

## Flash で動く関数グラフィングツールの開発

筑波大学教育開発国際協力研究センター 讃岐 勝 (Masaru Sanuki)  
磯田 正美 (Masami Isoda)  
Center for Research on International Cooperation in  
Educational Development,  
University of Tsukuba  
株式会社ゼータ 渡辺 央 (Hiroshi Watanabe)  
ZETA Co. LTD

### 1 はじめに

本稿では、Flash で動くアプリケーションの開発について述べる。当センター<sup>1</sup>では、Action Script 2/3 で書かれた電子教科書作成ツール dbook[dbook] の開発を行っており、そのツールによって作成された電子教科書は Flash ファイル (.swf) として利用が可能であり、WEB 上にもそのまま公開可能な電子教科書が作成できる。その電子教科書上で動作可能な Flash アプリケーション GCL Editor[GCL Editor] 上で動く関数描画アプリケーションの開発について述べる (図 1 参照)。

GCL Editor は dbook 上で動作する動的幾何アプリケーションである。Cabri 2D/3D などの動的幾何アプリケーションは関数描画の機能を有していない。逆に Grapes などの関数描画アプリケーションは作図機能を持たない。GeoGebra はいずれの機能も持つが、他のアプリケーションと連携して動かすことができないため、現在のままでは電子教科書との連携はできない。そのため、電子教科書 dbook とともに開発を進めている GCL Editor 上に関数描画の機能を実装する。

Flash はコンピュータに負担を掛けるとか、初期の iPad では Flash が読み込めず、Java, JavaScript や HTML5 で動くアプリケーションの開発が主流になるかもしれない現状であるが (本講録に多数収録)、HTML5 から Flash への変換を Adobe の Creative Suite CS5 でサポートされたり Skyfire<sup>2</sup> などブラウザの中で変換するプラグインが開発されるなど、Flash の需要はあり Flash 自身なくなるわけではない現状である。ゆえに Flash Player で動くアプリケーションを作ることは無意味ではなく、ユーザが多く得られる。

本稿では、動的幾何ソフトの機能を持ちつつ、代数入力された関数  $F_{a,b,\dots}(x,y)$  の描画が可能なアプリケーションを開発する。ここで、 $a, b, \dots$  はパラメータであり随時変化させることができ、そのたびに関数が描かれ、ディスプレイには関数族が表示されることを目指す。また、動的幾何および関数描画の機能を持たせることによって次の作図問題から曲線の性質を明かす機会を与える (かかれる曲線はいずれもパスカルのリマソン)。実験や探求を基盤とする作業的・体験的な活動が必須であることを確信する機会を提供する教材の作成をする。

<sup>1</sup>筑波大学教育開発国際協力研究センター (CRICED)

<sup>2</sup><http://www.skyfire.com/>

1. 点  $O$  を通る半径  $r$  の円  $\gamma$  があり円周上を点  $A$  が動く. 直線  $OA$  上の 2 点  $P, Q$  は  $AP = AQ = a$  をみたす点のとき, 点  $P, Q$  のかく曲線
2. 極方程式  $r = b + a \cos \theta$  の描画

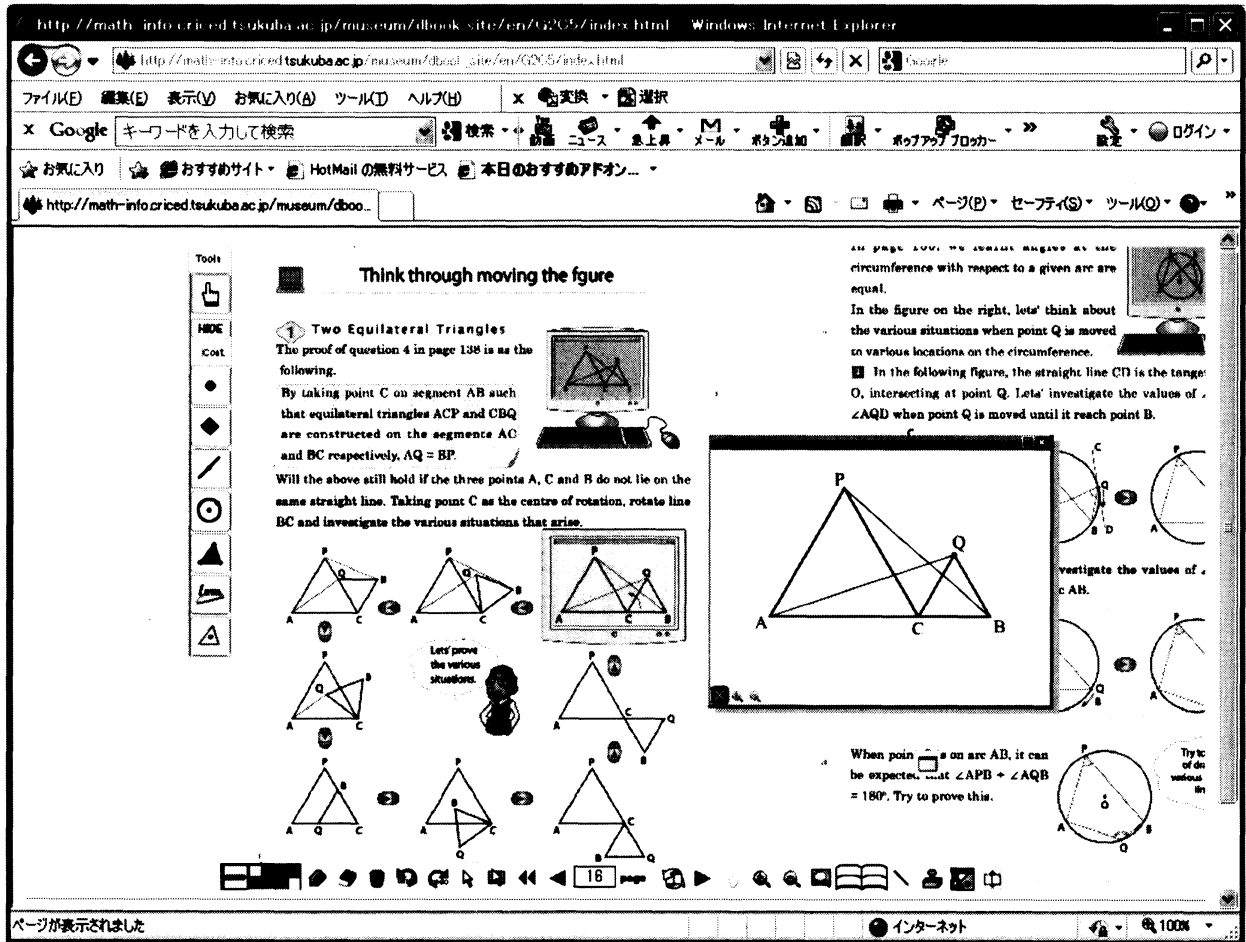


図 1: 電子教科書 dbook 上で動的幾何ソフト GCL Editor が動いている [math-info]

## 2 設計

Flash で動く関数グラフィングツールを次の設計により開発する. 関数描画に関するアプリケーションを作る場合, どうしても数式処理システム (数式の処理および多変数多項式系の根の計算ため) が必要となる. Mathematica や Maple では自前のシステムを利用して表示までを実装, Maxima では描画される点の計算までシステムで行い表示は Gnuplot に委託, Gnuplot 自身の関数描画は自前で実装が行われているが, 高度な関数の描画は Maxima を利用して描画した方が正確である. このような現状の元, 自身で描画ソフトを作る際には数式処理システムを, 1) 自身で開発する, 2) 既存のシステムを使用する, という二択に迫られる. 自身で開発する場合, 関数描画を行うのに必要なアルゴリズムの実装というのは, 既存する数式処理システムそのものを組みのと変わらないくらいの労力を必要とする. このような現状から, 数式を処理する部分については既存

のシステムを利用することにする。本稿では、Maximaを利用して関数描画アプリケーションを開発する。実際に描画点の計算のため、Maximaには次の順で処理をさせる。

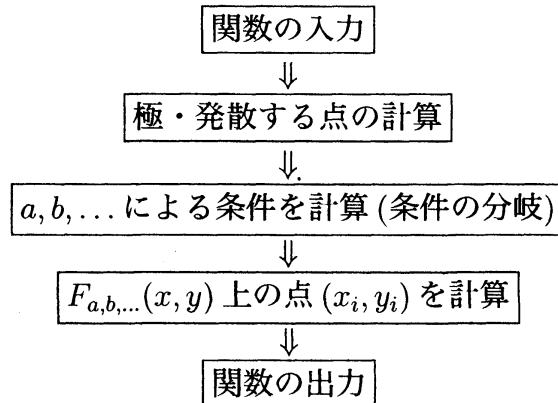


表 1: Maxima で計算する順序

Maxima の `plot` コマンドを使用すると  $F_{a,b,\dots}(x, y)$  上の点  $(x_i, y_i)$  を計算および描画点の連結する順序などが自ら実装することなく得ることができる。

**例 1 (条件の分岐)**  $F_{a,b,c}(x, y) = ax^2 + bx + c$  および  $F_a(x, y) = ax^2 + (a - 1)y^2 = 1$  を表 1 の手順で処理させると、条件の分岐は次のように書ける。

- $F_{a,b,c}(x, y) = ax^2 + bx + c$   
 $\Leftrightarrow$  主係数の係数の変化するとき
- $F_a(x, y) = ax^2 + (a - 1)y^2 = 1$   
 $\Leftrightarrow \begin{cases} \text{楕円} & a > 1 \\ \{x|x^2 - 1 = 0\} & a = 1 \\ \text{双曲線} & 0 < a < 1 \\ \{y|-y^2 = 1\} & a = 0 \end{cases}$

実際に Maxima で次を実行すると、Maxima の指定した場所に描画点の情報のあるファイル `maxout.gnuplot` が作成される。このファイルは Gnuplot で解釈できるように吐き出されているため、連結部分の情報が含まれていることがすぐにわかる。

```

(%i1) load(implicit_plot);
(%o1) /usr/share/maxima/5.16.3/share/contrib/implicit_plot.lisp
(%i2) implicit_plot(x^2+y^2 + x =2, [x, -2, 2], [y, -2, 2], [run_viewer, false],
      [plot_format, gnuplot_pipes]);
(%o2) done
  
```

ゆえに、開発者側でプログラミングするのは、零点/極の計算のみである。また、本来開発に当てるべき時間を現状の数式処理システムで描くことのできないような関数の描画プログラムにあてる [Sanuki11]。

```

maxout.gnuplot
set term x11 font "Helvetica,16"

set xrange [-2.0:2.0]
set yrange [-2.0:2.0]
set style data lines
plot '-' title 'y^2+x^2+x = 2' with lines 3
-1.928 -0.48
-1.92 -0.488

-1.944 -0.432
-1.936 -0.44

(以下, 省略)

```

次に、Maxima を関数描画アプリケーションの数式処理エンジンとして組み込む方法を解説する。Maxima は GPL2 というライセンスで利用・配付がされている。このため、関数描画アプリケーションの実行ファイルに組み込むとすべてのソースを公開する義務が発生し、GCL Editor のソースも公開する義務が発生する。すべてのソース情報を公開したくない今回の場合、データを何らかの方法 (PHP) で Maxima に送り、Maxima で計算をさせて、再度 GCL Editor にデータを戻し、関数の描画を行うように設計した。

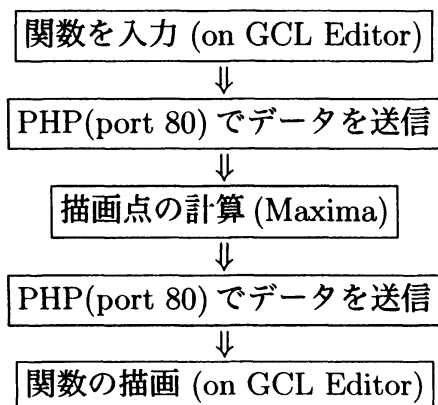


表 2: Maxima との連結

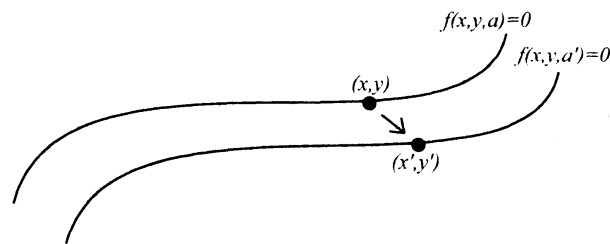
WEB 上で動くアプリケーションとして考えているので、通信には PHP を用いている。また、MaximaPHP<sup>3</sup> というアプリケーションがあるように WEB 上での Maxima の利用に関して多くのノウハウがあることが想像できる。MaximaPHP の内部をみると、より効率のよい方法で実装されていると思われるが、今回はシンプルに実装をした。PHP で

<sup>3</sup><http://maximaphp.sourceforge.net/>

データの転送をしているため、どのような関数の描画が行われたかを Maxima のインストールされている WEB サーバのログから知ることができる。

## 2.1 パラメータの変化

$F_{a_1, b_1, \dots}(x, y)$  の描画ができたとして、 $a_1 \rightarrow a_2 = a_1 + \varepsilon$  としたときの関数  $F_{a_2, b_1, \dots}(x, y)$  の描画については Maxima による再計算は行わない。表 1 でパラメータに関する条件を求めており、同じ関数族に含まれている関数であれば、関数の連続性からすでに与えられている関数から根を接続することによって、すぐに関数を描画することが可能である。根の計算には DKA 法などを使えばよく、これは ActionScript で実装する。



## 2.2 各アプリケーション間の接続

次の方法によって各アプリケーション間の接続ができる。

- FLASH  $\Leftrightarrow$  PHP  
PHP の動いているサーバの port 80 を解放していればよい。
- PHP  $\Leftrightarrow$  Maxima  
system 関数を用いて、PHP から maxima を実行させることができる。

## 3 サンプル

次は、2次曲線  $y = ax^2 + 1x + 1$  で  $a$  を動かしたときの様子である。直線上の点 P を動かすことで、曲線のパラメータを変化させている。

## 4 まとめと展望

Flash で動く関数描画アプリケーションの開発を数式処理システムを実装することのない方法によって行った。数式処理システムが必要な部分の計算について、PHP を利用して Maxima へ関数を受け渡す方法をとったが、状況によってはデータの転送に時間を要することが考えられる。そのため、1回のデータ転送の間に Maxima に多くの処理を行うように設計をしている。しかし、多くの処理を高速にさせる必要が出てきた場合には、使用上ライセンスに問題のない数式処理システムを組み込むことも考えなければいけない。

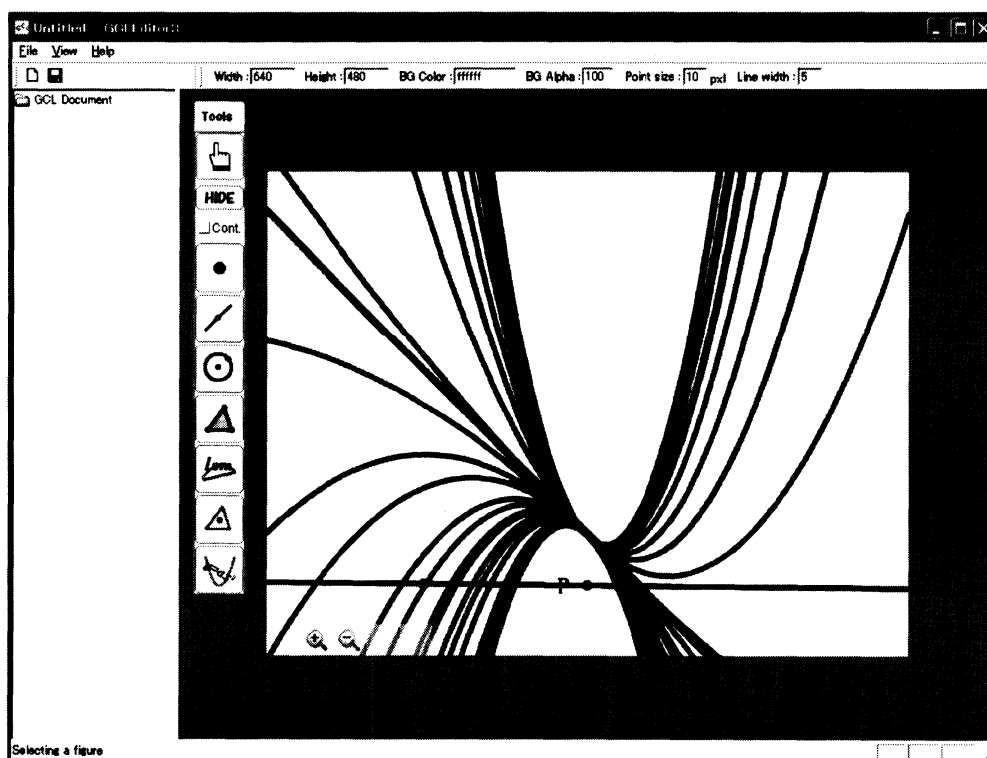


図 2:  $y = ax^2 + 1x + 1$

## 参考文献

### Softwares

[dbook] dbook :

[http://math-info.criced.tsukuba.ac.jp/software/dbook/dbook\\_eng/](http://math-info.criced.tsukuba.ac.jp/software/dbook/dbook_eng/)

[GCL Editor] GCL editor :

[http://math-info.criced.tsukuba.ac.jp/software/gcl/GCL\\_eng/](http://math-info.criced.tsukuba.ac.jp/software/gcl/GCL_eng/)

[Maxima] Maxima : <http://maxima.sourceforge.net/>

### References

[田端-讃岐-磯田 2009] 田端毅-讃岐勝-磯田正美. 曲線の事典 —性質・歴史・作図法—. 磯田正美 & Maria G. Bartolini Bussi 編, 共立出版, 2009.

[math-info] dbook の展示室. Exemplar of dbook :

[http://math-info.criced.tsukuba.ac.jp/museum/dbook\\_site/](http://math-info.criced.tsukuba.ac.jp/museum/dbook_site/)

[Sanuki11] M. Sanuki. *A study on implementation for drawing polar equation  $r = \sqrt[p]{f(\theta)}$* , Preprint of Univ. of Tsukuba, 8 page, 2010.