

# 数式予測アルゴリズムを実装した数式入力インターフェース MathTOUCHの試作と評価

白井 詩沙香<sup>1,a)</sup> 仲村 裕子<sup>2</sup> 福井 哲夫<sup>1,b)</sup>

**概要:** 本研究は、数式自動採点システムにおける数式入力方法の改善を目的に、数式曖昧表記変換方式を実装した数式入力インターフェース MathTOUCH のインテリジェント化を試みたものである。これまでに我々は提案インターフェースを数式自動採点システム STACK に実装し、教育利用の効果を評価した。その結果、従来方式と比べて 1.2~1.6 倍速く数式入力ができ、インターフェースの操作性に関する主観満足度で有意に高い評価を得たが、オペランド範囲の指示操作について改善の必要性が示唆された。本稿では、数式全体を予測し、その候補を提示するための数式予測アルゴリズムを実装した MathTOUCH を試作し、被験者実験によりパフォーマンスを評価した。その結果、実装前の MathTOUCH と比べて数式入力操作にかかる時間が約 1.4 倍速い傾向にあることを示した。

SHIZUKA SHIRAI<sup>1,a)</sup> YUKO NAKAMURA<sup>2</sup> TETSUO FUKUI<sup>1,b)</sup>

## 1. はじめに

国内外の多くの教育機関において、Blackboard Learn, Moodle, Canvas LMS などの LMS (Learning Management System) を活用した e ラーニングが導入されている。LMS には、コンテンツ配信・管理、電子掲示板、受講者の学習履歴・成績管理など、様々な基本機能があるが、特に、学習者の解答を自動的に採点し、即時フィードバックするオンラインテスト機能は、有効な e ラーニングの特徴として挙げられており [1]、受講生の理解度の確認や知識定着などに活用されている。

従来のオンラインテストで扱える問題形式は、多肢選択問題や空所補充問題などであったが、近年、数式の正誤判定を実現したオンラインテストシステム（以下、数式自動採点システム）が実用化され、注目を集めている [2], [3], [4]。数式の自動採点が可能となったことで、受講生の正確な理解度の計測が可能となり、また、提供する問題の幅も広げることができるようになった [3]。一方で、いくつかの課題も明らかになっており、その一つとして解答の数式入力方

法が煩わしく、学習者の負担となっていることが指摘されている [5], [6]。

これまでに我々は、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を目的に、数式を普段読むような曖昧な文字列（以下、数式曖昧表記文字列）で入力し、2 次元表記に変換することで所望の数式入力ができるインターフェース「MathTOUCH」を提案してきた [7]。ユーザビリティの観点から評価実験を行った結果、オペランド範囲を変換時に指定する点について、改善の必要性が示唆された [8], [9]。

本研究では、これらを改善するために、数式予測アルゴリズム [10] によって数式全体を予測し、その候補を提示する仕組みを実装した MathTOUCH を開発し、評価を行ったので報告する。

## 2. 先行研究

### 2.1 数式自動採点システムと従来の数式入力方法

2017 年現在、日本国内で利用されている数式自動採点システムとして、Edinburgh 大学の Chris Sangwin 氏によって開発され、Moodle のプラグインとして利用できる STACK [11]、Maple 社の Maple T.A. [12]、大阪府立大学で開発された Math on Web [13] などがある。これらのシステムは、数式の正誤判定に数式処理システム (Computer Algebra System, 以下、CAS) を活用しており、STACK は

<sup>1</sup> 武庫川女子大学生生活環境学部  
School of Human Environmental Sciences, Mukogawa  
Women's University

<sup>2</sup> 武庫川女子大学附属中学校・高等学校  
Mukogawa Women's University Junior & Senior High School

a) shirai@mukogawa-u.ac.jp

b) fukui@mukogawa-u.ac.jp

オープンソースの CAS である Maxima[14] を, Maple T.A. は Maple[15] を, Math on Web は webMathematica[16] を利用している\*1.

解答の数式入力方法は, いずれのシステムも各システムが使用している CAS の文法に従って入力しなければならない. 例えば,

$$\frac{3 - 5\sqrt{26}}{2} \quad (1)$$

の場合, Maxima 形式では“(3-5\*sqrt(26))/2”と入力する. オペランドの範囲や演算子等, システムが正しく認識できるような文法構造になっており, 逆に, 初学者が直感的に理解することは難しいと指摘されている[19].

一方, Maple T.A. ではテキストベース入力方式が苦手なユーザ向けに, 数式構造や数学記号のテンプレートを選択し, 数式を作図するように入力する GUI 型エディタも用意されている. 数式は 2 次元形式で表示されているが, 数式構造を把握してから順番に構造テンプレートを選択しなければならず, ユーザに負担がかかること[19]や構造の修正が難しいことが指摘されている[20].

## 2.2 MathTOUCH による数式入力方法

MathTOUCH は, 数式を普段読むように入力し, 変換することで数式入力ができる数式曖昧表記変換方式[9]を実装した数式入力インタフェースである[21]. 数式を普段読むように入力するとは, 数式に表示されていない記号(例えば, 暗黙積を表すアスタリスク記号や冪乗を表すキャレット記号など)は入力せず, 数式要素に対応するキーワードのみを読む順番に並べることを指す. 例えば, 数式(1)を MathTOUCH で入力する場合は, “3-5root26/2”のように数式に表示されていないオペランド範囲や演算子を省いた曖昧な状態で入力できる. この曖昧部分については, その後の変換フェーズで, 対話的にユーザの指示を受けて構成要素毎に確定していく(以下, 構成要素逐次確定方式).

数式  $\frac{2}{x^2-5}$  を例に, 具体的な入力手順を図 1 に示す. まず, Step 1 に示すように, 数式曖昧表記文字列を入力する\*2. 次に, Space キーを押して変換を開始すると, 左の数式要素から順番に変換が開始される. Step 2 に示すように, 変換対象は黄色背景で示され(今回はスラッシュ記号), その直下に変換候補一覧が表示される. また, 演算子記号が変換候補の場合はオペランド範囲を示す赤線も表示される. この赤線の範囲は左右矢印キーで調整可能である(Step 2 および Step 3 を参照). 変換候補と赤線の範囲を調整した後, Step 4 に示すように変換候補から所望の演算子を下矢印キーで選択し, Enter キーで確定すると, 変

換対象が次の要素に移る(Step 5). それ以降も Step 6~8 のように各要素毎にオペランド範囲の調整と変換候補の選択を行い, 最後の要素まで確定すると完成となる. 完成した数式は, LaTeX, MathML, JPG, EPS, Maxima, Mathematica, Maple 形式で出力可能である[22].

## 2.3 MathTOUCH のユーザビリティ評価

2014 年に MathTOUCH の操作性を評価するために, 高校生から大学生を対象に数式入力パフォーマンス実験を行った[9]. 評価実験では, 被験者に STACK 上で, MathTOUCH, Maxima 形式によるテキストベース UI, STACK で使用実績のある GUI 型エディタの DragMath[23] のいずれかを使って 6 つの数式を入力してもらい, その平均入力時間や主観満足度の比較をした.

実験の結果, Maxima 形式によるテキストベースや DragMath の GUI ベースといった従来の入力方式に比べて, 1.2~1.6 倍短い時間で入力できることを明らかにした. また, 数式入力後のユーザビリティに関する主観満足度アンケート調査の結果, 「効率性」, 「記憶のしやすさ」, 「再利用意向」の 3 項目において, 5 段階評定平均値が従来方式と比べ有意に高い結果となった.

一方で, アンケートの自由記述による感想より, 赤線の範囲(オペランド範囲)の指示操作方法が難しいという指摘があった. 動画分析を行った結果, 入力ミスの原因にもなっていることが明らかとなり, 改善の必要性が示唆された.

## 3. 数式予測アルゴリズムを実装した MathTOUCH の試作

2.3 節で述べたユーザビリティ評価実験により明らかになったオペランド範囲の指示操作方法に関する問題点を改善するために, 図 2 に示すように数式曖昧文字列から数式全体の候補を提示できる数式予測アルゴリズムを実装した MathTOUCH を開発した. 本章では, 数式予測アルゴリズムおよびその実装について述べる.

### 3.1 数式予測アルゴリズム

提案インタフェースに実装するアルゴリズムは, 2015 年に福井が提案した数式予測アルゴリズム[10]で, 構造化パーセプロトンによる機械学習技術を応用し, 数式曖昧表記文字列  $s$  から最適候補の数式  $y_p$  の予測を行えるようにしたものである.

本アルゴリズムでは, 数式の各構成要素に出現頻度に応じたスコアを付与する. そして, 数式  $y$  を構成する要素のスコアの合計値を  $Score(y)$  で表す. これにより, 数式曖昧表記文字列  $s$  から生成可能な全ての数式  $(Y(s))$  のうち, スコアの合計値が最大となる数式  $y_p$  を最適な予測候補として提示する.

\*1 WeBWorK[17] や Numbas[18] のように正誤評価に外部の CAS を利用していないシステムもある.

\*2 図 1 の Step 1 の文字列は, 英語読みに近いものであるが, “x2-5\2” や “x2-5bunno2” のように分母, 分子の順に入力することが可能である.

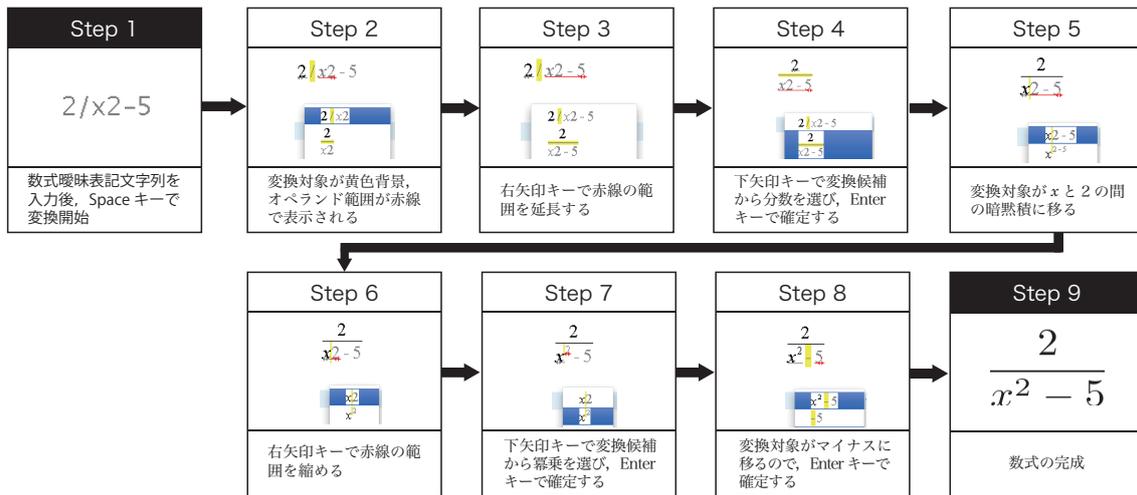


図 1 従来型 MathTOUCH (構成要素逐次確定方式) による数式変換の流れ

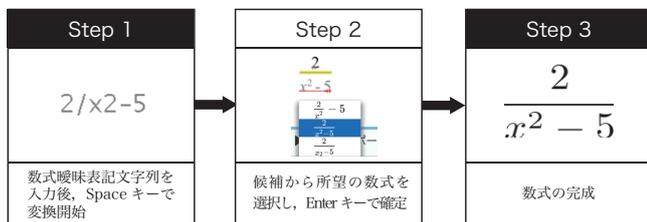


図 2 数式予測アルゴリズムを実装した MathTOUCH による数式変換の流れ

$$y_p \text{ s.t. } \text{Score}(y_p) = \max\{\text{Score}(y) | y \in Y(s)\} \quad (2)$$

本研究では、スコア (パラメタ) 決定のために表 1 に示す高校の教科書に出現する 4000 個の数式サンプルを教示データとするパーセプトロン学習を採用した。予備実験 [24] により、数式曖昧表記文字列が長いと候補算出に時間がかかりすぎるのが分かったため、文字数は 16 文字以内とした。サンプルのうち、500 個をテストデータ、残りを訓練データとする 8 分割の交差検定を行った結果、1 ベスト候補 (スコアが 1 位の候補) の正解率は 51.5%、3 ベスト候補 (スコアが 3 位以内のいずれかの候補) の正解率は 72.4%、10 ベスト候補 (スコアが 10 位以内のいずれかの候補) では 85.2% となった。

### 3.2 数式予測アルゴリズムの実装

前節で述べた数式予測アルゴリズムを実装した MathTOUCH の操作画面を図 3 に示す。図 2 に示したように、従来の MathTOUCH と同様に数式曖昧文字列  $s$  を入力し、Space キーで変換を開始した後、 $s$  から予測される数式候補  $y_p$  が提示される。

例えば、図 2 のように数式  $\frac{2}{x^2-5}$  を入力したい場合は、数式要素 (2, 割る, x, 2, マイナス, 5) の数式曖昧表記文字列は “2/x2-5” となり、これをテキストフィールドに入力し (図 2 の Step 1)、Space キーで変換開始指示をする。次に、図 2 の Step 2 のように数式全体を対象とした変換候補が

表 1 数式サンプル (高等学校数学 I, A, II, B, III より)

数学分野	データ数
代数学	1058
幾何学	702
ベクトル	250
集合・論理	75
整数論	208
確率・統計	267
三角関数	469
指数・対数	37
数列	178
関数と極限	257
微分・積分	499
合計	4000

提示されるため、所望の数式を Space キーまたは上下矢印キーで選択、Enter キーで確定し、変換操作は終了となる。この変換方式を以下では全体候補選択方式、この方式を実装した MathTOUCH を全体予測型 MathTOUCH と呼ぶ。

もし、変換候補の中に所望の数式が含まれない場合は、Shift + Space キーを押すことで、図 1 に示す従来通りの構成要素逐次確定方式に切り替えることができるようにした。

## 4. 評価

### 4.1 実験概要

2.3 節で述べた構成要素逐次確定方式の問題点を 3.2 節の全体候補選択方式によって改善できるか評価するために、STACK 上で従来型 MathTOUCH または全体予測型 MathTOUCH を使って数式を入力してもらい、数式入力タスクに対するタスク達成時間および主観満足度を測定し、比較した。

数式入力タスクテストは、先行研究 [9] と同じ手順を用いる。まずはじめに、被験者は数式入力方法について資料で説明を受けてから、STACK 上で基本的な数式要素 (表 2

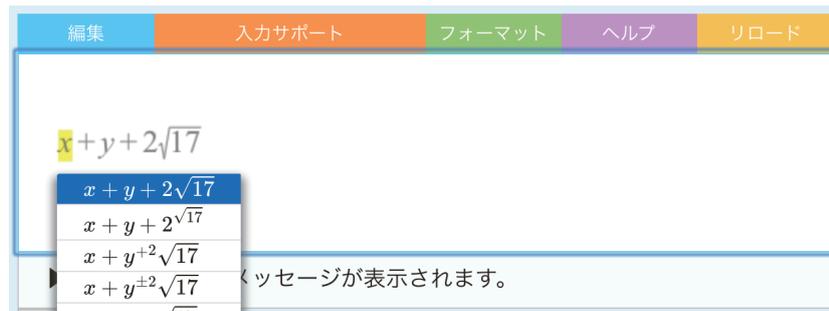


図 3 数式予測アルゴリズムを実装した MathTOUCH のスクリーンショット

表 2 練習で使用した数式

数式番号	数式	数式番号	数式
B1	$1 + 2$	P1	$x^2 - 8x + 16$
B2	$2 - 1$	P2	$x + y + 2\sqrt{30}$
B3	$2x$	P3	$\frac{4x-1}{3\sqrt{2x}}$
B4	$\frac{1}{10}x$	P4	$\frac{1}{27}(\pi - 2)$
B5	$x^2$	P5	$\frac{3 \sin x}{\cos^4 x}$
B6	$\sqrt{2}$	P6	$\sqrt{1 - \sin^2(\pi x)}$
B7	$\sqrt{2}x$		
B8	$2\pi$		
B9	$\sin x$		
B10	$\cos x$		
B11	$\sin^2 x$		
B12	$\cos^2 x$		

表 3 本番タスクで使用した数式

数式番号	数式
T1	$x^2 - 9x + 18$
T2	$x + y + 2\sqrt{17}$
T3	$\frac{3x-1}{4\sqrt{2x}}$
T4	$\frac{1}{27}(\pi - 1)$
T5	$\frac{2 \sin x}{\cos^4 x}$
T6	$\sqrt{1 - \cos^2(\pi x)}$

の B1~B12) の練習とマニュアルの字句や口頭説明の良し悪しによる影響を最小限に押さえるために、本番のタスクと同種の数式 (表 2 の P1~P6) を入力する練習を行い、理解レベルの統一を行う。その後で、本番タスクとして表 3 の T1~T6 に示す 6 個の数式を入力してもらった。タスクテスト終了後に、操作性に関する主観満足度アンケート調査を行い、実験は終了である。

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 2 年生、4 年生の 10 名で、プログラミングには比較的慣れている学生である。ただし、いずれの被験者も MathTOUCH の使用は初めてである。

#### 4.2 測定方法

時間の測定は、入力を始めてから STACK の確認エリア (図 4) に「正しく」数式が表示されるまでとした。すなわち、数式自動採点システムとしては、 $-\frac{1}{4}$  と  $\frac{-1}{4}$  は数学的に等価であり、どちらで入力した場合も正解となるが、本

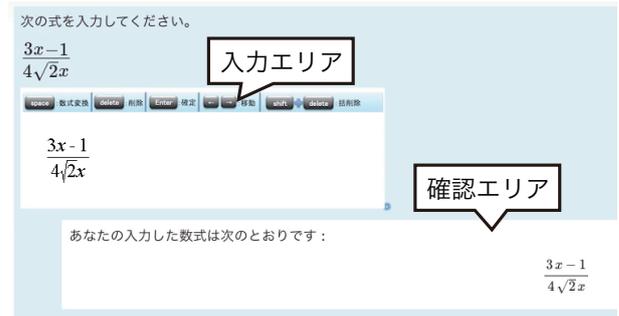


図 4 従来型 MathTOUCH を実装した STACK のスクリーンショット

実験の目的は数式入力のパフォーマンスを評価することであるため、修正行動も含め、2次元表記の数式構造が正確に一致するまでを入力時間とした。測定には、Quick Time Player (10.3) のデスクトップキャプチャ機能を使って録画し、計測した。

タスクで使用する数式は、先行研究 [9] と同様に、数式の偏りを避けるために数学記号およびギリシャ文字は「数学 I」までの範囲で扱われるものとし、キーボードに存在しない演算子または特殊記号が 1 つ含まれたもの、2 つ含まれたもの、3 つ含まれたものを 2 つずつ選んだ。分析で使用するタスク達成時間は、これら 6 つの数式の入力時間の平均値とした。

タスクテスト終了後に、ユーザビリティに関する主観満足度の評価として、5 段階の評定尺度法によるアンケート調査を行った。質問項目は先行研究 [9] と同じでユーザビリティ特性に関する「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 (表 4)。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) から当てはまるものを選んでもらった。また、操作性について自由記述で感想を求めた。

実験環境は、デスクトップコンピュータ (iMac 27 インチ, MacOS X 10.9.2) を使った。キーボードは、Apple Keyboard (テンキー付き-JIS) を使用した。テンキーの使用は被験者の任意とし、マウスは Apple Mouse を使って

表 4 MathTOUCH 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

(質問の意図) アンケートの質問文	5 段階評定尺度平均 (SD)	
	従来型	全体予測型
(A. 学習しやすさ) 数式の入力手順をマスターするのは簡単だ	4.2(0.4)	3.8(1.0)
(B. 効率性) 数式を早く、スムーズに入力できる	3.8(1.2)	3.6(1.0)
(C. 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	3.2(0.8)	3.6(1.0)
(D. 記憶しやすさ) 練習後、入力方法を覚えていた	4.6(0.5)	3.8(1.0)
(E. 再利用意向) 今後もし数式を入力する機会があった時はまたこのエディターを使ってみたいですか?	4.6(0.5)	4.0(1.1)

5段階評定尺度 (1 点: 全然そう思わない, 2 点: あまりそう思わない, 3 点: どちらともいえない, 4 点: ややそう思う, 5 点: とてもそう思う),  $N=10$

表 5 MathTOUCH による数式別のキー打鍵数および平均入力時間

タスク数式番号	T1	T2	T3	T4	T5	T6	全体平均	
全体予測型の正解候補順位	1	1	10	3	3	3		
打鍵数 (回)	従来型	15	17	22	19	19	23	19.2(2.7)
	全体予測型	10	12	22	12	14	17	14.5(4.0)
平均入力時間 (秒)	従来型	23.2(12.3)	17.2(4.7)	52.8(31.3)	29.3(9.6)	44.6(35.2)	31.9(13.1)	33.2(8.7)
	全体予測型	11.8(2.3)	13.0(1.9)	33.8(5.9)	27.7(10.4)	19.1(2.1)	31.8(17.3)	22.9(4.7)

括弧内の数値は SD,  $N=10$

もらった。各ソフトウェアのバージョンは STACK は 3.3, Moodle は 2.5.9, Chrome は 56.0.2924.87 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。構成要素逐次確定方式による従来の MathTOUCH を実装した STACK の画面を図 4 に示す。

### 4.3 結果と考察

実験の結果、いずれのインタフェースも全ての被験者がタスクを達成することができた。

動画分析による計測によって得られた各数式の平均入力時間と各数式の完成に要する打鍵数を表 5 に示す。この打鍵数は、変換・確定・候補選択に関するキー操作も含み、ミスなく入力した場合にかかる総打鍵数を表している。また、表 5 の 2 行目に示したのは、10 ベスト候補に含まれていた正解式の順位である。平均入力時間については、この順位に関わらず、全体予測型を利用した方がどの数式も短い時間で入力できていることが分かる。特に注目すべきは、数式 T3 と T5 の事例で、従来型での入力時間が 40~50 秒と大変手間取っている。この理由は、これらが分数構造をもち、先行研究 [9] で指摘されていたオペランド範囲の指示操作が面倒なためである。しかし、全体予測型では入力時間が大きく改善している。打鍵数についても従来型に比べて、ほとんど少ない打鍵数で数式入力完了している。

総じて、タスク全体の平均入力時間は、全体予測型が 22.9 秒 ( $SD:4.7$ )、従来型が 33.2 秒 ( $SD:8.7$ ) であった。正規性の判定に基づき、マン・ホイットニーの  $U$  検定を行った結果、有意傾向が見られた ( $p=.07$ )。全体予測型を利用した方が 10.3 秒ほど短く、約 1.4 倍速い入力時間でタスクを達成できていることが分かった。

次に、主観満足度のアンケート結果を表 4 に示す。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、いずれのインタフェースも全ての項目において、5 段階 (評

定値: 1~5) の主観満足度の評定平均値である 3.6[26] より高い評価であった。正規性の判定に基づき、マン・ホイットニーの  $U$  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。これにより、従来型の MathTOUCH と変わらぬ高い満足度を示すことが分かる。

最後に、正解式の候補順位が低い T3 と入力時間に差があまり見られなかった T6 について動画分析を行ったところ、候補リストには同じような数式が並び、その中から正解式を見つけ出す必要があるため、正解式を認識するのに時間を要していることが分かった。アンケートの自由記述項目に「選択候補のメニューが小さく見づらい」といった記述が見られ、候補の数式を判別しやすいようフォントサイズや余白等、候補の表示方法を見直す必要性が示唆された。

## 5. まとめと今後の課題

本研究では、従来の構成要素逐次確定方式のオペランド範囲の指示操作の問題点を改善するために、機械学習による数式予測アルゴリズムを実装した全体候補選択方式による MathTOUCH を試作した。改善効果を評価するために、従来型 MathTOUCH と全体予測型 MathTOUCH を使い、数式入力タスクのパフォーマンスを比較した結果、従来型より約 1.4 倍速く数式が入力できる傾向にあることを示した。

今後の課題は、変換候補の表示方法の改善と予測アルゴリズムの精度を向上させることである。

**謝辞** 本研究は JSPS KAKENHI Grant Number JP26330413 および 16K16178 の助成を受けたものです。

### 参考文献

- [1] Clarck, R. C. and Mayer, R. E.: *e-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*, Jossey-

- Bass/Pfeiffer (2003).
- [2] CIEC 研究会：「第 100 回研究会報告書」, CIEC 第 100 回研究会報告書, pp.1-6, 2014.
- [3] 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.
- [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] Sangwin, C. J., Ramsden, P. : Linear syntax for communicating elementary mathematics, Journal Article Journal of Symbolic Computation, Vol.42, No.9, pp.920-934 (2007).
- [6] 樋口三郎：「数式入力による数学評価システム Maple T.A. を利用した理工系学部での基礎教育」, 数理解析研究所講究録, No.1978, pp.72-78 (2015).
- [7] MathTOUCH: MathTOUCH(online), available from <http://math.mukogawa-u.ac.jp/> (2017.02.06).
- [8] 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.
- [9] 白井詩沙香, 仲村裕子, 福井哲夫：「数式自動採点システムにおける数式入力インタフェースの提案と評価」, 情報処理学会論文誌「教育とコンピュータ」, Vol.1, No.3, pp.11-21, 2015.
- [10] Fukui, T., Shirai, S.: *Predictive Algorithm from Linear String to Mathematical Formulae for Math Input Method* Proceedings of the 21st Conference on Applications of Computer Algebra, pp.17-22 (2015).
- [11] Ja STACK.org : Ja STACK.org(online), 入手先 <http://ja-stack.org/> (2017.02.06).
- [12] Maplesoft: Maple T.A.(online), 入手先 <http://maplesoft.com/products/mapleta/> (2017.02.06).
- [13] 大阪府立大学 高等教育推進機構: MATH ON WEB(online), 入手先 <http://www.las.osakafu-u.ac.jp/lecture/math/MathOnWeb/> (2017.02.06).
- [14] Maxima: Maxima(online), available from <http://maxima.sourceforge.net/> (2017.02.06).
- [15] Maplesoft: Maple(online), available from <http://www.maplesoft.com/products/Maple/> (2017.02.06).
- [16] Wolfram Research: webMathematica(online), available from <http://www.wolfram.com/products/webmathematica/> (2017.02.06).
- [17] The Mathematical Association of America: WeB-WorK(online), available from <http://webwork.maa.org/>, (2016.02.06).
- [18] Newcastle University: Numbas(online), available from <https://www.numbas.org.uk/>, (2016.02.06).
- [19] Pollanen, M., Wisniewski, T., Yu, X.: *XPRESS: A Novice Interface for the Real-Time Communication of Mathematical Expressions*, In Proceedings of MathUI200 (2007).
- [20] Smithies, S., Novins, K., Arvo, J.: *Equation Entry and Editing via Handwriting and Gesture Recognition*, Behaviour and Information Technology, Vol.20, No.1, pp.53-67 (2001).
- [21] Shirai, S., Fukui, T.: *MathTOUCH: Mathematical Input Interface for E-Assessment Systems*, MSOR Connections, Vol.15, No.2, pp.70-75 (2016).
- [22] 白井詩沙香, 福井哲夫：多種ある数式自動採点システムの統一的解答入力の提案, PC カンファレンス論文集 2016, pp.11-12 (2016).
- [23] DragMath : DragMath(online), available from [https://docs.moodle.org/29/en/DragMath\\_equation\\_editor](https://docs.moodle.org/29/en/DragMath_equation_editor) (2017.02.06).
- [24] 福井哲夫：「数式曖昧表記変換型数式入力の機械学習による数式予測と精度」, ARG Web インテリジェンスとインタラクション研究会 第 6 回研究会予稿集, pp.67-72(2015).
- [25] 侯野博, 河野俊丈, 他 27 名: 高等学校数学用文部科学省検定済教科書 数学 I・A・II・B・I・II, 東京書籍 (2014).
- [26] 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, 東京電気大学出版局 (2002).