

RESTful Web APIとマッシュアップ技術を活用した Web コンテンツ内数式処理と入出力及び可視化の検討

中野 裕司^{1,2,3,4,a)} 永井 孝幸^{1,3,2,4} 中村 泰之^{5,6} 稲垣 佑亮⁶ Wannous Muhammad^{7,1,8}
喜多 敏博^{4,2,1} 宇佐川 毅^{3,2,1}

概要：我々は Maxima とマッシュアップ技術による Web コンテンツ中での数式処理と可視化について研究を行ってきた。JSONP によりクロスドメインでどこからでもアクセスできる Web API を通して、Maxima による数式処理と MathML への変換サービスを提供し、クライアント上の JavaScript で HTML5 を利用して実装されたプロットライブラリ Flot によるグラフの可視化等を実現してきた。今回、Web API を RPC からより汎用的な REST へ変更し、MathDox を用いた GUI による数式入力や MathJax による数式表示に関する検討を行い、数式入力の可視性が高め入力を容易にし、Web ブラウザ異存を低減したので報告する。

1. はじめに

動的な数式処理やその結果のグラフ表示は、学習者が学習内容を理解する上で有用であると考えられるが、LMS (Learning Management System) 等の Web 上の学習コンテンツでこれを実現するには様々な問題がある。

数式処理を実現するには、Web インターフェースを含んだ専用の数式処理アプリケーションと、場合によっては利用範囲に応じたライセンスが必要となる。また、数式表示

には $\text{T}_\text{E}_\text{X}$ 等で出力したもののイメージ化やプラグイン等が必要で、グラフ表示もプロットツールのイメージ出力や専用プラグイン等が必要な場合が多い。また、それらを動的に連携して用いるには、セキュリティ上の制約(クロスサイトスクリプティング)から、コンテンツ配信サーバ側に専用 Web アプリケーションを開発する等の必要があった。

具体的にコンテンツ中で数式処理を扱うには、数式処理ソフトの Web フロントエンド [1] で取り組んだ例が多いが、我々は、Web コンテンツが LMS だけでなく、どのような Web サーバ上にあっても、また、ローカルファイルであっても、数式処理が可能であることを目指し、クライアントの JavaScript のみによる複数サービスのマッシュアップによる解決を試みてきた [7], [8]。具体的には、オープンソース数式処理ソフト Maxima[2], JSONP[3], Ajax (jQuery), MathML[4], FLOT[5] (HTML5 の Canvas[6] 利用) を活用により、その実現性と応用可能性を示すことができた。さらに、eラーニングコンテンツの標準仕様である SCORM[9] 上での動作するサンプルを作成することで、SCORM コンテンツ中で本システムが利用可能であることを確認した [10]。

ただし、いくつかの問題点もあり、比較的大きな問題として、数式入力形式が Maxima の入力形式であり、そのルールを知らなければならないこと、視覚的にわかり難いことが挙げられ、コンテンツ作成時にも問題となるが、特に、学習者自身に数式入力をさせるような動的なコンテンツでは大きな問題となる。図 1 に示すように、入力欄に Maxima 形式で数式をテキスト入力する必要があった。

¹ 熊本大学 総合情報統括センター
Center for Management of Information Technologies, Kumamoto University, Kumamoto 860-8555, Japan
² 熊本大学 大学院社会文化研究科 教授システム学専攻
Graduate School of Instructional Systems, Kumamoto University, Kumamoto 860-8555, Japan
³ 熊本大学 大学院自然科学研究科研究科 情報電気電子工学専攻
Graduate School of Computer Science and Electrical Engineering, Kumamoto University, Kumamoto 860-8555, Japan
⁴ 熊本大学 eラーニング推進機構
Institute for e-Learning Development, Kumamoto University, Kumamoto 860-8555, Japan
⁵ 名古屋大学大学院情報科学研究科
Graduate School of Information Science, Nagoya University, Nagoya464-0814, Japan
⁶ 名古屋大学情報文化学部
School of Informatics and Sciences, Nagoya University, Nagoya464-8601, Japan
⁷ 日本学術振興会 (外国人海外特別研究員)
Japan Society for Promotion of Science (JSPS Postdoctoral Fellowship for Overseas Researchers)
⁸ 国際科学技術大学, シリア
International University for Science and Technology (IUST), Syria
a) nakano@cc.kumamoto-u.ac.jp

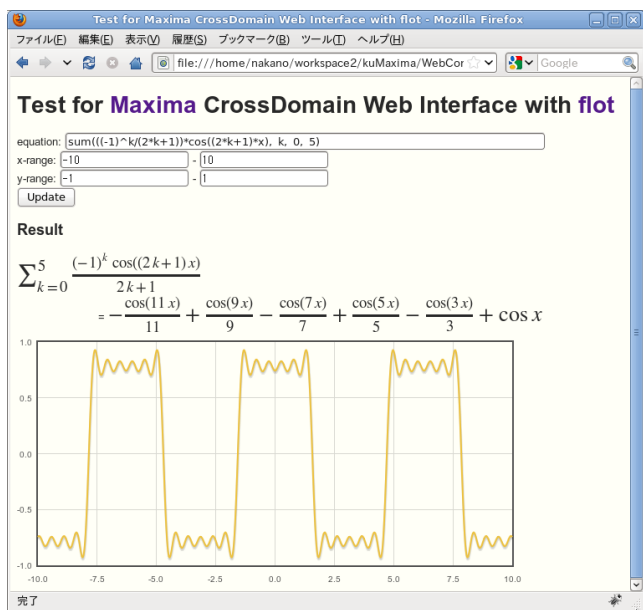


図 1 動的な数式入力と計算結果のプロットの例 [7], [8]

Fig. 1 A Sample web page for interactive inputting formula and plotting a graph [7], [8].

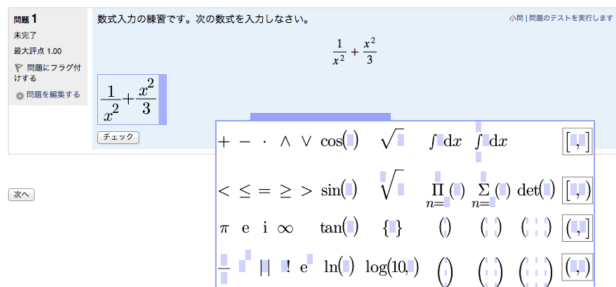


図 2 STACK 問題タイプの小テストの解答に MathDox を利用した様子 [11]

Fig. 2 Applying MathDox for answering the STACK's quiz [11].

この問題を解決するには、数式入力エディタで Maxima 形式の出力が可能なのが必要となる。Web コンテンツ自体に組み込んで利用可能な JavaScript による数式入力エディタには、画像イメージや $T_E X$ や MathML[4] 形式で出力可能なものは知られているが、Maxima 形式で出力できるものは殆ど知られていない [11]。ここでは、Moodle[12] の小テストの問題タイプの一つとして提供されている数学オンラインテスト・評価システム STACK[13] 用に開発された、MathDox[14] の数式エディタである MathDox formula editor の Maxima 形式出力対応 [11] を利用して数式入力の実装を図った。図 2 に、STACK 上で MathDox formula editor による数式入力を行っている様子を示す。

数式の表示は MathML を利用して行っていたが [7], [8]、ブラウザやそのバージョンに多少依存することや、ズーム機能等有用な機能が多数実装されており、多くの学会等で利用されていることから、今回新たに、数式表示に [15] を

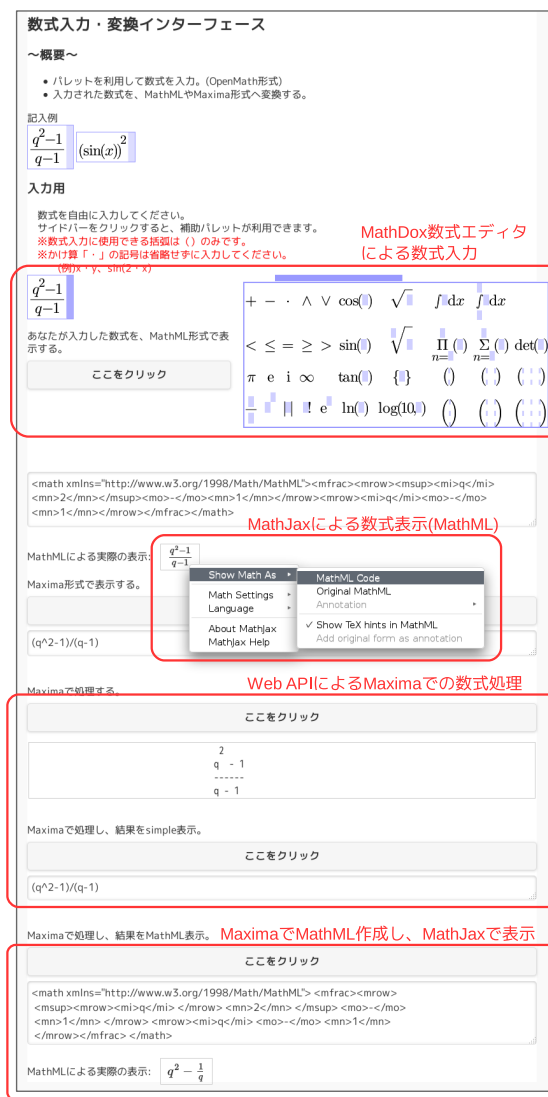


図 3 MathDox 数式エディタのデモに Maxima 連携を追加

Fig. 3 Applying Maxima's functions to the original demonstration of MathDox.

利用することにした。

また、Web API に関しても、これまでの RPC[16] から、最近 Web Service として多く使われている REST[17] へ変更し、より汎用性を高めた。

2. MathDox 数式エディタの Maxima 出力拡張

図 3 に、MathDox 数式エディタのデモ (日本語翻訳版) に、今回行った Maxima 連携や MathJax による表示等のデモを追加したものを示す。本稿では、基本的にはこのデモに沿って解説する。

MathDox 数式入力、すなわち、MathDox formula editor に、Maxima 形式の出力機能を付加したを利用しているが、基本的には [11] で開発したものを外部 JavaScript ファイル化したものである。実装は、MathDox の生成する MathML を Maxima 形式へ変換するものであって、図 3 上部に表示

MathDox の既存デモ

Maxima 連携、MathJax による表示の追加分

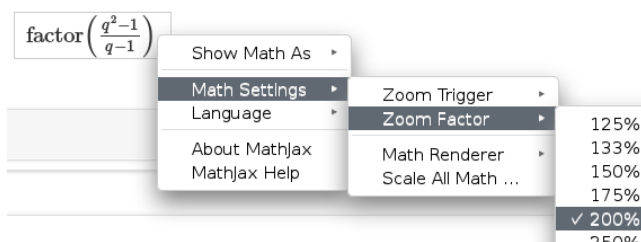
入力用

数式を自由に入力してください。
サイドバーをクリックすると、補助パレットが利用できます。
※数式入力に使用できる括弧は () のみです。
※かけ算「 \cdot 」の記号は省略せずに入力してください。
(例) $x \cdot y, \sin(2 \cdot x)$



あなたが入力した数式を、MathML形式で表示する。

(a) Maxima 対応版 MathDox 数式エディタによる数式入力
(a) Editing an equation by MathDox's equation editor.



(b) 上記入力の MathJax による表示
(b) Showing the above equation by MathJax.

図 4 数式入力と表示例

Fig. 4 Editing and showing an equation.

しているパレットの一部は未実装な部分も残っているが、多くは実装されている。この方式であれば、MathDox 以外でも MathML をリアルタイムに取り出せる数式エディタであれば、他のエディタにも応用できる可能性がある。この方法で、通常の数式入力に関しては、[11] でも評価されているように、格段に容易かつ視覚的になった。

図 4 に、Maxima 出力に対応させた MathDox 数式エディタによる数式入力と、入力された数式の MathJax による表示の例を示す。図 4(a) に示すように、数式を見やすい形で入力でき、右のパレットやキーボード [11] から比較的容易に入力でき、解説書等の必要性はあまり感じなかった。しかし、「*factor*」は因数分解の意味であるが、これは Maxima のコマンドを知らない限り入力できない。

図 4(b) では、図 4(a) で入力した数式を MathJax で表示している。図からわかるように、MathJax を用いると、多くのブラウザで数式表示が同様に行えるようになるだけでなく、マウスの右クリックで様々な付加機能が使えるようになり、表示方法として、CSS、MathML、SVG 等が選べたり、クリック時にズームしたり、MathML コードをコピーできる等、様々な利点がある。ただし、この場合のように、数式の編集等を行った場合は注意が必要であり、図 5 にあるように、数式に変更があった場合は、MathJax で更新を行わないと変更が反映されない。ユーザが直接このようなコードを書くことはあまりないと思われるが、ライブラリ化した際に組み込む必要がある。

```
$("#MMLxml").html(x2answer); // MathML 表示
MathJax.Hub.Queue(["Typeset", // MathJax 再適応
MathJax.Hub, "MMLxml"]);
```

図 5 MathJax による動的な MathML 表示用 JavaScript
Fig. 5 A JavaScript code for refreshing MathML by MathJax.

図 6 に、Web API 経由の Maxima による数式評価の結果出力について示す。Maxima は、式の評価結果をいくつかのタイプの出力で返すことができる。ここでは、MathML だけでよいようにも思われるが、特に、図 6(c) に示す「*simple*」タイプは、式の数学的意味を保持していることから、評価の結果を次の Maxima の入力とし、さらに評価を続けることがかくなることもあり、いくつかの出力タイプをサポートすることが必要であろう。

この例では、図 6(a) にあるような級数を Maxima に計算させており、MathJax のズーム表示をさせた例を示しているが、認識性はかなりよいと思われる。図 6(b) が Maxima の標準の出力であるが、式の意味が失われ、MathML より可視性も悪いいため、この場合は、あまり利用価値がないかもしれない。図 6(c) は、*simple* タイプで、可視性は悪いが、式の意味を保持している。

図 6(d) は、MathML で返された結果を MathJax で表示した例で、結果が可視性の高い数式として正しく表示されていることがわかる。

以上より、Maxima 形式に対応させた MathDox の数式エディタと MathDox による表示を組み込むことで、数式入力が容易かつ認識性が上がり、また出力もブラウザ依存が減りかつ様々な機能が付加された。しかし、まだ完全というわけではなく、現在わかっている問題点を以下に列挙する。

- 数式以外の Maxima コマンドへの対応 (微分、積分、展開等したいときに Maxima コマンドを知っている必要がある)
- Maxima の MathML 以外の出力形式 (*simple* 等) へのより完全な対応
- Maxima では許されるが数式エディタでは許されない文字、またはその逆への対策
- 未実装の関数等への対応
- MathDox の本家の開発が滞っている (止まっている?) ことへの対応

3. JSONP REST 形式 Web API

Web API に関しては、これまで RPC[16] 形式のインターフェイスを用いていたが [7], [8], [9]、最近 Web Service として REST[17] 形式が多く使われているため、移行することにした。幸い、サーバ側で利用していた、ライブラリである JSONIC[19] が、RPC、REST 両方に対応してい

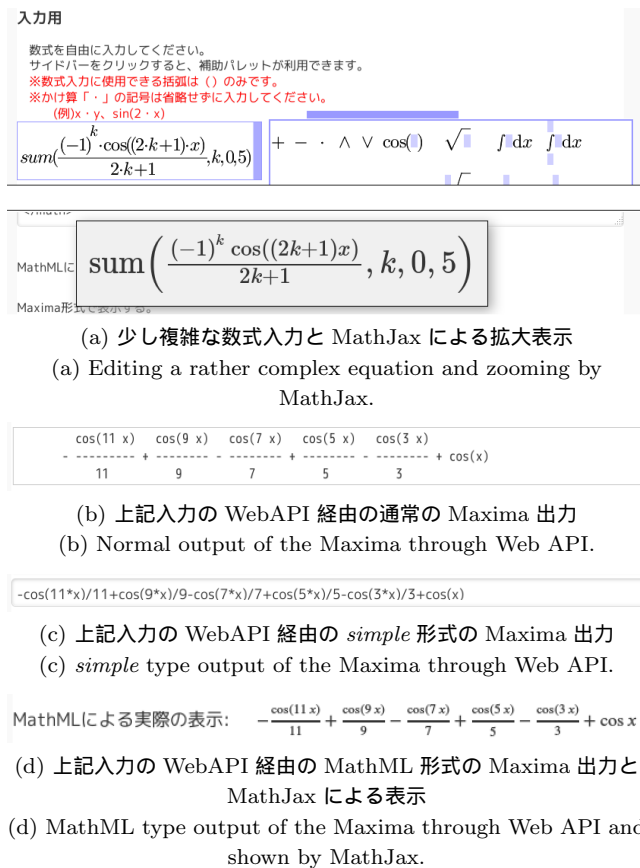


図 6 Web API 経由の Maxima の出力と表示例

Fig. 6 Various type of the Maxima's output through WebAPI.

ため、あまり大きな修正なしに REST 形式に移行できた。データ形式は以前同様にクロスドメインで利用可能な JSONP で、ローカルなファイルであっても Web ブラウザで開くと、Web API で通信が可能である。

図 7 に、Web ブラウザ側の jQuery を利用した Web API の呼び出し JavaScript コードの基本部分を示す。コンテンツ製作者はライブラリ経由で利用するため JavaScript に直接触れることはあまり想定されないが、この例に示すとおり非常に簡単に呼び出すことができる。他の WebAPI とマッシュアップを行うコンテンツや、他の Web アプリケーションやモバイルアプリケーションからも容易に呼び出せることがわかる。ここでは基本部分の紹介に留めるが、実際には、HTML への id による数式入出力の埋め込みインターフェース、グラフのプロット、例外処理等が必要である。

図 8 に、JSONIC を利用した Web API の待ち受け側の REST サーブレットの Java コードの基本部分を示す。パラメータの受け取りや JSON による戻り値のセットが、簡単に行えることがわかる。また、ここでは出力形式によって、Maxima に渡すコマンドを変えているが、MathML 形式の出力は、mathml.lisp[20] を利用している。ここでは基本部分の紹介に留めるが、実際には、パラメータが NULL の時の処理や、インジェクション攻撃対策 (Maxima で実

行してはならない命令の除去)、複文の処理、グラフのプロット、例外処理等々を行わなければならない。

4. MathJax による数式表示

図 9 に、MathJax を利用する場合の HTML のサンプルコードを示す。設定により異なるが、この場合は、[21] に従い、TeX と MathML が利用可能で、米数学会系の設定となっている。コードのなかで、article タグの開始・終了タグの間に、MathML 形式のコードを記述すれば、自動的に MathJax がそのコードに従った数式を表示する。ただし、動的に MathML を書き換える場合は、図 5 に示した更新処理を行わなければならない、再描画されない。

5. まとめと今後の課題

本研究では、これまで進めてきた、Maxima とマッシュアップ技術による Web コンテンツ中での数式処理と可視化において、Maxima 形式に対応させた MathDox の数式エディタを組み込むことで、数式入力を容易にするともに可視性を高めた。また、MathJax を利用することで、数式表示の Web ブラウザ依存性を削減するとともに、ズーム等の多くの付加機能を得た。さらに、JSONP によりクロスドメインでどこからでもアクセスできる機能はそのままに、Web API を RPC からより多く使われている REST に変更することで、より汎用性を高めた。

しかし、数式入力機能の実装に伴い、数式以外の Maxima コマンドへの対応、Maxima と数式エディタで利用可能な文字の違い、未実装の関数等への対応、MathDox の本家の開発が滞っていること等課題も多い。

ただし、以前より、数式が可視性が高く容易に入力可能になったことは大きく、今後さらに発展させていきたい。

謝辞

本研究は JSPS 科研費 24501195,25280124 の助成を受けたものです。

参考文献

- [1] webMathematica : <http://www.wolfram.com/products/webmathematica/> ,
MapleNet : <http://www.maplesoft.com/products/maplenet/> ,
Sage : <http://www.sagemath.org/> ,
MaximaPHP : <http://maximaphp.sourceforge.net/> 等
- [2] ホーム : <http://maxima.sourceforge.net/>
- [3] 高橋登史朗 : jQuery 入門 魔法の JavaScript リファレンス&サンプル, 秀和システム, (2009/11) p.349.
- [4] ホーム : W3C Math Home, <http://www.w3.org/Math/>
- [5] ホーム : <http://code.google.com/p/float/>
- [6] HTML5 Canvas :
<http://www.w3.org/TR/html5/the-canvas-element.html#the-canvas-element>
- [7] 中野裕司, 賈雲鵬, Muhammad Wannous, 永井孝幸, 喜

```
function execute(id, outstyle, command) {
  $.getJSON("http:// ホスト名 /maximaExecute.json?callback=?", // 形式ではJSONP callback=? が必須
    { outstyle: outstyle, command: command}, // 呼び出しパラメータ
    function(data) {
      $("#"+id).text(data.result); // 戻り値の処理
    });
}
```

図 7 Web API 呼び出し側 JavaScript の基本部分 (jQuery 使用)

Fig. 7 Essential JavaScript code for calling the Web API (using jQuery).

```
public class MaximaExecuteService {
  public HttpServletRequest request;
  public LinkedHashMap<String, Object> find(Map<String, Object> params) {
    LinkedHashMap<String, Object> ret = new LinkedHashMap<String, Object>();
    String outstyle = params.get("outstyle").toString();
    String command = params.get("command").toString();
    if(outstyle.equals("simple")) {
      command = "display2d:false$"+command + ";"; // simple 形式の出力
    } else if(outstyle.equals("mathml")) {
      command = "load(\"mathml.lisp\")$mathml(\"+command+\")$"; // MathML 形式の出力
    } else {
      command = command + ";"; // 通常の出力
    }
    String result = (new Maxima()).executeRawCommand(command).toString(); // Maxima の実行 (省略)
    ret.put("result", result); // JSON による戻り値のセット
    return ret;
  }
}
```

図 8 Web API 待ち受け側 Java の基本部分 (JSONIC 使用)

Fig. 8 Essential Java code for responding the Web API (using JSONIC)

```
<head>
...
<script type="text/javascript"
  src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML"></script>
...
<body>
...
<article>
( MathML で記述すると、 MathJax が表示する )
( MathJax automatically display MathML written here )
</article>
...
```

図 9 MathJax を利用するための JavaScript [21]

Fig. 9 JavaScript code for using MathJax

[8] 多敏博：数式の処理と表示のマッシュアップを実現する Maxima 活用システムの開発，教育システム情報学会第 35 回全国大会発表論文集，pp.381-382 (2010).
Nakano, H., Nagai, T., Jia Y., Wannous, M. and Kita, T.: Mashup approach for embedding algebraic manipulations, formulas and graphs in web pages, Global Engineering Education Conference (EDUCON), 4-6 April 2011, Amman, pp.691-694 (2011).

[9] ホーム：
<http://legacy.adlnet.gov/Technologies/scorm/default.aspx>
[10] 賈雲鵬，永井孝幸，Muhammad Wannous，喜多敏博，中野裕司：Maxima とマッシュアップ技術による SCORM コンテンツ中での数式処理と可視化の試み，情報処理学会研究報告. コンピュータと教育研究会報告，Vol.2011, No.3, pp.1-6 (2011-10).
[11] 中村泰之，稲垣佑亮，中原敬広：MathDox を活用した

STACK への数式入力インターフェースの追加, PC カンファレンス論文集 (CD-ROM), pp.188–191 (2014).

- [12] ホーム : <https://moodle.org/>
- [13] Chris Sangwin: Computer Aided Assessment of Mathematics, Oxford University Press (2013).
- [14] ホーム : <http://mathdox.org/>
- [15] ホーム : <https://www.mathjax.org/>
- [16] JSON-RPC 2.0 Specification : <http://www.jsonrpc.org/specification>
- [17] Roger L. Costello, Building Web Services the REST Way : <http://www.xfront.com/REST-Web-Services.html>
- [18] MathJax MathML Support : <http://docs.mathjax.org/en/latest/mathml.html>
- [19] jsonic.jar - simple json encoder/decoder for java, ホーム : <http://jsonic.sourceforge.jp/>
- [20] mathml.lisp - MathML Conversion in Maxima ホーム : <http://www.lurklurk.org/maxima.html>
- [21] The TeX-AMS-MML_HTMLorMML configuration file : <http://docs.mathjax.org/en/latest/config-files.html#the-tex-ams-mml-htmlormml-configuration-file>