

論文

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^{1,b)}

受付日 2014年8月5日, 再受付日 2015年2月14日,
採録日 2015年3月22日

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やすいとはいえず, 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. 本研究では, 大学数学で扱われる「行列」に着目し, 数式を普段読むように曖昧表記で入力し, 仮名漢字変換のように変換することで数式を入力できる数式入力インタフェース MathTOUCH によって, 数式自動採点システムの1つである STACK における行列入力方法の改善ができるかを評価した. その結果, 従来の数式入力インタフェースより有意に速く行列を入力できることを確認した. 一方, 行列入力のユーザビリティに関する主観満足度の調査では, 従来方式との有意差は見られなかった.

キーワード: 数式自動採点システム, 行列問題, 数式入力, STACK, ユーザインタフェース

Improving the Math Input Method for Matrices for Use in Linear Algebra on STACK

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

Received: August 5, 2014, Revised: February 14, 2015,
Accepted: March 22, 2015

Abstract: The purpose of our study is to improve a math input method on computer aided assessment (CAA) systems for mathematics. Current mathematics CAA systems enable users to directly input mathematical expressions to evaluate their answers with mathematical expressions automatically. However, it cannot be said that it is easy for novices of the mathematics to use the math input method. In this paper, we focus on the case of matrix input method for use in linear algebra, which is taught at many universities. We have investigated whether our proposal mathematics interface, named MathTOUCH, improves the input procedure for learners to answer in matrix form on STACK. As a result, we found that students could input matrix expressions with MathTOUCH significantly faster than with the current mathematics interface on STACK. Our questionnaire found no significant differences in regards to subjective satisfaction between the two interfaces.

Keywords: computer aided assessment systems for mathematics, matrix, input for mathematical expression, STACK, user interface

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

が数式そのもので解答できる数式自動採点システムが目ざされている [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

¹ 武庫川女子大学生生活環境学部
School of Human Environmental Sciences, Mukogawa
Women's University, Nishinomiya, Hyogo 663-8558, Japan
a) shirai@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
Table 1 Question examples involving matrices.

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 $x = \begin{bmatrix} 4 \\ 5 \\ 3 \end{bmatrix}$ に対し, $\frac{1}{2} * x$ を計算せよ.	$\begin{bmatrix} 2 \\ \frac{5}{2} \\ \frac{3}{2} \\ 1 \end{bmatrix}$
3	行列 $A = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix}$ に対し, 転置行列 A^T を求めよ.	$\begin{bmatrix} 4 & 3 & 1 \end{bmatrix}$

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 標準の数式入力インタフェースは, 数式で解答する際の学習者への負担が指摘されている [6], [7].

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, 2 次元形式の数式をイメージしにくい負担が大きく, 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のわずらわしさがある.

我々はこれまでに, 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方法を改善するために, 数式曖昧表記変換方式 [8], [9], [10] を実装した数式入力インタフェースを数式自動採点システムで扱えるように最適化し, 提案してきた [11]. この提案数式入力インタフェースを STACK に実装し, 大学 3 年生に数学学習で使ってもらった結果, 数式入力インタフェースのユーザビリティに関する満足度を一部改善できることを確認した [11].

本研究では, さらに行列問題を題材に STACK における行列入力方法の改善を試み, 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, 表 1 のような主に成分表示による行列の基本計算力を養うための学習問題である. もちろん行列成分で扱う数式は表 1 のような有理数に限定されるものではなく, 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を想定している. ただし本研究では, STACK が扱える行列表現の範囲で, 行列入力時の操作性に限定して議論する.

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

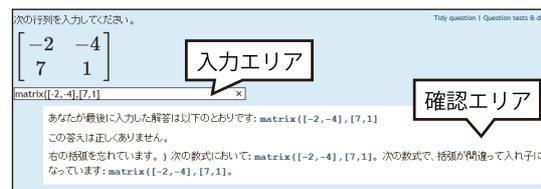


図 1 Maxima コマンドによる入力

Fig. 1 Screenshot of Maxima command interface on STACK.

形式入力). 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にくく, 入力ミスに気づきにくいという課題をかかえているが, 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 (たとえば, $(AB)^{-1} = B^{-1}A^{-1}$ 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ある. 特に文献 [12] によると, 括弧のミス発生率はある程度習熟した後もあまり減少しないことが報告されており, 注意が必要である.

そのため STACK では, 図 2 のような行列の成分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, あらかじ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

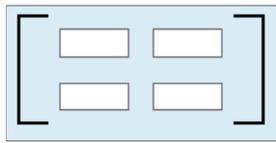


図 2 STACK の行列専用インタフェース

Fig. 2 Specialized interface for matrix inputs on STACK.

ことにもなる [13].

3. 提案数式入力インタフェースによる行列入力方法

3.1 数式曖昧表記変換方式

2012年に福井が提案した数式曖昧表記変換方式*1 [8], [9], [10]とは、初期入力文字列に表記されない区切り記号や演算子の指示が不要で、数式変換エンジンと数式辞書（変換キーと数式要素と順序の対応データ）によって、インテリジェントに候補を算出し、WYSIWYGでインタラクティブに数式を構築する方式である。ここで初期入力文字列のルール（数式曖昧表記法）は、所望する数式に含まれる数式要素に対応するキー文字（列）を読む順番に区切りなく並べるだけである。一般に、所望する数式の要素を読む順番に並べたものを (e_1, e_2, \dots, e_N) とし、 e_i ($i = 1, \dots, N$) に対応するキー文字（列）を k_i とすると、数式曖昧表記文字列は “ $k_1 k_2 \dots k_N$ ” で表される。たとえば、 $-\frac{a^2}{3}$ （マイナス、アルファ、2乗、割る、3）のための文字列は “a2/3” となる。このように数式曖昧表記法は数学記号（イタリック体、ボールド体、ローマン体、ギリシャ文字）などがほぼアルファベット1文字に対応しており、数式構造を決める演算子（べき乗*2や割り算など）が作用するオペランド範囲の区切り記号（括弧やコンマなど）が不要なため、CASコマンド形式に比べ短い文字数で済むという特徴を持つ。しかし、数式変換時に演算子のオペランド範囲をそのつど指定する必要がある。

数式曖昧表記変換方式の操作の流れを図3に示す。たとえば、 $\frac{1}{a+1}$ を入力したい場合は、数式要素 $(e_1, e_2, \dots, e_5) = (1, \text{割る}, a, \text{プラス}, 1)$ の数式曖昧表記文字列は “1/a+1” となり、これをテキストフィールドに入力し（図4のStep 1）、Spaceキーで変換開始指示をする。次に、図3のように (e_1, e_2, \dots, e_5) の順に要素をインタラクティブに確定していく。ただし、整数1などは第1候補しかないため自動確定され、次へ進む。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（図4のStep 2）。演算子の場合はそのオペランド範囲が赤のアンダーラインで強調表示され、左右矢印キーで調整することで、図4

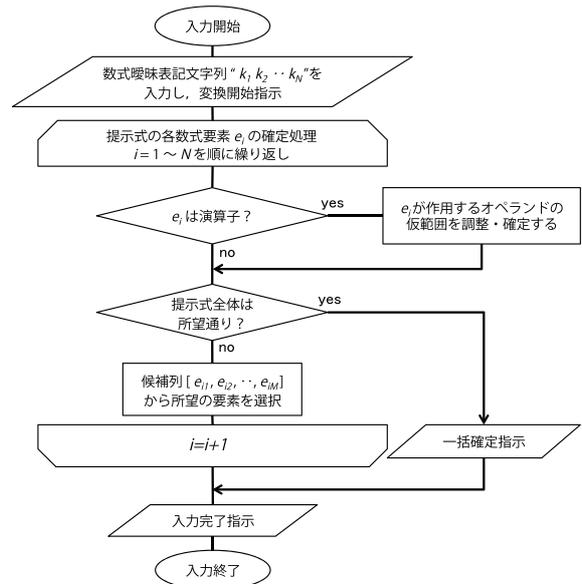


図 3 数式曖昧表記変換方式の入力操作手順

Fig. 3 The procedure for inputting mathematical expression.

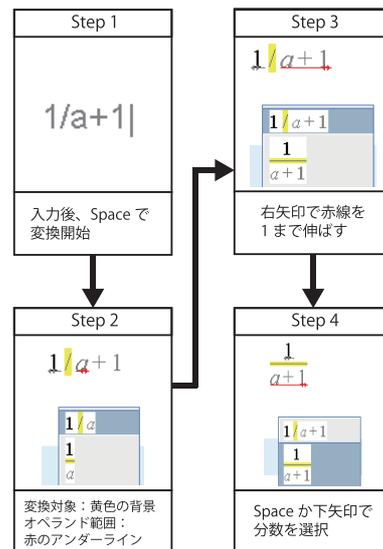


図 4 数式曖昧表記変換方式による入力例

Fig. 4 The steps for inputting mathematical expression with our method.

の Step 3 のように範囲を指定できる。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4の Step 4）。Enter キーで分数の表現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。

この方式を実装した数式入力インタフェースを Math-TOUCH と呼んでいる [14], [15]。MathTOUCH で扱える数式の範囲は、LaTeX で扱える数式をほぼカバーしており、大学の教養数学程度の数式は入力可能である。また、数式の行列配置も可能で、式 (2) のようなベクトル・行列

*1 参考文献 [8] ではインテリジェントな線形入力方式と呼んでいた。
 *2 べき乗は 2次元表記の数式に表記されない演算子として扱い、数式曖昧表記法では指示文字は不要（Maxima では “^” を使う）である。したがって、 x^2 の場合は “x2” となる。このとき、変換候補列には $[x2, x^2, x_2, ^2, x_2]$ がある。

の成分表記や式 (3) のような連立方程式の列成分の配置位置 (左寄せ, 中央など) の調整が行え, さらには式 (4) のように行列配置の罫線枠をつけることもできる [14], [16].

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix} \quad (2)$$

$$\begin{cases} y = 3x - 2 \\ y = x + 1 \end{cases} \quad (3)$$

$f(x)$	x	x^2
$f'(x)$	1	$2x$
$\int f(x)dx$	$\frac{x^2}{2}$	$\frac{x^3}{3}$

(4)

また, 次のような編集機能も備えている [17].

編集機能 1) 確定した数式の一部の削除機能

編集機能 2) 数式要素や演算子構造の追加挿入機能

編集機能 3) 記号や演算子構造の再変換機能

ここで, 編集機能 1), 2) の演算子構造の追加・削除機能とは, たとえば式 (5) の左辺から右辺へ分数構造 (分数の表現記号) の挿入を行ったり, 逆に右辺から左辺へ分数構造を削除できたりする機能のことである.

$$a b \Leftrightarrow \frac{a}{b} \quad (5)$$

また, 編集機能 3) の再変換機能とは, 一度確定した記号や演算子を変換候補選択状態 (たとえば図 4 の Step 2 の状態) に戻し, 別の候補に変更できる機能のことである. たとえば式 (6) の分数の表現記号を再変換して, 分母のオペランド範囲 (記号はそのまま分数構造のみ) を変更することもできる.

$$\frac{1}{a^2 + 1} \Leftrightarrow \frac{1}{a^2} + 1 \quad (6)$$

3.2 STACK 用 MathTOUCH インタフェースの実装

本研究では, MathTOUCH を STACK の行列問題学習で使用するために, MathTOUCH で構築した行列を Maxima コマンド形式で出力するためのモジュールを作成した. 具体的には, MathTOUCH の持つ内部表現である数式木構造を解釈し, 列ごとにカンマを, 行区切りとして括弧を追加し, Maxima コマンドに変換している.

また, 3.1 節で述べたように, MathTOUCH は教師の教材作成や研究者の論文作成にも使用できるような機能を持つが, 数学学習者にとっては, 不要な機能が混乱を招く可能性もある. そこで, 行列入力機能から列成分の配置位置の指定および行列配置の罫線枠指示の 2 項目を削除した. たとえば式 (3) のように連立方程式を綺麗に左寄せにしたり, 式 (4) のような罫線を表示したりする機能は, 学習者の解答のためには不要と考え, 指定できないようにした.

なお, 2012 年に発表したアプリケーション版 MathTOUCH には, 保持する数式辞書の優先順位情報と履歴式

に関する学習機能を有する. しかし, 本研究の STACK 向け MathTOUCH はオンラインシステムのため, 学習情報をローカルストレージに保存できるように対応しておらず, セッションが切れるたびに初期状態に戻る. そのため, 本研究の 4 章で述べる評価実験では学習機能は機能しない.

以上の変更を行った行列問題学習向け MathTOUCH を, STACK のプラグイン機能を利用し実装した. もし, STACK の利用者が本研究で提案する MathTOUCH のプラグインファイル群を STACK の所定のディレクトリに保存すれば, 自動で STACK が認識し, 問題作成画面の数式入力タイプを指定するメニューから MathTOUCH を選択できるようになる.

3.3 行列の入力手順

行列問題学習向け MathTOUCH では, 一般に式 (7) のような $m \times n$ 行列が入力できる. 各行列成分 a_{ij} ($i = 1, \dots, m, j = 1, \dots, n$) は 3.1 節で述べた手順で大学教養数学程度までの様々な数式が入力可能である.

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad (7)$$

数式曖昧表記変換方式における行列の入力手順は, 各行列成分 a_{ij} ($i = 1, \dots, m, j = 1, \dots, n$) に対応した数式曖昧表記文字列を所望する行列と同じ m 行 n 列に配置されたテキストフィールドに入力する点が異なる. すなわち, 式 (7) の場合は $m \times n$ 個のテキストフィールドに数式曖昧表記文字列を入力することになる. 列の移動には左右の矢印キー「←」, 「→」を使い, 行の移動には上下の矢印キー「↑」, 「↓」を使う. ただし, 標準では行列サイズは 1 行 1 列の状態となっており, カーソルが最終列のテキストフィールドにある状態で右矢印キーを押すとテキストフィールドが 1 列分追加され, 新しい列に対応する行列成分の数式曖昧表記文字列を入力できる. 同様に, 最下行のテキストフィールドで下向き矢印キーを押すとテキストフィールドが 1 行分追加される. もし, 行または列を追加しすぎた場合は図 5 の Step 1 に示すような行または列区切りマークを Delete キーで削除すればよい. ただし, 削除される行または列のテキストフィールドに入力文字列が残っている場合は削除できないので, 先に対応成分の文字列を削除してから実施する. このように, 変換を開始する前であれば行列のサイズを変更できる. さて, 行列の入力操作手順は図 6 のようになっており, まず, すべての行列成分 a_{ij} ($i = 1, \dots, m, j = 1, \dots, n$) に対応した数式曖昧表記文字列を入力したら, Space キーで数式変換を開始する. 各行列成分の数式構築の順番は 1 行ごとに 1 列目から行い, 図 6 のように a_{ij} の数式変換・確定操作を図 3 に

従って繰り返す。すべての成分が確定すると行列全体を囲む括弧が自動挿入される。もし、この時点で修正すべき行列成分が見つかった場合は、上下・左右の矢印キーを使って対応箇所へ移動し、編集機能 1)~3) によって修正することもできる。しかし、いったん確定した行列のサイズは変更できない。

実際の行列入力例を表 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、変換作業が必要な成分は“-2”と“ $\frac{3}{2}$ ”と“- $\frac{1}{2}$ ”の 3 つである。まず、右矢印キーおよび下矢印キーを押して 2 行 2 列のテキストフィールドを追加し、各成分の数式曖昧表記文字列を図 5 の Step 1 のように入力する。ここで、改行記号は行の、⇒記号は列の区切りマークを表す。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 Step 2)。マイナス記号は“-”以外に“干”が候補として表示される。今回は、“-”のままでもよいため、そのまま Enter キーで確

定する。確定すると (2,1) 成分である“ $\frac{3}{2}$ ”に変換対象が移る。図 5 の Step 3 に示すように、変換候補として $\frac{3}{2}$ や $\frac{3}{2}$ 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“- $\frac{1}{2}$ ”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。演算子が変換対象の場合、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、マイナス記号のオペランド範囲を示す赤のアンダーラインを“1”から“ $\frac{1}{2}$ ”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 Step 5)。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。

4. 従来方式との比較実験

4.1 実験概要

MathTOUCH によって STACK の行列入力を改善できるか検証するために、同一被験者に Maxima 形式入力と MathTOUCH を使い行列を入力してもらい、それぞれの入力時間および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、STACK 上でタスクと同種の数式を入力練習し、その後、表 2 の E1~E3 の 3 行列を入力するタスクテストを実施する。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操作性に関する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ごとに入れ替えた。

今回の実験では行列配置特有の入力操作性を比較することが目的であるため、行列成分の数式を極力シンプルにする必要がある。もし、成分式を複雑にすれば、行列の入力しやすさについて、要因の切り分けが難しくなると判断した。したがって、タスクテストに使用した行列 E1~E3 は行列成分が変数や関数などを含まない有理数のみとした。

被験者は武庫川女子大学情報メディア学科の 3 年生 13 名である。いずれも 2 年次に STACK を使い平方根の計算

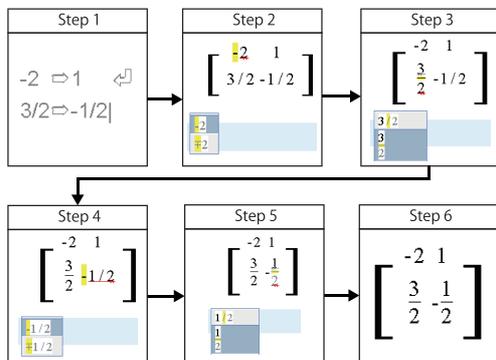


図 5 MathTOUCH による行列の数式変換例

Fig. 5 Conversion steps for a matrix using MathTOUCH.

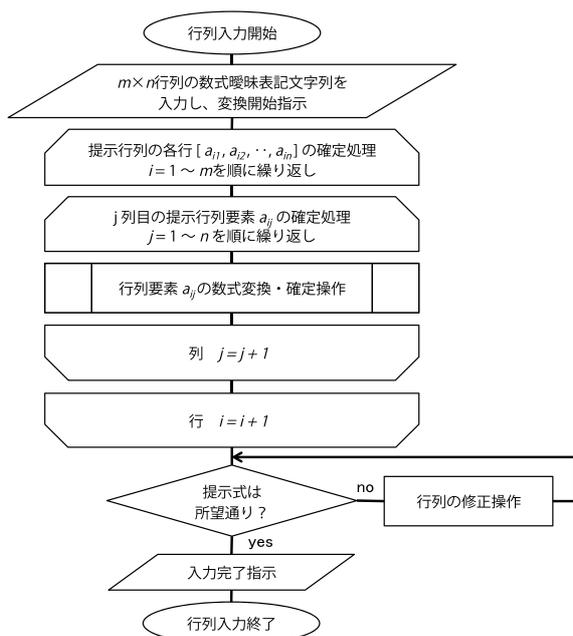


図 6 行列の入力操作手順

Fig. 6 The procedure for inputting a matrix.

表 2 タスクテストで使用した行列
Table 2 Matrices used for our task test.

問題番号	タスク行列
E1	$\begin{bmatrix} -3 & -9 \\ 6 & 1 \end{bmatrix}$
E2	$\begin{bmatrix} 41 & -3 & 30 \\ 35 & -15 & 0 \end{bmatrix}$
E3	$\begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$

表 3 インタフェースのユーザビリティに関する主観満足度のアンケート結果

Table 3 The questionnaire results regarding subjective satisfaction with the usability of each interface.

(質問の意図) アンケートの質問文	平均 (SD)	
	MathTOUCH	Maxima 形式入力
(理解しやすさ) 行列の入力手順をマスターするのは簡単だ	3.92(0.95)	3.46(1.27)
(効率性) 行列を早く, スムーズに入力できる	4.00(1.00)	3.00(1.29)
(間違えにくさ・修正しやすさ) 入力ミスが少なく, 間違えても修正は簡単だった	3.54(1.20)	3.62(0.96)
(記憶しやすさ) 練習後, 入力方法を覚えていた	4.08(1.04)	4.08(0.76)
(再利用意向) 今後もし行列を入力する機会があったときはまたこのエディターを使ってみたいですか?	3.85(1.34)	3.69(1.18)

5段階評定尺度:「とてもそう思う」(5点), 「ややそう思う」(4点), 「どちらともいえない」(3点), 「あまりそう思わない」(2点), 「全然そう思わない」(1点)

練習を行っている。過去の数式入力インタフェースの利用経験が実験結果に影響しないよう, MathTOUCH の利用経験者を 7 名, Maxima コマンドの利用経験者を 6 名とほぼ均等になるよう配慮した。

4.2 測定方法

評価する入力時間とは, 行列 E1~E3 それぞれの入力を始めてから確認エリアに「正しく」行列が表示されるまでの時間の平均値である。学習のうえでは, たとえば解答式が $-\frac{1}{2}$ でも $\frac{-1}{2}$ でも数学的に等価であるが, ここで「正しく」とは, 2次元表記した数式構造まで一致するよう入力することを要求する。この要求は一見, 本研究の目的である数式自動採点システムの良さ(数学的に等価などどんな数式も評価できる点)を損なうように思われるが, 本実験の論点は行列入力インタフェースによる入力時間の比較にあるので, 達成すべきタスクを統一する必要がある, あえてこのように設計した。また, 提示した行列をそのまま入力してもらう理由は, 数学の問題を実際に解いてもらった場合, 被験者の学力差も結果に影響するからである。本実験では入力ミスの修正操作も入力時間の一部と考え, 行列の入力時間を修正行動も含めて計測した。各被験者の行列 E_i ($i = 1, 2, 3$) の入力時間は操作画面をビデオキャプチャソフト Bandicam [18] を使い 0.1 秒単位で録画し, 計測した。

両インタフェースによる行列入力後の満足度の評価は, 測定終了後に行った 5 段階の評定尺度法に基づく操作性に関するアンケートで測定した。質問項目はユーザビリティ特性 [19] を参考に「理解しやすさ」, 「効率性」, 「記憶しやすさ」, 「間違えにくさ・修正しやすさ」に関する質問と総合評価として「再利用意向」を問う 5 項目とした。それら 5 項目に関する質問文を表 3 に示す。回答は「とてもそう思う」(5 点), 「ややそう思う」(4 点), 「どちらともいえない」(3 点), 「あまりそう思わない」(2 点), 「全然そう思わない」(1 点) からあてはまるものを選ぶよう求めた。また, 全体の感想についても自由記述してもらった。

実験の公平性を保つため, タスク前の操作説明・事前練習においてインタフェースによらず統一したのは次の 3 点

表 4 練習で使用した数式

Table 4 Practice mathematical expressions used in this study.

数式番号	基本練習	数式番号	行列練習
B1	$\frac{1}{2}$	P1	$\begin{bmatrix} -2 & -4 \\ 7 & 1 \end{bmatrix}$
B2	$-\frac{3}{4}$	P2	$\begin{bmatrix} 31 & -5 & 20 \\ 39 & -17 & 1 \end{bmatrix}$
B3	$-\frac{2}{5}$	P3	$\begin{bmatrix} -7 & 2 \\ \frac{5}{3} & -\frac{3}{7} \end{bmatrix}$
		P4	$\begin{bmatrix} 4 & -7 \\ -6 & 5 \end{bmatrix}$
		P5	$\begin{bmatrix} 61 & -3 & -32 \\ 42 & -11 & -34 \end{bmatrix}$
		P6	$\begin{bmatrix} 0 & -2 \\ \frac{1}{6} & -\frac{3}{5} \end{bmatrix}$

である。

- (i) 実施環境の統一
- (ii) 説明項目 (a), (b), (c) を統一
 - (a) 操作画面要素・エリアの役割
 - (b) 表示マーク(カーソル, 枠, 強調色など)の意味
 - (c) 有理数(表 4 の B1, B2, B3) の入力操作手順
- (iii) 練習問題(表 4 の P1~P6) の統一

実施環境は, ラップトップコンピュータ(HP ProBook 4740s, OS: Windows 7 Professional (32 bit))を使った。キーボード部分のキーピッチは 19mm, キーストロークは 2.2mm, JIS 標準準拠・OADG 準拠配列である。テンキーの使用は被験者の任意とし, 付属のタッチパッドは使用せず, 標準のマウスを使うよう指示した。各ソフトウェアのバージョンは STACK は 3.1, Moodle は 2.5.2+, Internet Explorer は 9 である。測定画面は, 上部にタスクの数式を表示し, その直下にインタフェースを表示した。

両者のインタフェースに対する説明項目は上記(ii)の(a), (b), (c)だけにして, 説明の仕方の良し悪しによる影響を極力少なくするために, 表 2 のタスクテストと同種の 6 つの行列 P1~P6 (表 4) を正しく入力できるまで練習してもらい, 理解レベルの統一を図った。

表 6 タスク行列入力時のキー打鍵数

Table 6 The key entering numbers of the matrices with each interface.

インタフェース	打鍵数			行列 E3 の場合の打鍵キー列 (最短の場合)														
	E1	E2	E3															
MathTOUCH	12	20	21	- 2	→	1	↓	3 / 2	→	- 1 / 2	SP	E	SP	E	→	E	SP	E
Maxima	21	29	25	matrix ([- 2 , 1] , [3 / 2 , - 1 / 2])														

SP : Space キー, E : Enter キー, → : 右矢印キー, ↓ : 下矢印キー

表 5 入力時間の平均値 (秒) の結果

Table 5 A summary of mean input times (in seconds).

	MathTOUCH	Maxima 形式入力
平均入力時間	17.4	23.9
SD	6.1	7.1

4.3 結果と考察

実験の結果、いずれのインタフェースもすべての被験者がタスク達成不能に陥ることもなく、すべてのタスクにおいて MathTOUCH が最大 69.4 秒、Maxima 形式入力が最大 66.5 秒以内に行列の入力を完了していた。

行列の平均入力時間は、MathTOUCH が 17.4 秒、Maxima 形式入力が 23.9 秒であった。結果を標準偏差とともに表 5 に示す。正規性の判定に基づき、Wilcoxon の符号付き順位検定を行った結果、有意な差が見られた ($Z=2.447, p<.05$)。

今回のタスクテストに使用した行列は、2 行 2 列または 3 列で、行列成分が有理数のため、やさしいものである。単純な予測として、行列成分の数式入力の手間が同じ場合、行列サイズが大きくなれば、それに比例して入力の負担も大きくなる。そこで、両者の行列入力操作におけるキー打鍵数を比較したところ、表 6 のようになった。参考のために、行列 E3 (表 2) に対する MathTOUCH および Maxima 形式入力における打鍵キー列を示しておいた。E3 の場合、行列成分を表す文字はどちらも 10 文字であるが、MathTOUCH の場合は 3 行列 E1~E3 の総打鍵数が 53 文字、Maxima 形式入力の場合は 75 文字であった。この比は 1.42 倍であり、平均入力時間の比 1.37 倍とほぼ近い値であることが分かる。一般に式 (7) のような $m \times n$ 行列の場合の行列成分式以外の打鍵文字数を検討してみると、MathTOUCH の場合はミスのない最短入力打鍵数は式 (8) で表される。

$$(mn - 1) + \mathcal{O} \tag{8}$$

ここで、 $(mn - 1)$ は初期入力時の行と列の追加矢印キー打鍵の総数で、 \mathcal{O} は $m \times n$ 成分に含まれる変換すべき成分の変換指示キー (矢印キーや Space, Enter など) の総数であり、行列サイズだけでは決まらない。たとえば、行列 E3 の場合は表 6 に示したように、 $\mathcal{O} = 15$ 打鍵となる。Maxima 形式入力の場合の行列成分以外 (すなわち

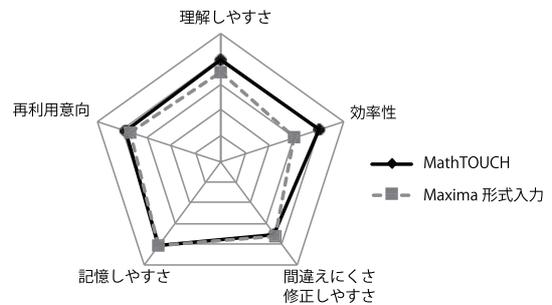


図 7 主観満足度結果のレーダーチャート

Fig. 7 A spider chart showing the results of the subjective satisfaction questionnaire.

“matrix()”, “[]”, “,”) の打鍵数は式 (9) で与えられる。

$$m(n + 2) + 7 \tag{9}$$

両者の差 (9)-(8) は $2m + 8 - \mathcal{O}$ となり、行数が増えれば Maxima 形式入力の方が打鍵数が増え、MathTOUCH は成分の変換指示操作のための打鍵数 \mathcal{O} が増えれば負担が増えると予想される。たとえば、4 行 4 列の場合は、MathTOUCH の変換指示打鍵数 $\mathcal{O} > 16$ で、Maxima 形式入力より負担が増えることになる。一方、式 (1) で指摘したように、Maxima 形式入力の場合は行列サイズが増えると誤入力が増える傾向もあり、打鍵数によるコストだけでは比較することができない。したがって、行列サイズが大きいタスクについては、別途検証を行う必要がある。

次に、主観満足度のアンケート結果を表 3 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。一般に 5 段階評定尺度に対する平均値は 3.6 といわれている [19]。MathTOUCH のユーザビリティ 5 項目に対する主観満足度の評定尺度平均は、「間違えにくさ・修正しやすさ」の項目以外は 3.6 より高い評価であった。特に、「学習しやすさ」、「効率性」、「再利用意向」の 3 項目で MathTOUCH の方が Maxima 形式入力より満足度が高い結果となった。一方、「間違えにくさ・修正しやすさ」については、MathTOUCH の方が満足度が低い結果となった。しかし、Wilcoxon の符号付き順位検定を行った結果、いずれの項目にも有意な差は見られなかった。

以上の結果から、行列入力操作に関して MathTOUCH は Maxima 形式入力と変わらない満足度で、Maxima 形式入力より有意に速く入力できるといえる。

しかし、「間違えにくさ・修正しやすさ」の点で、自由

記述の不満意見から、行列操作に関して無視できない意見 (13名中2名)として、「行列確定後に行列のサイズ変更ができない」点が指摘された。ほかに13名中3名が演算子構造の確定操作のミスに関する指摘があったが、行列の入力手順とは無関係であり、文献 [11] で詳しく扱っているため、ここでは議論しない。

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、数式自動採点システム向けに MathTOUCH の行列入力機能を最適化し、STACK 標準のインタフェースとの比較検証実験を行い、操作性を評価した。実験の結果、行列入力操作に関して MathTOUCH は Maxima 形式入力と変わらない満足度で、Maxima 形式入力より有意に速く入力できることが明らかになった。

本研究により、数式自動採点システムにおいて、数式曖昧表記変換方式を使った行列入力方法の提供ができ、STACK を使った学習者の数式入力方式の選択肢の幅を広げることができた。しかし上記の結果は、提案数式入力インタフェースを実装した STACK を使い、実際の行列問題を学習した場合の学習効果を改善できるかどうかは別の問題である。

今後の課題は、アンケート結果より示唆された行列確定後の行列サイズの変更機能の追加と、よりサイズの大きな行列でも検証を行うこと、そして提案インタフェースを使った行列問題学習を実施し、学習効果について評価していくことである。

謝辞 本研究の実験にご協力いただきました武庫川女子大学生活環境学部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.J.: STACK (online), available from <http://stack.bham.ac.uk/> (accessed 2014-05-26).

[3] Ja STACK.org: Ja STACK.org (online), available from <http://ja-stack.org/> (accessed 2014-05-26).

[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] Sangwin, C.J.: Computer Aided Assessment of Mathematics Using STACK, *Proc. ICME*, Vol.12 (2012).

[7] 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).

[8] 福井哲夫: 数式のインテリジェントな線形入力方式, 京都大学数理解析研究所講究録, Vol.1780, pp.160-171 (2012).

[9] 福井哲夫: 数式のインテリジェントな線形入力方式と評価, 数式処理, Vol.18, No.2, pp.47-50 (2012).

[10] 福井哲夫: 線形文字列変換による対話型数式入力方式の効果, 京都大学数理解析研究所講究録, Vol.1785, pp.32-44 (2012).

[11] 白井詩沙香, 福井哲夫: 数式自動採点システム STACK における数式入力方法の改善, コンピュータ&エデュケーション, Vol.37, pp.85-90 (2014).

[12] Sangwin, C.J. and Ramsden, P.: Linear syntax for communicating elementary mathematics, *Journal Article Journal of Symbolic Computation*, Vol.42, No.9, pp.920-934 (2007).

[13] 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).

[14] 福井哲夫: インテリジェントな対話型数式デジタル化技術による数式エディタの開発, 京都大学数理解析研究所講究録, Vol.1865, pp.10-17 (2013).

[15] 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/> (参照 2014-05-26).

[16] 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> (参照 2014-05-26).

[17] 福井哲夫: 線形文字列変換による機械学習型数式入力インタフェースと編集機能の設計, 情報処理学会全国大会講演論文集, Vol.2012, No.1, pp.1-3 (2012).

[18] 株式会社グレテックジャパン: Bandicam (online), 入手先 <http://www.gomplayer.jp/bandicam/> (参照 2014-12-27).

[19] ヤコブ・ニールセン: ユーザビリティエンジニアリング原論, 東京電機大学出版局 (2002).



白井 詩沙香 (正会員)

2007年武庫川女子大学情報メディア学科卒業。2012年同大学院生活環境学研究所修士課程修了。2015年同大学院同研究科博士課程修了。博士(情報メディア学)。現在、武庫川女子大学生活環境学部生活環境学科助教。ヒューマンインタフェース、学習支援システムに興味を持つ。コンピュータ利用教育学会、日本数式処理学会、日本教育工学会、ACM 各会員。



福井 哲夫 (正会員)

1985年大阪市立大学大学院理学研究科前期博士課程修了。1989年同大学院後期博士課程単位取得満期退学。同年博士(理学)。1989年詫間電波工業高等専門学校講師。1993年同高専助教授。1994年武庫川女子大学生活環境学部情報メディア学科助教。2005年同大学教授、現在に至る。数式ヒューマンインタフェースに関する研究に従事。日本物理学会、日本数学会、日本数式処理学会各会員。