

数式処理システムによる数式の自動生成に関する考察と実装

明治大学大学院先端数理科学研究科先端メディアサイエンス専攻 藤田祥一

1 はじめに

本稿では、高校数学で学ぶ特定の単元に現れる数式を適切な条件下でランダムに生成するようなソフトウェアのことを数式自動生成システム（以下、自動生成）と呼ぶことにする。一般に、自動生成は数式処理システムによって行われる。教育現場で自動生成が活用される場面は2つ想定される。1つは教員による問題作成であり、もう1つはeラーニングにおける問題作成である。授業における例題、演習問題、小テストなどのために、教員は多くの数式を翻案する¹ことが必要である。また、現状では教員経験が必要な作業の1つでもあり、教員が手作業で数式の作成を行っている。このことから、自動生成を用いることにより、教員が数式を翻案する手間の軽減が期待される。eラーニングやCBT（Computer Based Testing）においては、課題に取り組む学生間のカンニングを防止する目的から、学生ごとに異なる問いを呈示する場合があります。自動生成はこのような場面でも活用されている。

LMS（Learning Management System）であるMoodle上で動作するオンラインテスト・評価システムのSTACK[1]では、乱数を用いた出題が可能であり、同一の質問文の数式部分だけを受験者ごとに変えることができる。数式部分が異なれば正答も受験者ごとに異なるが、STACKでは数式処理システムMaximaを用いて正誤判定を行っている。また、長坂[2]は、線形代数の多肢選択型問題を乱数によって生成するシステムを開発した。このシステムは数式処理システムであるMathematica用とプログラム言語であるPython用が作られており、生成された問題がMoodleへ移植できるようになっている。

自動生成における問題点は主に2つある。1つは、乱数を用いて生成された数式がユーザである教員の望むものであるとは限らないことである。もう1つは、ユーザである教員が数式処理システムや開発環境の知識を持たないとスムーズな使用が難しいことである。

教員が自動生成に求めることは主に授業教材として活用できる数式の生成である。具体的には、授業内容や生徒のレベルに合った数式の生成を望む。しかし、自動生成は乱数によって異なる数式を幅広く生成する仕様になるのが普通である。そのため、生成された数式全てが教材に適した数式とは限らず、生成された数式1つ1つを使用・用途に合っているか教員が確認するという手間がかかる恐れがある。一方で、限定的な生成に絞った自動生成の設計すると、使用・用途ごとに自動生成を開発する必要が出てきてしまい、自動生成の汎用性が少なくなってしまう。教員が効率的に数式を翻案するための方策として、1つのソフトウェアを用いて、教員が生成する数式の特徴を選べて、教員の意図した数式のみを生成することができる自動生成の設計が求められる。

第2の問題点として、自動生成を使用する教員が必ずしも数式処理システムの使用経験や開発環境に関する知識を持ち合わせてるとは限らないことが想定される。ソフトウェアのダウンロードやインストールといった一般的なパソコン操作はできたとしても、ソフト

¹特に、単元の趣旨を踏まえた教育上効果が見込まれるような数式を作成すること。

ウェアに関連する詳細設定を CUI²によって行う作業がどの教員でも可能かと問われると難しい部分がある。開発環境によって数式の入力・表現など細かい部分が違うこともあり、開発環境に依存するコマンド入力や操作を求めてしまうと、教員によってはソフトウェアを使うための新たな知識を習得する手間が発生し、数式を翻案する手間の軽減という本来の目的が薄くなってしまう。教員間の技術の個人差も関係なく、幅広い教員が気軽に使用できるソフトウェア設計という観点で、使用するユーザの知識や経験に依存しない方法で使用できる自動生成が求められる。

以上2点の問題点を踏まえて、本研究では数式処理システムである Risa/Asir を用いて、ユーザである教員の使用・用途に合わせて生成する数式の特徴を細かく指定できるような自動生成の提案・実装を行った。なお、本稿ではソフトウェアを使用するための準備にあたる Risa/Asir 本体と Risa/Asir のライブラリである [6] のインストール等はユーザができていた環境であることを前提に述べる。

Risa/Asir とは、オープンソースの計算機代数 (数式処理) システムであり、非営利・非商用であればライセンスはフリーで配布・改変等も自由となっている。竹島、横山、野呂によって富士通株式会社国際情報社会科学研究所で開発が進み、1994年に初版が公開された [3]。現在は神戸大学にてソフトウェアが公開されている³。Risa/Asir は以下の特徴をもつ [3, 4, 5]。

- C 言語風のプログラム言語を使用
- 定数・変数等の型宣言が不要
- 因数分解 (整数上, 代数拡大体上), 最大公約多項式計算などの多項式演算の高速性 (特に, グレブナー基底の演算に強い)
- ユニークな陰関数の描画機能
- ユーザ定義関数によるツール

また、野呂や大島 [6] は Risa/Asir を用いて数学の問題解決に役立つ便利なユーザ関数を作成し、それらを集めたライブラリを自身のホームページなどで公開している。

本研究で作成した自動生成の特性は以下の3点である。1点目は、数式の生成方法を1つに固定しながらも、ユーザが指定できる数式の細かい特徴を多数用意し、かつ特徴を複数個満足するような数式を生成できることである。異なる数式を幅広く生成するという自動生成の目的をできるだけ薄めないように、生成時に用いる変数は多く設定してあるが、生成する数式の特徴によって変数の値を決める乱数の範囲に制限を設けたり、定数にすることができるよう自動生成を設計した。2点目は、ユーザが数式処理の操作に慣れていない場合を想定して、対話式の誘導文で入力内容を伝え、キーボード入力は数字またはアルファベット1文字の組み合わせで済むようにしたことである。対話式によって指定できる特徴を1つ1つ伝えることで、設定可能な数式の特徴を一通り漏れなくユーザに呈示することが可能になった。3点目は、生成された数式の中からユーザが選んだ数式のみ \TeX

²Character User Interface. コンピュータの操作において、命令や情報の表示を全て文字によって行うユーザインターフェースの様式。

³<http://www.math.kobe-u.ac.jp/Asir/asir-ja.html> から Risa/Asir をダウンロード可能。

形式の数式表記で出力できることである。特徴を指定して生成された数式の中から、ユーザである教員がさらに教育的な観点から適切な数式であるかを吟味する場合もある。生成された数式の中から、ユーザが効率よく数式を吟味して、選んだ数式を $\text{T}_{\text{E}}\text{X}$ 形式で取り出せるように自動生成を設計した。なお、選んだ数式を $\text{T}_{\text{E}}\text{X}$ 形式で出力できるようにした理由としては、数学の文章作成でよく使われているシステムが $\text{T}_{\text{E}}\text{X}$ であり、生成された数式をコピー&ペーストすることを想定したときに、 $\text{T}_{\text{E}}\text{X}$ 形式の数式の方が汎用性が高く使い勝手が良いと筆者が考えたからである。

本稿では、「因数分解」、「2次関数」、「微分」それぞれの単元に対応した3種類の自動生成について、数式の生成方法や指定できる数式の特徴などソフトウェアの概要を述べる⁴。また、「因数分解」と「微分」の自動生成については東京都高等学校数学教育研究会主催の研究会と東京理科大学数学教育研究会主催の研究会⁵にて実演を行った際に頂いたソフトウェアに対する意見も報告する。

2 Risa/Asir を用いた自動生成の実装

本研究で作成した自動生成の主な流れは以下になる。

1. 生成する数式の個数を指定してソフトウェアを開始する。数式は一度に最大100個まで生成が可能である。
2. 対話文に答える形で生成したい数式の特徴をキーボード入力で指定する。
3. 指定した数式の特徴の設定一覧がRisa/Asirの画面上に表れる。設定に誤りがなければ、0を入力して⁶数式の生成を開始する。もし設定に誤りがある場合は、該当する設定に対応した数字を入力することによって設定が変更できる。
4. ユーザが設定した特徴を満たす数式を指定した個数に達するまで、数式生成の過程を繰り返す。
5. `dviout`⁷によってRisa/Asirの画面とは別ウィンドウに生成された数式の一覧表を表示する。
6. 数式の一覧表からユーザが選んだ数式を $\text{T}_{\text{E}}\text{X}$ 形式の数式表記で出力する。
7. 0を入力してソフトウェアを終了する。なお、数式を追加生成や生成する数式の特徴を変更しての再出力も可能である。

本研究で作成した自動生成では、Risa/Asirで生成された数式全てを一度 $\text{T}_{\text{E}}\text{X}$ 形式の数式表記に変換している。[6]にはRisa/Asir上の数式表記を $\text{T}_{\text{E}}\text{X}$ 形式の数式に変換する`os_md.my_tex_form`というユーザ関数がある。この関数により、生成された数式を $\text{T}_{\text{E}}\text{X}$ 形式に数式を変換して`dviout`での表示やRisa/Asirの画面への出力を行っている。

⁴本稿ではソフトウェアのソースコードの掲載は省略する。ソースコードを必要とされる方は筆者宛(sf.08092627@gmail.com)にメールを頂ければ送付する。

⁵主な参加者は現職の中学・高等学校の教員であり、退職された教員や大学教員の方も参加している。

⁶0を入力する部分は、何も入力せずにEnter/Returnキーを押すだけでも良い。

⁷大島が開発した $\text{T}_{\text{E}}\text{X}$ のpreviewerである。

dviout 表示を行う理由としては、数式が手書きの数式表記と同様な表示になり、見やすくなるからである。Risa/Asir の画面に直接生成された数式を表示することも可能であるが、狭い画面に大量の数式が並ぶため数式 1 つ 1 つが見づらくなってしまふ。また、生成された数式は Risa/Asir に依った数式表示になっているため、そのまま画面に表示すると手書きの数式表記では省略する積記号や括弧などが残ってしまう。生成された数式を T_EX 形式に変換して dviout 表示を行うことによって、省略や T_EX コマンドに置き換えが可能な記号は変換され、より自然な形で表示された数式の一覧表が作成できる。さらに、ユーザが T_EX 形式で出力する数式を選ぶ際、dviout で表示を行ったときに各数式へ自動で振られる番号とユーザが入力する数字を対応させておくことによって、ユーザが選んだ数式をスムーズに T_EX 形式で出力できるようにも設計した。

以下からは、作成した 3 種類の自動生成の概要や実行例について述べる。

2.1 多項式の因数分解する問題の自動生成 afactor

afactor では、必ず整数係数の 1 次式の積に因数分解が可能な 2~10 次までの 1 変数または 2 変数同次多項式の生成、及びその多項式を因数分解した結果の呈示を行う。また、ユーザが選んだ多項式に対して、因数分解を答えさせる多肢選択型問題の選択肢を 2~10 択の中で生成することができる。

多項式の生成は、ユーザが設定する 2 つの変数 V_1, V_2 ⁸ に対して

$$\prod_{i=1}^n (a_i V_1 + b_i V_2) \quad (a_i \in \mathbb{N}, b_i \in \mathbb{Z} \setminus \{0\}, 2 \leq n \leq 10)$$

と 1 次式の積を取る形で生成している⁹。 a_i と b_i はユーザが設定した範囲の乱数によって選ぶが、互いに素になるように値を選んでいるため 1 つの括弧内で共通因数が存在することはない。また、誤答の選択肢は以下の流れで生成している。

1. 多項式の V_1^n と V_2^n の係数を素因数分解して、それぞれの素因数リストを作る。
(例. V_1^3 の係数 : 12 \rightarrow [2, 2, 3], V_2^3 の係数 : 16 \rightarrow [2, 2, 2, 2])
2. 長さが n で全ての要素がベクトル (1,1) であるリスト P を用意する。
(例. $P = [(1, 1), (1, 1), (1, 1)]$)
3. V_1^n の素因数リストの各要素を P からランダムに選んだ要素のベクトルの第 1 成分に掛けていく。(例. $P = [(2, 1), (3, 1), (2, 1)]$)
4. V_2^n の素因数リストの各要素を P からランダムに選んだ要素のベクトルの第 2 成分に掛けていく。(例. $P = [(2, 1), (3, 2 \cdot 2 \cdot 2 \cdot 2), (2, 1)]$)
5. P の各要素の第 2 成分の符号をランダムに変更する。(例. $P = [(2, 1), (3, -16), (2, -1)]$)
6. P の第 1 要素から第 n 要素まで順番に、ベクトルの第 1 成分を V_1 の係数、第 2 成分を V_2 の係数とした 1 次式を作成して、 $(c_1 V_1 + d_1 V_2)(c_2 V_1 + d_2 V_2) \cdots (c_n V_1 + d_n V_2)$ ($c_i \in$

⁸1 変数の場合は $V_2 = 1$ となる。

⁹本稿では \mathbb{N} を正の整数とする。

\mathbb{N} , $d_i \in \mathbb{Z} \setminus \{0\}$, $2 \leq n \leq 10$) の形になるように繋げて数式を生成する.

(例. $P = [(2, 1), (3, -16), (2, -1)] \Rightarrow (2V_1 + V_2)(3V_1 - 16V_2)(2V_1 - V_2)$)

7. 6. にて生成された数式が正答や他の誤答と一致していなければ選択肢として採用する. 一致した場合は生成し直す.

8. 以上の工程を繰り返して, 必要な個数の誤答を生成する.

上記のように誤答を生成することで, 各選択肢を展開したときに V_1^n と V_2^n の係数の絶対値がどれも同じになるため, 解答者が選択肢を展開して正答を探そうした場合でも多少の計算を必要となることが予想される. よって, 一目では正解と見分けがつかない誤答の生成が可能になる. 当初は誤答の生成もユーザが指定した範囲の乱数を用いて適当に選択肢を生成するように設計していたが, 高校教員の方から「なるべく生徒が間違えそうな良い誤答を生成することはできないのか」という意見を頂き, 今回の誤答の生成方法を考案した. 多項式の生成において設定できる特徴とその対話文は以下になっている.

- “生成する式の次数をいくつにしますか?”

2~10 次式までの中から生成する多項式の次数を 1 つ指定する. 多項式の次数と同じ数字 2~10 を入力する.

- “変数を指定して下さい (例. 1 変数:[x,1], 2 変数:[x,y] など)”

変数 V_1 と V_2 を指定する. 1 変数多項式の場合は [“変数”, 1], 2 変数同次多項式の場合は [“変数”, “変数”] とリスト形式で入力する. なお, 変数はアルファベット小文字でユーザが自由に選ぶことができる.

- “使用する乱数の範囲は +- いくつにしますか (推奨:5)? (1-50)”

a_i, b_i の取り得る値の範囲を指定する. 入力は 1~50 の数字であり, 入力した数 n に対して a_i は n 以下の自然数から選ばれ, b_i は 0 を除く $|n|$ 以下の整数から選ばれる. デフォルトは $n = 5$ であり, 何も入力せず Enter/Return キーを押すとデフォルト設定になる.

- “式の最高次数の係数を 1 に固定しますか? (0:いいえ, 1:はい)”

「はい」を選択すると $a_i = 1$ に固定して多項式を生成する.

- “展開後の符号の組み合わせはどうしますか? (0:ランダム, 1:全て +, -1:全て -, 2:(+, -) ほぼ均等)”

対応する数字を入力することによって, b_i の符号を制御することができる. 「全て +」を選ぶと b_i の符号が正に, 「全て -」を選ぶと b_i の符号が負に固定される. また, 「(+, -) ほぼ均等」を選ぶと b_1, \dots, b_n の値が正負半分ずつくらいを取るように符号を決める.

- “選択肢を作成しますか? (0:いいえ, 1:はい)” “何択にしますか?(2-10)”

多肢選択型問題の選択肢を作成することができる. この対話文は $\text{T}_{\text{E}}\text{X}$ 形式で出力する数式の番号を入力した後に出現し, 「はい」を選択すると選択肢の個数を聞かれる.

選択肢の個数は2~10の数字を入力して指定する。出力は $\text{T}_\text{E}\text{X}$ 形式で表された正答と誤答を合わせたリストとなる。なお、正答がリストの何番目に配置されるかはランダムに決まる。

`afactor`では、ユーザである教員が行う因数分解可能な2~10次までの多項式を翻案する作業や多肢選択型問題の作成を支援することができる。このソフトウェアの派生として、「1次式の積を展開する問題の自動生成」や「因数分解によって解ける方程式の自動生成」も開発した。しかし、`afactor`で可能なことは「因数分解できる多項式を生成する」、「多肢選択型問題の誤答を生成する」までに留まる。このソフトウェアをより教育的に活用するために高校教員の方から以下の意見を頂いた。

1つ目は「生徒に配る演習プリントを自動作成してくれる機能が欲しい」という意見である。2つ目は「生成された問題を Web 上でのドリル学習用教材として役立てることはできないか」という提案である。1つ目については、`dviout`への出力を上手く改良することによって、`Risa/Asir`単体で実装できる可能性がある。2つ目については、`Risa/Asir`単体での処理では厳しく、他のシステムと合わせた開発が必要となる。`Risa/Asir`にはブラウザ上でシステムを使用する方法が存在する。ゆえに、問題生成には`afactor`をブラウザ上で動かして、問題の呈示・解答欄等は他のシステムで形作ることによって、ブラウザ上でのドリル学習ツールとして実現が可能であると考えている。

以上の機能の実装も含め、`afactor`の教育利用に向けて今後も改良を加える。

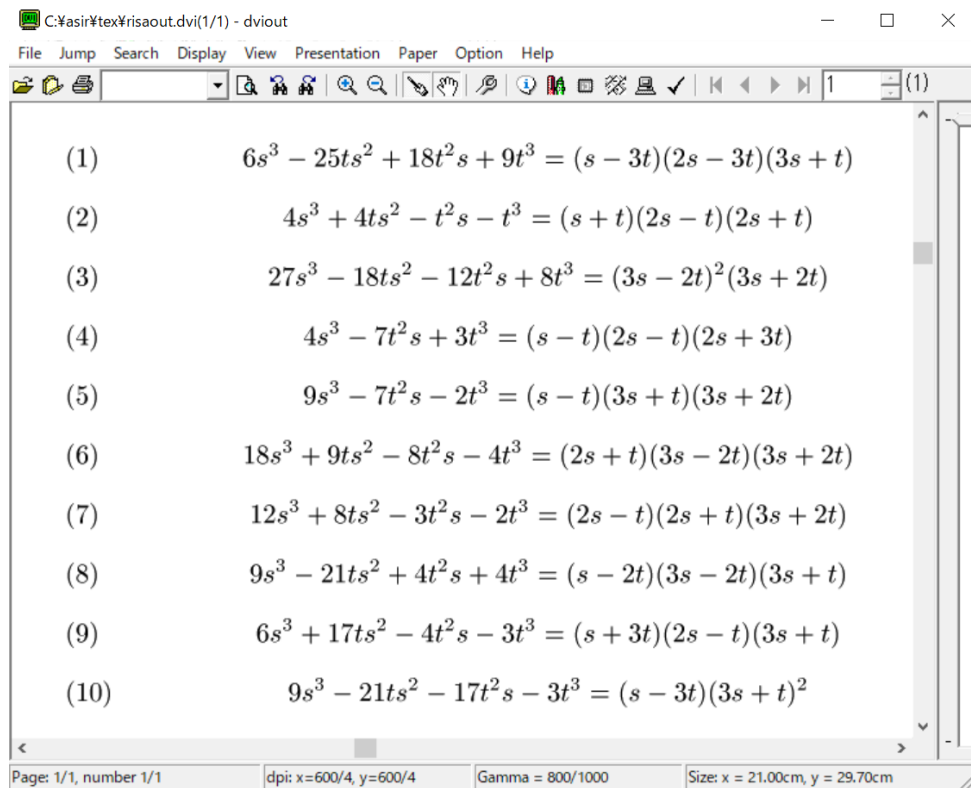


図 1: `afactor`にて生成された多項式と因数分解した結果の一覧表

Risa/Asir の画面での afactor の実行例 (図 1 を表示させたときの設定)

[2715] afactor(10);

生成する式の次数をいくつにしますか? (2-10): 3

変数を指定して下さい (例. 1 変数:[x,1], 2 変数:[x,y] など): [s,t]

使用する乱数の範囲は+-いくつにしますか(推奨:5)? (1-50): 3

式の最高次数の係数を 1 に固定しますか? (0:いいえ, 1:はい):

展開後の符号の組み合わせはどうしますか? (0:ランダム, 1:全て+, -1:全て-, 2:(+,-) ほぼ均等): 2

- (1) 問題生成数: 10
- (2) 式の次数: 3
- (3) 使用する乱数の範囲: 3
- (4) 変数: [s,t]
- (5) 最高次の係数: 一般
- (6) 展開後のパターン: (+,-) ほぼ均等

上記の設定で出力しますか? (0:出力, 1-6:各種設定変更):

(1 ~ 10) 問題の TeX 出力 (出力したい問題番号を数又はリストで入力)

(-1) 問題の追加出力

(-2) 出力の設定変更

(0) プログラム終了

TeX 出力したい問題または他のコマンドに該当する括弧内の数を入力してください: [4,10]

選択肢を作成しますか? (0:いいえ, 1:はい): 1

何択にしますか?(2-10): 4

$$4s^3-7t^2s+3t^3=(s-t)(2s-t)(2s+3t)$$

$$[(s-t)(s+t)(4s+3t) (s-t)(2s-t)(2s+3t) (s-3t)(2s-t)(2s+t)$$

$$(s-3t)(s+t)(4s-t)]$$

$$9s^3-21ts^2-17t^2s-3t^3=(s-3t)(3s+t)^2$$

$$[(s-3t)(3s-t)(3s+t) (s-3t)(3s+t)^2 (s-t)(s+3t)(9s-t) (s-3t)(s+t)(9s-t)]$$

追加で出力する問題または他のコマンドに該当する括弧内の数を入力してください:

プログラムを終了しました

2.2 2次関数の自動生成 qufu

qufu では、各次数の係数が有理数である 2 次関数の生成、及びその 2 次関数の頂点の座標と x 軸との交点の x 座標の呈示を行う。dviout の表示では生成された 2 次関数 1 つに対して

[“2 次関数”, [“頂点の x 座標”, “頂点の y 座標”], [“ x 軸との交点の x 座標”¹⁰]]

のようにリスト形式で値がまとめられたものが表示される。

2 次関数の生成はユーザが設定する変数 V に対して

$$y = \frac{a_1}{b_1} \left(V - \frac{a_2}{b_2} \right)^2 + \frac{a_3}{b_3}$$

の形で生成しており、リストの中には括弧を展開・整理した式が表示される。 $a_1, b_1, a_2, b_2, a_3, b_3$ はユーザが設定した範囲の乱数で選ばれるが、 $\frac{a_1}{b_1}, \frac{a_2}{b_2}, \frac{a_3}{b_3}$ は既約分数となるように設定している。また、係数の符号は $\frac{a_1}{b_1}, \frac{a_2}{b_2}, \frac{a_3}{b_3}$ を生成した後に決めている。

2 次関数の生成において設定できる特徴とその対話文は以下になっている。

- “使用する乱数の範囲は +- いくつにしますか?(推奨:5) (1-50)”
 $a_1, b_1, a_2, b_2, a_3, b_3$ の取り得る値の範囲を指定する。入力は 1~50 の数字であり、入力した数 n に対して a_1, a_2, a_3 は 0 以上 n 以下の整数が選ばれ、 b_1, b_2, b_3 は n 以下の自然数が選ばれる。デフォルトは $n = 5$ であり、何も入力せず Enter/Return キーを押すとデフォルト設定になる。
- “変数を 1 つ指定して下さい (小文字で入力)”
変数 V を任意に 1 つ指定する。変数はアルファベット小文字で入力する。
- “2 次項の係数をカスタムしますか? (0:いいえ, 1:はい)”
“数ほどのタイプにしますか? (0:有理数, 1:1 に固定, 2:整数)”
“凸の向きを指定しますか? (0:ランダム, 1:下に凸, -1:上に凸)”
 $\frac{a_1}{b_1}$ の形や符号を指定することができる。画面には項目に該当する数字を入力し、「はい」を選択すると 2 つの対話文が追加される。「1 に固定」を選択すると $a_1 = b_1 = 1$ に固定され、「整数」を選択すると $b_1 = 1$ に固定される。また、凸の向きを「下に凸」に指定すると $\frac{a_1}{b_1}$ の符号が正に、「上に凸」に指定すると符号が負に固定される。
- “頂点の座標をカスタムしますか? (0:いいえ, 1:はい)”
“頂点の V 座標¹¹の数ほどの種類にしますか? (0:有理数, 1:0 に固定, 2:整数)”
“頂点の V 座標の符号はどうしますか? (0:ランダム, 1:+ のみ, -1:- のみ)”
“頂点の y 座標の数ほどの種類にしますか? (0:有理数, 1:0 に固定, 2:整数)”
“頂点の y 座標の符号はどうしますか? (0:ランダム, 1:+ のみ, -1:- のみ)”

¹⁰生成された 2 次関数が x 軸と交点を 2 つもつ場合はこのリストの要素は 2 つになる。また、交点が 1 つの場合はリストの要素が 1 つに、交点がない場合は No point とリスト内に表示される。

¹¹対話文にある全ての V はユーザが指定した変数のアルファベットに置き換わる。

$\frac{a_2}{b_2}$ 及び $\frac{a_3}{b_3}$ の形や符号を指定することができる。画面には項目に該当する数字を入力し、「はい」を選択すると4つの対話文が追加される。座標の数の種類を「0に固定」に指定すると分子が0に固定され、「整数」に指定すると分母が1に固定される。また、符号を「+のみ」に指定すると正に固定され、「-のみ」に指定すると負に固定される。

- “V軸との交点をカスタムしますか? (0:いいえ, 1:はい)”
“V軸との交点をいくつにしますか? (0:なし, 1:1点, 2:2点)”
“符号の組み合わせはどうしますか? (0:ランダム, 1:+のみ, -1:-のみ)”
“交点のV座標の形を指定しますか? (0:ランダム, 1:整数, 2:有理数, 3:無理数)”
“符号の組み合わせはどうしますか? (0:ランダム, 1:(+,+)のみ, -1:(-, -)のみ, 2:(-, +)のみ)”
“ルートの中の数に上限を付けますか? (0:いいえ, 1:はい)”

V軸との交点の個数を指定することができる。画面には項目に該当する数字を入力し、「はい」を選択するといくつか対話文が追加される。交点が1点の場合は交点のV座標の正負が設定でき、2点の場合は交点2つのV座標の値の形と正負の組み合わせが指定できる¹²。また、交点が2点でV座標の形が「無理数」と指定された場合は、根号の中に表れる値に上限を設定して数式を生成することもできる。

qufuでは、2次関数を決定する上で鍵となる頂点の座標の位置や軸との交点の個数を細かく指定して生成することができる。また、生成された2次関数の頂点や軸との交点の座標も呈示するため、ユーザである教員が2次関数を翻案する際に行う補助計算も軽減される。しかし、qufuの欠点の1つとして、生成された2次関数のグラフの概形がソフトウェア内の機能で確認できない点がある。教育的な観点を考えると視覚的な情報も含め、生成された2次関数を選ぶ方が良い。Risa/Asirには関数のグラフをプロットする機能も備わっているが、グラフを表示するウィンドウの大きさをあらかじめ固定した上でプロットするため、表示する2次関数によって臨機応変にウィンドウの大きさや軸の目盛りの変更することができない。そのため、Risa/Asirの機能のみでグラフを見やすい形に表示する方法が筆者はまだ見つけられていない¹³。この欠点を解決するために他のシステムと組み合わせも視野に入れながら、「生成された2次関数の中からユーザが選んだ関数のグラフを確認できる」機能を追加することが今後の課題の1つとなる。また、2次関数に関連する演習問題（例えば、「ある定義域における最大値・最小値の求める」問題や「ある1次関数との交点の座標を求める」問題）と正答を生成する機能なども加えて教育ツールとして実用性を広げる。

¹²この設定によっては前に設定した条件と相反する条件を付けてしまう場合があるので注意が必要である。なお、相反する条件で2次関数の生成を始めた場合でも、ソフトウェアがエラーで止まることがない設計になっている。

¹³[6]には関数のグラフをdvioutへ出力するos_md.xygraphというユーザ関数が備わっている。しかし、dvioutの仕様上2つの内容を別々のウィンドウへ表示させることができない。そのため、os_md.xygraphを使ってグラフを出力すると、生成された2次関数の一覧表の下にグラフを表示させる形になってしまう。

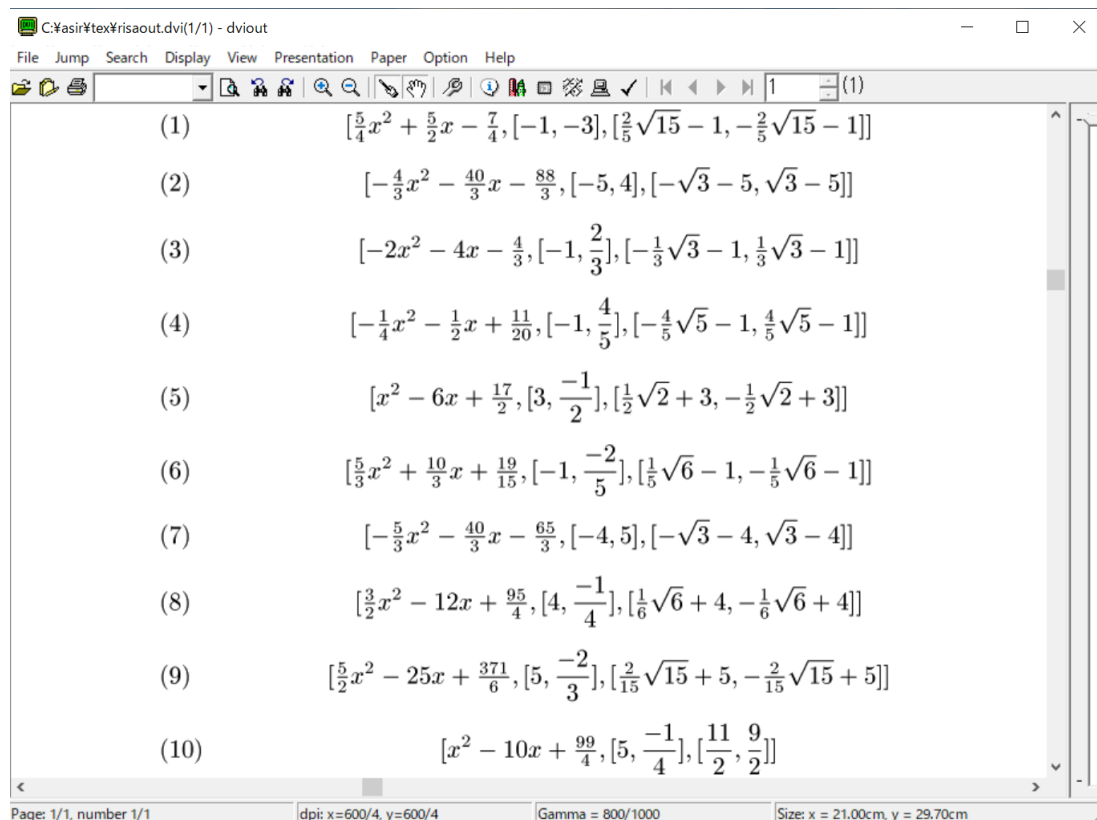


図 2: qufu にて生成された 2 次関数の一覧表

Risa/Asir の画面での qufu の実行例 (図 2 を表示させたときの設定)

[2719] qufu(10);

使用する乱数の範囲は+-いくつにしますか?(推奨:5) (1-50):

変数を 1 つ指定して下さい (小文字で入力): x

2 次項の係数をカスタムしますか? (0:いいえ, 1:はい):

頂点の座標をカスタムしますか? (0:いいえ, 1:はい): 1

頂点の x 座標の数はどの種類にしますか? (0:有理数, 1:0 に固定, 2:整数): 2

頂点の x 座標の符号はどうしますか? (0:ランダム, 1:+のみ, -1:-のみ):

頂点の y 座標の数はどの種類にしますか? (0:有理数, 1:0 に固定, 2:整数):

頂点の y 座標の符号はどうしますか? (0:ランダム, 1:+のみ, -1:-のみ):

x 軸との交点をカスタムしますか? (0:いいえ, 1:はい): 1

x 軸との交点をいくつにしますか? (0:なし, 1:1 点, 2:2 点): 2

交点の x 座標の形を指定しますか? (0:ランダム, 1:整数, 2:有理数, 3:無理数):

符号の組み合わせはどうしますか? (0:ランダム, 1:(+,+)のみ, -1:(-,-)のみ, 2:(-,+)のみ):

- (1) 使用乱数の範囲: 5
- (2) 2次項の係数: 有理数, ランダム
- (3) 頂点の x 座標: 整数, ランダム
- (4) 頂点の y 座標: 有理数, ランダム
- (5) x 軸との交点の個数: 2点, ランダム, ランダム
- (6) 生成数: 10
- (7) 変数: x

上記の設定で出力しますか? (0:出力, 1-7:各種設定変更):

- (1 ~ 10) 2次関数の TeX 出力 (出力した関数の番号を数又はリストで入力)
- (-1) 2次関数の再 (追加) 出力
 - (-2) 出力の設定変更
 - (0) プログラム終了

TeX 出力したい 2 次関数または他のコマンドに該当する括弧内の数を入力してください: [6,10]

```
[\tfrac{5}{3}x^2+\tfrac{10}{3}x+\tfrac{19}{15},[- 1,-\frac{ 2}{ 5}],
[\tfrac{1}{5}\sqrt{6}-1,-\tfrac{1}{5}\sqrt{6}-1]]
```

```
[x^2-10x+\tfrac{99}{4},[5,-\frac{ 1}{ 4}],[\frac{11}{2},\frac{9}{2}]]
```

追加で出力する 2 次関数または他のコマンドに該当する括弧内の数を入力してください:

プログラムを終了しました

2.3 関数を微分する問題の自動生成 biburan

`biburan` では, ユーザが選んだ関数と微分の手法¹⁴の中から関数を微分する問題¹⁵の生成, 及びその関数を微分した結果の呈示を行う. `dviout` の表示では [“関数”, ---, “関数を微分した結果”] のようにリスト形式でまとめられたものが表示される. 問題生成に使用する関数は以下の 10 種類の中から選択することができる. なお, a, b, c はランダムに選ばれる整数とし, V をユーザが任意に決める変数とする.

- 1~3 次の整数乗の単項式: $aV \quad aV^2 \quad aV^3$
- 1~ N 次の整数乗の単項式¹⁶: $aV^c \quad (2 \leq c \leq N)$
- 1~3 次の多項式: $aV^3 + aV^2 + aV + a \quad aV^2 + aV + a \quad aV + a$
- 累乗根の式: $aV^{\frac{b}{c}} \quad (b \geq 1, c \geq 2)$

¹⁴本稿では, 高等学校の教科書に載っている微分の公式の名称などを指す.

¹⁵生成する関数は全て微分可能であることを前提とする.

¹⁶ N はユーザが乱数の範囲を指定する際に入力する数字と同じ.

- 2乗根の式： $a\sqrt{V}$
- 指数関数： $ae^{bV} \quad c^{bV} \quad (b \geq 1, c \geq 2)$
- 対数関数： $a \log bV \quad (b \geq 1)$
- 三角関数： $a \sin bV \quad a \cos bV \quad a \tan bV$
- 逆三角関数： $a \arcsin bV \quad a \arccos bV \quad a \arctan bV$
- 双曲線関数： $a \sinh bV \quad a \cosh bV \quad a \tanh bV$

また、問題の種類は単純な微分 $f'(V)$ (このソフトウェアでは「通常微分」と呼んでいる。)に加え、積の微分 $(f(V)g(V))'$ 、商の微分 $\left(\frac{f(V)}{g(V)}\right)'$ 、合成関数の微分 $(f(g(V)))'$ 、冪乗の関数の微分 $(f(V)^{g(V)})'$ の5つの微分の手法から選択して問題を生成することが可能である。

問題生成の流れは次のようになる。

1. 全ての関数を1つにまとめたリストを作成する。この時点で a, b, c は具体的な値が決まっている。
2. 1. で作成したリストの中から、ユーザが選んだ関数の種類に合うような2つの関数を選ぶ。
3. 使用する微分の手法の中からランダムに1つ手法を選ぶ。
4. 選ばれた2つの関数と微分の手法によって、問題となる関数を生成する。
5. 生成された関数が前に生成された関数と重なっていなければ、問題として採用し、生成された関数とその関数を微分した結果をまとめたリストを作成する。
6. 上記の工程を指定した問題数になるまで繰り返す。

関数の生成において設定できる特徴とその対話文は以下のようになっている。

- “使用する乱数の範囲は+-いくつにしますか?(推奨:5) (1-50)”
 a, b, c の取り得る値の範囲を指定する。入力は1~50の数字であり、入力した数 n に対して a, b, c が各関数に付いている条件に従って $|n|$ 以下の整数が選ばれる。デフォルトは $n = 5$ であり、何も入力せず Enter/Return キーを押すとデフォルト設定になる。
- “変数を1つ指定して下さい(小文字で入力)”
 変数 V を任意に1つ指定する。変数はアルファベット小文字で入力する。
- “式の長さはいくつにしますか? (0 or 1:1項, 2:2項, 3:3項)”
 生成する関数の項数を1項から3項の間で指定する。入力は0~3の数字である。各項の生成は毎回 a, b, c を新しく設定し、かつ2つの関数と1つの微分の手法を改めてランダムに選んでいる。

- “係数を 1 に固定しますか? (0:いいえ, 1:はい)”

「はい」を選択すると、 $a = 1$ に固定して関数を生成する。

- “関数を単純にしますか? (0:いいえ, 1:はい)”

「はい」を選択すると、 $b = 1$ に固定して関数を生成する。

- “微分で使用する関数をカスタムしますか? (0:デフォルト, 1:はい)”

この対話文の後に、指定した数式の特徴、及び使用する関数と微分の手法を確認する画面が表示される。「デフォルト」を選択すると、「1～3 次の整数乗の単項式、指数関数、対数関数、三角関数」が使用する関数に、「通常微分、積の微分、商の微分」が使用する微分の手法に選ばれた状態で画面が始まる。「はい」を選択すると、使用する関数と微分の手法が全て選ばれていない状態から画面が始まる。

- “上記の設定で出力しますか?(0:出力, 1-21:各種設定変更)”

この対話文の前に指定した数式の特徴、及び使用する関数と微分の手法の一覧が画面に表示される。もし設定を変更したい場合は変更したい設定の左側にある番号を入力すると該当する設定が変更できる。なお、(1)～(6)については対話文が出現して設定を変更する形になるが、(7)～(21)については数字を入力すると「ある」または「なし」の表示が入れ替わって設定変更が完了する。

biburan では、ユーザが選んだ関数と微分の手法に限定した範囲で関数を微分する問題を生成することが可能である。使用する関数と微分の手法が細かく設定できるため、ユーザである教員が学習単元に合った微分の問題をこのソフトウェアを用いて翻案することが可能である。逆に、使用する関数と微分の手法を広く設定することによって、微分についての総合的な技術を測るための問題作成へ活用することも期待ができる。本来の使用・用途とは少し離れるが、ランダム生成という利点から場合によっては人力では思いつかないような問題を生成してくれるという面白さもこのソフトウェアは持ち合わせている。**biburan** の教育的な活用に向けては以下の追加機能に関する意見を高校教員の方から頂いた。

1 つ目は、「難易度による問題の判別機能」である。現段階では問題を生成するまではできるが、その問題がどの程度複雑で難しい微分であるのかを判定する機能まではない。生成された問題をどの程度の難易度であるかも参考にしながら選べる、さらには、難易度の制限を設けて問題を生成することができれば、目的に合った問題を選びやすくなって便利であるという意見を頂いた。微分する過程において、どの部分をもって難易度とするかという課題¹⁷もあるが、何らかの指標を検討して問題の難易度による判別の実装を目指す。

2 つ目は、「計算の途中過程を出力できる機能」である。教育的な観点から問題を選ぶ上で、実際にどのような計算過程によって微分した結果が得られるのかを確認できた方が便利であるという意見を頂いた。単に微分する程度であれば手計算でも計算過程を確認することは可能であるが、生成された問題の計算過程を一度に複数個確認するととなると、どうし

¹⁷案としては、微分する回数による判別方法を検討している。例えば、 $(f(g(V)))'$ であれば微分した結果が $f'(g(V))g'(V)$ となり微分が 2 回行われると数える。同様に $(f(V) \cdot g(V))'$ や $\left(\frac{f(V)}{g(V)}\right)'$ も 2 回となる。

でも手間がかかってしまう。必要に応じてユーザが選んだ問題の微分過程が表示できる機能の実装を目指す¹⁸。

3つ目は、「同値な正答を複数個呈示する機能」である。biburanでの正答は、生成された問題をRisa/Asirで微分した結果をそのまま表示している。小テストなどで出題した際に、想定される別解答と一緒に呈示してくれると生徒の解答を採点するのに便利であるという意見を頂いた。また、1つの問題に対して正答が複数個用意されていれば、生徒に「どの数式の表現が見やすかったり、分かりやすかったりするか」ということ考えさせる教材の題材にもなり、ソフトウェアの教育的な価値が上がるという意見も頂いた。一方で、対数関数や三角関数の同値変形の方法はいくつかあり、どの程度までの式変形を想定して同値な正答を作成するかという課題がある。この課題を検討しながら同値な正答の生成方法とその表示方法を考案する。

以上の機能の実装を視野に入れながら今後もbiburanの改良を進める予定である。

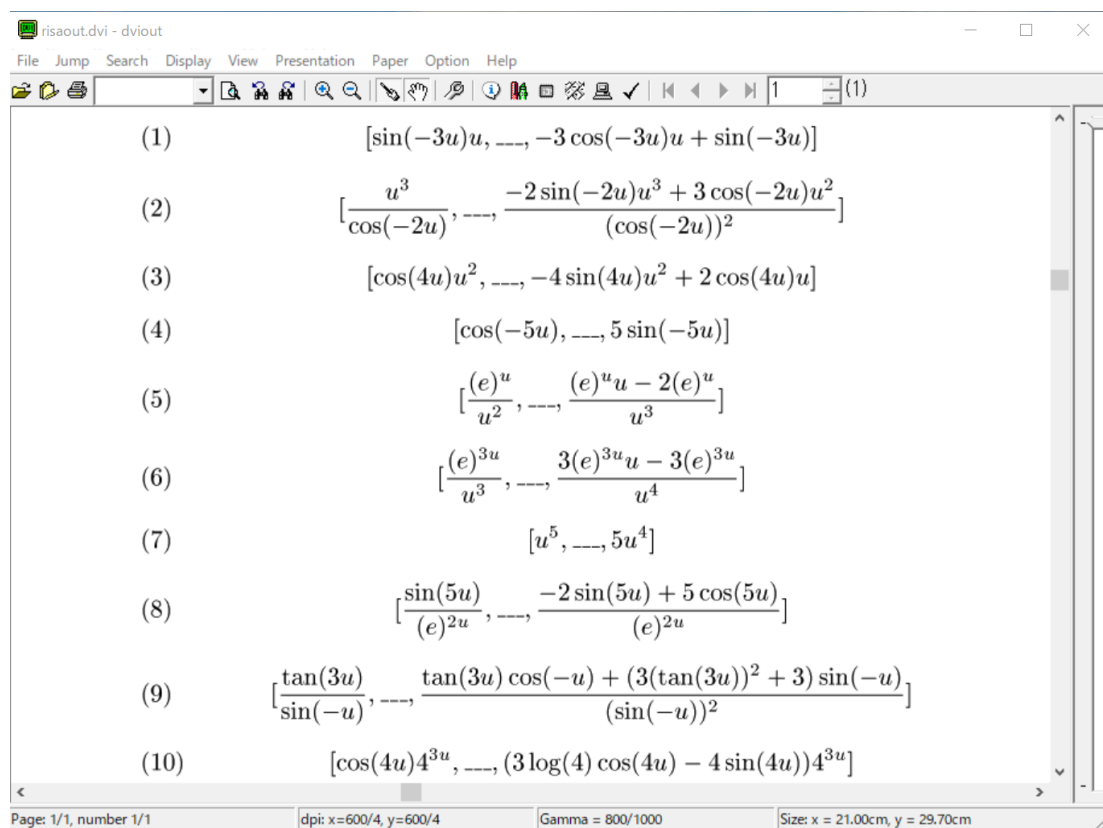


図 3: biburan にて生成された問題と正答の一覧表

¹⁸[6]には、関数の不定積分を行うユーザ関数 `os.md.integrate` のオプションとして積分の途中計算を出力する機能が実装されている。このユーザ関数を参考に微分の過程を出力する方法を検討中である。

Risa/Asir の画面での biburan の実行例 (図 3 を表示させたときの設定)

```
[2732] biburan(10);
```

使用する乱数の範囲は+-いくつにしますか? (推奨:5) (1-50):

変数を 1 つ指定して下さい (小文字で入力): u

式の長さはいくつにしますか? (0 or 1:1 項, 2:2 項, 3:3 項):

係数を 1 に固定しますか? (0:いいえ, 1:はい): 1

関数を単純にしますか? (0:いいえ, 1:はい):

微分で使用する関数をカスタムしますか? (0:デフォルト, 1:はい):

- (1) 問題生成数: 10
- (2) 使用乱数の範囲: 5
- (3) 変数: u
- (4) 項数: 1 項
- (5) 係数: 1 に固定
- (6) 関数の複雑さ: 一般

- (7) 1<->3 次の整数乗の単項式: あり
- (8) 1<-> 5 次の整数乗の単項式: なし
- (9) 1<->3 次の多項式: なし
- (10) 累乗根の式: なし
- (11) 2 乗根の式: なし
- (12) 指数関数: あり
- (13) 対数関数: あり
- (14) 三角関数: あり
- (15) 逆三角関数: なし
- (16) 双曲線関数: なし

- (17) 通常微分: あり
- (18) 積の微分: あり
- (19) 商の微分: あり
- (20) 合成関数の微分: なし
- (21) べき乗の微分: なし

上記の設定で出力しますか? (0:出力, 1-21:各種設定変更):

- (1 ~ 10) 関数の TeX 出力 (出力したい関数の番号を数又はリストで入力)
- (-1) 問題の再(追加)出力
- (-2) 出力の設定変更
- (0) プログラム終了

TeX 出力したい関数または他のコマンドに該当する括弧内の数を入力してください: [2,3]

```
[\frac{u^3}{\cos(-2u)},\frac{-2\sin(-2u)u^3+3\cos(-2u)u^2}{(\cos(-2u))^2}]
```

```
[\cos(4u)u^2,-4\sin(4u)u^2+2\cos(4u)u]
```

追加で出力する関数または他のコマンドに該当する括弧内の数を入力してください:

プログラムを終了しました

3 まとめ

本稿では, Risa/Asir で作成した「因数分解」, 「2次関数」, 「微分」それぞれの単元に対応した3種類の自動生成について概要を述べた。また, 「因数分解」と「微分」の自動生成では高校教員の方から頂いた意見についても報告した。本研究にて実装した自動生成では, 1つのソフトウェアで生成する数式の特徴を細かく指定できるため, ユーザである教員の使用・用途に合わせた様々なタイプの数式を限定的に生成することができる。さらに, 入力内容に対話式の誘導文で伝え, キーボード入力は数字またはアルファベット1文字の組み合わせで済むように設計したことで, ユーザ側の前提知識にできるだけ依らない形の自動生成の使用が可能になる。ユーザである教員が授業内容に合った数式の特徴を指定して数式を生成でき, かつ生成された数式の中からさらに教員が数式を選んで出力することができる自動生成は, 教員が毎授業行っている数式を翻案する手間を軽減する補助的なツールとしての活用が期待できる。しかし, 各ソフトウェアにて挙げた課題で示したように, 教育現場で活用できる教育ツールとしてはまだ機能が足りない。一方で, 頂いた意見の中には Risa/Asir 単体の処理では実装が厳しい機能もある。また, 根本的な問題として, 自動生成を使用するための環境設定をユーザである教員が滞りなく行えるのかという疑問が残っている。本稿では, Risa/Asir やライブラリ等が使用できる環境であることを前提に議論を進めた。ゆえに, 「幅広い教員が気軽に使用できる」という観点では, 本研究で作成した自動生成はユーザに一定の下準備を求めるという点で, この観点をまだ達成しておらず今後の大きな課題となる。これらを踏まえ, 今後は以下の課題について取り組む。

- 本稿で述べた3種類の自動生成について, 各ソフトウェアで挙げた追加機能の実装を行う。また, より効率的な数式の生成方法や生成された数式及び付随する情報の見やすい呈示方法についても考察する。
- 本研究で作成した自動生成をブラウザ上で使用できるように整備を行い, フリーの教育ツールとして自動生成を提供することを目指す。

- 現状では、自動生成によって呈示された数式の中から教員が教育効果が見込めるものを選ぶといったソフトウェアの活用の流れになっている。教員が行う「教育効果が見込めるものを選ぶ」という作業をソフトウェア側で担うことができないかと筆者は考えている。「どの観点に立って教育効果が見込めるというのか」といった指標作りの課題もあるが、数式を生成する際に一定の教育効果も踏まえた上で生成が可能になる自動生成の開発を目指す。

また、まだ作成していない分野での自動生成の開発も引き続き行っていく。

4 謝辞

本研究を始めるにあたって、修士課程在籍中に Risa/Asir を紹介してくださった城西大学の 大島利雄 教授に感謝を申し上げます。また、ソフトウェアの改良にあたり意見をくださった 明治大学の 阿原一志 教授を始め、東京都高等学校数学教育研究会と東京理科大学数学教育研究会の方々に感謝を申し上げます。

参考文献

- [1] 中村泰之, 数学 e ラーニング 数式解答評価システム STACK と Moodle による理工系教育, 東京電機大学出版局, 2010.
- [2] 長坂耕作, Moodle XML Question Generator for Python, 京都大学数理解析研究所 講究録 No.2142 “数学ソフトウェアとその効果的教育利用に関する研究”, 2019.
- [3] 竹島卓, Rias/Asir の開発, 数式処理 J.JSSAC Vol.12, No.1, p.9-22, 2005.
- [4] 野呂正行, 高山信毅, Risa/Asir ドリル, 2012.
- [5] 野呂正行, 下山武司, 竹島卓, Risa/Asir マニュアル, 2014.
<http://www.math.kobe-u.ac.jp/OpenXM/Current/doc/index-doc-ja.html>
- [6] 大島利雄, os_muldif, Risa/Asir のライブラリー, 2020.
<https://www.ms.u-tokyo.ac.jp/~oshima/index-j.html>