

(平成30年7月3日一部修正)

(平成18年12月28日一部修正)

平成18年11月1日

気象庁予報部

## 配信資料に関する技術情報(気象編)第237号

～台風情報の改善に伴う電文形式等の変更について その3～

平成19年4月より台風情報の改善を予定しており、この改善に伴う台風予報の表示方法の変更については、既に気象庁ホームページ等で「台風予報の図表示方法の指針」としてガイドラインを示すとともに、「気象庁における台風予報の図表示方法」を提示しているところです。このなかで、気象庁ホームページ等における表示方針として、各時刻の暴風警戒域の円の表示に代えて暴風警戒域の通過する範囲を線で表示することとしています。

この線表示について、与えられた台風予報の要素から範囲を示す線を導出する計算手法は複雑な場合分けもあることから、気象庁における導出手法を別添資料1にて示すとともに、サンプルのプログラムを別添資料2に示し、また、(一財)気象業務支援センターに提供します。サンプルプログラムが必要な場合は、同センターにお問い合わせください。

なお、サンプルのプログラムは導出手法の確認のために作成したものであり、参考として示すものです。本プログラムの全部又は一部を利用することは問題ありません。但し、気象庁が本プログラムの動作を保証するものではありませんし、本プログラムの一部又は全部を利用したことにより、利用者が被った直接的又は間接的ないかなる損害についても、気象庁は一切責任を負いません。また、本プログラムに関する個別の対応は行いかねますので、ご注意下さい。

台風予報の図表示方法の指針:

[http://www.jma.go.jp/jma/press/0603/28a/taifu\\_hyouji.html](http://www.jma.go.jp/jma/press/0603/28a/taifu_hyouji.html)

気象庁における台風予報の図表示方法:

[http://www.jma.go.jp/jma/press/0607/12a/typhoon\\_map.html](http://www.jma.go.jp/jma/press/0607/12a/typhoon_map.html)

### (修正履歴)

○平成18年12月28日

別添資料1に示すアルゴリズムにおける数式の誤りを修正しました。

○平成30年7月3日

サンプルプログラムにおける描画不具合を修正するとともに、サンプルプログラムの提供方法に、(一財)気象業務支援センターを通じた提供を追加しました。

(平成18年12月28日一部修正)

平成18年11月1日  
気象庁予報部

## 暴風警戒域の通過する範囲を示す弧・線分を選択するアルゴリズム

平成19年から始める台風予報の図表示方法では、予報時刻ごとに暴風警戒域の円を描くのではなく、円と接線が囲む領域を示す。この領域の境界線を定義する方法を説明する。

## 1. 暴風警戒域の通過する範囲の表示

下に示すAの円、Bの線分で囲まれた領域の一番外側の線だけを描くことで暴風警戒域の通過する範囲を表示できる。

- A 暴風域の円と暴風警戒域の円（両方あわせてこれ以降、「暴風警戒域の円」または単に「円」と呼ぶことにする）（予報円の表示を省略する時刻を含む）
- B 連続した（予報円の表示を省略する時刻を含む）二つの予報時刻の暴風警戒域の円の共通外接線上で、二つの接点を両端とする線分の部分

一番外側の線だけを表示するためには、円と線分のうち、円と線分で囲まれた領域の内側にある部分を消去すれば良い。領域は、円と、二つの円の共通外接線を二辺とする四角形（台形）からなる（図1.1）。ゆえに、円と線分のうち、領域を構成するひとつひとつの円・四角形の内側にある部分を消去することで目的が達成できる。四角形は外側に凸であれば良く、二辺が平行であるという台形の性質は必要ない。

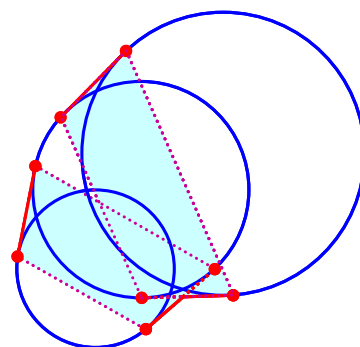


図 1.1 領域は円と台形で構成されている。

以下、目的を達成するためのアルゴリズムを解説する。第2節では領域（円と台形）の内側に円周や線分が入るかどうかの判別法を、第3節では判別のために必要な主な計算を説明する。個々の条件で円周や線分の描かない部分が分かった後、描く部分だけ残しておく（記憶しておく）方法を第4節で述べる。このアルゴリズムは一例であり、ほかにも方法がある。

また、このアルゴリズムを実現する FORTRAN サンプルプログラムも付ける。このプログラムは試験してみた例では目的どおりに動作したが、ほかの計算機、ほかの事例では必ずしも目的どおりに動作しないこともあり得る。

## 2. 領域（円と台形）の内側の判別法

接点の座標は分かっているとす。本解説ではその求め方は記述しない（サンプルプログラムでは求めている）。

## 2. 1 円弧の判別法

## 2. 1. 1 隣り合う円の向かい合った部分の弧

連続する予報時刻の円の共通外接線より内側の部分（弧）は描かない。弧のうち、円内、台形内の部分を描かない（それぞれ 2. 1. 2、2. 1. 3 で後述）という条件でも処理できるが、隣り合う円の向かい合った部分の弧は、台形や円の内外側の判定が必要ないので、描かない条件をより簡単に求められる。このため、この条件の処理をまず行う。

図 2.1 に示す円  $A$  と円  $B$  の共通外接線として  $A_1-B_1, A_2-B_2$  という線分が描ける。ここで  $A_1, A_2$  は円  $A$  の接点、 $B_1, B_2$  は円  $B$  の接点。円  $A$  は  $A_1$  と  $A_2$  で二つの弧に分けられる。円  $B$  も同様。弧  $A_1-A_2$  で円  $B$  に近い側、弧  $B_1-B_2$  で円  $A$  に近い側は描かない。

「近い側の弧」は次のようにして判別できる。円の中心を通過して  $x$  軸と平行な直線からの角度で円周上の点の位置を指定できる。角度は  $x$  軸の方向を  $0$  度として、反時計回りに大きくなるようにとる。

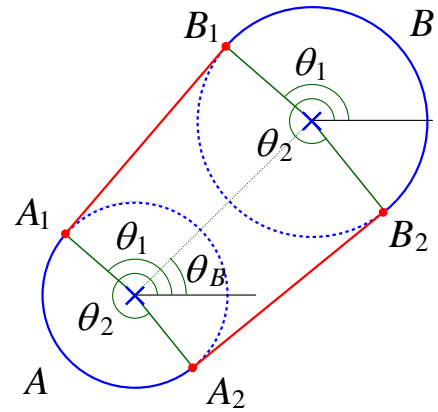


図 2.1 隣り合う円の向かい合った部分の弧.

1. 円  $A$  の二つの接点を表す角度  $\theta_1, \theta_2$  を求める。 $\theta_1, \theta_2$  は  $0^\circ$  から  $360^\circ$  の間をとり、 $\theta_1 < \theta_2$  とする。円  $B$  の二つの接点を表す角度も  $\theta_1, \theta_2$  に等しい。
2. 円  $A$  の中心から円  $B$  の中心へ方向の角度  $\theta_B$  を求める。
- 3a.  $\theta_1 < \theta_B < \theta_2$  の場合：円  $A$  の  $\theta_1 \sim \theta_2$  の部分を描かない。円  $B$  の  $0^\circ \sim \theta_1, \theta_2 \sim 360^\circ$  の部分を描かない。
- 3b.  $\theta_B < \theta_1$  または  $\theta_2 < \theta_B$  の場合：円  $A$  の  $0^\circ \sim \theta_1, \theta_2 \sim 360^\circ$  の部分を描かない。円  $B$  の  $\theta_1 \sim \theta_2$  の部分を描かない。

### 2. 1. 2 円の内側の円弧

ある円のうち、ほかの円に含まれている円弧は描かない。ただし、隣り合う円で重なる部分は「2. 1. 1 隣り合う円の向かい合った部分の弧」によって描かれない部分が分かっているので、この条件を考慮しないで良い。二つの円  $C_1, C_2$  の半径を  $r_1, r_2$ 、中心の間の距離を  $L$  とする。

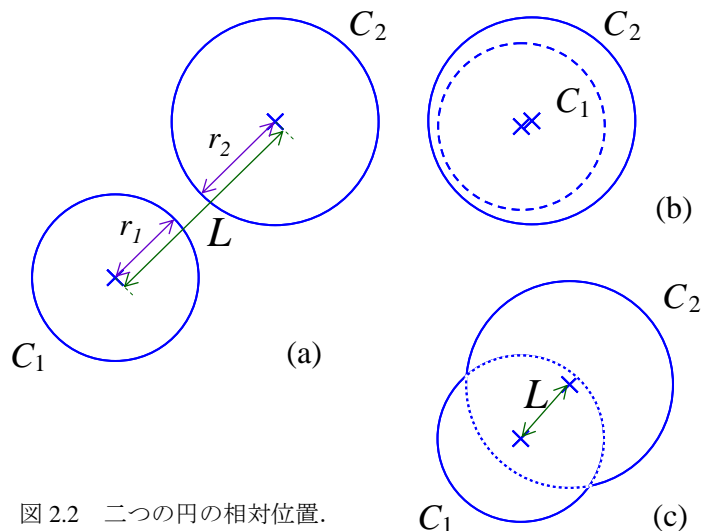


図 2.2 二つの円の相対位置.

- (i)  $L > r_1 + r_2$  : 二つの円は重ならないので、円の内側の円弧は存在しない (図 2.2(a)).
- (ii)  $L + r_1 < r_2$  :  $C_1$  は  $C_2$  の内側にあるので描かない (図 2.2(b)).
- (iii)  $L + r_2 < r_1$  :  $C_2$  は  $C_1$  の内側にあるので描かない。
- (iv) それ以外の場合：二つの円は交わり、円は交点で二つの弧に分かれる。二つの弧のうち、他方の円の内側にある方は描かない (図 2.2(c)).

「他方の円の内側の弧」の次のようにして判別できる（図 2.3）。

1. ある円  $C_1$  のうち、他方の円  $C_2$  の内側にある弧の角度の半分  $\alpha$  は、余弦定理を利用して次の式で求められる。

$$\alpha = \cos^{-1} \left\{ \frac{L^2 + r_1^2 - r_2^2}{2 L r_1} \right\}$$

ここで、 $L$  は二つの円の間  
の距離、 $r_1, r_2$  はそれぞれ  
円  $C_1, C_2$  の半径。

2. 円  $C_1$  の中心から円  $C_2$  の中心へ  
の方向の角度  $\theta$  を求める。
3. 円  $C_1$  の  $\theta - \alpha \sim \theta + \alpha$  の  
部分は他方の円  $C_2$  の内側  
にあるので描かない。
4. 描かない部分を示す角度  
は常に  $0$  と  $360$  の間だと都  
合が良いので、角度が  $0^\circ$   
または  $360^\circ$  をまたぐ場合  
には、描かない角度を次のように書き換える。

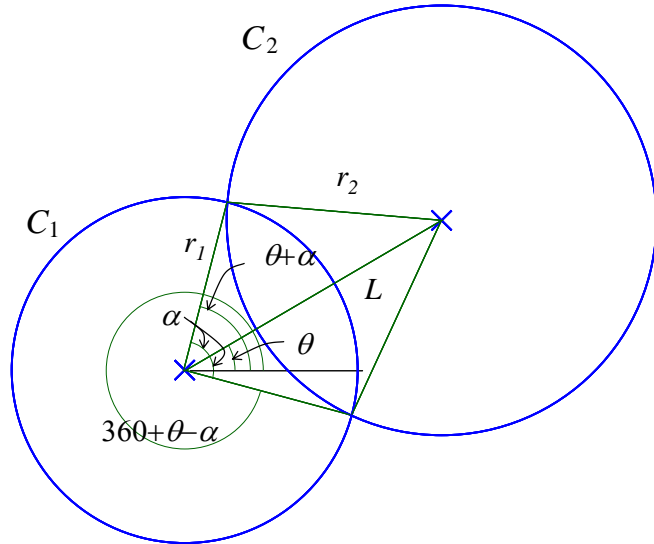


図 2.3 二つの円が交差する場合.

- 4a.  $\theta - \alpha$  が  $0$  よりも小さい場合： $360 + \theta - \alpha \sim 360^\circ, 0^\circ \sim \theta + \alpha$  の部分を  
描かない。
- 4b.  $\theta + \alpha$  が  $360$  よりも小さい場合： $\theta - \alpha \sim 360^\circ, 0^\circ \sim \theta + \alpha - 360^\circ$  の部  
分を描かない。

## 2. 1. 3 台形の内側の円弧

ある円のうち、台形に含まれている円弧は描かない。ただし、ある円とその円に接する線分を辺とする台形との関係は「2. 1. 1 隣り合う円の向かい合った部分の弧」によって描かない部分が分かっているので、この条件を考慮しなくて良い。円と台形との相対位置を図 2.4 と表 1 に示す。

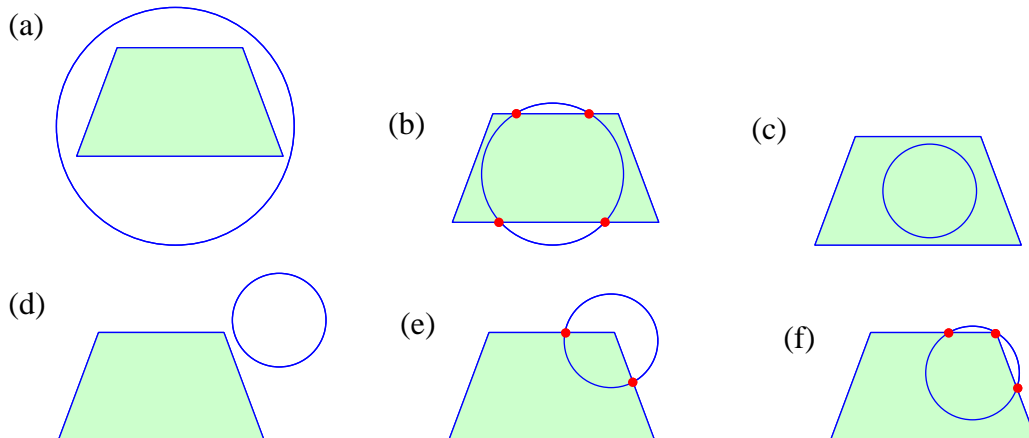


図 2.4 円と台形との相対位置.

表 1 円と台形との相対位置.

		台形の頂点が円の内側にあるかないか	
		頂点は4つとも円の内側	円の外側に少なくとも一つの頂点がある
交 点	なし	台形はすっぽり円の内側にある (図 2.4(a)).	a. 円の中心が台形の内側にある場合:円はすっぽり台形の内側にある (図 2.4(c)). b. 円の中心が台形の外側にある場合:円は台形の外側にある (図 2.4(d)).
	あり		円周は台形の外側の弧と内側の弧に分けられる (図 2.4(b), (e), (f)).

次のような方法で台形の内側の弧を特定できる。図 2.5 にフローチャートを示した。

1. 台形の頂点が円の内側にあるかどうかを判定する。円の中心と台形の頂点と間の距離が円の半径よりも小さければ、台形の頂点は円の内側にある。
2. 台形の4つの頂点がすべて円の内側にあれば、台形はすっぽり円の内側にある (図 2.4(a))。台形の内側の円弧は存在しない。円弧と台形の処理終了。
3. 台形の4辺と円との交点を求める。台形の頂点がすべて円の外でも辺と円がまじわる場合がある (図 2.4(b))。線分と円との交点を求める方法は3. 1を参照。
- 4a. 交点がない場合: 円の中心が台形の内側にあるかどうかを調べる。方法は3. 3を参照。
  - 4a-a. 円の中心が台形の内側にある場合: 円はすっぽり台形の内側にあるので、この円は描かない (図 2.4(c))。円弧と台形の処理終了。
  - 4a-b. 円の中心が台形の外側にある場合: 円は台形の外側にあり、台形の内側の円弧はない (図 2.4(d))。円弧と台形の処理終了。

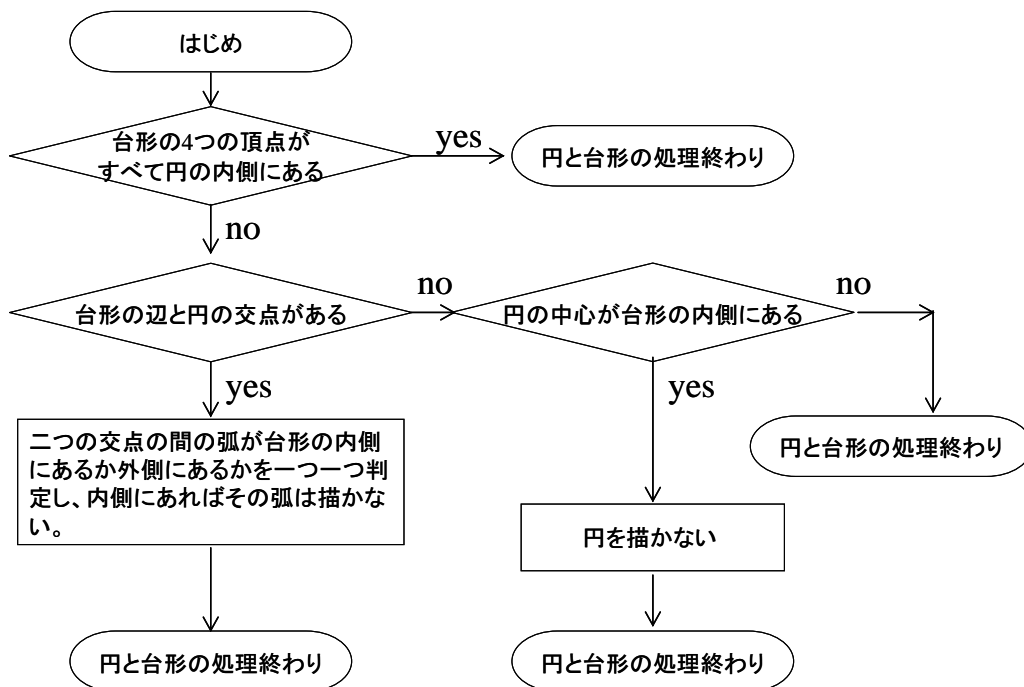


図 2.5 台形の内側の円弧に関する処理のフローチャート.

4b. 交点がある場合（図 2.4(b), (e), (f)）：

4b-1. 交点の角度を求め、角度の小さい順に交点を並べる。

4b-2. 二つの交点の間の弧が台形の内側にあるか外側にあるかを一つ一つ判定する。内側にあればその弧は描かない。図 2.4(f)のように、交点を境に交互に内/外/内/外/とはならない場合もあるので、一つ一つの弧を対象に内/外を判定する。具体的には、二つの交点の角度の中間の角度の円周上の点が入形の中にあるかないかで判定できる。

## 2. 2 線分の処理

線分のうち、円の内側、台形の内側は描かない。線分と円または台形との相対関係は図 2.6 に示す A, B, C, D の 4 通りがある。線分の端が円または台形の内側にあるかないか、線分と円または台形の辺と交点の数によって表 2 のように 4 通りを区別でき、線分をどう処理するか

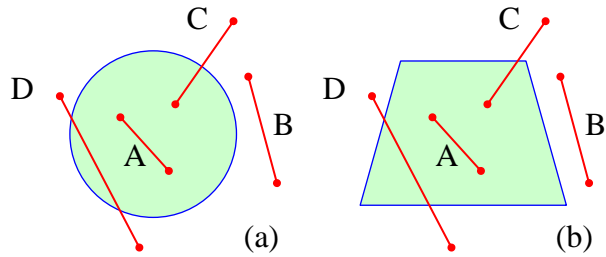


図 2.6 線分と円(a)または台形(b)との相対位置。

が決められる。4 通りのどれに当たるかを判別するアルゴリズムは一通りではない。

表 2 線分と円または台形との位置関係。( ) 内は図 2.6(a), (b) の図の線分につけた文字に対応する。

		線分の端が円または台形の内側にあるかないか		
		両端内側	片方内側	両端外側
交点の数	なし	線分を描かない (A)		線分を消す条件はない (B)
	1		内側の点と交点の間を描かない (C)	
	2			交点と交点の間を描かない。線分は二つに分かれる。(D)

### 2. 2. 1 円の内側の線分

線分のうち、円の内側は描かない。線分の端が円の内側にあるかどうかを先に判定するアルゴリズムを説明する（サンプルプログラムも同様）。図 2.7 にフローチャートを示す。

1. 線分の端が円の内側にあるかどうかを判定する。

2a. 線分の両端とも円の内側なら、その線分は描かない。図 2.6(a)の A の場合。線分と円の処理終わり。

2b. 線分の片方または両方の端が円の外側なら、線分と円との交点を求める。線分の両端が円の外でも線分と円がまじわることがある。線分と円との交点を求める方法は 3. 1 を参照。

2b-a. 交点がない場合：円の内側に線分は存在しない。図 2.6(a)の B の場合。線分

と円の処理終わり。

2b-b. 交点が一つの場合：線分のうち、円周との交点と円の内側にある端との間は描かない。図 2.6(a)の C の場合。線分と円の処理終わり。

2b-c. 交点が二つの場合：線分の両端は円の外にあるので、交点と交点との間は描かない。線分は二つに分かれる。図 2.6(a)の D の場合。線分と円の処理終わり。

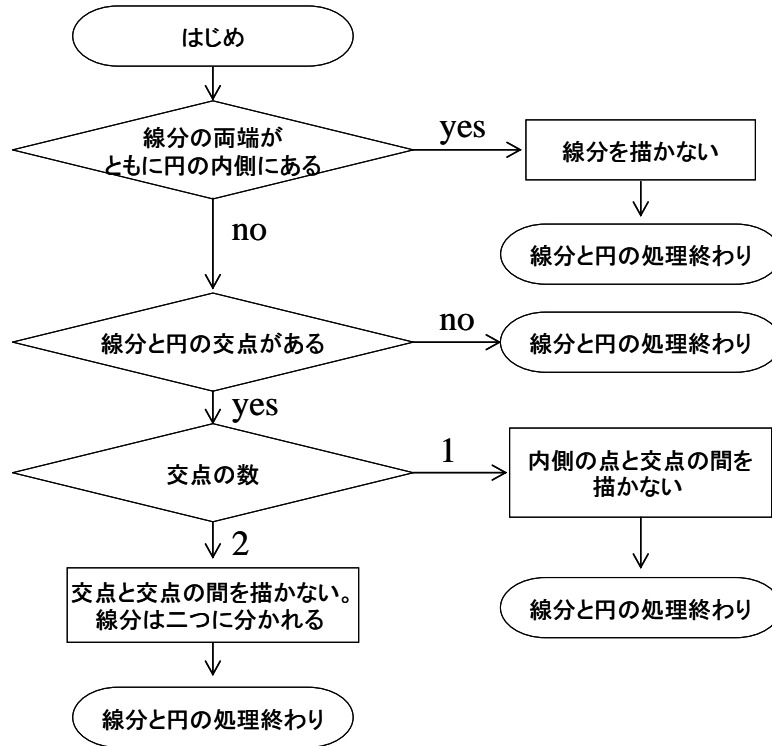


図 2.7 円の内側の線分に関する処理のフローチャート。

### 2. 2. 2 台形の内側の線分

線分のうち、台形の内側は描かない。線分が台形を構成する辺の一つなら、この判別は行わない。線分と台形の交点の数による分類を先に行うアルゴリズムを説明する（サンプルプログラムも同様）。図 2.8 にフローチャートを示す。

1. 線分と台形の 4 辺との交点を求める。線分と線分との交点を求める方法は 3. 2 を参照。
- 2a. 交点有二つの場合：交点と交点との間は描かない。線分は二つに分かれる。図 2.6(b)の D の場合。線分と台形の処理終わり。
- 2b. 交点が一つの場合：線分のうち、台形との交点と台形の内側にある端との間は描かない。図 2.6(b)の C の場合。線分と台形の処理終わり。
- 2c. 交点がない場合：線分の始点・終点が台形の内側にあるかどうかを判定する。方法は 3. 3 を参照。
  - 2c-a 両端とも台形の内側にある場合：この線分は描かない。図 2.6(b)の A の場合。線分と台形の処理終わり。

2c-b 両端とも台形の外側にある場合：台形の内側に線分は存在しない。図 2.6(b) の B の場合。線分と台形の処理終わり。

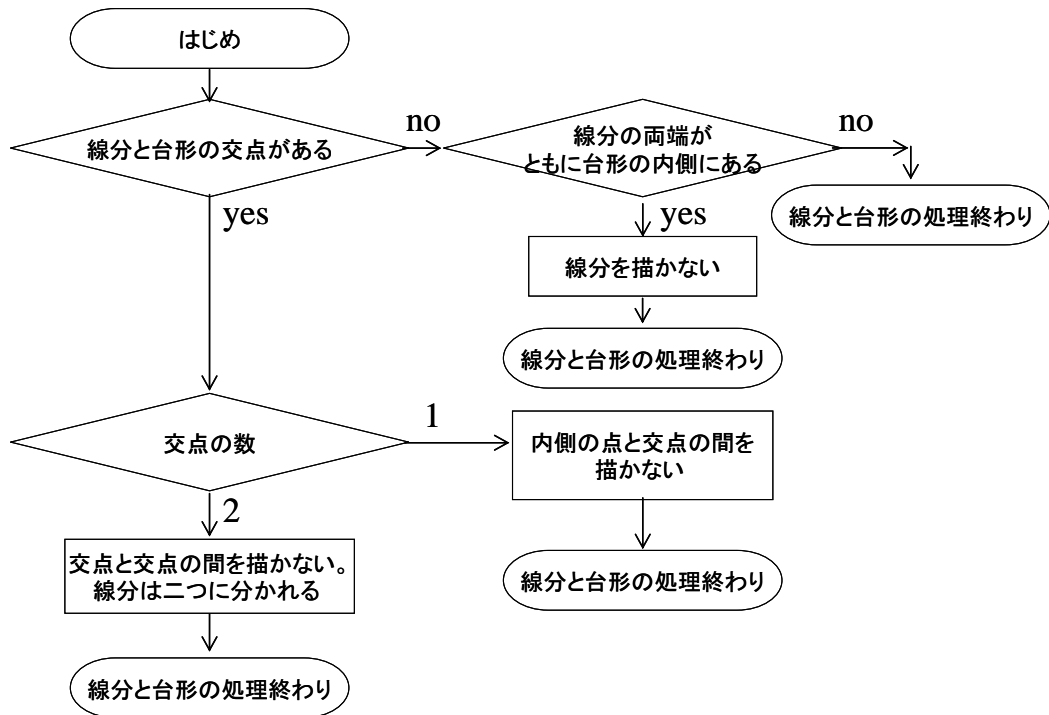


図 2.8 台形の内側の線分に関する処理のフローチャート。

### 3. 円弧や線分を描く/描かないを判別するために必要な主要な計算

線分が y 軸に平行な場合などを別個に取り扱わなくてもすむように、ベクトルで計算する。以下、ベクトルを  $\langle OP \rangle$  のように  $\langle \quad \rangle$  で表す。

#### 3. 1 線分と円との交点を求める

円の中心を  $O$ ，線分を  $PQ$  とした時、円と線分との交点  $R$  を求める (図 3.1)。

$\langle OR \rangle = a \langle OP \rangle + b \langle OQ \rangle$  とおいた時、 $a + b = 1, 0 < a < 1, 0 < b < 1$  であり、 $\langle OR \rangle$  の長さが円の半径  $r$  に等しいような  $a, b$  が求まれば、 $R$  は線分  $PQ$  と円との交点である。

円の中心  $O$  を原点とし、 $P, Q$  の座標をそれぞれ  $(x_p, y_p), (x_q, y_q)$  とする。

$$\begin{aligned} & [\langle OR \rangle = a \langle OP \rangle + b \langle OQ \rangle] \\ & [\langle OR \rangle \text{ の } x \text{ 成分} = a x_p + b x_q] \\ & [\langle OR \rangle \text{ の } y \text{ 成分} = a y_p + b y_q] \\ & [|\langle OR \rangle|^2 = r^2 = (a x_p + b x_q)^2 + (a y_p + b y_q)^2] \end{aligned}$$

$b=1-a$  を代入すると  $a$  の 2 次方程式となる。

$$[A a^2 + 2B a + C = 0]$$

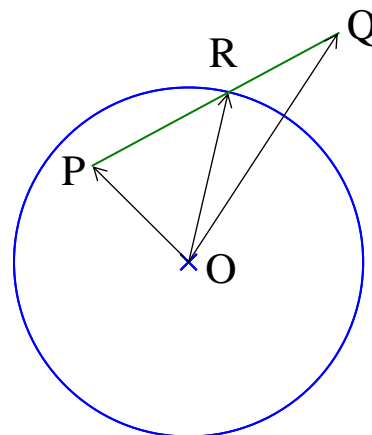


図 3.1 線分  $PQ$  と円との交点  $R$ 。



ここで、

$$\begin{aligned} [A &= (x_p - x_q)^2 + (y_p - y_q)^2] \\ [B &= x_q(x_p - x_q) + y_q(y_p - y_q)] \\ [C &= x_q^2 + y_q^2 - r^2] \end{aligned}$$

この2次方程式の判別式  $D$  は

$$[D = B^2 - AC = \{ (x_p - x_q)^2 + (y_p - y_q)^2 \} r^2 - \{ x_p y_q - x_q y_p \}^2]$$

と表せる。

$D$  が正であれば、線分を含む直線と円が交点を持つ。このとき、解は

$$[a = \{ -B \pm \sqrt{D} \} / A]$$

と表される。解  $a$  が  $0 < a < 1$  を満たせば、 $R$  は線分  $PQ$  上にあり、線分  $PQ$  と円は交点を持つ。

### 3. 2 線分と線分との交点を求める

線分  $OA$  と線分  $PQ$  の交点  $R$  を求める (図 3.2)。

$\langle OR \rangle = a \langle OA \rangle = b \langle OP \rangle + (1 - b) \langle OQ \rangle$  と表した時、 $0 < a < 1, 0 < b < 1$  を満たす  $a, b$  が存在すれば、線分  $OA$  と線分  $PQ$  は交点  $R$  を持つ。

$O, A, P, Q$  の  $x, y$  座標をそれぞれ  $(x_o, y_o), (x_a, y_a), (x_p, y_p), (x_q, y_q)$  とする。

$$[\langle OR \rangle \text{ の } x \text{ 成分} = a(x_a - x_o) = b(x_p - x_o) + (1 - b)(x_q - x_o)]$$

$$[\langle OR \rangle \text{ の } y \text{ 成分} = a(y_a - y_o) = b(y_p - y_o) + (1 - b)(y_q - y_o)]$$

から、次のような  $a$  と  $b$  の連立1次方程式ができる。

$$[a(x_a - x_o) = b(x_p - x_o) + (x_q - x_o)]$$

$$[a(y_a - y_o) = b(y_p - y_o) + (y_q - y_o)]$$

これを解いて

$$[a = \{ (y_p - y_q)x_o - (y_p - y_o)x_q + (y_q - y_o)x_p \} / \{ (y_a - y_o)(x_p - x_q) - (x_a - x_o)(y_p - y_q) \}]$$

$$[b = \{ (y_a - y_q)x_o - (y_a - y_o)x_q + (y_q - y_o)x_a \} / \{ (y_a - y_o)(x_p - x_q) - (x_a - x_o)(y_p - y_q) \}]$$

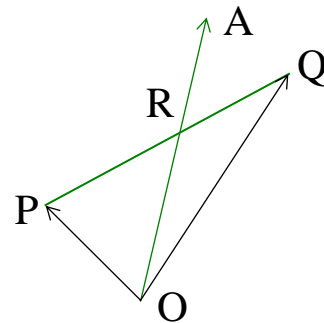


図 3.2 線分  $PQ$  と線分  $OA$  との交点  $R$ .

解  $a, b$  が  $0 < a < 1, 0 < b < 1$  を満たせば二つの線分  $OA$  と  $PQ$  は交わる。

### 3. 3 ある点が台形の内側にあるかどうかの判定

点  $R$  が台形  $ABCD$  の中にあるかどうかを判定する (図 3.3)。台形は三角形  $ABC$  と三角形  $ACD$  に分けられる。点  $R$  がこれらの三角形の内側にあるかどうかを判定すればよい。ここで四角形  $ABCD$  はへこみのない四角形 (内角が常に  $180$  度よりも小さい) であれば良く、二辺が平行であるという台形の性質は必要ない。

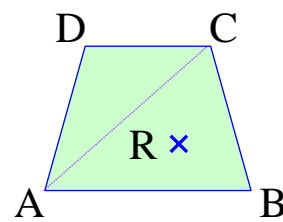


図 3.3 台形  $ABCD$  の中にある点  $R$ 。

点  $R$  が三角形  $OPQ$  の内側にあるかどうかを判定する (図 3.4)。

$\langle OR \rangle = a \langle OP \rangle + b \langle OQ \rangle$  と表した場合、 $a > 0, b > 0, a + b < 1$  ならば、点  $R$  は三角形  $OPQ$  の内側にある。簡単のため、 $O$  を原点に取る。 $P, Q, R$  の  $x, y$  座標をそれぞれ  $(x_p, y_p), (x_q, y_q), (x_r, y_r)$  とする。

$$[\langle OR \rangle = a \langle OP \rangle + b \langle OQ \rangle]$$

次のような  $a$  と  $b$  の連立 1 次方程式ができる。

$$\langle OR \rangle \text{ の } x \text{ 成分 } [x_r = a x_p + b x_q]$$

$$\langle OR \rangle \text{ の } y \text{ 成分 } [y_r = a y_p + b y_q]$$

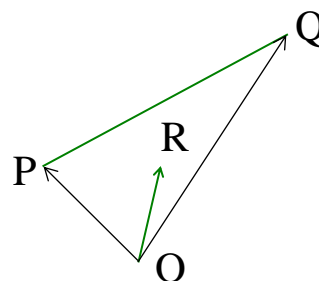


図 3.4 三角形  $OPQ$  の中にある点  $R$ 。

$$[a = \{ y_r x_q - x_r y_q \} / \{ x_q y_p - x_p y_q \}]$$

$$[b = \{ x_r y_p - y_r x_p \} / \{ x_q y_p - x_p y_q \}]$$

解  $a, b$  が  $a > 0, b > 0, a + b < 1$  を満たせば点  $R$  は三角形  $OPQ$  の内側にある。

## 4. 円・線分の描かない条件を集積していく (描く部分だけ残していく) 方法

### 4. 1 円

円のうち、描かない部分を集積していく。

それぞれの円で描く弧を、弧の始点と終点の角度  $(\theta_1^S, \theta_1^E), \dots, (\theta_N^S, \theta_N^E)$  の  $N$  組で表す (図 4.1)。下付の添え字の数字または  $i, j, n, N$  などは弧の番号、上付の添え字の  $S, E$  はそれぞれ弧の始点、終点を示す。常に

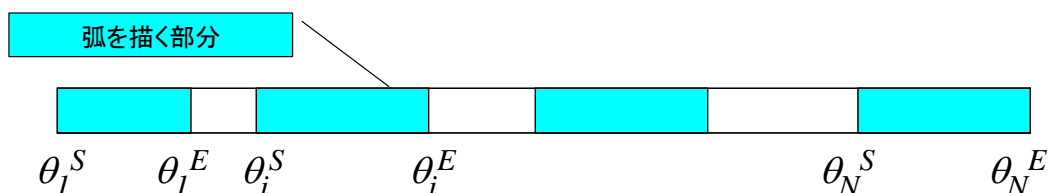


図 4.1 円周のうち、弧を描く部分と描かない部分。

$$0 \leq \theta_1^S < \theta_1^E < \dots < \theta_N^S < \theta_N^E \leq 360$$

が成り立つような順序にする。0度をまたぐ弧は $(0, \theta_1^E), (\theta_N^S, 360)$ の二つに分ける。

「2. 1 円弧の判別法」に基づいて、円周の $\phi^S$ から $\phi^E$ まで( $\phi^S < \phi^E$ )の角度を描かないことが分かった時、これを $(\theta_i^S, \theta_i^E), i=1 \sim N$ に反映させる。

$\phi^S, \phi^E$ が弧を描く部分かどうか( $\phi^S, \phi^E$ と $\theta_i^S, \theta_i^E$ の大小)によって、次の4つに場合に分ける。

- (1)  $\phi^S, \phi^E$ ともに弧を描く部分
- (2)  $\phi^S$ は弧を描く部分、 $\phi^E$ は弧を描かない部分
- (3)  $\phi^S$ は弧を描かない部分、 $\phi^E$ は弧を描く部分
- (4)  $\phi^S, \phi^E$ ともに弧を描かない部分

(1)  $\phi^S, \phi^E$ ともに弧を描く部分： $\theta_i^S < \phi^S < \theta_i^E, \theta_j^S < \phi^E < \theta_j^E$

(1-a)  $i=j$ の場合 (図 4.2)

弧 $(\theta_i^S, \theta_i^E)$ を弧 $(\theta_i^S, \phi^S)$ と弧 $(\phi^E, \theta_i^E)$ に分割する。プログラムでは、 $\theta_i^E = \phi^S, \theta_{i+1}^S = \phi^E, \theta_{i+1}^E = \theta_i^E$ とし、 $n$ が1から $N-i-1$ まで、 $\theta_{i+n+1}^S = \theta_{i+n}^S, \theta_{i+n+1}^E = \theta_{i+n}^E$ のように一つずつずらす。(実際には $n$ の大きい順にずらした後に $\theta_i^E = \phi^S, \theta_{i+1}^S = \phi^E$ を代入する)

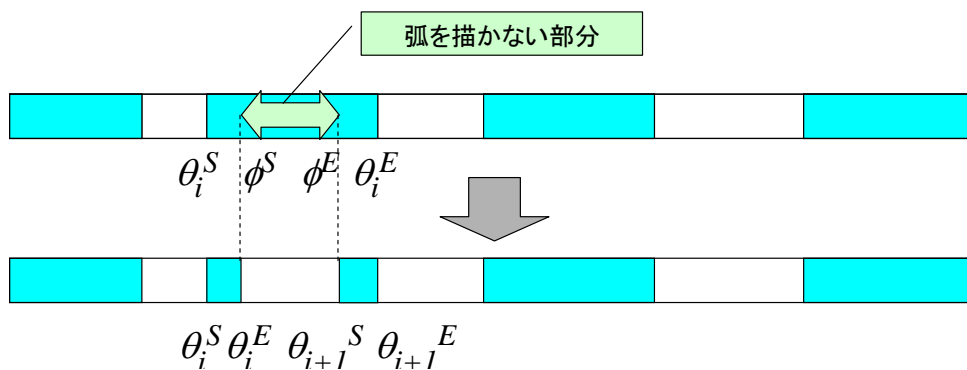


図 4.2  $\phi^S, \phi^E$ ともに弧を描く部分で  $i=j$  の場合。

(1-b)  $i \neq j$ の場合 (図 4.3)

弧 $(\theta_i^S, \theta_i^E)$ を $(\theta_i^S, \phi^S)$ に、弧 $(\theta_j^S, \theta_j^E)$ を $(\phi^E, \theta_j^E)$ に短くする。 $i < n < j$ になるような弧 $(\theta_n^S, \theta_n^E)$ は描かない。プログラムでは、 $\theta_i^E = \phi^S, \theta_{i+1}^S = \phi^E, \theta_{i+1}^E = \theta_j^E$ とし、 $n$ が1から $N-j$ まで、 $\theta_{i+n+1}^S = \theta_{j+n}^S, \theta_{i+n+1}^E = \theta_{j+n}^E$ のようにずらす。

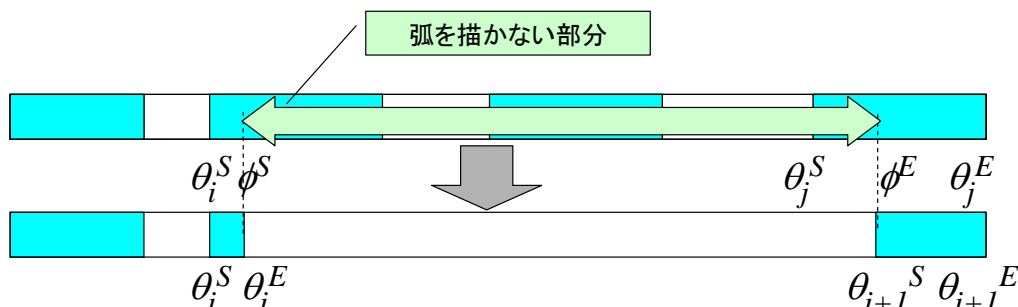


図 4.3  $\phi^S, \phi^E$ ともに弧を描く部分で  $i \neq j$  の場合。

(2)  $\phi^S$ は弧を描く部分、 $\phi^E$ は弧を描かない部分： $\theta_i^S < \phi^S < \theta_i^E, \theta_j^E < \phi^E < \theta_{j+1}^S$  (図 4.4)  
 弧  $(\theta_i^S, \theta_i^E)$  を  $(\theta_i^S, \phi^S)$  に短くし、 $i < n \leq j$  になるような弧  $(\theta_n^S, \theta_n^E)$  は描かない。  
 プログラムでは、 $\theta_i^E = \phi^S$  とし、 $n$  が 1 から  $N-j$  まで、 $\theta_{i+n}^S = \theta_{j+n}^S, \theta_{i+n}^E = \theta_{j+n}^E$  のようにずらす。

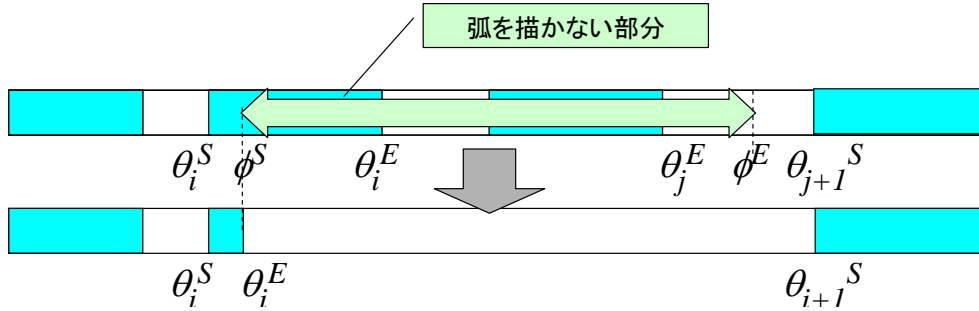


図 4.4  $\phi^S$ は弧を描く部分、 $\phi^E$ は弧を描かない部分.

(3)  $\phi^S$ は弧を描かない部分、 $\phi^E$ は弧を描く部分： $\theta_i^E < \phi^S < \theta_{i+1}^S, \theta_j^S < \phi^E < \theta_j^E$  (図 4.5)  
 弧  $(\theta_j^S, \theta_j^E)$  を  $(\phi^E, \theta_j^E)$  に短くし、 $i < n < j$  になるような弧  $(\theta_n^S, \theta_n^E)$  は描かない。  
 プログラムでは、 $\theta_{i+1}^S = \phi^E, \theta_{i+1}^E = \theta_j^E$  とし、 $n$  が 1 から  $N-j$  まで、 $\theta_{i+n+1}^S = \theta_{j+n}^S, \theta_{i+n+1}^E = \theta_{j+n}^E$  のようにずらす。

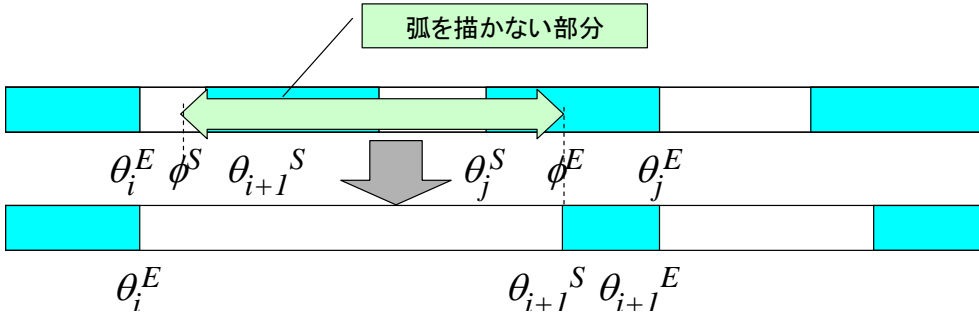


図 4.5  $\phi^S$ は弧を描かない部分、 $\phi^E$ は弧を描く部分.

(4)  $\phi^S, \phi^E$ ともに弧を描かない部分： $\theta_i^E < \phi^S < \theta_{i+1}^S, \theta_j^E < \phi^E < \theta_{j+1}^S$  (図 4.6)  
 $i < n \leq j$  になるような弧  $(\theta_n^S, \theta_n^E)$  は描かない。プログラムでは、 $n$  が 1 から  $N-j$  まで、 $\theta_{i+n}^S = \theta_{j+n}^S, \theta_{i+n}^E = \theta_{j+n}^E$  のようにずらす。

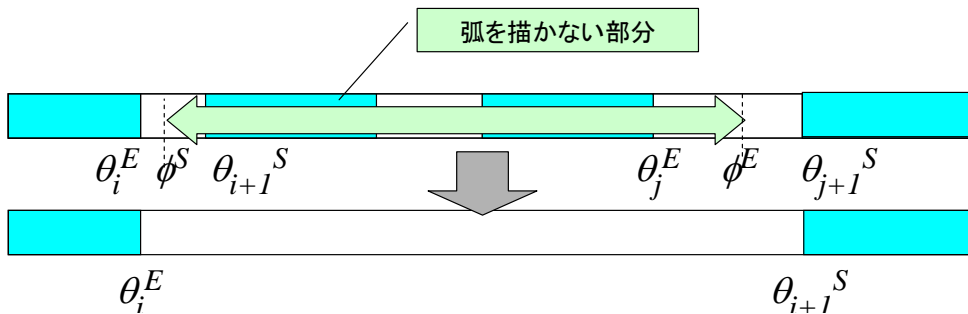


図 4.6  $\phi^S, \phi^E$ ともに弧を描かない部分.

#### 4. 2 線分

線分 AB は両端 A, B を示す二組の  $x, y$  座標  $(x_a, y_a), (x_b, y_b)$  で定義される。交点 P, Q の座標をそれぞれ  $(x_p, y_p), (x_q, y_q)$  とする。

線分の一部を消す処理を次の 2 つに場合分けする。

- (1) 線分の片方の端が台形または円の内部にあり、他方の端が外部にある場合（一つ交点 P がある）
- (2) 線分の両端は台形または円の外部にあるが、線分の途中が台形または円の内部にある場合（二つの交点 P と Q がある）

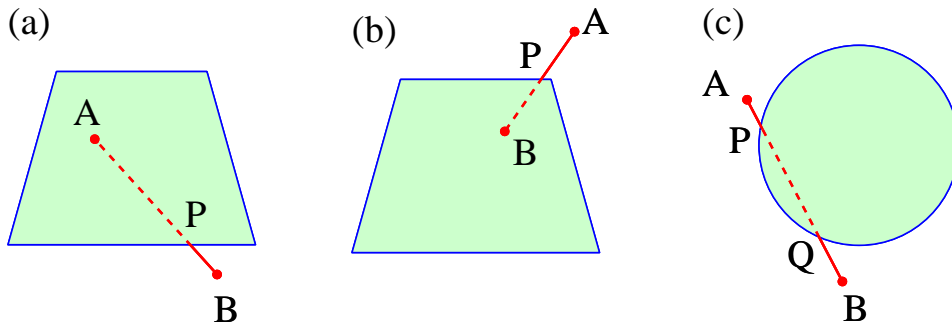


図 4.7 線分と台形または円との相対位置.

(1) 線分の片方の端が台形または円の内部にあり、他方の端が外部にある場合。

(1-a) 点 A が台形または円の内部にある場合（図 4.7 (a)）

線分 AB を線分 PB に短くする。プログラムでは  $(x_a, y_a)$  を  $(x_p, y_p)$  に置き換える。

(1-b) 点 B が台形または円の内部にある場合（図 4.7 (b)）

線分 AB を線分 AP に短くする。プログラムでは  $(x_b, y_b)$  を  $(x_p, y_p)$  に置き換える。

(2) 線分の両端は台形または円の外部にあるが、線分の途中が台形または円の内部にある場合（図 4.7 (c)）

$x_a \neq x_b$  の場合、 $x_a < x_p < x_q < x_b$ ,

$x_a = x_b$  の場合、 $y_a < y_p < y_q < y_b$  とする。

元の線分 AB を  $(x_a, y_a), (x_p, y_p)$  を両端とする線分 AP と  $(x_q, y_q), (x_b, y_b)$  を両端とする線分 QB の二つの線分に分ける。

暴風域の通過する線を描画するために作成したサンプルプログラム

注意点

本プログラムの全部又は一部を利用することは問題ありません。  
 但し、気象庁が本プログラムの動作を保証するものではありませんし、  
 本プログラムの一部又は全部を利用したことにより、  
 利用者が被った直接的又は間接的ないかなる損害についても、  
 気象庁は一切責任を負いません。  
 また、本プログラムに関する個別の対応は行いかねますので、  
 ご注意下さい。

いくつかの円（中心位置と半径）を与えられた時、  
 これらの円とその接線のうち、一番外側を構成する線分・弧の要素  
 （線分なら始点・終点の座標、  
 弧なら元の円の中心位置と半径と始点・終点の角度）  
 を計算するfortranプログラム

H18.11.01. 気象庁予報部予報課太平洋台風センター

H30.07.03改訂 気象庁予報部予報課アジア太平洋気象防災センター

```

program eggplant
PARAMETER (MXC=11, MXC1=MXC-1,
&          MACX=MXC*2, MXCMP=MXC*4, MXLN=MXC*4 )
PARAMETER (MXSG=12)
*   MXC: 円の数の最大値
*   MXC1: 台形の数の最大値
*   MXCX: 最終製品でプロットする弧の数の最大値
*   MXCMP: 最終製品でプロットする線分・弧の数の合計の最大値
*   MXLN: プロットする線分の数；
*           製品ではなく作業途中の状態も含む最大値
*   MXSG: 一つの円を弧に分割した場合のそれぞれの弧の両端の数；
*           製品ではなく作業途中の状態も含む最大値

*   円の中心位置(XC,YC), 半径(RC)
REAL XC(MXC), YC(MXC), RC(MXC)

*   共通外接線(XA1, YA1)-(XB1, YB1), (XA2, YA2)-(XB2, YB2)
REAL XA1(MXC1), YA1(MXC1), XB1(MXC1), YB1(MXC1),
&    XA2(MXC1), YA2(MXC1), XB2(MXC1), YB2(MXC1)

*   台形の頂点
REAL TRPZ(4,2,MXC1)
*   1つめの添え字で頂点を区別、2つ目の添え字で x,y を区別
*   TRPZ(1), TRPZ(2), TRPZ(3), TRPZ(4)    A1, B1, B2, A2

*   それぞれの円でプロットする区間の角度、作業用配列
REAL DEGPLT(MXSG, MXC), DEGPLTWK(4,2,MXC), DEGWK(2,6)

*   プロットする線分（接点と接点を結ぶ線分の全部or一部）
    
```

```

* 線分Mの両端の(X,Y):
*      (DLNX(1,M),DLNY(1,M)),(DLNX(2,M),DLNY(2,M))
*      (DLNX(1,M)<DLNX(2,M)
REAL DLNX(2,MXLN), DLNY(2,MXLN)
* 線分を引くかどうかのフラッグ(LONOFF=0/1 で引く/引かない)
* 線分がどの円の接線か
INTEGER LONOFF(MXLN), NTAN(2,MXLN)

```

```

* プロットする弧
* 弧Mの両端の(X,Y)
*      (ARCX(1,M),ARCY(1,M)),(ARCX(2,M),ARCY(2,M))
* 弧Mの両端の角度
*      ARCD(1,M),ARCD(2,M)
*      ARCD(1,M)<ARCD(2,M) で角度が増える方向に描く
* 弧Mのもとの円の中心の位置(x,y)と半径 r
REAL ARCX(2,MACX),ARCY(2,MACX), ARCD(2,MACX),
& ARCCRC(3,MACX)
* 弧Mのもとの円の番号
INTEGER MARC(MACX)

```

```

* プロットする線分・弧の終点/始点がつながる順番などを示す
INTEGER JUN(3,MXCMP)

```

```
DATA VALNO / -99.E9/
```

```

* -----
* 【入力データ：円の中心位置の座標、半径、円の数】
* -----

```

```

CALL CIRCLDATA (XC,YC,RC,NFC, MXC)
NFC1=NFC-1
      write(6,'(/4x, 60(1H=) //4x,a8, i3)') '<CIRCLE>', NFC
WRITE(6,6510) 'XC', (XC(I),I=1,NFC)
WRITE(6,6510) 'YC', (YC(I),I=1,NFC)
WRITE(6,6510) 'RC', (RC(I),I=1,NFC)
6510 FORMAT (4X,A2, 12F7.1)

```

```

* -----
* 【接点を求める】
* -----

```

```

      write(6,'(/4x,a18/)') '<POINT OF CONTACT>'
CALL SETTEN ( XA1,YA1,XB1,YB1, XA2,YA2,XB2,YB2,
&            XC,YC,RC, VALNO, NFC,NFC1 )
DO 110 N = 1,NFC1
      WRITE(6,6520) N, 'x', XA1(N),XB1(N), XA2(N),XB2(N),
&            'y', YA1(N),YB1(N), YA2(N),YB2(N)
110 CONTINUE
6520 FORMAT(1X,I3,7X, 'A1      B1      A2      B2',
&          2 (/1X,A6,4F8.2) )
      write(6,'(/4x, 30(2H -) /)')

```

```

* -----
* 【線分の処理】

```

```

* -----
CALL SETLINE ( TRPZ, DLNX,DLNY, LONOFF, NTAN, NPLN,
&                XA1,YA1,XB1,YB1, XA2,YA2,XB2,YB2,
&                VALNO, NFC1, MXLN )
DO 120 L = 1,NPLN
WRITE(6,6790) L, (DLNX(I,L),DLNY(I,L),I=1,2), LONOFF(L)
120 CONTINUE
6790 FORMAT(2X,'line', I3, 2X,2( '(' ,2F7.1,' ) ' ),I4 )

```

```

* < 円の中の線分 >
write(6,'(/4x,a16)') '<line in circle>'
CALL LINECIRCL ( DLNX,DLNY, LONOFF, NTAN, NPLN,
&                XC,YC,RC,    NFC, MXLN )
DO 130 L = 1,NPLN
WRITE(6,6790) L, (DLNX(I,L),DLNY(I,L),I=1,2), LONOFF(L)
130 CONTINUE

```

```

* < 台形の中の線分 >
write(6,'(/4x,a19/)') '<line in trapezoid>'
CALL LINETRPZ ( DLNX,DLNY, LONOFF, NTAN, NPLN,
&                TRPZ, VALNO, MXLN,MXC1,NFC1 )

DO 140 L = 1,NPLN
WRITE(6,6790) L, (DLNX(I,L),DLNY(I,L),I=1,2), LONOFF(L)
140 CONTINUE
write(6,'(/4x, 30(2H -) /)')

```

```

* -----
* 【円の処理】
* -----

```

```

CALL SETDEGPLT (DEGPLT, MXSG,MXC)
write(6,'(/4x,a18)') '<neighbouring arc>'
* < 隣り合った円の向かい合った部分の弧 >
CALL SETDEGPLTWK (DEGPLTWK, 8*MXC)
CALL SETDEGPLTWK (DEGWK, 12)
CALL DELARC ( DEGPLT, DEGPLTWK, DEGWK,
&            XA1,YA1, XA2,YA2, XC,YC,RC, VALNO, NFC,NFC1, MXSG )

DO 160 N = 1,NFC
160 WRITE(6,6640) N, (DEGPLT(I,N),I=1,6)
6640 FORMAT(2X,I3,4 ( '(' , 2F7.1,' ) ' ) )

```

```

* < 重なる円 >
write(6,'(/4x,a20/)') '<overlapping circle>'
CALL SETDEGPLTWK (DEGWK, 12)
CALL OVERLAP ( DEGPLT, DEGWK,
&            XC,YC,RC, MXSG,NFC,NFC1 )
DO 170 N = 1,NFC
170 WRITE(6,6640) N, (DEGPLT(I,N),I=1,6)

```

```

* < 台形の中の円弧 >
write(6,'(/4x,a20/)') '<arc in trapezoids>'

```



```
* subroutine SETLINE has to be called prior to TRPZCRCL.
* TRPZ is used in subroutine TRPZCRCL.
CALL SETDEGPLTWK (DEGWK, 12)
CALL TRPZCRCL ( DEGPLT, DEGWK,
& TRPZ, XC, YC, RC, VALNO, NFC,NFC1,MXSG )
```

```
DO 180 N = 1,NFC
180 WRITE(6,6640) N, (DEGPLT(I,N),I=1,6)
```

```
* <円弧のパラメタ取り出し>
CALL DEG2ARC ( ARCX,ARCY, ARCD, ARCCRC, MARC, NARC,
& XC,YC,RC, DEGPLT, NFC, MXSG, MACX )
```

```
* -----
* 【一覽出力】
* -----
```

```
WRITE(6,6248)
6248 FORMAT(/2x, 90(1H=)//15X,2( '( x y ) ' ) )
```

```
DO 150 L = 1,NPLN
IF(LONOFF(L).NE.0)
& WRITE(6,6255) L, (DLNX(I,L),DLNY(I,L),I=1,2)
150 CONTINUE
6255 FORMAT(6X,'line', I3, 2X,2( '(' ,2F7.1,' ) ' ) )
```

```
WRITE(6,6245)
6245 FORMAT(/15X,2( '( x y ) ' ),': (deg start-end )',
& 14X, '(center x y radius)' )
DO 190 N = 1,NARC
WRITE(6,6265) N, (ARCX(I,N),ARCY(I,N),I=1,2),
& (ARCD(I,N),I=1,2), MARC(N),(ARCCRC(I,N),I=1,3)
190 CONTINUE
6265 FORMAT(6X,'arc ', I3, 2X,2( '(' ,2F7.1,' ) ' ),
& ': (' ,2F7.1,' ) of circle', I3,' (' ,3F7.1,' ) ' )
```

```
* -----
* 【描く要素のまとめ】
* -----
```

```
write(6,'(/2x, 90(1H=) /)')
IF(NARC.EQ.1) THEN
LALL=0
DO 195 L = 1,NPLN
LALL=LALL+LONOFF(L)
195 CONTINUE
IF(LALL.EQ.0) THEN
WRITE(6,*) ' ++ only one circle is plotted ++'
ELSE
WRITE(6,*) ' ?? one arc and (a) line segment(s) ??', LALL
ENDIF
ELSE
CALL SNGLSTRK ( JUN, NCMP,
```

```

&          DLNX,DLNY, LONOFF, NPLN,
&          ARCX,ARCY, ARCD, ARCCRC, NARC, MXCMP )

CALL PRNTCMP ( JUN, NCMP,
&          DLNX,DLNY, NPLN,
&          ARCX,ARCY, ARCD, ARCCRC, MARC, NARC )
* IF(NARC.EQ.1) THEN --> ELSE --> ENDIF
ENDIF

                                write(6,'(/4x,a8)') '<circle>'
WRITE(6,6510) 'XC', (XC(I),I=1,NFC)
WRITE(6,6510) 'YC', (YC(I),I=1,NFC)
WRITE(6,6510) 'RC', (RC(I),I=1,NFC)
                                write(6,'(/2x, 90(1H=)  /)')

STOP
END

* =====

SUBROUTINE SETTEN ( XA1,YA1,XB1,YB1, XA2,YA2,XB2,YB2,
&          XC,YC,RC, VALNO, NFC,NFC1 )

* < 接点を求める >
* 円の中心位置(XC,YC), 半径(RC)
REAL XC(NFC), YC(NFC), RC(NFC)

* 共通外接線(XA1,YA1)-(XB1,YB1), (XA2,YA2)-(XB2,YB2),
REAL XA1(NFC1), YA1(NFC1), XB1(NFC1), YB1(NFC1),
& XA2(NFC1), YA2(NFC1), XB2(NFC1), YB2(NFC1)

DO 100 N=1,NFC1
  DST=SQRT((XC(N+1)-XC(N))**2+(YC(N+1)-YC(N))**2)
  AR=(RC(N+1)-RC(N))/DST
* H30.07.03 start
  IF(AR.GT.-1. .AND. AR.LT.1.
& .AND. RC(N).GT.0. .AND. RC(N+1).GT.0. ) THEN
* H30.07.03 end
* - - 一方の予報円を原点に、
* 他方の予報円の中心をX軸上に移動した場合の接点の座標
COAR=SQRT(1.-AR*AR)
XAO=-AR*RC(N)
YAO=RC(N)*COAR
XBO=DST-AR*RC(N+1)
YBO=RC(N+1)*COAR
*
* - - -
* XA1(N)=XAO
* YA1(N)=YAO
* XA2(N)=XAO
* YA2(N)=-YAO
* XB1(N)=XBO
* YB1(N)=YBO
* XB2(N)=XBO

```

```

*          YB2(N)=-YB0
*
* - - 座標軸回転
SINT=(YC(N+1)-YC(N))/DST
COST=(XC(N+1)-XC(N))/DST
XA1(N)=+XA0*COST-YA0*SINT
YA1(N)=+XA0*SINT+YA0*COST
XA2(N)=+XA0*COST+YA0*SINT
YA2(N)=+XA0*SINT-YA0*COST
XB1(N)=+XB0*COST-YB0*SINT
YB1(N)=+XB0*SINT+YB0*COST
XB2(N)=+XB0*COST+YB0*SINT
YB2(N)=+XB0*SINT-YB0*COST

* - - 座標軸平行移動
XA1(N)=XA1(N)+XC(N)
YA1(N)=YA1(N)+YC(N)
XA2(N)=XA2(N)+XC(N)
YA2(N)=YA2(N)+YC(N)
XB1(N)=XB1(N)+XC(N)
YB1(N)=YB1(N)+YC(N)
XB2(N)=XB2(N)+XC(N)
YB2(N)=YB2(N)+YC(N)
CXX          WRITE(6,6610) N,  XA1(N),XB1(N), XA2(N),XB2(N),
CXX          &                  YA1(N),YB1(N), YA2(N),YB2(N)
CXX 6610     FORMAT(1X,I3, '  A1   B1   A2   B2', 2 (/4X,4F7.2) )
          ELSE
          WRITE(6, '(3X,I3,A10)') N, ' no point'
          XA1(N)=VALNO
          YA1(N)=VALNO
          XA2(N)=VALNO
          YA2(N)=VALNO
          XB1(N)=VALNO
          YB1(N)=VALNO
          XB2(N)=VALNO
          YB2(N)=VALNO
          ENDIF
100 CONTINUE
RETURN
END

* =====

SUBROUTINE DELARC ( DEGPLT, DEGPLTWK, DEGWK,
&                XA1,YA1, XA2,YA2, XC,YC,RC,  VALNO, NFC,NFC1, MXSG )

* < 隣り合った円の向かい合った部分の弧 >
* それぞれの円でプロットする区間の角度、作業用配列
REAL DEGPLT(MXSG,NFC),  DEGPLTWK(4,2,NFC), DEGWK(2,6)

* 円の中心位置(XC,YC), 半径(RC)

```

```

REAL XC(NFC), YC(NFC), RC(NFC)
*  共通外接線の接点(XA1, YA1), (XA2, YA2)
REAL XA1(NFC1), YA1(NFC1),
&    XA2(NFC1), YA2(NFC1)

*  -----
CALL DELKO ( DEGPLTWK,
&          XA1, YA1, XA2, YA2, XC, YC, RC,  VALNO, NFC, NFC1 )
*  -----
                                                    write(6,*)

DO  210  N = 1, NFC
    WRITE(6, 6620) N, ((DEGPLTWK(I, J, N), I=1, 4), J=1, 2)
210  CONTINUE
6620 FORMAT (5X, 'circle', I2, '/ noplot:', 2(2('(', 2F6.1, ') '), 2X) )

*  【 描かない弧を示す角度の配列を分割 】
DO  220  N = 1, NFC
    LL=0
    DO  240  J = 1, 2
    DO  240  I = 1, 2
        II=I*2-1
        IF(DEGPLTWK(II, J, N).LT.360.) THEN
            LL=LL+1
            DEGWK(1, LL)=DEGPLTWK(II, J, N)
            DEGWK(2, LL)=DEGPLTWK(II+1, J, N)
        ENDIF
240  CONTINUE
cx    WRITE(6, 6630) N, LL, ((DEGWK(I, L), I=1, 2), L=1, LL)
cx 6630 FORMAT(1X, 'circle', I2, ':', I2, ' segments w/o plot: ',
cx    &      7( '(', 2F6.1, ') ') )

*  【 描かない弧を示す角度合成 】
DO  260  L = 1, LL
CALL SELSGMNT (DEGPLT(1, N), DEGWK(1, L), MXSG)
260  CONTINUE
220  CONTINUE

RETURN
END

*  --V----X----V----X----V----X----V----X----V----X----V----X----V----X--
SUBROUTINE DELKO ( DEGPLTWK,
&          XA1, YA1, XA2, YA2, XC, YC, RC,  VALNO, NFC, NFC1 )
*  隣り合った円の向かい合った部分の弧
*  それぞれの円でプロットしない区間の角度、作業用配列
REAL DEGPLTWK(4, 2, NFC)

*  円の中心位置(XC, YC), 半径(RC)
REAL XC(NFC), YC(NFC), RC(NFC)
*  共通外接線の接点(XA1, YA1), (XA2, YA2)

```

```

REAL XA1(NFC1), YA1(NFC1),
& XA2(NFC1), YA2(NFC1)

```

```

DO 200 N=1,NFC1

```

```

IF (XA1(N).NE.VALNO)THEN

```

```

CALL XY2DEG ( DEGA1, XA1(N),YA1(N), XC(N),YC(N),RC(N) )

```

```

CALL XY2DEG ( DEGA2, XA2(N),YA2(N), XC(N),YC(N),RC(N) )

```

```

IF(DEGA2.LT.DEGA1) THEN

```

```

    DWK=DEGA1

```

```

    DEGA1=DEGA2

```

```

    DEGA2=DWK

```

```

*      --> DEGA2 is larger than DEGA1

```

```

ENDIF

```

```

CXX write(6,*) DEGA1, DEGA2

```

```

* 隣り合う二つの円で接点の角度は同じ。

```

```

CALL LINEDEG(DEG, XC(N+1),YC(N+1),XC(N),YC(N))

```

```

*      0<DEG<360

```

```

IF (DEG.GE.DEGA1 .AND. DEG.LE.DEGA2) THEN

```

```

    DEGPLTWK(1,2,N)=DEGA1

```

```

    DEGPLTWK(2,2,N)=DEGA2

```

```

CXX WRITE(6,*)

```

```

CXX & ' circle ',N, ' not plot:',DEGA1,' --', DEGA2

```

```

    DEGPLTWK(1,1,N+1)=0.

```

```

    DEGPLTWK(2,1,N+1)=DEGA1

```

```

    DEGPLTWK(3,1,N+1)=DEGA2

```

```

    DEGPLTWK(4,1,N+1)=361.

```

```

CXX WRITE(6,*)

```

```

CXX & ' circle ',N+1,' not plot:0--',DEGA1, DEGA2, ' -- 360'

```

```

ELSE

```

```

    DEGPLTWK(1,2,N)=0.

```

```

    DEGPLTWK(2,2,N)=DEGA1

```

```

    DEGPLTWK(3,2,N)=DEGA2

```

```

    DEGPLTWK(4,2,N)=361.

```

```

CXX WRITE(6,*)

```

```

CXX & ' circle ',N,' not plot:0--',DEGA1, DEGA2, ' -- 360'

```

```

    DEGPLTWK(1,1,N+1)=DEGA1

```

```

    DEGPLTWK(2,1,N+1)=DEGA2

```

```

CXX WRITE(6,*)

```

```

CXX & ' circle ',N+1, ' not plot:',DEGA1,' --', DEGA2

```

```

ENDIF

```

```

* IF (XA1(N).NE.VALNO)THEN ----> ENDIF

```

```

ENDIF

```

```

200 CONTINUE

```

```

RETURN

```

```

END

```

```

* --V---X---V---X---V---X---V---X---V---X---V---X---V---X---

```

```

SUBROUTINE OVERLAP ( DEGPLT, DEGWK,

```

```

&                XC,YC,RC,  MXSG,NFC,NFC1 )
REAL DEGPLT(MXSG,NFC), DEGWK(2,6)
REAL XC(NFC), YC(NFC), RC(NFC)

*   < 重なる円 >
DO 280 I = 1,NFC1
DO 285 J = I+1,NFC
  DST=SQRT((XC(I)-XC(J))**2+(YC(I)-YC(J))**2)
  IF(DST.GE.RC(I)+RC(J)) THEN
*   重ならない : do nothing
    CONTINUE
  ELSE IF(DST+RC(I).LE.RC(J)) THEN
*   円Iは円Jの内部にある : 円Iを描かない
    DEGWK(1,1)=0.
    DEGWK(2,1)=360.
    CALL SELSGMNT (DEGPLT(1,I), DEGWK(1,1), MXSG)
    WRITE(6,6638) I, J, RC(I),RC(J),DST
6638   FORMAT(5X,'circle',I3,' is inside of the circle',I3,
&         ' . ri,rj,dist=',3f6.1)
  ELSE IF(DST+RC(J).LE.RC(I)) THEN
*   円Jは円Iの内部にある : 円Jを描かない
    DEGWK(1,1)=0.
    DEGWK(2,1)=360.
    CALL SELSGMNT (DEGPLT(1,J), DEGWK(1,1), MXSG)
    WRITE(6,6638) J, I, RC(I),RC(J),DST
  ELSE
*   円Iと円Jが重なる ; 交点がある : 相手の内側にある弧を描かない
*   円Iの弧
*   -----
    CALL OVLP (DEGWK(1,1),
&           DST, XC(I),YC(I),RC(I), XC(J),YC(J),RC(J) )
*   -----
    write(6,6635) I,J, ((DEGWK(N,M),N=1,2),M=1,2)
    CALL SELSGMNT (DEGPLT(1,I), DEGWK(1,1), MXSG)
    IF(DEGWK(1,2).LT.360.)
&   CALL SELSGMNT (DEGPLT(1,I), DEGWK(1,2), MXSG)
6635   FORMAT(5X,'circle',2I3, '/ noplot:', 7( '(', 2F6.1,') ' ) )

*   円Jの弧
*   -----
    CALL OVLP (DEGWK(1,3),
&           DST, XC(J),YC(J),RC(J), XC(I),YC(I),RC(I) )
*   -----
    write(6,6635) J,I, ((DEGWK(N,M),N=1,2),M=3,4)
    CALL SELSGMNT (DEGPLT(1,J), DEGWK(1,3), MXSG)
    IF(DEGWK(1,4).LT.360.)
&   CALL SELSGMNT (DEGPLT(1,J), DEGWK(1,4), MXSG)
  ENDIF
285 CONTINUE
280 CONTINUE

```

RETURN  
END

```
* -----  
  
SUBROUTINE OVLP (DEGWK,  
& DST, XCI, YCI, RC1, XCJ, YCJ, RCJ )  
REAL DEGWK(2,2)  
  
* 余弦定理を利用  
* PI=ACOS(-1.)  
R2D=180./ACOS(-1.)  
DEGC=(ACOS((DST**2+RC1**2-RCJ**2)/(2.*DST*RC1)))*R2D  
* 0<DEGC<180  
CALL LINEDEG(DEG, XCJ, YCJ, XCI, YCI)  
* 0<DEG<360  
DEG1=DEG-DEGC  
DEG2=DEG+DEGC  
CXX write(6, '(2x,2f7.1)') DEG1, DEG2  
IF (DEG1.LT.0.) THEN  
* 相手の円の内側にある弧は DEG1--0, 0--DEG2  
DEGWK(1,1)=360.+DEG1  
DEGWK(2,1)=360.  
DEGWK(1,2)=0.  
DEGWK(2,2)=DEG2  
ELSE IF (DEG2.GT.360.) THEN  
* 相手の円の内側にある弧は DEG1--360, 360--DEG2  
DEGWK(1,1)=DEG1  
DEGWK(2,1)=360.  
DEGWK(1,2)=0.  
DEGWK(2,2)=DEG2-360  
ELSE  
* 相手の円の内側にある弧は DEG1--DEG2  
DEGWK(1,1)=DEG1  
DEGWK(2,1)=DEG2  
DEGWK(1,2)=361.  
DEGWK(2,2)=361.  
ENDIF  
CXX write(6, '(2(3x,2f7.1))') DEGWK(1,1), DEGWK(2,1),  
CXX & DEGWK(1,2), DEGWK(2,2)  
RETURN  
END  
  
* --V---X---V---X---V---X---V---X---V---X---V---X---V---X---V---X---
```

```
SUBROUTINE TRPZCRCL ( DEGPLT, DEGWK,  
& TRPZ, XC, YC, RC, VALNO, NFC, NFC1, MXSG )
```

```
PARAMETER (MXP=10)
```

```
* < 台形の中の円弧 >
```

```

* 円の中心位置(XC,YC), 半径(RC)
REAL XC(NFC), YC(NFC), RC(NFC)

* 台形の頂点
REAL TRPZ(4,2,NFC1)
* 1つめの添え字で頂点を区別、2つ目の添え字で x,y を区別
* TRPZ(1),TRPZ(2),TRPZ(3),TRPZ(4) A1,B1,B2,A2

* それぞれの円でプロットする区間の角度、作業用配列
REAL DEGPLT(MXSG,NFC), DEGWK(2,6)

* 円と台形の交点(最大8個)+0.,360.=最大10個
REAL XX(MXP), YY(MXP), DEG(MXP)
INTEGER IORDG(MXP), IRC(MXP)

* PI=ACOS(-1.)
D2R=ACOS(-1.)/180.
D360=360.

DO 600 N = 1,NFC
cx write(6,6720) N,XC(N),YC(N),RC(N)
cx 6720 FORMAT(3X,'circle ', I2,' x,y,r:',3F7.1)
DO 610 M = 1,NFC1
IF (TRPZ(1,1,M).NE.VALNO) THEN
cx write(6,6525) M,'x', (TRPZ(i,1,M),i=1,4),
cx & 'y', (TRPZ(i,2,M),i=1,4)
cx 6525 FORMAT(3x,'quadr.', I3,':', A4,4F8.2/13x,A4,4F8.2)

* 円 N に接する線分を辺とする台形 M は飛ばす
* <隣り合った円の向かい合った部分の弧>で処理
IF ( M.EQ.N .OR. M+1.EQ.N ) THEN
* do nothing
write (6,'(5x,a11,i2,a7,i2)') 'skip circle', n, '- quadr.',m
ELSE
* 台形 M の頂点が円 N の中かどうか判定
INSIDE=0
DO 620 I = 1,4
DST=SQRT((XC(N)-TRPZ(I,1,M))**2+
& (YC(N)-TRPZ(I,2,M))**2)
IF(DST.LE.RC(N)) THEN
INSIDE=INSIDE+1
cx WRITE(6,'(4x,2f7.1,a15,i3)')
cx & TRPZ(I,1,M), TRPZ(I,2,M), ' inside circle ', N
ENDIF
620 CONTINUE
cx write(6,'(5x,a6,i2,a12,i2,a1,i2,a14 )')
cx & 'circle', n, '/quadrangle', m,':',
cx & inside, ' points inside'
IF (INSIDE.EQ.4) THEN
* 台形がすっぽり円の内部にある
* do nothing

```



```

        continue
*   IF (INSIDE.EQ.4) THEN --> ELSE
ELSE
*   台形の辺と円との交点を求める
*   台形の頂点がすべて円の外でも辺と円がまじわることがある。
*   交点は最大8個
CALL QUADCIRCRS (XX,YY,DEG,MCRS,
&                TRPZ(1,1,M), XC(N),YC(N),RC(N), MXP )
IF(MCRS.EQ.0) THEN
CX   write(6,*) '      no crossing point'
CALL INQUAD ( INSIDE, XC(N),YC(N), TRPZ(1,1,M) )
IF (INSIDE.GT.0) THEN
*   円Nはすっぽり台形Mの内部にある：円Nを描かない
        DEGWK(1,1)=0.
        DEGWK(2,1)=360.
        CALL SELSGMNT (DEGPLT(1,N), DEGWK(1,1), MXSG)
        WRITE(6,6638) N, M, XC(N),YC(N),RC(N)
6638   FORMAT(5X,'circle',I3,' is inside of the trapez',I3,
&         ' . x, y, r=',3f6.1)
        ENDIF
ELSE
CX   WRITE(6,'( i6,a16, 8(/5x,a25,3f7.1))')
CX   &   mcrs, ' crossing points', ('circle and side cross at',
CX   &   xx(i),yy(i),deg(i),i=1,mcrs)

*   adding 0. and 360., DEG is sorted in increasing order.
NOO=0
NO360=0
DO 640 I = 1,MCRS
    IF(DEG(I).EQ.0.) NOO=99
    IF(DEG(I).EQ.D360) NO360=99
640 CONTINUE
IF(NOO.EQ.0) THEN
    MCRS=MCRS+1
    DEG(MCRS)=0.
    XX(MCRS)=XC(N)+RC(N)
    YY(MCRS)=YC(N)
ENDIF
IF(NO360.EQ.0) THEN
    MCRS=MCRS+1
    DEG(MCRS)=D360
    XX(MCRS)=XC(N)+RC(N)
    YY(MCRS)=YC(N)
ENDIF

DO 642 I = 1,MXP
    IORDG(I)=0
    IRC(I)=0
642 CONTINUE
III=0
644 CONTINUE

```

```

SMAL=361.
DO 646 I = 1, MCRS
  IF (DEG(I).LT.SMAL .AND. IRC(I).EQ.0) THEN
    SMAL=DEG(I)
    II=I
  ENDIF
646 CONTINUE
  III=III+1
  IORDG(III)=II
  IRC(II)=99
  IF(III.LT.MCRS) GOTO 644

* find whether each arc is inside or outside of the quadrangle
LL=0
DO 650 II = 1, MCRS-1
  I1=IORDG(II)
  I2=IORDG(II+1)
cx   DCT=0.5*(DEG(I1)+DEG(I2))
cx   write(6, '(6x,a4,3f7.2)') 'deg:', DEG(I1), DCT, DEG(I2)
  RCT=(0.5*(DEG(I1)+DEG(I2)))*D2R
  XCT=XC(N)+RC(N)*COS(RCT)
  YCT=YC(N)+RC(N)*SIN(RCT)
cx   write(6, '(6x,a4,3f7.2)') 'x: ', xx(I1), xct, xx(I2)
cx   write(6, '(6x,a4,3f7.2)') 'y: ', yy(I1), yct, yy(I2)
  CALL INQUAD ( INSIDE, XCT, YCT, TRPZ(1,1, M) )
  IF(INSIDE.GT.0) THEN
cx   write(6, '(4x,2f7.1,a54)') DEG(I1), DEG(I2),
cx   &   ': the arc is inside the quadrangle: not to be plotted.'
    LL=LL+1
    DEGWK(1, LL)=DEG(I1)
    DEGWK(2, LL)=DEG(I2)
  ELSE
cx   write(6, '(4x,2f7.1,a54)') DEG(I1), DEG(I2),
cx   &   ': the arc is outside the quadrangle: to be plotted. '
*   do nothing
    CONTINUE
  ENDIF
650 CONTINUE
DO 660 L = 1, LL
  CALL SELSGMNT (DEGPLT(1, N), DEGWK(1, L), MXSG)
660 CONTINUE

WRITE(6, 6635) N, M, ((DEGWK(i, j), i=1, 2), j=1, II)
6635 FORMAT(5X, 'circle', I2, ': trapz', I2,
&         '/ noplot:', 7( '(', 2F6.1, ') ' ) )

* IF(MCRS.EQ.0) THEN --> ELSE --> ENDIF
  ENDIF
* IF (INSIDE.EQ.4) THEN --> ELSE --> ENDIF
  ENDIF
* IF ( M.EQ.N .OR. M+1.EQ.N ) THEN --> ELSE --> ENDIF

```

```

ENDIF
* IF (TRPZ(1,1,M).NE.VALNO) THEN --> ENDIF
ENDIF
610 CONTINUE
600 CONTINUE
RETURN
END

```

\* -----

```

SUBROUTINE QUADCIRCRS (XX,YY,DEG,MCRS,
& TRPZ, XC,YC,RC, MXP )

```

```

* 台形の辺と円との交点を求める
* 交点は最大8個
* 円と台形の交点
REAL XX(MXP), YY(MXP), DEG(MXP)
* 台形の頂点
REAL TRPZ(4,2)
* 台形の辺の両端：作業用
REAL DLX(2), DLY(2)

MCRS=0
DO 630 N = 1,4
IF(N.EQ.4) THEN
N2=1
ELSE
N2=N+1
ENDIF
DLX(1)=TRPZ(N,1)
DLY(1)=TRPZ(N,2)
DLX(2)=TRPZ(N2,1)
DLY(2)=TRPZ(N2,2)
CX write(6,'(5x,a4,2f7.1,2x,2f7.1)') 'side:',(dlx(i),dly(i),i=1,2)
* 円と線分の交点を計算
CALL LINCIRCRS (XX(MCRS+1),YY(MCRS+1),DEG(MCRS+1),NSL,
& DLX,DLY, XC,YC,RC )
CX IF(NSL.GE.1) THEN
CX write(6,*) nsl, ' crossing points'
CX WRITE(6,'(5x,a25,3f7.1)')
CX & ('circle and side cross at',
CX & xx(i),yy(i),deg(i),i=mcrs+1,mcrs+nsl)
CX ENDF
MCRS=MCRS+NSL
630 CONTINUE
CX IF(MCRS.EQ.0) THEN
CX write(6,*) ' no crossing point'
CX ELSE
CX WRITE(6,'( i6,a16,8(/5x,a25,3f7.1))')
CX & mcrs, ' crossing points', ('circle and side cross at',
CX & xx(i),yy(i),deg(i),i=1,mcrs)

```

```

CX      ENDIF
        RETURN
        END

```

```

* --V---X---V---X---V---X---V---X---V---X---V---X---V---X---

```

```

SUBROUTINE SELSGMNT (DEGPLT, DEGWK, MXSG)
  REAL DEGPLT(MXSG), DEGWK(2)
  *   DEGPLT:二つの数字の組でプロットする区間を示す
  *           {DEGPLT(1)-DEGPLT(2)}, {DEGPLT(3)-DEGPLT(4)}, ...
  *           {DEGPLT(2*N-1)-DEGPLT(2*N)} の区間をプロットする。
  *           当然 DEGPT(1)<DEGPLT(2)<DEGPLT(3) ... < DEGPT(2*N)
  *           DEGPLT(1)=0. のこともある。
  *   DEGWK:プロットしない区間を示す一組の二つの数字。DEGWK(1)<DEGWK(2)
CXX    write(6,*) ' input'
CXX    write(6,6645) DEGWK
CXX    write(6,6640) (DEGPLT(I),I=1,MXSG)
CXX 6640 FORMAT(2X, 6 ( '(', 2F7.1,') ' ) )
CXX 6645 FORMAT(2X, '(', 2F7.1,')' )

  IF(DEGWK(1).GE.360.) RETURN
  *   do nothing
  IF(DEGPLT(1).EQ.0. .AND. DEGPLT(2).GE.360.) THEN
  *   DEGPLT 初期設定 0. 361. 361. 361. 361....
    IF(DEGWK(1).EQ.0.) THEN
      DEGPLT(1)=DEGWK(2)
    ELSE
      DEGPLT(2)=DEGWK(1)
      DEGPLT(3)=DEGWK(2)
    ENDIF
  ELSE
    II=0
    JJ=0
    DO 310 K = 1,MXSG
      IF(DEGWK(1).GT.DEGPLT(K)) II=K
      IF(DEGWK(2).GT.DEGPLT(K)) JJ=K
310    CONTINUE
CXX    write(6,*)'  ii,jj=', II,JJ
    IF(MOD(II,2).EQ.0) THEN
  *    IIは偶数 : DEGWK(1)は描かない区間内
    IF(MOD(JJ,2).EQ.0) THEN
  *    JJは偶数 : DEGWK(2)は描かない区間内
      IF(II.EQ.JJ) THEN
        CONTINUE
  *      do nothing
      ELSE
CXX    write(6,*) ' do 320'
        DO 320 K = 1,MXSG-JJ
          DEGPLT(II+K)=DEGPLT(JJ+K)
320    CONTINUE
        ENDIF

```

```

ELSE
*
JJは奇数 : DEGWK(2)は描く区間内
CXX      write(6,*) ' do 330'
          DEGPLT(II+1)=DEGWK(2)
          DO 330 K = 2,MXSG-JJ+1
            DEGPLT(II+K)=DEGPLT(JJ+K-1)
330      CONTINUE
        ENDIF
ELSE
*
IIは奇数 : DEGWK(1)は描く区間内
IF(MOD(JJ,2).EQ.0) THEN
*
JJは偶数 : DEGWK(2)は描かない区間内
CXX      write(6,*) ' do 340'
          DEGPLT(II+1)=DEGWK(1)
          DO 340 K = 2,MXSG-JJ+1
            DEGPLT(II+K)=DEGPLT(JJ+K-1)
340      CONTINUE
        ELSE
*
JJは奇数 : DEGWK(2)は描く区間内
IF(II.EQ.JJ) THEN
*
描く区間に描かない区間を割り込み分割
CXX      write(6,*) ' do 350'
          DO 350 K = II+3,MXSG
            KK=MXSG+II+3-K
            DEGPLT(KK)=DEGPLT(KK-2)
350      CONTINUE
          DEGPLT(II+1)=DEGWK(1)
          DEGPLT(II+2)=DEGWK(2)
        ELSE
CXX      write(6,*) ' do 360'
          DEGPLT(II+1)=DEGWK(1)
          DEGPLT(II+2)=DEGWK(2)
          DO 360 K = 3,MXSG-JJ+2
            DEGPLT(II+K)=DEGPLT(JJ+K-2)
360      CONTINUE
        ENDIF
      ENDIF
    ENDIF
  ENDIF

*
DEGWK(1)=DEGPLT(K) or DEGWK(2)=DEGPLT(K) 対策
DO 370 I = 1,MXSG,2
IF (DEGPLT(I).LE.360.) THEN
  IF(DEGPLT(I+1).EQ.DEGPLT(I)) THEN
*
  delete the interval between DEGPLT(I) and DEGPLT(I+1)
*
  if DEGPLT(I)=DEGPLT(I+1) , where I=2*N-1
ccxxx  write(6,*) i, ' DEGPLT(i+1).eq.DEGPLT(i)', DEGPLT(i)
        DO 380 J = I+2,MXSG
          DEGPLT(J-2)=DEGPLT(J)
380      CONTINUE
        ENDIF

```

```
ENDIF
370 CONTINUE
```

```
CXX      write(6,*) ' output'
CXX      write(6,6640) (DEGPLT(I),I=1,MXSG)
```

```
RETURN
END
```

```
* --V---X---V---X---V---X---V---X---V---X---V---X---V---X---
```

```
SUBROUTINE XY2DEG ( DEG, X,Y, XC,YC,RC )
```

```
* 半径RC, 中心位置(XC,YC)の円の円周上の点(X,Y)の角度(DEG)を計算
```

```
D=SQRT((X-XC)**2+(Y-YC)**2)
```

```
DR=(D-RC)/RC
```

```
SMALL=1.E-4
```

```
IF(DR.LE.-SMALL .OR. DR.GE.SMALL)
```

```
&      WRITE(6,6910) X,Y, XC,YC,RC
```

```
6910  FORMAT(// ' THE POINT(' ,2F9.3,') DOES NOT SEEM TO BE ',
```

```
&      'ON THE CIRCLE', 3F9.3//)
```

```
CALL LINEDEG(DEG, X,Y,XC,YC)
```

```
RETURN
```

```
END
```

```
* -----
```

```
SUBROUTINE LINEDEG(DEG, X,Y,XC,YC)
```

```
* 線分(XC,YC)-(X,Y)がX軸となす角度(DEG)を計算
```

```
*  PI=ACOS(-1.)
```

```
R2D=180./ACOS(-1.)
```

```
DST=SQRT((X-XC)**2+(Y-YC)**2)
```

```
COSC=(X-XC)/DST
```

```
SINC=(Y-YC)/DST
```

```
IF(SINC.GE.0.) THEN
```

```
  DEG=ACOS(COSC)*R2D
```

```
ELSE
```

```
  DEG=360.-ACOS(COSC)*R2D
```

```
ENDIF
```

```
CXX      write(6,*) DEG, X,Y, XC,YC,DST, SINC,COSC
```

```
RETURN
```

```
END
```

```
* -----
```

```
SUBROUTINE SETDEGPLT (DEGPLT, MXSG,MXC)
```

```
* それぞれの円でプロットする区間の角度、初期値設定=全部描く
```

```
*      0. 361. 361. 361. 361.....
```

```
REAL DEGPLT(MXSG,MXC)
```

```
DO 20 N = 1,MXC
```

```
DEGPLT(1,N) = 0.
```

```
DO 20 M = 2,MXSG
```

```

DEGPLT(M,N) = 361.
20 CONTINUE
RETURN
END

```

\* -----

```

SUBROUTINE SETDEGPLTWK (DEGPLTWK, MAX)
REAL DEGPLTWK(MAX)
DO 30 M = 1,MAX
DEGPLTWK(M) = 361.
30 CONTINUE
RETURN
END

```

\* =====

```

SUBROUTINE SETLINE ( TRPZ, DLNX,DLNY, LONOFF, NTAN, NPLN,
&                    XA1, YA1, XB1, YB1, XA2, YA2, XB2, YB2,
&                    VALNO, NFC1, MXLN )

```

```

* 台形の頂点
REAL TRPZ(4,2,NFC1)
* 1つめの添え字で頂点を区別、2つ目の添え字で x,y を区別
* TRPZ(1),TRPZ(2),TRPZ(3),TRPZ(4)  A1,B1,B2,A2

* プロットする線分 ( 接点と接点を結ぶ線分の全部or一部 )
* 線分Mの両端の(X,Y):
* (DLNX(1,M),DLNY(1,M)),(DLNX(2,M),DLNY(2,M))
* (DLNX(1,M)<DLNX(2,M))
REAL DLNX(2,MXLN), DLNY(2,MXLN)
* 線分を引くかどうかのフラグ(=0/1 で引く/引かない)
* 線分がどの円の接線か
INTEGER LONOFF(MXLN), NTAN(2,MXLN)

* 共通外接線(XA1, YA1)-(XB1, YB1), (XA2, YA2)-(XB2, YB2),
REAL XA1(NFC1), YA1(NFC1), XB1(NFC1), YB1(NFC1),
& XA2(NFC1), YA2(NFC1), XB2(NFC1), YB2(NFC1)

DO 410 N = 1,NFC1
TRPZ(1,1,N)=XA1(N)
TRPZ(2,1,N)=XB1(N)
TRPZ(3,1,N)=XB2(N)
TRPZ(4,1,N)=XA2(N)
TRPZ(1,2,N)=YA1(N)
TRPZ(2,2,N)=YB1(N)
TRPZ(3,2,N)=YB2(N)
TRPZ(4,2,N)=YA2(N)
410 CONTINUE

DO 420 N = 1,NFC1

```

```
NN=N*2
NTAN(1,NN-1)=N
NTAN(2,NN-1)=N+1
NTAN(1,NN)=N
NTAN(2,NN)=N+1
```

```
IF(XA1(N).LT.XB1(N))THEN
  JUN=+1
ELSE IF (XA1(N).EQ.XB1(N))THEN
  IF(YA1(N).LT.YB1(N))THEN
    JUN=+1
  ELSE
    JUN=-1
  ENDIF
ELSE
  JUN=-1
ENDIF
IF(JUN.GT.0) THEN
  DLNX(1,NN-1)=XA1(N)
  DLNX(2,NN-1)=XB1(N)
  DLNY(1,NN-1)=YA1(N)
  DLNY(2,NN-1)=YB1(N)
ELSE
  DLNX(1,NN-1)=XB1(N)
  DLNX(2,NN-1)=XA1(N)
  DLNY(1,NN-1)=YB1(N)
  DLNY(2,NN-1)=YA1(N)
ENDIF
```

```
IF(XA2(N).LT.XB2(N))THEN
  JUN=+1
ELSE IF (XA2(N).EQ.XB2(N))THEN
  IF(YA2(N).LT.YB2(N))THEN
    JUN=+1
  ELSE
    JUN=-1
  ENDIF
ELSE
  JUN=-1
ENDIF
IF(JUN.GT.0) THEN
  DLNX(1,NN)=XA2(N)
  DLNX(2,NN)=XB2(N)
  DLNY(1,NN)=YA2(N)
  DLNY(2,NN)=YB2(N)
ELSE
  DLNX(1,NN)=XB2(N)
  DLNX(2,NN)=XA2(N)
  DLNY(1,NN)=YB2(N)
  DLNY(2,NN)=YA2(N)
ENDIF
```



420 CONTINUE

```
DO 430 N = 1,NFC1
  NN=N*2
  IF(XA1(N).EQ.VALNO) THEN
    LONOFF(NN-1)=0
    LONOFF(NN)=0
  ELSE
    LONOFF(NN-1)=1
    LONOFF(NN)=1
  ENDIF
```

430 CONTINUE

```
NPLN=NFC1*2
DO 435 N = NPLN+1,MXLN
  LONOFF(N)=0
  NTAN(1,N)=0
  NTAN(2,N)=0
```

435 CONTINUE

```
RETURN
END
```

\* --V----X----V----X----V----X----V----X----V----X----V----X----V----X--

```
  SUBROUTINE LINECIRCL ( DLNX,DLNY, LONOFF, NTAN, NPLN,
&                        XC,YC,RC,   NFC, MXLN )
```

\* < 円の中の線分 >

```
  REAL DLNX(2,MXLN),DLNY(2,MXLN)
  INTEGER LONOFF(MXLN), NTAN(2,MXLN)
```

```
  REAL XC(NFC), YC(NFC), RC(NFC)
```

```
  REAL SOLX(2),SOLY(2),SOLD(2)
```

```
DO 450 N = 1,NFC
```

\* WRITE(6,6720) N,XC(N),YC(N),RC(N)

\* 6720 FORMAT(3X,'circle ', I2, ' x,y,r:',3F7.1)

```
  INCR=0
```

```
DO 455 L = 1,NPLN
```

```
IF(LONOFF(L).GE.1) THEN
```

```
  IF(N.EQ.NTAN(1,L) .OR. N.EQ.NTAN(2,L) ) THEN
```

\* 線分 L は円 N の接線なので、円の内部には入らない

\* write(6,\*) ' line', L, ' is tangent of circle',N

cx WRITE(6,6710) N, L, (DLNX(I,L),DLNY(I,L),I=1,2)

cx 6710 FORMAT(3X,'circle', I3, ' /line',I3,

cx & ' (x,y),(x,y):', 2('(',2F7.2,')'), ' skip' )

```
  GOTO 455
```

```
ENDIF
```

\* 線分 L の端が円 N の中かどうか判定

```

XP=DLNX(1,L)-XC(N)
XQ=DLNX(2,L)-XC(N)
YP=DLNY(1,L)-YC(N)
YQ=DLNY(2,L)-YC(N)
DP=SQRT(XP**2+YP**2)
DQ=SQRT(XQ**2+YQ**2)
IF(DP.LE.RC(N) .AND. DQ.LE.RC(N) ) THEN
*   線分 L は描かない
    LONOFF(L)=0
c   write(6,6730) L,N, 'the line is inside of the circle:noplot'
c 6730   format(7X,'line ',I2,'/circle ',I2,': ', A40)
        WRITE(6,6740) L, (DLNX(I,L),DLNY(I,L),I=1,2),
            &               N, XC(N),YC(N),RC(N),
            &               'the line is inside of the circle: noplot'
6740   FORMAT(/6X,'line',I3,' (x,y):', 2('(','2F7.1,')') /
            &       6X,'circle', I2,' x,y,r:',3F7.1, ' [' , A40,']',/
            &       8X, 2('(','2F7.1,')') )
*   IF(DP.LE.RC(N) .AND. DQ.LE.RC(N) ) THEN --> ELSE
ELSE
*   DPかDQのどちらかあるいは両方がRCより大きい
*   円Nと線分Lの交点を計算
*   -----
        CALL LINCIRCRS (SOLX,SOLY,SOLD,NSL,
            &               DLNX(1,L),DLNY(1,L), XC(N),YC(N),RC(N) )

        IF(NSL.EQ.1) THEN
            XX=SOLX(1)
            YY=SOLY(1)
            IF(DP.LE.RC(N)) THEN
c   write(6,*) '      point P is inside'
            WRITE(6,6740) L, (DLNX(I,L),DLNY(I,L),I=1,2),
                &               N, XC(N),YC(N),RC(N),
                &               'the line and the circle cross at      ', XX,YY
            WRITE(6,6780) DLNX(1,L), DLNY(1,L)
            CALL DELINE ( DLNX(1,L), DLNY(1,L), XX,YY, -1)
            ELSEIF(DQ.LE.RC(N)) THEN
*   write(6,*) '      point Q is inside'
            WRITE(6,6740) L, (DLNX(I,L),DLNY(I,L),I=1,2),
                &               N, XC(N),YC(N),RC(N),
                &               'the line and the circle cross at      ', XX,YY
            WRITE(6,6780) DLNX(2,L), DLNY(2,L)
            CALL DELINE ( DLNX(1,L), DLNY(1,L), XX,YY, +1)
6780   FORMAT(8X,'(',2F7.1,') is inside')
            ENDIF
            ELSEIF(NSL.EQ.2) THEN
*   write(6,*) '      both point P & Q are outside'
*   write(6,*) '      - two answers on the limited line'
            XX1=SOLX(1)
            YY1=SOLY(1)
            XX2=SOLX(2)
            YY2=SOLY(2)

```

```

        WRITE(6,6740) L, (DLNX(I,L),DLNY(I,L),I=1,2),
&          N, XC(N),YC(N),RC(N),
&          'the line and the circle cross at      ',
&          XX1,YY1,XX2,YY2
        CALL DVLIN ( DLNX,DLNY, LONOFF, NTAN, INCR,
&          XX1,YY1,XX2,YY2, L, NPLN,MXLN )
    ELSE
*      NSL=0 : 円Nと線分Lは交わらない
*          do nothing
*          continue
*          write(6,*) '          - no answer between line and circle'
    ENDIF
*      IF(DP.LE.RC(N) .AND. DQ.LE.RC(N) ) THEN --> ELSE --> ENDIF
    ENDIF
*      IF(LONOFF(L).GE.1) THEN --> ENDIF
    ENDIF
455 CONTINUE
    NPLN=NPLN+INCR
*      線分を分割した場合には線分の数を一つ増やし、次の円に移る。
450 CONTINUE
    RETURN
    END

```

\* -----

```

    SUBROUTINE LINCIRCRS (XX,YY,DEG,NSL,
&          DLX,DLY, XC,YC,RC )
*      円と線分の交点を計算

    REAL DLX(2),DLY(2)
    REAL XX(2),YY(2),DEG(2)

    XP=DLX(1)-XC
    XQ=DLX(2)-XC
    YP=DLY(1)-YC
    YQ=DLY(2)-YC
    DLXPQ=DLX(1)-DLX(2)
    DLYPQ=DLY(1)-DLY(2)
    NSL=0
    BSQAC=RC**2*(DLXPQ**2+DLYPQ**2) - (XP*YQ-XQ*YP)**2
    IF(BSQAC.GT.0) THEN
*      円Nと直線Lは交わる。線分が交わるか判定
        AA=DLXPQ**2+DLYPQ**2
        BB=XQ*DLXPQ+YQ*DLYPQ
        ANS1=(-BB+SQRT(BSQAC))/AA
        ANS2=(-BB-SQRT(BSQAC))/AA
***      XX=ANS*XP+(1.-ANS)*XQ
***      YY=ANS*YP+(1.-ANS)*YQ
***      RR=SQRT(XX**2+YY**2)
***      WRITE(6,*) RR,RC,RR-RC
***      XX=XX+XC

```

```

***      YY=YY+YC
        IF (ANS1.GE.0. .AND. ANS1.LE.1. ) THEN
            NSL=NSL+1
            XX(NSL)=XC+ANS1*XP+(1.-ANS1)*XQ
            YY(NSL)=YC+ANS1*YP+(1.-ANS1)*YQ
            CALL XY2DEG ( DEG(NSL), XX(NSL),YY(NSL), XC,YC,RC )
        ENDIF
        IF (ANS2.GE.0. .AND. ANS2.LE.1. ) THEN
            NSL=NSL+1
            XX(NSL)=XC+ANS2*XP+(1.-ANS2)*XQ
            YY(NSL)=YC+ANS2*YP+(1.-ANS2)*YQ
            CALL XY2DEG ( DEG(NSL), XX(NSL),YY(NSL), XC,YC,RC )
        ENDIF
*      IF(BSQAC.GT.0) THEN --> ELSE --> ENDIF
ELSE
*      BSQAC<0 : 円Nと直線Lは交わらない
*      do nothing
*      continue
*      write(6,*) '          - no answer between line and circle'
ENDIF

RETURN
END

* --V----X----V----X----V----X----V----X----V----X----V----X----V----X--

SUBROUTINE DELINE ( DLNX, DLNY, XX,YY, LLL)
REAL DLNX(2),DLNY(2)
IF (LLL.LT.0) THEN
*      (x,y)=(DLNX(1),DLNY(1)) と (XX,YY) の間を消す
        DLNX(1)=XX
        DLNY(1)=YY
ELSE IF (LLL.GT.0) THEN
*      (x,y)=(DLNX(2),DLNY(2)) と (XX,YY) の間を消す
        DLNX(2)=XX
        DLNY(2)=YY
ENDIF
RETURN
END

* -----

SUBROUTINE DVLINE ( DLNX,DLNY, LONOFF, NTAN, INCR,
&                  XX1,YY1,XX2,YY2, L, NPLN,MXLN )
*      線分を二つに分割する。新たな線分は NPLN+INCR番目
REAL DLNX(2,MXLN),DLNY(2,MXLN)
INTEGER LONOFF(MXLN), NTAN(2,MXLN)
INCR=INCR+1
L1=L
L2=NPLN+INCR
IF(XX1.GT.XX2) THEN

```

```

XX=XX2
XX2=XX1
XX1=XX
YY=YY2
YY2=YY1
YY1=YY
ELSE IF (XX1.EQ.XX2) THEN
  IF (YY1.GT.YY2) THEN
    XX=XX2
    XX2=XX1
    XX1=XX
    YY=YY2
    YY2=YY1
    YY1=YY
  ENDIF
ENDIF

```

```

* DLNX(1,L1)<XX1<XX2<DLNX(2,L1)
* LINE [ DLNX(1,L1)---DLNX(2,L1) ]
* --> (delete between XX1 and XX2) -->
* LINE [ DLNX(1,L1) -- DLNX(2,L1)=XX1 ] and
* [ XX2=DLNX(1,L1)---DLNX(2,L2) ]

```

```

DLNX(1,L2)=DLNX(1,L1)
DLNX(2,L2)=DLNX(2,L1)
DLNY(1,L2)=DLNY(1,L1)
DLNY(2,L2)=DLNY(2,L1)
NTAN(1,L2)=NTAN(1,L1)
NTAN(2,L2)=NTAN(2,L1)
LONOFF(L2)=LONOFF(L1)
CALL DELINE ( DLNX(1,L1), DLNY(1,L1), XX1,YY1, +1)
CALL DELINE ( DLNX(1,L2), DLNY(1,L2), XX2,YY2, -1)

```

```

write(6, '(6x,a5,i3,a15,i3,a5,i3)' )
& 'line ', L1, ' is divided to ', L1, ' and ', L2

```

```

RETURN
END

```

```

* =====

```

```

SUBROUTINE LINETRPZ ( DLNX,DLNY, LONOFF, NTAN, NPLN,
& TRPZ, VALNO, MXLN,MXC1,NFC1 )

```

```

* < 台形の中の線分 >

```

```

* プロットする線分 ( 接点と接点を結ぶ線分の全部or一部 )

```

```

* 線分Mの両端の(X,Y):

```

```

* (DLNX(1,M),DLNY(1,M)),(DLNX(2,M),DLNY(2,M))

```

```

* (DLNX(1,M)<DLNX(2,M)

```

```

REAL DLNX(2,MXLN), DLNY(2,MXLN)

```

```

* 線分を引くかどうかのフラッグ(LONOFF=0/1で引く/引かない)
* 線分がどの円の接線か
  INTEGER LONOFF(MXLN), NTAN(2,MXLN)
* 台形の頂点
  REAL TRPZ(4,2,NFC1)
* 1つめの添え字で頂点を区別、2つ目の添え字で x,y を区別
*   TRPZ(1),TRPZ(2),TRPZ(3),TRPZ(4)   A1,B1,B2,A2
* 交点の座標
  REAL XXX(4), YYY(4)

  DO 510 M = 1,NFC1
  IF (TRPZ(1,1,M).NE.VALNO) THEN
  INCR=0
*   WRITE(6,6760) M, (TRPZ(1,1,M),TRPZ(1,2,M),I=1,4)
* 6760  FORMAT(3X,'trapez',I3,' (x,y)...(x,y):', 4('(',2F7.2,')'))
  DO 515 L = 1,NPLN
  IF(LONOFF(L).GE.1) THEN
  IF(NTAN(1,L).NE.M) THEN
*   線分 L が台形 M を構成する線分ならskip
cx   WRITE(6,6710) M, L, (DLNX(1,L),DLNY(1,L),I=1,2)
cx 6710  FORMAT(3X,'trapez', I3, ' /line',I3,
cx   &   ' (x,y),(x,y):', 2('(',2F7.2,')'))

  CALL LNTRPCRS ( XXX,YYY,MSL,
&   DLNX(1,L),DLNY(1,L), TRPZ(1,1,M) )

  IF(MSL.EQ.2) THEN
*   WRITE(6,*) ' a part of the line segment, which is located',
*   &   ' in the quadrangle, should be deleted.'
  WRITE(6,6770) M, L, (DLNX(1,L),DLNY(1,L),I=1,2),
&   (XXX(1),YYY(1),I=1,2)
*   線分の一部消去・分割
  CALL DVLINE ( DLNX,DLNY, LONOFF, NTAN, INCR,
&   XXX(1),YYY(1),XXX(2),YYY(2), L, NPLN,MXLN )
  ELSE
*   線分の端が台形の内部にあるかどうか。
  CALL INQUAD ( INSID1, DLNX(1,L),DLNY(1,L), TRPZ(1,1,M) )
  CALL INQUAD ( INSID2, DLNX(2,L),DLNY(2,L), TRPZ(1,1,M) )
  IF(MSL.EQ.1) THEN
*   WRITE(6,*) ' one end is in the quadrangle,',
*   &   ' another end is outside.',
*   &   ' line segment inside the quad. should be deleted.'
  WRITE(6,6770) M, L, (DLNX(1,L),DLNY(1,L),I=1,2),
&   XXX(1),YYY(1)
6770  FORMAT(3X,'trapez', I3, ' /line',I3, 2('(',2F7.1,')') /
&   8X, 'cross at',2('(',2F7.1,')') )
  IF(INSID1.EQ.1) THEN
    WRITE(6,6780) DLNX(1,L), DLNY(1,L)
    CALL DELINE ( DLNX(1,L), DLNY(1,L), XXX(1),YYY(1), -1)
  ELSEIF(INSID2.EQ.1) THEN
    WRITE(6,6780) DLNX(2,L), DLNY(2,L)

```

```

        CALL DELINE ( DLNX(1,L), DLNY(1,L), XXX(1),YYY(1), +1)
    ENDIF
6780  FORMAT(8X, '(' ,2F7.1,') is inside')
    ELSEIF(MSL.EQ.0) THEN
c      WRITE(6,*) ' no crossing points, ',
c      &          ' but all of the line segment might be ',
c      &          ' inside of the quadrangle.'
        IF(INSID1.EQ.1 .AND. INSID2.EQ.1) THEN
            WRITE(6,6770) M, L, (DLNX(I,L),DLNY(I,L),I=1,2)
            WRITE(6,6780) DLNX(1,L), DLNY(1,L)
            WRITE(6,6780) DLNX(2,L), DLNY(2,L)
            LONOFF(L)=0
        ENDIF
    ENDIF
ENDIF
ENDIF
*   IF(NTAN(1,L).NE.M) THEN --> ENDIF
*   IF(LONOFF(L).GE.1) THEN --> ENDIF
ENDIF
ENDIF
515  CONTINUE
*   線分を分割した場合には線分の数を一つ増やし、次の台形に移る。
    NPLN=NPLN+INCR
*   IF (TRPZ(1,1,M).NE.VALNO) THEN ----> ENDIF
ENDIF
510  CONTINUE

    RETURN
    END

```

\* --V----X----V----X----V----X----V----X----V----X----V----X----V----X---

```

    SUBROUTINE LNTRPCRS ( XXX,YYY,MSL,
&                      DLX,DLY, QUAD )
*   線分と四角形の交点を求める
    REAL XXX(4), YYY(4)
    REAL DLX(2), DLY(2), QUAD(4,2)
    MSL=0

    DO 520 N = 1,4
        IF(N.EQ.4) THEN
            N2=1
        ELSE
            N2=N+1
        ENDIF
        CALL LINECRS ( XX,YY,NSL,
& DLX(1),DLY(1), DLX(2),DLY(2),
& QUAD(N,1),QUAD(N,2), QUAD(N2,1),QUAD(N2,2) )
        IF(NSL.EQ.1) THEN
            MSL=MSL+1
            XXX(MSL)=XX
            YYY(MSL)=YY
        ENDIF
    END DO

```

```

**      WRITE(6,'(3X,A8,2F7.1)') 'solution', XXX(MSL),YYY(MSL), MSL
      IF(MSL.GE.2) THEN
*      頂点で交わる場合、同じ答えが2回出てくる。
          DO 525 M = 1,MSL-1
              IF(XXX(MSL).EQ.XXX(M) .AND. YYY(MSL).EQ.YYY(M) ) THEN
                  MSL=MSL-1
                  write(6,*) ' delete '
              ENDIF
525      CONTINUE
          ENDIF
      ENDIF
520 CONTINUE
      RETURN
      END

*  --V----X----V----X----V----X----V----X----V----X----V----X----V----X----
      SUBROUTINE LINECRS ( XX,YY,NSL,
&      XO,YO,XA,YA, XP,YP,XQ,YQ )
*      線分と線分の交点を求める
C      line segment OA: (XO,YO)--(XA,YA)
C      line segment PQ: (XP,YP)--(XQ,YQ)
C      crossing point R: (XX,YY)          (point of intersection)
*      <OR> = a<OA> = (1-b)<OP>+b<OQ>
*      where 0<a<1 and 0<b<1
*      write(6,*) ' given line',XO,YO,'-',XA,YA,'xx', XP,YP,'-',XQ,YQ
      T1=(XP-XQ)*(YA-YO)
      T2=(YP-YQ)*(XA-XO)
      IF(T1.EQ.T2) THEN
*      WRITE(6,*) ' No solution: parallel line or same point'
          NSL=0
      ELSE
          TA=(YP-YQ)*XO+(YQ-YO)*XP+(YO-YP)*XQ
          TB=(YA-YQ)*XO+(YQ-YO)*XA+(YO-YA)*XQ
          AA=TA/(T1-T2)
          BB=TB/(T1-T2)
          XX=XO+(XA-XO)*AA
          YY=YO+(YA-YO)*AA
          IF(AA.GT.0. .AND. AA.LT.1. .AND.
&      BB.GT.0. .AND. BB.LT.1. )          THEN
              NSL=1
          ELSE
*      WRITE(6,*) ' The crossing point is not on the line segments'
              NSL=0
          ENDIF
      ENDIF
      RETURN
      END

```

\* -----



```

SUBROUTINE INQUAD ( INSIDE, XX,YY, QUAD )
* 点 (XX,YY) が四角形 quadrangle の内部にあるかどうかを返す
REAL QUAD(4,2)
REAL TRIANG1(3,2), TRIANG2(3,2)
INSIDE=0
DO 530 J = 1,2
DO 530 I = 1,3
  IF(I.LT.3) THEN
    II=I+2
  ELSE
    II=1
  ENDIF
  TRIANG1(I,J)=QUAD(I,J)
  TRIANG2(I,J)=QUAD(II,J)
530 CONTINUE
CALL INTRI ( INSIDE, XX,YY, TRIANG1 )
IF(INSIDE.EQ.1) RETURN
CALL INTRI ( INSIDE, XX,YY, TRIANG2 )
RETURN
END

```

\* -----

```

SUBROUTINE INTRI ( INSIDE, XX,YY, TRIANG )
* 点 (XX,YY) が三角形 triangle の内部にあれば INSIDE=1
REAL TRIANG(3,2)
* <OR>=a* <OP>+b* <OQ>
* a>0, b>0, a+b<1 なら、点Rは三角形OPQの内部
XR=XX-TRIANG(1,1)
YR=YY-TRIANG(1,2)
XP=TRIANG(2,1)-TRIANG(1,1)
YP=TRIANG(2,2)-TRIANG(1,2)
XQ=TRIANG(3,1)-TRIANG(1,1)
YQ=TRIANG(3,2)-TRIANG(1,2)
AA=(YR*XQ-XR*YQ)/(XQ*YP-XP*YQ)
BB=(XR*YP-YR*XP)/(XQ*YP-XP*YQ)
IF ( AA.GT.0. .AND. BB.GT.0. .AND. AA+BB.LT.1. ) THEN
  INSIDE=1
ELSE
  INSIDE=0
ENDIF
RETURN
END

```

\* --V---X---V---X---V---X---V---X---V---X---V---X---V---X---

```

SUBROUTINE DEG2ARC ( ARCX, ARCY, ARCD, ARCCRC, MARC, NARC,
& XC, YC, RC, DEGPLT, NFC, MXSG, MACX )
* <円弧を描くパラメタ取り出し>
*
* プロットする弧

```

```

* 弧Mの両端の(X,Y)
*      (ARCX(1,M), ARCY(1,M)), (ARCX(2,M), ARCY(2,M))
* 弧Mの両端の角度
*      ARCD(1,M), ARCD(2,M)
*      ARCD(1,M) < ARCD(2,M) で角度が増える方向に描く
* 弧Mのもとの円の中心の位置 (x,y) と半径 r
REAL ARCX(2,MACX), ARCY(2,MACX), ARCD(2,MACX),
& ARCCRC(3,MACX)
* 弧Mのもとの円の番号
INTEGER MARC(MACX)

REAL XC(NFC), YC(NFC), RC(NFC)
REAL DEGPLT(MXSG, NFC)

PI=ACOS(-1.)
D2R=ACOS(-1.)/180.

* to make NARC sets of the arcs
* with starting points and ending points
NARC=0
DO 800 N = 1, NFC
* H30.07.03 start
    IF (RC(N) .GT. 0.) THEN
* H30.07.03 end
**    write(6,6220) N, XC(N), YC(N), RC(N)
** 6220 FORMAT(3X, 'circle ', I2, ' x,y,r:', 3F7.1)
    DO 820 K = 1, MXSG, 2
        IF (DEGPLT(K,N) .GE. 360. ) GOTO 800
        RAD=DEGPLT(K,N)*D2R
        NARC=NARC+1
        XX=XC(N)+RC(N)*COS(RAD)
        YY=YC(N)+RC(N)*SIN(RAD)
**    write(6,6230) K, XX, YY, DEGPLT(K,N)
        ARCX(1,NARC)=XX
        ARCY(1,NARC)=YY
        ARCD(1,NARC)=DEGPLT(K,N)
        IF (DEGPLT(K+1,N) .GE. 360. ) THEN
            DEGPLT(K+1,N)=360.
            RAD=2.*PI
        ELSE
            RAD=DEGPLT(K+1,N)*D2R
        ENDIF
        XX=XC(N)+RC(N)*COS(RAD)
        YY=YC(N)+RC(N)*SIN(RAD)
**    write(6,6230) K+1, XX, YY, DEGPLT(K+1,N)
        ARCX(2,NARC)=XX
        ARCY(2,NARC)=YY
        ARCD(2,NARC)=DEGPLT(K+1,N)

        MARC(NARC)=N
        ARCCRC(1,NARC)=XC(N)

```

```

        ARCCRC(2,NARC)=YC(N)
        ARCCRC(3,NARC)=RC(N)
        IF(DEGPLT(K+1,N).GE.360.) GOTO 800
** 6230   FORMAT (3X,I3,2F7.1,' (' ,F7.1,')' )
820   CONTINUE
* H30.07.03 start
        ENDIF
* H30.07.03 end
800   CONTINUE
* -----
        RETURN
        END
* =====

```

```

        SUBROUTINE SNGLSTRK ( JUN, NCMP,
&           DLNX,DLNY, LONOFF, NPLN,
&           ARCX,ARCY, ARCD, ARCCRC, NARC, MXCMP )

*   プロットする線分・弧の終点/始点がつながる順番などを示す
        INTEGER JUN(3,MXCMP)
*   lines and arcs are re-ordered with one end attaching the next.
*   1: line(=1) or arc(=2)
*   2: number of line or arc
*   3: increasing (in x or degree)(=+1) or decreasing (= -1)

        REAL DLNX(2,NPLN), DLNY(2,NPLN)
        INTEGER LONOFF(NPLN)

        REAL ARCX(2,NARC), ARCY(2,NARC), ARCD(2,NARC),
&   ARCCRC(3,NARC)

        CHARACTER CLORC(2)*4
        DATA CLORC / 'line', 'arc ' /

        BIG=ARCCRC(3,NARC)*10.

        PI=ACOS(-1.)
        D2R=ACOS(-1.)/180.

```

```

* -----
*   The first point to plot:
*   An arc is always plotted,
*   though all the line segments might be deleted.
        LORC=2
        NUM=1
        INCR=+1
        NCMP=1
        JUN(1,NCMP)=LORC
        JUN(2,NCMP)=NUM
        JUN(3,NCMP)=INCR

```

```

LORCO=LORC
NUMO=NUM
* -----
880 CONTINUE
* -----
  IF(INCR.GT.0) THEN
    NEND=2
  ELSE
    NEND=1
  ENDIF

  IF(LORC.EQ.1) THEN
    XEND=DLNX(NEND,NUM)
    YEND=DLNY(NEND,NUM)
  ELSE IF(LORC.EQ.2) THEN
    XEND=ARCX(NEND,NUM)
    YEND=ARCY(NEND,NUM)
  ENDIF

* search the nearest point to (XEND,YEND)
DSTLSQ=BIG**2
DO 850 L = 1,NPLN
  IF(LONOFF(L).NE.0) THEN
  IF(LORC.EQ.1 .AND. L.EQ.NUM) THEN
    CONTINUE
C skip: the same line segment
  ELSE
    DSTSQ=(XEND-DLNX(1,L))**2+(YEND-DLNY(1,L))**2
    IF(DSTSQ.LE.DSTLSQ) THEN
      NUML=L
      INCL=+1
      DSTLSQ=DSTSQ
    ENDIF
    DSTSQ=(XEND-DLNX(2,L))**2+(YEND-DLNY(2,L))**2
    IF(DSTSQ.LE.DSTLSQ) THEN
      NUML=L
      INCL=-1
      DSTLSQ=DSTSQ
    ENDIF
  ENDIF
ENDIF
850 CONTINUE
DSTCSQ=BIG**2
DO 860 N = 1,NARC
  IF(LORC.EQ.2 .AND. N.EQ.NUM) THEN
    CONTINUE
C skip: the same arc
  ELSE
    DSTSQ=(XEND-ARCX(1,N))**2+(YEND-ARCY(1,N))**2
    IF(DSTSQ.LE.DSTCSQ) THEN
      NUMC=N
      INCC=+1

```

```

        DSTCSQ=DSTSQ
    ENDIF
    DSTSQ=(XEND-ARCX(2,N))**2+(YEND-ARCY(2,N))**2
    IF(DSTSQ.LE.DSTCSQ) THEN
        NUMC=N
        INCC=-1
        DSTCSQ=DSTSQ
    ENDIF
ENDIF
860 CONTINUE
    IF(DSTLSQ.LE.DSTCSQ) THEN
        LORC=1
        NUM=NUML
        INCR=INCL
        GAP=DSTLSQ
    ELSE
        LORC=2
        NUM=NUMC
        INCR=INCC
        GAP=DSTCSQ
    ENDIF
    GAP=SQRT(GAP)
    IF(GAP.GT.1.E-5*ARCCRC(3,1)) WRITE(6,6290)
&   CLORC(JUN(1,NCMP)),JUN(2,NCMP), CLORC(LORC),NUM, GAP
6290  FORMAT (/ 2X,'!!!!!! GAP BETWEEN <',A4,I3,'> AND <',A4,I3,
&   '> SEEMS TO BE TOO LARGE:',F9.3, '!!!!!!' /)
    IF(LORC.EQ.LORCO .AND. NUM.EQ.NUMO) THEN
*      write(6,*)' return to the first segment (arc)'
        CONTINUE
*      GOTO 890
    ELSE
        NCMP=NCMP+1
        JUN(1,NCMP)=LORC
        JUN(2,NCMP)=NUM
        JUN(3,NCMP)=INCR
        IF(NCMP.GE.MAX) THEN
            WRITE(6,*)' TOO MANY SEGMENTS OR A BUG OF THE PROGRAM'
            GOTO 890
        ENDIF
        GOTO 880
    ENDIF
ENDIF
* -----
890 CONTINUE
* -----

    RETURN
    END

* --V---X---V---X---V---X---V---X---V---X---V---X---V---X---

```

```

SUBROUTINE PRNTCMP ( JUN, NCMP,

```

```

&          DLNX,DLNY, NPLN,
&          ARCX,ARCY, ARCD, ARCCRC, MARC, NARC )

*   プロットする線分・弧の終点/始点がつながる順番などを示す
INTEGER JUN(3,NCMP)
*   lines and arcs are re-ordered with one end attaching the next.
*   1: line(=1) or arc(=2)
*   2: number of line or arc
*   3: increasing (in x or degree)(=+1) or decreasing (= -1)

REAL DLNX(2,NPLN), DLNY(2,NPLN)

REAL ARCX(2,NARC), ARCY(2,NARC), ARCD(2,NARC),
&   ARCCRC(3,NARC)
INTEGER MARC(NARC)
CHARACTER CLORC(2)*4
DATA CLORC / 'line', 'arc ' /

WRITE(6,6240)
6240 FORMAT(15X,2( '( x y ) ',': (deg start-end )',
&   14X, '(center x y radius)' )

DO 910 N = 1,NCMP
LORC=JUN(1,N)
NUM=JUN(2,N)
IF(JUN(3,N).GT.0) THEN
  I1=1
  I2=2
ELSE
  I1=2
  I2=1
ENDIF
IF(LORC.EQ.1) THEN
  X1=DLNX(I1,NUM)
  Y1=DLNY(I1,NUM)
  X2=DLNX(I2,NUM)
  Y2=DLNY(I2,NUM)
  WRITE(6,6250) N,CLORC(LORC),NUM, X1,Y1,X2,Y2
6250 FORMAT(2X,I2,2X, A4, I3, 2X,2( '(' ,2F7.1, ' ) ' ) )
ELSE
*   Since the first segment to plot is arc,
*   and increasing degree (anti-clockwise plot) is assumed,
*   the arc should be always plotted in anti-clockwise way.
IF(JUN(3,N).LT.0)
&   write(6,*) ' CAUTION: The arc is plotted clock-wise'
  X1=ARCX(I1,NUM)
  Y1=ARCY(I1,NUM)
  X2=ARCX(I2,NUM)
  Y2=ARCY(I2,NUM)
  D1=ARCD(I1,NUM)
  D2=ARCD(I2,NUM)

```

```

        WRITE(6,6260) N,CLORC(LORC),NUM, X1,Y1,X2,Y2,
&      D1,D2, MARC(NUM), (ARCCRC(I,NUM),I=1,3)
6260 FORMAT(2X,I2,2X,      A4,I3, 2X,2( '(' ,2F7.1,' ) ' ),
& ': (' ,2F7.1,' ) of circle', I3,' (' ,3F7.1,' ) ' )
        ENDIF
910 CONTINUE

        RETURN
        END

```

```

*****
*****

```

```

SUBROUTINE CIRCLDATA (XC,YC,RC,NFC, MXC)
*   円の中心位置(XC,YC), 半径(RC)
    REAL XC(MXC), YC(MXC), RC(MXC)
    CALL SAMPLE (XC,YC,RC,NFC, MXC)
    RETURN
    END

```

```

* - - - - -

```

```

SUBROUTINE SAMPLE (XC,YC,RC,NFC, MXC)
PARAMETER (MAX=3)
REAL XC(MXC), YC(MXC), RC(MXC)
REAL XX(MAX), YY(MAX), RR(MAX)
DATA XX / 7., 13., 0. /
DATA YY / 0., 0., 0. /
DATA RR / 2., 3., 4. /
NFC=3
DO 20 N = 1,NFC
    XC(N)=XX(N)
    YC(N)=YY(N)
    RC(N)=RR(N)
20 CONTINUE
RETURN
END

```