REVISTA DE OBRAS PUBLICAS, Agosto 1981. Pág. 557 a 566.

Determinación gráfica de la intersección de recintos planos, por medio de un computador (*)

Por A. RECUERO
Dr. Ing. Caminos, Canales y Puertos
Inst.° Eduardo Torroja. Madrid

Se presenta un algoritmo que permite resolver de forma general el problema de la determinación gráfica de la intersección de un número cualquiera de recintos planos. Los recintos a intersectar deben aproximarse mediante recintos poligonales. Dichos recintos pueden ser conexos, simple o múltiplemente, o no conexos.

El algoritmo determina la intersección de dos recintos, y está preparado para poder ser utilizado de forma repetitiva con comodidad. De este modo puede aplicarse para determinar la intersección de un número cualquiera de recintos.

Se incluyen listados de computador escritos en FORTRAN IV para un miniordenador HP-21MX, así como varios ejemplos de utilización.

1. INTRODUCCION.

La presentación gráfica de los resultados de un programa de ordenador tiene notables ventajas sobre la presentación numérica de los mismos en un gran número de aplicaciones, tanto de tipo técnico como de tipo comercial o financiero.

En todos aquellos casos en los que la solución del problema planteado se extiende a un intervalo, de tiempo o de espacio, más o menos extenso, la presentación numérica de la solución ha de resolverse por medio de tablas. La interpretación de una tabla, su forma de variación, o su comparación con otras tablas semejantes resulta una labor tediosa y propensa a ocasionar equivocaciones. En estos casos, la presentación gráfica de los resultados, no sólo reduce notablemente el volumen de salida, sino que además facilita considerablemente la interpretación general de la solución, y las comparaciones entre distintas soluciones.

Hay incluso algunas aplicaciones en las que la presentación gráfica de los resultados es la única viable de obtener una solución util. Uno de estos casos es el problema de la superposición de mapas. Puede describirse este problema de forma sucinta como sigue.

Considerese una determinada región de la cual se dispone de un conjunto de mapas, cada uno de los cuales está hecho con referencia a algún tipo de atributo. Por ejemplo: Sismicidad de la zona, tipo de suelo, velocidad media del viento, profundidad de la capa freática, propiedad pública o privada del suelo, clasificación urbanística del suelo, pendiente media del terreno, etc.

Para planificar adecuadamente la realización de obras públicas, grandes instalaciones, ordenaciones del territorio, prospecciones mineras, explotaciones agrícolas y muchos otros temas puede resultar imprescindible disponer de nuevos mapas de la región en los cuales se localicen las zonas en las que concurren varias de las características recogidas en el conjunto de los mapas básicos antes citados.

Este problema puede resolverse de forma simple y cómoda, con ayuda de un ordenador. Para ello bastará con disponer de una descripción de los mapas básicos, de un medio de salida gráfica y del algoritmo adecuado para determinar la intersección de los mapas básicos.

Actualmente, las disponibilidades de terminales de salida gráfica para ordenadores, se han incrementado notablemente. Sus precios se han reducido, siguiendo la tónica general de los restantes componentes de las instalaciones de ordenadores. Su manejo se ve también notablemente facilitado por disponer de un amplio soporte de subrutinas tanto para el dibujo como para la rotulación de las figuras.

(*) Se admiten comentarios sobre el presente artículo, que podrán remitirse a la Redacción de esta Revista hasta el 31 de Octubre de 1981.

Los terminales de computador de salida gráfica más comunes son:

- Plotters, con una amplia gama de tamaños, velocidades, resolución, y de posibilidades de dibujo con varios colores.
- Impresoras gráficas de línea o de página.
- Pantallas de rayos catódicos, en blanco y negro o en color, dotadas con posibilidad de obtener copia en papel del contenido de la pantalla.

Estos tipos de salida gráfica pueden utilizarse incluso con equipos de tamaño y precio extremadamente reducido, pues existen incluso calculadoras programables de bolsillo que pueden producir salida gráfica.

Todos estos factores hacen aconsejable que los centros de cálculo electrónico dispongan de los programas adecuados para poder utilizar estas posibilidades de su equipo.

Se considera el problema de determinar un recinto en el cual se dan simultaneamente una serie de propiedades o circunstancias a partir de un conjunto de recintos dentro de cada uno de los cuales se verifica por separado una de las citadas propiedades.

Sí bién el problema no presenta conceptualmente ninguna dificultad, y su resolución manual es trivial, al tratar de resolver el problema con un computador de modo que éste dibuje automáticamente el recinto solución, la resolución del problema con caracter totalmente general encierra algunas dificultades, que el algoritmo que se presenta en este trabajo resuelve de forma completa y eficiente.

Considérese tres recintos A, B y C en cada uno de los cuales se cumple respectivamente las propiedades a,b y c. El recinto en el cual se cumplen simltáneamente las propiedades a, b y c será el recinto.

$$A \cap B \cap C$$

Puesto que la operación de intersección goza de la propiedad asociativa tendremos que

$$A \cap B \cap C = (A \cap B) \cap C \qquad (1)$$

En consecuencia, un algoritmo que permita resolver la intersección de dos recintos cualesquiera, permitirá, mediante la aplicación reiterada del mismo, obtener la intersección de un número cualquiera de recintos.

Desde un punto de vista computacional, la forma más simple de representar dominios planos es por medio de un recinto de forma poligonal que aproxime el recinto real.

En el presente trabajo se describe de modo completo un algoritmo que permite resolver de forma general la intersección de dos recintos poligonales. De acuerdo con lo dicho inmediatamente antes, la aplicación reiterada de este algoritmo, y la representación de un recinto cualquiera por medio de un recinto poligonal aproximado hace que el presente algoritmo permita resolver de forma completa el problema de la intersección de un número cualquiera de recintos planos.

Se incluye también la programación, en FORTRAN IV, del algoritmo, organizado en forma de subrutinas, junto con los resultados de aplicación del algoritmo a distintos ejemplos.

2. DESCRIPCION DEL ALGORITMO

Se describe aquí, de forma completa, el algoritmo propuesto para hallar la intersección de dos recintos poligonales. Para ello se pasa revista a:

- Forma de describir los recintos.
- Determinación de la posición de un punto respecto a una recta.
- Determinación de la intersección de dos segmentos.
- Determinación de sí un punto es interior a un recinto.
- Descripción completa del algoritmo, haciendo uso de lo expuesto en los puntos anteriores.

2.1. Descripción de un recinto

Un recinto se describe por la cadena poligonal cerrada que constituye su contorno.

Dicha cadena se supone orientada según el siguiente criterio: Las aristas de la cadena se recorren de modo tal que, al avanzar por ellas, el interior del recinto quede situado a la derecha. En un recinto convexo este criterio coincide con un sentido horario del recorrido del contorno.

Cada arista de la cadena se caracteriza por un vértice origen. Se supone que dicho vértice origen pertenece a la arista, mientras que el vértice final se supone que no pertenece a la misma.

Un recinto dato debe definirse mediante una única cadena. En consecuencia, si un recinto no es simplemente convexo, para poderlo definir, deberán darse los cortes adecuados de modo que pueda definirse el contorno mediante una única cadena.

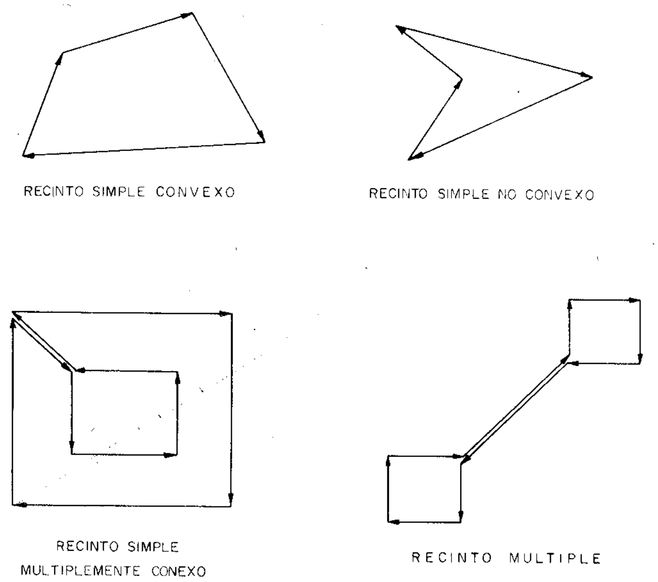


Fig. 1. Algunos tipos de recintos.

Análogamente, en el caso de recintos múltiples, deberán establecerse las conexiones adecuadas para definirlas por medio de una sola cadena.

En la figura 1 se muestra el modo de describir distintos recintos convexos o no, simple o múltiplemente conexos, o no conexos.

2.2. Posición de un punto respecto a una receta.

El algoritmo que aquí se describe se basa fundamentalmente en las propiedades del producto vectorial de vectores.

Consideremos en el espacio un triedro dextrógiro OXYZ. Supondremos que los recintos cuya intersección se desea hallar están sobre el plano XY. Si consideramos en dicho plano XYun vector $\overline{AB}$ y un punto C cualquiera del plano, la posición de dicho

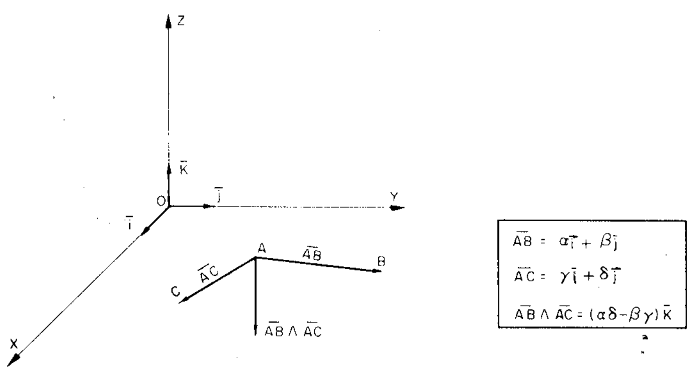


Fig. 2. Sistema de referencia y producto vectorial.

punto C respecto a la recta definida por $\overline{AB}$ recorrida en el sentido de A hacia B, se determina del siguiente modo:

Sea $\overline{AB} \wedge \overline{AC} = p\bar{k}$ (1)

donde $\bar{k}$ es el versor del eje OZ

Si $P < 0$, C se encontrará a la derecha de la recta.
Si $p > 0$, el punto C se encontrará a la izquierda.
Si $p = 0$, el punto C se encontrará sobre la recta.
(Ver figura 2).

INTERSECCION DE DOS SEGMENTOS

Considerese los segmentos orientados $\overline{AB}$ y $\overline{CD}$. (Ver figura 3).

Supongamos que dichos segmentos son sendas aristas pertenecientes a las cadenas de contorno de dos recintos. De acuerdo con lo dicho respecto a la definición de los recintos supondremos que el punto B no pertenece al segmento $\overline{AB}$ y que el punto D no pertenece al segmento $\overline{CD}$.

Para determinar si existe intersección de ambos segmentos se considerarán los siguientes productos vectoriales.

$$\begin{aligned} \overline{AB} \wedge \overline{AC} &= p_1\bar{k} \\ \overline{AB} \wedge \overline{AD} &= p_2\bar{k} \\ \overline{CD} \wedge \overline{CA} &= p_3\bar{k} \\ \overline{CD} \wedge \overline{CB} &= p_4\bar{k} \end{aligned} \qquad (2)$$

Considerese ahora la expresión lógica (3).

$(p_1p_2 \leqslant 0)$ AND $(p_3p_4 \leqslant 0)$ AND $(p_2 \neq 0)$ AND $(p_4 \neq 0)$ (3)

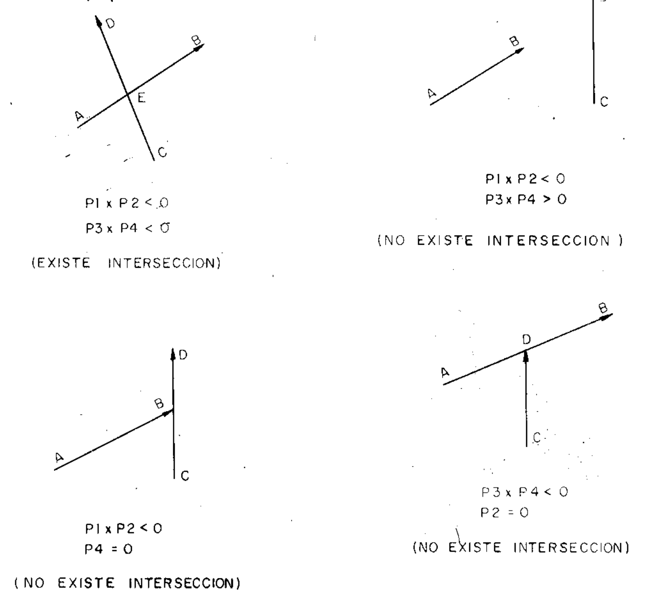


Fig. 3. Posiciones relativas de dos segmentos.

Si el valor que toma la expresión (3) es verdadero, existirá intersección de los segmentos $\overline{AB}$ y $\overline{CD}$. En caso contrario no existirá intersección.

Si existe intersección de ambos segmentos, llamando E a dicho punto, podemos determinar su posición del siguiente modo.

Un vector $\overline{AN}$, siendo N un punto genérico de la recta CD, vendrá dado por la expresión (4).

$$\overline{AN} = \lambda \,.\, \overline{AC} + (1-\lambda)\,.\,\overline{AD} \qquad (4)$$

donde λ es un parámetro variable.

Si se impone la condición de que $\overline{AN}$ sea paralelo a $\overline{AB}$, tendremos que,

$$\overline{AN} \wedge \overline{AB} = 0 \qquad (5)$$

en tal caso el punto genérico N coincidirá con el punto de intersección E lo cual equivale a que

$$\lambda \,.\, p_1 + (1-\lambda)\,.\, P_2 = 0 \qquad (6)$$

de donde,

$$\overline{AE} = \frac{-p2}{p1-p2} \cdot \overline{AC} + \frac{p1}{p1-p2} \cdot \overline{AD} \qquad (7)$$

2.3. Posición de un punto respecto a un recinto

Se considera que un punto pertenece a un recinto si es interior a él o pertenece a su contorno.

Para determinar si un punto pertenece a un recinto dado, se propone el siguiente procedimiento.

Considerese la cadena que define un recinto, formada por los putos ABCD..., consecutivos según el sentido de recorrido adoptado. El proceso que se describe permite la eliminación sucesiva de vértices de la cadena, formando en consecuencia una cadena auxiliar derivada, tal que cuando el número de vértices de la cadena auxiliar sea 2 el proceso se detenga, a menos que en un paso anterior se haya comprobado que el punto en cuestión es interior al recinto.

Sea N el punto cuya posición respecto al recinto limitado por la cadena ABCD se desea determinar.

1. Considerese el triángulo de vértices ABC. Determinar si los puntos del interior de dicho triángulo son interiores al recinto inicial. Para ello habrá que considerar

$$\overline{AB} \wedge \overline{AC} = p\overline{k} \qquad (8)$$

1.1 Si p < O los puntos del interior del triángulo ABC pueden ser interiores al recinto. A continuación se comprobará si el segmento $\overline{AC}$ corta a alguna arista de la cadena auxiliar derivada. En caso de no ser así se continuará en el punto 2. En caso de haber alguna intersección, alguna parte del triángulo ABC puede no ser interior al recinto. En este caso se tomará como nuevo punto A el punto B, como nuevo punto B el punto C, como nuevo punto C el punto D, y se volverá al punto 1.

1.2 Si p > O los puntos del triángulo ABC no son interiores al recinto dado. En consecuencia, se tomará como nuevo punto A el punto B, como nuevo punto B el punto C y como nuevo punto C el punto D y se volverá al punto 1.

1.3 Si p = O los puntos AB y C están alineados. En tal caso se eliminará el punto B, y se tomará como nuevo punto B el punto C y como nuevo punto C el punto D y se volverá al punto 1.

2. Comprobar si el punto N es interior al triángulo ABC. Para ello habrá que considerar los siguientes productos vectoriales.

$$\begin{aligned} \overline{AB} \wedge \overline{AN} &= p_1\overline{k} \\ \overline{BC} \wedge \overline{BN} &= p_2\overline{k} \\ \overline{CA} \wedge \overline{CN} &= p_3\overline{k} \end{aligned} \qquad (10)$$

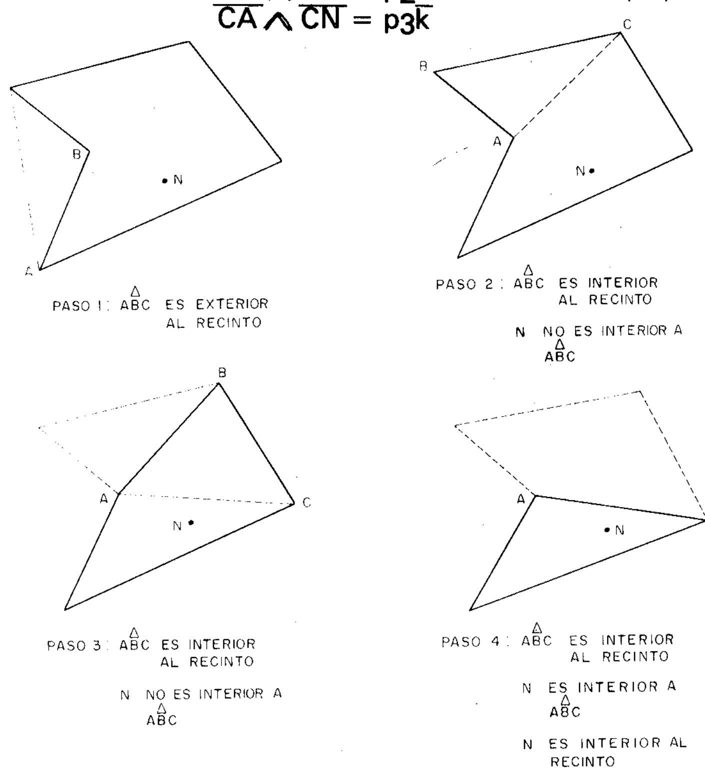


Fig. 4. Esquema del proceso de comprobación de la posición de un punto respecto a un recinto.

Considerese ahora la expresión lógica (11)

$$(p_1 \leqslant 0) \text{ AND } (p_2 \leqslant 0) \text{ AND } (p_3 \leqslant 0) \qquad (11)$$

Si esta expresión es cierta el punto N será interior al triángulo ABC y en consecuencia al recinto total. Por consiguiente el proceso termina aquí con resultado afirmativo.

Si la expresión (11) es falsa el punto N no pertenece al triángulo ABC. En tal caso se eliminará el punto B, se tomará como nuevo punto B el punto C y como nuevo punto C el punto D y se pasará de nuevo al punto 1.

Tal como se ha dicho, cuando la cadena auxiliar, tras la sucesiva eliminación de puntos B, se reduzca a dos puntos sin haber obtenido solución afirmativa el proceso se detendrá dando una solución negativa. El proceso se esquematiza en la figura 4.

2.4. Algoritmo para la determinación de la solución

La solución del problema planteado puede ser vacía, puede consistir en un solo recinto, o puede consistir en varios recintos disjuntos. El algoritmo que se describe contempla todas estas posibilidades. La solución puede ser vacía, en cuyo caso no dará ninguna cadena solución.

En caso de que la solución conste de varios recintos disjuntos el algoritmo proporcionará tantas cadenas solución como recintos en general, salvo que incluya algún corte que pertenezca a alguna de las cadenas que definen los recintos originales.

En el caso de que alguno de los recintos solución se reduzcan a un punto o a una cadena poligonal no cerrada, cada una de estas soluciones se da como recinto solución independiente, descrito por la cadena adecuada.

A continuación se pasa a describir el algoritmo. Se llamará primera cadena a la que define el primero de los dos recintos y segunda cadena a la que define al segundo.

1. Determinar todas las posibles intersecciones de aristas de la primera cadena con aristas de la segunda cadena.

   Cada uno de los puntos de intersección se identificará de modo que sea posible distinguir los vértices de las cadenas de los puntos de intersección.

   Si el número de intersecciones fuese 0 se seguirá en el punto 2, y si fuese distinto de 0 se seguirá en el punto 3.

2. Cuando no existe ninguna intersección entre aristas de ambas cadenas, puede producirse una de las tres siguientes situaciones:
   a) El primer recinto contiene al segundo.
   b) El segundo recinto contiene al primero.
   c) La intersección de ambos recintos es vacía.

   Para determinar cual de estas 3 situaciones es la que ocurre se hará:

   2.1. Comprobar si un punto de la segunda cadena es interior al recinto 1. Si la respuesta es afirmativa la solución es el recinto segundo y el proceso queda determinado. En caso contrario pasar al punto 2.2.

   2.2. Si un punto de la primera cadena es interior al segundo recinto, la solución es el primer recinto, con lo cual el proceso queda terminado. En caso contrario la intersección es vacía y el proceso queda terminado.

3. En este caso se formarán sendas cadenas auxiliares que se denominan: cadena primera ampliada y cadena segunda ampliada.

   Las cadenas ampliadas se forman a partir de las cadenas originales incluyendo en ellas los puntos de intersección, de forma que se respete el sentido de recorrido de las cadenas adoptado. En las cadenas ampliadas deben distinguirse los vértices procedentes de las cadenas originales y los vértices procedentes de intersecciones. En el caso de que una intersección se produzca en un vértice de la cadena original ocurrirá que un mismo punto aparece dos veces en la cadena ampliada con distintas denominaciones. Además, los puntos de las cadenas ampliadas deberán marcarse para indicar si han sido o no utilizados en cadenas solución.

   Las cadenas que definan los recintos solución deberán estar necesariamente formadas por aristas pertenecientes a estas cadenas ampliadas y solo por ellas.

   Entonces, se toma el primer punto de la primera cadena ampliada y se pasa a 3.1.

   3.1. Comprobar si el punto analizado ha sido marcado como utilizado. En tal caso tomar el punto siguiente de la cadena y volver a 3.1.
   El proceso se detendrá cuando se hallan analizado todos los puntos de la primera cadena ampliada. Si el punto analizado no está marcado como utilizado se pasará al punto 3.2.

   3.2. Si el punto analizado es una intersección se pasará al punto 3.5. En caso contrario, se pasará al punto 3.3.

3.3. Se comprobará si el punto analizado es interior al segundo recinto. Si no lo es se marcará el punto como utilizado y se pasará al punto 3.4. y si el punto es interior al segundo recinto se pasará a 3.4.

3.4. Se considerará el punto siguiente de la primera cadena ampliada y se volverá a 3.1.

3.5. El punto considerado se tomará como origen de una cadena solución. Se incorporará el punto a dicha cadena solución, se marcará como utilizado y se pasará a 3.6.

3.6. El punto analizado se marcará como utilizado y se comprobará si la arista que sigue al punto, cuya longitud no sea nula, dentro de la primera cadena ampliada es interior al segundo recinto. En caso afirmativo se indicará que se ha de seguir por la primera cadena ampliada y se pasará a 3.8. En caso contrario, se seguirá en 3.7.

3.7. Se dará el punto como utilizado dentro de la segunda cadena ampliada. A continuación, se comprobará si la arista siguiente al punto, de longitud no nula, en la segunda cadena ampliada es interior al primer recinto. En caso afirmativo se indicará que debe seguirse por la segunda cadena ampliada y se pasará al punto 3.8. En caso contrario se pasará al punto 3.10.

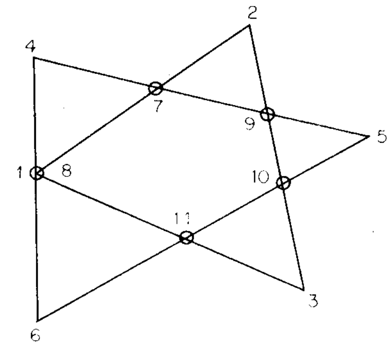
3.8. Se tomará el punto siguiente de la cadena elegida como continuación. Se comprobará si dicho punto está marcado como utilizado. En tal caso se pasaría a 3.10. En caso contrario se pasaría a 3.9.

3.9. El punto se marca como utilizado y se incorpora a la cadena solución. Si dicho punto fuese un vértice de la cadena original, se pasaría a 3.8. Si fuese una intersección, se pasaría a 3.6.

3.10. La cadena solución en curso está terminada. Se considerará entonces el punto siguiente al origen de la cadena solución, dentro de la primera cadena ampliada y se pasará al punto 3.1.

Una vez agotado este algoritmo se habrán obtenido todos los recintos solución, intersección de dos recintos poligonales cualesquiera.

La figura 5 esquematiza el proceso de intersección de dos recintos triangulares. En ella se marca los puntos de intersección de las aristas y se indican las cadenas originales, las cadenas ampliadas y la cadena que define el recinto solución.



CADENA 1: 1 2 3
CADENA 2: 4 5 6

CADENA 1 AMPLIADA: 1 8 7 2 9 10 3 11
CADENA 2 AMPLIADA: 4 7 9 5 10 11 6 8

CADENA SOLUCION: 8 7 9 10 11 1

Fig. 5. Proceso de formación de la cadena solución.

3. PROGRAMACION DEL ALGORITMO

El algoritmo descrito en el apartado anterior, ha sido programado en Fortran IV para su utilización en un mini-ordenador HP-21MX dotado de salida gráfica (plotter).

El programa se estructura mediante una serie de subrutinas y funciones cuyo cometido básico es:

- —Function VP (N1, N2, N3). Calculo el producto vectorial $\overline{N1}$, $\overline{N2}$, $\overline{N1}$, $\overline{N3}$.
- —Integer Function IN (I1, N). Determina si el punto N es interior a la cadena cuya etiqueta es I1.
- —Subroutine Chint (I1, J1, K1, NCS). Determina la intersección de los recintos definidos por las cadenas cuyas etiquetas son I1 y J1. Situa en NCS el número de cadenas que definen el recinto solución, y etiqueta dichas cadenas solución a partir de K1.

La descripción general del problema se hace mediante las variables NTP, X, Y, NPC y NV. Estas variables deben trasmitirse entre las subrutinas y con el programa principal por medio del Common. Su significado es el siguiente:

- —NTP, indica el número más alto asignado a algún punto, para su identificación, que intervenga en el problema. Contempla tanto los puntos que sirven para definir los datos originales, como aquellos que se generan durante la solución.

—X(I) e Y(I), contienen respectivamente las coordenadas X e Y del punto al que se ha asignado el número I.

—NPC(I), contiene el número de vértices que definen la cadena cuya etiqueta es I.

—NV(J,1), contiene el número correspondiente al vértice J ésimo de la cadena cuya etiqueta es I. NPC y NV se refieren tanto a cadenas que definen recintos dato como a las que definen recintos solución.

En algunos subprogramas, junto a estas variables principales, se utilizan algunas variables auxiliares, de las cuales las principales son:

—En Integer Function IN, la matriz de una dimensión NV (1) se utiliza para contener la descripción de la cadena auxiliar derivada a la que se hace referencia en el apartado "Posición de un punto respecto a un recinto". A este respecto hay que señalar que se toma como punto A el punto NV1(1), como punto B el punto NV1(2), y como punto C el punto NV1(3).

—En Subroutine Chint, el significado de las matrices auxiliares NA, NU, INTER, XL e IC, es como siguen. En la determinación de todas las posibles intersecciones entre aristas de ambas cadenas, INTER contendrá la descripción de cada intersección, registrando los números de orden de las aristas de la primera y segunda cadena que intersectan y el número de identificación asignado a la intersección. NU, contendrá en este caso el número de intersecciones que se produce en cada arista de ambas cadenas.

En la determinación de las cadenas ampliadas intervienen todas las matrices auxiliares. Al final de este proceso NA contendrá la descripción de las dos cadenas ampliadas, distinguiendo en ella los puntos que eran vértices de las cadenas originales de los puntos incorporados debidos a intersecciones, afectando a éstos últimos del signo menos. Para la formación de estas cadenas ampliadas se hace uso de los datos contenidos en las matrices auxiliares INTER y NU, procedentes de la determinación de las intersecciones. En el caso de que alguna arista perteneciente a una cadena tenga varias intersecciones, se utilizan las matrices XL e IC con objeto de ordenar estas intersecciones de forma que se respete el sentido de recorrido adoptado. La matriz NU se utilizará en lo que sigue para indicar si los puntos de las cadenas ampliadas han sido utilizados en la formación de cadenas solución. A tal fin cuando el valor correspondiente sea cero significará que el punto está aún disponible o por el contrario cuando sea uno que ya ha sido utilizado o desechado.

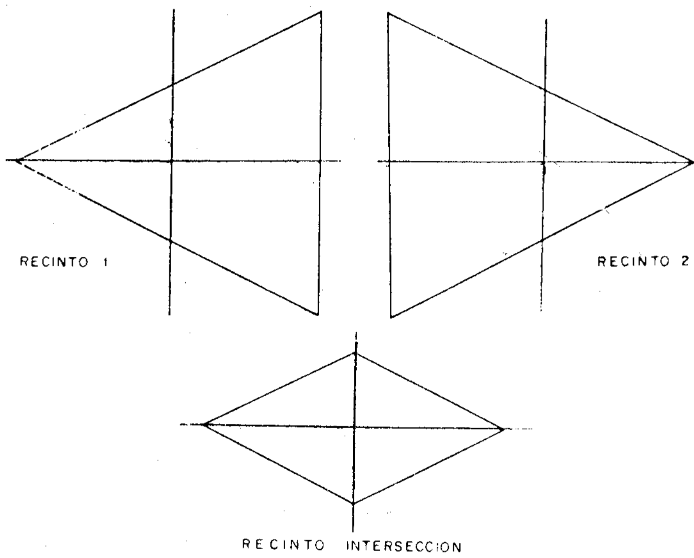


Fig. 6. Intersección de dos recintos simples y convexos.

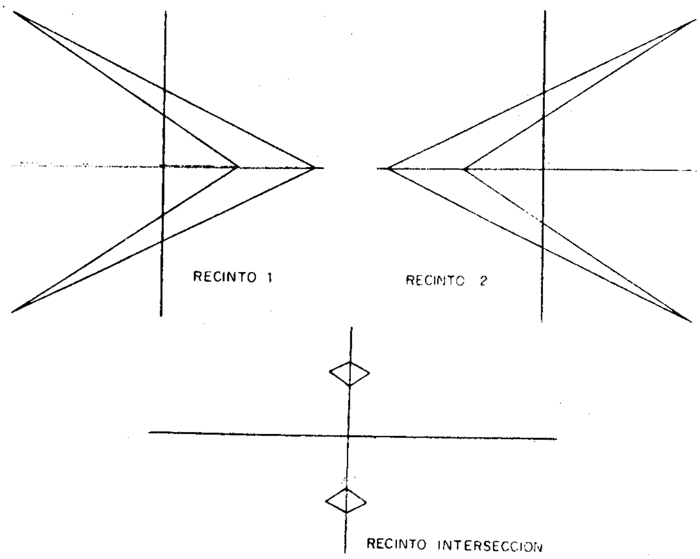


Fig. 7. Intersección de dos recintos simples no convexos, que da lugar a un recinto múltiple.

Debe señalarse que el control automático del valor de la variable NTP, así como la utilización de las matrices NPC y NV para describir las cadenas que definen tanto los recintos dato como los recintos solución facilita considerablemente la utilización repetitiva de la subroutina Chint en el caso de tener que determinar la intersección de varios recintos.

El apéndice que acompaña a este trabajo contiene los listados de ordenador de los subpro-

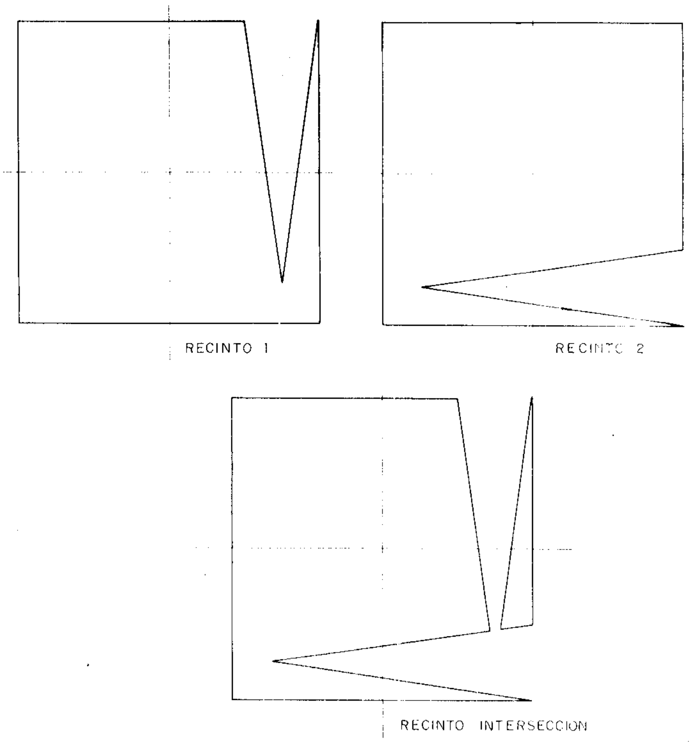


Fig. 8. Intersección de dos recintos simples no convexos, que da lugara un recinto múltiple.

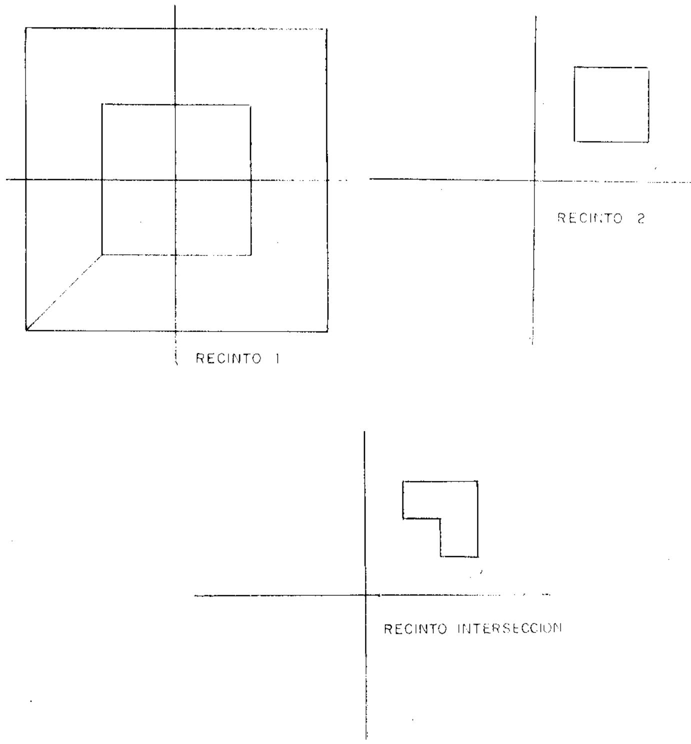


Fig. 10. Intersección de recinto múltiplemente conexo con recinto simple, que da lugar a recinto simple.

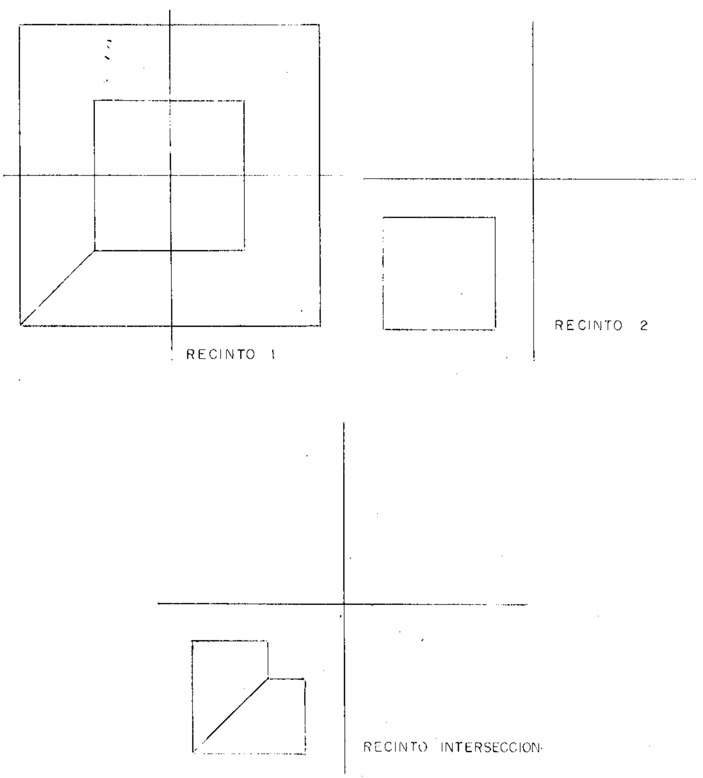


Fig. 9. Intersección de recinto múltiplemente conexo con recinto simple, que da lugar a recinto múltiple.

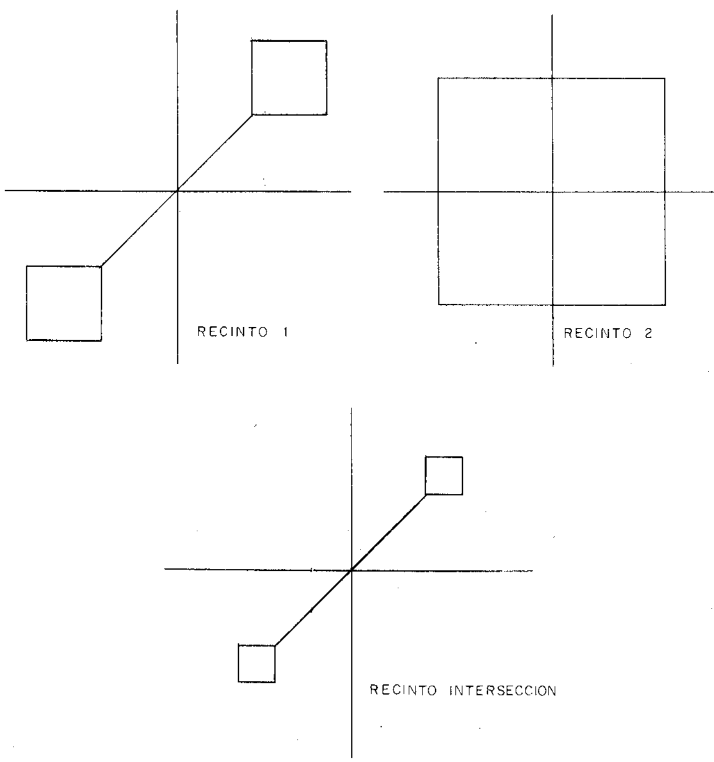


Fig. 11. Intersección de un recinto múltiple con un recinto simple, que da lugar a un recinto múltiple.

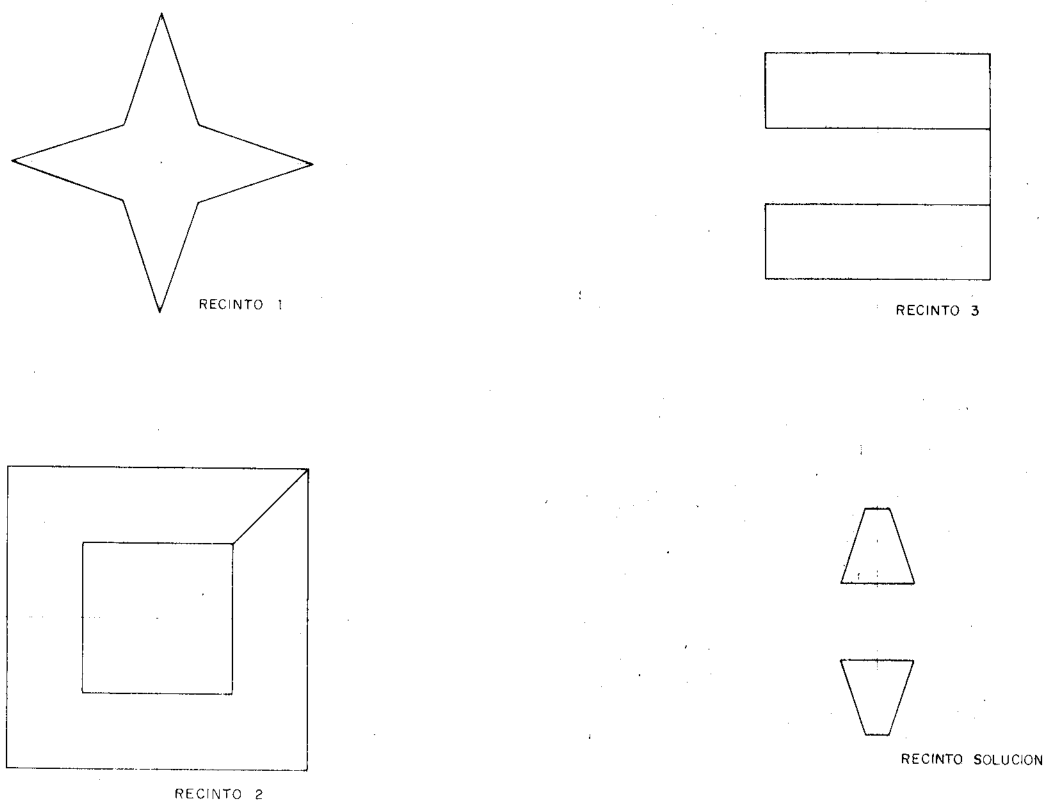


Fig. 12. Intersección de tres recintos.

gramas citados. Así mismo las figuras 6 a 12 muestran una serie de ejemplos de utilización del algoritmo, cubriendo una amplia gama de casos.

Para facilitar la comprensión de los ejemplos, los recintos empleados se sitúan referidos a un sistema de ejes cartesianos, que permiten la comprobación de los resultados mediante la superposición de las figuras.

4. CONCLUSIONES

El algoritmo que se presenta permite resolver el problema de intersección de recintos de forma cualquiera, y con cualquier tipo de conexión. Su programación es sencilla, tal como puede verse en los listados de ordenador que se acompañan, y su utilización resulta altamente eficiente.

Está básicamente orientado para la utilización de salida gráfica. No se incluye las subrutinas de dibujo que pueden variar de uno a otro computador pero que en cualquier caso son triviales puesto que, la subrutina de resolución proporciona la descripción del contorno mediante sus vértices y las coordenadas de los mismos.

La disposición de los datos permite encadenar de forma muy sencilla la utilización reiterada de la subrutina principal con objeto de obtener la intersección de un número cualquiera de recintos.

Los ejemplos que se presentan muestran una amplia casuística, incluyendo la intersección de dos o más recintos en los que, los recintos dato son simple o múltiplemente conexos, no conexos, convexos o no convexos. Los recintos solución pueden ser así mismo conexos o no e incluso reducirse a un punto o a una línea de contacto.

```
PAGE 0001        FTN4 COMPILER: HP24177  (SEPT. 1974)

      FUNCTION VP(N1,N2,N3)
C     PRODUCTO VECTORIAL DE LOS VECTORES N1,N2 Y N1,N3
      COMMON NTP,X(500),Y(500)
      VP=(X(N2)-X(N1))*(Y(N3)-Y(N1))-(X(N3)-X(N1))*(Y(N2)-Y(N1))
      RETURN
      END

**  NO ERRORS**    PROGRAM = 00108      COMMON = 02001
```

```
PAGE 0001        FTN4 COMPILER: HP24177  (SEPT. 1974)

      INTEGER FUNCTION IN(I1,N)
C     VERIFICA SI EL PUNTO ES INTERIOR A LA CADENA I1
C     IN VALDRA 0 SI EL PUNTO ES EXTERIOR Y 1 SI ES INTERIOR O DEL CONTO
      COMMON NTP,X(500),Y(500),NV(50,30),NPC(30)
      DIMENSION NV1(50)
      NPC1=NPC(I1)
      DO 1 I=1,NPC(I1)
    1 NV1(I)=NV(I,I1)
      IN=0
   10 IF(NPC1.LT.3)RETURN
      IF(VP(NV1(1),NV1(2),NV1(3)))20,30,40
C     EL TRIANGULO NV1(1),NV1(2),NV1(3) ES EXTERIOR A LA CADENA I1
   40 NAUX=NV1(1)
      DO 2 I=2,NPC1
    2 NV1(I-1)=NV1(I)
      NV1(NPC1)=NAUX
      GO TO 10
C     LOS PUNTOS NV1(1),NV1(2) Y NV1(3) ESTAN ALINEADOS
   30 NPC1=NPC1-1
      DO 3 I=2,NPC1
    3 NV1(I)=NV1(I+1)
      GO TO 10
C     EL TRIANGULO NV1(1),NV1(2),NV1(3) ES INTERIOR A LA CADENA I1
   20 IF(NPC1.EQ.3)GO TO 25
      DO 21 I=4,NPC1
      N1=NV1(1)
      N2=NV1(3)
      N3=NV1(I)
      N4=NV1(I-1)
      P1=VP(N1,N2,N3)
      P2=VP(N1,N2,N4)
      P3=VP(N3,N4,N1)
      P4=VP(N3,N4,N2)
      IF(P1*P2.LT.0.AND.P3*P4.LT.0)GO TO 40
   21 CONTINUE
   25 IF(VP(NV1(1),NV1(2),N).GT.0.OR.VP(NV1(2),
     *NV1(3),N).GT.0.OR.VP(NV1(3),NV1(1),N).GT.0)GO TO 30
      IN=1
      RETURN
      END

**  NO ERRORS**    PROGRAM = 00389      COMMON = 03531
```

```
PAGE 0001        FTN4 COMPILER: HP24177  (SEPT. 1974)

      SUBROUTINE CHINT(I1,J1,K1,NCS)
      COMMON NTP,X,Y,NV,NPC
      DIMENSION NV(50,30),NPC(30),X(500),Y(500),NA(100,2),NU(100,2)
     *INTER(3,50),XL(10),IC(10)
C     DETERMINACION DE LAS INTERSECCIONES
      NINT=0
      NCS=0
      DO 16 I=1,2
      DO 16 J=1,100
   16 NU(J,I)=0
      DO 1 I=1,NPC(I1)
      N1=NV(I,I1)
      N2=NV(I+1,I1)
      IF(I.EQ.NPC(I1))N2=NV(1,I1)
      DO 1 J=1,NPC(J1)
      N3=NV(J,J1)
      N4=NV(J+1,J1)
      IF(J.EQ.NPC(J1))N4=NV(1,J1)
      P1=VP(N1,N2,N3)
      P2=VP(N1,N2,N4)
      P3=VP(N3,N4,N1)
      P4=VP(N3,N4,N2)
      IF(P1*P2.GT.0.OR.P3*P4.GT.0.OR.P2.EQ.0.OR.P4.EQ.0)GO TO 1
      NINT=NINT+1
      NTP=NTP+1
      C1=-P2/(P1-P2)
      C2=P1/(P1-P2)
      X(NTP)=C1*X(N3)+C2*X(N4)
      Y(NTP)=C1*Y(N3)+C2*Y(N4)
      INTER(1,NINT)=I
      INTER(2,NINT)=J
      INTER(3,NINT)=NTP
      NU(I,1)=NU(I,1)+1
      NU(J,2)=NU(J,2)+1
    1 CONTINUE
      IF(NINT.GT.0)GO TO 10
C     NO EXISTE NINGUNA INTERSECCION ENTRE AMBAS CADENAS
      IF(IN(J1,NV(1,I1)).EQ.0)GO TO 2
C     EL PRIMER RECINTO ESTA CONTENIDO EN EL SEGUNDO
      NCS=1
      NPC(K1)=NPC(I1)
      DO 3 I=1,NPC(I1)
    3 NV(I,K1)=NV(I,I1)
      RETURN
    2 IF(IN(I1,NV(1,J1)).EQ.0)RETURN
C     EL SEGUNDO RECINTO ESTA CONTENIDO EN EL PRIMERO
      NCS=1
      NPC(K1)=NPC(J1)
      DO 4 I=1,NPC(J1)
    4 NV(I,K1)=NV(I,J1)
      RETURN
C     FORMACION DE LAS CADENAS AUXILIARES
   10 DO 11 L=1,2
      L1=I1
      IF(L.EQ.2)L1=J1
      NCONT=0
```

```
PAGE 0002 CHINT   FTN4 COMPILER: HP24177  (SEPT. 1974)

      DO 11 I=1,NPC(L1)
      NCONT=NCONT+1
      NA(NCONT,L)=NV(I,L1)
      IF(NU(I,L).EQ.0)GO TO 11
      NN=0
C     ORDENACION DE LOS PUNTOS DE INTERSECCION DE UNA ARISTA
      DO 12 K=1,NINT
      IF(INTER(L,K).NE.I)GO TO 12
      NN=NN+1
      IC(NN)=INTER(3,K)
      XDIF=X(NV(I,L1))-X(IC(NN))
      YDIF=Y(NV(I,L1))-Y(IC(NN))
      XL(NN)=SQRT(XDIF*XDIF+YDIF*YDIF)
   12 CONTINUE
      DO 13 K=1,NN
      K10=1
      DO 14 K2=1,NN
      IF(XL(K2).GE.XL(K10))GO TO 14
      K10=K2
   14 CONTINUE
      NCONT=NCONT+1
      NA(NCONT,L)=-IC(K10)
      XL(K10)=1.E+10
   13 CONTINUE
   11 CONTINUE
      DO 15 I=1,2
      DO 15 J=1,100
   15 NU(J,I)=0
C     FASE FINAL. FORMACION DE LAS CADENAS DE INTERSECCION
      NN=NPC(I1)+NINT
      K10=K1
      DO 30 I=1,NN
      IF(NU(I,1).EQ.1)GO TO 30
      N1=NA(I,1)
      IF(N1.LT.0)GO TO 31
      IF(IN(J1,N1).EQ.0)NU(I,1)=1
      GO TO 30
C     ORIGEN DE UNA NUEVA CADENA SOLUCION
   31 NCS=NCS+1
      NCONT=1
      NV(1,K10)=-N1
      NU(I,1)=1
C     CONTINUACION DE CADENA SOLUCION
C     CASO DE INTERSECCION
   40 DO 45 INDES=1,2
      L1=I1
      IF(INDES.EQ.2)L1=J1
      NN1=NPC(L1)+NINT
      DO 46 L=1,NN1
      IF(NA(L,INDES).EQ.N1)GO TO 47
   46 CONTINUE
   47 L2=L
      NU(L,INDES)=1
   48 L2=L2+1
      IF(L2.GT.NN1)L2=1
      N2=IABS(NA(L2,INDES))
```

```
PAGE 0003 CHINT   FTN4 COMPILER: HP24177  (SEPT. 1974)

      NTP1=NTP+1
      X(NTP1)=(X(N2)+X(-N1))/2.
      Y(NTP1)=(Y(N2)+Y(-N1))/2.
      IF(X(NTP1).EQ.X(-N1).AND.Y(NTP1).EQ.Y(-N1))GO TO 48
      L3=I1
      IF(INDES.EQ.1)L3=J1
      ICONT=IN(L3,NTP1)
      IF(ICONT.EQ.1)GO TO 50
   45 CONTINUE
      GO TO 60
C     BUSQUEDA DEL SIGUIENTE PUNTO DE LA CADENA SOLUJCION
   50 L=L+1
      IF(L.GT.NN1)L=1
      N1=NA(L,INDES)
      IF(NU(L,INDES).EQ.1)GO TO 60
      NU(L,INDES)=1
      NCONT=NCONT+1
      NV(NCONT,K10)=IABS(NA(L,INDES))
      IF(N1.GT.0)GO TO 50
      GO TO 40
   60 NPC(K10)=NCONT
      K10=K10+1
   30 CONTINUE
      RETURN
      END

**  NO ERRORS**    PROGRAM = 01810      COMMON = 03531
```