

数式自動採点システム STACK の 行列問題における解答入力方法改善の取り組み

白井 詩沙香^{1,a)} 福井 哲夫^{2,b)}

概要：本研究の目的は、e-Learning のオンラインテストにおいて数式そのものを解答形式として扱える数式自動採点システムにおける数式入力方法を改善することである。現在の数式入力インタフェースは、学習者にとって使い易いとは言えず、数式自動採点システムの普及のための課題の1つとなっている。我々はパソコン環境における数式自動採点システムの数式入力インタフェースの研究を継続的に行っており、本稿ではその一環として行った行列問題における行列入力方法の改善の取り組みについて報告する。

SHIRAI SHIZUKA^{1,a)} FUKUI TETSUO^{2,b)}

1. はじめに

近年、e-Learning におけるオンラインテストで、学習者が数式そのものを解答できる数式自動採点システムが注目されている [1]。従来のオンラインテストの解答形式は、正誤判定や多肢選択、数値入力などに限定されていたが、数式自動採点システムは、数式処理システム (Computer Algebra System, 以下、CAS) を活用することにより、数式による解答を実現した。

数式自動採点システムの代表的なものに英国で開発された STACK がある [2]。STACK は LMS に Moodle, CAS に Maxima を使っており、Moodle の小テストとして簡単に利用できる。中村らにより日本語化作業が行われており [3]、すでに国内においてもいくつかの大学で活用されている [4], [5]。しかし、STACK のさらなる普及のためには解決すべき問題があり、その1つに数式入力の際の学習者への負担が挙げられる [6]。

現在、STACK をはじめとする数式自動採点システムは主にパソコン環境で利用されており、代表的な数式入力方法は CAS コマンドによるテキストを使った入力方法と GUI エディタによる入力方法の2つである。CAS コマン

ドによる入力方法は初学者にとっては負担が大きく、GUI エディタによる入力方法は敷居が低いものの、キーボードとマウスを併用する操作がわずらわしく、繰り返し学習には向かない。

我々はこれまでに、STACK をはじめとする数式自動採点システムのパソコン環境における数式入力方法を改善するために、数式曖昧表記変換方式 [7] を実装した数式入力インタフェース MathTOUCH [8] を提案してきた。本学学生に MathTOUCH を実装した STACK を使い、リメディアルを目的とした平方根問題の練習を行った結果、ユーザビリティに関する満足度が改善できることを確認した [9]。

本稿では、さらに行列問題を題材に STACK における数式入力方法の改善を行い、検証した結果について報告する。本研究で扱う行列問題とは、主に成分表示によるベクトルや行列の基本計算力を養うための学習問題である (表 1 参照)。教育対象は「線形代数」など大学の数学で扱われる行列計算や高校数学 II で扱われる「平面と空間のベクトル」もその範疇に入る。特に理工系の大学では、ベクトル方程式や固有値問題などを解くために重要な概念であるため、工学・物理学・統計学など応用範囲は広い。それゆえ、行列を自動採点できる学習支援システムの意義は大きい。

2. STACK 標準の行列入力方法

STACK による標準の数式入力方法は、Maxima コマンドをテキストエリアに入力するものである (以下、Maxima コマンド方式)。本来、CAS コマンドによる入力方法は、自分が入力した数式がイメージしづらく、入力ミスに気づ

¹ 武庫川女子大学大学院 生活環境学研究所
Graduate School of Human Environmental sciences, Mukogawa Women's University

² 武庫川女子大学
Mukogawa Women's University

a) mw419064 @ mukogawa-u.ac.jp

b) fukui @ mukogawa-u.ac.jp

表 1 本研究で想定している行列問題例

項番	問題例	学習者の解答
1	行列 $A = \begin{bmatrix} -4 & 6 \\ 1 & \frac{1}{2} \end{bmatrix}$, $B = \begin{bmatrix} 2 & -5 \\ \frac{1}{2} & -1 \end{bmatrix}$ に対し, $A+B$ を計算せよ.	$\begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$
2	行列 $x = \begin{bmatrix} 4 \\ 5 \\ 3 \\ 2 \\ 2 \end{bmatrix}$ に対し, $\frac{1}{2} * x$ を計算せよ.	$\begin{bmatrix} 2 \\ \frac{5}{2} \\ \frac{3}{2} \\ 1 \\ 1 \end{bmatrix}$
3	行列 $A = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$ に対し, 転置行列 A^T を求めよ.	$\begin{bmatrix} 4 & 3 & 1 \end{bmatrix}$

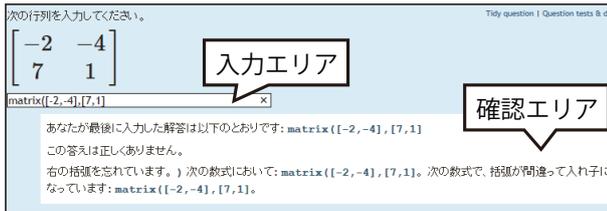


図 1 Maxima コマンドによる入力

きづらいという課題を抱えているが, STACK ではコマンドを入力するとリアルタイムで 2 次元形式の数式が表示される Validation エリア (以下, 確認エリア) により, ユーザフレンドリーになっている. また, 文法を間違えて入力した場合は, この確認エリアには図 1 のような括弧やアスタリスク忘れなどを指摘するコメントが表示される.

さて, STACK で扱える行列問題は, 答えを成分表示による行列で表せるものに限られている (例えば, $(AB)^{-1} = B^{-1}A^{-1}$ のような行列の成分情報を抽象化した代数的表記は扱えない). したがって, 行列を入力するには Maxima の “matrix()” コマンドを使用する. たとえば, 2 行 2 列の行列の場合は次のようになる.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} := \text{matrix}([1, 2], [3, 4]) \quad (1)$$

このような表現はプログラミングの文法に似ており, 丸括弧と角括弧, さらにカンマが入れ子になるため, 誤入力が多発する可能性が高い. 書式のヒントを表示させることも可能であるが, プログラミング未経験の学生にとっては, 表示されていたとしても入力時の負担は大きいと予想される.

そのため STACK では, 図 2 のような行列の要素のみを入力させる行列用の入力インタフェースが用意されている. このインタフェースは, 予め行列成分の位置に配置されたテキストエリアをクリックし, 各成分の数値 (あるいは数式) のみを入力すればよいため, 入力時の負担や誤入力は軽減できる. しかし, この方法では行列の答えのサイズを知らせることになるため, 教育効果を阻害することにもなる [10].

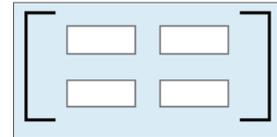


図 2 STACK の行列専用インタフェース

3. MathTOUCH による行列入力方法

3.1 数式入力インタフェース MathTOUCH

MathTOUCH とは, 2012 年に福井より提案された数式曖昧表記変換方式を実装した数式入力インタフェースである. 本方式は, CAS コマンドのように数式に表示されない指示記号 (x^2 入力時のキャレット記号など) を使わず, 普段数式を読むような曖昧表記で数式要素を入力し, 日本語の仮名漢字変換のように変換をすることで数式を入力できる特徴をもつ. 例えば, $\frac{1}{a+1}$ を入力したい場合は, まず “1/a+1” と入力し, Space キーで変換を開始する. するとスラッシュ記号が変換対象となり, WYSIWYG で変換候補が表示される. “1/a+1” から出来る数式は $\frac{1}{a+1}$ と $\frac{1}{a} + 1$ があり, CAS コマンドでは丸括弧でオペランド範囲を明示し, この 2 つの数式を区別する. 一方, MathTOUCH では, 丸括弧など表示されない指示記号は入力しないため, 変換時に同時に表示される赤線を左右矢印キーで調整することで, 図 3 の Step3 のようにオペランド範囲が指定できる. オペランド範囲を調整するとリアルタイムで変換候補も変わるため, 調整後は Space キーまたは上下矢印キーで所望の数式を選択する (図 3 の Step4). Enter キーで分数記号の変換を確定し, 次の要素の変換に移る. 残りの要素の変換が不要であれば, Shift キーを押しながら Enter キーを押すことで一括確定ができ, 変換操作は終了となる. その他にも, $\sin x$ は “sinx” (CAS コマンドでは $\sin(x)$), $\sqrt{2}$ は “root2” (CAS コマンドでは $\text{sqrt}(2)$) などと入力し, 同様の手順で変換を行い入力ができる.

したがって, 入力文字列は曖昧な表記でよく, 数学学習とは関係のないコマンドを覚える必要がないため, CAS を使ったことのない初学者の負担を軽減できると期待できる.

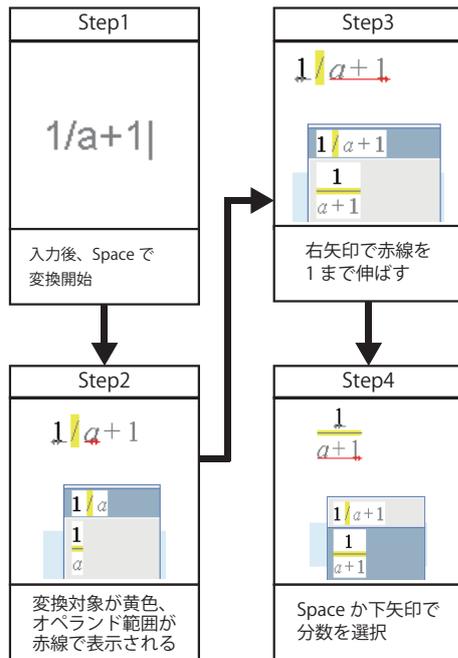


図3 オペランドの調整方法

3.2 行列の入力手順

開発当初、MathTOUCHは $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ やWordなどを使った数学教材作成の支援を主な目的としていたため、行列入力機能は存在したが、枠線の種類を選択できるオプションといったe-Learningでは必要のない機能が含まれていた[11]。そこで、本研究では数式自動採点システム向けに入力操作を直感的に、且つ変換操作をシンプルに行えるようMathTOUCHの行列入力機能を見直した。

改良したMathTOUCHによる行列入力手順を図4に示す。MathTOUCHでは、第1要素の式入力に続いて、 \rightarrow (右矢印キー)を入力すると1列追加され、カーソルが次の列へ移動する。同様に、 \downarrow (下矢印キー)により1行追加され、カーソルが次の行へ移動し、直感的に入力できるようになっている。したがって、数式(1)と同じ行列を入力するには、“ $1 \rightarrow 2 \downarrow 3 \rightarrow 4$ ”と入力する。始めの“ \rightarrow ”で2列目となり、“ \downarrow ”で2行目に移るので、各要素は出来上がりと同じ2行2列に配置されていることを確認しながらキー入力ができる。そして、Spaceキーで変換を開始する。各要素に分数やべき乗など変換が必要な数式が含まれていた場合は各要素の変換に移り、変換が不要であれば、STACKに数式が出力され、確認エリアにも行列が表示される。

3.3 行列各要素の数式変換方法

各要素に変換が必要な数式が含まれていた場合の変換手順を表1の項番1に示した解答行列の例で述べる。この行列の場合、変換作業が必要な要素は“-2”と“ $\frac{3}{2}$ ”と“- $\frac{1}{2}$ ”の3つである。Spaceキーで変換を開始すると、まずは(1,1)成分である-2が変換対象となる(図5のStep2)。マイナ

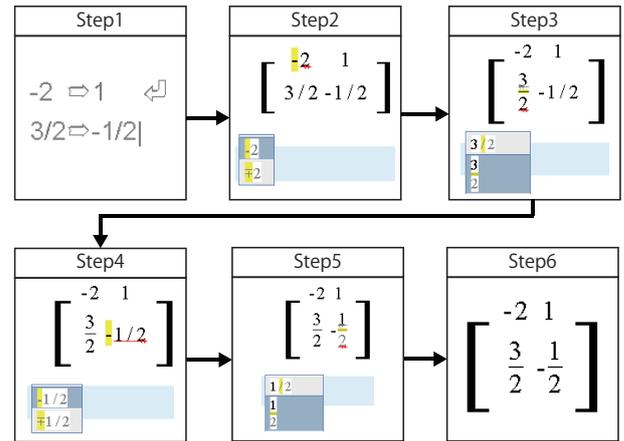


図5 MathTOUCHによる行列各要素の数式変換手順

ス記号は“-”以外に“干”が候補として表示される。今回は、-のままでよいため、そのままEnterキーで確定する。確定すると(2,1)成分である“ $3/2$ ”に変換対象が移る。図5のStep3に示すように、変換候補として $\frac{3}{2}$ や $3/2$ などが表示されるので、分数を選択し、Enterキーで確定する。すると最後に(2,2)成分の“- $1/2$ ”に変換対象が移る。左から順に変換が行われるため、まずはマイナス記号が変換対象となる。演算子に変換対象の場合、演算子の種類(今回は“-”か“干”)に加え、演算子が作用する部分式の範囲によっても、変換後の数式が異ってくる。そのため、マイナスの作用範囲の赤線を“1”から“ $1/2$ ”まで伸ばし、分数全体にかかるようにする必要がある。指定後、Enterキーで候補を確定すると、変換対象がスラッシュ記号に移る(図5のStep5)。数式候補から分数を選択し、Enterキーを押すと確定され、入力完了となる。

3.4 数式要素の修正方法

MathTOUCHには、パソコンで文書を修正する際と同様に、確定した数式の一部を削除したり、数式要素を追加挿入できる機能がある。また、独自の機能として、一度確定した記号や演算子を再変換し直すことができ、ミス確定を直すことができる。

ここでは、表1の項番1の(2,2)成分を“- $\frac{1}{2}$ ”と入力してしまった場合を例に再変換手順を述べる。

修正が必要な場合は、まずMathTOUCHの入力エリアにフォーカスを戻す必要がある。MathTOUCHにおけるフォーカスは、式の挿入位置を表すカーソルとその左側に接する削除または再変換対象を表す対象枠(図6のStep2)で表示されるため、修正作業を開始できる。今回は、マイナス記号の修正が必要なため、まずは対象枠をマイナス記号に移動させる必要がある(図6のStep3)。対象枠の表示が必要な理由は、数式の場合、文書と違い対象が1文字とは限らないからである。矢印キーを押すことで、カーソルと共に対象枠を所望の要素に移動できる。

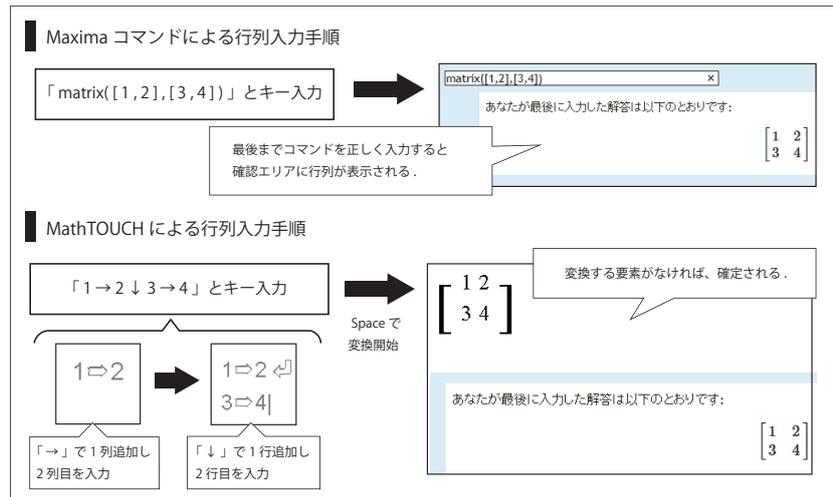


図 4 MathTOUCH による行列入力手順

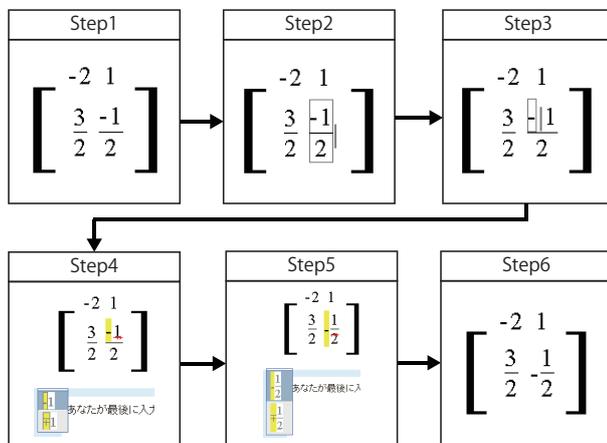


図 6 MathTOUCH による数式要素の修正手順

マイナス記号が選択されたら、Space キーを押すことで再変換が開始される。この場合、マイナス記号の作用範囲が分数全体に指定されていなかったことが原因であるため、右矢印キーで赤線を「2」まで伸ばすことで、マイナスの作用範囲を分数全体に修正することができる。修正後は、通常の変換時と同様に確定作業を行い、全ての要素が確定できれば、修正作業が完了となる。

変換忘れや変換候補の選択ミスの場合、再変換機能を利用することで再入力の手間を省くことが可能である。

4. 従来方式との比較実験

4.1 実験概要

MathTOUCH によって STACK の行列入力時の負担を軽減できるか検証するために、同一被験者に Maxima コマンド方式と MathTOUCH を使い行列を入力してもらい、その入力時間と主観満足度の差を比較した。

被験者は数式の入力方法について資料で説明を受けてから、練習として STACK 上で 9 種類の数式を入力し、その後、表 2 に示す行列を 3 種類入力するタスクテストを実施

表 3 入力時間（秒）と分散分析結果

	Maxima	MT	主効果	
入力方式	-	23.9(7.1)	17.4(6.1)	10.1**
利用経験	Maxima	22.3(2.3)	18.6(8.7)	1.8
	MT	25.2(9.5)	16.3(2.9)	

() 内は $SD^{**} p < .01$

する。タスクテストを終えた被験者は、再び資料でもう一方の数式入力方法について説明を受け、同様の手順で練習とタスクテストを行う。両種類のタスクテストが終わった後でアンケートに回答し、実験は終了である。なお、各入力方法の練習およびタスクテストを行う順番は約半数毎に入れ替えた。

入力時間は、入力を始めてから確認エリアに正しい行列が表示されるまでとし、修正行動も含めた時間を録画ソフトで計測した*1。主観満足度の評価は、測定終了後に 5 段階の評定尺度法に基づくアンケートで行った（5 が最良）。質問項目は「学習しやすさ」、「効率性」、「記憶しやすさ」、「間違えにくさ・修正しやすさ」、「再利用意向」の 5 項目である。

被験者は本学情報メディア学科の情報技術系のゼミに所属する 3 年生 13 名で、プログラミングには比較的慣れている学生である。また、いずれも 2 年次に STACK を使い平方根の計算練習を行っており、STACK の操作にも慣れている（MathTOUCH 利用者：7 名、Maxima コマンド利用者：6 名）。したがって分析の際は、測定時間、主観満足度ともに、以前の利用経験による結果への影響も確認する必要がある。このため、利用経験と今回の入力方法を独立変数とした 2 要因の分散分析を行った。

4.2 結果と考察

平均練習時間は、MathTOUCH が 4.61 分、Maxima コ

*1 PC: HP ProBook 4740s, OS: Windows 7 Professional(32bit)

表 2 タスクテストで使用した行列

問題番号	行列	Maxima コマンド	MathTOUCH
1	$\begin{bmatrix} -3 & -9 \\ 6 & 1 \end{bmatrix}$	matrix([-3,-9],[6,1])	-3 \rightarrow -9 \downarrow 6 \rightarrow 1
2	$\begin{bmatrix} 41 & -3 & 30 \\ 35 & -15 & 0 \end{bmatrix}$	matrix([41,-3,30],[35,-15,0])	41 \rightarrow -3 \rightarrow 30 \downarrow 35 \rightarrow -15 \rightarrow 0
3	$\begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$	matrix([-2,1],[3/2,-(1/2)])	-2 \rightarrow 1 \downarrow 3/2 \rightarrow -1/2

表 4 主観満足度アンケートと分散分析結果

質問項目	Maxima	MT	主効果
学習しやすさ	3.46(1.27)	3.92(0.95)	0.71
効率性	3.00(1.29)	4.00(1.00)	2.59
間違えにくさ・修正しやすさ	3.62(0.96)	3.54(1.20)	0.06
記憶しやすさ	4.08(0.76)	4.08(1.04)	0.00
再利用意向	3.69(1.18)	3.85(1.34)	0.05

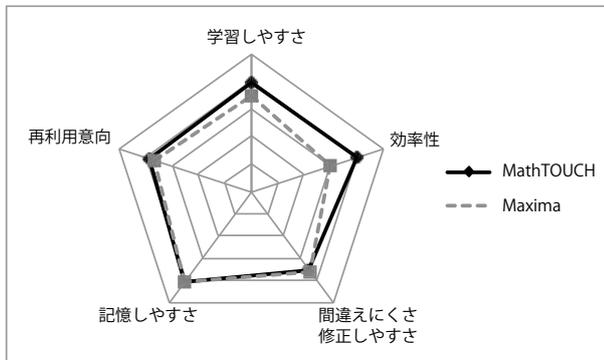


図 7 主観満足度結果のレーダーチャート

マンド方式が 4.03 分となり、MathTOUCH の方が練習にはやや時間がかかる結果となった。今回の被験者は全員がプログラミング経験者で処理系の考え方には慣れているため、Maxima コマンド方式の練習は比較的スムーズに進んだ。しかし MathTOUCH の場合は、全く新しいインタフェースであるため、行列の入力方法の前に、数式の変換や修正といった MathTOUCH の概念に慣れる必要があったため、時間がかかったと考えられる。

しかしタスクテストでは、平均入力時間は Maxima コマンド方式が 23.9 秒、MathTOUCH が 17.4 秒となった(表 3 の 2 行目)。分散分析を行った結果、以前の利用経験と入力方法には交互作用は見られず、入力方法で有意な主効果が見られた ($F(1,11)=10.1, p<.01$)。過去の利用経験に関係なく、MathTOUCH の方が行列を有意に速く入力できることが判る。したがって、概念に慣れる必要があるため、初回の練習にはやや時間を要するが、プログラミング経験者であったとしても、MathTOUCH を使用することで効率よく行列を入力できることが確認できた。

次に、主観満足度のアンケート結果を表 4 に、5 つの指標の平均値をレーダーチャートで表したものを図 7 に示す。学習しやすさ、効率性、再利用意向の 3 項目で Math-

TOUCH の方が満足度が高い結果となった。しかし、分散分析を行った結果はいずれの項目にも有意な差は見られなかった。また、間違えにくさ・修正しやすさについては、MathTOUCH の方がやや満足度が低い結果となった。自由記述による感想を確認したところ、“修正が難しかった”という意見が多かった。実際に動画を確認したところ、問題 3 のマイナス記号がつく分数の要素でマイナス記号の作用範囲の指定がよくわからず、“ $-\frac{1}{2}$ ”と変換してしまい、その後、修正をスムーズに行えない被験者が見られた。この場合の修正方法は、“-1”のマイナス記号を消去し、正しい位置にマイナス記号を挿入する方法とマイナス記号を再変換し、赤線の範囲を分数全体に変更する方法(3.4 節参照)の 2 通りがあるが、再変換機能を使おうとした被験者が多かった。しかし、赤線の意味を正しく理解できていない被験者は少なく、スムーズに修正を行えない結果となった。丸括弧による範囲の指定に慣れているプログラミング経験者にとって、赤線によるオペランド範囲の指定や再変換などの操作に慣れるには時間を要することが判った。

全体の感想は、MathTOUCH は“直感的に入力できたので簡単だった”、“ミスも少なく、覚えやすいのでスムーズに出来た”といった感想があった。一方で、“消すのは面倒だった”、“変換が少ないと便利だけど多いと迷う気がする”といった意見もあった。Maxima は“変換の回数が少なく、打ち易かった”、“括弧でくくることで変換の手間が少なくてよかった”という意見や“入力数が多いとかなり手間がとられた”、“[] や () を多用されると間違いが多くなる”といった意見があった。MathTOUCH により、Maxima コマンド方式の短所である効率性の問題を解消することはできたが、現在の MathTOUCH では要素毎に変換を行う必要があるため、入力時間としては速く入力できているにも関わらず、有意差が出るほど主観満足度に差が出る結果には至らなかった。また、修正しやすさについては、どのような数式であっても修正を容易にできるような改善が必要がある。

5. まとめと今後の課題

本研究では、STACK における行列の入力方法を改善することを目的に、MathTOUCH の行列入力機能を改良し、STACK 標準のインタフェースとの比較検証実験を行った。

MathTOUCHによって、入力効率を上げることができたが、主観満足度については従来のMaximaコマンド方式との間に有意な差は見られなかった。その主な原因は要素毎の変換操作と修正操作であることが明らかになった。

今後の課題は、変換効率を上げることと修正操作のさらなる改善を行うことである。また、今回は被験者がプログラミング経験者中心で偏っているため、プログラミングとは無関係な学習者に対しても検証を行っていくことが課題である。

謝辞 本研究の実験にご協力いただきました本学情報メディア学科の中村勝則先生に心より感謝申し上げます。なお、本研究はJSPS 科研費（課題番号：26330413）の助成を受けたものです。

参考文献

- [1] CIEC 研究会：第100回研究報告書，CIEC 第100回研究会報告書 (2014).
- [2] Sangwin,C.:STACK(online), available from <http://stack.bham.ac.uk/> (2014.07.16).
- [3] Ja STACK.org : Ja STACK.org(online), 入手先 <http://ja-stack.org/>(2014.07.16).
- [4] 谷口哲也, 根本洋明, 五十嵐正夫: 数学教育における Moodle と STACK の利用, 数理解析研究所講究録, No.1865, pp.121-129 (2013).
- [5] 亀田真澄, 宇田川暢: 大学の数学教育に対する主体的な学びとなる学修環境作り, ICT 活用教育方法研究, Vol.16, No.1, pp.36-41(2013).
- [6] 中村泰之, 大俣友佳, 中原敬広: STACK の問題作成ツールの開発と STACK3 に向けて (第21回日本数式処理学会大会報告), 数式処理, Vol.19, No.2, pp.33-36 (2013).
- [7] 福井哲夫: 数式のインテリジェントな線形入力方式, 数理解析研究所講究録, Vol.1780, pp.160-171 (2012).
- [8] MathTOUCH プロジェクト, 福井研究室: 数式エディタ MathTOUCH(online), 入手先 <http://math.mukogawa-u.ac.jp/> (2014.07.16).
- [9] 白井詩沙香, 仲村裕子, 福井哲夫: STACK による数学 e ラーニングシステムにおける数式入力方法改善の試み, 情報処理学会研究報告. Vol.2014-CE-123, No.14, pp.1-7 (2014).
- [10] 中村泰之: 数学 e ラーニング数式評価システム STACK と Moodle による理工系教育, 東京電機大学出版局, 東京 (2010).
- [11] 福井哲夫: MathTOUCH Web(ver.0.60) ユーザーズマニュアル (online), 入手先 <http://math.mukogawa-u.ac.jp/document/manual/manualWeb060.pdf> (2014.07.16).