが期待されるが、工学のような理系でもこの問題が発生した。

もう 1 つは、スマートフォン特有のキーボードにある。記号が出しづらいこともあり、数式の入力が困難な場合がある。x³+3*x+1 くらいならば問題ないが、((x+2)²*(x-1))/(x⁴-3*x³-2*x) のような長い式になると学生に負担となる。実際、文系クラスで図 1 の課題を出したところ、「こんな式の入力をさせて、答が正しいのに入力ミスで間違いなるような問題は認められない。」という強硬な異論がメールで送られてきた事例もある。

また、行列の入力にも問題があり、例えば PC や Mac ならば Tab キーを用いて入力欄の移動が容易だが、スマートフォンの場合は実際にやってみて非常に煩雑な作業であった。行列の成分が 20 個あるような問題を 3~5 題やらせるとそれだけで負担である。このことは 2017 年度の学生のアンケートから判明しており、実際に実験して確認した。

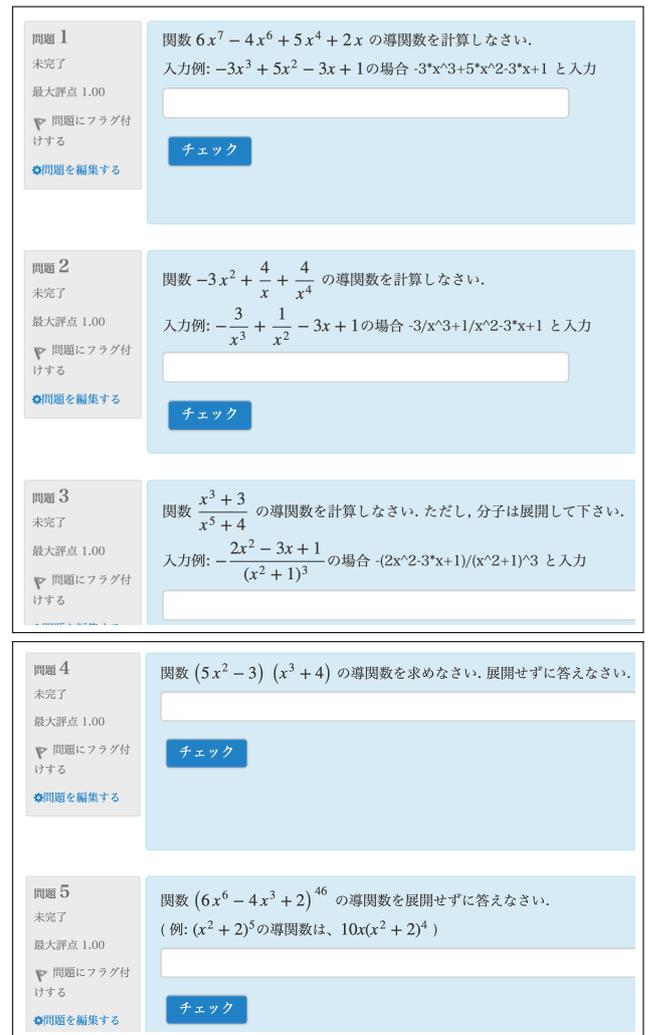


図 1 STACK の例 (基礎数学)

なお、STACK では FlickMATH⁽⁷⁾ や MathTOUCH⁽⁸⁾ という優れた入力支援の研究もあるが、まだ、一般的にシステムに採用されているわけではない上、行列の入力などで苦労する点はやはり問題となる。

このように、数学の問題を作るには非常に効果的な CAS ベースのシステムであるが、スマートフォンを日常的に道具として使用するようになった学生には入力という点で困難があり負担となる点は否めなくなってきたのである。

1.4 多肢選択問題の利点と欠点

一方、入力が平易な問題としては正誤問題や多肢選択問題があげられる。多肢選択問題は単にチェックして送信すれば解答できる問題形式であり、解答形式としては最も簡便なものであることは言うまでもない。したがって、上述のような評価や学習目的で数多くの問題を学生に解かせる場合には適した問題形式と言える。

多肢選択問題はその活用用途として、知識習得が中心で、一度解いてしまえば覚えてしまい、繰り返し利用

するのはあまり意味がないというイメージがある。

しかし、実際には単に正解はどれか、という知識のみを問うだけではなく、知識ベースの推論やヒヤリング等の言語テストでもよく用いられる。

ただし、従来数学の ICT 活用課題の形式としてはあまり使用されてこなかったと言える。一つには数学では知識よりも理解を重視する点が原因と考えられる。知識として問われる内容は非常に少ないので、固定化された問題では、教え合いや繰り返しにより簡単に覚えてしまい、ICT を活用する意味はほとんどないと言える。また、答が1つでない問題もあり、そのような問題に対して、選択肢をどのように作るかという問題は、大きな障壁となったのではないかと考えられる。

しかしながら、毎回異なる数値に対して正しい認識や理解がなければ正解を選択できないような(大量のパターンからなる)多肢選択問題を作ることができれば多肢選択問題形式も数学において活用できる可能性があると考えられる。さらに、語学テストと異なり、数学の場合は乱数をうまく用いることにより比較的所謂錯乱肢が作り易いという利点もある。

一般に Moodle 等の LMS における多肢選択問題では、高度なプログラミングを用いてランダムな問題を生成し、多様なパターンの選択肢を提示する機能はない。

しかし、カテゴリ内にある相当数の問題群から指定された問題数の問題をランダムに選択して提示する機能(ランダム出題機能)はある。したがって、ランダム化された多肢選択問題を潤沢に(例えば 100 題単位で)用意し、ランダム出題機能を用いて利用すればランダムな多肢選択問題が実現できる。

動画の視聴確認や授業時間外で日々の学習用途に多肢選択問題でも十分利用できるというのが本研究での着想である。特に計算が必要な問題というよりも計算が不要もしくは暗算程度しか要求されない、明確な概念理解が要求されるような問題が理想的である。

このような問題の開発と利用について、本稿では紹介したい。

2. 多肢選択問題

2.1 構造

Moodle における問題データはすべて XML 形式でやりとりすることができる。構造はシンプルで、XML を自

- 単一選択 (A 形式) か複数選択 (XX 形式) か。
- 問題文
- 正解とそれに対するフィードバックのリスト
- 誤答とそれに対するフィードバックのリスト
- 正答と誤答に対する配点 (%)

図2 多肢選択問題の構造

動生成することは比較的容易である。なお、多肢選択問題に必要なデータは図2の通りである。

他にいくつかオプションはあるが、「選択肢の並びをシャッフルするか?」などのフラグであり、基本的に固定値で運用して問題ない。

これらのデータを基に XML を生成し保存する基本関数群と線形代数など科目毎に多用する関数群をライブラリ化しておけば、XML の生成は比較的容易である。

2.2 開発方針

技術的には平易であることは前述の通りであるが、実際の問題の開発においては注意すべき点もある。

問題の開発方針として重要なポイントは「どのような誤答を用意するか」である。明らかに誤答とわかる選択肢を入れても意味がない。学生が自分の理解の誤りに気付くような問題が望ましい。

また、パターンを覚えてしまうような選択肢も論外である。見た目はほとんど同じだが、論理的な思考により排除することが要求される選択肢を設置すべきである。用語の用法の誤りなどはパターン化してしまうおそれがあるが正しいフィードバックと数値パターンとの組み合わせによって学習者に気付きを促すことは十分可能であろうと考える。

以上のような気付きや学びを誘発するような誤答選択肢は実際に学生が間違いやすそうな観点に基づいて生成されたものが適切であろう。

すなわち、過去の紙での演習やテスト・課題が記録された PDF に基づき、そこでの誤答を分析して、それらを基にした誤選択肢を生成することが重要である。

また、このような方法を用いれば、例えば数学では ICT で問にくいと考えられるような論述式の答案であってもよくある誤答や記号・用語の誤用・乱用例を織り交ぜて誤答パターンを作れば、学生がそれらを見比べ違いを分析し検討することによって、学習効果が上がることが期待される。このように学習した上で、紙のテストを実施したり、より高度な対面授業設計を行

1. 行列とベクトルの積を逆順に書いてしまう。
2. 微分なのに積分してしまう, または積分と混同してしまう。
3. 合成関数 $f(g(x))$ の微分を $f'(g(x))$ としてしまう。
4. 三角関数の微分の符号を間違える。
5. 積 $f(x)g(x)$ の微分を $f'(x)g'(x)$ とする。

図3 基礎数学における誤答例

1. 固有多項式の実数が行列のサイズと一致していないのに気づかない。
2. 対角成分の和と固有値が等しいという必要条件を用いたエラー判定を使わず固有値の計算をミスする。
3. 固有値の計算ミスや基本変形の違いにより $A - \lambda E$ を基本変形した結果正則になる。
4. 固有空間の次元が重複度と一致しないにもかかわらず対角化不可であることに気付かない。
5. 対角化判定のみ要求された問題で重複度 1 の固有空間の次元を調べようとする。
6. 重複度が 1 より大きい固有値があるだけで対角化不可能と判定してしまう。
7. 上のようなミスに気付かず, 基底をならべた正則行列 P の列ベクトルとして同じベクトルを並べたり, 0 ベクトルを補って強引に正方行列にする, 正方行列でないものをそのまま答える。
8. 行列 A のサイズを rank A と表記してしまう。
9. 固有空間 V_λ の次元を $\dim(A - \lambda E)$ と表記する。

図4 対角化判定(線形代数)でのさまざまな誤答要因

なえば, 学生の論述でよく散見されるような不正確な用語使用例や不正確な議論が改善されたり, 学生の理解をより促進させることができると期待できる。

誤りの例として, 基礎数学の場合によく見られる誤答パターンを図3にあげた。また, 線形代数における対角化判定の問題における誤答の例を図4にあげた。このように多様な誤りが学生の答案には見られるが, 誤りパターンは例年だいたいこのような分類の範囲に収まるのも事実である。

これらをうまく段階を追って分類し, 誤答のパターンを組み合わせることにより, 適切な数のパターンを生成すれば, 学生がそこから正解を選ぶときに自分のミスに気付く可能性があると考えられる。

ただし, 正解が複数ある問題の場合は, 選択肢があまり多いと再び学生の負担となってしまうので, 誤答の数もある程度抑える必要がある。そのためには, 段階を追って問題を細分化する工夫が必要であろう。

2.3 開発の実際

現在, 開発は *Mathematica* を用いて行っている。

```
GenerateMCxml[Title_ , Single_ , Qhtml_ , CorrectList_ , IncorrectList_ , IncorrectPer_] :=
Module[{PERLIST = {"100", "50", "33.33333", "25", "20",
"16.66666", "14.28571", "12.5", "11.11111", "10"},
CORRECTPER, INCORRECTPER, SingleTF, ct, Cnum = Length[CorrectList],
ICnum = Length[IncorrectList], Tmpl1 = "", Per, Pat, Fdb, CIL, CILen, Ret},
(* If[Cnum=1, SingleTF=true, SingleTF=Single]; *)
SingleTF = Single;
If[SingleTF == "true", CORRECTPER = PERLIST[[1]], CORRECTPER = PERLIST[[Cnum]];
INCORRECTPER = "" -> PERLIST[[2]]; (* always -50 *)
Ret = " <!-- question: " <> ToString[RandomInteger[9 000 000] + 1 000 000] <> " -->\n";
Ret = Ret <> " <question type=\"multichoice\">\n";
Ret = Ret <> " <name>\n <text>" <> Title <> "\n </name>\n";
Ret = Ret <> " <questiontext format=\"html\">\n";
Ret = Ret <> " <text><![CDATA[<p>" <> Qhtml <> "\n"]></text>\n";
Ret = Ret <> " </questiontext>\n";
Ret = Ret <> " <generalfeedback format=\"html\">\n";
Ret = Ret <> " <text></text>\n </generalfeedback>\n";
Ret = Ret <> " <defaultgrade>1.000000</defaultgrade>\n";
Ret = Ret <> " <penalty>0.333333</penalty>\n <hidden>0</hidden>\n";
Ret = Ret <> " <single>" <> SingleTF <> "\n";
Ret = Ret <> " <shuffleanswers>true</shuffleanswers>\n"; (* Shuffle default true *)
Ret = Ret <> " <answernumbering>none</answernumbering>\n"; (* no numbering default *)
Ret = Ret <> " <correctfeedback format=\"html\">\n";
Ret = Ret <> " <text></text>\n </correctfeedback>\n";
Ret = Ret <> " <partiallycorrectfeedback format=\"html\">\n";
Ret = Ret <> " <text></text>\n </partiallycorrectfeedback>\n";
Ret = Ret <> " <incorrectfeedback format=\"html\">\n";
Ret = Ret <> " <text></text>\n </incorrectfeedback>\n";
Ret = Ret <> " <shownumcorrect>\n";

Tmpl1 = Tmpl1 <> " <answer fraction=\"__Per__" format=\"html\">\n";
Tmpl1 = Tmpl1 <> " <text><![CDATA[<p>__Pat__</p>]]></text>\n";
Tmpl1 = Tmpl1 <> " <feedback format=\"html\">\n";
Tmpl1 = Tmpl1 <> " <text><![CDATA[<p>__Fdb__</p>]]></text>\n";
Tmpl1 = Tmpl1 <> " </feedback>\n";
Tmpl1 = Tmpl1 <> " </answer>\n";
(* Print["Debug: Cnum=", Cnum, ", ICnum=", ICnum]; *)
For[ct = 1, ct <= Cnum + ICnum, ct++,
Ret = StringJoin[Ret,
If[ct <= Cnum, Per = CORRECTPER; CIL = CorrectList[[ct]],
Per = INCORRECTPER; CIL = IncorrectList[[ct - Cnum]];
If[ListQ[CIL], Pat = CIL[[1]]; Fdb = CIL[[2]]; Pat = CIL; Fdb = ""];
StringReplace[Tmpl1, {
"__Per__" -> Per, "__Pat__" -> Pat, "__Fdb__" -> Fdb}]];
Ret = Ret <> " </question>\n";
Return[Ret];
```

図5 プログラム例 (XML生成ルーチン 1)

```
GenerateMCxml[Fname_ (* save filename *) , Cat_ (* category string *) ,
Title_ (* title string (auto numbering) *) ,
Num_ (* Number of questions *) , Single_ (* "true" or "false" *) ,
Qtext_ (* Question text string or Generate Qtext function *) ,
CListFunc_ (* Function name which returns {Correct Answer1, CA2, ...}
or {{CA1, Feedback1}, {CA2, Fdb2}, ...} s *) ,
ICListFunc_ (* Function name which returns {Incorrect Answer1, IA2, ...}
or {{IA1, Fdb1}, {IA2, Fdb2}, ...} s *) ] :=
Module[{ct, i, j, ij, EM, mm, dt, mmS, TitleN, XMLstring = "", Qtext1,
CList, ICList},
dt = Floor[N[Log[Num] / Log[10], 2]] + 1;
XMLstring = XMLstring <> "<?xml version='1.0' encoding='UTF-8'?'>\n";
XMLstring = XMLstring <> " <quiz>\n";
XMLstring =
XMLstring <> " <!-- question: 0 -->\n <question type=\"category\">\n";
XMLstring = XMLstring <> " <category>\n <text>";
XMLstring = XMLstring <> Cat;
XMLstring = XMLstring <> "\n </category>\n </question>";
For[mm = 1, mm <= Num, mm++, mmS = PrintF[mm, dt];
TitleN = Title <> " (" <> mm <> ")";
If[StringQ[Qtext], Qtext1 = Qtext, Qtext1 = Qtext[mm];
CList = CListFunc[mm]; ICList = ICListFunc[mm];
XMLstring = XMLstring <> GenerateMCxml[TitleN, Single, Qtext1, CList,
ICList, "50"];
XMLstring = XMLstring <> "\n";
XMLstring = XMLstring <> " </quiz>\n";
SetDirectory[NotebookDirectory[]];
Export[Fname, XMLstring, "Text"];
];
FuncList = Append[FuncList, GenerateMCxml];
```

図6 プログラム例 (XML生成ルーチン 2)

図 5,6,7,8 にプログラムの例を掲載した。

誤りパターンは, 過去の紙によるテストや課題のログを参考に分析して生成するのが効果の最も期待できる方法である。しかし, これは非常に時間がかかる作業のため, 一部を除いて学生が間違えそうなパターンを想定し, パターンを生成している。

学生の誤りパターンを考慮した選択肢の例として, 工学向け線形代数の授業で用いた多肢選択問題の例と

```

NUM = 300; PatNum = 6;
RI [m_] := RandomInteger[m];

EFL = {};
For[i = 1, i <= 100, i++, (* Echelon Form のリストを作る *)
EFL = Append[EFL, RandomEchelonPV2[RI[1] + 3, 5, {1, 3}, {-2, -1, 1, 2, 3}]];
EFL = Append[EFL, RandomEchelonPV2[RI[1] + 3, 5, {1, 4}, {-2, -1, 1, 2, 3}]];
EFL = Append[EFL, RandomEchelonPV2[RI[1] + 3, 5, {2, 4}, {-2, -1, 1, 2, 3}]];
      ⋮ (中略) ⋮ 被約階段行列のパターン生成
EFL = Append[EFL, RandomEchelonPV2[RI[1] + 3, 5, {1, 4}, {-2, -1, 1, 2, 3}]];
EFL = Append[EFL, RandomEchelonPV2[RI[1] + 3, 5, {1, 4}, {-2, -1, 1, 2, 3}]];
EFL = Append[EFL, RandomEchelonPV2[4, 6, {1, 2, 4, 5}, {-1, 2, 3}]];
EFL = Append[EFL, RandomEchelonPV2[4, 6, {1, 2, 4, 6}, {-2, 1, 2, 3}]];
EFL = DeleteDuplicates[EFL]; (* 重複削除 *)
QtList = {}; (* 問題リスト. 最初が正解, 後の5つ (PatNum-1個) は他のEchelonFormから作ったもの *)
For[i = 1, i <= 1000, i++,
EFM = RandomSample[EFL, PatNum];
Qt = {};
(* QtList は {EchelonForm, 行列} の PatNum からなるリスト. 誤答の応答に EchelonForm を使う *)
For[j = 1, j <= PatNum, j++, n = Length[EFM][[j]]; (* n は EchelonForm の行数 *)
While[Det[RM = Table[Table[RndN2[3], {i, 1, n}], {j, 1, n}]] = 0];
Qt = Append[Qt, {RM, EFM[[j]], EFM[[j]]}];
QtList = Append[QtList, Qt];
QtList = RandomSample[QtList, NUM];

TextF[mm_] := Module[{Qt, ret}, Qt = QtList[[mm]];
ret = "次の行列のうち, 行に関して基本変形して被約階段行列に変形すると\\(";
ret = ret <> TOT[Qt[[1, 2]]] <> "\\) となるものを1つ選びなさい.";
Return[ret];
通常の逆問題 (計算の負荷が少ない)

CListF[mm_] := Module[{Qt, ret = {}}, Qt = QtList[[mm]];
ret = Append[ret, {"\\(" <> TOT[Qt[[1, 1]]] <> "\\)", "正解です."}]; Return[ret];
ICListF[mm_] := Module[{Qt, ret = {}}, Qt = QtList[[mm]];
For[ct = 2, ct <= PatNum, ct++,
ret = Append[ret, {"\\(" <> TOT[Qt[[ct, 1]]] <> "\\)",
"被約階段行列は\\(" <> TOT[Qt[[ct, 2]]] <> "\\) となります."}]];
Return[ret];

GenerateMCxml[DFname[], "LA2-MC/与えられたEchelonFormを持つ行列", "与えられたEchelonForm
CListF, ICListF]

```

図7 プログラム例 (実際の生成例1)

```

(* Select correct derivative of x sin or cos(ax+b) *)
QtList = {}; (* 選択肢 5, 正解数 1 *)
For[ct = 1, ct <= 300, ct++,
a = (2 * RandomInteger[1] - 1) * (RandomInteger[9] + 2);
b = (2 * RandomInteger[1] - 1) * (RandomInteger[9] + 2);
c = RandomInteger[1]; (* c = 0: sin, c = 1: cos *)
While[Abs[a] = Abs[b], b = (2 * RandomInteger[1] - 1) * (RandomInteger[9] + 2)];
QtList = Append[QtList, {a, b, c}];
QtList = DeleteDuplicates[QtList];

TextF[mm_] := Module[{a, b, c, csf, ret}, {a, b, c} = QtList[[mm]];
If[c = 0, csf = "\\sin", csf = "\\cos"];
ret = "\\(x\\) の関数 \\(y=x\\), \\(y=csf <math>\\cos</math> \\(x <math>+ a x + b</math>)</math> を求めよ.";
Return[ret];
問題文出力

CListF[mm_] := Module[{a, b, c, f0, f1, s1, tmp, ret = {}}, {a, b, c} = QtList[[mm]];
If[c = 0, f0 = "\\sin"; f1 = "\\cos"; s1 = 1, f0 = "\\cos"; f1 = "\\sin"; s1 = -1];
tmp = "\\(" <> TT[s1 * 2 * a] <> f1 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)";
tmp = tmp <> "x" <> f0 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\)";
ret = Append[ret, {tmp, "正解です."}]; Return[ret];

ICListF[mm_] := Module[{a, b, c, ret = {}}, f0, f1, f0s, f1s, s1, f0m1, f0m2, f0m3, f0m4},
{a, b, c} = QtList[[mm]];
f0m1 = "積の微分法を確認しよう.";
f0m2 = "2階導関数は一部分を微分していいということではありません.";
f0m3 = "微分を1回し頻なっているよう.";
f0m4 = "何らかの計算ミスか?";

If[c = 0, f0 = "\\sin"; f1 = "\\cos"; s1 = 1, f0 = "\\cos"; f1 = "\\sin"; s1 = -1];
f0s = f0 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)";
f1s = f1 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)";
ret = Append[ret, {f0s <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m1}];
ret = Append[ret, {f1s <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m2}];
ret = Append[ret, {"\\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m3}];
ret = Append[ret, {"\\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m4}];
ret = Append[ret, {"\\(" <> TT[s1 * 2 * a] <> f1 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m1}];
ret = Append[ret, {"\\(" <> TT[-s1 * 2 * a] <> f1 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m2}];
ret = Append[ret, {"\\(" <> TT[s1 * 2 * a] <> f1 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m3}];
ret = Append[ret, {"\\(" <> TT[-s1 * 2 * a] <> f1 <> "\\) <math>+</math> \\(" <> TT[a * x + b] <> "\\) <math>+</math> \\(" <> TT[-a^2] <> "\\)", f0m4}];
Return[RandomSample[ret, 4]];
誤答パターンは 4 パターンを選択

```

図8 プログラム例 (実際の生成関数例)

文系向け基礎数学で用いた多肢選択問題の例をあげる (図9, 10).

2.4 データの公開と利用

筆者が開発を行うための関数群や生成した XML は自サイトで公開している⁽⁹⁾. ライセンスは明記していないが, MIT ライセンスで自由に利用可能してもらってよい. また, 神戸大学の長坂耕作氏も同じように多肢選

問題1 最大得点 1.00
xの関数 e^{10x+4} の導関数を求めよ.
1つ選択してください:
 $4e^{10x+4}$
 e^{10x+4}
 $40e^{10x+4}$
 $10e^{10x+3}$
 $10e^{10x+4}$

問題4 最大得点 1.00
xの関数 9^{2x-5} の導関数を求めよ.
1つ選択してください:
 $(\log_e 9)^{9^{2x-5}}$
 $(\log_e 9)^{9^{2x-5}}$
 $9 \cdot 9^{2x-5}$
 $-5 \cdot 9^{2x-5}$
 $(-\log_e 5)^{9^{2x-5}}$

問題7 最大得点 1.00
xの関数 $y = \cos(10x - 3)$ および $z = e^{10x-3}$ の二階導関数の
1つ選択してください:
 $y'' = \cos(10x - 3), z'' = e^{10x-3}$
 $y'' = -100 \cos(10x - 3), z'' = 100e^{10x-3}$
 $y'' = -10 \cos(10x - 3), z'' = 10e^{10x-3}$
 $y'' = 100 \sin(10x - 3), z'' = 100e^{10x-3}$
 $y'' = \cos(10x - 3), z'' = 100e^{10x-3}$

問題1~6は他のクラスと共通だが, 問題7だけ自クラスに追加, 例年紙の課題で間違い ← の多い問題の変化を見る実験

図9 問題例1(基礎数学抜粋)

次の行列Aは被約階段行列に変形すると右のようになる。このとき, 方程式 $Ax = 0$ の解空間UとAの列ベクトル $a_j, j = 1, 2, \dots$ が生成する空間 $W = \langle a_1, \dots \rangle$ の基底と次元について正しいものを1つ選びなさい。 $A = \begin{pmatrix} -4 & 8 & 1 \\ 3 & -6 & 4 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

1つ選択してください:

- Uの基底として $\begin{pmatrix} -4 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \end{pmatrix}$ がとれて次元は2, Wの基底として $\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}$ がとれて次元は1
- Uの基底として $\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}$ がとれて次元は1, Wの基底として $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ がとれて次元は2
- Uの基底として $\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}$ がとれて次元は1, Wの基底として $\begin{pmatrix} -4 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \end{pmatrix}$ がとれて次元は2

図10 問題例2(線形代数の1例)

択問題の自動生成に取り組みされており, 氏のサイトに公開されている⁽⁶⁾. どちらのXMLも自由に利用することができる。

ただし, 大学毎に用語や用法が異なることがあるので, 注意が必要である. 例えば本学では pivot という用語を用いていなかったため, 相互利用するために授業で導入した. 本質的にはその程度の補足で相互利用は可能であると考えられる。

3. 今後の課題

3.1 反転学習教材の開発

反転授業で活用するには, 動画毎に確認する問題を開発することが望ましい. 動画を見たことの何らかの達成感が期待されるからであり, 自習教材としての学習後テストの意味もある。

筆者は線形代数の全課程の動画を開発済みであるが, それらは短か目に開発しているものの, ある研究者は3分が集中力の限界と述べていたようにまだまだ短くす

る必要がある。その結果、必要な確認項目も増え、より多くの問題開発が必要となる。

筆者は比較的容易に問題を開発できるようになったが、重要なのは問題の設計であり、反転学習用教材の他、テストやテストの復習課題の利用など幅広い用途に対応して問題を開発するのは通常の STACK の問題を開発するより困難である。

ただし、最新の STACK では多肢選択問題で数式が表示できるようになっており、将来は STACK を使って開発することも可能になることが期待される。(本学では残念ながら古くて多肢選択問題までは開発できるが数式が \TeX 形式のまま表示され使いものにならない)。いずれにせよ、多数の問題の設計が必要である。

3.2 効果検証

過去の誤答例を基に多肢選択問題を開発した場合に、それらを実際に紙の課題やテストと併用してどのような効果があるかを検証したい。

本稿執筆時の前の週に、試行的に基礎数学の課題で例年紙の課題で間違いの多い問題を出題した。紙の課題は回収済みで採点も終わっている。

オンライン課題には期限がないため、全員はまだ実施していないが、実施した学生の紙の課題の正答率は非常に高かった。対して、実施していない学生には例年通り、積の微分や合成関数の微分を理解していないと思われる学生が多数見られた。具体的なデータについては本稿には間に合わなかったので、当日公開し検証したい。

今後はこのような過去に見られる誤答例を基準にした多肢選択問題の実施と併用した場合にどのように変化するかを検証を実施し、効果的な多肢選択問題のブラッシュアップも検討していきたい。

3.3 開発環境の改善

現在、前述の長坂氏とは共同研究を行っており、XML の生成手法には互換性はなくそれぞれ独自に行っているが、生成した XML は相互利用も行っている。

しかし、問題の種類を今後も飛躍的に増やす必要があり、そのためには広く一般の教員とも共有して開発できるような体制も必要であろう。MeLQS⁽¹⁰⁾とも協調して、どのような問題が必要か、を含め、問題データの共有だけでなく開発体制も共有できるようにしていきたいと考えている。

謝辞

本研究は JSPS 科研費 18K02941 「気づきと深い学びを誘発する大学数学用スマートフォン適応型反転学習教材の開発と検証」の助成を受けたものです。

参考文献

- (1) https://github.com/maths/moodle-qtype_stack [Accessed 10 Dec. 2018]
- (2) Kawazoe, M. and Yoshitomi, K., 2017. E-learning/e-assessment systems based on webMathematica for university mathematics education. *MSOR Connections*, 15(2), pp.1724.
- (3) DigitalEd.com, 2018. M³OBIUS ASSESSMENT ONLINE ASSESSMENT SYSTEM FOR STEM COURSES. Available at: <https://www.digitaled.com/products/assessment/>. [Accessed 10 Dec. 2018]
- (4) The Mathematical Association of America, 2018. WeB-WorK. Available at: <http://webwork.maa.org> [Accessed 10 Dec. 2018].
- (5) Numbas.org.uk, 2018. Numbas: really versatile maths e-assessment. Available at: <https://www.numbas.org.uk> [Accessed 10 Dec. 2018].
- (6) Moodle XML Questions Generator - http://wwwmain.h.kobe-u.ac.jp/~nagasaka/research/xml_quiz/ [Accessed 11 Dec. 2018].
- (7) Yasuyuki NAKAMURA, Takahiro NAKAHARA: "DEVELOPMENT OF A MATH INPUT INTERFACE WITH FLICK OPERATION FOR MOBILE DEVICES", *Proceedings of 12th International Conference Mobile Learning 2016*. 113-116(2016).
- (8) Shirai, S. and Fukui, T., 2017. MathTOUCH: Mathematical input interface for e-assessment systems. *MSOR Connections*, 15(2), pp.7075.
- (9) http://www.las.osakafu-u.ac.jp/~yositomi/moodle_xml/
- (10) 吉富 賢太郎, 川添 充, 中原 敬広, 中村 泰之, 福井 哲夫, 白井 誌沙香, 加藤 克也, 谷口 哲也: "数式オンラインテストの標準仕様 MeLQS の提案と仕様書作成ツールの試作", *教育システム情報学会 第 42 回全国大会*, pp.447-448(2017).