

KeTCindy の mnr 法による日本の定理 II の 2 解法と教材化

KeTCindy センター, 日本数学教育学会名誉会員

高遠 節夫

Setsuo Takato

KeTCindy Center, H.M. of the Japan Society for Mathematics Education

1 はじめに

江戸時代に日本で独自に発展した和算には、多くの図形問題があり、各地の神社仏閣に掲額された算額にも描かれている ([1]). それらの問題には、図形的には美しいが実際に解くのは難しいものも少なからず存在する。



例えば、図 1 の右側は、上垣渉 ([3]) によって「日本の定理 II」と呼ばれている問題で、現代語に訳すと以下のようになる。

問 三角形の中に全円及び 3 線を隔てて 4 円(元, 亨, 利, 貞)を入れる。ここで全は三角形に接し、元, 利, 貞は三角形の 2 辺と 3 線に接し、亨は 3 線に接する。全径¹が 1 寸のとき元, 亨, 利, 貞の円径の和はいくらか。

答 元, 亨, 利, 貞 4 円径の和は 2 寸

術 全径を 2 倍すると 4 円の和を得て間に合う

このような図形の問題をオープンソースの数式処理システム Maxima を用いて解くことにした。しかし、座標幾何によって連立方程式を作っても、長大な式や無理式が現れるため、解を得ることはほとんど不可能である。例えば、三角形の内心を点と直線の距離により求めようとすると、次の Maxima のスクリプトを記述することになる²。

¹和算では径は直径の意味であるが、半径としても意味は同じである。

²スクリプトで、: は代入式 (通常の =) を表し ; (\$) は結果を表示する (しない) ことを意味する。

```

d1:((y2-y1)*(x-x1)-(x2-x1)*(y-y1))^2/((y2-y1)^2+(x2-x1)^2)$
d2:((y3-y1)*(x-x1)-(x3-x1)*(y-y1))^2/((y3-y1)^2+(x3-x1)^2)$
d3:((y3-y2)*(x-x2)-(x3-x2)*(y-y2))^2/((y3-y2)^2+(x3-x2)^2)$
eq1:d1-d2$
eq2:d2-d3$
ans:algsys([eq1,eq2],[x,y]);

```

しかし、Maximaで上記のスクリプトを実行しても、結果を得ることはできない。`solve`や`grobner`パッケージの`poly_buchberger`など、他の求解コマンドでも同様である。そこで、三角形の両底角 B, C の半角の正接 m, n と内接円の半径 r によって三角形の諸量を表す方法を考案して、これを mnr 法と呼ぶことにした³。

$$m = \tan \frac{B}{2}, \quad n = \tan \frac{C}{2}$$

また、 $B = (m)$, $C = (n)$ の標記を用いることにする。

このとき、三角形の諸量は m, n, r の有理式で表される。これを用いて、著者は Maxima のライブラリ `mnr.max` を作成した ([10])⁴。

2014 年、著者らは動的幾何 Cinderella2(Cindy) で対話的に作成した図から描画コードファイルを即時に出力する KeTCindy を開発した ([6])。

さらに、KeTCindy に Maxima や gcc コンパイラを呼び出す機能を追加した ([7], [8])。KeTCindy は、Cindy をインストールして、「ketcindy home」で検索されるページ

<https://s-takato.github.io/ketcindyorg/indexj.html>
からダウンロードして、`doc/ketcindysettings.cdy` を実行すればインストールできる⁵。
Maxima で `mnr.max` を読み込むには `batch` コマンドを用いる⁶。

`mnr.max` の主なコマンドと大域変数は以下の通りである。

<code>putT(m,n,r)</code>	原点を中心とする三角形をおく
<code>slideT(P,Q)</code>	点 P が点 Q に一致するように平行移動
<code>rotateT(m,P)</code>	点 P を中心に (m) だけ回転
<code>vtxT,vtxL,vtxR</code>	三角形の頂点 (図 2 の A,B,C)
<code>edgB,edgL,edgR</code>	辺の長さ (BC,AB,AC)
<code>inC,cirC,cirR,angT</code>	内心, 外心, 外接円の半径, 頂角 (mnr)

次節以降では、KeTCindy と Maxima で mnr 法を実行する方法について述べる。

³平面上の直線の傾き $\tan \theta$ を通常 m と表すことによる。

⁴通常の Maxima の拡張子 `verb.mac` を用いないのは、著者の PC が Mac であるためである。

⁵TeX の `texmf-dist/tex/latex/ketcindy` にコピーされる。

⁶KeTCindy から Maxima を呼び出すときは、コマンド列 `cmdL` の冒頭に `Mxbatch("mnr")` を入れる。

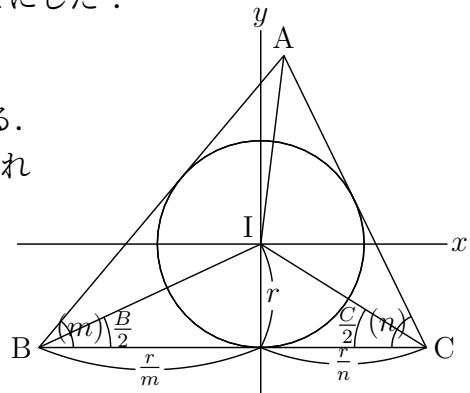


図 2. mnr 法

2 ファイルの準備と解法例

2.1 単純な問題の場合

1つのファイルにすべてのスクリプトを記述する。手順は以下の通りである。

1. KeTCindy のファイルを立ち上げて、実行するスクリプトを CindyScript の figure スロットに書く。このスロットは状態変化があった時に常に実行される。一方、KETlib スロットは、最初または右上のギヤマークを押した時のみ実行される。

The screenshot shows the KeTCindy script editor interface. The title bar says "スクリプトエディタ - exmaxima.cdy". The left sidebar shows a tree view of events: Events > Draw > figures. The main area contains the following CindyScript code:

```
1 Ketinit();
2
3 cmdL=concat(Mxbatch("mnr"), [
4   "putT(m,n,r)",
5   "A:vtxT; B:vtxL; C:vtxR",
6   "I:inC",
7   "end"
8 ]);
9 var="A::B::C::I";
10 CalcbyMset(var, "mxans", cmdL, []);
11 m=tanhalf(40); n=tanhalf(55); r=2;
12 Parsevv(var);
13 Listplot("1", [A,B,C,A]);
14 Circledata("1", [I,r]);
15
16 Windisp();
```

図 3. CindyScript の画面

- 注 1. 1 行は初期化、16 行は画面に図を描く KeTCindy のコマンドである。
- 注 2. 3 行から 8 行は Maxima で実行するコマンドのリストで、Mxbatch は mnr.max をバッチコマンドで実行する要素を作る。concat はリストの連結をする Cindy の命令である。
- 注 3. 7 行の end は、その前のコマンドにはいつもコンマをつけてよいことにするための dummy コマンドである（以下の例示では省略する場合もある）。
- 注 4. 9 行は結果として返される変数のリストである。
- 注 5. 10 行は、cmdL を実行して、mxans というファイルに書き出し、さらに各変数に文字列として割り当てる命令である。
- 注 6. 11 行の tanhalf は、角度（単位は度）の半分の正接を返す関数である。
- 注 7. 12 行の Parsevv は、各変数を評価してその数値を同じ変数に再代入する。
- 注 8. 13,14 行は、画面に描画するデータを作成するコマンドである。

2. Script 画面の右上にある実行ボタン(ギヤマーク)を押すと、Maxima が実行されて、計算結果が var の各変数に代入されるとともにスクリーンに図が表示される(図4).
3. Figure または Parent のボタンを押すことにより TeX の描画ファイルができる。

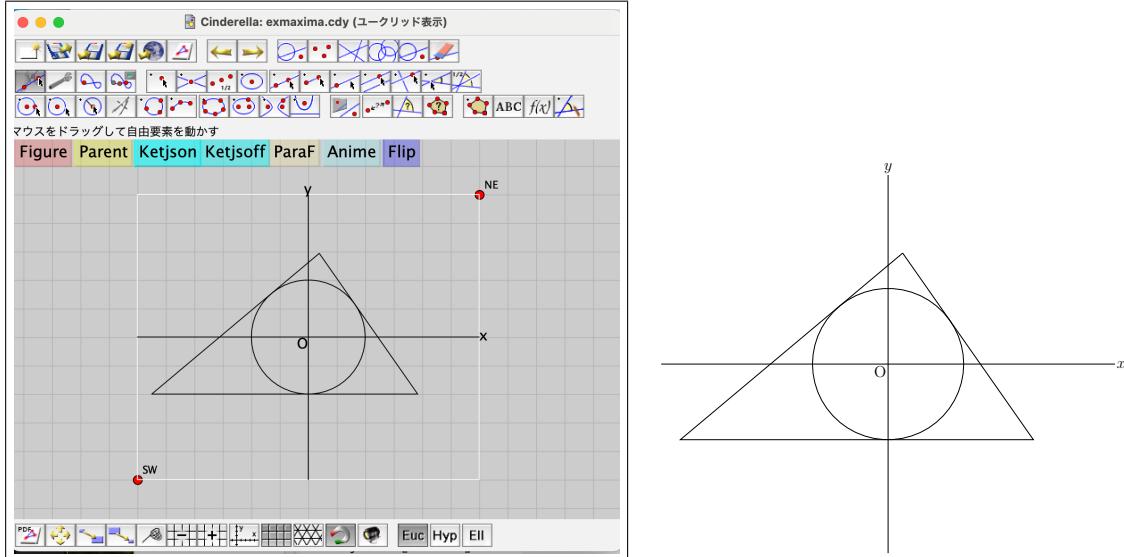


図4. Cindy の表示画面と TeX 描画

2.2 複数のステップに分ける場合

以下の手順に従う。

1. 例えば naishin.cdy を開き、KETlib に Readymnr(1,1,1) を追加して実行する。
 - (1) Cindy 画面上に実行ボタンができる。これらを上枠に移動して固定する。
 - (2) naishinfigures.txt, naishinketlib.txt をそれぞれ figure, KETlib にコピーする⁷.
 - (3) mkcmd.txt に Maxima のスクリプトを記述する。

```

8 mkcmd1():=( 
9 cmdL1=concat(Mxbatch("mnr"),[ 
10 "putT(m,n,r)", 
11 "A:vtxT; B:vtxL; C:vtxR; I:inC", 
12 "a:edgB; b:edgR; c:edgL", 
13 "P:frfactor(a*A+b*B+c*C)", 
14 "end" 
15 ]); 
16 ); 
17 var1="A::B::C::I::P"; 
18 
19 
20 
21
22 mkcmd2():=( 
23 cmdL2=concat(Mxbatch("mnr"),[ 
24 "I:[x,y]", 
25 "putT(m,n,r); slideT(inC,I)", 
26 "A:vtxT; B:vtxL; C:vtxR", 
27 "a:frfactor(edgB)", 
28 "b:frfactor(edgR)", 
29 "c:frfactor(edgL)", 
30 "Ians:frfactor((a*A+b*B+c*C)/(a+b+c))", 
31 "end" 
32 ]); 
33 ); 
34 var2="A::B::C::I::a::b::c::Ians"; 
35

```

図5. mkcmd.txt の記述(ステップ別)

⁷//import(Cdynamne() +"mkcmd.txt"); は figure と KETlib に書かれているが、どちらかだけを用いる。

2. CindyScript にステップごとの実行スクリプトを記述する.

```

10 mkcmd1();
11 if(contains(Ch,1),
12 Setmnstep(1);
13 CalcbyMset(var1,"mxans1",cmdL1,op(5));
14 DispTex(Pos,Dy,1,var1);
15 dispfigure(50,65,2);
16 );
17
19 mkcmd2();
20 if(contains(Ch,2),
21 Setmnstep(2);
22 CalcbyMset(var2,"mxans2",cmdL2,op(5));
23 DispTex(Pos,Dy,2,var2);
24 x=0.5; y=1;
25 dispfigure(50,65,2);
26 );
27

```

図 6. ステップごとの Cindy スクリプト

3. 各ステップの結果は以下の通りであり、数式は Cindy の機能によって TeX 表示される。ステップ 2 では内心公式 $I_{\text{ans}} = [x, y] = (a*A + b*B + c*C) / (a + b + c)$ が得られる。

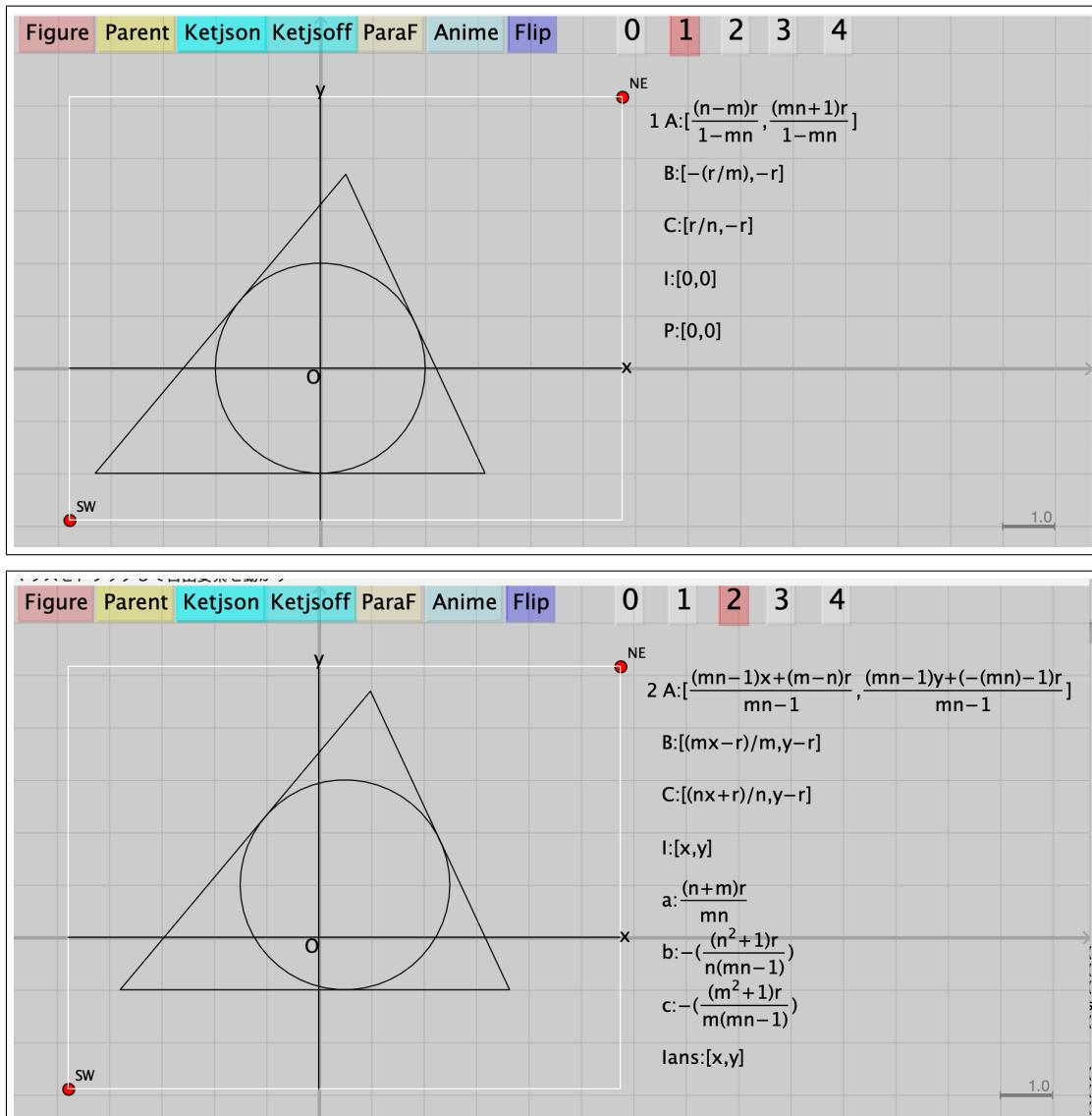


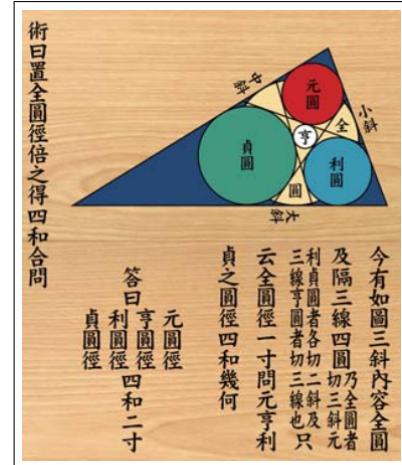
図 7. ステップ 1, 2 の結果

3 日本の定理 II (1) 中からの証明

亨を内接円とする三角形から始めると、元利貞の3円はその傍接円であり、全円はそれらの共通外接線からできる三角形の内接円である。
傍接円などはすべて mnr.max で定義されているため、
方程式は最後に現れるだけで、求解は容易である。

```
mkcmd1():=(
    cmdL1=concat(Mxbatch("mnr"), [
        "putT(m,n,r)",
        "A:vtxT; B:vtxL; C:vtxR; I:inC; r:inR",
        "Ca:exCa; Ra:exRa; Cb:exCb", //exCa などは傍心
        "Rb:exRb; Cc:exCc; Rc:exRc", //傍接円の半径
        "eq4a:contCL(I3,r3,D,E); eq4b:contCL(I3,r3,K,F)",
        //I3 は DE, KF に接する。結果の式は因数分解される
        "eq4a1:nthfactor(eq4a,6)",
        //因数分解された式の 6 番目の要素をとる
        "eq4a2:nthfactor(eq4a,7)",
        "eq4b1:nthfactor(eq4b,6); eq4b2:nthfactor(eq4b,7)",
        "rd11:reduceD([eq4a1,eq4b1],n1,10)", //m1,n1 の連立方程式から n1 を消去
        "rd12:reduceD([eq4a1,eq4b2],n1,10); rd21:reduceD([eq4a2,eq4b1],n1,10)",
        "rd22:reduceD([eq4a2,eq4b2],n1,10)",
        "rd11mm:nthfactor(rd11[2],6); rd12mm:nthfactor(rd12[2],5)",
        "rd21mm:nthfactor(rd21[2],4); rd22mm:nthfactor(rd22[2],6)",
        "rdn:numer(remainder(eq4b2,rd12mm)); rdn:nthfactor(rdn,2)",
        "solm:solve(rd12mm,m1); soln:solve(rdn,n1)",
        "ansm1:frev(m1,solm[2]); ansn1:frev(n1,soln[2])",
        "P1:meetLine([K,F],[H,L]); P2:meetLine([D,E],[H,L])",
        "P3:meetLine([D,E],[K,F])",
        "putT(plusA(m1,n2),minusA(n1,n2),r0)",
        "slideT(vtxL,P1); rotateT(supA(n2),P1)",
        "eq:vtxR[1]-P2[1]; sol:solve(eq,r0)",
        "fe:frevL([r0,inC,vtxT,vtxL,vtxR],sol)",
        "r0:fe[1]; I0:fe[2]; Q1:fe[3]; Q2:fe[4]; Q3:fe[5]", "P3:Q1"
    ]);
)
```

```
mkcmd2():=(
    cmdL2=concat(cmdL1, [ //cmdL1 と以下を合わせたリストを作成(concat は Cindy コマンド)
        "out:comTan1(Ca,Ra,Cb,Rb)", //comTan1 は共通外接線(上の 2 点を求める)
        "pts1:[out[1],out[2]]", //共通外接線上の 2 点からできる線分
        "out:comTan1(Cc,Rc,Ca,Ra); pts2:[out[1],out[2]]",
        "out:comTan1(Cb,Rb,Cc,Rc); pts3:[out[1],out[2]]",
    ])
```



```

"D:frfactor(meetLine(pts1,pts2))", //2線分の交点を求めて分数式を簡略化
"E:frfactor(meetLine(pts2,pts3)); F:frfactor(meetLine(pts3,pts1))",
"end"
]);
);
mkcmd3():=
cmdL3=concat(cmdL2,[

"assume(m>0,n>0,m<1,n<1,m*n-1<0); assume((m+1)*n+m-1>0)",
"DE:edge(D,E); EF:edge(E,F); FD:edge(F,D)", //2点を結ぶ線分の長さ
"mD:edg2m(EF,DE,FD)", //3辺の長さからmを求める
"mE:edg2m(FD,EF,DE); mF:edg2m(DE,FD,EF)",
"mrotD:cos2m((D[1]-F[1])/FD)", //cosの値からmを求める
"putT(mF,mD,r1); slideT(vtxR,D); rotateT(-mrotD,D)",
"eq:vtxT[1]-E[1]; sol:solve(eq,r1)", //頂点がEと一致するようにr1を決める
"R:frev(r1,sol); CR:frev(inC,sol)", //解をR,CRに代入
"r0:r; sum:frfactor(r+Ra+Rb+Rc)", //4円の和を簡単化
"end"
]);
);

```

図9はステップ3を実行したときの画面で、算額にある術の等式 $sum = 2R$ が得られる。

$$R = \frac{(m^2 n^2 - n^2 - 2mn - m^2 - 1)r}{2mn(mn - 1)}$$

$$r + Ra + Rb + Rc = \frac{(m^2 n^2 - n^2 - 2mn - m^2 - 1)r}{mn(mn - 1)} = 2R$$

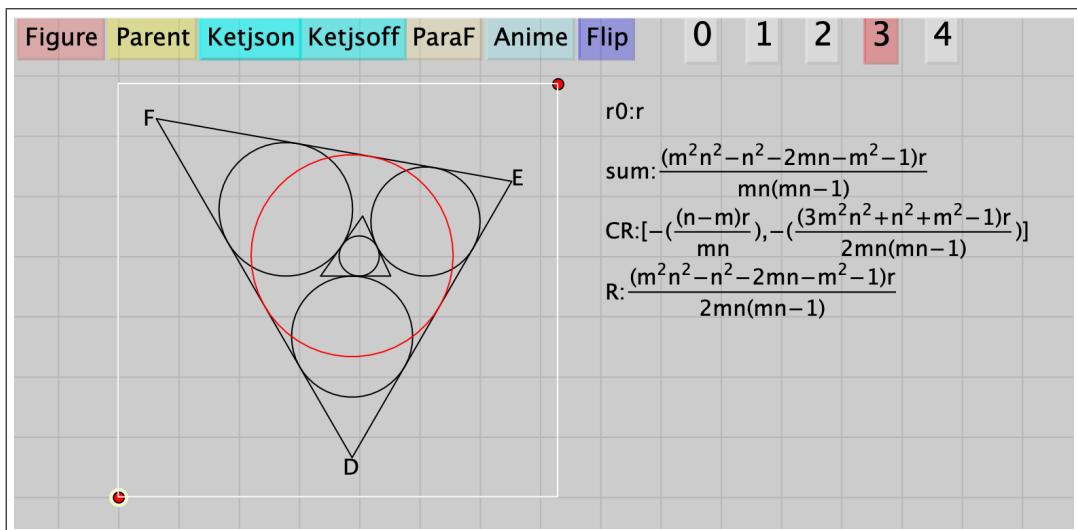


図9. 中からの証明 ($sum = 2R$)

4 日本の定理II (2) 外からの証明

算額の間によれば、まず外の三角形と内接円を描いてから、三角形の中に3線分を描き、三角形の辺と3線分に接する元、利、貞と3線分だけに接する亭をとっている。本節ではこれに沿った解法を考えることにする。

Step 1. 図10の $m, n, m1, n1, n2$ をおいて、DBE, KF, DE に接する円 I1 を求める。

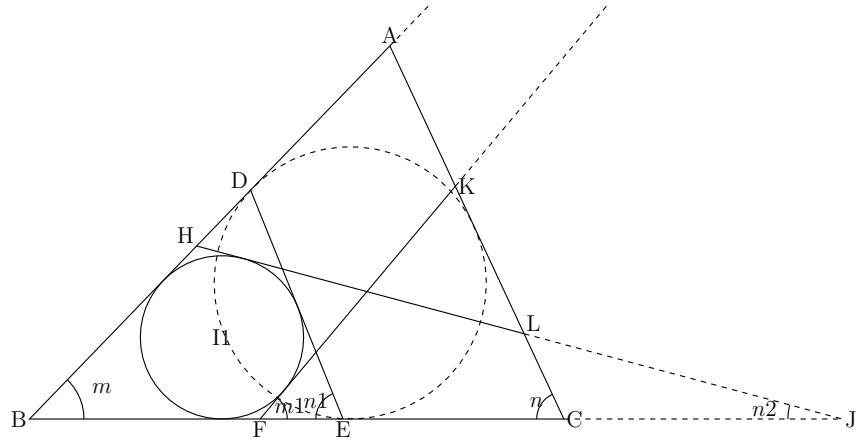


図10. ステップ1の図

```

cmdL1=concat(Mxbatch("mnr"), [ //mnr ライブライを読み込む
"putT(m,n,R)", //内心を原点とする三角形をおく
"A:frfactor(vtxT); B:frfactor(vtxL); C:frfactor(vtxR)",
"I:frfactor(inC)", //頂点と内心の分数式を簡単化
"putT(m,n1,r1); slideT(vtxL,B)", //三角形 DBE をおく
"D:vtxT; E:vtxR; I1:inC",
"putT(m,supA(m1),r1); slideT(vtxL,B)", //supA は補角
"G:vtxT; F:vtxR", //G は三角形 GBF の頂点 (直線 BD, FK の交点)
"putT(m,n2,r1); slideT(vtxL,B); H:vtxT; J:vtxR", //三角形 HBJ をおく
"K:meetLine([F,G],[A,C]); L:meetLine([H,J],[A,C])" //線分の交点
]);

```

Step 2. KFC, DE, HL に接する円 I2 を求める。

```

cmdL2=concat(cmdL1, [ //cmdL1 と連結
"putT(m1,n,r2); slideT(vtxR,C); I2a:frfactor(inC)", //三角形 KFC
"eq1:numer(vtxL[1]-F[1])", //左頂点が F となる条件
"putT(supA(n1),n,r2); slideT(vtxR,C); I2b:frfactor(inC)", //三角形*EC
"eq2:numer(vtxL[1]-E[1])", //左頂点が E となる条件
"sol12:solve([eq1,eq2],[r1,r2])", //eq1, eq2 を解く
"fe:frevL([r1,r2,D,E,I1,G,F,H,J,K,I2a],sol12)", //解を各変数に代入
"r1:fe[1]; r2:fe[2]; D:fe[3]; E:fe[4]; I1:fe[5]",
"G:fe[6]; F:fe[7]; H:fe[8]; J:fe[9]; K:fe[10]; I2:fe[11]",
"putT(m1,n2,r2); slideT(vtxL,F); I2c:frfactor(inC)", //三角形*FJ
];

```

```
"eq2b:numer(vtxR[1]-J[1]); sol2b:solve(eq2b,n2)", //右頂点が J の条件
"fe:freval([n2,H,L,J],sol2b); n2:fe[1]; H:fe[2]; L:fe[3]; J:fe[4]"
]);
```

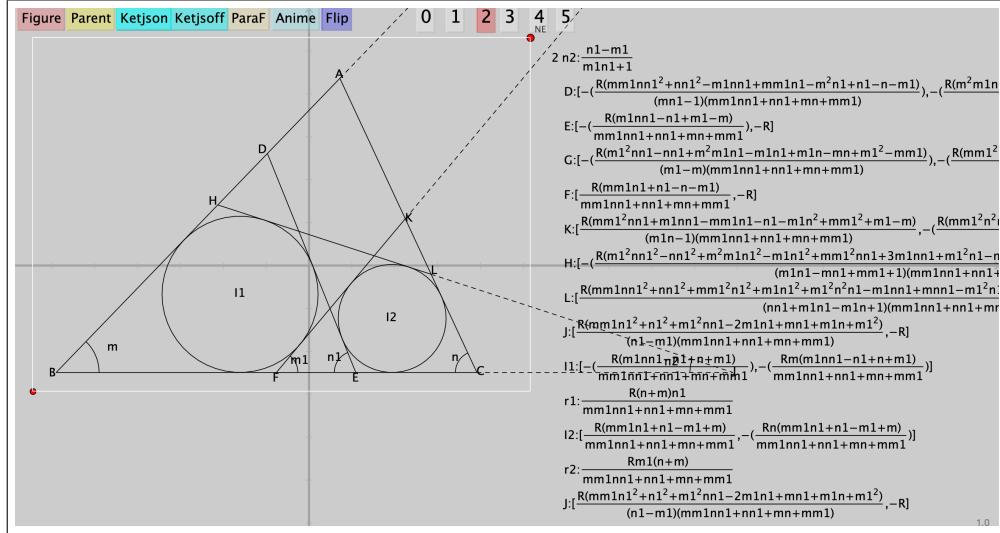


図 11. ステップ 2 の実行結果

Step 3. $\angle AHL = m + n2$, $\angle ALH = n - n2$ に注意して AHL の内接円 I_3 をおく.

```
cmdL3=concat(cmdL2, [ //cmdL2 と連結
"putT(plusA(m,n2),minusA(n,n2),r3)", //plusA,minusA は角度の和と差
"slideT(vtxT,A); rotateT(-n2,A)", //Aを中心回転
"H3:frfactor(vtxL); L3:frfactor(vtxR)", //左右の頂点を H3,L3 とおく
"eq3:numer(H3[1]-H[1]); sol3:solve(eq3,r3)", //H3$=$H の方程式を解く
"fe:freval([H3,L3,inC,r3],sol3)", //sol3 を代入
"H:fe[1]; L:fe[2]; I3:fe[3]; r3:fe[4]"
]);
```

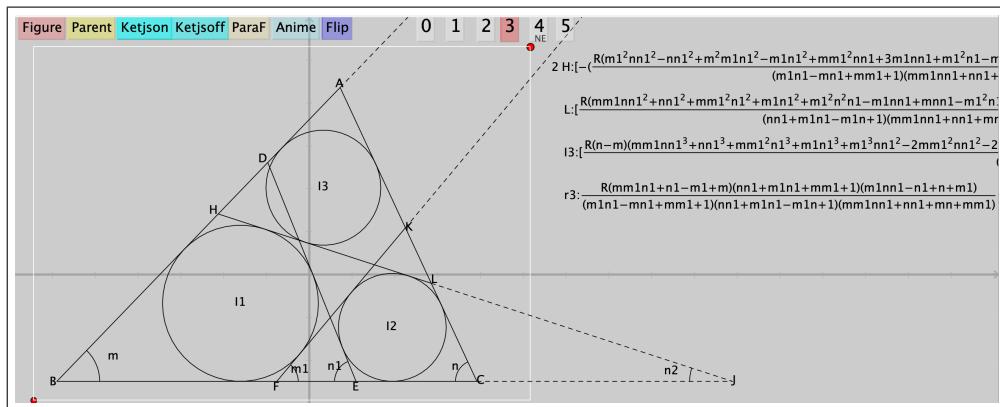


図 12. ステップ 3 の実行結果

Step 4. I_3 が DE, KF に接するように $m1, n1$ を定める.

```

cmdL4=concat(cmdL3,[  

"eq4a:contCL(I3,r3,D,E); eq4b:contCL(I3,r3,K,F)", //I3 は DE, KF に接する  

"eq4a1:nthfactor(eq4a,6)", //因数分解された式の 1 つの要素をとる  

"eq4a2:nthfactor(eq4a,7)",  

"eq4b1:nthfactor(eq4b,6); eq4b2:nthfactor(eq4b,7)",  

"rd11:reduceD([eq4a1,eq4b1],n1,10)", //m1,n1 の連立方程式から n1 を消去  

"rd12:reduceD([eq4a1,eq4b2],n1,10)",  

"rd21:reduceD([eq4a2,eq4b1],n1,10)",  

"rd22:reduceD([eq4a2,eq4b2],n1,10)",  

"rd11mm:nthfactor(rd11[2],6); rd12mm:nthfactor(rd12[2],5)",  

"rd21mm:nthfactor(rd21[2],4); rd22mm:nthfactor(rd22[2],6)",  

"rdn:numer(remainder(eq4b2,rd12mm)); dn:nthfactor(rdn,2)",  

"solm:solve(rd12mm,m1); soln:solve(rdn,n1)",  

"ansm1:frev(m1,solm[2]); ansn1:frev(n1,soln[2])",  

"P1:meetLine([K,F],[H,L]); P2:meetLine([D,E],[H,L])",  

"P3:meetLine([D,E],[K,F])",  

"putT(plusA(m1,n2),minusA(n1,n2),r0)",  

"slideT(vtxL,P1); rotateT(supA(n2),P1)",  

"eq:vtxR[1]-P2[1]; sol: solve(eq,r0)",  

"fe:frevL([r0,inC,vtxT,vtxL,vtxR],sol)",  

"r0:fe[1]; I0:fe[2]; Q1:fe[3]; Q2:fe[4]; Q3:fe[5]",  

"P3:Q1"  

]);

```

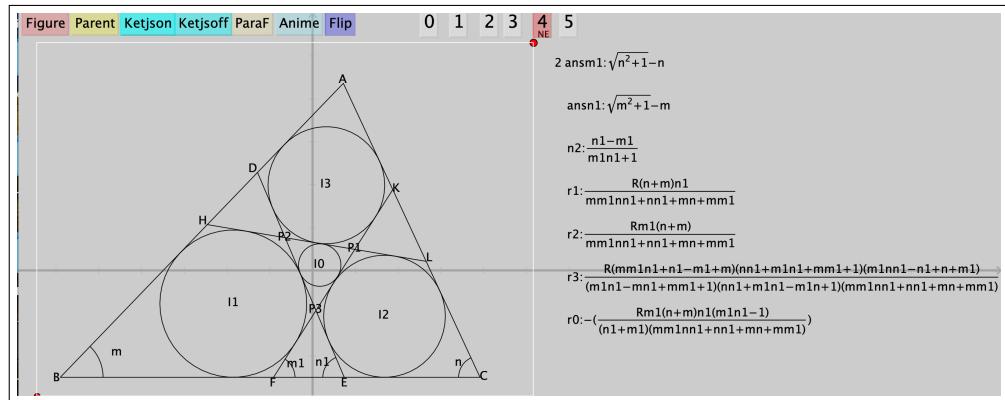


図 13. ステップ 4 の実行結果

Step 5. ステップ 4 の結果を用いれば、4円の半径 r_0, r_1, r_2, r_3 を求めて作図することはできるが、 $r_1 + r_2 + r_3 + r_4 = 2R$ を証明するためには、図 13 の右側の結果にある $m_1 = \sqrt{n^2 + 1} - n$, $n_1 = \sqrt{m^2 + 1} - m$ を代入する必要がある。しかし、これらは無理式のため、このまま代入した式を Maxima で簡単化すること

ができない。そこで、次の式変形を用いた置き換えをする。

$$\begin{aligned}\tan 2\theta &= \frac{2 \sin \theta \cos \theta}{\cos^2 \theta - \sin^2 \theta} = \frac{2 \tan \theta}{1 - \tan^2 \theta} \\ \sqrt{1 + \tan^2 2\theta} &= \frac{1}{\cos 2\theta} = \frac{1 + \tan^2 \theta}{1 - \tan^2 \theta} \\ \sqrt{1 + \tan^2 2\theta} - \tan 2\theta &= \frac{(1 - \tan \theta)^2}{1 - \tan^2 \theta} = \frac{1 - \tan \theta}{1 + \tan \theta}\end{aligned}$$

$m = \tan \frac{B}{2} = \tan 2\theta, M = \tan \frac{B}{4} = \tan \theta$ とおくと

$$\begin{aligned}m &= \frac{2M}{1 - M^2} \\ \sqrt{m^2 + 1} - m &= \frac{1 - M}{1 + M}\end{aligned}$$

M は B の 4 半角であり、この置き換えにより上の式は M の有理式で表される。 n, N についても同様である。

これを用いて以下のスクリプトを実行すると、 $r0 + r1 + r2 + r3 = 2R$ が得られる。

```
cmdL5=concat(cmdL4,[  
"m1:quasubs(ansm1,n,N)",  
    //ansm1 の n に N の式を代入  
"n1:quasubs(ansn1,m,M)",  
"m:quasubs(m,m,M)",  
"n:quasubs(n,n,N)",  
"sbs:[m=m,n=n,m1=m1,n1=n1]",  
"fe:frevL([n2,r0,r1,r2,r3],sbs)",  
"n2:fe[1]; r0:fe[2]; r1:fe[3];  
"r2:fe[4]; r3:fe[5]",  
"S:frfactor(r0+r1+r2+r3)"  
]);
```

```
5 m1:-(N-1)/(N+1)  
n1:-(M-1)/(M+1)  
m:-(2M/(M-1)(M+1))  
n:-(2N/(N-1)(N+1))  
n2:(N-M)/(MN+1)  
r0:(M-1)(N-1)(N+M)R/(M^2N^2+MN^2+M^2N-2MN+N+M+1)  
r1:(M-1)(N+1)(MN-1)R/(M^2N^2+MN^2+M^2N-2MN+N+M+1)  
r2:(M+1)(N-1)(MN-1)R/(M^2N^2+MN^2+M^2N-2MN+N+M+1)  
r3:(M+1)(N+1)(N+M)R/(M^2N^2+MN^2+M^2N-2MN+N+M+1)  
S:2R
```

図 14. 4 円の和 $S = 2R$

5 まとめと今後の課題

和算家たちは、算額にあるような図形問題を、幾何学で知られている多くの定理を用いることなしに彼らの卓越した計算力で解いていた。文献 [9] によれば、1911 年発行の Mathesis には、日本の定理 II の問題と当時の数学者による証明についての林鶴一 ([2]) の報告があり、それらの証明では 9 点円の定理やフォイエルバッハ定理などが用いられ

ていたという。しかし、これらの定理は、解析幾何の初步や三角関数だけを学んだ学生のほとんどは知らず、和算の問題を理解したとしても、自らの力だけでは解くことができないであろう。これに対して、mnr 法は、ある程度の数学の知識と Maxima のスクリプトの記述さえできれば、通常の学生でも扱える方法であり、実際に問題を解くことで、和算に対する興味をより高めることが期待される。時には 4 半角を用いるなどの工夫が必要であり、それらを考案することも、数学の応用力を高めて興味を倍加することにつながると考えられる。ただし、多くの学生にとって、Maxima のプログラミングは慣れていますとはいはず、代入は : (コロン), 式の区切りは ; (セミコロン), block の区切りは , (コンマ)などを間違えて、エラーで動かなくなることもしばしば発生する。そのため、mnr 法による記述についての文法チェックプログラムを開発することも必要であり、今後の課題となっている。

参考文献

- [1] 和算の館 <http://www.wasan.jp>
- [2] 林鶴一, Un Théorème Japonais, Mathesis 4, 1, p.208-209, 1911
- [3] 上垣涉, Japanese Theorem の起源と歴史, 三重大学教育学部研究紀要. 自然科学 / 三重大学教育学部編 52, p.23-45, 2001
- [4] H. Fukagawa, Tony Rothman, Sacred Mathematics: Japanese Temple Geometry, Princeton University Press, 2008
- [5] 涌田和芳, 外川一仁, 新潟白山神社の紛失算額, 長岡高専研究紀要 47巻, p.7-16, 2011
- [6] 高遠節夫, KeTCindy 開発チーム, KeTCindy の開発について, RIMS 講究録 1978, p.173-182, 2015
- [7] 小林茂樹, 高遠節夫, KeTCindy と Maxima,Risa/Asir との連携, RIMS 講究録 2022, p.128-134, 2017
- [8] 野田健夫, 高遠節夫, KeTCindy の C呼び出し機能と曲線・曲面論の教材の作成, RIMS 講究録 2067, p.132-141, 2018
- [9] 涌田和芳, Mathesis に掲載された新潟白山神社の算額の証明, 和算ジャーナル 6号, 会報通巻 56, 群馬県和算研究会, 2022
- [10] 高遠節夫, 北本卓也, Maxima と KeTCindy による和算問題の解法と教材化, RIMS 講究録 2320, p.1-10, 2025