

HTMLによる数式表示

元吉文男

(独)産業技術総合研究所*

F.MOTOYOSHI

AIST

1 はじめに

Web ページで数学に関することを記述しようとする、数式を使用する機会が多くなるが、現状では、決め手となる手段が一般的になっていない。ここでは、html のタグだけを使用して数式を記述する試みを紹介する。

2 従来の手法

現在、多く Web ページで数式の表示に使用されている手法には、概ね以下のものがある。

簡単なタグによる方法

一応、html にも数式を表現するために、上付、下付というタグがあり、添字、べきという程度の数式ならば、記述することが可能である。しかし、分数、行列などという複雑な式になると、記述できない。

“pre” タグの利用

“pre” タグを使用して、入力したままのテキストとして記述する手法であり、簡単な絵を文字で記述するときなどにも使用される。数式の場合も、文字を二次元に配置して表現しようという手法であり、文字しか出力できない端末において数式を表現するのに使用された方法である。しかし、この手法では、読む際に見易いとはいえ、またフォントが限定される、編集の際に数式としての構造が失われるという問題がある。

latex2html による方法

latex2html は、latex の原稿から html ファイルを生成するプログラムであり、現在、数式を含んだページを表示するのに、多く使用されている。この手法は tex の出力イメージで数式を表示できるため、きれいな表現が可能である。ただ、数式を画像として表現しているために、ファイル数が多くなる、数式としての構造が失われる、html の表現機能が自由には使用できないなどの問題がある。

*f.motoyoshi@aist.go.jp

$A(x)$ と $B(x)$ は可換環 R 上の x に関する次の多項式とする。

$$A(x) = \sum_{i=0}^m a_i x^i, \quad a_i \in R \quad (0 \leq i \leq m)$$

$$B(x) = \sum_{i=0}^n b_i x^i, \quad b_i \in R \quad (0 \leq i \leq n)$$

このとき

$$\text{resultant}_x(A(x), B(x)) = \text{Sylvester}_x(A(x), B(x))$$

$$= \begin{array}{c} \left| \begin{array}{cccc} a_m & a_{m-1} & \cdots & a_0 \\ & a_m & a_{m-1} & \cdots & a_0 \\ & & \ddots & \ddots & \ddots \\ & & & a_m & a_{m-1} & \cdots & a_0 \\ b_n & b_{n-1} & \cdots & b_0 \\ & b_n & b_{n-1} & \cdots & b_0 \\ & & \ddots & \ddots & \ddots \\ & & & b_n & b_{n-1} & \cdots & b_0 \end{array} \right| \end{array} \begin{array}{l} \left. \vphantom{\begin{array}{c} a_m \\ a_m \\ \ddots \\ a_m \end{array}} \right\} n \\ \left. \vphantom{\begin{array}{c} b_n \\ b_n \\ \ddots \\ b_n \end{array}} \right\} m \end{array}$$

図 1: 数式表示例 (画面ハードコピー)

MathML による方法

xml である MathML は、数式を表現するのに最も適した方法であろうが(もともと、そのために設計された手法である)、現在では、MathML を処理できるブラウザが限定されており (Mozilla 4.2 以降)、汎用的に使用できるわけではない。さらに MathML では表現できない数式もあり、規格として決定されているために、機能を追加しようとする際の拡張性が弱いように思われる。

3 本稿での方法

本稿では、Web ページ上で数式を表示するに際して次のことを念頭に置いた。

- 新たに plugin などをインストールしなくとも表示が可能なこと。
- フォントもシステムに標準でインストールされているものを使用する。
- テキストとして記述し、1つのファイルで閉じていること。

以上のことを考慮して、html の table タグを利用して表示することにした。table タグは入れ子にすることが可能であり、TeX における box の代用として用いることができ、二次元への配置に利用できる。さらに、分数の分子分母間の線や行列の線などは、罫線として引くことで、それらしく見せることができる。

フォントは Times New Roman を使用すると、小文字の筆記体が数式表示に適しているようであり、多くのシステムでそのまま使用できることも合わせて、このフォントにした。

このようにして作成したページ例を図 1 に示す。

図 1 を表示するためにはソーステキストが膨大になるため、簡単な数式を例にとって、表示原理を説明する。図 2(a) として表示される数式を記述したソースが図 2(b) である。このソースにおいて、"class="以下の記述で文字のフォントや大きさなどを使い分けている (そのために CSS — Cascading Style Sheets と呼ばれるファイルを作成して、クラスと属性の対応を定義してあるが、これはスタイル記述の簡単化のためのマクロのようなものである)。

実際にスクリーンに表示されるときの構造を示したものが、図 2(c) である。数式全体を 1 つの table とするようしており (table はデフォルトでは前後の改行を伴うようになっているが、テキスト中に含まれる場合には inline という属性を持たせることができる)、添字や Σ 式のように縦方向に移動を伴うも

$$A(x) = \sum_{i=0}^m a_i x^i$$

(a) 表示画面

```
<table class=mathblock><tr>
<td><table class=mathinner><tr>
  <td><span class=mathi>A</span>(<span class=mathi>x</span>) = </td>
  <td class=sum><span class=mathi>m</span><br>
    <span style="font-size:144%;">
    &Sigma;</span><br>
    <span class=mathi>i</span>=0</td>
  <td><span class=mathi>a</span></td>
  <td class=suxscript>&thinsp;<br><span class=mathi>i</span></td>
  <td><span class=mathi>x</span></td>
  <td class=suxscript><span class=mathi>i</span><br>&thinsp;</td>
</tr></table>
</td>
```

(b) ソーステキスト



(c) 画面構造

図 2: ソーステキストと表示の関係

のは1つのセルとして表現されている。

この例にはないが分数を表示するときには、2行1列の table として記述し、分母分子間の線はその部分だけ罫線を表示させることによって表現している。行列式するときには、行列の要素(それが入れ子になった table になることもある)がそれぞれ table の各セルとして対応させ、左右の罫線のみを表示すればよい。行列を表示するには、左右端に空で小さな幅のセルを加え、対応する側(左端ならば左側)の罫線に加えて、上下の罫線を表示することによって、行列らしく見せることができる。

4 table 形式の自動生成

前節の図2からも判るように、簡単な数式でさえ、ソーステキストが多くの行に渡るようになるために、少し複雑な式になると膨大な量になってしまう。それを、構文的に正しく記述するのは手作業では非常に困難であるので、何らかの形で計算機によってソースを生成しないことには、実用にならない。

そこで、数式処理システムに入力するような記述から table タグによる表現を生成するプログラムを作成してみた。本時点では、処理できる数式の構造として「べき」と「添字」のみが処理可能であるが、これを拡張することによって多くの構造に対応できる。その原理は、TeX と同様にボックスをレイアウトするという手法であり、表現したい数式の構造に対して、ボックスの配置に変換する機能を作成することによって、処理できる構造の種類を拡張することができる設計となっている。なお、ボックスは html では table として表現されているために、現状では、TeX のように自由にベースラインを設定することがうまくできていない。

自動変換の元になる数式の記述はとりあえずは macsyma 方式を採用しているが、構文解析部分は自由に文法を定義できるようになっているために、ダイナミックに文法を変更しながら、数式を記述することも可能である。

5 問題点

ブラウザ OS, CPU	表示画面
Mozilla 5.0 Linux, P4	$A(x) = \sum_{i=0}^m a_i x^i \quad \frac{x^2+1}{x^4+x^2+1} \quad \sqrt{x+3} \quad d_m^i$
Mozilla 5.0 MacOS X, PPC	$A(x) = \sum_{i=0}^m a_i x^i \quad \frac{x^2+1}{x^4+x^2+1} \quad \sqrt{x+3} \quad d_m^i$
Netscape 7.0 WindowsXP, P4	$A(x) = \sum_{i=0}^m a_i x^i \quad \frac{x^2+1}{x^4+x^2+1} \quad \sqrt{x+3} \quad d_m^i$
Internet Explorer 6.0 WindowsXP, P4	$A(x) = \sum_{i=0}^m a_i x^i \quad \frac{x^2+1}{x^4+x^2+1} \quad \sqrt{x+3} \quad d_m^i$
Opera 7.2 Windows2000, P4	$A(x) = \sum_{i=0}^m a_i x^i \quad \frac{x^2+1}{x^4+x^2+1} \quad \sqrt{x+3} \quad d_m^i$


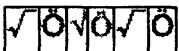
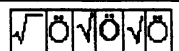
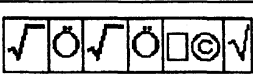
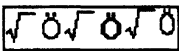
表 1: ブラウザによる表示の違い

前節でのようにして数式を表示できるが、同じソースを様々なシステムやブラウザで表示してみると、表 1 に示すような違いがあった。

最上位の行がターゲットにしたシステムであり、そこでの表示が最も適切になるようにソースを記述してある。他のシステムでもそれなりに表現されているが、部分式が中央に揃えられていないものや、根号の表示がおかしなものがある。これは、html の規格 (本稿での手法は html 4.01 を想定している) の解釈が異なっているためであると思われ、記述法を試行錯誤してみたが、その差を吸収することができなかった。

ただし、根号の表記については切実な問題であるので、根号文字に関して、使用フォントと文字の指定方法と変えて別に調査を行なった (表 2)。

文字の指定方法には”√” というように名前で指定する方法と”Ö” というようにコードで指定する方法がある。フォントとしてはブラウザのデフォルト、ここで数式用に使用した Times New Roman、browser が数式用に持っている symbol の 3 種類を組み合わせしてみた。それによると、ここでの使用で望ましいと考えている上部に線のない形 (上部の線はテーブルの罫線で表現している) が、ブラウザによって異なった指定方法でしか表示されていないことがわかる (いずれの方法でも表示できないブラウザもある)。

Mozilla 5.0 Linux	
Mozilla 5.0 Mac	
Netscape 7.0 WinXP	
MSIE 6.0 WinXP	
Opera 7.2 Win2000	

右列の説明

フォント、文字の指定 (左から)

1. 既定フォント ”√”
2. 既定フォント ”Ö”
3. Times New Roman ”√”
4. Times New Roman ”Ö”
5. symbol ”√”
6. symbol ”Ö”

表 2: 根号の表示

また、これが一番の問題であるが、そもそも html は文書の構造を記述するための言語であり、それをレイアウトに使用するの (しかも table という表の記述法を用いて) マークアップ言語としての用法を逸脱しているとも考えられることである。なお、MathML では、content 型と presentation 型と 2 種類の用法を使い分けられているが、そのために、処理系の負担が増大している (同じ数式構造であっても表示形が異なるのも有り得る)。しかし、実際に数式を Web 上で表示する機会は多くあり、そのための記述法が必要になるのは確かである。

6 まとめ

以上のように、多くのシステムに標準で備わってるブラウザを前提に Web 上でソースファイルをテキスト形式で記述した数式を表示することが可能であることが示せたが、処理系による違いを完全に吸収するまでには至っていない。

また、数式を記述するために規格化されたマークアップ言語である MathML が多くのブラウザでサポートされるようになったときの本稿の手法の優位性を考えると、ボックスのレイアウトという原始的な手法を採っているために、新しい形態の数式にも対応できるということであるが、互換性、汎用性のことを考慮すると、少なくとも、MathML 表現との相互変換を行なう機能は必須であると思われる。

参 考 文 献

- [1] B.Bos etc. (ed.): Cascading Style Sheets, level 2 — CSS Specification, W3C, 1998, (<http://www.w3.org/TR/CSS2/>).
- [2] D.Carlisle, etc. (ed.): Mathematical Markup Language (MathML) Ver. 2.0 (2nd ed.), W3C, 2003, (<http://www.w3.org/TR/2003/REC-MathML2-20031021/>).
- [3] D.Ragget (ed.): HTML 4.01 Specification, W3C, 1999, (<http://www.w3.org/TR/1999/REC-html401-19991224/>).
- [4] 水津弘幸, 石井歩, C&R 研究所: HTML+CSS Handbook, ソフトバンクパブリッシング (株), 東京, 2003.