

2つの円の交点座標と弓形図形のグラフィックスーその2 Jと非線形連立方程式

西川 利男

こののはじまりは群論であったが、このところはその応用としてアラベスク模様
魅せられ、図形のタイリングの数学とグラフィックスにはまっている。

前回の例会[1]ではもっとも簡単な導入として、半径の異なる2つの円が重なって
できる弓形図形の内部を塗りつぶすことから始めたが、これがなかなか一筋縄ではいか
ないことがわかった。

[1] 西川利男「2つの円で囲まれた弓形図形の内部を塗りつぶす
ーアラベスク図形タイリングに向けてー」JAPLA 研究会資料 2017/12/9

1. 円の方程式をそのまま解くー非線形連立方程式

簡単な例題で、やってみる。

例題1

$$\text{円1 } x^2 + y^2 = 5 \quad (1)$$

$$\text{円2 } (x-1)^2 + (y-2)^2 = 4 \quad (2)$$

円1は原点に中心をもつ半径 $\sqrt{5} = 2.23$ の円、円2は点(1, 2)に中心をもつ半径
2の円である。片方の円は原点に中心をもつとしても一般性は保たれる。

式(1)から $y^2 = 5 - x^2$ であるので、これを式(2)の y^2 に入れば、 y の2次の
項は消えて1次の項だけになる。つぎのようにして2つの円の交点の座標は求められ
る。

式(2)から

$$x^2 - 2x + 1 + (5 - x^2) - 4y = 0$$

$$-2x + 1 + 5 - 4y = 0$$

$$x = 3 - 2y \quad (3)$$

となるので、これを式(1)に入れば、 y だけの2次方程式になる。

$$(3 - 2y)^2 + y^2 = 5$$

$$4y^2 - 12y + 9 + y^2 - 5 = 0$$

$$5y^2 - 12y + 4 = 0$$

因数分解すると

$$(y-2)(5y-2) = 0$$

$$y = 2 \quad \text{または} \quad y = 2/5$$

$$y = 2 \quad \text{のとき} \quad x = 3 - 2 \times 2 = -1 \quad \text{つまり} \quad (-1, 2)$$

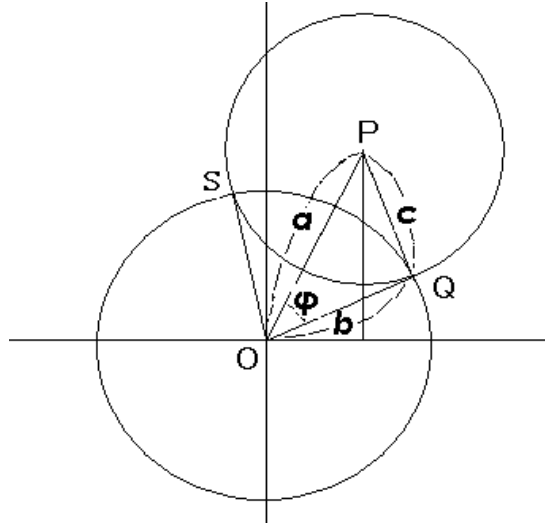
$$y = 2/5 \quad \text{のとき} \quad x = 3 - 2 \times 2/5 = 11/5 \quad \text{つまり} \quad (11/5, 2/5)$$

以上のようにして、2つの交点の座標は求められたが、あまりにトリッキーであり
強引すぎる。もっと素直に平易にできないだろうか。

円は2次曲線であるので、これを代数的に解くとすると2次の未知数を含んだ、つ
まり非線形の連立方程式となる。したがって、その解はあるときには根号を含んだ複
雑な式になるのは当然といえる。

2. 極座標を用いて解く

元来、円の表示には直交座標より極座標がより自然である。極座標を用いてあらた
めて計算してみる。



上の図のよう
と円 P, 中心
の交点を Q と S
の各辺 **c, a, b** の間に次の余弦法則を適用する。

$$c^2 = a^2 + b^2 - 2ab \cos \varphi$$

$$\varphi = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

2つの円の座標、半径を使うと

$$\varphi = \arccos\left(\frac{(p^2 + q^2) + r^2 - R^2}{2\sqrt{p^2 + q^2} * r}\right)$$

$$\alpha = \arctan(q / p) - \varphi$$

$$\beta = \alpha + 2\varphi$$

ここで α と β とは、円 O において、中心から点 Q と S とを見込む角である。

すると円 O でできる弓形は、この円弧の一部の上で

点 $(r * \cos \alpha, r * \sin \alpha)$ から点 $(r * \cos \beta, r * \sin \beta)$

をつなげた図形として描くことができる。

同様にして円 P でできる弓形は、この円弧の一部の上で

点 $(R * \cos \beta, R * \sin \beta)$ から点 $(R * \cos \alpha, R * \sin \alpha)$

をつなげた図形として描くことができる。

に円 O, 中心 (0, 0), 半径 r
(p, q), 半径 R およびそ
ときめて、三角形 $\triangle OPQ$

3. 計算とグラフィックスのJプログラム

前回発表のウィンドウズ・グラフィックスのプログラム・フォームを元に、そこで作ったいろいろな円のサブルーチン・ツールを使って、前節の極座標を用いた計算から、目的とする図形を描くことができる。プログラムの主要部分のみを記した。

```
NB. The Latest New Version / 2018/3/10 =====
```

```
NB. Enter_In Position and Radius of Circles in Edit Box =====
```

```
alhambra_X2Enter_button=: 3 : 0
XB =: ". X2Enter
)
alhambra_Y2Enter_button=: 3 : 0
YB =: ". Y2Enter
)
alhambra_R2Enter_button=: 3 : 0
RB =: ". R2Enter
)
```

```
NB. ***** Display Two Circles and Cross Positions *****
```

```
alhambra_CircA_button=: 3 : 0 NB. =====
wr '== Two Circles Problem =='
wr ', '
wr ' X^2 + Y^2 = ', (": *: RA)
wr '(X-', (": XB), ')^2 + (Y-', (": YB), ')^2 = ', (": *: RB)
wr ', '
glrgb 0 0 255
glpen 2 0
```

```
NB. circle-1, (0, 0) R0 = 5
```

```
XA =. 0
```

```
YA =. 0
```

```
NB. RA =: 5 NB. Radius Global value
```

```
gncircle_empty (circ2_adjXY XA), (circ2_adjXY YA), (circ2_adjR RA)
```

```
NB. circle-2, (1, 2) R1 = 2
```

```
glrgb 0 255 0
```

```
glpen 2 0
```

```
NB. XB =: 4 NB. X2 Global value
```

```
NB. YB =: 3 NB. Y2 Global value
```

```
NB. RB =: 5 NB. Radius Global value
```

```
gncircle_empty (circ2_adjXY XB), (circ2_adjXY YB), (circ2_adjR RB)
```

```
NB. plot cross points =====
```

```
glrgb 0 0 0
```

```
glbrush ''
```

```
NB. Calculate Cross Points by Using circross subroutine
```

```
'C4PX1 C4PY1 C4PX2 C4PY2' =: circross RA, RB, XB, YB NB. global value
```

```
wr 'cross points: (X1, Y1): ', ": C4PX1, C4PY1
```

```
wr 'cross points: (X2, Y2): ', ": C4PX2, C4PY2
```

```
gncircle (circ2_adjXY C4PX1), (circ2_adjXY C4PY1), 10
```

```
gncircle (circ2_adjXY C4PX2), (circ2_adjXY C4PY2), 10
```

```
NB. draw X,Y axis 4=====
```

```

glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)
alhambra_CircB_button=: 3 : 0
glrgb 255 0 0
glpen 4 0
glbrush ''

alpha =. adarctan (C4PX1, C4PY1)
wr 'alpha: ', (' ': alpha), ' deg'

beta =. adarctan (C4PX2 , C4PY2)
if. ((beta - alpha) < 0) do. beta =. beta + 360 end.
wr 'beta: ', (' ': beta), ' deg'

(alpha, beta) gnarc (circ2_adjXY 0), (circ2_adjXY 0), (circ2_adjR RA)

NB. draw X,Y axis =====
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)

alhambra_CircC_button=: 3 : 0
glrgb 0 255 0
glpen 4 0
glbrush ''

gamma =. adarctan ((C4PX2 - XB), (C4PY2 - YB))
wr 'gamma: ', (6j2 ' ': gamma), ' deg'

delta =. adarctan ((C4PX1 - XB) , (C4PY1 - YB) )
if. ((delta - gamma) < 0) do. delta =. delta + 360 end.

wr 'delta: ', (6j2 ' ': delta), ' deg'

(gamma, delta) gnarc (circ2_adjXY XB), (circ2_adjXY YB), (circ2_adjR RB)

NB. draw X,Y axis =====
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)

```

ここで用いたサブルーチン・プログラムは次のものである。すべて末尾のJプログラム・リストにあげてある。

```

gncircle ..... 円の内部を塗りつぶして描く
gncircle_empty..... 円の内部を空白にして描く
gnarc ..... 円弧の一部と弦とで囲まれた図形の内部を塗りつぶす
adarctan ..... arctangent の値を角度の象限に合わせて、度数で示す

```

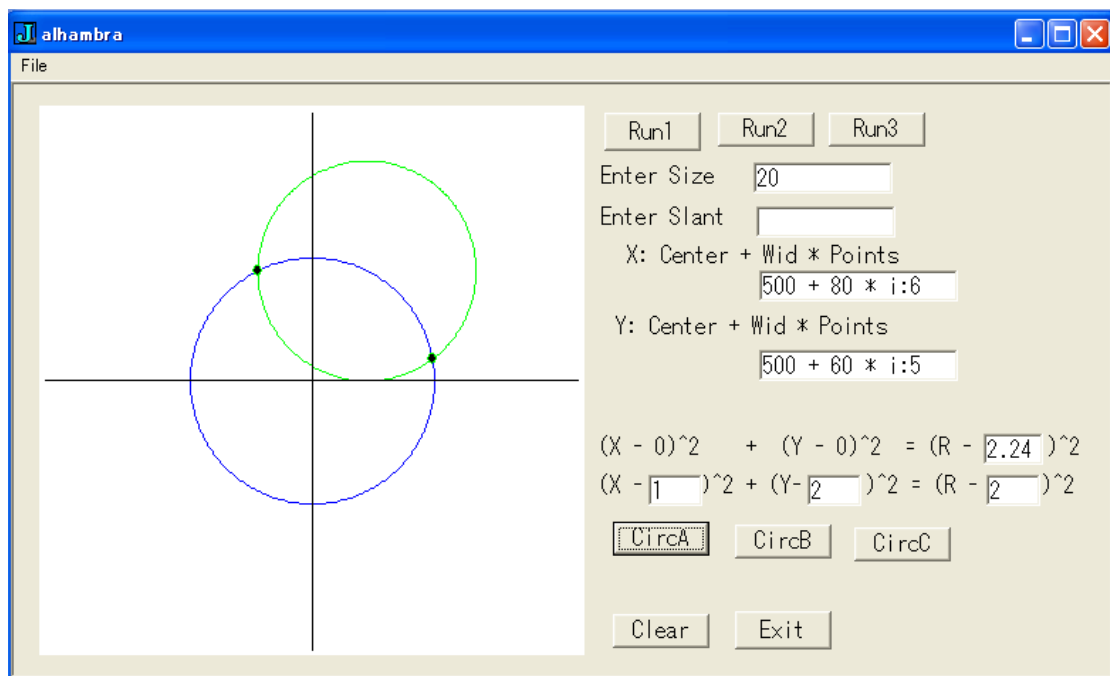
circ2_adjXY …… 計算した位置座標値をディスプレイ画面の値に調整する

circ2_adjR …… 計算した半径値をディスプレイ画面の値に調整する

4. Jによるグラフィック・プログラムの実際

前回のJグラフィック・プログラムに追加、拡張して、任意の位置と大きさを持つ2つの円の重なり部分の弓形図形の内部を塗りつぶすグラフィックスを作った。

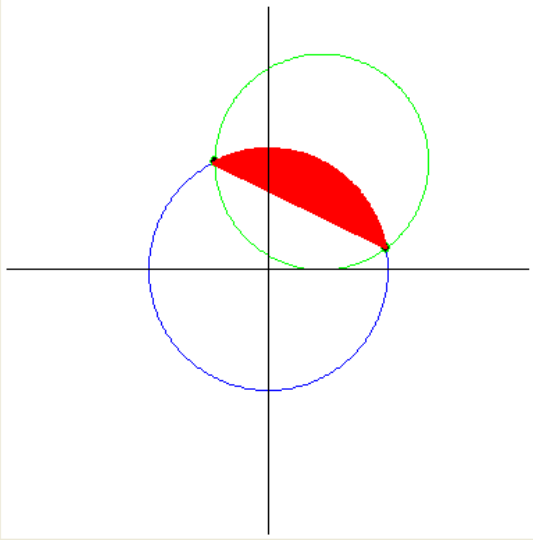
まず、2つの円の方程式の半径、中心座標の値を入力する。そしてボタンCircAを押すと、交点が表示される。



次にボタンCircBを押すと、円1（円の中心が座標の中心）による弓形図形の半分がアカで塗りつぶされる。

alhabra

File



Run1 Run2 Run3

Enter Size

Enter Slant

X: Center + Wid * Points

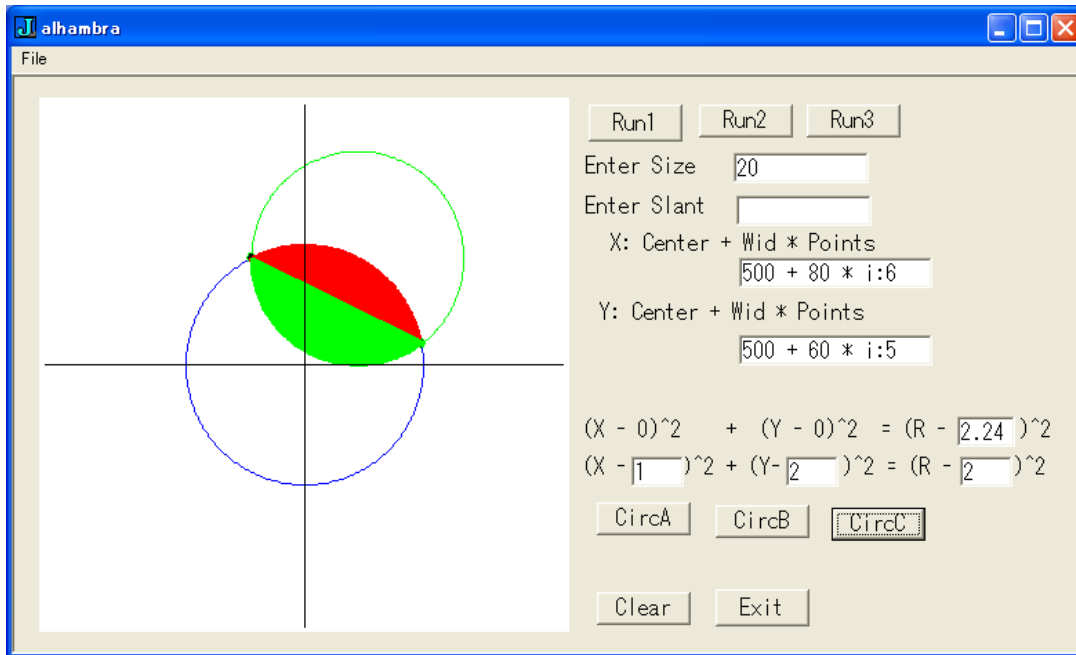
Y: Center + Wid * Points

$(X - 0)^2 + (Y - 0)^2 = (R - \underline{2.24})^2$
 $(X - \underline{1})^2 + (Y - \underline{2})^2 = (R - \underline{2})^2$

CircA CircB CircC

Clear Exit

続いて、ボタンCircCを押すと、円2（円の中心は任意の位置）による弓形図形の残りの半分がミドリで塗りつぶされる。



プログラム実行の計算値は、次のようになる。
run

== Two Circles Problem ==

$$X^2 + Y^2 = 5.0176$$

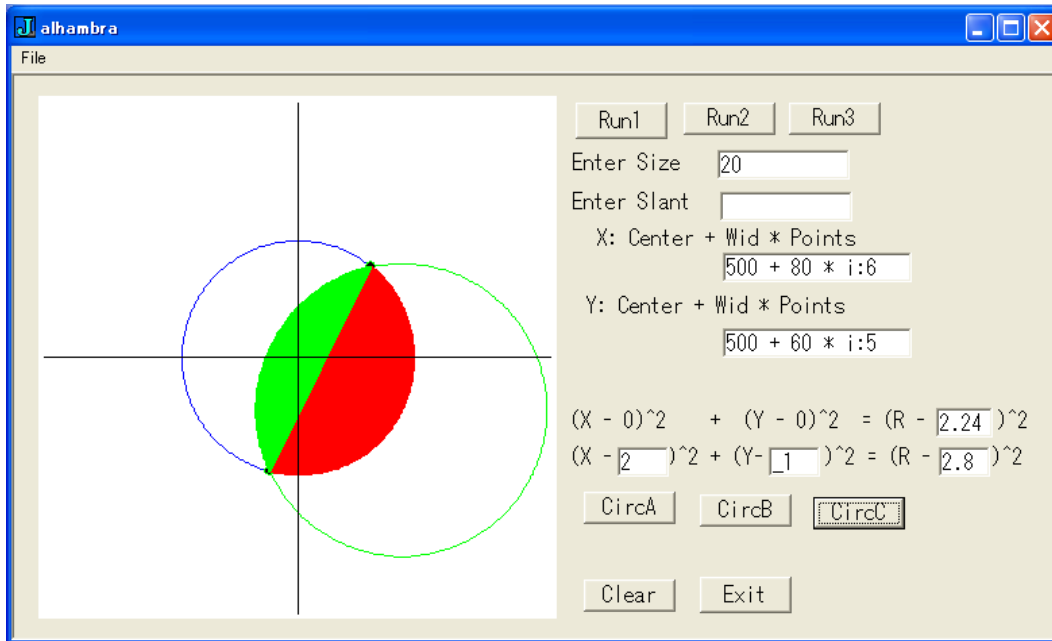
$$(X-1)^2 + (Y-2)^2 = 4$$

cross points: (X1, Y1): 2.20352 0.402642
cross points: (X2, Y2): -0.999995 2.0044

alpha: 10.3553 deg
beta: 116.515 deg

gamma: 179.87 deg
delta: 307.00 deg

2つの円の大きさ、位置を変えた別の例でやってみる。



```
run ''
== Two Circles Problem ==
```

$$X^2 + Y^2 = 5.0176$$

$$(X-2)^2 + (Y-1)^2 = 7.84$$

```
cross points: (X1, Y1): _0.542284 _2.17337
cross points: (X2, Y2): 1.41332 1.73785
```

```
alpha: 255.99 deg
beta: 410.88 deg
```

```
gamma: 102.09 deg
delta: 204.78 deg
```

Jプログラム・リスト

```
NB. two_cicles.ijs extended from alhambra.ijs
NB.      for Alhambra_graphics
```

```
require 'trig'
load 'gl2'
ALHAMBRA=: 0 : 0
pc alhambra;
menupop "File";
menu new "&New" "" "" "" ";
menu open "&Open" "" "" "" ";
menusep ;
menu exit "&Exit" "" "" "" ;
```



```

menupopz;
xywh 208 9 34 12;cc Run1 button;
xywh 254 165 34 12;cc cancel button;cn "Exit";
xywh 9 7 192 172;cc Alham isigraph;
xywh 248 9 34 11;cc Run2 button;
xywh 211 166 34 11;cc Clear button;
xywh 207 38 50 12;cc label static;cn "Enter Slant";
xywh 261 38 50 11;cc Entslant edit ws_border es_autohscroll;
xywh 216 50 104 10;cc label static;cn "X: Center + Wid * Points";
xywh 262 58 71 11;cc EntX edit ws_border es_autohscroll;
xywh 212 72 97 10;cc label static;cn "Y: Center + Wid * Points";
xywh 262 83 71 11;cc EntY edit ws_border es_autohscroll;
xywh 207 25 50 10;cc label static;cn "Enter Size";
xywh 260 24 50 11;cc EntR edit ws_border es_autohscroll;
xywh 287 9 34 11;cc Run3 button;
xywh 211 137 34 11;cc CircA button;
xywh 254 138 34 11;cc CircB button;
xywh 296 139 34 11;cc CircC button;
xywh 206 110 132 10;cc label static;cn "(X - 0)^2 + (Y - 0)^2 = (R - ";
xywh 360 110 20 10;cc label static;cn ")^2";
xywh 341 109 22 11;cc R1Enter edit ws_border es_autohscroll;
xywh 206 122 20 10;cc label static;cn "(X - ";
xywh 238 122 45 10;cc label static;cn ")^2 + (Y- ";
xywh 296 122 46 10;cc label static;cn ")^2 = (R - ";
xywh 358 122 20 10;cc label static;cn ")^2";
xywh 223 122 20 11;cc X2Enter edit ws_border es_autohscroll;
xywh 279 122 20 11;cc Y2Enter edit ws_border es_autohscroll;
xywh 342 122 20 11;cc R2Enter edit ws_border es_autohscroll;
pas 6 6;pcenter;
rem form end;
)
run =: alhambra_run
alhambra_run=: 3 : 0
wd ALHAMBRA
NB. initialize form here
SIZE =: 20
wd 'set EntR *', '20'
SLA =: 0
XX =. '500 + 80 * i:6'
wd 'set EntX *', XX
YY =. '500 + 60 * i:5'
wd 'set EntY *', YY
wd 'pshow;'
)

alhambra_close=: 3 : 0
wd'pclose'
)

alhambra_cancel_button=: 3 : 0
alhambra_close''
)

alhambra_Clear_button=: 3 : 0
XWP =: EntX
YWP =: EntY
glclear ''
glshow ''
)

```

```

NB. draw circle x, y, r / paint inside =====
gncircle =: 3 : 0
'x y r' =. y.
glellipse (x-r), (y-r), (2*r), (2*r)
)

```

```

alhambra_Run1_button=: 3 : 0
glrgb 255 0 0
glbrush ''
glpen 1 0
gncircle L:0 AIJR
glshow ''
)

```

```

NB. slant x values version =====
alhambra_Entslant_button=: 3 : 0
SLA =: ". Entslant
)

```

```

slant =: 3 : 0
:
SL =. y.
A0 =. x.
'Row Col' =. $ A0
NB. A =. (10*i:5),. 0
A =. (SL * i: -: <: Row),. 0
AA =. <"(1) A
AAA =. (Row, Col)$ (Col#AA)
A0 + L:0 AAA
)

```

```

alhambra_EntX_button=: 3 : 0
XWP =: EntX
)

```

```

alhambra_EntY_button=: 3 : 0
YWP =: EntY
)

```

```

alhambra_EntR_button=: 3 : 0
SIZE =: ". EntR
)

```

```

alhambra_Run2_button=: 3 : 0
AIX =. ". XWP
AJY =. ". YWP
AIJXY =. AIX grid AJY

```

```

glclear ''
glrgb 255 0 0
glbrush ''
glpen 1 0
AIJ2 =. AIJXY slant SLA
NB. AIJR2 =. (,&20) L:0 AIJ2
AIJR2 =. (,&SIZE) L:0 AIJ2
gncircle L:0 AIJR2
glshow ''
)

```

```

NB. draw circle x, y, r / empty inside =====
gncircle_empty =: 3 : 0
0 360 gncircle_empty y.
:
'alph beta' =. x.
'x y r' =. y.
t =. alph
while. t < beta
do.
  t1 =. t + 1
  gllines (x + (r*cosd t)), (y + (r*sind t)), (x + (r*cosd t1)), (y +
(r*sind t1))
  t =. t + 1
end.
)

gncircle_empty_bind =: 3 : 0
0 360 gncircle_empty y.
:
'alph beta' =. x.
'x y r' =. y.
t =. alph
while. t < beta
do.
  t1 =. t + 1
  glpolygon (x + (r*cosd t)), (y + (r*sind t)), (x + (r*cosd t1)), (y +
(r*sind t1))

  t =. t + 1
end.
glpolygon (x + (r*cosd beta)), (y + (r*sind beta)), (x + (r*cosd alph)), (y +
(r*sind alph))
)

NB. 重なった部分を塗りつぶす =====
=====
NB. Input Overlap Value into Slant Edit Box
wr =: 1!:2&2
gnarc_alpla =: 3 : 0
:
'D, E' =. x.
alph =. arctan (E % D)
X Y =. y.
XA =. (X*(cos alph)) + (Y*(sin alph))
YA =. (X*(sin alph)) - (Y*(cos alph))
)

NB. 2 circles problem =====
NB.  $X^2 + Y^2 = 5$ 
R0 =: %: 5

NB.  $(X-1)^2 + (Y-2)^2 = 4$ 
R1 =: 2
P1 =: 1
Q1 =: 2

```

NB. 2 Circle Cross Point / Revised Version 2018/2/4

=====

NB. adjust to graphic screen =====

```

circ2_adjXY =: 3 : '500 + 100 * y.'
circ2_adjXYXY =: 3 : '500 + 50 * y.'
circ2_adjR =: 3 : '100 * y.'
circ2_adjRR =: 3 : '50 * y.'

```

```

X0 =: 0
Y0 =: 0
R0 =: %: 5

```

```

circross =: 3 : 0
'r1 r2 x1 y1' =. y.
a =. -: (*: x1) + (*: y1) + (*: r1) - (*: r2)
R =. (*: x1) + (*: y1)
X1 =. (a*x1) + y1 * %: ((R * *:r1) - *:a)
X1 =. X1 % R
X2 =. (a*x1) - y1 * %: ((R * *:r1) - *:a)
X2 =. X2 % R
Y1 =. (a*y1) - x1 * %: ((R * *:r1) - *:a)
Y1 =. Y1 % R
Y2 =. (a*y1) + x1 * %: ((R * *:r1) - *:a)
Y2 =. Y2 % R
X1, Y1, X2, Y2
)

```

NB. arcsin, arccos, arctan returns in degree =====

```

NB. sind 60 (deg) => 0.866, darcsin 0.866 => 59.9971
NB. tand 30 (deg) => 0.57735, darctan 0.577 => 29.9849
darcsin =: 3 : '(180&%pi) * arcsin y.'
darccos =: 3 : '(180&%pi) * arccos y.'
darctan =: 3 : '(180&%pi) * arctan y.'

```

```

adarctan =: 3 : 0
'ax ay' =. y.
z =. | darctan (ay % ax)
if. (ax > 0) *. (ay > 0) do. z end.
if. (ax < 0) *. (ay > 0) do. z =. 180 - z end.
if. (ax < 0) *. (ay < 0) do. z =. 180 + z end.
if. (ax > 0) *. (ay < 0) do. z =. 360 - z end.
)

```

NB. angles of arc for draw arc figure =====

```

NB. circ_angle 5, 4, 2, 1
circ_angle =: 3 : 0
'r1 r2 x1 y1' =. y.
b =. r1
a =. -: (*: x1) + (*: y1)
c =. r2
ph =: darccos ((*: a) + (*: c) - (*: b)) % (2 * a * c)
)

```

```

alhambra_X2Enter_button=: 3 : 0
XB =: ". X2Enter
)

```

```
alhambra_Y2Enter_button=: 3 : 0  
YB =: ". Y2Enter  
)
```

```
alhambra_R2Enter_button=: 3 : 0  
RB =: ". R2Enter  
)
```

```

NB. Current New Version =====
alhambra_CircA_button=: 3 : 0 NB. =====
wr '== Two Circles Problem =='
wr ',,'
wr ' X^2 + Y^2 = ', (': *: RA)
wr '(X-', (': XB), ')^2 + (Y-', (': YB), ')^2 = ', (': *: RB)
wr ',,'
glrgb 0 0 255
glpen 2 0

NB. circle-1, (0, 0) R0 = 5
XA =. 0
YA =. 0
NB. RA =: 5 NB. Radius Global value
gncircle_empty (circ2_adjXY XA), (circ2_adjXY YA), (circ2_adjR RA)

NB. circle-2, (1, 2) R1 = 2
glrgb 0 255 0
glpen 2 0
NB. XB =: 4 NB. X2 Global value
NB. YB =: 3 NB. Y2 Global value
NB. RB =: %: 5 NB. Radius Global value

gncircle_empty (circ2_adjXY XB), (circ2_adjXY YB), (circ2_adjR RB)

NB. plot cross points =====
glrgb 0 0 0
glbrush ''

NB. Calculate Cross Points by Using circross subroutine
'C4PX1 C4PY1 C4PX2 C4PY2' =: circross RA, RB, XB, YB NB. global value

wr 'cross points: (X1, Y1): ', ': C4PX1, C4PY1
wr 'cross points: (X2, Y2): ', ': C4PX2, C4PY2

gncircle (circ2_adjXY C4PX1), (circ2_adjXY C4PY1), 10
gncircle (circ2_adjXY C4PX2), (circ2_adjXY C4PY2), 10

NB. draw X,Y axis 4=====
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)
alhambra_CircB_button=: 3 : 0
glrgb 255 0 0
glpen 4 0
glbrush ''

alpha =. adarctan (C4PX1, C4PY1)
wr 'alpha: ', (': alpha), ' deg'

beta =. adarctan (C4PX2, C4PY2)
if. ((beta - alpha) < 0) do. beta =. beta + 360 end.
wr 'beta: ', (': beta), ' deg'

(alpha, beta) gnarc (circ2_adjXY 0), (circ2_adjXY 0), (circ2_adjR RA)

```

```

NB. draw X,Y axis =====
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)

alhambra_CircC_button=: 3 : 0
glrgb 0 255 0
glpen 4 0
glbrush ''

gamma =. adarctan ((C4PX2 - XB), (C4PY2 - YB))
wr 'gamma: ', (6j2 ": gamma), ' deg'

delta =. adarctan ((C4PX1 - XB) , (C4PY1 - YB) )
if. ((delta - gamma) < 0) do. delta =. delta + 360 end.

wr 'delta: ', (6j2 ": delta), ' deg'

(gamma, delta) gnarc (circ2_adjXY XB), (circ2_adjXY YB), (circ2_adjR RB)

NB. draw X,Y axis =====
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)

```