

論文

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

白井 詩沙香^{1,a)} 仲村 裕子² 福井 哲夫^{1,b)}

受付日 2014年3月31日, 再受付日 2015年1月22日,
採録日 2015年3月22日

概要: 数学系オンラインテストなどで, 数式による解答を評価できる数式自動採点システムは, 数式入力方法の改善が課題となっている. 一方, 2011年に福井は仮名漢字変換に似た数式入力方式を発表し, 専門家仕様の数式入力インタフェース MathTOUCH を開発した. 本研究では, MathTOUCH の数式変換エンジンを使い, 数式自動採点システム向けに最適化した数式入力インタフェースを新たに開発し, 数式入力方法改善のために提案する. 提案インタフェースの評価のために, 数式自動採点システム STACK に実装し, 高等学校生および大学生に数式入力パフォーマンス実験を行った. その結果, 数式の平均入力時間は従来方式と比べて 1.2~1.6 倍速く, インタフェースの操作性に関する主観満足度においても有意に高い評価を得た.

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

An Interactive Math Input Method for Computer Aided Assessment Systems in Mathematics

SHIZUKA SHIRAI^{1,a)} YUKO NAKAMURA² TETSUO FUKUI^{1,b)}

Received: March 31, 2014, Revised: January 22, 2015,
Accepted: March 22, 2015

Abstract: Math input interfaces of most current computer aided assessment (CAA) systems that are able to assess answers using mathematical expressions, are cumbersome for novice learners. In 2011 Professor Tetsuo Fukui of Mukogawa Women's University proposed a new mathematical input method similar to the ones used for inputting Japanese characters in many operating systems. He developed an interface that implemented this method for professional users, called MathTOUCH. In this paper, we have presented a new math input interface that optimizes the MathTOUCH's math conversion engine for mathematical CAA systems. This proposed math input interface was implemented in STACK. We used an experimental performance test to investigate whether students can input mathematical expressions using our proposed interface more smoothly than with the standard interfaces found in current CAA systems. Our results show that this new input interface allows task times of approximately 1.2 - 1.6 times faster than with the standard interfaces. Moreover, our system was shown to have a high level of user satisfaction in regards to math input usability.

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

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

² 武庫川女子大学附属中学校・高等学校
Mukogawa Women's University Junior & Senior High
School, Nishinomiya, Hyogo 663-8143, Japan

a) shirai@mukogawa-u.ac.jp

b) fukui@mukogawa-u.ac.jp

1. はじめに

インターネット利用人口の拡大とともに 2000 年頃から普及するようになった e ラーニング [1] の重要な機能として, 学習者の理解度を確認するためのオンラインテストがある. 従来のオンラインテストの解答形式は, 空所補充,

正誤選択, 単一選択, 多肢選択などで, 数式による解答を扱うことは困難であり, 理数系科目の演習であっても従来の解答形式の範囲内で工夫して問題を作成しなければならなかった [2], [3].

これを解決したのが, 数学的な等価性に基づいて数式の正誤判定を実現した数式自動採点システムである [4]. 数式自動採点システムの登場により, 数式による直接解答が実現できるようになった. 現在, 数式自動採点システムは, 主に大学を中心にリメディアル教育や大学数学の学習支援として使用されている [5], [6], [7]. しかし, 現在の数式自動採点システムで利用されている解答の数式入力方法は, 学習者の負担となっていることが報告されている [8], [9], [10]. したがって, 数式自動採点システムの数式入力方法の改善は重要な課題である.

一方, 2011年に福井が提案した数式曖昧表記変換方式 [11], [12], [13] は, 初期入力文字列がふだん数式を読むような曖昧な表記でよく, WYSIWYG で数式を確認しながらキーボードのみで入力できるという特徴を持つ. 先行研究では, 多項式の入力において本方式の実装インタフェースである MathTOUCH は Microsoft® Word®*1 [14] の数式エディタより, 1.7 倍速く入力できることが報告されている [12]. それゆえ, 数式自動採点システムに実装できれば, 学習者への負担改善が期待できる. しかし, MathTOUCH は教師の教材作成や研究者の論文作成支援を目的に開発されたため, このままでは数式自動採点システムで扱うことができない.

本研究の目的は, 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方法を改善するために, MathTOUCH の数式変換エンジンのみを利用して数式自動採点システム向けに最適化した新たな数式入力インタフェースを開発し, 提案することである. 本研究では, 数式自動採点システム STACK [15], [16], [17] を検証環境とし, 提案インタフェースが数式自動採点システムの数式入力改善に貢献できるかを明らかにするために, 数式入力パフォーマンス実験によってその有効性を検証する.

2. 数式自動採点システムと数式入力方式

2.1 数式自動採点システム

数式自動採点システムとは, 学習支援システムのオンラインテストにおいて, 数式による解答の自動採点ができるシステムである. 多くの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) を活用し, 入力された数式が, あらかじめ問題作成者が設定した模範解答と数学的に等価であるかを計算し, 正誤判定を行っている*2.

商用の CAS を使用した数式自動採点システムとして, Maple® [19] を使用した数式自動採点システムである Maple T.A.®*3 [20] や webMathematica®*4 [21] を使用した大阪府立大学の数学学習支援サイト MATH ON WEB [22] がある. 一方, STACK は CAS には Maxima [23] が, LMS には Moodle [24] が使われており, すべてオープンソースソフトウェアで開発されている [16]. Moodle との連携は 2007 年に公開された STACK2.0 から行われており, 2013 年に公開された STACK3.0 以降は, Moodle の小テストの問題タイプとして利用できるようになったため, 連携がより容易になった [25].

このように, STACK はオープンソースで, なおかつ国内外の多くの大学で利用されている Moodle で容易に利用できるため, 本研究で提案する数式入力インタフェースの検証環境には STACK を利用する. ただし, 本研究で得られる結果の多くは STACK に限定されるものではない.

2.2 数式自動採点システムにおける従来の数式入力方式

数式は 2 次元構造を持つため, 1 次元的な文字列のように扱うことができず, デジタルデバイスで扱うことは容易ではない. 2014 年現在, 数式自動採点システムの多くは, 主にパソコン環境で使用されているため, 一般的にパソコンで数式を入力する際に利用されているテキストベース入力方式または構造ベース入力方式で学習者に解答の数式を入力させている.

2.2.1 テキストベース入力方式

テキストベース入力方式は, テキストフィールドにマークアップ言語や CAS コマンド形式に従って, 数式を入力する方式である. STACK や Maple T.A. など多くの数式自動採点システムがこの方式を使っている.

2 次元表記の数式を 1 次元的な文字列で表現できるため, キーボードのみで入力できるという特徴を持つ. しかし, 入力すべき文字列は機械が完全にフォーマットできるような文法構造になっているため, ユーザにとって可読しにくく, 初学者には直感的に理解することが難しい [26]. たとえば, STACK では数式 $e^{\pi x}$ を Maxima 形式の “e^(%pi*x)” と入力するので, 数式に表示されないキャレット記号やアスタリスク記号や括弧記号が必要となる.

数学学習中の入力トラブルとして, アスタリスク記号や括弧忘れなどの CAS コマンドの文法ミスによる事例が報告されている [10], [27]. 入力補助のために明確な文法のヒントを表示していても, 数学学習に集中していると学習者はヒントを見ないためエラーを起こすことや入力方法を習得した後もアスタリスク記号や括弧などの入力忘れが発生すると報告されている [27].

誤入力時の対策を行っている数式自動採点システムもあ

*1 Microsoft, Word は Microsoft Corporation の登録商標である.

*2 WeBWorK [18] など, CAS を使用していない数式自動採点システムもある.

*3 Maple, Maple T.A. は Waterloo Maple Inc. の商標である.

*4 webMathematica は Wolfram Research, Inc. の商標である.

表 1 MathTOUCH Jr. による Maxima 形式出力の例
Table 1 Example Maxima format outputs using MathTOUCH Jr.

数式タイプ	数式例	数式曖昧表記	Maxima 形式出力
多項式	$3x^2 + 1$	3x2+1	3*x^2+1
根号	$\sqrt{2}$	root2	sqrt(2)
三角関数	$\sin^2 x$	sin2x	sin(x)^2
対数関数	$\log_{10} x$	log10x	log(x)/log(10)
指数関数	$e^{\pi x}$	epx	e^(%pi*x)
総和	$\sum_{k=1}^n k^2$	sumk=1nk2	sum(k^(2),k,1,n)
微分	f' または $\frac{df}{dx}$	f' または df/dx	diff(f,x)
積分	$\int_a^b f dx$	intabfdx	integrate(f,x,a,b)

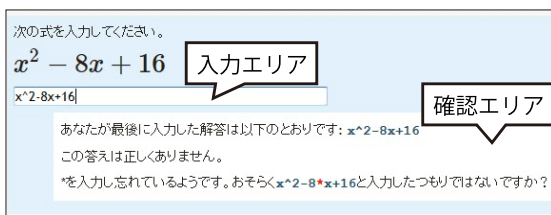


図 1 STACK 標準の Maxima コマンド形式の数式入力インターフェース

Fig. 1 A standard Math input interface using STACK's Maxima format.

り、たとえば STACK では、文法を間違えると入力エリア直下に表示される Validation エリア（以下、確認エリア）に誤入力の修正を支援するコメントが表示される。図 1 の確認エリアには、 $8x$ の間にアスタリスク記号が抜けていることを指摘するメッセージが表示されている。なお、正しく入力できれば確認エリアには 2 次元表記の数式が表示される。

2.2.2 構造ベース入力方式

構造ベース入力方式は、グラフィカルユーザインターフェース（以下、GUI）を使い、はじめに分数やルートなどの数式構造をツールバー上のアイコンテンプレートから選択し、数式を作図するように入力する方式である。WYSIWYG のため、入力した数式がイメージしやすい。

Maple T.A. がテキストベース入力方式が難しい学習者向けに、構造ベース入力方式実装インターフェースを用意している。STACK も ver.2.X まではテキストベース入力方式に加え、DragMath [28] という構造ベース入力方式実装インターフェースを使っていた。DragMath では、数式構造のシンボルアイコンをクリックもしくはドラッグ&ドロップするとワークスペースにテンプレートが挿入されるため、テンプレートの空欄をクリックし、数値やさらにテンプレートを挿入することで数式入力を行う。ver3.0 以降は STACK で使用していないが、現在は Moodle の数式入力インターフェースとして利用されている。

しかし構造ベース入力方式は、入力する際に頭の中で入力したい数式の構造を解析し、構造ベース入力方式に合わせ

た手順で入力する必要があるため、初学者には難しく [29]、修正による構造の変更が難しいとの指摘がある [30]。

2.3 数式入力の先行研究

2.3.1 手書き入力方式

上記以外の代表的な数式入力方法として、ポインティングデバイスなどを使用し、紙面上に数式を書くように数式を入力できる手書き入力方式がある。しかし、手書き認識技術は認識精度の問題が指摘されている [31]。

2007 年に糟谷らは、数式構造の認識の困難さを指摘し、記号文字の認識のみで数式入力を行える手書き数式入力システム MathBox を提案している [32]。MathBox ではインタラクティブに表示される数式構造の枠表示により入力補助を行う。認識率は、ユーザ自身のサンプルを使用し、修正を 3 回まで許せば 92.5% の精度で入力できることを明らかにしている。また、構造ベース入力方式である MathType^{*5}と比較した結果、平均 33.1% 入力時間を削減できたと報告している。

しかし、数式自動採点システムに関しては、2014 年時点では、MathBox をはじめ、手書き入力方式を実装したものは筆者の調べた限り報告されていなかった。また、手書き入力方式はポインティングデバイスが必要であり、主に数式自動採点システムが導入されている大学におけるパソコン端末環境が手書き入力方式に対応しているとは限らないため、本研究では手書き入力方式ではなく、次に示す数式入力方式を検討した。

2.3.2 数式曖昧表記変換方式

1 章で述べた、もう 1 つの数式入力方式、数式曖昧表記変換方式は、初期入力文字列が表記されない区切り記号や演算子は入力しない曖昧な文字列でよく、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。また、 \LaTeX で定義されている数学記号やイタリック体、ボールド体、ローマン体、カリグラフィック体、サンسكريット体およびギ

*5 MathType は Design Science, Inc. の商標である。

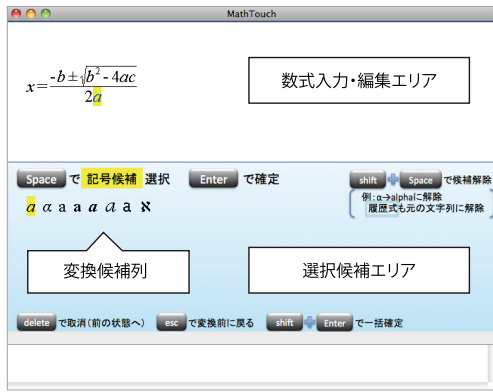


図 2 従来の MathTOUCH

Fig. 2 A sample screenshot of MathTOUCH.

リシャ文字などは、ほぼアルファベット 1 文字に対応しており、変換することで入力可能となっている。表 1 に主な数式例に対応する数式曖昧表記の例を示す。たとえば、数式 $e^{\pi x}$ は文字列 “epx” から変換される。それゆえ、初期入力文字列はテキストベース入力方式に比べ、短い文字数で済む [12]。この考え方は仮名漢字変換に似ているが、数式構造は日本語文と異質のため、数式変換エンジンのアルゴリズムは大きく異なることが報告されている [13]。

この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が MathTOUCH (図 2) である。1 章で述べたように、先行研究 [12] では、限定的ではあるが情報系女子大学生による多項式入力の比較実験により、MathTOUCH は Microsoft® Word® の数式エディタ (構造ベース入力方式) より効率の良さが示された。

3. 提案する数式入力インタフェース

3.1 従来の MathTOUCH の最適化の方針

しかし、MathTOUCH は教師の教材作成や研究者の論文作成支援を目的に開発されたため、図 2 に見られるように初期入力文字列から数式候補に変換するための表示方法や数式辞書が専門的で、学習者の混乱を招く可能性がある。また、数式自動採点システムに組み込むためには、CAS コマンドの出力機能が必要であり、インタフェースを見直す必要もある。

さらに、一般的に数式自動採点システムでは、図 1 のように出題エリア、解答入力エリア、結果表示エリア (STACK では確認エリア) を提示する。そのため、従来の MathTOUCH の画面サイズ (W710px×H550px) では、解答入力エリアに実装するには大きく、画面サイズの省スペース化が求められる。

以上の検討から、本研究では数式曖昧表記変換方式を実装した MathTOUCH の変換エンジンだけを活用し、数式自動採点システム向けに次の 4 つの方針に従って最適化を行った新たな数式入力インタフェースを開発する。

最適化方針 1 CAS コマンドの出力機能を追加実装

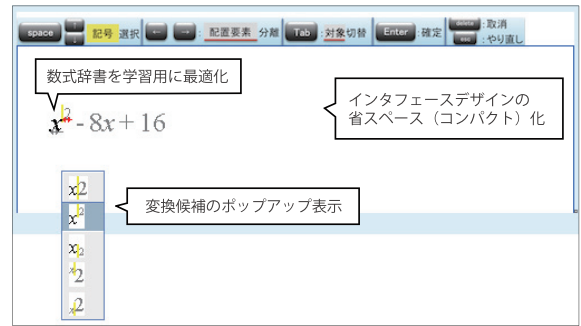


図 3 提案数式入力インタフェース MathTOUCH Jr.

Fig. 3 The interface for MathTOUCH Jr., a proposed alternative to the original MathTOUCH.

表 2 MathTOUCH Jr. で使用できるギリシャ文字と数学記号

Table 2 A sample of greek and mathematical symbols which can be used in MathTOUCH Jr.

アルファベットキー	a	b	c	i	p	t	w	D
数学記号	α	β	γ	∞	π, φ	θ	ω	Δ

最適化方針 2 数式辞書を数学学習用に見直し

最適化方針 3 変換候補の表示方法の改良

最適化方針 4 インタフェースデザインの省スペース化

3.2 数式入力インタフェース MathTOUCH Jr. の開発

前節で述べた方針に従って MathTOUCH の最適化を行い Java 言語を使い開発したのが、図 3 に示す数式自動採点システム向け数式入力インタフェース MathTOUCH Jr. (ジュニア) である。以下でその特徴を報告する。

3.2.1 CAS コマンドの出力機能

MathTOUCH Jr. は、検証環境である STACK の正誤判定のための数式表現 (Maxima コマンド形式) を生成するモジュールを実装している。ただし、Maxima 形式で表現できない数式はすべて Null を返す。実際、表 1 に示した各数式タイプを上記モジュールで変換させたところ、表 1 の 4 列目に示した Maxima 形式文字列を出力できることを確認した。

3.2.2 MathTOUCH Jr. 用数式辞書データ

最適化方針 2 のために高校の教科書「数学 (I~III, A~C)」を調査し、数式辞書には小文字のアルファベットに対してはイタリック体と一部のギリシャ文字・数学記号 (表 2) のみに対応するよう絞り込んだ。特に、現時点における数式自動採点システムに字体の違いを判別する機能はない。

3.2.3 インタフェースデザインの改良

最適化方針 3 として、変換候補列の表示を図 3 に示すような WYSIWYG でポップアップ表示となるように変更した。変換候補を変換対象の文字列の直下に表示することで、視線を落とす必要もなくなった。変換候補の選択操作についても、従来は Space キーのみで行っていたが、上下

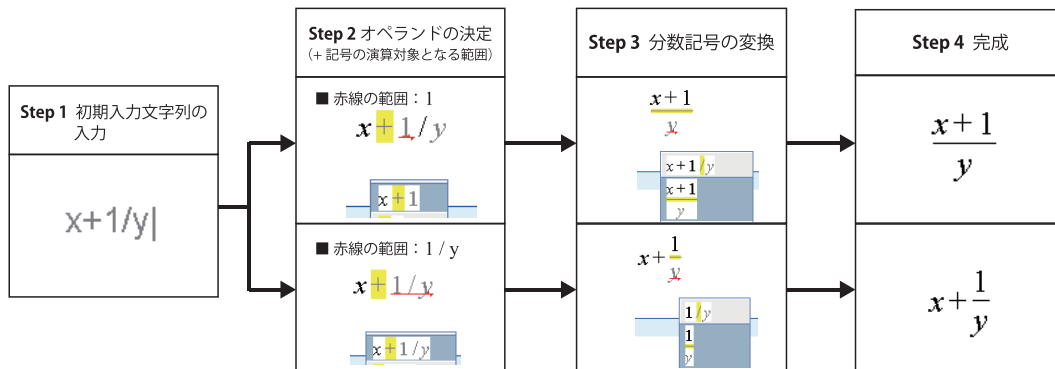


図 4 MathTOUCH Jr. による数式入力手順
Fig. 4 MathTOUCH Jr. input procedure.

矢印キーでも選択できるようにした。

なお、この改良によって、選択候補エリア (図 2 参照) が不要になり、省スペース化を実現した。

3.3 MathTOUCH Jr. による数式入力手順

MathTOUCH Jr. による具体的な入力手順を $\frac{x+1}{y}$ の例に説明する。まず、キーボードで“x+1/y”と入力する (図 4 の Step 1)。次に Space キーを押し、変換を開始すると、図 4 の Step 2 のように、確定済みの要素は黒色、未変換の数式は灰色で表示され、変換対象は黄色の背景色で表示される。変換は“+”演算子から始まり、対象演算子のオペランド範囲 (赤線) を調整するとポップアップ表示された候補列の表示もリアルタイムに変更が反映される。オペランド範囲を指定できたら、Enter キーで確定する。

特に、MathTOUCH Jr. は数式曖昧表記変換方式を採用しているため、図 4 のように 1 つの数式曖昧文字列から複数の数式が構築できる。たとえば、“+”演算子の右側オペランド (赤線の範囲) を $\frac{1}{y}$ とすれば $\frac{x+1}{y}$, $\frac{1}{y}$ まで広げれば $x + \frac{1}{y}$ となって結果が異なる。MathTOUCH Jr. にまだ慣れていない利用者にとっては、一方の結果を求めてもう一方が出力されると、戸惑う場合がある。

3.4 提案数式入力インタフェースの STACK への実装

STACK では新しい数式入力インタフェースを利用するためのプラグイン機能がある。このプラグイン機能を利用し、STACK 標準の入力エリアへ解答を入力したかのように、MathTOUCH Jr. からの Maxima コマンドを入力エリアに挿入し、適切にフォーカスが移動する連携プログラムを作成した。STACK へ実装した MathTOUCH Jr. の画面を図 5 に示す。

技術的には、プラグイン用の PHP ファイルに HTML 内 Document オブジェクトに対して、MathTOUCH Jr. からの結果を渡すプログラムを作成し、特に、入力エリアへ完成した数式を出力した後、Enter を押すことにより、フォーカスが自動で MathTOUCH Jr. に戻るよう実装した。こ

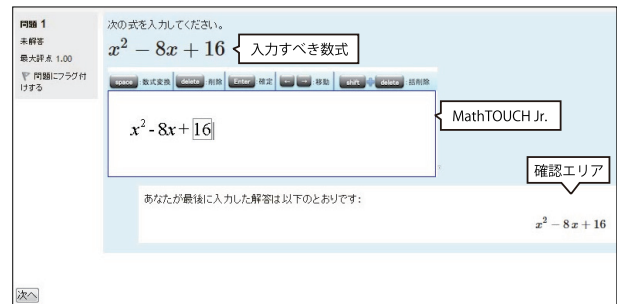


図 5 STACK に実装した MathTOUCH Jr.
Fig. 5 A screenshot of MathTOUCH Jr.'s STACK implementation.

れにより、キーボードとマウスを持ち替えるの必要がなくなり、入力エリアに直接 Maxima コマンドを入力する場合と同様の操作で数式入力が行える。

問題作成者が MathTOUCH Jr. を使った小テストを作成したい場合は、本研究で提案する MathTOUCH Jr. のプラグインファイルを所定のディレクトリ*6に保存するだけでよい。そうすれば、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プルダウンメニューから MathTOUCH Jr. が指定できるようになる。

以上のように、MathTOUCH Jr. のプラグイン化によって、STACK の数式入力インタフェースの選択肢の幅を広げることができる。

4. 評価

4.1 評価の概要

提案した MathTOUCH Jr. の有効性を示すために、数式自動採点システムにおいて学習者を想定した被験者に実際に数式を入力してもらい、その際の使いやすさ (ユーザビリティ) に関する定量的データを測定し評価を行う。ここで、ユーザビリティに関する定量的データとは、樽本 [33]

*6 Moodle 内の STACK の input ディレクトリ内にプラグイン用のディレクトリを作成し、当該ディレクトリ内に MathTOUCH Jr. プラグイン (STACK の必須関数を含む PHP ファイル) を保存すれば、STACK の数式入力タイプとして認識され、新しい数式入力インタフェースとして利用できる。

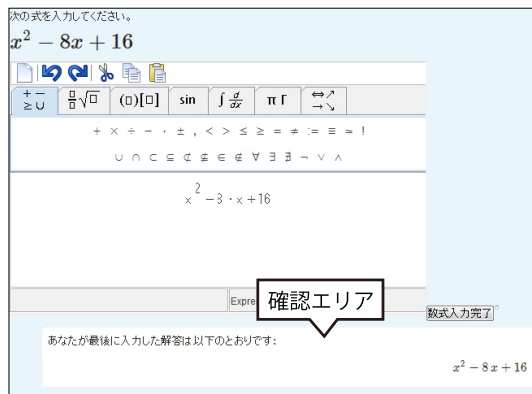


図 6 STACK に実装した DragMath

Fig. 6 A screenshot of DragMath's STACK implementation.

表 3 被験者数一覧 (人)

Table 3 A breakdown of task test participants.

	所属	性別	UI1	UI2	UI3
高等学校生	文系	男性	3	2	2
		女性	2	3	3
	理系	男性	5	7	5
		女性	7	9	6
大学生	文系	男性	2	4	3
		女性	6	3	4
	理系	男性	5	5	6
		女性	5	6	5
総数 (N_k)			35	39	34

によるユーザビリティ 3 要素 (効果・効率・満足度) のことで、一定のタスクに対する達成率を効果、達成時間を効率の指標とし、使用後のアンケートにより満足度を調べる。

また、比較のために (UI1) MathTOUCH Jr. のほかに、2.2 節で述べた (UI2) STACK 標準の Maxima コマンド形式による数式入力 (以下、Maxima 形式入力) および (UI3) STACK で使用実績のある DragMath の計 3 つのインタフェースに対して同一のタスクテストを行う。ただし、テストの経験が他のテストに影響を与えるかもしれないので、被験者は必ず 1 つのインタフェース k (UI1, UI2, UI3 のいずれか) のみでタスクテストを実施してもらう。なお、確認エリアへの数式表示は (UI2) Maxima 形式入力はリアルタイム、(UI1) MathTOUCH Jr. は数式確定後に Enter キーを押すことで表示できる。(UI3) DragMath は図 6 のように数式入力後に数式入力完了ボタンを押すことで表示できるようにした。

4.1.1 実験方針

本評価実験は、特に、数式自動採点システムにおけるユーザビリティ評価の知見となるよう次の 4 点に配慮した。

- (1) 被験者には学習者を想定して、年齢が高等学校生から大学生までの男女で文系・理系がバランスよく含まれるようにする (表 3)。
- (2) タスクとして入力してもらう数式は被験者の数学知識

表 4 タスクで使用した数式

Table 4 The six mathematical expressions used in our task test.

数式番号	数式	キーボードに存在しない特殊記号の数
E1	$x^2 - 9x + 18$	1 つ
E2	$x + y + 2\sqrt{17}$	
E3	$\frac{3x-1}{4\sqrt{2x}}$	2 つ
E4	$\frac{1}{27}(\pi - 1)$	
E5	$\frac{2\sin x}{\cos^4 x}$	3 つ
E6	$\sqrt{1 - \sin^3(\pi x)}$	

の差が影響しないように配慮する。

- (3) STACK で実績のある数式入力インタフェース (UI2), (UI3) を比較対象とする。
- (4) 実施環境 (マシン・提示画面・進行手順) を統一する意味で数式入力 UI だけが異なる STACK で実施する。

実験方針 (2) で選定したタスクのための数式は、表 4 に示す 6 種類の数式 E1~E6 である。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に限定し、総文字数が 8~11 文字とし、キーボードに存在しない演算または特殊記号が 1 つ含まれたもの、2 つ含まれたもの、3 つ含まれたものをそれぞれ 2 つずつ選び、偏りを避けるため、それら 6 つの入力時間の平均値を評価に使用する。

4.1.2 タスクの定義

本実験において被験者に実施してもらったタスク J_k とは、インタフェース k を使い表 4 に示す 6 つの数式 E1~E6 すべてを「正しく」入力することである。ただし、修正にかかる時間も効率の一部と考え、操作マニュアルを見てもよく、ミスがあれば正しい形になるまで何度でも修正してよいものとした。ここで、「正しく」とは、2 次元表記した数式構造まで一致していることを指している。たとえば、数式 E4 は代数的には $\frac{\pi-1}{27}$ でも等価であるが、今回の実験では STACK ではなくインタフェース部分の操作性のみを評価するため、不可とし、E4 の表記どおりに入力してもらった。また、予備実験から、どの UI であっても数式 E_j ($j=1\sim 6$) の入力時間がほとんど 2 分以内で完了していたことから、1 式あたりの制限時間を 2 分とした。したがって、インタフェース k を使って被験者 i_k ($=1\sim N_k$) が行ったタスク J_k において数式 E1~E6 のいずれかの入力中、制限時間を過ぎた場合は未達成と判断する。ここで、 N_k はタスク J_k を受験した有効被験者総数で、気づかずに E1~E6 とは違う数式を入力してしまった場合は無効とする。

そこで、被験者 i_k のタスク達成の判断記録 (タスク達成を 1, 未達成を 0) を $\delta(i_k)$ で表す。

$$\delta(i_k) = \begin{cases} 0 & \text{未達成} \\ 1 & \text{達成} \end{cases} \quad (1)$$

これにより、インタフェース k のタスク達成者数 M_k は

表 5 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 5 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	5段階評定尺度平均 (SD)		
	UI1	UI2	UI3
(A. 理解しやすさ) 数式の入力手順をマスターするのは簡単だ	3.94(0.97)	3.64(1.11)	3.73(1.23)
(B. 効率性) 数式を早く, スムーズに入力できる	3.83(0.89)	2.97(1.16)	3.42(1.09)
(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	3.86(1.12)	3.56(1.12)	3.33(1.22)
(D. 記憶しやすさ) 練習後, 入力方法を覚えていた	4.46(0.70)	3.82(0.76)	3.91(1.01)
(E. 再利用意向) 今後もし数式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?	4.43(0.78)	3.79(0.95)	3.91(1.10)

$N_{UI1}=35, N_{UI2}=39, N_{UI3}=33$

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)

式 (2) で表される.

$$M_k = \sum_{i_k=1}^{N_k} \delta(i_k) \quad (2)$$

したがって, タスク達成率 P_k は式 (3) で表すことができる.

$$P_k = \frac{M_k}{N_k} \quad (3)$$

次に, タスク J_k のうち, 数式 E_j を正しく入力することをタスク j と呼ぶ. 被験者 i_k がタスク j を達成するのにかかった時間を $T(i_k, j)$ とすると, インタフェース k の平均タスク達成時間 T_k は式 (4) で定義される.

$$T_k = \frac{1}{M_k} \sum_{i_k=1}^{N_k} \delta(i_k) \left(\frac{\sum_{j=1}^6 T(i_k, j)}{6} \right) \quad (4)$$

すなわち, タスク J_k を達成できた者のうち, 1つの数式を入力するのにかかる平均時間である.

4.1.3 満足度の測定

インタフェース k を使用した満足度を測るために, タスクテスト終了後に5段階の評定尺度法に基づくアンケートを実施した (5が最良). 質問項目には, ユーザビリティ特性 [34] 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」に関する質問に回答してもらった. それら5項目に対する具体的な質問文を表5の1列目に示す. また, 被験者にインタフェースについて自由にコメントを記述してもらった. ただし, 我々の提案するインタフェースがどれであるかは知らせないように配慮した.

4.2 実験の実施

4.2.1 実施手順

タスク J_k を実施するためには, インタフェース k による数式入力操作の方法を理解する必要があり, 被験者へタスクの前に操作説明をし, 練習を行ってもらった. しかし, 特定の被験者だけインタフェース k の使用経験があったり, 実験者の説明や事前練習が k によって偏っている公平性を失ってしまう. その点の配慮については次の 4.2.2 項で

表 6 練習で使用した数式

Table 6 Practice mathematical expressions used in this study.

数式番号	基本数式	数式番号	実践数式
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	2π		
B9	$\sin x$		
B10	$\cos x$		
B11	$\sin^2 x$		
B12	$\cos^2 x$		

述べる. したがって, 実施手順は, 次の 1)~6) のように実施した.

- 1) 実施環境の準備と実験趣旨説明
- 2) インタフェース k の操作説明
- 3) 基本要素練習問題 (表 6 の B1~B12) による操作練習
- 4) 実践問題 (表 6 の P1~P6) による未理解部分の解消
- 5) タスク J_k の実施
- 6) 満足度に関するアンケートの実施

タスク実施環境として, マシンはラップトップコンピュータ (HP ProBook 4740s, OS: Windows 7 Professional (32 bit)) を使い, 特にキーボード部分のキーピッチが 19mm, キーストロークが 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た. また, 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. ソフトウェアに使用した STACK のバージョンは 3.1 で, Moodle は 2.5.2+, ブラウザは Internet Explorer9 である.

タスク J_k の実施画面は, ($k=UI1$) MathTOUCH Jr. の場合が 3.4 節で示した図 5, ($k=UI2$) が 2.2 節で示した図 1, ($k=UI3$) が 4.1 節で示した図 6 である. 数学的能力に関係なく, すぐに入力を開始できるように, 画面上部に

表 7 タスク達成率 (%) の結果

Table 7 A summary of task-performance rates (in percentage terms).

インタフェース k		UI1. MathTOUCH Jr.		UI2. Maxima		UI3. DragMath		
列項目名		P_{UI1}	M_{UI1}	P_{UI2}	M_{UI2}	P_{UI3}	M_{UI3}	
総合結果		97.1	34	89.7	35	94.1	32	
内訳	専攻	文系	92.3	12	75.0	9	83.3	10
		理系	100.0	22	96.3	26	100.0	22
	プログラミング	なし	93.8	15	80.0	16	86.7	13
	経験	あり	100.0	19	100.0	19	100.0	19

$N_{UI1} = 35, N_{UI2} = 39, N_{UI3} = 34$

入力すべきタスク j (ただし, j は 1~6 の順に) の数式 E_j を 2 次元表記で提示した. 被験者は中央のインタフェース k によって数式を入力すると画面下の確認エリアからの入力結果のフィードバックを確認して次のタスク $j := j + 1$ へ進む.

4.2.2 操作説明・練習の統一性

タスク J_k は必ずインタフェース k の未経験者に対して実施し, タスクの前の操作説明・事前練習においてインタフェース k によらず統一したのは次の 4 点である.

- (i) 実施環境の統一 (前項参照)
- (ii) 説明項目 (a), (b), (c)
 - (a) 操作画面要素・エリアの役割
 - (b) 表示マーク (カーソル, 枠, 色, 強調など) の意味
 - (c) 基本要素 (表 6 の B1~B12) の入力操作手順
- (iii) 基本要素練習問題 (表 6 の B1~B12 を実際に入力)
- (iv) 実践問題 (表 6 の P1~P6)

(ii) の説明項目のみを記載したマニュアルを被験者に渡し, 項目 (a), (b) をマニュアルに沿って口頭説明した後, 残りの項目 (c) を見ながら, 自力で (iii) の基本要素練習問題に取り組んでもらった. ただし, 口頭説明時間はインタフェース k によって同じでない. この理由は, Maxima 形式入力や DragMath の場合はテキスト編集やマウス操作などふだんの経験が活かされている部分は説明が短く済むのに対して, MathTOUCH Jr. の場合は, 図 4 で示したようなオペランド範囲を表す表示マークの意味など説明項目は同じでも, 新しい概念を説明しなければならないからである. 実施手順 3), 4) の練習では, 入力方法が分からない場合は質問を受け付け, 質問がない場合も操作につまずいてから 1 分経過した場合は実験者より入力方法の説明を行った. 実践問題 (iv) の数式 P_j ($j=1\sim 6$) は E_j と同種 (数字のみが異なる) であり, この入力が理解できていないと, タスク達成率 P_k には影響するが, 達成時間 T_k には未達成者は除かれるので無関係である. しかし, T_k は (i)~(iv) の配慮だけでは, 特徴の異なる k のマニュアルの字句や口頭説明の仕方の良し悪しには影響される. その違いをできるだけ小さくするため, 実施手順 4) の実践問題練習は時

間制限を設けず, 自力で正しく入力できるまで未理解部分を解消してもらい, 理解レベルの統一を図った.

4.2.3 タスク達成時間の測定

上記のように, 本実験ではインタフェース k によらず同一の実践問題の入力方法を一度理解した状態で, タスク達成時間の測定を行う. したがって, この実施方法ではタスク達成時間によって, 4.1.3 項で述べたユーザビリティ特性のうち, 「理解しやすさ」および「記憶のしやすさ」を客観的に測ることはできない. しかし, タスク j のタスク達成時間 $T(i_k, j)$ は, 4.1.2 項で定義したように修正を許しているため, 「効率性」および「間違えにくさ・修正しやすさ」の特性を総合的に測っていると見なせる.

タスク達成時間 $T(i_k, j)$ の測定は, 操作中のデスクトップ画面をビデオキャプチャーソフト Bandicam [35] によって録画し, MathTOUCH と Maxima 形式入力方式は, 1 文字目を入力してから, 構造ベース入力方式は初回のテンプレートをクリックしてからを開始と見なし, 入力された数式が確認エリアに表示されるまでを 0.1 秒単位で計測した. また, Bandicam のマウスクリック効果機能を使い, 録画データからマウスクリックのタイミングが分かるようにした.

4.3 実験結果

4.3.1 タスク達成率の結果

実験の結果, タスク達成者数 M_k は, $M_{UI1} = 34, M_{UI2} = 35, M_{UI3} = 32$ であった. したがって, タスク達成率 P_k は, MathTOUCH Jr. が $P_{UI1}=97.1\%$, Maxima 形式入力が $P_{UI2}=89.7\%$, DragMath は $P_{UI3}=94.1\%$, となった. これらの結果を文系・理系別, プログラミング経験の有無別の内訳とともに表 7 に示す. 達成率に関して Fisher の正確確率検定を行った結果, 有意水準 5% で有意差は見られなかった ($p=0.4971$).

4.3.2 タスク達成時間の測定結果

平均タスク達成時間 T_k は, MathTOUCH Jr. が $T_{UI1}=24.3$ 秒, Maxima 形式入力が $T_{UI2}=28.8$ 秒, DragMath が $T_{UI3}=39.0$ 秒であった. これらの結果および文系・理系別, プログラミング経験の有無別の内訳を標準偏

表 8 タスク達成時間の平均値 (秒) の結果

Table 8 A summary of mean task-performance times (in seconds).

インタフェース k		UI1. MathTOUCH Jr.	UI2. Maxima	UI3. DragMath
列項目名		T_{UI1}	SD	T_{UI2} SD T_{UI3} SD
総合結果		24.3	7.4	28.8 8.5 39.0 10.3
内訳	専攻	文系 28.2	8.3	32.8 9.3 45.6 12.4
		理系 22.2	6.1	27.4 7.9 36.1 7.8
	プログラミング	なし 27.7	8.0	32.4 9.3 44.3 11.6
	経験	あり 21.6	5.8	25.7 6.4 35.4 7.8

$M_{UI1} = 34, M_{UI2} = 35, M_{UI3} = 32$

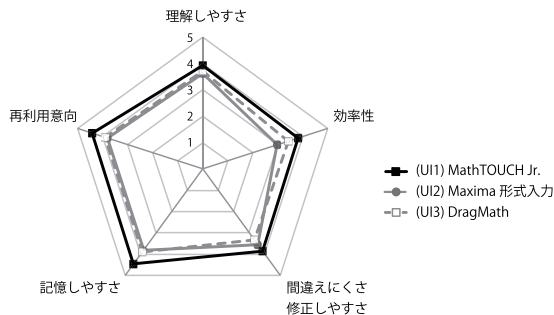


図 7 主観満足度アンケート結果のレーダーチャート
Fig. 7 A radar chart showing the results of our subjective satisfaction questionnaire.

差とともに表 8 に示す。正規性の判定に基づき、クラスカル・ウォリスの H 検定を行った結果、有意な差が見られた ($\chi^2(2) = 39.2, p < .05$)。したがって、マン・ホイットニーの U 検定を用いてライアン法を適用し、多重比較を行った結果 [36]、すべてのインタフェース間で有意差があることが分かった (Maxima \times DragMath: $U = 204, p < .05$, Maxima \times MathTOUCH Jr.: $U = 385, p < .05$, DragMath \times MathTOUCH Jr.: $U = 95, p < .05$)。表 8 の内訳に示した文系・理系別、プログラミング経験の有無別のタスク達成時間から専攻やプログラミング経験にかかわらず、 $T_{UI1} < T_{UI2} < T_{UI3}$ であることが分かる。

4.3.3 主観満足度の測定結果

主観満足度を問う 5 つの質問項目に対する結果を表 5*7 に、各項目の平均値によるレーダーチャートを図 7 に示す。すべての項目で MathTOUCH Jr. が最も満足度が高いことが分かる。各項目についてクラスカル・ウォリスの H 検定を行った結果、効率性 ($\chi^2(2) = 10.6, p < .05$) と記憶しやすさ ($\chi^2(2) = 15.2, p < .05$)、再利用意向 ($\chi^2(2) = 9.7, p < .05$) の 3 項目で有意差が見られた。マン・ホイットニーの U 検定を用いてライアン法を適用し、多重比較を行った結果、効率性は MathTOUCH Jr. が Maxima 形式入力より有意に満足度が高い結果となった ($U = 397, p < .05$)。記憶しやすさについては、MathTOUCH Jr. が Maxima 形式入力および DragMath より有意に満足度が高い結果と

なった (MathTOUCH Jr.: \times Maxima: $U = 361, p < .05$, MathTOUCH Jr.: \times DragMath: $U = 389, p < .05$)。また、総合評価である再利用意向の項目では、MathTOUCH Jr. が Maxima 形式入力より有意に満足度の高い結果となった ($U = 413, p < .05$)。

4.4 考察

4.4.1 総合評価

タスク達成率 P_k は、3 つのインタフェースすべてがほぼ 90% 以上あり、 $P_{UI1}, P_{UI2}, P_{UI3}$ の間に有意差は認められなかったが、提案方式のタスク達成率が $P_{UI1} = 97.1\%$ と、最も高く、効果の信頼性は高い。逆に、有意差がないということは、実験前の練習による理解度が偏らなかったとみることができる。また、文系の被験者に限ってみると、タスク未達成者が Maxima 形式入力では 25.0%、DragMath では 16.3% であったのに対して提案方式の場合は 7.7% であり、提案方式は文系の初学者にも導入しやすいことが分かる。

平均タスク達成時間は、MathTOUCH Jr. が 24.3 秒と、Maxima 形式入力と比べて約 1.2 倍、DragMath と比べて約 1.6 倍速くなっており、数式の入力にかかる時間が有意に最も短いことが実証できた。

次に、アンケートによる主観満足度調査の結果、提案方式がすべての特性項目で最も高い評価平均値を得た。簡単のために、5 つの特性項目「理解しやすさ」、「効率性」、「間違えにくさ・修正しやすさ」、「記憶しやすさ」、「再利用意向」に対するインタフェース k の評価平均値をそれぞれ、 A_k, B_k, C_k, D_k, E_k で表す。一般に 5 段階 (評価値: 1~5) の評価平均値は 3.6 といわれている [34]。Maxima 形式入力および DragMath は 2 特性がそれぞれ $B_{UI2}, C_{UI2} < 3.6, B_{UI3}, C_{UI3} < 3.6$ となり、必ずしも好評価を得ているわけではないが、提案方式はすべて 3.8 以上で好評価を得た。特に、統計的にも以下の 3 特性において有意差が示され、満足度の高さが実証された。

- 効率性 : $B_{UI1} > B_{UI2}$
- 記憶のしやすさ : $D_{UI1} > D_{UI2}, D_{UI3}$
- 再利用意向 : $E_{UI1} > E_{UI2}$

*7 主観満足度アンケートの被験者数は、アンケートに未記入項目があった被験者が 1 人いたため 107 人となった。

MathTOUCH Jr. に関する自由記述のコメントでは、「見たまま打てばよいので簡単だった」、「式を直感的に読んでそのまま入力すれば大体そのとおり入力できた」といった意見があり、数式曖昧表記法に好感が持たれていることが分かった。

4.4.2 エラーについて

数式入力の場合、入力ミスによる軽微なものから本質的なものまで、質の違う様々なエラーが存在するため、本研究ではエラー率を測ることはできなかった。しかし、エラーもタスク達成率や達成時間に影響するため、タスク未達成者を含むすべての被験者を対象に、数式 E_j (j=1~6) 別のエラー内容を録画データより調査した。

Maxima 形式入力のエラーの内容は、E1~E4 までは、ほとんどが括弧のミス（括弧のつけ忘れ、丸括弧閉じの位置など）やアスタリスクのつけ忘れ、単なる入力ミス（特殊記号や数字のタイプミス）などであった。しかし、E5 と E6 についてはエラーの半数以上が \cos や \sin の冪乗の文法が原因であった。エラーの中には、 $\cos^4 x$ を “ $\cos^4 * x$ ” のように入力したものもあり、2次元形式の数式では見た目が同じため、注意が必要である。

DragMath は、E5、E6 のエラーが多く、数式が複雑になるほど操作によるエラーが増えることが分かった。自由記述のコメントにテンプレートの選択順序を迷ったという意見があり、2.2 節で述べた先行研究 [29] の指摘より、数式が複雑になると、数式構造の解析が難しくなるため、エラーが増えたと考えられる。

MathTOUCH Jr. のエラー内容は、いずれのタスクも変換ミス（変換忘れや誤った候補の選択）、オペランド範囲の指定操作のミスであった。自由記述のコメントを確認したところ、「赤線の範囲（オペランド範囲）の指定方法が難しかった」との意見があった。特にオペランド範囲の指定操作については、エラーの原因にもなっており、改善の必要性が示唆された。

5. まとめと今後の課題

本研究では、数式自動採点システムの数式入力方法の改善のために数式曖昧表記変換方式を採用し、数式自動採点システム向けに最適化を行った数式入力インタフェース MathTOUCH Jr. を開発・提案した。オープンソースの数式自動採点システムである STACK への実装に成功し、STACK における新たな数式入力の選択肢として、MathTOUCH Jr. プラグインを提供できるようになった。提案システムを高等学校生から大学生までの文系・理系を含む男女 35 人にタスクと同等の数式入力を一度自力で確認してもらった状態で 6 つの数式（表 4）入力タスクを実施してもらった結果、制限時間内に自力で正しく入力できた人数割合（タスク達成率）は 97.1%であった。さらに、1 つの数式あたりの平均入力時間は 24.3 秒であり、従来方式に比

べて有意に 1.2~1.6 倍短い時間であった。また、タスク実施後の提案インタフェースに対するユーザビリティ特性の満足度に関するアンケート調査の結果、3 特性「効率性」、「記憶のしやすさ」、「再利用意向」の 5 段階評定平均値が従来方式と比べ有意に高かった。特に、「再利用意向」の主観満足度平均値は 4.43（5 が最良）となり、好評であった。残りの 2 特性「理解しやすさ」と「間違えにくさ・修正しやすさ」に関する満足度も一般平均値より高い評価を得ており、提案システムの主観満足度の高さが実証された。

また、表 8 で示したように、文系と理系によって、タスクの負担に明らかな差があることから、IT を使った教育支援を行ううえでこれらの違いに配慮することは重要である。提案数式入力インタフェース MathTOUCH Jr. は、理系だけでなく、文系の数学初学者に対しても適用効果が高く、数式自動採点システム STACK を使った数学学習において数式入力の改善が期待できる。

しかし、以上の結果は数式入力インタフェース部分の操作性を評価したにすぎず、数式自動採点システムを使った学習支援に教育上有効に働くかどうかは明らかでない。また、評価実験の分析からオペランド範囲の指定操作については改善の必要性が示唆された。これらは今後の課題としたい。

謝辞 評価実験のために実験場所の提供および被験者集めにご協力いただいた甲南大学の高橋正教授、阿南工業高等専門学校 杉野隆三郎教授、太田健吾助教、児島雄志特命助教、大阪府立吹田東高等学校の中田裕省校長、大阪府立茨木西高等学校の若林智子教頭、武庫川女子大学附属中学校・高等学校の古川明教諭、武庫川女子大学の山崎彰教授、小花和 W. 尚子教授、本城精二教授、水谷孝子教授に心から感謝申し上げます。また、ご協力いただいた生徒、学生の皆様に心より御礼申し上げます。

参考文献

- [1] 特定非営利活動法人 日本イーラーニングコンソシアム：eラーニング白書 2008/2009 年版，東京電機大学出版局（2008）。
- [2] 勝野喜以子，勝野弘康，真野博史，入沢寿美：物理学科新入学者に対する入学前補習学習用 Web 教材の開発，学習院大学計算機センター年報，Vol.33，pp.139-145（2013）。
- [3] 船久保公一：物理専門科目における学習管理システムの活用，メディア教育研究，Vol.5，No.1，pp.67-75（2008）。
- [4] 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）。
- [5] 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）。
- [6] 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）。
- [7] 吉富賢太郎，川添 充：学習目標データベースを基盤とする数学到達度評価システムの開発，教育システム情報学会研究報告，Vol.27，No.2，pp.113-118（2012）。
- [8] Sangwin, C.J.: Computer Aided Assessment of Mathe-

matics Using STACK, *Proc. ICME*, Vol.12 (2012).

[9] 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, 数式処理, Vol.19, No.2, pp.33-36 (2013).

[10] 北本卓也: Maple T.A. の授業援用について, 京都大学数理解析研究所講究録, Vol.1907, pp.182-187 (2012).

[11] 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).

[12] 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).

[13] 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).

[14] Microsoft Corporation: Microsoft Word (online), available from <http://office.microsoft.com/ja-jp/word/> (accessed 2014-12-16).

[15] Sangwin, C.: STACK (online), available from <http://stack.bham.ac.uk/> (accessed 2014-03-28).

[16] 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).

[17] Ja STACK.org: Ja STACK.org (online), available from <http://ja-stack.org/> (accessed 2014-03-28).

[18] WeBWorK, available from <http://webwork.maa.org/index.html> (accessed 2014-12-16).

[19] Maplesoft: Maple (online), available from <http://www.maplesoft.com/products/Maple/> (accessed 2014-12-16).

[20] Maplesoft: Maple T.A. (online), available from <http://maplesoft.com/products/mapleta/> (accessed 2014-12-16).

[21] Wolfram Research: webMathematica (online), available from <http://www.wolfram.com/products/webmathematica/> (accessed 2014-12-16).

[22] 大阪府立大学高等教育推進機構: MATH ON WEB Learning College Mathematics by webMathematica (online), 入手先 <http://www.las.osakafu-u.ac.jp/lecture/math/MathOnWeb/> (参照 2014-03-28).

[23] Maxima: Maxima (online), available from <http://maxima.sourceforge.net/> (accessed 2014-12-16).

[24] Moodle: Moodle (online), available from <https://moodle.org/> (accessed 2014-12-16).

[25] Butcher, P., Sangwin, C.J. and Hunt, T.: Embedding and enhancing eAssessment in the leading open source VLE, *Proc. Higher Education Academy Conference*, Birmingham (2013).

[26] Pollanen, M., Hooper, J., Cater, B. and Kang, S.: A Tablet-Compatible Web-Interface for Mathematical Collaboration, *International Congress on Mathematical Software 2014*, Lecture Notes in Computer Science 8592, pp.614-620, Springer (2014).

[27] 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).

[28] DragMath: DragMath (online), available from <http://www.dragmath.bham.ac.uk/> (accessed 2014-03-28).

[29] Pollanen, M., Wisniewski, T. and Yu, X.: XPRESS: A Novice Interface for the Real-Time Communication of Mathematical Expressions, *Proc. MathUI2007* (2007).

[30] Smithies, S., Novins, K. and Arvo, J.: Equation Entry and Editing via Handwriting and Gesture Recognition, *Behaviour and Information Technology*, Vol.20, No.1, pp.53-67 (2001).

[31] Anthony, L., Yang, J. and Koedinger, K.R.: A paradigm for handwriting-based intelligent tutors, *International*

Journal of Human-Computer Studies, Vol.70, Issue 11, pp.866-887 (2012).

[32] 糟谷勇児, 山名早人: 手書き数式入力システム MathBox, 電子情報通信学会技術研究報告, PRMU, パターン認識・メディア理解, Vol.106, No.606, pp.1-6 (2007).

[33] 樽本徹也: ユーザビリティエンジニアリング—ユーザ調査とユーザビリティ評価実践テクニック, 株式会社オーム社 (2009).

[34] 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).

[35] 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 2014-12-027).

[36] 森 敏昭, 吉田寿夫: 心理学のためのデータ解析テクニカルブック, 株式会社北大路書房 (1992).



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科修士課程修了。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。



仲村 裕子 (正会員)

2012年武庫川女子大学情報メディア学科卒業。2014年同大学院生活環境学研究科修士課程修了。現在, 武庫川女子大学附属中学校・高等学校非常勤講師。



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。1994年武庫川女子大学生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。