

KETpic の Risa/Asir への移植について

諒間電波高専・電子制御工学科 近藤 祐史 (Yuji Kondoh)

Department of Control Engineering,
Takuma National College of Technology

(株) アルファオメガ

齋藤 友克 (Tomokatsu Saito)

AlphaOmega Inc.

東邦大学・薬学部

高遠 節夫 (Setsuo Takato)

Faculty of Pharmaceutical Sciences,
Toho University

1 はじめに

筆者の一人は、印刷教材における作図を支援するための $\text{T}_\text{E}\text{X}$ マクロパッケージ KETpic を開発した [1]。KETpic は、当初 Maple や Mathematica といった商用数式処理ソフト上で開発されたため、高価なソフトウェアを購入しなければ利用できなかった。そこで、広く普及させるためフリーソフトへの移植が試みられている [2]。本稿では、KETpic を現在フリーソフトである Risa/Asir [3] へ移植することを検討する。特に、グラフ描画機能における移植の可能性について検討する。

2 Asir への移植

2.1 移植の方法

KETpic を Risa/Asir へ移植する場合、その方法として次のものが考えられる。

- 組み込み関数として実現
 - C 言語でのプログラミングすることにより、Asir 内部に組み込む。
- ユーザ言語 (Asir 言語) を用いて実現
 - Asir 言語のみでのプログラミングを行う。
- 不足する機能のみ組み込み関数を改良または追加し、ユーザ言語で実現
 - Asir 言語でのプログラミングを行うが、他の数式処理システムが標準的に搭載している関数で実現されていないものは、C 言語プログラミングにより組み込みを行う。

これらの方法の中、本稿では2番目のユーザ言語（Asir 言語）のみで実現することを検討する。

2.2 KETpic 移植の条件

KETpicを移植するにあたり、数式処理システムがもつべき条件として次が挙げられる。

- プロットデータの取得
- TeX コマンド・Tpic コードの書き出し
- パッケージ化
- 文字列処理
- 引数の柔軟性（関数, リスト, 文字列）
- リスト処理
- 数式処理

2.3 移植の条件への対応

- プロットデータの取得
 - `memory_plot()`, `memory_ifplot()`, `memory_conplot()` があるので, これらの関数を利用する.
- TeX コマンド・Tpic コードの書き出し
 - `print()`
 - 標準出力をファイルへ切替える `output()` で対応可能である.
- パッケージ化
 - プログラムのロード `load()` で対応できる.
- 文字列処理
 - 文字列を処理するための関数が用意されている.
- 引数の柔軟性（関数, リスト, 文字列）
 - ユーザ言語では, 可変長引数を書けないが, オプション指定という形式で対応可能である.

例 `drwline(PD);`
`drwline(PD| width=0.5);`

- また、引数全体をリスト表現にしてしまうことでも実現可能である。

例 `drwline(PD);`
`drwline([PD, 0.5]);`

- リスト処理

- `car()`, `cdr()`, `append()`, `cons()` などがあるので利用できる。

- 数式処理

- 数式処理機能はある。

2.4 `memory_plot()` や `memory_ifplot` の出力

プロットデータの取得で用いる `memory_plot()` や `memory_ifplot` からの出力例は次のようなものである。

```
[0] memory_plot(sin(x));
[300,300,100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 0f 00 e0 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00 06
.....
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00]
```

この出力は、

- 300 × 300 ドットの点データをバイナリーデータ化している。
- x 軸方向に 8 ドット分が 1 バイトとして表現されている。
- グラフの形のみの情報である。

という特徴がある。そのため、`memory_plot()` を利用する場合、

- 1 ビット毎にスキャンし、隣点を線分で結ぶ

必要がある。

また、`memory_ifplot()` を利用する場合、

- 点の位置データしかない。点と点の間を補う方法はない。
- アルゴリズム上、必要な精度でのデータを取得する必要がある。
- そのため、データ増になる。

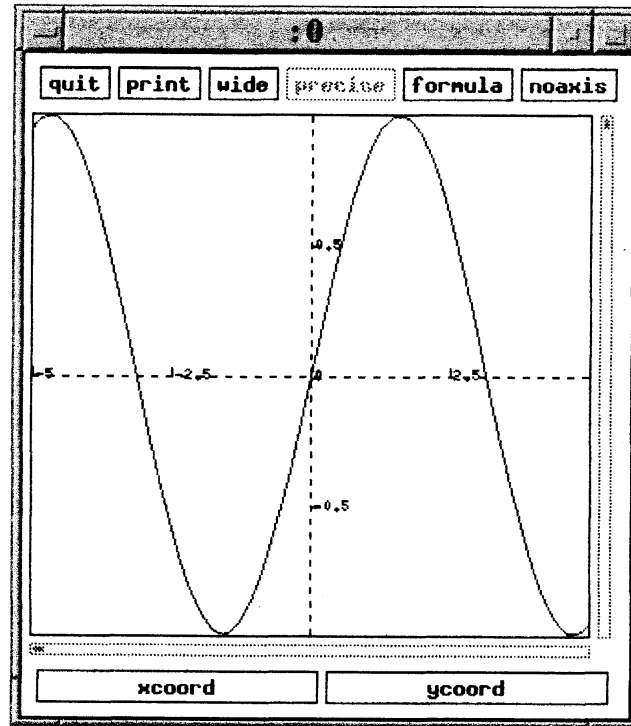
という点が問題となると考えられる。

3 実行例

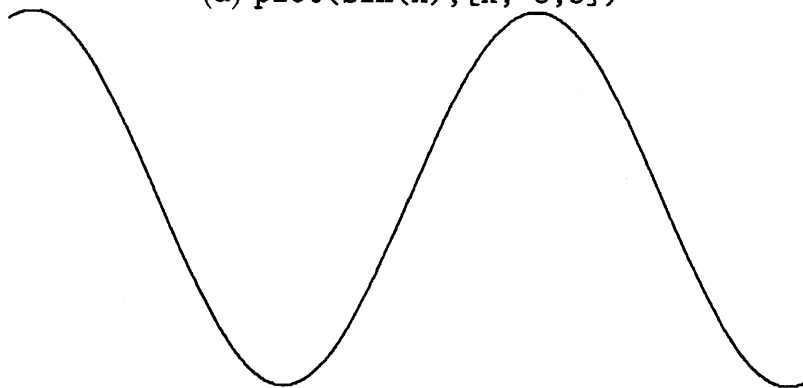
前節までの検討を踏まえ、プロット関連の試作を行った。

3.1 memory_plot() を利用した場合

例として、 $y = \sin x$ の出力例を図 1 に示す。



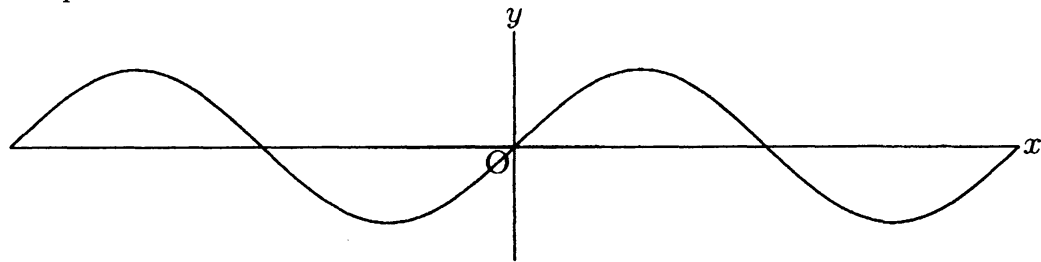
(a) `plot(sin(x), [x, -5, 5])`



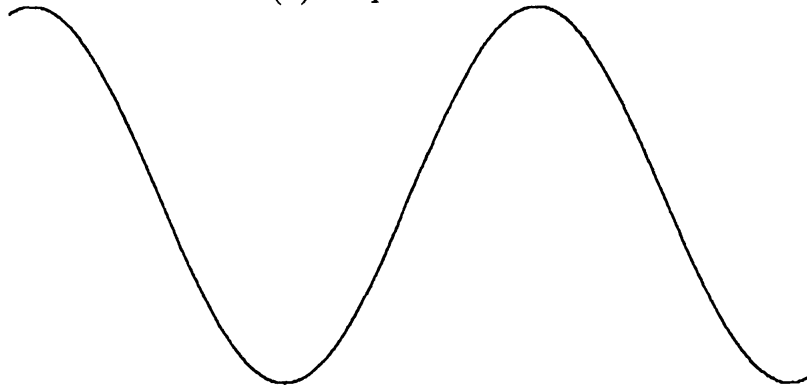
(b) 試作システムの出力

図 1: $y = \sin x$ の画面出力と試作システムの出力

また, Maple 版との比較を図 2 に示す.



(a) Maple 版の出力



(b) asir 版の出力

図 2: Maple 版との比較

3.2 memory_plot() 利用での問題点

memory_plot() 利用する上で, 次のような問題点が明らかになった.

- 点の値ではなく, ウィンドウ内の位置情報しかわからない.
 - 精度的に問題になる場合が出る.
- 並行移動した物は, 同じデータしか得られない.
- x の領域は与えるので良いが, y の領域が分からない.
よって, このままでは軸が書けない.
- 現在配布中の Windows 版では未実装である.

3.3 memory_ifplot() を利用した場合

Asir の特徴の一つでもある陰関数描画は, 点を書くアルゴリズム (実際には, 消すアルゴリズムである. 詳細は [4] 参照) であるため, 試作システムでも点と点を直線で補間せず, 点を置いて実現している. Asir のライブラリで準備されているハート型関数での出力例を図 3 に示す.

3.4 memory_ifplot() 利用での問題点

memory_ifplot() 利用する上で、次のような問題点が明らかになった。

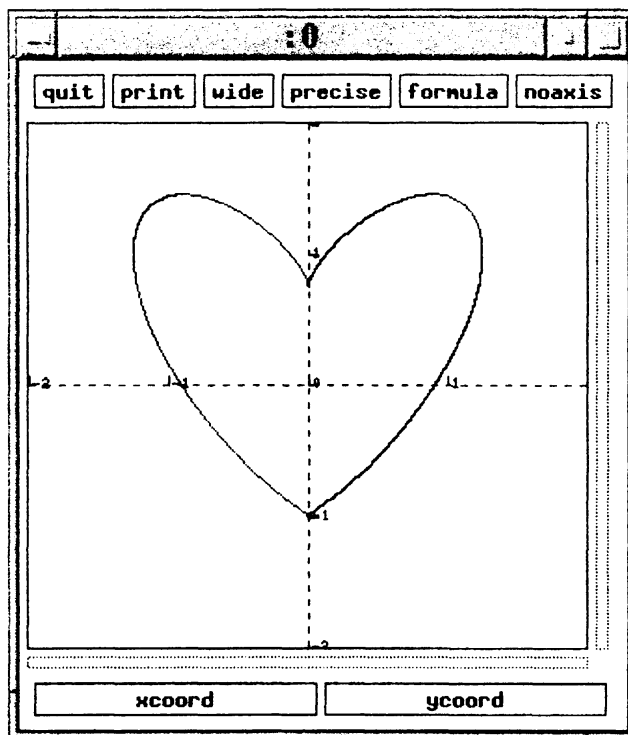
- デフォルトでは、解像度不足である。
- 点データしか得られない。(アルゴリズム上)
- そのため、安易な図でもデータ量が増える。

4 まとめと課題

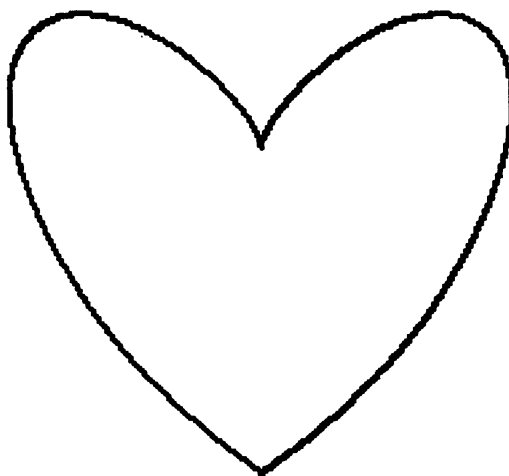
本稿では、挿図教材作成支援システムである KETpic を Risa/Asir へ移植するための検討を行った。特に、プロット関連での Asir 言語のみでの移植を試みた。結果として、Asir 言語だけでの実現は、いろいろな情報が不足するため、困難であることがわかった。今後は、組み込み関数の修正や追加を行いながら C 言語による組み込み関数と Asir 言語での実現を使い分ける必要がある。また、3D プロットなどの他の数式処理システムで実現されているが、Asir では利用できないものへの対応も行う必要がある。

参考文献

- [1] 阿部, 泉, 金子, 北原, 関口, 深澤, 山下, 高遠, T_EX 描画のための CAS マクロパッケージ KETpic の開発と改良, 数式処理, 15(2), 2008, 101-104.
- [2] 高遠節夫, 越川浩明, KETpic の移植について—CAS と Scilab の距離—, 数式処理, 15(2), 2008, 109-112.
- [3] M. Noro, T. Takeshima, Risa/Asir — A Computer Algebra System, *Proceedings of ISSAC'92*, 1992, 387-396.
- [4] 齋藤友克, 近藤祐史, 三好善彦, 竹島卓, Displaying real solution of mathematical equations, 数式処理, 6(2), 1998, 2-21.



(a) ハート型関数の画面出力



(b) 試作システムの出力

図 3: ハート型の画面出力と試作システムの出力