

OpenXM 1.2.2 の概要

OpenXM 開発グループ

OPENXM COMMITTEES

OpenXM プロジェクトでは数学での並列計算, 数学ソフトウェアの統合化 または Conglomerate 化 [13], 数学的知識のマネージメント (Mathematical Knowledge Management) などの諸問題を考察するとともに, 実際に数学の研究や数学の応用に使えるパッケージの開発をおこなっている.

パッケージは改訂を 4 回おこなった.

1. OpenXM 1.1.1 (January 24, 2000): 最初の実験版.
2. OpenXM 1.1.2 (March 20, 2000): とりあえず使える版.
3. OpenXM 1.1.3 (September 26, 2000): 1.1 系の最終版. OpenXM RFC 100 形式のプロセス木. 1077 個の数学関数を提供. 提供しているサーバは `ox_asir`, `ox_sm1`, `ox_phc`, `ox_gnuplot`, `ox_m2`, `ox_tigers`, `ox_math(ematica)`, `OMproxy` [6].
4. OpenXM 1.2.1 (March 2, 2002): Cygwin (Windows) への対応開始. マニュアル自動生成 (`gentexi`) など.

1.2 系列では新規サーバの導入や新プロトコルのおおがかりな導入は試みておらず, OX-RFC (OpenXM-RFC) 100, 101 プロトコルをより安定して動作させるための改良, さまざまなプラットフォームへの対応, 公式集プロジェクトなどにみられるように OpenXM の応用に重点をおいている. ここではちかじか予定している 1.2.2 系での新しい機能と設計における諸問題点を議論したい.

1 oxd スーパーサーバ

OpenXM では 1.2 系列より Cygwin を用いて Windows にも部分的に対応してきた. また 1.1 系列から OpenXM 対応の Java クラスを提供してきた. これらの開発で経験したことは, OX-RFC 100 による サーバの起動が煩雑 ということである. とくに C と unix の枠組をはずれると OpenXM サーバをクライアントが起動するのがとても面倒になる. さらにクライアントのコードは OS 毎に分岐させる必要が生じる. この煩雑さを避ける解決策としては以下のような方法があるであろう.

1. Corba などの Agent の仕組みを用いる.
2. ssh と `oxc` を用いる OpenXM RFC 101 (小原) を用いる.

“実際に使えるパッケージを配布する” という立場では, Corba 等の実験的インフラストラクチャを用いるのはなるべく避けたい. また効率重視のパッケージを作成したいときの自由度がない. OpenXM RFC 101 を用いるのは良い方法であるが, 実際にはメインのクライアントである `asir` が OpenXM RFC 100 にしか対応していないし, 100 対応のみのサーバも多い. したがって, RFC 100, 101 両方に対応しかつ, OpenXM サーバを容易に起動できるようなスーパーサーバの開発を試みた. これが `oxd` である (ソースは `OpenXM/src/kxx/oxd.c`).

サーバ側のログ:

```
bash-2.03$ cd OpenXM/src/kxx
bash-2.03$ ./oxd
Hello from open. serverName is localhost
and portnumber is 8089
Done the initialization. port =8089
デフォルト ポート 8089 で待つ.
Waiting a connection
```

クライアント側のログ:

```
bash-2.03$ cd OpenXM/src/OpenMath
bash-2.03$ java testclienttr
たとえば java で接続してみる.
```

oxd との通信コマンドは以下のように XML 形式である。トップノードは `<login>` ~ `</login>` である。

サーバ側:

```
... Serial=0
Trying to accept from localhost... len= 16
Accepted.

Connected. 8089 にクライアントが繋がった.
Waiting a connection... Serial=1
Trying to accept from localhost...
Serial=1: command=<login method="file">
クライアントより送られて来たコマンドは <login>
Serial=1: command=<done/>

こちらより送付した challenge を正しく処理
できたので、認証が完了.
Serial=1: command=<launch> ox -ox ox_asir
-reverse -control 3167
-data 3168 -finish </launch>
クライアントより ox サーバの起動コマンドを
受け取る。エンジン名はフルパスでは与えない.

Serial=1 : Executing command
-/home/taka/OpenXM/bin/openxm ox
-ox ox_asir -reverse -control 3167
-data 3168 -finish
The server ox_asir was not found.
Trying to find it under OpenXM/bin
Starting OX server :
/home/taka/OpenXM/bin/ox_asir

resultCode=0
Serial=1: command=<login/>
oxd への login は終了.

Waiting the termination of
the child process (ox server).
1: The child process is terminated.
```

クライアント側:

```
<login method="file">
認証方法は file を選択.

<challenge file="/home/taka/.oxd10793"/>
oxd より ファイル /home/taka/.oxd10793
に touch できたら認証すると連絡をうける.
?
<done/> touch したので <done/> を送る.

Listening...
<launch> ox -ox ox_asir -reverse -control 3167
-data 3168 -finish </launch>
oxd ~ OpenXM サーバの起動コマンドを送る.

<succeeded/> oxd より起動成功のメッセージ
をもらう.
?
```

Java 側の接続プログラムは簡単で、localhost の oxd で起動するなら

```
asir = new OpenXM("ox_asir");
```

と OX サーバ名を引数として OpenXM コンストラクタを呼べばよい。

エンジン側の出力:

```
controlByteOrder=0
OpenXM/Risa/Asir-Contrib(20020922),
Copyright 2000-2002, OpenXM.org
Loading "/.asirrc"
I'm an ox_asir. Version 20020802.
#0 Got OX_COMMAND SM_mathcap

以下略.
```

クライアント側のエンジンとの通信状況:

```
Accepted the control port.
Accepted the data port.
start
CMD> (CMD_MATHCAP, (CMD_LIST, 3, (CMD_LIST, 2,
(CMD_INT32, 199901160),
(CMD_STRING, 7, ox_asir)),
(CMD_LIST, 18, (CMD_INT32, 276)), 以下略.
```

このように oxd をもちいることにより、OS や 言語による起動法の違いを吸収可能である。さらに oxd の実装に agent の仕組みを組み込むことも可能である。

開発: 高山

2 いつでもどこでも “因数分解♡”

いつでもどこでも “因数分解♡” は、OpenXM のスローガンである。この章では、PDA や Web への対応を紹介する。

2.1 ザウルス 対応

シャープのザウルス SL-A300 で Risa/Asir が不完全ながら動作している。SL-A300 では Embedix という Linux を搭載している。GC と整数計算のためのマシン語プログラムの部分の移植が問題であったが、GC についてはすでに開発グループ (H-J. Boehm, A. J. Demers) が Xscale CPU に対応していた。整数計算についてはポータブル整数計算コード

OpenXM.contrib2/asir2000/asm/asmalpha.c を用いた。

開発: 藤本

2.2 OpenXM Online

OpenXM では HTTP プロトコルの部分集合 (GET および POST の一部) を実装し、Web ブラウザを OpenXM エンジンのフロントエンドとして利用したり、OpenXM エンジンに対する Remote Procedure Call をおこなうために利用している。この応用として、現在 Asir Online を <http://fe.math.kobe-u.ac.jp:8090> で実験的に運用している。HTTP を理解するサーバ (httpd) は、kan/sml のポストスクリプト言語で実装されている (OpenXM/src/kan96xx/Doc/httpd*.sml)。

この実験運用で問題となったのは、中断の伝播とリソースの制限の問題である。OpenXM RFC 100 ではエンジンの計算を中断するロバストなプロトコルを定めており、これが OpenXM RFC 100 の特徴の一つである [7], [10]。しかしながら、エンジン間で中断を伝播させる仕組みについては、標準をなにも決めていない。Web で計算をサービスするにはこれらの仕組みは不可欠であり、中断を伝播する仕組みとリソースの制限を実験的に実装した。

中断に関しては、中断したときに実行される関数を登録する仕組みが OpenXM サーバやクライアントにある。昔の Basic 風なら `On stop goto ...`, 最近の Java 風なら `try { ... } catch { ... }` みたいなものであるが、asir では `register_handler` 関数, sml では `ctrlC-hook` 関数で登録する。この関数の中で次の上意下達の処理をおこない中断を伝播させる。

上意下達の中断処理:

1. 自分の子供プロセスをすべてリストする (たとえば `ox_get_serverinfo()`)。
2. 子供プロセスに順番に OpenXM-RFC 100 の中断メッセージを送る。

上意下達法で中断はきちんと動作するが、この方法の欠点は中断が逐次的におこなわれるため、多数のサーバが含まれる木構造の場合は動作が遅いことである。今後必要に応じて中断メッセージを同報する方法を新しいプロトコルとして考察すべきであろう。

リソース制限に関しても類似の上意下達方式でリソース量を保持している大域変数の値を伝達することにより OpenXM-RFC 100 の枠内でコマンド毎のリソース制限が可能である。しかしながら shared memory の仕組みがあればより簡単であろう。

開発: 高山

3 OpenXM のフロントエンド

Infty Editor は九州大学の鈴木らが中心となり開発している手書き数式の入力機能ももつエディターである。このエディターから OpenXM サーバをよびだす機能が藤本により設計実装された。詳しい報告は [3].

開発: 藤本

4 OpenXM/Risa/Asir/Contrib

OpenXM/Risa/Asir (OpenXM 版 Risa/Asir) は OpenXM パッケージのメインのクライアントである。OpenXM/Risa/Asir/Contrib プロジェクトでは, asir 言語でかかれた各種数学関数, ユーティリティ関数および OpenXM サーバとのインタフェース関数を開発している。中期目標として“超幾何関数に関連する計算はなんでもできるように”を掲げている。超幾何関数に関連する計算は, 計算代数, 多面体幾何, 数値解析, ビジュアライゼーションなどいろいろな分野に関係している。

現在の関数一覧は以下のとおり。

base_cancel	matrix_list_to_matrix	print_tex_form
base_choose	matrix_matrix_to_list	print_tfb_form
base_flatten	matrix_rank	print_xdvi_form
base_intersection	matrix_solve_linear	print_xv_form
base_memberq	matrix_submatrix	sm1
base_permutation	matrix_transpose	sm1_ahg
base_position	number_abs	sm1_appell1
base_prune	number_ceiling	sm1_appell4
base_replace	number_floor	sm1_auto_reduce
base_set_minus	number_imaginary_part	sm1_bfunction
base_set_union	number_is_integer	sm1_deRham
base_subsetq	number_real_part	sm1_distraction
base_subsets_of_size	om_start	sm1_gb
dsolv_dual	om_xml	sm1_generalized_bfunction
dsolv_starting_term	om_xml_to_cmo	sm1_genericAnn
glib_line	ox_check_errors2	sm1_gkz
glib_open	ox_sm1_forAsir	sm1_hilbert
glib_plot	phc	sm1_mul
glib_print	phc_start	sm1_push_int0
glib_putpixel	plucker_relation	sm1_rank
glib_tops	poly_degree	sm1_reduction
glib_window	poly_elimination_ideal	sm1_restriction
gnuplot	poly_factor	sm1_saturation
gnuplot_heat	poly_gcd	sm1_slope
gnuplot_output	poly_grobner_basis	sm1_start
gnuplot_plot_dots	poly_hilbert_polynomial	sm1_syz
gnuplot_plot_function	poly_initial	sm1_wTensor0
gnuplot_start	poly_initial_coefficients	sm1_xml_tree_to_prefix_string
hilbert_polynomial	poly_initial_term	tigers
m_rtostr	poly_solve_linear	util_filter
m_start	print_dvi_form	util_find_and_replace
m_tree_to_string	print_em	util_find_substr
matrix_clone	print_gif_form	util_index
matrix_det	print_input_form	util_load_file_as_a_string
matrix_diagonal_matrix	print_open_math_tfb_form	util_part
matrix_eigenvalues	print_open_math_xml_form	util_read_file_as_a_string
matrix_identity_matrix	print_output	util_remove_cr
matrix_inner_product	print_ox_rfc100_xml_form	util_v
matrix_inverse	print_png_form	util_write_string_to_a_file

`matrix_kernel` `print_terminal_form`

各関数のマニュアルは

<http://www.math.kobe-u.ac.jp/OpenXM/1.2.2/doc/asir-contrib/html-en/cman-en-toc.html>

開発: 野呂, 中山, 高山

5 サーバの新しい数学関数

1.2 系ではまだ新しいサーバの接続作業をおこなっていない。1.3 系の機能になると思うが、多面体の幾何 (polymake, cdd), 3D グラフィックス関連のサーバ (3D Java, JavaView, OpenGL をもちいた自前のサーバ) の接続を予定している。

`ox_asir` には次の新しい数学的機能が加わった (野呂, 横山)。

1. 有限体での因数分解 (`OpenXM.contrib2/asir2000/engine/Fgfs.c`).
2. 有限体での準素イデアル分解

`ox_sm1` には次の新しい数学的機能が加わった (高山)。

1. Granger-Oaku の Tangent Cone アルゴリズムの実装のため解析的微分作用素環における同次化を実装 (`ring_of_differential_operators`, `weightHomogenization`, `OpenXM/src/kan96xx/Kan/poly3.c`).
2. M.J.Ucha の新しい b -関数アルゴリズムの実装のため, ある微分差分作用素環を実装 (`ring_of_differential_difference_operators`).

6 fb — 超幾何関数デジタル公式集と OpenMath

OpenXM ではコードネーム fb で超幾何関数のデジタル公式集を研究している。fb は 1.2 系の目玉といえるモジュールである。数学公式集の研究はたとえば日本では, 1980 年代後半の佐々木らの研究が先駆的である [12]。この研究では数式の検索法, 数式の公式集による書き換え, およびそれに適したデータ構造が研究された。さらに岩波公式集をもとに実際のデジタル公式集を作成したが, 著作権の問題があり, 公開されていない。この流れをくみ, さらに OpenMath の XML で公式を表現し検索する研究が森永, 村上, 野田 [8] により最近進展している。

OpenXM の fb プロジェクトでは, 上の流れとは異なり, 検索などの問題を考える前の段階の問題 “いかに数学公式を計算機上に蓄積するか?” という問題意識のもと, 次のような機能をもつデジタル公式集が欲しいという観点から設計実装をおこなっている。

1. さまざまな数学概念に柔軟に対応できる公式集。
2. 公式の数学的意味や参考文献などの情報を機械可読形式で保持できる公式集。
3. XML の機能をフルに生かした公式集。
4. 著作権の問題がない公式集。
5. 入力ミスの発見を補助する機能など。

以下にわれわれの設計と実装の概要を記す。

fbOM-DocBook document	DocBook タグ (Top), OpenMath タグ, MathBook タグ, formula タグ (独自)
fb-DocBook document	DocBook タグ (Top), tfb タグ (独自), MathBook タグ, formula タグ (独自),
fbOM-formula document	formula タグ (Top, 独自), OpenMath タグ, DocBook タグ (一部)
fb-formula document	formula タグ (Top, 独自), tfb タグ (独自), DocBook タグ (一部)

図 1: fb XML 文書のバリエーション

1. OpenMath (www.openmath.org) はさまざまな数学概念に柔軟に対応できる仕組みである。OpenMath を基礎に公式集を記述する。functions.wolfram.com では Mathematica 形式で格納された多数の数学公式が提供されているが、OpenMath を利用することにより、Mathematica の制約をうけない。
2. OpenMath オブジェクトを表現するための tfb という言語を開発した。OpenMath プロジェクトでは OpenMath オブジェクトを入力するための RIACA CD Editor(オランダ) や STAR/MathWriter (Stilo Technology) などのシステムが研究されている。たとえば STAR/MathWriter では semantic tex という OpenMath XML とはある意味でかけ離れた言語をもちいて数学オブジェクトを入力する。この tfb は OpenMath オブジェクトとの 1 対 1 対応がすぐわかる形式であり、OpenMath シンボルの階層構造の美しさがそのまま見える設計になっているのを特徴とする。
3. DocBook [2] および OpenMath [4] の XML タグを拡張し、formula を トップノードとする独自のタグをいくつか定義した。これらをまとめて fb-formula document タグセットとよんでいる。超幾何関数に関連する OpenMath Content Dictionary が存在しないため、自分達で CD hypergeo0, hypergeo1, hypergeo2, hypergeon0, hypergeon1, hypergeon2, weylalgebra1 を開発した (OpenMath CD の contrib を参照 [4])。これにより、公式の数学的意味や参考文献などの情報を機械可読形式で保持できるようになった。なお、A.Cohen らの Interactive Math Book のプロジェクトも DocBook と OpenMath のタグを拡張する形で研究している [1]。
4. OpenMath の XML と Content Dictionary はさまざまな数学概念に柔軟に対応できる仕組みである。たとえば前記の超幾何関数に関連する Content Dictionary をもとに高山は 1 次元の積分路を表現する Content Dictionary intpath1 を定義し、この Content Dictionary は任意の 1 次元の積分路を表現できることを twisted (co)homology の理論を用いて証明した。
5. fb では DOM や XSL など XML の機能をフルに生かし公式集を実装する。XML の機能がデジタル公式集にも有効であることを確認した。図 2 の各コマンドは主に Java で実装されている。
6. fb では 19 世紀、20 世紀の超幾何関数のオリジナル論文をもとに tfb という言語をもちいて入力をおこなっており、著作権の問題がない。
7. 公式の入力にはミスがつきものである。fb では、数値計算をもちいて明らかな誤りを検出する仕組みを提供している。この仕組みでは OpenXM 版の mathematica および asir サーバおよび前述の OpenXM online を用いている。この checker の実装のために退化接続公式を用いた新しい超幾何関数の数値計算アルゴリズムも考案された。

次は fb-formula 文書の例 (fb/formula/h21-0060.tfb)。

```
<?xml version="1.0" encoding="ISO-2022-JP"?>
<?xml-stylesheet type="text/xsl" href="fb.xsl"?>
<formula>
  <tfb macroset="http://www.openxm.org/fb/hfb.txt">
    2 * arith1.root(nums1.pi,2)
```

tfb2fb.sh	fb-formula 文書を fbOM-formula 文書へ変換
	fb-DocBook 文書を fbOM-DocBook 文書へ変換
tfb2ocd.sh	(互換性のため残してある)
fb2tex.sh	fbOM-DocBook 文書より LaTeX への変換
ocd2tex.sh	(互換性のため残してある)
formula2tex.sh	fbOM-formula XML 文書より LaTeX への変換
asir2tfb.sh	asir 形式から tfb 形式への変換
tfb2asir-check*.sh	fb-formula 文書 より asir checker へ.
tfb2math-check.sh	fb-formula XML 文書 より Mathematica checker へ.

図 2: fb の変換コマンド (一部)

```

* hypergeo0.gamma(a + b + (1 / 2))
/ hypergeo0.gamma(a + (1 / 2))
/ hypergeo0.gamma(b + (1 / 2))
* hypergeo1.hypergeometric2F1(a,b,1 / 2,x)
=
(hypergeo1.hypergeometric2F1(2 * a, 2 * b,
                             a + b + (1 / 2),
                             1 + arith1.root(x,2) / 2)
+
hypergeo1.hypergeometric2F1(2 * a, 2 * b,
                             a + b + (1 / 2),
                             1 - arith1.root(x,2) / 2));
</tfb>

<editor> Yasushi Tamura </editor>

<description>
  Quadratic transformation of independent variable
</description>

<reference linkend="goursat1" page="118"/>

<evidence checker="Mathematica">
  @@ /. {a->1/2,b->3/5,c->-2/11,x->0.2}
</evidence>

</formula>

```

この公式を TeX への変換ツールでプレゼンテーション形式に変換したものは

$$\frac{2\sqrt{\pi}\Gamma(\frac{1}{2}+a+b){}_2F_1(a,b,\frac{1}{2},x)}{\Gamma(\frac{1}{2}+a)\Gamma(\frac{1}{2}+b)} = {}_2F_1(2a,2b,\frac{1}{2}+a+b,\frac{1-\sqrt{x}}{2}) + {}_2F_1(2a,2b,\frac{1}{2}+a+b,\frac{1+\sqrt{x}}{2})$$

この公式は 1881 年の E.Goursat の論文に記載されている超幾何関数の独立変数の変換公式である。

開発チーフ: 田村

開発: 田村, 高山

公式編集メンバー: 田村, 高山, 中山, 藤堂, 長谷川

7 OpenXM 関連の出版物

OpenXM 1.2 以降に出版された資料のリスト:

1. [9] では Risa/Asir の内部構造の概要や Risa/Asir の OpenXM への対応状況が解説してある。
2. [11] は日本語による Risa/Asir プログラミングの入門書である。OpenXM を用いた簡単な分散計算への入門の章がある。
3. OpenXM の中断機能, 競争計算, などについての短いビデオ。

以上 OpenXM 1.2.2 の新機能を概説した。詳しくは <http://www.openxm.org> の CVS-web を用いてソースコードの変更履歴およびコミット時のコメントを参照されたい。

参 考 文 献

- [1] A.Cohen, H.Cuyppers, E.R.Barreiro, H. Sterk, Interactive Mathematical Documents on the Web, edited by M.Joswig and N.Takayama: *Algebra, Geometry and Software Systems*, Springer, 289–308, 2003 (to appear).
- [2] <http://www.docbook.org>
- [3] M.Fujimoto, M.Suzuki, A Handwriting Interface to Various Computer Algebra Systems via OpenXM Framework, Abstracts of 8th International Conference on Applications of Computer Algebra, 2002, Volos, Greece. 103–104.
- [4] <http://www.openmath.org>
- [5] <http://www.openxm.org>
- [6] 小原, 高山, 田村, 野呂, 前川, OpenXM 1.1.3 の概要, 数理研講究録 1199 (2001 年 4 月), 179–191.
- [7] M.Maekawa, M.Noro, N.Takayama, Y.Tamura, K.Ohara, The Design and Implementation of OpenXM-RFC 100 and 101. Computer Mathematics, Proceedings of the Fifth Asian Symposium (ASCM 2001), edited by Shirayanagi and Yokoyama, World Scientific, 102–111, 2001.
- [8] 森永, 村上, 数学公式データベースと G 関数, 数式処理 9 No.2 (2002), 20–31.
- [9] M.Noro Computer Algebra System: Risa/Asir, edited by M.Joswig and N.Takayama: *Algebra, Geometry and Software Systems*, Springer, 147–162, 2003 (to appear).
- [10] M.Noro, N.Takayama, Design and Implementation of OpenXM Client-Server Model and Common Mathematical Object Format (OpenXM-RFC 100). 1997, 2000.
<http://www.math.sci.kobe-u.ac.jp/OpenXM/OpenXM-RFC.html> or <http://www.openxm.org>
- [11] 野呂, 高山, Risa/Asir ドリル, <http://www.math.kobe-u.ac.jp/Asir>
- [12] T.Sasaki, Y.Masunaga, Y.Saigusa, A.Abe, F.Motoyoshi, M.Sasaki, Formula Database in Computer Algebra System GAL, 数理研講究録 663, 群と微分方程式の数式処理システムの研究 (1988), 3–22.
- [13] A.Solomon, Distributed Computing for Conglomerate Mathematical Systems, edited by M.Joswig and N.Takayama: *Algebra, Geometry and Software Systems*, Springer, 309–326, 2003 (to appear).