

2つの円で囲まれた弓形図形の内部を塗りつぶす —アラベスク図形タイリングに向けて—

西川 利男

先の JAPLA 例会では、志村氏によるアルハンブラグラフィックスに大いに触発された。簡単そうに見えるグラフィックスにも大変な苦勞と精力的なプログラミングがあることが分かった。

私自身も先の群論・壁紙群の応用として、アラベスク図形に興味を持っている。その手初めとして、1つの図形を画面一杯に敷き詰めるタイリング(p 2 斜方格子群)のグラフィックスから始めたが、これは比較的容易にできた。しかしその後、たとえば円では、2つの円が重なり合ったとき、その円の境界線のできる弓形図形の内部のみを塗りつぶそうとすると、志村氏も指摘されているように中々筋縄ではいかない。今回、これを解決すべく、私なりの試行錯誤のプログラミングをおこなった。その過程を報告する。もちろん、目標とするのはアラベスク図形のグラフィックスである。

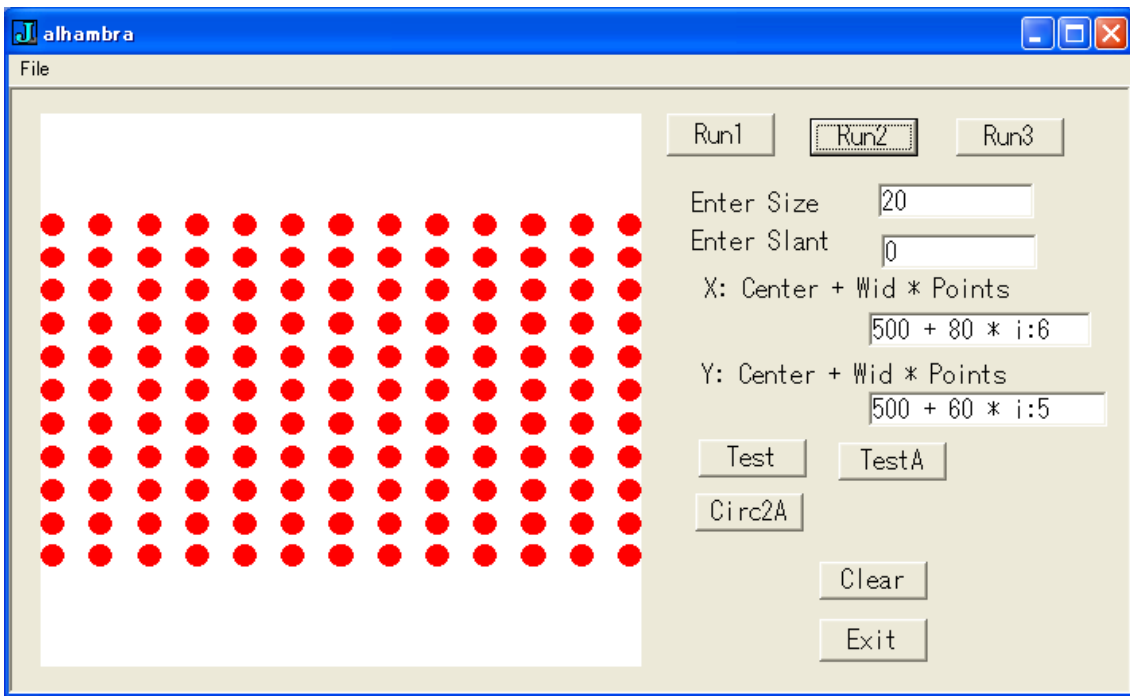
1. Jのgl2グラフィックスコマンドの制約と特徴

円を描くのに、gl2ではコマンド glellipse を用いるが、あまり使い易くない。そこで円の中心座標と半径をもとにして、内部を塗りつぶす gncircle と塗りつぶさない gncircle_empty とそれぞれ次の2つのコマンドを作った。なお gncircle_empty では円の一部、つまり弧も描けるようにした。

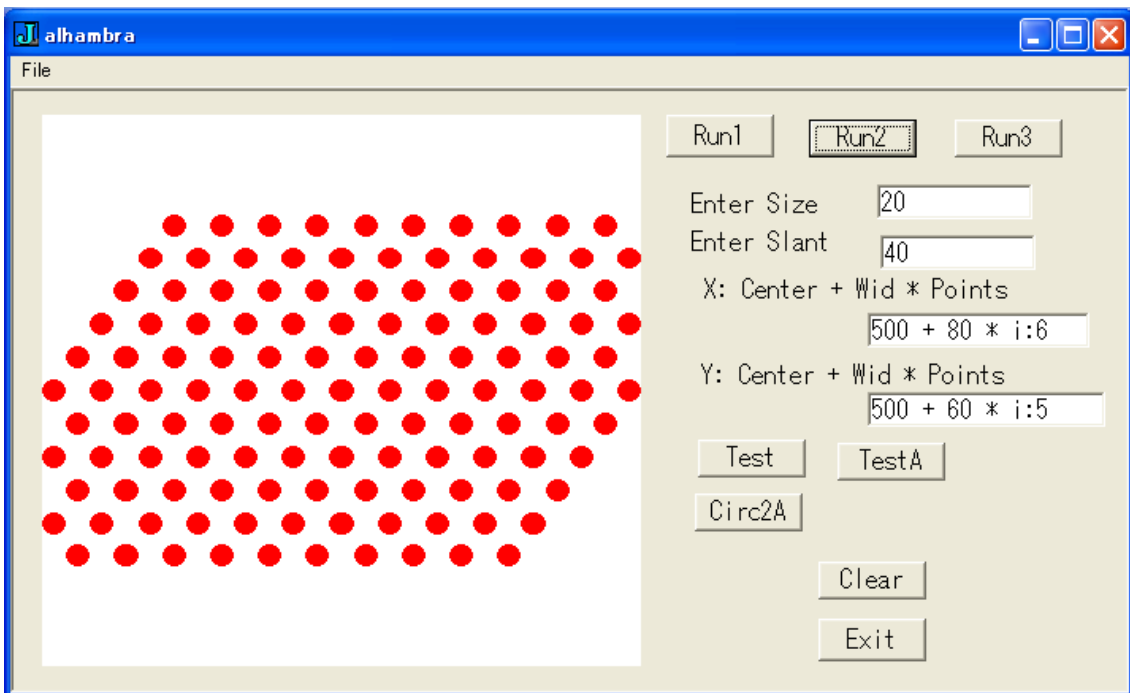
```
NB. draw circle x, y, r / paint inside =====
gncircle =: 3 : 0
'x y r' =. y.
glellipse (x-r), (y-r), (2*r), (2*r)
)
NB. draw circle x, y, r / empty inside =====
gncircle_empty =: 3 : 0
360 gncircle_empty y.
:
te =. x.
'x y r' =. y.
t =. 0
while. t < te
do.
  t1 =. t + 1
  glines (x + (r*cosd t)), (y + (r*sind t)), (x + (r*cosd t1)),
    (y + (r*sind t1))
  t =. t + 1
end.
)
```

2. 円を例としたの直交格子と斜交格子のタイリング

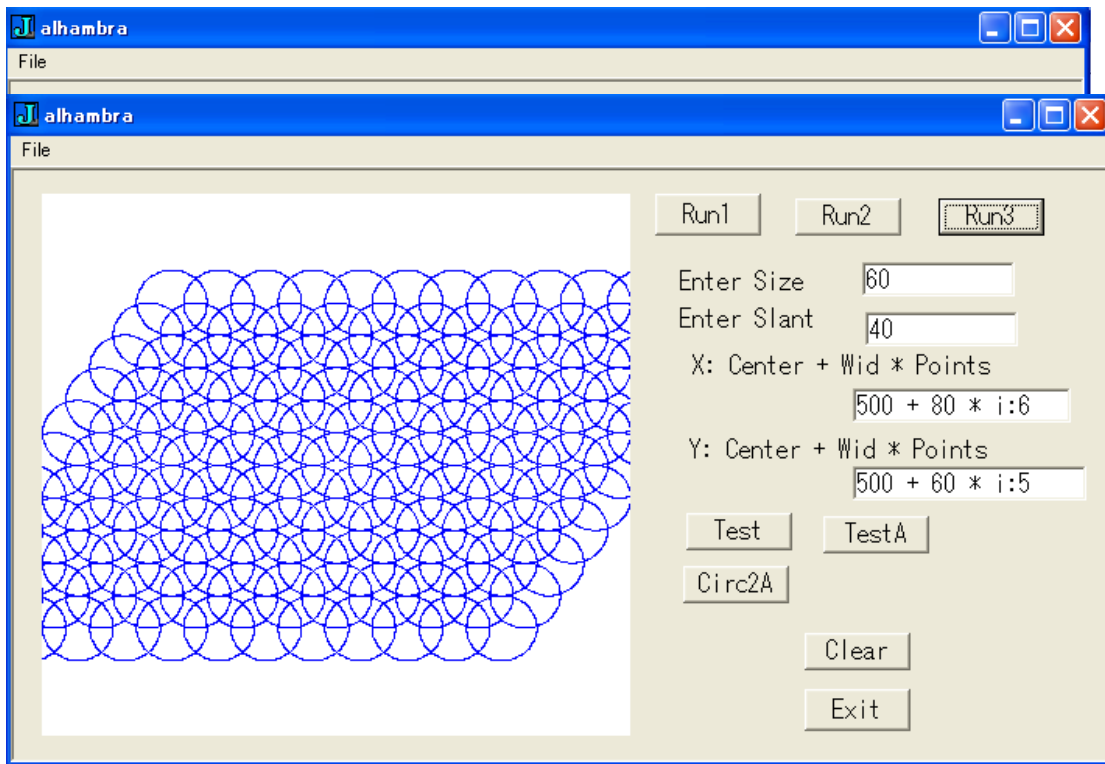
まず、円を例として直交格子をベースにタイリングしてみる



つぎに、上の円を斜交格子をベースとしてタイリングする。



はじめに基準となる位置座標、タイリングの幅、繰り返し回数を指定する。[Run2]のボタンでは円の大きさ、斜交の傾き角をいろいろ変えて実行できる。
なお、プログラムの詳細は末尾の付録としてあげる。 [Run3]のボタンでは内部を塗りつぶさない円でタイリングする。



さらに円の大きさを大きくしてみる。これだけでも仲々きれいな図ができる。

ここで、重なり合っただけの弓形図形の内部を塗りつぶせば、さらに美的なタイリング図形ができるだろう。ところが、志村氏が大変な苦勞されたように、これがそう容易ではない。かつてMSDOSのBASICではGRAPH_FLOODのような命令があつて、点を囲む図形の内部を塗りつぶせたが、Jのg12には以前はあつたが、今はない。

3. 2つの円でできる図形

これからが、私の試行錯誤のプログラミングであり、その経過を報告する。

例題として、インターネットの中から、次のような2つの円を取りあげる。

$$\text{円1: } x^2 + y^2 = 5$$

$$\text{円2: } (x-1)^2 + (y-2)^2 = 4$$

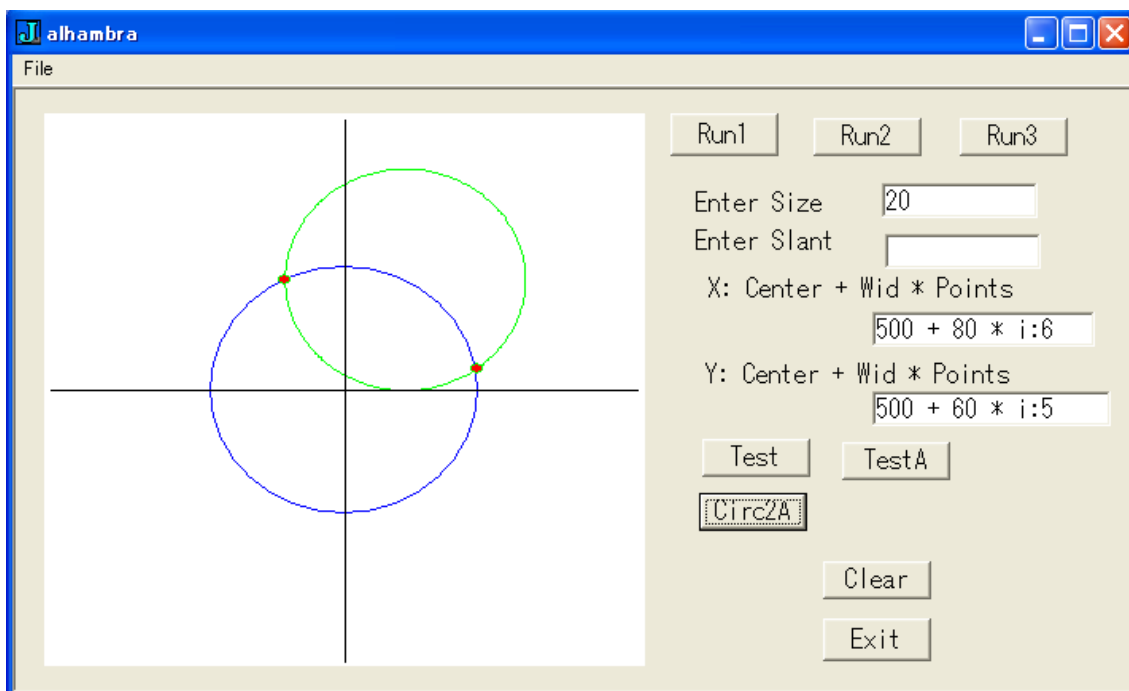
解析幾何学の問題として、上の2つの円の交点の座標は、2つの式を連立方程式として解いて、交わるときには2組の実根として計算で求められる。

$$(-1, 2) \text{ と } (11/5, 2/5)$$

これを元にグラフ表示をおこなう。

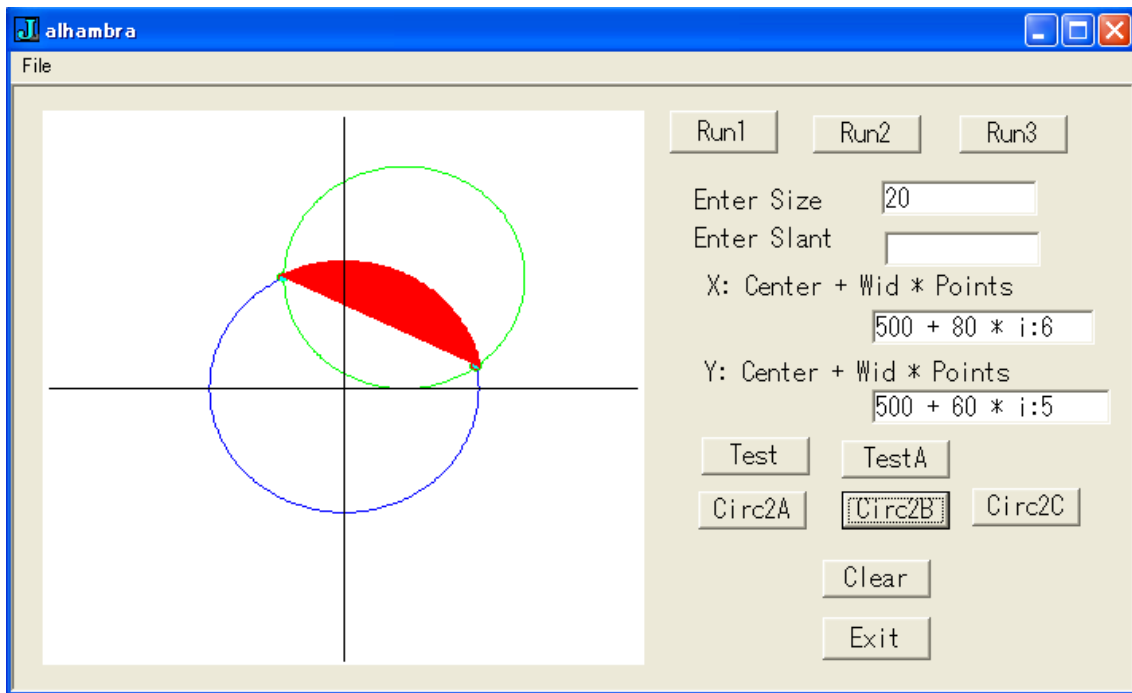
3. 1 2つの円の交点

ボタン[Circ2A]により実行される。交点は赤の点で示す。

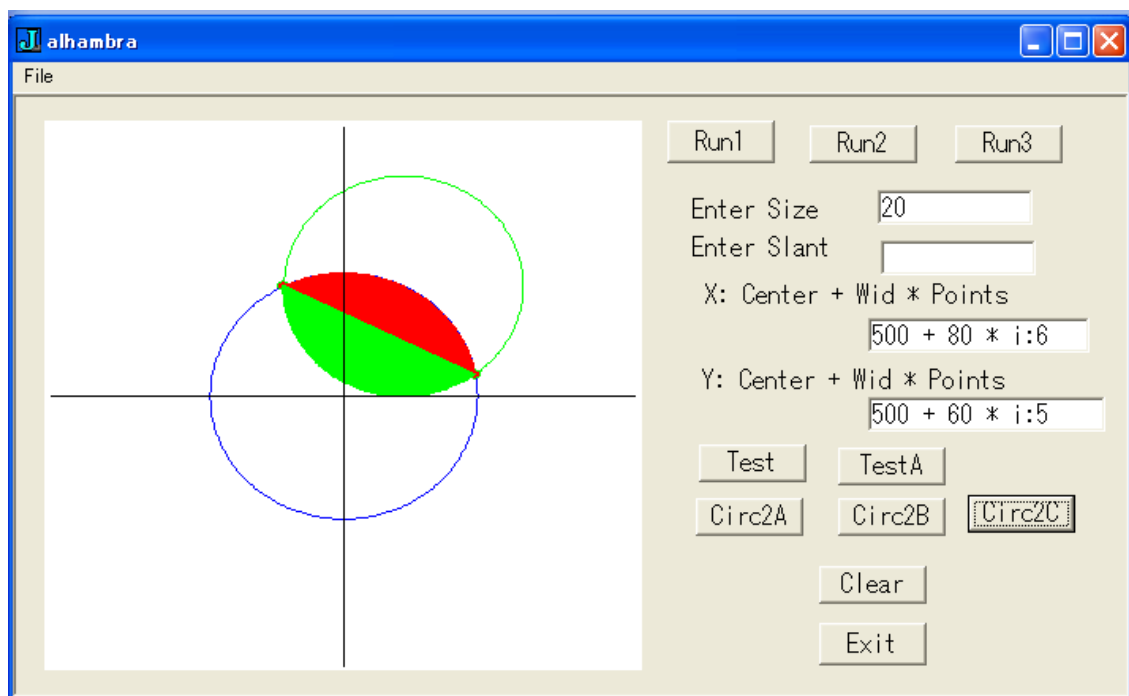


3. 2 2つの円で囲まれた弓形図形

円1による弓形は、ボタン[Circ2B]により以下のように実行される。



次に円2による弓形は、ボタン[Circ2C]により以下のように実行される。



上のようにして、2つの円の交点による弓形図形の内部の塗りつぶしができあがった。

これを可能にしたJプログラムのポイントは次である。

- 準備として次の描画を作った。
 - `gncircle` `glellipse` を中心、半径で動かすようにした (前述)
 - `gncircle_empty` 円の方程式に従って、輪郭点の各座標値を計算して、それを `gllines` により結ぶ (前述)
 - `gnarc` 上と同様だが、座標値の集合を `glpoygon` で結ぶことで塗りつぶしがおこなわれる。このとき、`glrgb`, `glpen`, `glbrush` をに適切な指定することで、
- 2つの円の交点の座標を元に、それぞれの中心から交点に向けての回転角を `arctan` で求める。その上でコマンド `gnarc` により、色を決めて (開始角, 終了角) `gnarc` 円の中心の X 座標, 円の中心の Y 座標, 円の半径) により、弓形をその色でぬりつぶす。

プログラムの詳細は付録として、稿末にあげてある。

プログラム・リスト

NB. Alhambra_graphics

```
NB. I =: 10 + 5*i:4
NB. I
NB. _10 _5 0 5 10 15 20 25 30
NB. J =: 20 + 3 *i:5
NB. J
NB. 5 8 11 14 17 20 23 26 29 32 35
NB. IJ =: <"(1) > I (<@,.)/"(1 0) J
```

NB. generate grid coordinates =====

AI =: 500 + 80*i:6

AJ =: 500 + 60*i:5

grid =: 3 : 0

:

<"(1) > x. (<@,.)/"(1 0) y.

)

AIJ =: AI grid AJ

NB. X, Y plus R =====

AIJR =: (,&20) L:0 AIJ

require 'trig'

load 'gl2'

ALHAMBRA=: 0 : 0

pc alhambra;

menupop "File";

menu new "&New" "" "" "";

menu open "&Open" "" "" "";

menusep ;

menu exit "&Exit" "" "" "";

menupopz;

xywh 206 7 34 12;cc Run1 button;

xywh 254 148 34 12;cc cancel button;cn "Exit";

xywh 9 7 189 155;cc Alham isigraph;

xywh 251 8 34 11;cc Run2 button;

xywh 254 132 34 11;cc Clear button;

xywh 214 39 50 12;cc label static;cn "Enter Slant";

xywh 273 40 50 11;cc Entslant edit ws_border es_autohscroll;

xywh 218 52 104 10;cc label static;cn "X: Center + Wid * Points";

xywh 269 62 71 11;cc EntX edit ws_border es_autohscroll;

xywh 217 76 97 10;cc label static;cn "Y: Center + Wid * Points";

xywh 269 84 76 11;cc EntY edit ws_border es_autohscroll;

xywh 214 28 50 10;cc label static;cn "Enter Size";

xywh 272 26 50 11;cc EntR edit ws_border es_autohscroll;

xywh 297 8 34 11;cc Run3 button;

xywh 216 98 34 11;cc Test button;

xywh 260 99 34 11;cc TestA button;

xywh 215 113 34 11;cc Circ2A button;

xywh 260 113 34 11;cc Circ2B button;

xywh 301 112 34 11;cc Circ2C button;

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
NB. glellipse 500 500 200 200
NB. gncircle 500 500 20
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_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 ''
)

```

```

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

```

```

alhambra_Run3_button=: 3 : 0
AIX =. ". XWP
AJY =. ". YWP
AIJXY =. AIX grid AJY

```

```

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

```

NB. 囲った図形の塗りつぶし =====

```

gnarc =: 3 : 0
:
'th0 th' =. x.
'x y r', =. y.
xy =.
t =. th0
while. t < th
  do.
    t1 =. t + 1
    xy =. xy, (x + (r*cosd t)), (y + (r*sind t)), (x + (r*cosd t1)), (y +
(r*sind t1))
    t =. t + 1
  end.
xy =. xy, (x + (r*cosd th0)), (y + (r*sind th0))
glpolygon xy
NB. gllines ''
glshow ''
)

```

NB. 2 Circle Cross Point

```

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

```

```

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

```

NB. 円1と円2の交点 =====

```

alhambra_Circ2A_button=: 3 : 0
glrgb 0 0 255
glpen 2 0

```

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

```

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

```

```

X =: 500 + X0
Y =: 500 + Y0
R0 =: 100 * R0
gncircle_empty (circ2_adjXY X0), (circ2_adjXY Y0), (circ2_adjR R0)

```

```

NB. circle-2, (1, 2) R1 = 2
glrgb 0 255 0
glpen 2 0

P =: 1 NB. X2
Q =: 2 NB. Y2
R2 =: 2
NB. X =. circ2_adjXY P
NB. Y =. circ2_adjXY Q
NB. R =. circ2_adjR R2
NB. gncircle_empty X, Y, R
gncircle_empty (circ2_adjXY P), (circ2_adjXY Q), (circ2_adjR R2)

NB. plot cross points =====
glrgb 255 0 0
glbrush ',,'
gncircle (circ2_adjXY 11%5) , (circ2_adjXY 2%5) , 10
gncircle (circ2_adjXY _1) , (circ2_adjXY 2) , 10

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

```

```

NB. 円1の弓形 =====
alhambra_Circ2B_button=: 3 : 0

glrgb 255 0 0
glpen 4 0
glbrush ''
(10.3 , (180-63.43)) gnarc (circ2_adjXY 0), (circ2_adjXY 0), (circ2_adjR %:
5)

```

```

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

```

```

NB. 円1の弓形 =====
alhambra_Circ2C_button=: 3 : 0
glrgb 0 255 0
glpen 4 0
glbrush ''
(180 306.87) gnarc (circ2_adjXY P), (circ2_adjXY Q), (circ2_adjR 2)

```

```

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

```

```

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

```