

# MathML 編集環境における近似代数の利用について

片岡 正彰

嶋津 央礼

甲斐 博

MASAOKI KATAOKA \*

HIRONORI SHIMAZU †

HIROSHI KAI ‡

愛媛大学大学院理工学研究科

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING, EHIME UNIVERSITY

宮本 敦史

ATSUSHI MIYAMOTO §

株式会社ジャストシステム

JUSTSYSTEMS CORPORATION

## 1 まえがき

近年, Web 上の電子化された技術文書や e-Learning システムの普及に伴い, 文字や図だけでなく, 数式のような様々なボキャブラリを有効に活用できる文書編集システムが開発されている. 特に, 科学技術文書では数学情報を取り扱う機会が多く, 電子文書中の数式に対して検索や再計算を行うことができれば便利である.

数式の再計算が可能な文書編集システムとしては,  $\text{T}_{\text{E}}\text{X}$ macs[3] や WIRIS[4] などがあげられる.  $\text{T}_{\text{E}}\text{X}$ macs はテキストや数式などの構造化された文書を編集するシステムであるが, 数式処理のフロントエンドとしても利用できる. 計算された数式はシームレスに文書編集環境の中で利用でき, 記述した文書は  $\text{T}_{\text{E}}\text{X}$  や HTML/MathML で保存できる. 一方, WIRIS は数学教育での利用を目的としたソフトウェア群である. 数式編集用の WIRIS Editor や数式処理用の WIRIS CAS などを利用して, e-Learning システム Moodle 中の HTML 文書を編集できる. 数学情報は MathML を含む属性付きの画像として文書中に挿入されており, 付加された属性を利用して再計算が行われる. このように様々な取り組みがされているが, 我々はこの問題に対して複数の XML ボキャブラリを組み合わせた文書 (複合 XML 文書) をベースとした編集システムを開発することを検討している. ここで複合 XML 文書を利用する利点は,

- 様々なボキャブラリを含む文書構造を自然に表現できる
- 数式表現として MathML や OpenMath といった標準的な XML を利用できる
- 今後増えていくと考えられる XML アプリケーションに柔軟に対応できる

---

\*kataoka@hpc.cs.ehime-u.ac.jp

†shimazu@hpc.cs.ehime-u.ac.jp

‡kai@cs.ehime-u.ac.jp

§atsushi.miyamoto@justsystems.com

という点である。

xfy[1] は複合 XML 文書を編集することができるソフトウェアの一つでありジャストシステム社により開発された。文書には XHTML を用い、数式には MathML を用いることで、XML を利用した数学文書が作成可能となる。本研究では、再計算が可能な数学文書編集システムを xfy を用いて開発する。

MathML には意味上の構造を保持する Content Markup と表記上の構造を保持する Presentation Markup の 2 つが定義されている。数式の再計算を行うには数学的意味構造を保持できる Content Markup を用いるのが適している。本研究では xfy 上の Content Markup 編集エディタ [6] と数式処理システムとの連携を行う。これにより WIRIS と同様のコンセプトを持ちながらあらゆる情報に柔軟に対応可能な文書編集システムが実現できる。本研究では幅広い数学情報への対応を目指して、近似代数計算の利用を検討した。

## 2 xfy

xfy は複合 XML 文書の編集を目的としてジャストシステム社で設計・開発された統合 XML アプリケーション開発/実行プラットフォームである。xfy の特徴として、XML 文書の作成・編集を WYSIWYG で行えることが挙げられる。また、xfy コンポーネントによる機能拡張を可能にするプラグgable・アーキテクチャを採用している。

xfy コンポーネントとは、XML 文書の編集、XML 部分木に対する加工や計算、ファイル入出力等の機能をそれぞれ個別に実装したものである。xfy コンポーネントは主に Java により開発され、それらを xfy 上で利用するために JAR 形式でアーカイブされる。xfy には予め基本的な機能を提供する xfy コンポーネントが用意されており、XML 文書の閲覧・編集ができる。また、新規に xfy コンポーネントを開発することで、既存の xfy コンポーネントと組み合わせて利用することも可能である。

xfy における複合 XML 文書に対する処理の流れを図 1 に示す。この例では XHTML と SVG が複合 XML 文書に含まれている。

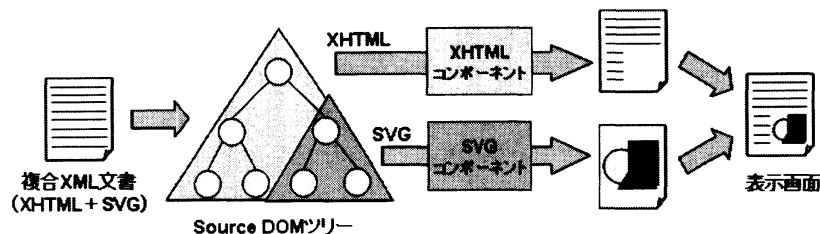


図 1: xfy による複合 XML 文書の処理

**step1** 複合 XML 文書から DOM ツリー (Source DOM ツリー) を生成する

**step2** それぞれの名前空間をもとに DOM ツリーを区画化し、各ボキャブラリコンポーネントが対応部分をそれぞれ処理する

**step3** step2. で得られた処理結果を統合し、複合 XML 文書として表示する

xfy では複合 XML 文書のシームレスな処理を目的としているため、文書構造に関する多くの情報を取得し、利用する機会が多い。必要に応じて要素の並び替えや、文書内の任意位置への要素の移動・追加・削除を行う必要があるため DOM を利用している。

### 3 Content Markup エディタへの数式処理機能の実装

Content Markup エディタ [6] は xfy 上で Content Markup の WYSIWYG 編集を可能にする xfy コンポーネントである。Content Markup を編集することで、xfy 上で正確な数学的意味を保持した数式を生成することが可能である。

Content Markup エディタでは一般的な数式編集エディタでもよく利用されているテンプレートによる入力を行う。テンプレート入力は一定の構造を維持したまま入力することが可能であるため、常に正しい数学的意味構造を保持しておく必要がある Content Markup の編集に適している。図 2 及び図 3 にテンプレートによる数式入力の例を示す。

出力方式には二次元表記を用いる。二次元表記は、一般的に人間が読み書きを行う際の数式の形であるため、直感的に理解しやすいのが特徴である。二次元表記の実現には既存エディタでもよく用いられているテキストとボックスの組み合わせを利用する。

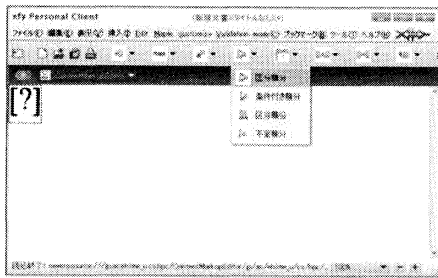


図 2: テンプレートの選択

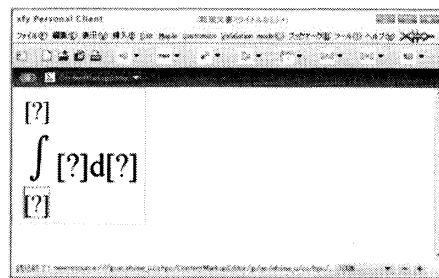


図 3: 入力されたテンプレート

Maple や Mathematica 等の数式処理システムは入出力として MathML が利用できる。本研究では数式処理システムと Content Markup エディタの連携について検討する。これにより、数式処理結果を容易にエディタへ反映できるようになり、数式編集におけるユーザビリティと利便性の向上が期待できる。

本研究では数式処理システム Maple との連携を検討した。Maple を用いた理由としては、Java から OpenMaple の API を利用して計算エンジン呼び出せることである。MathML への変換は Maple 中のパッケージを用いる。図 4 にシステムの概要を示す。

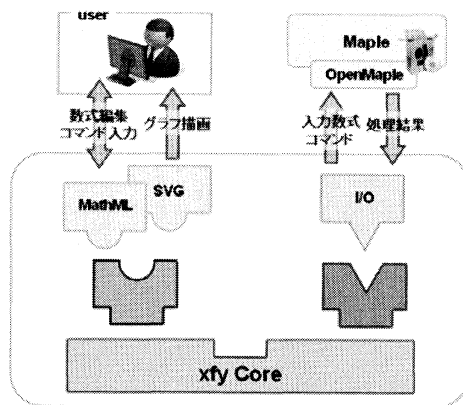


図 4: システム概要

Content Markup エディタの数式処理機能では、現在、展開や積分などの数式処理を実行することが可能であるが、MathML で表現された数式が Maple の関数の引数として意味のあるものであり、戻り値が MathML で表現できるものであれば、どのようなコマンドでも原理的には実装できる。また、plot 等のコマンドでは関数のグラフ描画が可能であるが、xfy で描画結果を利用できるようにするため SVG 形式で描画結果を得るようにしている。plot で生成された PostScript 形式のグラフを pstoeedit を用いて SVG 形式とすることで xfy 上での描画を可能としている。グラフの表示には xfy 既存の SVG コンポーネントが使用される。

数式処理機能の実行は、始めに OpenMaple を起動する。数式処理コマンドを適用する部分式を選択し、コンテキストメニューを用いて部分式に適用する数式処理コマンドを選択する。実際の実行画面を図 5、図 6 に示す。

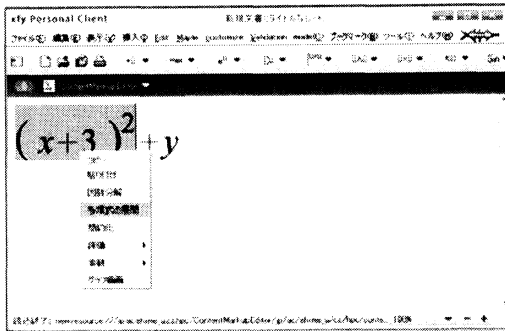


図 5: 部分式の選択

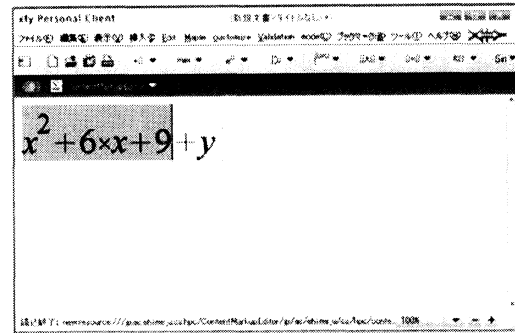


図 6: 実行結果

## 4 ユーザビリティ評価

本研究では数式処理機能を用いた場合の入力コストに対する変化を調べるため、数学文書作成時の入力コストについて評価実験を行った。評価の方法として図7のような問題文、問題の数式、解答の数式を含む数学文書を3名のテストユーザに作成してもらい、文書完成までにかかった入力コストを計測する。ここで入力コストとは、

$$\text{入力コスト} = \text{タイプ数} + \text{クリック数} + \text{動線距離} \quad (1)$$

で定義し、キーボードおよびマウスの入力が必要なコストの総和を検討した。ここでタイプ数、クリック数、動線距離は、それぞれ操作開始から終了までにかかるテストユーザの平均値である。動線距離の単位はセンチメートルである。比較対象には、従来の数学文書編集環境として Word と Maple を併用する場合、WIRIS[4]を用いる場合を取り上げる。テストの課題としては9つの課題を用意した。課題1を図7に示す。課題2から課題9までは図7と基本的には同じ構成であるが処理内容(図7では「積分」というテキストを指す)と問題の数式と解答の数式が異なる。表1に各課題の処理内容、問題の数式、解答をそれぞれ示す。

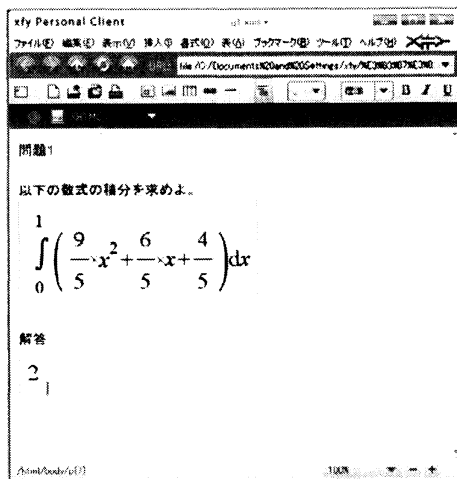


図 7: 課題1の文書

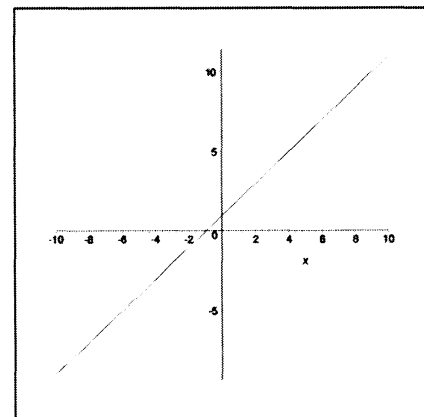


図 8: 課題9の解答

表 1: 評価に用いた課題

	処理	問題	解答
課題 1	積分	$\int_0^1 (\frac{9}{5}x^2 + \frac{6}{5}x + \frac{4}{5}) dx$	2
課題 2	展開	$(x^3 + 2x^2 - 6x - 3)^4$	$81 + 648x + 1188x^3 + 1728x^2$ $+ 822x^6 - 1944x^5 - 1728x^4 - 124x^9$ $- 128x^8 + 744x^7 + x^{12} + 8x^{11}$
課題 3	積分	(2) 式	(3) 式
課題 4	簡単化	$(x^2 + 3x)(y^3 - 4y^2)$	$x(x + 3) * y^2(y - 4)$
課題 5	総和	$\sum_{i=1}^n (2i + 4)$	$(n + 1)^2 + 3n - 1$
課題 6	極限	$\lim_{n \rightarrow \infty} (n^2 - n)$	$\infty$
課題 7	積分	$\int (x^3 * \sin x) dx$	$-x^3 \cos x + 3x^2 \sin x - 6 \sin x + 6x \cos x$
課題 8	和集合	$\{1, 2, 3, 4, 5\} \cup \{2, 4, 6, 8, 10\}$	$\{1, 2, 3, 4, 5, 6, 8, 10\}$
課題 9	グラフ	$x + 1$	図 8

表 2 に各課題に対する入力コストの計測値を示す.  $c$  はクリック数,  $t$  はタイプ数,  $d$  は動線距離,  $cost$  は (1) 式の入力コストを表す. ただし, クリック数, タイプ数, 動線距離は, (1) 式と同様に平均値であり, 少数点第 1 位で四捨五入した値である. また, 課題 3 と課題 5 は WIRIS では解答の数式を求めることができないため “\*” で表記している. 課題 3 は次節で述べるハイブリッド計算を用いて Maple 上で計算できる.

表 2: 入力コストの計測結果

	xfy				Word/Maple				WIRIS			
	c	t	d	cost	c	t	d	cost	c	t	d	cost
課題 1	142	26	105	273	208	30	132	370	216	48	263	527
課題 2	129	23	113	266	258	55	279	592	200	23	156	379
課題 3	279	26	140	445	531	71	526	1128	*	*	*	*
課題 4	156	15	66	237	191	27	177	395	196	16	125	337
課題 5	95	15	73	183	149	19	143	311	*	*	*	*
課題 6	104	22	163	289	141	27	195	363	107	56	222	384
課題 7	106	27	116	249	185	25	152	362	142	39	183	364
課題 8	109	40	174	323	156	23	150	329	112	20	139	270
課題 9	93	25	105	224	139	40	296	475	138	44	388	570

表2において、タイプ数、クリック数、動線距離の操作別に見た場合、ほとんどの課題において我々の実装は値が小さくなっているが、課題8におけるクリック数と動線距離の値は我々の方が大きくなった。集合の和を示す演算子“ $\cup$ ”の入力について、Content Markup エディタではアイコン選択後、ダイアログから演算子の数を指定する。他の方法では、アイコン選択後に演算子一つが入力される方法を採用している。我々のエディタではダイアログ操作を行う分だけクリック数と動線距離が増加したと考えられる。この点に関しては今後改善していく必要がある。

また、表2の入力コストについて検定を行った。検定にはスチューデントのt検定(片側検定)を用いた。その結果、Word/Mapleを用いた場合と比べて、Content Markup エディタを用いて数学文書を作成した場合の方が入力コストが小さいという結果が得られた。特に、課題2や課題3、課題5、課題7において入力コストが大幅に削減できている。これらの課題は他の課題と比べて解答の数式が長く、複雑なものとなっている。つまり、Content Markup Editorを用いた場合、複雑な数式を編集する際に、特に大きなコスト削減効果があるということが分かる。但し、前述のように個別のタイプ数、クリック数、動線距離について同様にt検定を行うと、課題8については我々の実装がコストが大きいという結果が得られる。

WIRIS との比較では、入力コストの計測値では全体的にWIRISよりも小さい値が得られているが、検定ではContent Markup エディタとWIRISの間に有意な差は見られなかった。従って、数式処理システムとの連携という同一コンセプトのシステムにおけるユーザビリティについて、本システムとの間に大きな差は無いと考えられる。

しかし、ユーザビリティ以外の観点から見ると、Content Markup エディタはx<sub>fy</sub>上で実装されているため、文書を複合XML文書として取り扱うことで、WIRISでは対応できない化学式などの様々な情報や新たに定義された新しい情報に対しても柔軟に対応できるという利点がある。

## 5 近似代数計算の利用

前節の問題についてWIRISではいくつか計算できない課題があった。幅広い数学情報への対応を考えて、未対応の関数やユーザ定義関数も利用できることが望ましい。ここでは数式処理機能の拡張の例としてハイブリッド積分[5]の計算の実現を行う。

前述の数式処理機能を用いて、以下のような不定積分のグラフを求めることを考える。

$$\int \frac{1}{x^5 - x^4 - 0.75x^3 + x^2 - 0.25x - 10^{-6}} dx \quad (2)$$

(2) 式をMapleのintコマンドで計算すると以下ようになる。

$$\begin{aligned} & -3.9999 \ln(x + 0.000004) + 0.2222 \ln(x + 1.0000) \\ & + (0.88885 - 816.48i) \ln(x - 0.5 + 0.0016330i) \\ & + (0.88885 + 816.48i) \ln(x - 0.5 - 0.0016330i) + 2.0000 \ln(x - 1.0000) \end{aligned}$$

ここで得られた数式は複素係数を持つ。しかし、ハイブリッド積分を用いれば、実係数の近似により不定積分を計算し、以下のような数式が得られる。

$$\begin{aligned} & 1.99996 \ln|x - 1.00000| - 1632.96 \arctan(612.376x - 306.189) \\ & + 0.888635 \ln|x^2 - x + 0.250004| - 3.99987 \ln|x + 0.00004| + 0.222223 \ln|x + 1| \end{aligned} \quad (3)$$

この数式では複素係数を含まないため、Mapleのplotコマンドでグラフ描画を行うことが可能となる。これをContent Markup エディタ上で利用できるようにすることで、文書編集等にグラフを取り込むことができる。

ハイブリッド積分はユーザ定義関数で実装されており、そのコマンドの実行には

- ユーザ定義関数の登録
- 計算に必要なパラメータの数などの登録

が必要である。現在はこれらをプラグイン開発者が登録し必要なパラメータはダイアログなどを使って入力することができる。今後、関数の登録を自動化する手法を検討する必要がある。現在登録しているハイブリッド積分の実行方法は通常の数式処理機能のコマンドを実行する際とほぼ同じである。図 9, 図 10 に実行画面を, 図 11 にグラフ描画の結果を示す。

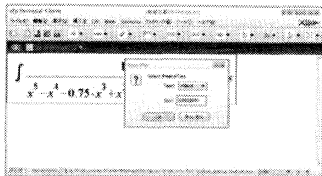


図 9: ダイアログへの入力

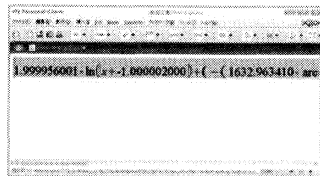


図 10: 実行結果

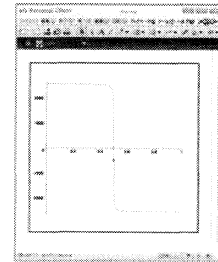


図 11: グラフ描画の結果

## 6 おわりに

本研究では、電子文書等における様々な数学情報をより有効に活用するため、数学情報を再計算可能な文書編集ソフトウェアの検討・開発を行った。また複合 XML 文書をベースにしているため数学情報以外の情報にも柔軟に対応できることが我々の手法の利点である。具体的には以下のことを行った。

- xfy 上の Content Markup エディタと数式処理システム Maple との連携
- ハイブリッド積分を実行するユーザ定義関数の利用

しかし、未対応関数やユーザ定義関数の自動登録機能など今度検討していく部分は多く残っている。また、Maple 以外の数式処理システムへの対応も今後の課題である。

## 参 考 文 献

- [1] xfy, <https://www.xfytec.com/>
- [2] MathML, <http://www.w3.org/TR/MathML2/>
- [3] A.G.Grozin, TeXmacs interfaces to Maxima, MuPAD and REDUCE, Proceedings of International Workshop on Computer Algebra and Its Application to Physics, pp.149-156, 2001.
- [4] Daniel Marques, Ramon Eixarch, Gloria Casanellas, Bruno Martinez, Tim Smith, WIRIS OM Tools a Semantic Formula Editor, Proceedings of MathUI, 2006.
- [5] Matu-Tarow Noda, Ei-ichi Miyahiro, On the Symbolic/Numerical Hybrid Integration, Proceedings of the International Symposium on Symbolic and Algebraic Computation, p.304, 1990.
- [6] 河田貴幸, 片岡正彰, 甲斐博, 田村恭士, xfy 上の MathML content markup エディタの実装と評価, 京都大学数理解析研究所講究録 1666, pp.164-171, 2009.