

MathDox を活用した STACK への数式入力インターフェースの追加

中村泰之*1・稲垣佑亮*2・中原敬広*3
Email: nakamura@nagoya-u.jp

*1: 名古屋大学情報科学研究科

*2: 名古屋大学情報文化学部

*3: 合同会社三玄舎

◎Key Words 数学オンラインテスト, STACK, 数式入力インターフェース

1. はじめに

英国バーミンガム大学の Sangwin (現英国ラフバラ大学) らによって、2004 年から開発が始まった数学オンラインテスト・評価システム STACK (System for Teaching and Assessment using a Computer algebra Kernel)⁽¹⁾は、数式処理システムを活用することにより、オンラインテストにおいて数式で提示された解答の正誤評価を行うことのできるシステムである。現在は Moodle の小テストの問題タイプの一つとして提供されている。自然科学教育などにおいては、従来からよく用いられた問題タイプである、正誤選択タイプ、多肢選択タイプでは計ることのできない、個々の問題に対する理解度を把握することが STACK 問題タイプには期待できる。つまり、多肢選択タイプでも多数の問題を課せば、受験した学生の全体的な理解度の参考にすることはできるが、多肢選択式だと、個々の問題について、「あてずっぽう」の度合いを除外できないために、各問題の理解度を個別に計ることは困難である。しかし、数式によって解答させることで、「あてずっぽう」の要素を省くことができ、理解度をより正確に判定することが可能となると考えられる。

我々は STACK のそのような特性に注目すると同時に、オープンソース・ソフトウェアであることを活かして、様々な機能拡張を行い、STACK 利用の普及にも努めてきた⁽²⁾。その過程で STACK の普及を妨げるいくつかの問題点があると考え、その解決に向け様々な対応を行っている。例えば、良質な問題を提供することは、STACK を利用する場合に限らず、全てのオンラインテストに求められることであるが、STACK での問題作成は、Moodle の他の問題タイプの問題作成と比較して複雑である。これは、学生が提示した解答に対して、適切なフィードバックを与えることのできるポテンシャル・レスポンス・ツリーという機構を備えていることに起因する。ポテンシャル・レスポンス・ツリーは非常に特徴的であり、有益な機構である反面、これを適切に活かした問題を作成することは、STACK を運用するにあたっての高いハードルとなっていた。我々はその問題点を解消するために、問題作成ツールの開発⁽³⁾、問題を共有するためのしくみである、問題バンクシステムの開発^(4,5)を行うことにより、一定の解決策を示したと考えている。

しかし、未解決のもう一つの問題点があると考えて

いる。それは、学生が解答する際に、数式を入力するためのインターフェースの問題である。本報告では、この問題を解決するための一つの提案として、MathDox formula editor というシステムを活用した STACK の数式入力インターフェースの追加について報告したい。

本報告は次のように構成されている。まず、次節で既存の数式入力インターフェースについていくつか紹介し、今回我々が採用した MathDox formula editor の STACK への導入方法を 3 節で述べる。そして、4 節で利用実証実験の結果を紹介し、最後にまとめと今後の課題について述べる。

2. いくつかの数式入力インターフェース

2.1 テキスト入力

STACK での基本的な数式入力はテキスト入力である。 $\frac{x}{2}$ は $x/2$ 、 x^2 は x^2 などと入力する方法であり、単純な数式であれば入力ミスも少なく、比較的手軽に入力することが可能である。一方、 $\frac{1}{1+\cos(x^2)}$ を微分せよという問題に対して、正答である $\frac{2x \sin(x^2)}{(1+\cos(x^2))^2}$ を解答するためには $(2*x*\sin(x^2))/(1+\cos(x^2))^2$ と入力する必要があり、注意深さと慣れが要求される。

2.2 DragMath

DragMath⁽⁶⁾という数式入力インターフェースは、図1のように数式の形式をパレットからドラッグ・ドロップを主としたマウス操作により選択し、数式を構成していく方法を採用しており、複雑な数式の入力ミスが軽減されることが期待できる。また、構成された数式を TeX や MathML 形式、さらに STACK が採用している数式処理システムの Maxima 形式で出力することが可能で、STACK2 では数式入力インターフェースとして採用されていた^a。一方短所として、Java アプレットによって動作するため、Java プラグインのインストー

^a 現在の STACK の最新バージョンは STACK3.2 であるが、DragMath は数式入力インターフェースとしては採用されていない。

ルが必要である点、近年普及しているスマートフォンやタブレット端末の中には、Java アプレットに対応していないものが多く、環境に依存するという点で、STACK を利用する場合には注意が必要となる。

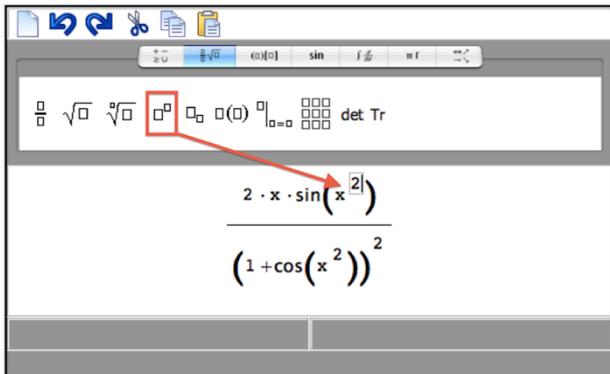


図1 DragMathによる数式入力例

2.3 Suim

Suim⁽⁷⁾は町野によって開発された、Web上で動作する数式入力インターフェースで、JavaScriptによって開発されている。Suimでは日本語インプット・メソッド・エディタ(IME)と同様に、数式入力のキーワードを入力した後にキーボードのスペース・キーを押すことにより数式に変換する方式を採用している。また、日本語変換などにも採用されている予測変換機能が備わっており、例えば根号($\sqrt{\quad}$)を入力したいときには、**sqrt**や**root**と入力する必要があるが、図2のように**ro**と入力したところで、根号を候補として提示してくれる。このように、日本語IMEとの類似性により、特に日本人には馴染みやすいであろうということ、非常に軽快に入力が可能であることなどの利点がある一方で、数式の出力形式はTeXおよびMathML形式であり、STACKで利用するために必要なMaxima形式の出力に対応していない点、対応しているブラウザがFirefoxのみである点などが欠点としてあげられる。

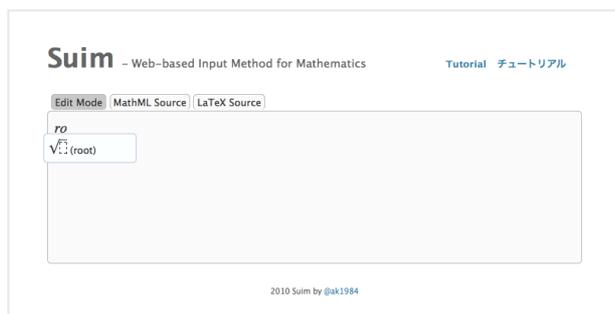


図2 Suimにおける数式入力予測変換機能

2.4 MathTOUCH

MathTOUCH⁽⁸⁾は福井によって開発された数式入力インターフェースで、Suimと同様に日本語IMEのように、入力後、スペース・キーで変換していく入力方式である。MathTOUCHの特徴は数式入力の順番が、人が数式を読みあげるのと同じ順番を採用していることである。

例えば、 x^2 は「 x の2乗」というふうに読み上げるが、その場合「**x2**」と入力し、変換すればよい(図3)。

MathTOUCHは数式の出力形式としてTeX, MathMLだけでなく、PNGやJPEGなどの図としての出力も可能であり、さらに、最近STACKの数式入力インターフェースとしても利用できるように拡張がなされた⁽⁹⁾。実際にSTACKの数式入力インターフェースとしてMathTOUCHを採用した場合にも、少なくとも従来のテキスト入力方式とは変わらない数式入力効率であったことが報告されている。一方で、DragMathと同様にMathTOUCHはJavaアプレットとして動作するため、スマートフォンやタブレット端末では利用できない場合があることに留意しておく必要がある。



図3 MathTOUCHによる数式入力例

2.5 MathDox formula editor

MathDox formula editor^(10, 11)はオランダのアイントホーフェン工科大学で開発されている、数学ドキュメントを作成するためのソフトウェア群であるMathDox⁽¹²⁾の一つとして提供されている、数式入力のためのWebインターフェースである。JavaScriptで開発されており、利用環境にあまり依存しないという特徴があり、入力はキーボード操作により「2次元的」な数式の入力を可能とするものである。つまり、 $\frac{x}{2}$ を入力するために、

キーボードから**x**を入力し、その後「/」記号を入力することにより、2次元的な分数形式の表示となり、分母の部分が入力可能状態となる(図4)。さらに、必要に応じて図5に示される数式入力パレットを利用して数式入力を行うことも可能である。ただし、数式の出力形式はMathMLとOpenMath⁽¹³⁾形式のみであり、STACKで必要とされるMaxima形式には対応していない。



図4 MathDox formula editorによる「2次元的」な数式入力例

我々は MathDox formular editor の、テキスト入力による直感的な 2 次元的な数式入力が可能であること、必要に応じてパレットを用いて複雑な数式入力の支援がなされていること、また、JavaScript によって開発されているため端末の環境にあまり依存しないことが期待されることなどから、STACK の数式入力インターフェースとして採用することを検討した。

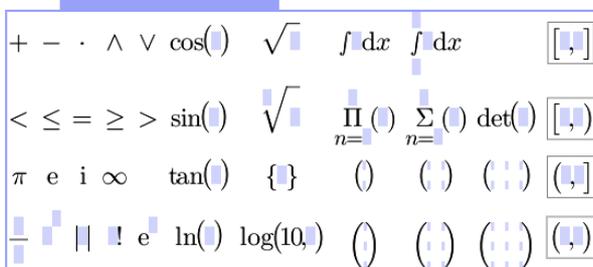


図5 MathDox formula editor の数式入力パレット

3. MathDox formula editor と STACK との連携

MathDox formula editor (以後, MathDox と略記する) と STACK とを連携させるにあたっては、数式の出力形式として Maxima 形式を可能にすること、その上で、MathDox を STACK の新しい数式入力タイプとして追加・実装するという作業が必要となる。

3.1 Maxima 形式への出力変換フィルタの作成

我々は、MathDox で作成した数式を Maxima 形式に変換するための方法として、元々 MathDox が出力している MathML, OpenMath 形式を介して Maxima に変換することとした。具体的には、(i) タグを利用して、MathML 形式で表現された数式を項に分割すること、(ii) 各項ごとに数式のタイプ (分数, 指数関数, 三角関数, 対数, 絶対値, 行列など) に応じて Maxima 形式への変換処理を行うものとする。

3.2 STACK の数式入力タイプとしての実装

我々は、MathDox を STACK で利用するために、STACK 専用の input タイププラグインとして実装を行った。STACK では、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、必須関数を含むファイルを梱包することで input タイプとして認識され、解答欄での利用が可能となる。この input プラグイン用のディレクトリの中に、MathDox を動作させるために必要な JavaScript ファイルや CSS ファイル、およびフォントファイルなどを梱包した。また、前項で紹介した MathML 形式、OpenMath 形式から Maxima 形式へ変換を行う関数も JavaScript の外部スクリプトファイルとして、ここに同梱した。

3.3 STACK での MathDox の利用

図6はSTACK問題タイプの小テストの解答において、実際に MathDox を利用している様子である。テキスト入力だけでなく、数式入力パレットが利用出来るように実装されていることがわかる。なお、パレットは表示・非表示の切り替えが可能であることのみならず、

表示させた場合には好みの位置に移動させることも可能である。

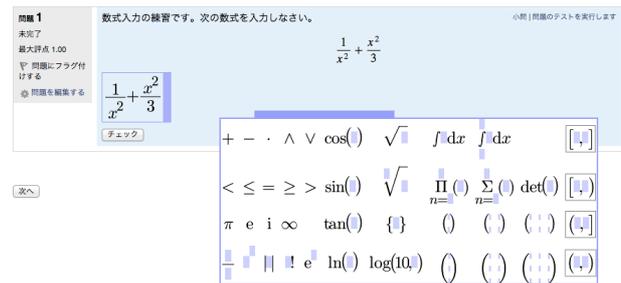


図6 STACK 問題タイプの小テストの解答に MathDox を利用した様子

4. MathDox の利用実証実験

我々が作成した変換フィルタにより、MathDox が Maxima 形式に正しく変換されていることの検証と、他の数式入力インターフェースと比較して、実際に MathDox を利用した場合の効率を検証するために、簡単な実証実験を行った。実験方法は次のとおりである。実験協力者は名古屋大学情報文化学部自然情報学科 4 年の学生 5 名である。テキスト入力, DragMath, MathDox の 3 種類の方法で、提示された 5 つの数式を入力してもらう。5 つの数式入力に要した合計時間を計測し、入力方式毎に 5 人の平均入力時間を算出し、それを効率として比較する。また、実験終了後に、実験協力者 5 名に 3 種類の入力方式の使いやすさを順位付けしてもらった。なお、この実証実験では、各入力方式が STACK の入力インターフェースに組み込まれた状態のものではなく、単体の数式入力インターフェースのものを利用し、テキスト入力の場合は Maxima のインターフェースに直接入力することとした。実験に用いた数式は次の 5 種類である。 $(a+b)^2$, $\frac{y}{1+x}$, $\sqrt{2} + \sqrt[3]{x+y}$, $e^x + \sin(\pi x)$, $\frac{1+2x \sin(x^2)}{(1+\cos(x^2))^2}$ 。

実証実験の結果は表 1 のとおりである。入力時間ではテキスト入力が最も短いことがわかる。しかし、一方で使いやすさでは MathDox が上回っており、5 人中 4 人が MathDox の使いやすさを 1 位に順位付けている。このことから、多少入力時間は長くかかったとはいえ、それはわずかであるし、直感的に数式を構成できる MathDox のほうが入力方式としては望ましいことがうかがえる。

表1 数式入力方法の利用実証実験の結果(表中の A~E は実験協力者 5 名に対応する)

	平均入力時間	使いやすさ (順位)				
		A	B	C	D	E
テキスト入力	2分48秒	2	2	3	1	2
DragMath	5分18秒	3	3	2	3	3
MathDox	3分09秒	1	1	1	2	1

実験協力者からのコメントとしては次のようなものが挙げられた。「短い数式はテキスト入力で十分だが、長い式は MathDox のほうが入力しやすい」「MathDox はテキスト入力と DragMath を合わせたような操作ができるのが良い」「テキスト入力は、キーボードのみで入力できるため、書式がわかっているれば使いやすい。つまり、MathDox は 2 次元的な数式表現故にわかりやすい一方、短い数式はテキスト入力で十分であるため、入力方式を選択できればよいということが推測される。

今回の実験協力者は、卒業論文作成の経験などから、TeX など数式のテキスト入力に関して全くの初心者というわけではないため、テキスト入力にもあまり抵抗のない様子がかがえた。したがって、高校生や、理数科系を専門としないような学生など、数式入力にあまり慣れていない学生が参加して同様の実証実験を行えば、異なった結果が得られる可能性もあり、利用者に応じた柔軟な数式入力インターフェースを提供できる環境が望まれる。

5. まとめと今後の課題

本報告では、MathDox formula editor を、数学 e ラーニングシステム STACK の input タイププラグインとして実装することにより、新たな数式入力インターフェースを提供することができた。また、簡単な利用実証実験を行うことにより、MathDox の有効性が示唆される一方で、入力する数式の複雑さの程度、あるいは利用者の経験などに応じて、数式入力方法が選択的に利用できることが望ましいことも判明した。

また、技術的な問題点として次のようなことを指摘しておく必要がある。実際に MathDox を利用して STACK 上で解答を行った際に、入力する数式の確認画面において、入力した数式が解答欄から消えてしまうという不具合が発生した。これは、Maxima 形式に変換してデータベースに保存されたものを読み込んだ際に OpenMath 形式でのみ表示が可能な MathDox 入力エリアにおいて、Maxima 形式の数式を描画することができないために発生する。そこで、当面の解決策として利用者のブラウザに COOKIE として OpenMath 形式の数式を保存し、画面遷移後 COOKIE を取り出し解答欄に値を挿入し描画を行うこととした。取り出された数式データは即時 COOKIE から削除される。これにより、受験中に回答データが解答欄から消失することは無くなったが、正式な処理としては Maxima 形式を OpenMath 形式に変換する必要がある。早期にこの変換処理を開発する必要がある。また、MathDox は JavaScript で開発されているが、スマートフォン、タブレットでの利用には対応しておらず、今後の課題として対応を検討していきたい。

謝辞

本研究は JSPS 科研費 26282033 の助成を受けたものです。

参考文献

- (1) Chris Sangwin: "Computer Aided Assessment of Mathematics", Oxford University Press (2013)
- (2) Ja STACK: <http://ja-stack.org/>
- (3) 中村泰之, 大俣友佳, 中原敬広: "STACK の問題作成

ールの開発と STACK3 に向けて", 数式処理, Vol. 19, No. 2, pp.33-36 (2013).

- (4) 中原敬広, 中村泰之, 谷口哲也: "数学問題バンク構築に向けた Moodle プラグインの開発", 第 38 回教育システム情報学会全国大会講演論文集, A1-2 (2013)
- (5) Mathbank.jp: <http://mathbank.jp/>
- (6) DragMath: <http://www.dragmath.bham.ac.uk/>
- (7) Suim: <http://suim.herokuapp.com/>
- (8) MathTOUCH: <http://math.mukogawa-u.ac.jp/>
- (9) 白井詩沙香, 仲村裕子, 福井哲夫: "数式入力 UI 「MathTOUCH」の Maxima 形式出力機能の拡張と数学 e ラーニングへの応用", 数式処理, 発行予定 (2014)
- (10) MathDox formula editor: <http://mathdox.org/formulaeditor/>
- (11) Matthijs Brouwer, Hans Cuypers, Jan Willem Knopper: "MathDox editor", Proceedings of Mathematical User-Interfaces Workshop 2009 at the eighth Mathematical Knowledge Management Conference, 2009 <http://www.activemath.org/workshops/MathUI/09/proc/Cuypers-Knoppers-Brouwer-MathdoxEditor-MathUI09.pdf>
- (12) MathDox, Interactive Mathematics: <http://mathdox.org/new-web/index.html>
- (13) OpenMath: <http://www.openmath.org/>