

システム記述言語の拡張言語「画匠」 6 P-9 を用いた文書作成方式

鈴木未来子, 狩野敦, 曾谷俊男, 並木美太郎, 高橋延匡

東京農工大学 工学部 情報工学講座

1. はじめに

我々は、大学の研究室などの環境を前提とし、事務文書から論文、ソフトウェア開発に関連する文書に至る、総合的な日本語文書清書機能をもつインテリジェントプリントサーバ「浄書」の開発を行っている[1]。浄書には、プログラムリスト出力ツールや、和欧混合文に従った行頭行末禁則処理やスペーシング処理などを施し整形出力を行う日本語文書フォーマッタなど、様々な出力ツールが用意されている。そして、これらの出力ツールを統合して利用することを可能とする統合化文書フォーマッタ「浄書参」を現在、開発している。

本稿で述べる浄書参における図形出力処理の方式は、インタラクティブな作図システムで行うような、プリミティブの指定だけをサポートする方式ではない。我々は、科学技術論文などを作成する場合に、数式で表されるような関数グラフなどを、数値計算により厳密に描けるようにすることの必要性を痛感した。そのためには、インタラクティブな作図システムでは実現が難しい。したがって、我々は、図形作成言語を用いて文書中の図をプログラミングする方式を採用した。そして、この図形作成言語を「画匠」と名付け、浄書参において、「画匠」で記述した図形情報と文書情報を一体で管理し、整形出力する方式を実現した。

2. 図形作成言語「画匠」の設計方針

2.1 図形を含む文書作成の現状

日本語ワードプロセッサなどを含む卓上出版システムは、年々機能が向上しており、その普及も目覚ましい。これにより、従来、印刷会社に印刷を依頼していたマニュアル、論文、広報用パンフレットなどを身近なところで作成することが多くなってきた。しかし、図形の編集機能に関しては、システムに用意されたプリミティブ図形を、インタラクティブな方式で作成するという段階にとどまっている。したがって、科学技術論文で引用されるような図形、例えば、

- (1) 合成関数
- (2) 三角関数の合成波
- (3) 3次元透視図

など、数値計算を必要とする正確な図形は、手書きやプロッタなどの別システムで作成し、後から手作業により

切り貼りしなければならない。

2.2 「画匠」とその処理方式

我々は、関数グラフなどを作成するには、インタラクティブ方式では不可能であると考え、言語による図形記述方式を採用した。また、浄書はパッチ型のフォーマッタであるので、言語で図形を指定するという体型をとることにより、文書と図を同一のテキスト上で一体管理できるという利点が生じる。

この方式を選択するとしても、現状のページ記述言語(例えば、PostScript [2] や pic) では、関数を正確に定義することは難しい。また、一から言語を設計したとしても、その言語処理系の構築が困難であるうえ、汎用のプログラミング言語に匹敵する機能をすべて備えた言語を開発することは容易でない。そこで、我々は、「浄書」やその開発システムのシステム記述言語である言語Cをベースとした図形作成言語「画匠」を設計し、実現することにした。すなわち、図形プリミティブや図形変換などの指定や、再利用可能な図形の定義などを、文書中に記述する方式を採用した。

3. 「画匠」の利用方法

3.1 「画匠」の機能と特徴

図形作成言語として、「画匠」には次にあげる機能を用意した。

(1) プリミティブ描画機能

“直線”, “点”, “円”, “円弧”, “楕円”, “文字列”, “プロット” という必要最低限のプリミティブ関数

(2) 図形修飾機能

「画匠」でプログラミングした図形に対して施す線修飾、面修飾などの制御文

(3) 図形変換機能

“回転”, “平行”, “大小” という、図形変換に関する制御文

(4) 図形変換文の定義機能

上記の他に自前の図形変換を関数定義すること

図形修飾と図形変換の指定は、言語Cの制御文のスタイルで、構造的に行えるように、制御文を用いる。すなわち、図形修飾や図形変換の制御文の先頭には、“@”を

付け、図形制御文であることを示し、その制御を施す画匠プログラムの範囲を“{”と“}”で囲み、明示する。

「画匠」の特徴の一つは、単に図形を書くだけにとどまらないことにある。その本質は、言語Cを拡張した言語である点にある。したがって、「画匠」には、言語Cの機能がすべて含まれている。また、OSが提供する言語Cのライブラリ関数も画匠プログラムの中で利用可能である。したがって、画匠プログラム中でファイル入出力関数を用いるなど、多彩なプログラミングが可能となる。

3.2 浄書テキスト中の「画匠」プログラム

画匠プログラムは、浄書のソーステキスト中において、“@箱(画匠)”というコマンドに対し、位置、領域の大きさなどを指定する。その時に、次の引数を指定する。

(1) 実行関数名

図を描く関数名を明記する。

(2) 原点の位置

原点(0,0)の位置を領域のどの位置にとるかを指定する。

(3) スケールの大きさ

プログラム中に現われる数値の1スケールに対する大きさを、XY軸方向ともに指定する。

(4) 画匠プログラム

これらの指定の後に“{”と“}”の間に画匠プログラムを記述する。

原点の位置とスケールの大きさの情報は、画匠プログラムの定義の外で与えるので、一度作成した図形プログラムを用い、大きさや位置のみ変更して別の文書に出力するという利用が可能である。

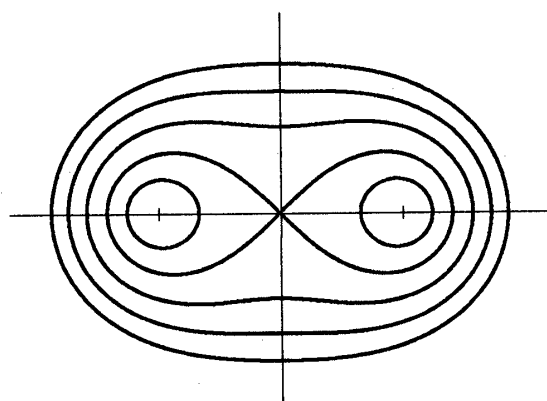


図1 カッシーニの卵形線

カッシーニの卵形線とは、2定点(焦点)の距離の積が一定になる点の軌跡である。この軌跡の極座表示は、

$$r^2 = a \cos 2\theta \pm b$$

$$(a = \text{焦点距離}^2, b = \sqrt{C^2 - \text{焦点距離}^4} \sin^2 2\theta)$$

で表わされる。パラメータC、および、焦点距離の関係により、以下のように軌跡が変化する。

- (1) $C^2 < \text{焦点距離}^2$: 焦点を囲む左右2つの卵形線
- (2) $C^2 = \text{焦点距離}^2$: ペルヌーイのレムニスケート
- (3) $C^2 < 2 \cdot \text{焦点距離}^2$: 外側に凹なまゆ形
- (4) $C^2 = 2 \cdot \text{焦点距離}^2$: 扁平な卵形
- (5) $C^2 > 2 \cdot \text{焦点距離}^2$: 外側に凸な楕円状の卵形

図1 カッシーニの卵形線

4. 画匠プログラムを含むテキスト展開処理

我々は、「画匠」の処理に、我々の研究室で開発した言語CコンパイラCATを用いている。つまり、画匠を含む文書ファイルから全ての画匠プログラムを切りだし、まず、言語Cのプログラムに変換する。そして、それらをコンパイル、リンクし、図形を描くための実行形式ファイルGORO (GASHO Output ROutline) を作成する。このGOROは、文書フォーマッタ浄書参において、図形出力を請け負うルーチンとなる。

テキスト中の「画匠」処理方式には、別タスクである言語処理系を文書走査系の中から起動するという作業が必要となる。我々は、これを我々の研究室で開発したOS/omeronのタスク管理を利用して実現した。

5. おわりに

言語Cの言語処理系を利用した図形作成言語の処理方式を提案し、実現したことは、言語の機能・拡張の面で有効であった。今後の課題としては、

- (1) 「画匠」用の図形入力エディタ
- (2) 統計量を表現するグラフィックス・ライブラリなどの構築である。

6. 参考文献

- [1] 里山元章、他: “文書の論理構造を備えた日本語清書システム「浄書」の設計と実現”, 情報処理学会論文誌, Vol.30, No.9, pp.1126-1134(1989).
- [2] 小方一郎: “レイアウト言語PostScript”, bit, Vol.18, No.12, pp.4-11(1986).

```

/*-----*/
/*      カッシーニ関数      */
/*-----*/
#include      (type.h)
#include      (stdio.h)
#define      pi      3.141592653
#define      平方根(x)      sqrt(x)

カッシーニ関数()
{
    カッシーニ( 2.0, 1.5, 20000 );
    カッシーニ( 2.0, 2.0, 20000 );
    カッシーニ( 2.0, 平方根( 6.0 ), 3600 );
    カッシーニ( 2.0, 平方根( 8.0 ), 3600 );
    カッシーニ( 2.0, 平方根(10.0), 3600 );
    座標( -5.0, 5.0, -4.0, 4.0 );
}

カッシーニ( 焦点距離, パラメータ, n )
/* パラメータ: このパラメータの変化により軌跡が異なる */
/* 焦点距離: 原点と焦点の距離 */
/* n: 座標の個数 */
{
    FLOAT      Δθ, θ, x, y, r, α, β;
    INT      i;

    Δθ = 2.0 * pi / (FLOAT)n;
    θ = 0.0;
    α = ベキ乗(焦点距離, 2);
    for ( i = 0; i < n; ++i ) {
        β = ベキ乗(パラメータ, 4) - ベキ乗(焦点距離, 4) * ベキ乗(sin(2.0*θ), 2);
        if ( β >= 0.0 ) {
            if ( (α * cos(2.0*θ) - 平方根(β)) >= 0.0 ) {
                r = 平方根( α * cos(2.0*θ) - 平方根(β) );
                x = r * cos( θ );
                y = r * sin( θ );
                ペンダウン( x, y );
            }
            if ( (α * cos(2.0*θ) + 平方根(β)) >= 0.0 ) {
                r = 平方根( α * cos(2.0*θ) + 平方根(β) );
                x = r * cos( θ );
                y = r * sin( θ );
                ペンダウン( x, y );
            }
        }
        θ += Δθ;
    }
}

```

図2 カッシーニの卵形線を描画する画匠プログラム