幾何言語（ＧＥＯＭＥＴＲＹ）の開発

名古屋大学工学部　正員　島田静雄

要旨

幾何学は本質的に代数学とは異なった数学であって、人間の感覚に訴えて理解することが極めて多い。幾何学的性質をコンピューターで処理させようとすると、座標系の定義や種々の約束事を定めて計算に乗せるのであるが、解が一つとは限らないし、解の性質それぞれに幾何学的な意味を伴う。我々が幾何の命題を言葉（言語）で表現することを、コンピューターも解読して必要な処理を施すようなプログラム言語をＧＥＯＭＥＴＯＲＹと呼ぶことにした。例えば、「２点Ａ，Ｂを結ぶ直線と、Ｃ，Ｄを結ぶ直線との交点Ｅを求める。」という命題を記述するとき

Ｌ１＝Ａ@Ｂ：　Ｌ２＝Ｃ@Ｄ：　Ｅ＝Ｌ１&Ｌ２

または一行で　Ｅ＝（Ａ@Ｂ）&（Ｃ@Ｄ）

のように書いてみようというのである。

このような表現を可能とする言語の開発は、コンパイラーを開発することと同じである。この場合、データのタイプとして実数型、整数型という考えを拡張して、ベクトル、直線、平面、マトリックスなども含めることが必要になる。代入則や演算則は当然のことながら代数計算とは異なった定義とした。

１．概説

本報文で述べる幾何言語（ＧＥＯＭＥＴＲＹ）の特徴をかいつまんで説明すると、ベクトル、点、直線、平面などの幾何学的要素を整数、実数などと共に変数の型として扱うようにしたもので、ベクトルの加減算やスカラー倍の算法も代数式の形で書けるようになっている。演算の記号に特殊なものを加えて、例えば二点を通る直線をＬ＝Ａ@Ｂ、二直線の交点をＰ＝Ｌ１&Ｌ２のように書けるようにしたものである。

そもそも、幾何は非常に感覚的な理解の上に成り立っていて、数（デジタル的）の学問とはかなり質が異なっている。幾何の解析に数学を応用するのは幾何の発展に役だってきたが、同時に厳密さと引き替えに直観的な理解を妨げて来ていることも事実である。例えば三角形の外心を図的に求める方法は誰しも理解している。しかし、これをコンピュータで計算させることを考えると、かなり面倒な儀式が必要になる。まず座標系の定義を決める。三点の座標から幾何の原理に従って外心の座標の理論式を求める。この理論式がかなり面倒な式になるので、相当に数学に強い人でも式の誘導に苦労するに違いない。この計算をプログラムに書いたとすると、説明がなければ何を計算するかを理解することは恐らく不可能である。この理由は、幾何の問題を説明する文章（言語）とコンピュータのプログラム言語とに大きな差があるからである。

我々はＦＯＲＴＲＡＮなどのプログラム言語の習慣で幾何の問題を解こうと努力して来たが、これらの言語が幾何処理に適しているか否かにさして疑いを持っていなかった。言い換えると、扱うデータの型は整数、実数、倍精度などである。最近は文字型を扱う言語も増え、それとともに文字型の演算の約束や関数が現われた。幾何処理に当たって、点、直線、平面などにデータの型を定義して処理する言語を考えてもよいわけである。この発想が幾何言語（ＧＥＯＭＥＴＲＹ）の開発の出発点である。具体的に作業を進めるに当たって、二つの重大な課題がある。一つは文法の確立であり、もう一つはコンパイラーもしくはインタプリターの開発である。

まず文法についてであるが、少なくとも中学、高校程度の幾何の知識があれば理解できる簡明さが必要である。このために初等幾何に立ち戻って幾何の見直しを行ない、基本的な型の定義、型同士の間の演算則の決定、関数の定義などをつめた。第二はインタプリターの開発であるが、これをＦＯＲＴＲＡＮで書いた。従って、ＧＥＯＭＥＴＲＹはＦＯＲＴＲＡＮの環境の下で実行されるが、ユーザーはマイクロコンピュータのＢＡＳＩＣに似た使い方になるように設計してある。幾何の問題を扱うにはグラフィック処理が切り離せない。特に作図命令を用いなくても、点や直線を計算した時点で画面上にこれを表示させる、と言う仕様になっている。これは、我々が紙に図を描きながら問題を考える習慣を生かしたいと考えたためである。

２．幾何学的要素

幾何学の定義とは、点・直線・平面並びにその組み合わせで得られる図形の性質を解析する数学の一分野であるとされている。数の学問である代数学の出発は、有理数・無理数・実数・整数などのように数を分類することから始めた。幾何学の出発も同様に点・直線・平面などを基本的な要素と考えるのであるが、これらの要素は数とは全く性質の異なるものである。まず、単位となるものを持っていない。加減乗除の四則演算は成り立たない。二直線の交点は、二直線を掛け合わせるという数字の演算則とは全く異なった法則で決まる。強いて理屈をつけると、二直線を集合要素と考えたときの論理積である。

幾何学に数の学問を応用する、つまりデジタル的処理をさせる、には座標系の概念を必要とする。点・直線・平面などを定義するには、この座標系の助を借りる。世界座標系（ワールド座標系）は最初から既に存在しているものと考えるのである。２つの三角形の相似や合同を判別するのは小学校の児童にも感覚的に理解できる。この場合、座標系の概念は必要がない。したがって、後で述べるように直線の定義を３個の数の集合（ａ，ｂ，ｃ）で表わすことを納得し、また、この数値を見てどのような直線であるかを感覚的に理解するには、多少の予備的な説明をする必要がある。

まず、幾何でいう直線は空間で無限に長く延びた直線を指す。図として紙に書いた２点Ａ，Ｂを結ぶ線分と同じ扱いをすることはできない。データとしてこれを区別するには、前者を（ａｘ＋ｂｙ＋ｃ＝０）の式を考えた３つの係数（ａ，ｂ，ｃ）を考え、後者は２点の座標値（x_1，y_1），（x_2，y_2）を一まとめにした４つの数値を考える。点、直線、線分にＡ，Ｌ，Ｅのように名前をつけておき、この名前で引用された場合には、それぞれの構成成分である一組の数値が参照されるようにする。これを変数名の型の定義という。幾何言語ＧＥＯＭＥＴＯＲＹには差し当たって16種類の型が使用される（表－１）。この中で二次元の座標値の扱いはＦＯＲＴＲＡＮの複素数の定義と近い。

英字の名前ＬＡを直線として扱うと約束するには型の定義文ＤＥＦ２ＬＮ　ＬＡを宣言する。データを与えるにはＢＡＳＩＣ言語のようにＲＥＡＤ－ＤＡＴＡ文を使うが、この際３個の数値が要求される。ところで、幾何言語ＧＥＯＭＥＴＯＲＹの扱いにおいては、読み込まれたデータがそのまま記憶されるのではなく、あらかじめ決められた標準の構造に変えられる。ＬＡは（ａ，ｂ，ｃ）を構成要素とするが、（ａ，ｂ）が単位ベクトルとなるように調整される。このようにすることでベクトル（ａ，ｂ）はこの直線の法線方向を与え、ｃはこの直線と原点の距離となる。（ａ，ｂ，ｃ）の符号をつけ変えたものは図形的には同じ直線であるが、単位ベクトルの定義は反対向きに定められる。

実は、このような約束にはその直線に正負の向きを含ませていて、平面領域を相対的に直線の正側、負側に区別するにのに用いる。また距離ｃの符号は原点の位置がその直線に対して正側にあるか、負側にあるかをも示すようになっている。代数学において直線の式をｙ＝ａｘ＋ｂの形で扱うと、ｙ軸に平行な直線を表現し難い欠点があり、また直線の正側・負側を入れ替えることができない。

表－１に示したものは、それぞれ幾何学的に一つの量を表わすつまり幾何学的要素として定義した。平面成分と立体成分とを考えるので種類が多くなっている。従来の整数・実数はスカラーと呼ぶが、一次元の幾何での座標値の意味も持っている。円、矩形、線分などは厳密には基本図形要素である。また変換マトリックスは代数的に扱うマトリックスではあるが、幾何学的には局所座標系を定義する量として原点座標と座標軸方向ベクトルから構成されている量として扱うことにしている。これらの幾何学的要素は、次節に示すような演算則を使って計算できるようになっている。

Table - 1　Data Types

No.	Type	Size by words	Command to define	Default letters
1	integer	1	DEFINT	I-N
2	real	1	DEFSNG	A-H,O-Z
4	string	2(8 bytes)	DEFSTR	
11	2D-point	2	DEF2PT	
12	2D-line	3	DEF2LN	
13	2D-circle	3	DEF2CR	
14	2D-rectangle	4	DEF2BX	
15	2D-edge	4	DEF2LS	
16	(2x3)matrix	6	DEF2TR	
21	3D-point	3	DEF3PT	
22	3D-line	6	DEF3LN	
23	3D-plane	4	DEF3PL	
24	3D-sphear	4	DEF3SP	
25	3D-cube	6	DEF3BX	
26	3D-edge	6	DEF3LS	
27	(3x4)matrix	12	DEF3TR	

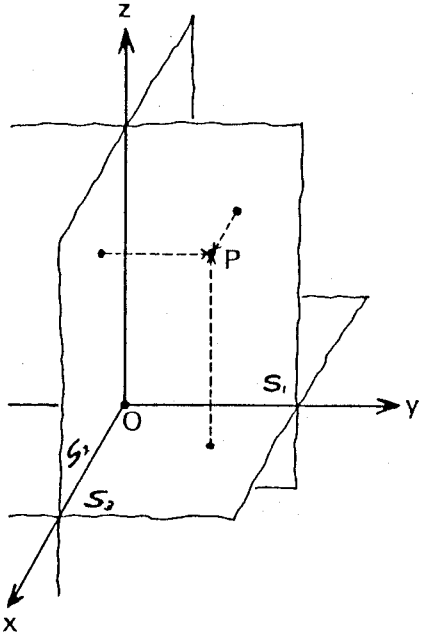


Fig.1 Coordinate System

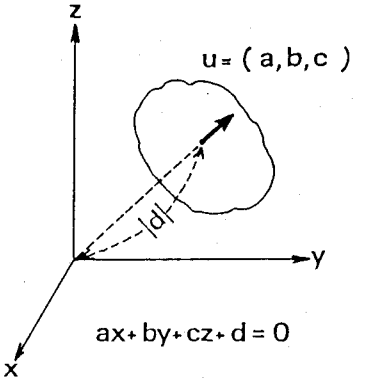


Fig.2 Definition of a plane

３．幾何学的要素間の算法

整数・実数間の算法はＦＯＲＴＲＡＮ言語などに使われるように　＋　－　＊　／　＊＊　の記号が加減乗除ならびにベキ乗に用いられる。ベクトル量として定義される点のデータは同じ型どうしで　＋　－　＊　＊＊　が用いられる。これらはベクトルとしての和、差、スカラー積、およびベクトル積が求まる。積の演算結果はデータの型が変わる。ベクトル積は二次元においてはスカラー、三次元においてはベクトルとして求められる。当然のことながら　／　の記号は定義されない。

型の異なるデータ間の代数演算は、スカラーとベクトルとが相手の時以外は原則として認めていない。そしてその演算は幾何学的な意味を持たせている。スカラー倍、スカラーでの除算は、原点を中心としてその幾何学的要素が図形として拡大、または縮小するように変換される。したがって、たとえば直線（ａ，ｂ，ｃ）の場合はｃの成分だけが変化ずる。つまり直線は原点からの距離がスカラー倍された位置に平行移動されることになる。ベクトルについては加減算だけが定められている。この幾何学的な意味は、そのベクトル方向への幾何学的要素の平行移動を行なわせる。積の記号は変換マトリックスを幾何学的要素に作用させる記号に用いる。また実際の演算は代数的なマトリックス演算とは異なっていて、相手の幾何学的要素に合わせた変換を行なうように定めてある。

幾何の処理を便利にするため、新しく２つの演算記号@と&とを定めた。@は共有（include とでも言うべきか）記号である。たとえば点ＡとＢとをＡ@Ｂと書くのは、ＡとＢとを含むような直線を作る。Ａ@ＢとＢ@Ａは向きの異なる直線である。&記号は論理積（logical ＡＮＤ）を表わす。２直線ＬＡとＬＢについてＬＡ&ＬＢは２直線の交点を求める。この記号は２つの円の交点、線分と直線の交点などのように用いることができる。三次元幾何では２平面の交差による空間直線や、空間直線の交点の計算にも用いる。演算記号として@&ならびにベクトル積記号に＊＊を用いたことに特に理由があるわけではなくて、ごく普通のコンピューターターミナルから使用できる記号の中から選んだだけである。

代入文、つまりｙ＝ｅの形は、等号をはさんだ両側の型が同じであるときには単純な代入文として働くが、型が異なるときは右側の型の性質を左側の型が取り込むようになっている。したがって常に代入文が成立するわけではない。まず左辺がスカラーであるとき、右辺の幾何学的要素の寸法や長さに関するデータが計算

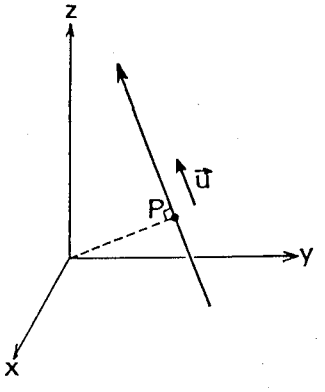


Fig.3 Defintion of a 3D-line

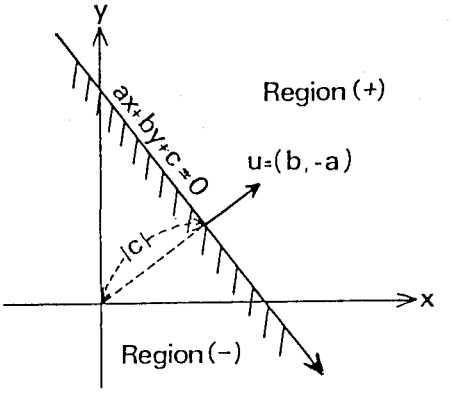


Fig.4 Defintion of a 2D-line

される。たとえば、ベクトルはベクトルの長さが得られる。直線は原点からの距離、円は半径などである。変換マトリックスはデターミナントを与える。一方、右辺がスカラーであるものはこの逆で、寸法や長さがこの値になるように左辺の幾何学的要素が変換される。

左辺がベクトルであるものは、右辺の幾何学的要素の中心位置を代入する。また、右辺にベクトルを書けば、その位置に幾何学的要素を絶対移動させる。このほかに、たとえば円＝矩形の演算は矩形の外接円を求める、という約束を定めてある。

Table-2 Assignment conditions

y= \ e	in	re	ST	A	L	R	RC	LS	MX	P	V	F	SP	BX	ED	TX
in	=	c	.	fc	pc	pc	fc	fc	fc	fc	fc	pc	pc	fc	fc	fc
re	c	=	.	f	p	p	f	f	f	f	f	p	p	f	f	f
ST	.	.	=	.	.	.	.	.	.	.	.	.	.	.	.	.
A	.	.	.	=	f	p	p	f	p	p	p	.	p	p	p	p
L	cp	p	.	f	=	.	.	f	.	.	f	.	.	.	f	.
R	cp	p	.	p	.	=	f	f	.	p	.	.	f	f	f	.
RC	.	.	.	p	.	f	=	f	.	p	.	.	f	p	f	.
LS	.	.	.	f	.	.	f	=	.	p	f	.	.	.	p	.
MX	.	.	.	p	.	.	.	.	=	p	.	.	.	.	.	p
P	.	.	.	p	.	p	p	p	p	=	f	f	p	p	f	p
V	.	.	.	.	f	.	.	p	.	f	=	.	.	.	f	.
F	cp	p	.	.	.	.	.	.	.	f	.	=	.	.	.	.
SP	cp	p	.	.	.	p	.	.	.	p	.	.	=	f	f	.
BX	.	.	.	.	.	.	.	.	.	p	.	.	f	=	f	.
ED	.	.	.	.	.	.	.	p	.	f	.	.	.	f	=	.
TX	.	.	.	p	.	.	.	.	p	p	.	.	.	.	.	=

type; in(integer), re(real), ST(string), A(2D-point), L(2D-line),
R(2D-circle), RC(2D-rectangle), LS(2D-line segment),
MX(2D-transformation matrix), P(3D-point), V(3D-line),
F(3D-plane), SP(3D-sphear), BX(3D-cube), ED(3D-line segment),
TX(3D-transformation matrix).

c ; conversion type of value(s) to real or to integer.
f ; functional calculatin is carried out then assign.
p ; part of components are assigned.
= ; simple assignment.
. ; not allowed.

4．関数

関数には、スカラーを求める関数（スカラー関数）と、ベクトルや直線のような幾何要素を求めるもの（これをベクトル関数という）の２種がある。まずスカラー関数はＡＢＳ，ＡＴＮ，ＡＴＮ２，ＣＢＲＴ，ＣＯＳ，ＤＩＳ，ＥＸＰ，ＬＯＧ，ＭＯＤ，ＲＮＤ，ＳＧＮ，ＳＩＮ，ＳＱＲＴ，ＴＡＮの14個を組み込み関数とした。これらの中でＤＩＳは距離を求める関数であり、引数として点と直線、点と平面のように使用する。

ＡＢＳは絶対値を求めるという意味を広げて、ベクトルの長さ、線分の長さ、マトリックスのデターミナントが計算できる。三角関数は角度をすべて度で扱うようになっている。そしてＳＩＮ，ＣＯＳ，ＴＡＮはベクトルや線分も引数として使用できる。逆関数ＡＴＮ，ＡＴＮ２は、ベクトルのｘ軸からの角度、もしくは２つのベクトル間の角度を度で求めるようにも定義した。

ベクトル関数として代表的なものは回転角と回転中心を与えて回転マトリックスを求めるものＭＲＯＴである。幾何の計算において２線分の交点を求めることも、１つの処理であるが、これは関数としてでなく、演算記号＆を使うことにしている。代人文や算術記号を幾何学的に定義することによって、多種類のベクトル関数を定める必要があまりない。２点間の二等分線を求める計算や、三角形の外接円を求めることは代人文、算術式ならびに回転マトリックスを使って誘導できる。しかし、使いやすさを考えて、これらもベクトル関数に加えることにした。

なお引数のない関数として、たとえば円周率πの値などをシステム常数として引用できるようにしてある。非常に幾何学的なベクトル関数として、向きの反転などに使用するＲＥＶというベクトル関数を定めた。これは、直線・線分・平面などの方向の定義を逆にするときに用いる。変換マトリックスに使用すると逆変換のマトリックスを計算する。円に用いると円の円周に沿う回転方向の定義を変えるのに使用される。この応用によって、２つの円の接線を求める算式は図－５のように一義的に定めることができる。

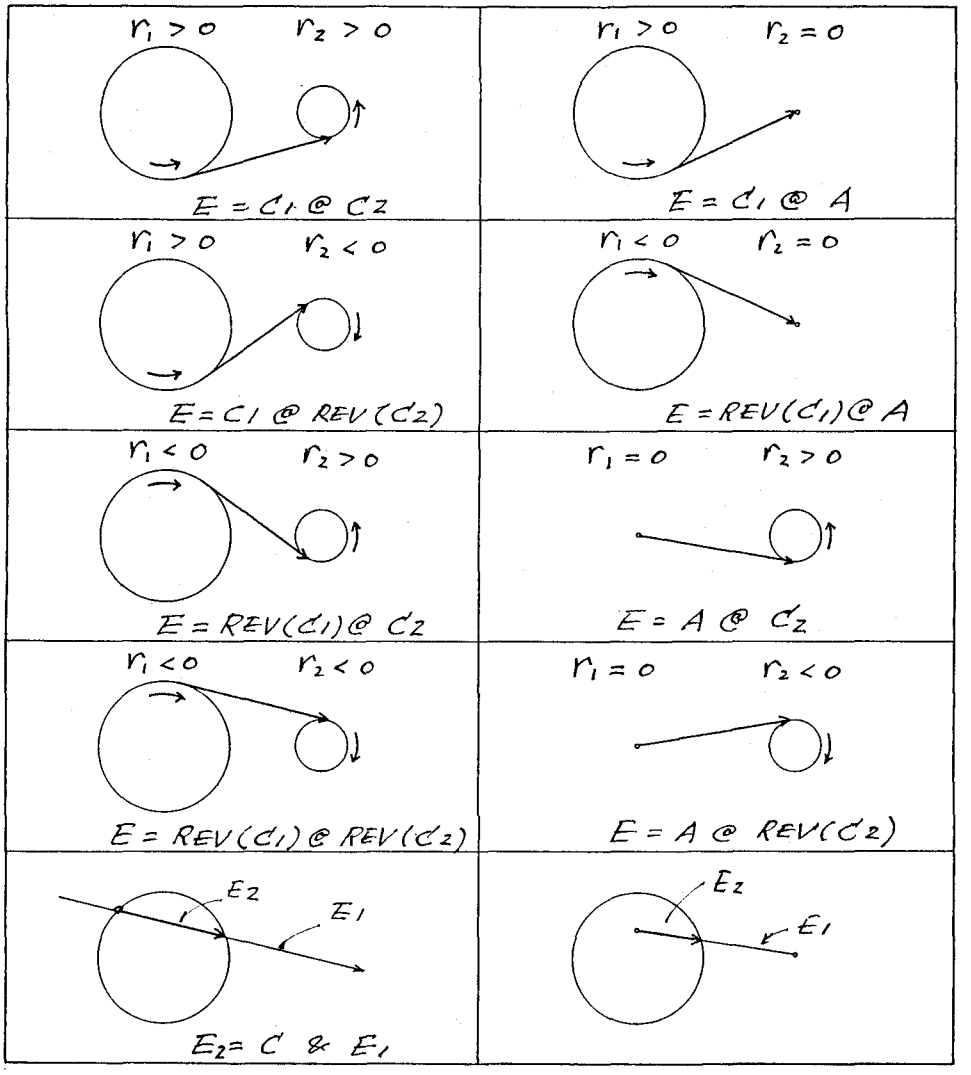
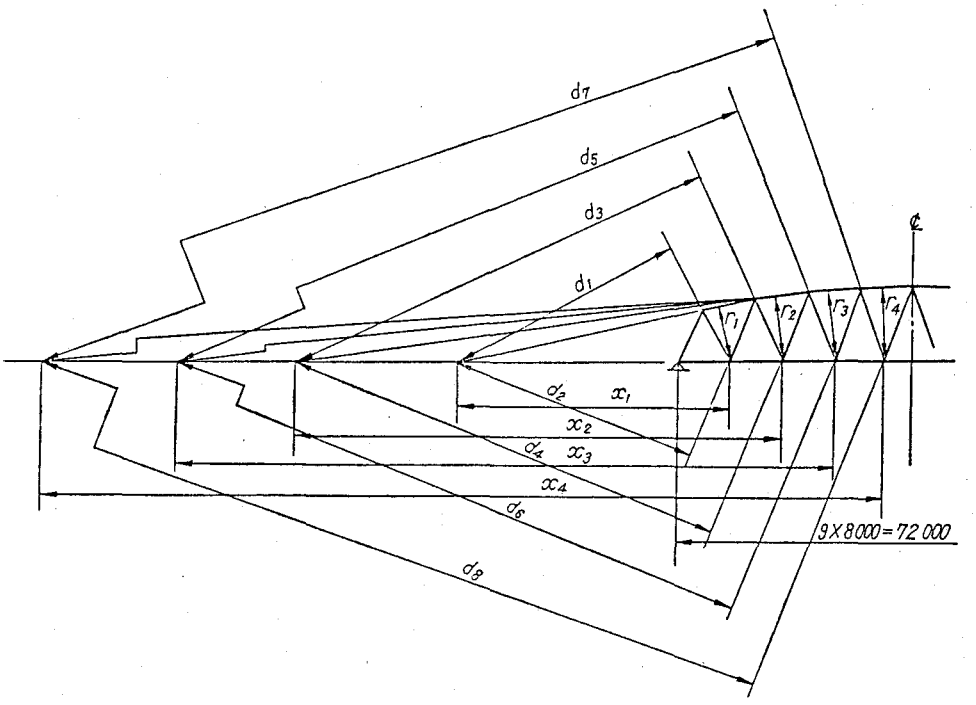


Fig. 5 Tangent lines

5．応用例

幾何言語ＧＥＯＭＥＴＲＹの応用は特に測量計算などには極めて便利である。種々の応用は今後の研究の課題であるが、図－６にトラスの解析の一例を示した。曲弦トラスの部材力の解析に切断法を応用するにあたって、２部材の延長線の交点を求め、そこから求める部材の延長線に垂線を降ろし、その垂線の長さを求めたものである。例題は参考文献によった。




```
LIST
    10 REM *** ANALYSIS OF TRUSS CHORD GEOMETRY ***
    20 DEF2PT P : DEF2LN L
    30 READ PA,PB,PC,PD,PE,PF,PG,PH,PI,PJ
    40 LBC=PB@PC : LCA=PC@PA : LBD=PB@PD : LCD=PC@PD : LDE=PD@PE
    50 LDF=PD@PF : LEF=PE@PF : LFG=PF@PG : LFH=PF@PH : LHI=PH@PI
    60 LGH=PG@PH : LIJ=PI@PJ : LHJ=PH@PJ
    70 PW=LCA&LBD : PX=LCA&LDF : PY=LCA&LFH : PZ=LCA&LHJ
    80 X1=PC-PW : X2=PE-PX : X3=PG-PY : X4=PI-PZ
    90 D1=DIS(PW,LBC) : D2=DIS(PW,LCD) : D3=DIS(PX,LDE) : D4=DIS(PX,LEF)
   100 D5=DIS(PY,LFG) : D6=DIS(PY,LGH) : D7=DIS(PZ,LHI) : D8=DIS(PZ,LIJ)
   110 R1=DIS(PC,LBD) : R2=DIS(PE,LDF) : R3=DIS(PG,LFH) : R4=DIS(PI,LHJ)
   120 PRINT "X1,X2,X3,X4=",X1,X2,X3,X4
   130 PRINT "D1,D3,D5,D7=",D1,D3,D5,D7
   140 PRINT "D2,D4,D6,D7=",D2,D4,D6,D8
   150 PRINT "R1,R2,R3,R4=",R1,R2,R3,R4
   160 DATA 0  0  4  7  8  0  12  8.3125 16 0
   170 DATA 20 9.25 24 0 28 9.8125 32 0 36 10
OK
>RUN
X1,X2,X3,X4=     46.6667       74.9333       135.556       422.667
D1,D3,D5,D7=    -40.5180      -67.5224      -124.421      -391.396
D2,D4,D6,D7=     42.0513       68.7781       125.527       392.436
R1,R2,R3,R4=    -7.55524      -8.72157      -9.50778      -9.90353
OK
>
```

参考文献

1．S.SHINADA; Computational Geometry for Structural Engineering,
IABSE proceedings pp.93-124, May,1985

2．島田静雄、熊沢周明，トラス橋の理論と設計，山海堂 1983

付録

A short Description of the Language GEOMETRY

Character Set
    Alphabetical(upper case); A B C _____ Z
    Numerical digit; 0 1 2 ___ 9
    Symbols; ■space ,comma .period " * + - / @ & ( ) = # < > ; :
    Any printable letters for string data.
    Function codes; system dependent.

Operation Phase
    Direct mode; Statements are immediately executed by CR.
    Indirect mode; Statements are stored in local program area.
    Batch mode; Statements are read from a file and executed.

Operators
    +(addition), -(subtraction), *(multiplication or product),
    /(division), **(power or vector product), @(inclusive)
    &(intersection), =(assignment or equal to), >(greater than),
    <(less than), <= or =< (less than or equal to), >= or =>(greater
    than or equal to), <> or >< (not equal)

Scalor functions
    ABS, ATN, ATN2, CBRT, COS, DIS, EXP, LOG, RND, SIN, SQRT, TAN

Vector functions
    LBSEC, MROT, PROT, REV, TROT, XROT

System constants
    ERRN, FLAG, PI, TORR

Commands
    AUTO, CONT, DECK, DELETE, END, EXIT, LIST, LOAD, MERGE, NEW, RENUM,
    RUN, SAVE

Statements
    DATA, DEFINT, DEFSNG, DEFSTR, DEF2PT, DEF2LN, DEF2CR, DEF2BX, DEF2ED
    DEF2TR, DEF3PT, DEF3LN, DEF3PL, DEF3SP, DEF3BX, DEF3ED, DEF3TR,
    DIM, ERASE, FOR, GOSUB, GOTO, IF, LET, NEXT, ON, READ, REM, RESTORE
    RETURN STOP

Input/output statements
    INPUT, PRINT, WIDTH, TRON, TROFF, LOGON, LOGOFF, ECHON, ECHOFF

Graphic controls(dependent to the device)
    GRON, GROFF, CAMERA, WINDOW, GERASE