数理解析研究所講究録
第646巻 1988年 93-108

作図問題解答システム

広島大学工学部　　加古富志雄 (Fujio Kako)
筑波大学社会工学系　岸本一男 (Kazuo Kishimoto)
広島大学工学部　　山本潔 (Kiyoshi Yamamoto)
広島大学工学部　　翁長健治 (Kenji Onaga)

初等幾何学は古い伝統をもつ一方で、現在においても、製図、グラフィクスなどをはじめとして、さまざまな分野の基礎となっている。 このような理由から、初等幾何の定理自動証明、作図問題などの初等幾何学の諸問題は注目を集め、解答システムの設計、試作が試みられてきた。 しかしながら、これらの多くは人工知能の手法を用いて、試行錯誤的なアプローチを取るものであり、偶然正しい解答にたどりつける場合もあるが、正誤の判定が行なえない場合も起こりえるという意味で不完全なシステムであった。このような不完全性を解決し、常に正しい解答を得るためには決定的でアルゴリズミックな算法が必要となってくる。

初等幾何学の大半の問題は座標を持ち込むことによって代数的な問題に変換しうることが知られている。 問題が一旦代数的な形に定式化されたならば、計算機によってその解をアルゴリズミックに求めることが可能となる。 ここでは、数式処理システム（ＲＥＤＵＣＥ）を用いた、作図問題の自動解法システム作成の試みについて述べる。

作図問題とは一般には、

『ある特定の器具だけを有限回用い、与えられた条件を満たす図形を描く』

という問題であるが、ここでは初等作図問題と言われる、

定規　　　：与えられた２点を通る直線を引く
コンパス　：与えられた点を中心とし、与えられた点を通る円を描く

の二つの器具を用いて作図を行なう問題を考える。

作図問題を代数学に翻訳すると、定規とコンパスを用いて行なえる操作は、四則演算と平方根を開く演算のみが可能である。 また四則演算と平方根を開く演算のみで表さ

れた点は、全て作図可能である。 一般には、作図条件としては、等式で表される条件のみではなく、不等式で表される条件も考えられるが、ここでは等式条件の場合のみに限って考える。 従って問題は、指定された条件を代数方程式に変換したときに、その方程式が四則と平方根を開くことのみで解くことが可能か否を判定すること及び可能な場合にはその根を実際に求めることに帰着できる。

システムの構成としては、

1）　作図問題の入力と方程式への変換：
　入力された作図条件をそれと等価な連立代数方程式に翻訳する。

2）　連立代数方程式の平方根による代数的解法：
　四則演算と平方根へ開く操作のみで解くという制約のもとで、連立方程式の求解を行なう。 これには、まず連立方程式から変数を消去し、解くべき１変数代数方程式を導出する。 得られた方程式が、二次方程式を次々に解くことで解きうるか否かを判定し、可解であればその根を求める。 この解を元に、連立方程式の解を構成する。 もし可解でなければ、作図不可能と返答する。

3）　作図手順の構成：
　四則演算と平方根で表されている連立方程式の解から、作図問題の解となる作図手順を構成する。 さらには、この手順に従い、与えられた数値に対して実際に作図を行なう。

となる。

１．　作図問題の入力と方程式への変換

作図問題は、基本となる図形（ここでは点、直線と円周からなる）に対する条件からなる。 基本図形は、座標系を設定することで、適当な定数を用いて、

点　　：$x=P_1,\quad y=P_2$

直線　：$L_1x+L_2y+L_3=0,\quad L_1{}^2+L_2{}^2=1$

円周　：$(x-Q_1)^2+(y-Q_2)^2-Q_3{}^2=0$

と表現できる。 システムの内部では、基本図形はこれ等の定数（点は（P_1，P_2）、直線は（L_1，L_2，L_3）、円周は（Q_1，Q_2，Q_3）の組で）で表現する。 作図条件は、これ等の図形間の関係を述語の形で指定することで表される。 基本的な述語としては、

ON（点Ｐ、直線Ｌまたは円周Ｑ）

点Ｐが直線Ｌまたは円周Ｑの上にあることを表す。

CENTER(点Ｐ、円周Ｑ)

円周Ｑの中心が点Ｐであることを表す。

PERPENDICURE(直線L_1、直線L_2)

二つの直線が垂直であることを表す。

PARALLEL(直線L_1、直線L_2)

二つの直線が平行であることを表す。

を用意している。また、二点間の距離を与える関数DISTANCEと二つの直線の成す角度を与える関数ANGLEを用意している。

これ等の述語や、関数を元に、新しい述語や関数を定義することも可能で、これは、

```
<述語の定義>  ：：＝  PREDICATE 述語名（図形名1，．．．）；
                        BEGIN
                          図形名の宣言1；
                          図形名の宣言2；
                             ．
                          作図条件の記述1；
                          作図条件の記述2；
                             ．
                        END;

<関数の定義>  ：：＝  FUNCTION 関数名（図形名1，．．．）；
                        BEGIN
                          図形名の宣言1；

                          作図条件の記述1；
```

```
            RETURN 関数の値；
          END;
```

のシンタックスで定義される。　例えば、点Pに於ける円Qの接線がLであるという述語Tangentは、

```
PREDICATE Tangent(P,Q,L);
  BEGIN POINT A,B,P;       %点の宣言；
        LINE M,L;          %直線の宣言；
        CIRCLE Q;          %円の宣言；
    ON(B,Q); CENTER(A,Q);
    ON(A,M); ON(B,M); ON(B,L); ON(P,L);
    PERPENDICURE(M,L);
  END;
```

と定義できる。

実際に作図を行なう図形は、コマンドCONSTRUCTで指定する。　これは、次のように指示する。

```
CONSTRUCT(Tangent(P,Q,L),L);
```

第一引数は、作図条件の指定であり、第二引数は、未知図形即ち作図すべき図形の指定である。　未知図形の指定がされない図形は、既知図形である。　このコマンドにより、作図問題の入力・解析部は、直交座標系を導入し、図形を方程式で表現する。　そして、作図条件から、連立方程式を立てる。　例えば、点AとBが与えられたときに、点A，B，Cが正三角形の頂点を成すような点Cを求める問題を考えると、作図条件は、

```
PREDICATE Triangle(A,B,C);
  BEGIN POINT A,B,C;
    DISTANCE(B,C)=DISTANCE(A,B);
    DISTANCE(C,A)=DISTANCE(A,B);
  END;
```

により、述語Triangleで与えられる。　更に、この条件下で、点Cを求めることがあたえられた問題であり、従って、

```
CONSTRUCT(Triangle(A,B,C),C);
```

で作図問題が指定される。

これから、方程式は、点Ａの座標を（Ａ１，Ａ２）等と表せば、

$$(B1-C1)^2 + (B2-C2)^2 = (A1-B1)^2 + (A2-B2)^2,$$
$$(C1-A1)^2 + (C2-A2)^2 = (A1-B1)^2 + (A2-B2)^2$$

を未知数Ｃ１，Ｃ２について解く問題に変換される。

２．　連立代数方程式の代数的解法

連立方程式の根を代数的に求めるためには、まず変数を消去して一変数の方程式を導出し、その解をもとに残りの方程式を解くことになる。　変数を消去するアルゴリズムとしては、

（ａ）　消去法
（ｂ）　一般消去法
（ｃ）　Groebner基底による方法

が知られている。　ここでは、最も計算量が少ないと思われるGroebner基底を用いた方法によって消去を行なう。　Groebner基底の計算において、変数の順序として辞書式順序を使うと、得られる基底の内には必ず一変数の方程式が含まれる。　従って、まずこの方程式を解き、これを付加した体（拡大体）を上で、残りの方程式を解けば良い。
しかし、辞書式順序の場合にはGroebner基底を構成するための手数はその変数の順序の取り方に強く依存しているため、場合によってはかなり計算時間がかかる。　一般に全次数－辞書式順序を用いたほうが、計算時間が早くなる。　Moritsuguは、n変数の方程式系から、全次数－辞書式順序を用いて得られたGroebner基底に対して変換を行なうことにより、適当な仮定のもとで次の様な標準型にもっていけることをしめした。

$$\{h_1(x_1), x_2-h(x_1), x_3-h_3(x_1), \ldots, x_n-h(x_n)\}$$

ここでは、解くべき方程式はx_1対するもののみで、他の変数x_j $(j=2,\ldots,n)$については得

られた解x_1を他の方程式に代入することで得られる。

アルゴリズム１　$x_j-h(x_1)$となる多項式を計算する(Moritugu 1987)

```
%input: Groebner基底G ;
%output: p = x_j - h(x_1);
      M := deg(h_1(x_1)) - 1;
      for k:=1:M do p_k := NormalForm(G,x_1^k);
      NormalForm(G,x_j)+d_0+d_1p_1...+d_Mp_M = 0をd_k(k=0,..,M)について解く;
      p := x_j + d_0 + d_1x_1 + ... + d_Mx_1^M;
      return p;
```

ここで、NormalForm(G,f)は、ｆを基底Gで簡約した多項式を表す。

３．方程式の平方根による可解性と解の構成

必要なら素因数に分解することにより、ここでは多項式ｆは既約とする。　方程式ｆ＝０が可解であるとは、多項式に基礎体（ここでは有理数体または有理関数体）に巾乗根を次々に付加して得られた拡大体上で一次因子にまで因数分解できるかどうかということである。　方程式$f=0$が代数的に可解であることと、方程式のGalois群（$f=0$の根$\{a_1,a_2,...,a_n\}$の置換全体のなす群）が可解であることは同値であることが示されている。　ここで群Ｇが可解であるとは、部分群の有限鎖：

$$1 = G_0 \subsetneq G_1 \subsetneq \cdots\cdots \subsetneq G_r = G,\ \text{ただし}\ 1\ \text{は単位群}$$

があって、$i = 0,1,...,r-1$に対して

(1) G_iが G_{i+1}の正規部分群である
(2) G_{i+1}/G_iはアーベル群である。　即ち、全ての元に対して交換法則が成り立つ。

が成り立っている場合をいう。

したがって、可解性の判定はまず方程式からそれに対応するGalois群を求め、このGalois群が可解であるか否かで判定できる。　しかし、Galois群の構成には現在知られている方法では、一般の多項式では次数の指数時間かかってしまう。

ここでは、方程式が平方根のみで解きうるかの判定をGalois群の構成によることなく行なうアルゴリズムを示す。 これには、まずBlockを定義する。

> 集合 $\Omega = \{a_1, a_2, \ldots, a_n\}$ とそれに作用する可推移な置換群Gが与えられた時、任意の$\sigma \in G$に対して、$\sigma(B)=B$ or 0を満たす Ω の部分集合BをBlockと呼ぶ。 単位要素のみからなる集合と Ω は共に自明なBlockであり、それ以外のBlockを自明でないBlock(Block of Imprimitivity)と呼ぶ。

また、各ブロックに対して、ブロックを不変にするようなGalois群の部分群が対応する。方程式が平方根のみで解きうる場合には、方程式の次数nは2の巾乗($n=2^N$)であり、必ずサイズ2^i ($i = 1, .., N$)のBlockを持つ。 したがって、多項式f(x)にその一根aを付加した体上で因数分解すると、

$$f(x) = (x-a)(x-g_1(a))(x-g_2(a)) \ldots (x\text{の2次以上の因子}) \ \ldots$$

と分解できる。 ここで、一次因子の数は2以上であり、また2次以上の既約因子を持つ場合にはその次数は必ず2の巾乗である。 そうでない場合には、この方程式は平方根のみでは解きえない。

これから、

$$g_j(g_j(a)) = a$$

となるjを求めると $\{a, g_j(a)\}$ はサイズ2のブロックを成す。 この時、

$$(x-a)(x-g_j(a)) = x^2 + h_1(a)x + h_2(a)$$

として(もし二次の既約因子を持てば、その係数をh_1, h_2としてもよい)、基礎体をKとすると

(1) $h_1(a) = c \in K$
(2) $h_2(a) = c \in K$
(3) $h_1(a) = b\, h_2(a) + c,\quad (b, c \in K)$
(4) それ以外の場合

の4つの場合が考えられる。 この内最初の3つの場合にはそれぞれ、

(1) $z = (x + c)x$
(2) $z = (x^2 + c)/x$
(3) $z = (x^2 + c\,x)/(b\,x + 1)$

と置くことで方程式f(x)=0はzに関するn/2次の方程式に変換される。 なぜならば、b,cは基礎体の要素でありしたがって方程式のGalois群による変換に対して不変で、この変換式は、全てのブロックに共通となるからである。 問題は４番目の場合で、この時は次に大きいBlockを探す。 ４次の既約因子を持てば、それがサイズ４のブロックを成す。
また、

$$g_j(g_j(g_j(g_j(a)))) = a,\ g_j(g_j(a)) \neq a$$

であれば、 $\{a, g_j(a), g_j(g_j(a)), g_j(g_j(g_j(a)))\}$ がサイズ４のブロックになる。 これから作ったxの４次式を

$$x^4 + h_1(a)x^3 + h_2(a)x^2 + h_3(a)x + h_4(a)$$

としたとき、各$h_j(a)$が全て一次関係にあり、

$$(x^4 + b_1x^3 + b_2x^2 + b_3x + b_4) + h(a)(c_1x^3 + c_2x^2 + c_3x + c_4)$$

と表されれば、これから

$$z = (x^4 + b_1x^3 + b_2x^2 + b_3x + b_4)/(c_1x^3 + c_2x^2 + c_3x + c_4)$$

と置くことで元の方程式はzに関するn/4次の方程式に変換される。 以下同様に行ない、もしサイズn/2のブロックを作っても変換できない場合は、この方程式は平方根のみでは可解でないと判定できる。

このようにして元の方程式を二組の方程式への分解を行ない、最終的に二次式の方程式の組にまで変換できれば可解であると判定できる。 また、解の構成は、得られた二次方程式を解き、この解を変換式に代入して元のxについて解くことによって得られる。

ここで示した方法は、拡大体上の因数分解の手間だけであり、Lenstraのlatticeを使った因数分解のアルゴリズムを使用すれば方程式の次数の多項式時間で実行できる。
しかしながら、ノルムを使った拡大体上の因数分解のアルゴリズムでは、方程式の次数nに対してn(n-1)次の多項式を因数分解しなければならず、現在の計算機では８次方程式が限界であり、より高速のアルゴリズムを開発する必要がある。

４．　作図手順の構成

ここでは、四則演算と平方根を開く操作のみで表現された連立方程式の根をコンパスと定規で作図する手順の構成について説明する。　これには、基本操作として、

(1) L=line(P1,P2)

二点を結ぶ直線を描き、その直線をＬとする。

(2) L=parallel(L1,P)

点Ｐを通り、直線Ｌ１に平行な直線Ｌを描く。

(3) P=intersection(L1,L2)

二つの直線Ｌ１とＬ２の交点をＰとする。

(4) P=circle+(P1,P2-P3,L)
(5) P=circle-(P1,P2-P3,L)

点Ｐ１を中心とし、二点Ｐ２，Ｐ３を両端に持つ線分Ｐ２－Ｐ３を半径とする円を描き、その円と直線Ｌが交わる点をＰとする。　この交点は二つ存在するが、circle+の場合には、点Ｐ３がＰ２より座標が大きい場合には交点の座標の大きな方を採用し、逆の場合には、交点の座標の小さい方を採用する。　circle-の場合には、その逆である。

(6) P=midpoint(P1,P2)

点Ｐ１と点Ｐ２の中心点をＰとする。

の６つの操作で作図手順を記述する。
次に、これ等の基本操作を組合せて、四則演算と平方根へ開く作図操作を示す（図１）。

(1) C = A + B

```
C = circle+(A,O-B,Ox)
```

(2) C = A - B

```
C = circle-(A,O-B,Ox)
```

(3) C = A * B

```
P2 = circle+(O,O-B,Oy)
L1 = line(y1,P2)
L2 = parallel(L1,P2)
C = intersection(L2,Ox)
```

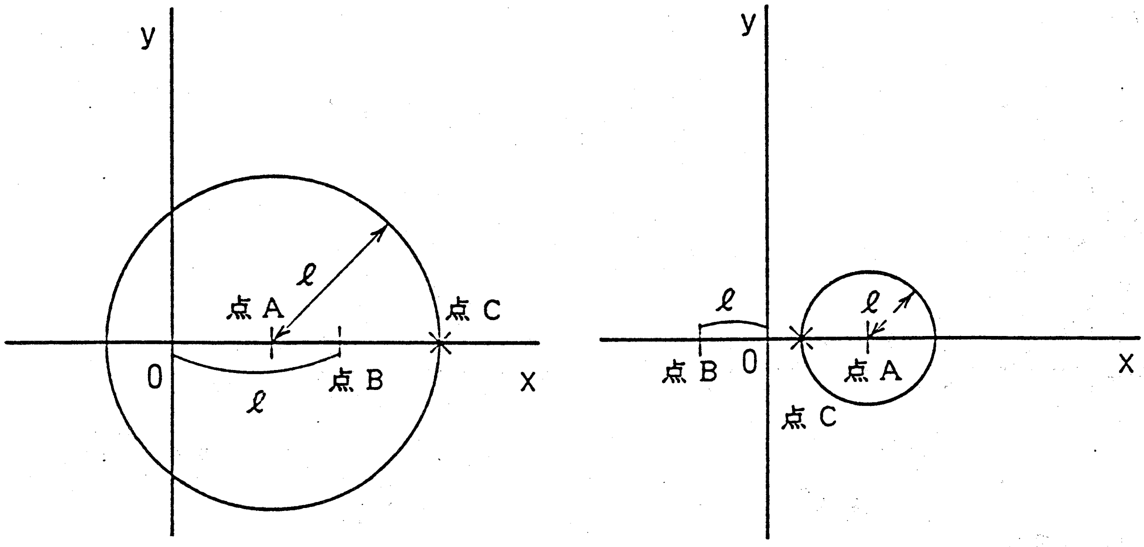
(4) C = A / B

```
P1 = circle+(O,O-A,Oy)
P2 = circle+(O,O-B,Oy)
L1 = line(x1,P1)
L2 = parallel(L1,P2)
C = intersection(L2,Ox)
```

(5) C = $\sqrt{A}$

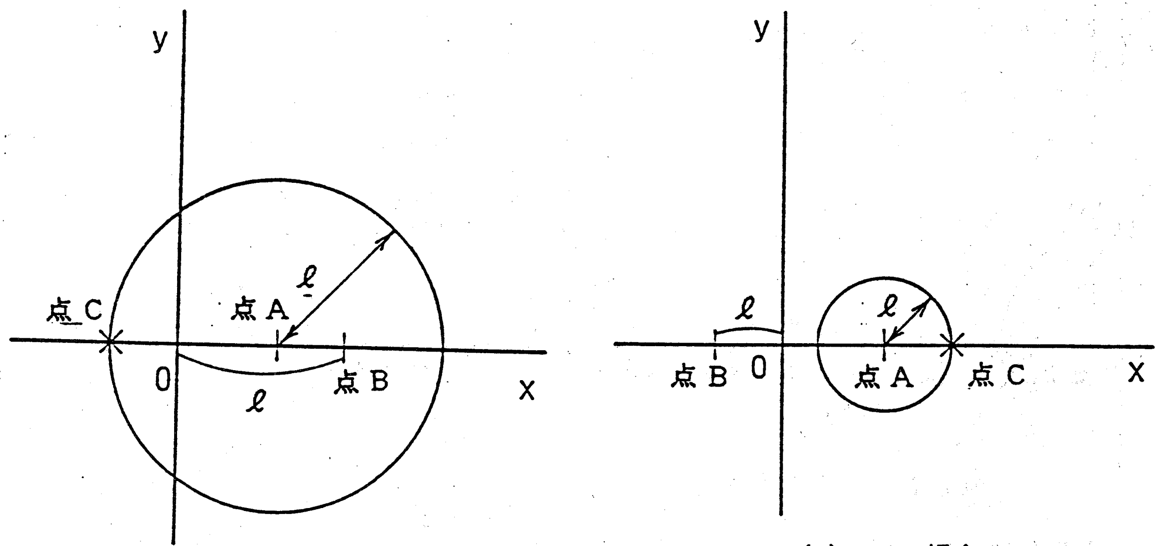
```
P1 = circle-(O,O-A,Oy)
P2 = midpoint(P1,y1)
C = circle+(P1,P2-y1,Ox)
```

ここで、Ox,Oy,x1とy1はそれぞれ、X軸、Y軸、X軸上単位長さの点とY軸上単位長さの点を表す。　未知数が既知の数の四則演算と平方根とで表されていれば、その式にしたがって以上の操作を順に繰返すことによって、作図手順は構成出来る。



(a) B>0の場合　　　(b) B<0の場合

図1．1　C = A + B の作図



(a) B>0の場合　　　(b) B<0の場合

図1．2　C = A − B の作図

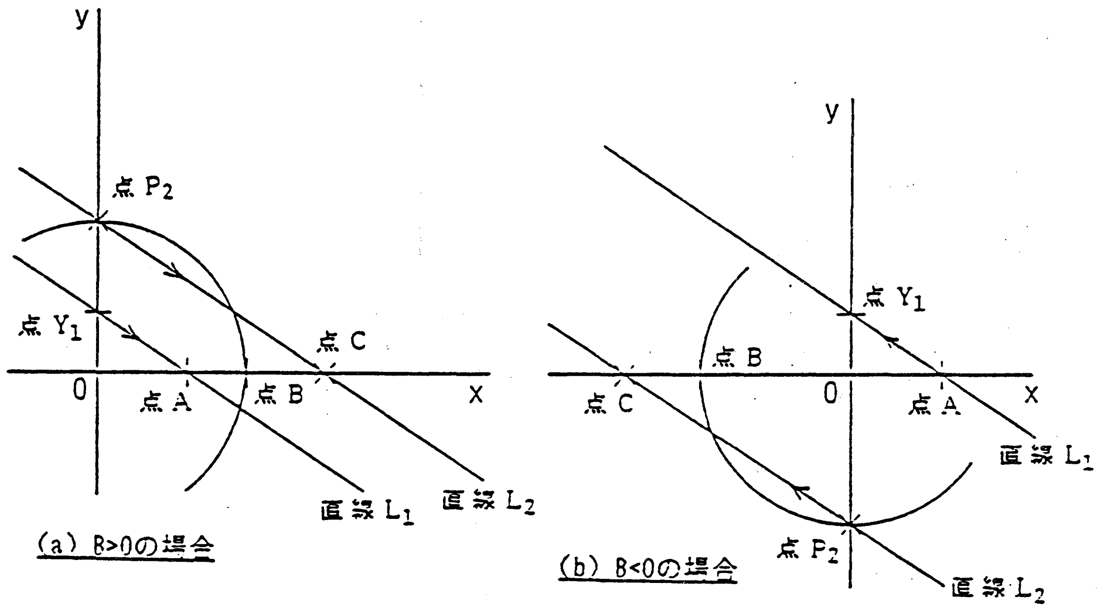


図1．3　C＝A＊B の作図

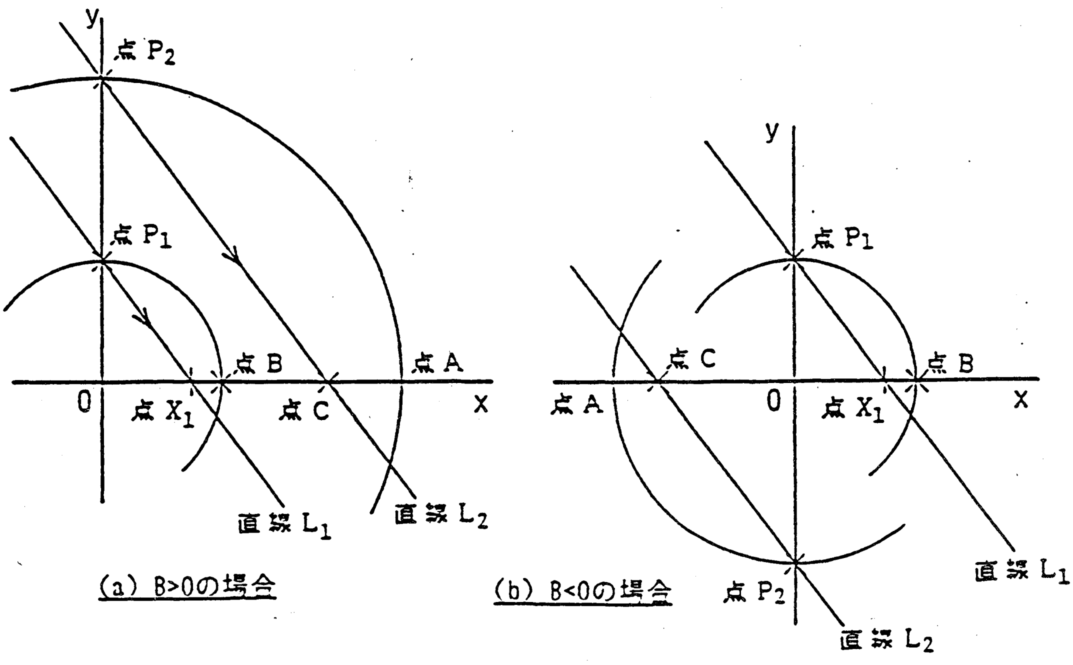


図1．4　C＝A／B の作図

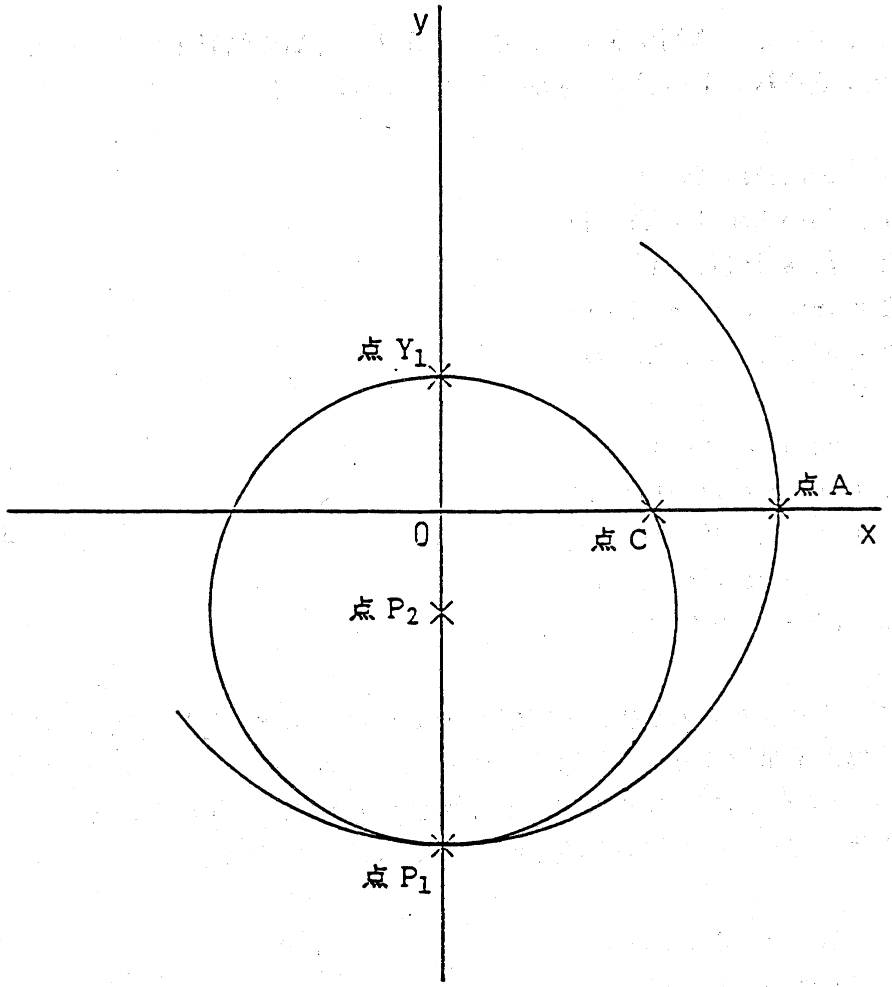


図1．5　C = $\sqrt{A}$ の作図

最初に挙げた正三角形の作図問題の例では、連立方程式の解は、

```
C1 = (sqrt(3)*(A2 - B2) + A1 + B1)/2
C2 = (sqrt(3)*(A1 + B1) + A2 + B2)/2
```

```
C1 = (sqrt(3)*(A2 + B2) + A1 + B1)/2
C2 = (sqrt(3)*(A1 - B1) + A2 + B2)/2
```

となる。　この解から作図手順を求める。　まず、既知の図形である、点A，Bを一意に定める数を表す線分を作図する手順は以下のようになる。

```
L1 = parallel(Oy,A)
P1 = intersection(Ox,L1)
L2 = parallel(Ox,A)
P2 = intersection(L2,Oy)
P3 = circle+(O,O-P2,Ox)

L3 = parallel(Oy,B)
P4 = intersection(Ox,L3)
L4 = parallel(Ox,B)
P5 = intersection(L4,Oy)
P6 = circle+(O,O-P5,Ox)
```

この手順により点AのX座標、Y座標はそれぞれ線分O-P1,O-P3で与えられる。　また点BのX座標、Y座標はそれぞれ線分O-P4,O-P6で与えられる。　以上で得られた点P1,P3,P4,P6を用いて点CのX,Y座標を求める作図手順を方程式の解から構成する。　一番目の解のC1を例にとり、その作図手順を示すと以下のようになる。

```
P7 = circle-(O,O-x3,Oy)              ;sqrt(3)
P8 = midpoint(P7,y1)
P9 = circle+(P7,P8-y1,Ox)

P10 = circle+(O,O-P9,Ox)             ;sqrt(3)*A2
P11 = circle+(O,O-P3,Oy)
L5 = line(y1,P10)
L6 = parallel(L6,P11)
P12 = intersection(L6,Ox)

P13 = circle-(O,O-x3,Oy)             ;sqrt(3)
P14 = midpoint(P13,y1)
P15 = circle-(P13,P14-y1,Ox)
```

```
P16 = circle+(0,0-P15,0x)          ;sqrt(3)*B2
P17 = circle+(0,0-P16,0y)
L7 = line(y1,P16)
L8 = parallel(L7,P17)
P18 = intersection(L8,0x)

P19 = circle-(P12,0-P18,0x)        ;sqrt(3)*(A2-B2)

P20 = circle+(P19,0-P1,0x)         ;sqrt(3)*(A2-B2)+A1

P21 = circle+(P20,0-P4,0x)         ;sqrt(3)*(A2-B2)+A1+B1

P22 = circle+(0,0-P21,0y)          ;(sqrt(3)*(A2-B2)+A1+B1)/2
P23 = circle+(0,0-x2,0y)
L9 = line(x1,P1)
L10 = parallel(L9,P23)
P24 = intersection(L10,0x)
```

以上により、線分0-P24で与えられる。 同様にしてC2の作図手順も構成できる。 また、二番目の解についても、同様である。 ここでの作図手順は冗長な部分を多く含んでいる。 例えば、 sqrt(3) の作図手順を複数回含んでいる。

最終的に点CのX座標、Y座標であるC1とC2を長さにもつ線分を0-C1,0-C2とする。 これらから点Cの作図手順は、

```
L100 = parallel(0y,C1)
P100 = circle+(C1,0-C2,L100)
```

により、点P100で与えられる。

ここで見るように、このシステムよって生成される作図手順は、かなり、冗長なものになっている。 この冗長な部分を除き、最適化された手順を出力することは、今後の課題である。 また、最終的には、この手順に従い、実際の作図を行なうFORTRANプログラムを生成するようなシステムを作成する。

参考文献

S.Landau and G.L.Miller: Solvability by Radicals Is in Polynomial Time, J. Comp. and System Sci. Vol. 30 (1985) pp. 179-208

R.P.Stauduhar: The Determination of Galois Groups Math. Comp., Vol. 27, No. 124 (1973) pp. 981-996

M.D.Atkinson: An Algorithm for Finding the Blocks of a Permutation Group, Math. Comp., Vol. 29, No. 131 (1975) pp. 911-913

C.C.Sims: Computational methods in the study of permutation group, Computational Problems in Abstruct Algebra (Proc. Conf., Oxford 1967) Pergamon Press, Oxford 1970 pp. 169-183

A.K.Lenstra: Lattice and factorization of polynomials over algebraic number field, Computer Algebra (ed. J.Calmet), Lecture Notes in Computer Science, Vol. 444 (Springer Verlag 1982) pp. 32-39

J.McKay: Some remarks on computing Galois groups, SIAM J. Comput., Vol. 8, No. 3 (1979) pp.344-347

S.Landau: Factoring Polynomials over Algebraic Number Fields SIAM J. Comput., Vol. 14, No. 1 (1985) pp.184-195

M.Moritugu: On solving a system of algebraic equations by using Groebner basis 数理解析研究所講究録 612 (1987) pp.25-37

藤瀬哲朗、小林英恒：連立方程式のある種の解法について ー代数的・数値的算法による全解構成法ー、 数式処理と数学研究への応用、数理解析研究所講究録 ５２０（昭和５９） pp. 1-22

稲永健太郎、志村正道：学習能力を持った幾何学問題解答システム、情報処理学会第３３回（昭和６１年後期）全国大会（昭和６１－１０） pp. 1673-1674

エム・ポストニコフ：ガロアの理論、 （日野寛三訳） 東京図書（昭和３９）

ファン・デル・ヴェルデン：現代代数学１，２，３、（銀林浩訳）東京図書（昭和３４）

A.C.Hearn: REDUCE User's Manual version 3.1, Rand Corporation (1984)

J.Fitch: Manual for Standard Lisp on IBM System 360 and 370, (1978)

J.B.Marti, A.C.Hearn, M.L.Griss and C.Griss: Standard LISP Report, (1978)