A Generalized Apollonius Problem

Greg Fee

Abstract

The Apollonius problem is: Find all circles that touch 3 given circles, in the Euclidean plane.
In general there are 8 such tangent circles.
We generalize this problem to: Find all circles that touch 3 given ellipses in the Euclidean plane.
We conjecture that there are at most 68 such circles.

Apollonius Problem

The Apollonius problem is described in [1] and [2], which are both on the internet.
Maple's geometry[Apollonius] can solve the Apollonius problem, but we illustrate the solution as follows.

An example of an Apollonius problem is:

Given: the following 3 circles in equation form.

```
> cL[1] := [14,22,13]:
  cL[2] := [97,19,11]:
  cL[3] := [53,83,17]:
```

```
> for i to 3 do
   g_cir[i] := (x-cL[i][1])^2 + (y-cL[i][2])^2 - cL[i][3]^2;
  od;
```

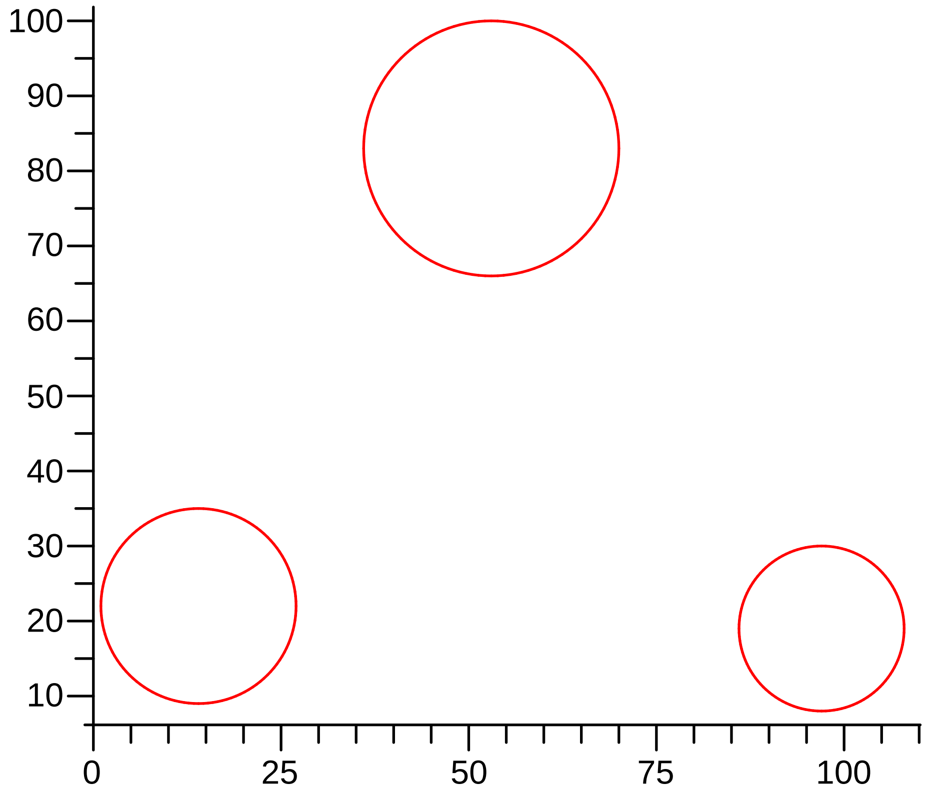
$$g\_cir_1 := (x-14)^2 + (y-22)^2 - 169 \tag{2.1}$$

$$g\_cir_2 := (x-97)^2 + (y-19)^2 - 121$$

$$g\_cir_3 := (x-53)^2 + (y-83)^2 - 289$$

```
> for i to 3 do
   pl_G[i] := plot_cir(cL[i],129,COLOUR(RGB,1,0,0));
  od:
```

```
> plots[display]({seq(pl_G[i],i=1..3)},scaling=constrained);
```

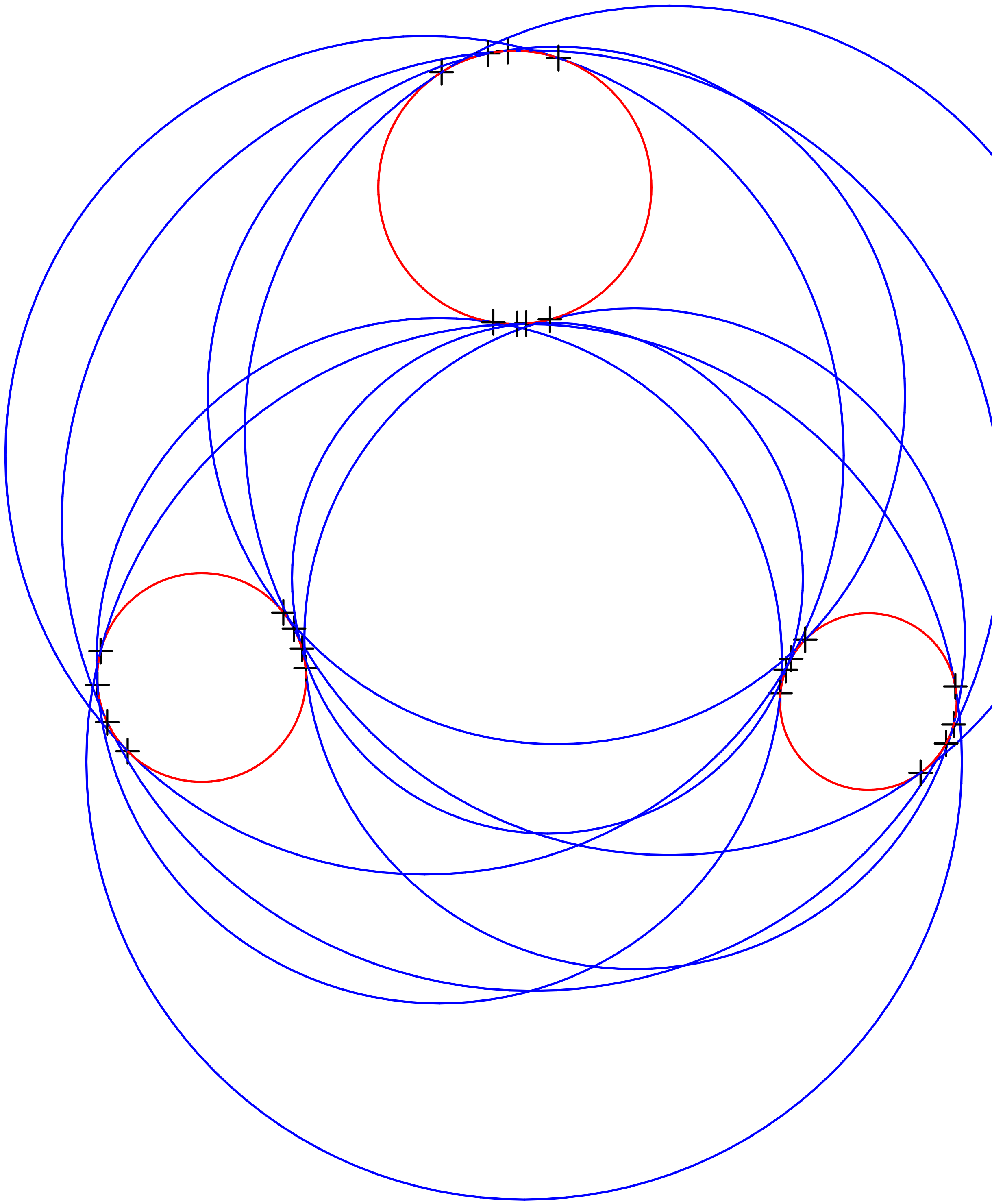


```
> make_tangent_cir_eq := proc(eq)
   local cir,rs,di,co,abr;
   cir := (x-a)^2+(y-b)^2-r^2;
   rs := resultant(eq,cir,y);
   di := discrim(rs,x);
   co := content(di,a);
   divide(di,co,'abr');
   abr*sign(abr);
  end:
> for i to 3 do
   Aeq[i] := make_tangent_cir_eq(g_cir[i]);
  od:
> gs:=Groebner[gsolve]([Aeq[1],Aeq[2],Aeq[3]],[a,b,r]):
> ng := nops(gs);
```

$$ng := 8 \quad (2.2)$$

```
> for i to 8 do
   r_cir[i] := solve(gs[i][1][1])[1];
   a_cir[i] := solve(subs(r=r_cir[i],gs[i][1][3]),a);
   b_cir[i] := solve(subs(r=r_cir[i],gs[i][1][2]),b);
   A_cir[i] := (x-a_cir[i])^2+(y-b_cir[i])^2-r_cir[i]^2;
  od:
> TPc := Matrix(8,3):
> for i to 8 do
   for j to 3 do
    TPc[i,j] := find_tangent_point_cir(A_cir[i],g_cir[j]);
   od;
  od;
> for i to 8 do
   pl_c[i] := plot_cir([a_cir[i],b_cir[i],r_cir[i]],
    129,COLOUR(RGB,0,0,1));
  od:
> for i to 8 do
   TPpl[i] := PLOT(POINTS(seq(evalf(TPc[i,j]),j=1..3)),
   COLOUR(RGB,0,1,0),SYMBOL(CROSS));
  od:
```

```
> plots[display]({seq(pl_G[i],i=1..3),
  seq(pl_c[i],i=1..8),seq(TPpl[i],i=1..8)},
  scaling=constrained,axes=none);
```



ellipse procedures

```
> make_tangent_eq := proc(eq)
   local cir,res_y,con_x,dis_x,g;
   cir := (x-a)^2+(y-b)^2-r2;
   res_y := resultant(eq,cir,y);
   con_x := content(res_y,x);
   divide(res_y,con_x,evaln(res_y));
   dis_x := discrim(res_y,x);
   dis_x := dis_x/icontent(dis_x);
   g := gcd(dis_x,diff(dis_x,a));
   divide(dis_x,g^2,evaln(dis_x));
   sign(dis_x)*dis_x;
  end:
```

find_tangent_point

```
> find_tangent_point := proc(eq1,eq2)
   local r,r1,fs1,n,fv,mf,k,x1,yv1,yv2,md,i,j,td,ansy;
   r := resultant(eq1,eq2,y);
   r1 := diff(r,x);
   fs1 := [fsolve(r1,x)]:
   n := nops(fs1);
   fv := [seq(abs(subs(x=fs1[i],r)),i=1..n)];
   mf := min(op(fv));
   member(mf,fv,evaln(k));
   x1 := fs1[k];
   yv1 := [fsolve(subs(x=x1,eq1),y)];
   yv2 := [fsolve(subs(x=x1,eq2),y)];
   md := infinity;
   for i in yv1 do
    for j in yv2 do
     td := abs(i-j);
     if td<md then md := td; ansy := .5*(i+j); fi;
    od;
   od;
   [x1,ansy];
  end:
```

circles touching 3 ellipses

```
> Aep[1] := [4.81,1.09,0.08,-0.01,0.041]:
  Aep[2] := [5.1,1.07,-0.06,0.021,1.]:
  Aep[3] := [4.92,1.03,0.04,0.031,-.981]:
> ell[1] := round_c(expand(642*ell_par_eq(op(Aep[1]))));
  ell[2] := round_c(expand(737*ell_par_eq(op(Aep[2]))));
  ell[3] := round_c(expand(579*ell_par_eq(op(Aep[3]))));
```

$$ell_1 := 29\,x^2 - 5\,x - 42\,x\,y - 642 + 14\,y + 539\,y^2 \quad (4.1)$$

$$ell_2 := 464\,x^2 + 67\,x - 560\,x\,y - 735 - 42\,y + 208\,y^2$$

$$ell_3 := 384\,x^2 - 46\,x + 482\,x\,y - 578 - 31\,y + 185\,y^2$$

```
> ep[1] := ell_eq_par(ell[1]):
  ep[2] := ell_eq_par(ell[2]):
  ep[3] := ell_eq_par(ell[3]):
> pe[1] := plot_ell_par(ep[1]):
  pe[2] := plot_ell_par(ep[2]):
  pe[3] := plot_ell_par(ep[3]):
> plots[display]({pe[1],pe[2],pe[3]}):
> eq[1] := make_tangent_eq(ell[1]):
  eq[2] := make_tangent_eq(ell[2]):
  eq[3] := make_tangent_eq(ell[3]):
> eqs := {eq[1],eq[2],eq[3]}:
> eqL := [eq[1],eq[2],eq[3]]:
> save eqL,"/home/gfee/Apollonius/eqL";
> deg[a]=degree(eq[1],a),deg[b]=degree(eq[1],b),
  deg[r2]=degree(eq[1],r2),total_deg=degree(eq[1]);
```

$$deg_a = 8,\ deg_b = 8,\ deg_{r2} = 4,\ total\_deg = 8 \quad (4.2)$$

```
> number_of_terms=nops(eq[1]);
```

$$number\_of\_terms = 95 \quad (4.3)$$

```
> number_of_digits=map(length@maxnorm,eqL);
```

$$number\_of\_digits = [19, 19, 18] \quad (4.4)$$

search for floating-point solutions

```
> fss := {}:
> for i from -40 by 5 to 40 do
   for j from -40 by 5 to 40 do
    for k from 1 by 5 to 61 do
     fs := fsolve(eqs,{a=i/10,b=j/10,r2=k^2/100});
     fss := fss union {fs};
    od;
   od;
  od:
> nc := nops(fss);
```

$$nc := 68 \qquad (4.1.1)$$

```
> number_of_solutions_found=nc;
```

$$number\_of\_solutions\_found = 68 \qquad (4.1.2)$$

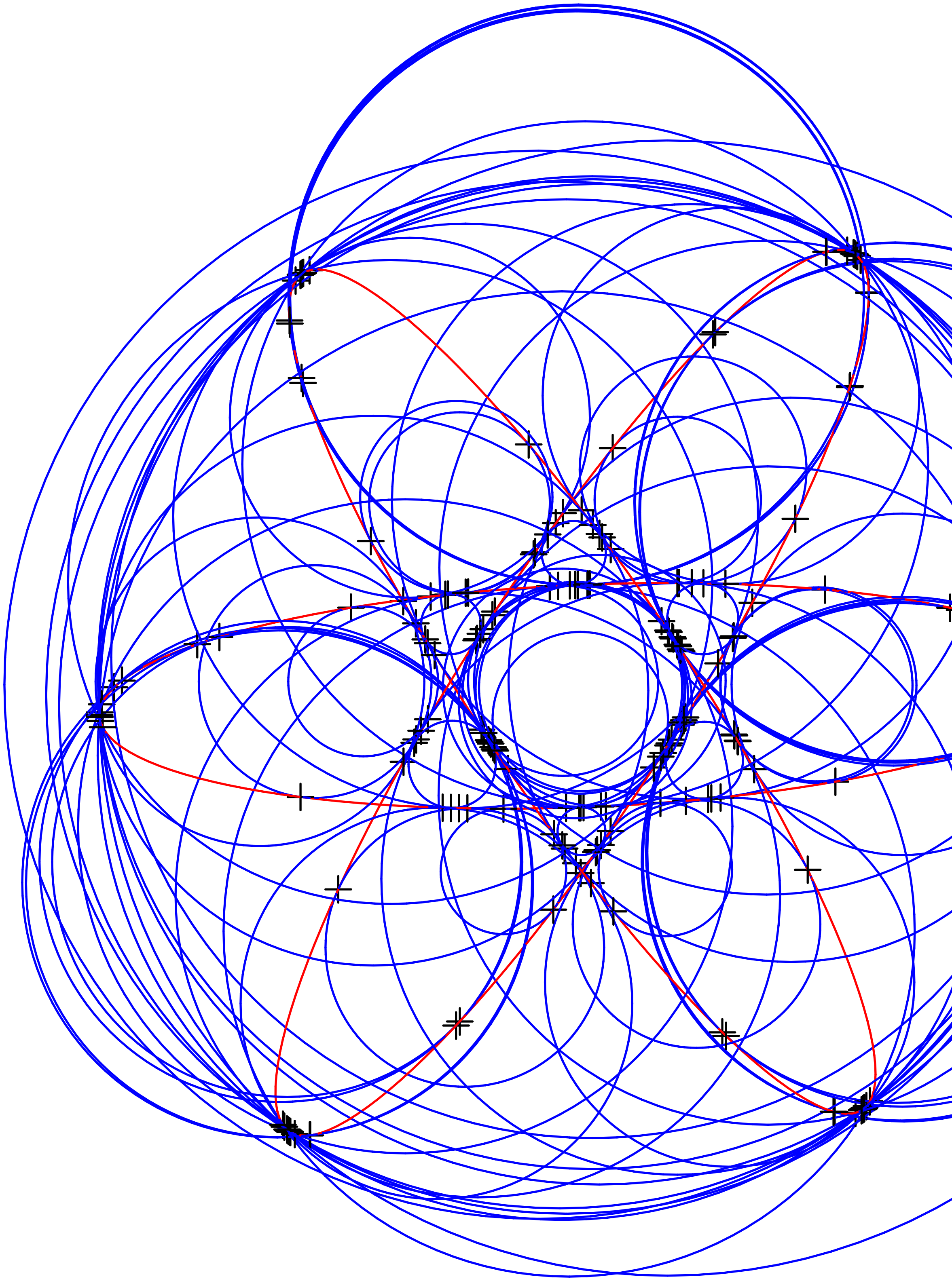
floating point solutions

```
> fss := fss:
> fsl := [op(fss)]:
> fsl := sort(fsl,proc(c,d) evalb(subs(c,r2)<subs(d,r2)) end):
> cir := (x-a)^2+(y-b)^2-r2:
> TP := Matrix(nc,3):
> for i to nc do
   for j to 3 do
    TP[i,j] := find_tangent_point(subs(fsl[i],cir),ell[j]);
   od;
  od:
>
> for i to nc do
   pc[i]:=plot_cir(subs(fsl[i],[a,b,sqrt(r2)]));
  od:
>
> colE := COLOUR(RGB,1.,0.,0.):
  colC := COLOUR(RGB,0.,0.,1.):
  colP := COLOUR(RGB,0.,1.,0.):
> sym_pt := SYMBOL(CROSS):
> for i to nc do
   plpt[i] := POINTS(seq(TP[i,j],j=1..3),colP,sym_pt);
  od:
```

Create plot

```
> fr[0] := [TITLE("3 given ellipses"),
   CURVES(op([1,1],pe[1]),colE),
   CURVES(op([1,1],pe[2]),colE),
   CURVES(op([1,1],pe[3]),colE),
  NULL]:
  for i to nc do
   af,bf,rf := op(subs(fsl[i],[a,b,sqrt(r2)]));
   tp := sprintf("sol=%2d, a=%6.3f, b=%6.3f, r=%6.3f",i,af,bf,rf);
   tpp := TITLE(tp);
   fr[i] := [tpp,
    CURVES(op([1,1],pe[1]),colE),
    CURVES(op([1,1],pe[2]),colE),
    CURVES(op([1,1],pe[3]),colE),
    CURVES(op([1,1],pc[i]),colC),
    plpt[i],
   NULL];
  od:
> sc := SCALING(CONSTRAINED):
> ax := AXESSTYLE(NORMAL): axno := AXESSTYLE(NONE):
> rbgP := PLOT(
   CURVES(op([1,1],pe[1]),colE),
   CURVES(op([1,1],pe[2]),colE),
   CURVES(op([1,1],pe[3]),colE),
   seq(CURVES(op([1,1],pc[i]),colC),i=1..nc),
   seq(plpt[i],i=1..nc),
   sc,axno,
  NULL):
```

```
> rbgP;
```



```
> PLOT(ANIMATE(seq(fr[i],i=0..nc)),sc,ax,NULL):
```

exact minimal polynomial for radius calculation

```
> kernelopts(printbytes=false):
> rs[1] := resultant(eq[1],eq[2],b):
> rs[2] := resultant(eq[1],eq[3],b):
> rs[3] := resultant(eq[2],eq[3],b):
> for i to 3 do rs[i] := rs[i]/icontent(rs[i]) od:
> time();
```

63.170 (4.4.1)

```
> rs[4]:=resultant(rs[1],rs[2],a):
  time();
  rs[5]:=resultant(rs[1],rs[3],a):
  time();
  mpx := gcd(rs[4],rs[5]):
  time();
  mpx := mpx/icontent(mpx):
  time();
  save mpx,
  "/private/automount/home/gfee/Desktop/Apollonius/mpx";
```

38779.650

79361.230

84482.530

84482.550 (4.4.2)

```
> degree(mpx);
```

184 (4.4.3)

```
> length(maxnorm(mpx));
```

2724 (4.4.4)

approximate Sturm sequence calculation to show there are 68 positive roots

```
> Digits := 16;
```

$Digits := 16$ (4.5.1)

```
> ss:=sturmseq(1.*mpx,r2):
> sturm(ss,r2,-infinity,infinity);
```

2 (4.5.2)

```
> Digits := 2^5;
```

$Digits := 32$ (4.5.3)

```
> ss:=sturmseq(1.*mpx,r2):
> sturm(ss,r2,-infinity,infinity);
```

22 (4.5.4)

```
> Digits := 2^6;
```

$Digits := 64$ (4.5.5)

```
> ss:=sturmseq(1.*mpx,r2):
> sturm(ss,r2,-infinity,infinity);
```

68 (4.5.6)

```
> Digits := 2^7;
```

$Digits := 128$ (4.5.7)

```
> ss:=sturmseq(1.*mpx,r2):
> sturm(ss,r2,-infinity,infinity);
```

68 (4.5.8)

```
> Digits := 2^8;
```

$Digits := 256$ (4.5.9)

```
> ss:=sturmseq(1.*mpx,r2):
> sturm(ss,r2,-infinity,infinity);
```

68 (4.5.10)

```
> sturm(ss,r2,0,infinity);
```

68 (4.5.11)

```
> sturm(ss,r2,.05,32.);
```

68 (4.5.12)

References

1

Jim Wilson
The problem of Apollonius
http://jwilson.coe.uga.edu/emt725/Apollonius/Prob.Apol.html

2

Robert H. Lewis and Stephen Bridgett
Conic Tangency Equations and Apollonius Problems in Biochemistry and Pharmacology
preprint, May 2002
http://www.bway.net/~lewis/lewbrid.pdf