Universidad Nacional Autónoma de México

Facultad de Ingeniería

REPRODUCCION Y ANALISIS DE CURVAS PLANAS Y SOLIDOS DE REVOLUCION MEDIANTE DIGITALIZACION DE INFORMACION GRAFICA

TESIS PROFESIONAL

Que para obtener el Título de

INGENIERO EN COMPUTACION

P r e s e n t a n

ZENG CHAOMING

HE NING

RICARDO DUARTE PEREZ

Dir. Dr. JORGE ANGELES ALVAREZ

México, D. F. 1984

RESUMEN

En este trabajo se presenta un sistema de "software" desarrollado por los autores para obtener información gráfica de figuras planas mediante un digitalizador. Una vez digitalizada esta información, se forma un banco de datos que se utiliza para reproducir las curvas originales en un graficador o en una pantalla, y calcular algunas propiedades geométricas globales y locales, ya sea de curvas planas o de sólidos de revolución generados por éstas.

Se describen las bases teóricas y los métodos usados para lograr el desarrollo del sistema, así como la manera de usarlo y el equipo de digitalización y graficación empleado. Se muestran algunas pruebas y resultados experimentales obtenidos utilizando el sistema.

La aplicación de este trabajo a la solución de problemas de ingeniería se ilustra con múltiples ejemplos tomados de diversas aplicaciones.

El desarrollo detallado de las fórmulas computacionales para el cálculo de las propiedades geométricas globales y los programas se encuentran en los apéndices incluidos al final del trabajo.

Este trabajo se desarrolló totalmente en el Laboratorio de Cálculo Automatizado para el Diseño de la División de Estudios de Posgrado de la Facultad de Ingeniería, de la UNAM.

Cd. Universitaria, México.
Junio, 1984.

REPRODUCCION Y ANALISIS DE CURVAS PLANAS Y SOLIDOS DE REVOLUCION MEDIANTE DIGITALIZACION DE INFORMACION GRAFICA

CONTENIDO

CAPITULO 1

INTRODUCCION

Frecuentemente el ser humano necesita representar y modelar los objetos visibles del mundo real mediante dibujos planos simples. Por ejemplo, en la ingeniería, la arquitectura, el diseño y la manufactura de todo tipo de máquinas y equipo de transporte (automóvil, avión, barco, etc.), así como de bienes de consumo y de capital, los dibujos planos se utilizan continuamente a tal grado que llegan a jugar un papel esencial en estas actividades [1].

El uso de la información gráfica de los dibujos planos produce problemas de almacenamiento, duplicación, procesamiento y transporte. Esto se ha realizado tradicionalmente en forma manual. Sin embargo, no es económico en el sentido de eficiencia, exactitud, espacio de almacenamiento y tiempo de ejecución. A la luz del desarrollo trascedente de la tecnología de computadoras digitales y equipo periférico, los problemas mencionados anteriormente pueden resolverse adecuadamente mediante las técnicas modernas de digitalización y graficación en forma automática o semiautomática [2].

En el presente trabajo se presenta un sistema de "software" cuyo objetivo es realizar la digitalización, el almacenamiento, el procesamiento, la síntesis y la reproducción de los dibujos planos formados por curvas planas, así como algunas aplicaciones. El sistema consiste principalmente en un paquete de programas, que realizan algoritmos cuyas bases teóricas se describen en forma sucinta.

El equipo utilizado consiste en una minicomputadora digital PDP-11/40 con los siguientes periféricos: una teleimpresora, una pantalla, dos unidades de disco, un digitalizador y un graficador.

El sistema de "software" se puede dividir lógicamente en tres partes. La primera se encarga del manejo del digitalizador, de la obtención y del almacenamiento de información gráfica de curvas planas. El objetivo de la segunda parte es el procesamiento de los datos digitalizados, la síntesis de las curvas planas originales y su reproducción en el graficador o en la pantalla de manera automática e interactiva. La tercera parte es una aplicación de la digitalización y la graficación de las curvas planas, que consiste en

a) Cálculo de algunas propiedades geométricas globales de curvas cerradas y de sólidos de revolución.

b) Cálculo y representación gráfica de algunas propiedades locales de curvas planas.

Este trabajo está dividido en nueve capítulos e incluye dos apéndices. En el capítulo 2 se presentan los conceptos básicos de interpolación de funciones y de curvas planas. Se describe un método eficiente para la síntesis de funciones y de curvas planas, que se basa en el uso de funciones "spline" cúbicas, tanto no paramétricas como paramétricas.

En el capítulo 3 se describe brevemente la computadora digital y el equipo periférico utilizado.

El capítulo 4 trata del problema de digitalización y almacenamiento de información sobre curvas planas mediante el digitalizador. Se describe detalladamente el diseño de un menú para el digitalizador y el procedimiento de digitalización utilizando este menú.

Los problemas de procesamiento y de síntesis de curvas planas digitalizadas y su reproducción, tanto en el graficador como en la pantalla, se resuelven en el capítulo 5. Se aplican ampliamente las funciones "spline", en el proceso de síntesis de funciones y de curvas planas. También se incluye en este capítulo una breve descripción de las rutinas para el graficador y para la pantalla.

En el capítulo 6 se desarrollan las fórmulas y los algoritmos computacionales para calcular algunas propiedades geométricas globales de contornos encerrados por curvas planas y sólidos de revolución producidos por una curva generatriz plana; en este trabajo se considera solamente el área, el volumen, la localización del centroide, los momentos y los ejes principales de inercia. Estas curvas son digitalizadas y reproducidas mediante los métodos descritos en los capítulos anteriores. El desarrollo detallado de las fórmulas algebraicas se encuentra en el apéndice A, el cual se anexa al final del trabajo.

En el capítulo 7 se trata de una aplicación de la digitalizacion y la graficación de curvas planas. Esto es, calcular y representar gráficamente las propiedades geométricas locales como las derivadas primera y segunda, la curvatura, etc.

El capítulo 3 contiene algunos resultados experimentales del sistema descrito en los capítulos anteriores. Las pruebas de la digitalización y la reproducción incluyen dibujos tanto mecánicos como artísticos. Tambien se muestran los resultados de los cálculos de algunas propiedades geométricas de figuras planas irregulares y de sólidos de revolución tales como envases, etc. Por último, se presentan gráficamente las propiedades geométricas locales de algunas curvas planas de prueba.

Finalmente, en el capítulo 4, se concluye el trabajo con algunas recomendaciones para trabajo futuro y posibles aplicaciones.

Los listados de los programas y subprogramas brevemente documentados se encuentran en el apéndice B, todos estos están escritos en el lenguaje FORTRAN IV, y están disponibles en los discos del Laboratorio de Cálculo Automatizado para el Diseño (CAD) de la División de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI) de la UNAM, donde se desarrolló totalmente este trabajo.

CAPITULO 2

CONCEPTOS BASICOS DE INTERPOLACION Y FUNCIONES "SPLINE"

La interpolación es un problema de aproximación. El objetivo de este capitulo es presentar los conceptos basicos y algunos métodos de interpolación que se usan en el problema de síntesis de curvas planas. Especialmente se hace énfasis en la interpolación mediante funciones "spline" cúbicas tanto no paramétricas como paramétricas.

2.1 Conceptos básicos de interpolación

El problema de interpolación de funciones se puede describir de la manera siguiente [3, 4]:

Considérese una función real de una sola variable

$$y = f(x)$$

Por razones prácticas, se supone que esta función es "suave", es decir, f(x) varia en forma regular en el intervalo [a,b], donde f(x) es continua y sus dos primeras derivadas son igualmente continuas.

Dado un conjunto de puntos, denominados "puntos de apoyo", $\{x_k, f(x_k)\}$, $k = 1,\ldots,n$, se desea determinar una función $p(x)$ que aproxime $f(x)$ y que pase por todos los puntos de apoyo, es decir, $p(x_k) = f(x_k)$, $k = 1,\ldots,n$.

Es evidente que p(x) no es única, y depende del método de interpolación que se use. Los polinomios, por su simplicidad, son generalmente usados como funciones de interpolación. En casos especiales otros tipos de funciones (por ejemplo, trigonometricas, exponenciales, racionales) también se pueden usar [5].

Existen numerosos métodos de interpolación, de los cuales el más conocido tal vez es el de Lagrange [5], que consiste en evaluar el siguiente polinomio para interpolar n puntos de la función y=f(x):

$$P(x) = \sum_{k=1}^{n} L_k(x)\, y_k \qquad k = 1,\ldots,n$$

donde $y_k = f(x_k)$

$$L_k(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)}$$

Además del método de Lagrange, los métodos de Newton, de Hermite, etc., también se utilizan ampliamente [6].

En la siguiente sección se explicará el método de interpolación de funciones mediante las funciones "spline" cúbicas.

Sin embargo, los métodos anteriores no se aplican directamente cuando los puntos de apoyo son los puntos tomados de alguna curva plana arbitraria, la cual no es una función de tipo $y=f(x)$. Esta dificultad se supera, ya que en general, cualquier curva plana puede ser descrita por una ecuación de la siguiente forma:

$$F(x,y) = 0$$

Introduciendo un parámetro apropiado, por ejemplo t, la expresión anterior se puede convertir en dos funciones del parámetro t, es decir

$$x = x(t)$$
$$y = y(t)$$

Entonces, cualquier curva plana se puede tratar como dos funciones de un parámetro y los métodos de interpolación de funciones se aplican igualmente en el caso de que la curva plana no sea realmente una función del tipo $y = f(x)$.

En la última sección de este capítulo se presentan las funciones "spline" paramétricas para interpolar curvas planas; el parámetro de este tipo de funciones se puede interpretar como la longitud de la curva medida a partir de un punto inicial.

2.2 Funciones "spline" no paramétricas

En varios campos de la ingeniería se usa frecuentemente la "spline" (regla elástica), la cual se puede deformar de tal manera que pase por un conjunto de puntos (x_k, y_k), $k=1,\ldots,n$ [6]. Matemáticamente, $y = f(x)$ es la función llamada "spline" que, bajo las hipótesis de la teoría de la elasticidad, puede ser descrita por un conjunto de polinimios cúbicos $\{f_k(x)\}$, $k=1,\ldots,n-1$, de tal manera que $f(x)$ y sus dos primeras derivadas, $f'(x)$ y $f''(x)$, sean continuas para todo x dentro de un intervalo $[a,b]$. Sin embargo, la tercera derivada de una "spline" cúbica es discontinua. Los puntos

x_k, $k = 1,\ldots,n$ son llamados nodos de la "spline" y el conjunto $\{(x_k,y_k)\}$, $k=1,\ldots,n$, "puntos de apoyo".

El concepto de función "spline" se puede formalizar mediante la siguiente

DEFINICION [6-9]: Sea $a=x_1<x_2<\ldots<x_n=b$ una subdivisión del intervalo $[a,b]$. Una función "spline" $S(x)$ de grado p con nodos x_k, $k = 1,\ldots,n$, tiene las siguientes propiedades:

a) $S(x)$ es un polinomio de grado p en cada subintervalo $[x_k,x_{k+1}]$, $k = 1,\ldots,n-1$.

b) $S(x)$ y sus primeras $p-1$ derivadas son continuas en el intervalo $[a,b]$.

Las funciones "spline" empleadas en este trabajo son cúbicas, esto es, $p = 3$ en la definición anterior, por lo que

$$S_k(x) = A_k(x-x_k)^3 + B_k(x-x_k)^2 + C_k(x-x_k) + D_k$$
$$\text{para } k = 1,\ldots,n-1,\ x_k <= x <= x_{k+1} \qquad (2.1)$$

donde A_k, B_k, C_k, D_k son los coeficientes del polinomio cúbico definido en el intervalo $[x_k, x_{k+1}]$, aún por determinar. Nótese que existen $4(n-1)$ coeficientes desconocidos para los $n-1$ polinomios cúbicos de $S(x)$.

Ahora, dado un conjunto de puntos (x_k,y_k), $k = 1,\ldots,n$, de una función arbitraria continua $y = f(x)$, se desea aproximar $f(x)$ por una función "spline" cúbica $S(x)$. Las condiciones que se deben de cumplir son [7, 10]:

1) La función aproximada $f(x)$ y la función de interpolación $S(x)$ coinciden en los nodos x_k , $k = 2,\ldots,n-1$

$$y_k = S_k(x_k) = D_k$$
$$y_{k+1} = S_{k+1}(x_{k+1}) = A_k\Delta x_k^3 + B_k\Delta x_k^2 + C_k\Delta x_k + D_k$$
$$\text{donde } \Delta x_k = x_{k+1} - x_k \qquad (2.2)$$

2) Las derivadas primera y segunda de $f(x)$ y de $S(x)$ deben coincidir en los nodos, esto es:

$$y'_k = S'_k(x_k) = C_k$$
$$y'_{k+1} = S'_k(x_{k+1}) = 3A_k\Delta x_k^2 + 2B_k\Delta x_k + C_k \qquad (2.3)$$

$$y''_k = S''_k(x_k) = 2B_k$$
$$y''_{k+1} = S''_k(x_{k+1}) = 6A_k\Delta x_k + 2B_k \qquad (2.4)$$

Haciendo algunas operaciones algebraicas con las ecuaciones

(2.2), (2.3) y (2.4), se obtiene los coeficientes de la función "spline" en cada intervalo:

$$A_k = (y''_{k+1} - y''_k)/6\Delta x_k$$

$$B_k = y''_k/2$$

$$C_k = \Delta y_k/\Delta x_k - \Delta x_k(y''_{k+1} + 2y''_k)/6$$

$$D_k = y_k \qquad (2.5)$$

Hay que notar que las condiciones a satisfacer son insuficientes para determinar unívocamente todos los coeficientes de la "spline". Se necesita, por tanto, imponer dos condiciones adicionales de "frontera". Dependiendo de las diferentes condiciones de frontera, existen varios tipos de "spline". Dos de ellos se describen a continuación, los cuales son la "spline" natural y la "spline" periódica.

a)Función "spline" natural:

Por la continuidad de la función "spline", se sabe que:

$$S'_{k-1}(x_k) = S'_k(x_k) \qquad k=2,\dots,n-1 \qquad (2.6)$$

De las ecuaciones (2.3) y (2.6) se puede obtener la siguiente ecuación:

$$3A_{k-1}\Delta x_{k-1}^2 + 2B_{k-1}\Delta x_{k-1} + C_{k-1} = C_k \qquad (2.7)$$

Sustituyendo los coeficientes obtenidos en la ecuación (2.5), se tiene

$$\Delta x_{k-1} y''_{k-1} + 2(\Delta x_{k-1} + \Delta x_k) y''_k + \Delta x_k y''_{k+1} = 6\left(\frac{\Delta y_k}{\Delta x_k} - \frac{\Delta y_{k-1}}{\Delta x_{k-1}}\right) \qquad (2.8)$$

para $k=2,\dots,n-1$

donde $\Delta y_k = y_{k+1} - y_k \qquad k=1,\dots,n-1.$

Su forma matricial es:

$$\begin{bmatrix} 2(\Delta x_1+\Delta x_2) & \Delta x_2 & & & \\ \Delta x_2 & 2(\Delta x_2+\Delta x_3) & \Delta x_3 & & \\ & \Delta x_3 & 2(\Delta x_3+\Delta x_4) & \Delta x_4 & \\ & & \ddots & \ddots & \ddots \\ & & & \Delta x_{n-2} & 2(\Delta x_{n-2}+\Delta x_{n-1}) \end{bmatrix} \begin{bmatrix} y''_2 \\ y''_3 \\ y''_4 \\ \vdots \\ y''_{n-1} \end{bmatrix} = \begin{bmatrix} 6\left(\frac{\Delta y_2}{\Delta x_2} - \frac{\Delta y_1}{\Delta x_1}\right) - \Delta x_1 y''_1 \\ 6\left(\frac{\Delta y_3}{\Delta x_3} - \frac{\Delta y_2}{\Delta x_2}\right) \\ 6\left(\frac{\Delta y_4}{\Delta x_4} - \frac{\Delta y_3}{\Delta x_3}\right) \\ \vdots \\ 6\left(\frac{\Delta y_{n-1}}{\Delta x_{n-1}} - \frac{\Delta y_{n-2}}{\Delta x_{n-2}}\right) - \Delta x_{n-1} y''_n \end{bmatrix} \qquad (2.9)$$

La función "spline" natural considera que

$$y''_1 = 0 \;,\; y''_n = 0 \qquad (2.9a)$$

Esto implica geométricamente que la curvatura de la función sea nula en los dos puntos extremos.

De (2.9) y (2.9a) se puede obtener las y''_k, $k=1,\dots,n$. Sustituyendo las y''_k obtenidas en (2.5), se obtiene los coeficientes A_K, B_K, C_K, D_K de la función "spline" natural que aproxima la función $f(x)$.

b)Función "spline" periódica:

Se desea aproximar una función periódica $y=f(x)$ en un período, dados los puntos de apoyo (x_k,y_k) $k=1,\dots,n$ de esta función. Supóngase que los dos puntos extremos de $f(x)$ coinciden en la forma [7, 11]:

$$y_1^{(j)}=y_n^{(j)} \quad j=0,1,2 \qquad (2.10)$$

Se desea que la función "spline" periódica $S(x)$ tenga la misma característica, es decir:

$$S_1^{(j)}(x_1)=S_{n-1}^{(j)}(x_n)$$

$$j=0,1,2 \qquad (2.11)$$

Obsérvese que en los nodos x_k, $k=3,\dots,n-2$, las y''_k son iguales que en el caso de la función "spline" natural. Nada más en dos intervalos extremos de la función hay modificaciones. Utilizando

$$S'_1(x_1)=S'_{n-1}(x_n)$$

y sustituyendo los coeficientes de (2.5), se tiene

$$2(\Delta x_{n-1}+\Delta x_1)y_1'' +\Delta x_1 y_2'' +\Delta x_{n-1}y_{n-1}''=6\left(\frac{\Delta y_1}{\Delta x_1} - \frac{\Delta y_{n-1}}{\Delta x_{n-1}}\right) \qquad (2.12)$$

Por otra parte, de (2.8) y (2.10), se tiene

$$\Delta x_{n-1}y_1'' +\Delta x_{n-2}y_{n-2}''+2(\Delta x_{n-2}+\Delta x_{n-1})y_{n-1}''=6\left(\frac{\Delta y_{n-1}}{\Delta x_{n-1}} - \frac{\Delta y_{n-2}}{\Delta x_{n-2}}\right) \qquad (2.13)$$

Ahora se forma una ecuación matricial, con (2.12), (2.8) y (2.13), de la siguiente manera:

$$\begin{bmatrix} 2(\Delta x_{n-1}+\Delta x_1) & \Delta x_1 & & & \Delta x_{n-1} \\ \Delta x_1 & 2(\Delta x_1+\Delta x_2) & \Delta x_2 & & \\ & \Delta x_2 & 2(\Delta x_2+\Delta x_3) & \Delta x_3 & \\ & \ddots & \ddots & \ddots & \\ \Delta x_{n-1} & & & \Delta x_{n-2} & 2(\Delta x_{n-2}+\Delta x_{n-1}) \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \vdots \\ y_{n-1}'' \end{bmatrix} = \begin{bmatrix} 6\left(\frac{\Delta y_1}{\Delta x_1}-\frac{\Delta y_{n-1}}{\Delta x_{n-1}}\right) \\ 6\left(\frac{\Delta y_2}{\Delta x_2}-\frac{\Delta y_1}{\Delta x_1}\right) \\ 6\left(\frac{\Delta y_3}{\Delta x_3}-\frac{\Delta y_2}{\Delta x_2}\right) \\ \vdots \\ 6\left(\frac{\Delta y_{n-1}}{\Delta x_{n-1}}-\frac{\Delta y_{n-2}}{\Delta x_{n-2}}\right) \end{bmatrix} \qquad (2.14)$$

Nótese que $y''_n=y''_1$.

Resolviendo (2.14), se obtiene las y''_k, $k=1,\dots,n$. Los coeficientes de la función "spline" periódica se determinan al evaluar (2.5) con las y''_k obtenidas.

Existen algorítmos muy eficientes para resolver los sistemas de las ecuaciones (2.9) y (2.14). En el Apéndice B.7 se encuentran las rutinas para calcular las segundas derivadas [7].

2.3 Funciones "spline" paramétricas

En la vida real, las curvas planas no siempre son tan sencillas como para que puedan expresarse con funciones, es decir, no necesariamente se cuenta con una relación funcional dada entre las coordenadas de los puntos de una curva plana. En esta sección se introduce otro tipo de funciones "spline", las llamadas "splines" paramétricas [7], [12].

Sea una curva plana arbitraria C y un conjunto de puntos sobre C, (x_k, y_k), para $k=1,..,n$. Se puede aproximar C por la siguiente curva S descrita por sus ecuaciones paramétricas con un parámetro t :

$$S: \; x_k(t)=Ax_k(t-t_k)^3+Bx_k(t-t_k)^2+Cx_k(t-t_k)+Dx_k$$
$$y_k(t)=Ay_k(t-t_k)^3+By_k(t-t_k)^2+Cy_k(t-t_k)+Dy_k \qquad (2.15)$$
$$k=1,...,n-1$$
$$0<t_1<t_2<...<t_n, \; t_k<=t<=t_{k+1}$$

Donde $t_k=t_{k-1}+\Delta t_{k-1}$ para $k=2,...,n$

$t_1=0, \; t_k=(\Delta x_k^2+\Delta y_k^2)^{1/2}$ para $k=1,...,n-1$

Tanto $x_k(t)$ como $y_k(t)$ son funciones "spline" cúbicas, por lo que los coeficientes $Ax_k,...,Dx_k$ y $Ay_k,...,Dy_k$ se determinan utilizando los resultados obtenidos en la sección anterior.

Dos tipos de funciones "spline" paramétricas comúnmente usados son: función "spline" paramétrica natural y función "spline" paramétrica periódica, cuyas definiciones son:

a) Función "spline" paramétrica natural

$x''_1=x''_n=y''_1=y''_n=0$

b) Función "spline" paramétrica periódica

$x_1^{(j)}=x_n^{(j)}$

$y_1^{(j)}=y_n^{(j)} \quad j=0,1,2$

Las rutinas para calcular los coeficientes de las funciones "spline" paramétricas se encuentran en el Apéndice B.7.

Las funciones "spline" paramétricas son muy versatiles en

la síntesis de curvas planas [12], [13]. En este trabajo, las **funciones "spline"** paramétricas son utilizadas para calcular algunas **propiedades geométricas** de un contorno plano o de un sólido de **revolución,** lo cual se tratará posteriormente, en el capítulo **6.**

CAPITULO 3

DESCRIPCION DEL EQUIPO UTILIZADO

Para poder entender el sistema de "software" desarrollado en este trabajo, es necesario saber algo sobre el equipo en que se trabaja. En este capítulo se describe en forma breve el equipo (herramienta) usado como medio principal de operación. El equipo usado consiste principalmente de: Una Computadora PDP-11/40, una pantalla VT-11, un digitalizador Calcomp-622 y un graficador Calcomp-907/1039, como se muestran en las Fig.3.0.1-3.0.3, respectivamente.

3.1 La computadora PDP-11/40 y la pantalla VT-11

El sistema operativo RT-11 versión 4.0 de la computadora PDP-11/40 está diseñado para un solo usuario, que permite una amplia gama de periféricos; pero un solo programa en su memoria principal.

La computadora PDP-11/40 tiene una memoria de 28K palabras (56K "bytes"), dos unidades de disco, teniendo cada disco una capacidad de 2.2 Megabytes y una consola terminal. La memoria principal usada es ferromagnética ("magnetic core"), con tiempo de ciclo de 850 nanosegundos y tiempo de acceso de 350 nanosegundos [14]. Para facilitar su uso, la computadora cuenta con un sistema de programas que se mencionan a continuación [15].

a) Editor: Edición (Creación y modificación) de textos de programa.
b) Ensamblador: Convierte procedimientos(programas y subprogramas) de lenguaje ensamblador a lenguaje de máquina.
c) Compiladores e intérpretes: FORTRAN-IV, BASIC, etc. que generan código objeto.
d) Ligador: Combina y relocaliza programas objeto separados y genera programas ejecutables.
e) Bibliotecas: Rutinas de uso común.

La computadora PDP-11/40 usa un bus("UNIBUS") para comunicar y conectar los perifericos del sistema, cuya arquitectura se muestra esquemáticamente en la Fig. 3.1.1

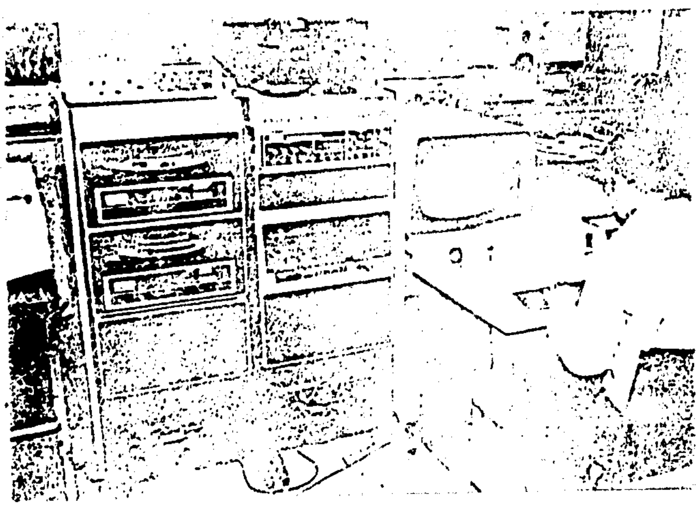


Fig 3.0.1 La computadora PDP-11/40 y la pantalla VT-11

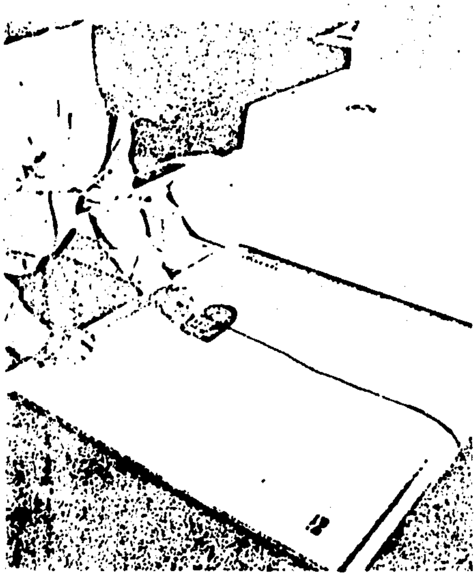


Fig 3.0.3 El graficador CALCOMP-907/1039

Fig 3.0.2 El digitalizador CALCOMP-622

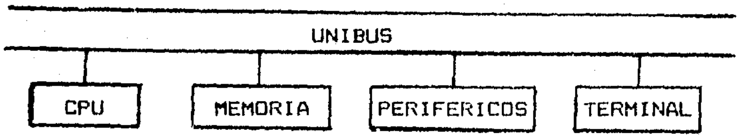


Fig. 3.1.1. PDP-11/40 y sus periféricos.

La pantalla VT-11 es un periférico de la PDP-11/40 y consiste de una terminal de tubo de rayos cátodicos (CRT) y de una pluma electrónica [16]. El área de la pantalla VT-11 es cuadrada y mide por lado 9.25 pulgadas (20.74cm). Normalmente, el número máximo de caracteres producidos en una línea es 73 y el número total de líneas que acepta es 31.

Para desplegar una imagen sobre la pantalla, el sistema coordenado x-y consiste de 1024 puntos individuales, con un total de 1.048,576 posiciones direccionables individualmente sobre el área efectiva de la pantalla. Si su dirección (posición) cae fuera del área, se trunca la imagen.

Los elementos de una imagen típica consisten de puntos y segmentos de recta y se definen de acuerdo con sus coordenadas relativas a la posición de un punto anterior. Estos combinan y forman imágenes de pantalla que son refrescadas por el CRT.

La pantalla VT-11 tiene un procesador autónomo (DPU, periférico de la PDP-11/40 para acceso directo a la memoria principal), cuyo requerimiento de memoria depende del requerimiento de programación, normalmente de 8K a 16K palabras. El DPU maneja instrucciones gráficas, de tal manera que libera el CPU de la PDP-11/40 de toda carga al ejecutar el programa gráfico.

El área de memoria donde se guardan las instrucciones gráficas y datos para construir una imagen se llama archivo de desplegado. Al ejecutar el DPU, esta área de memoria inicia con la primera instrucción y termina cuando encuentra una instrucción de fín. Este proceso se repite para formar una visión continua (refrescamiento de imagen producida).

El "software" requerido es el siguiente: un sistema operativo RT-11/04 y un paquete "DECgraphic" FORTRAN [16].

El "hardware" requerido es el siguiente: un CPU PDP-11, una memoria de 16K o más, un sistema de desplegados VT-11, una terminal de usuario, y dispositivos de almacenamiento usados por el sistema operativo RT-11/04.

3.2 El digitalizador CALCOMP-622

El digitalizador es un equipo especial que realiza conversión de posiciones físicas en un plano a datos digitales, los cuales pueden ser almacenados, transmitidos o procesados en forma apropiada por un sistema digital de cómputo. El digitalizador CALCOMP-622 comprende los siguientes componentes principales [17], [18], a saber, unidad de control, tabla de digitalización, cursor y cables de interfaces como se muestra en la Fig. 3.2.1

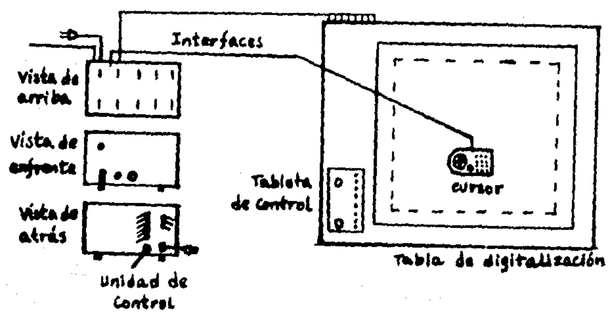


Fig.3.2.1 El digitalizador y sus componentes

Los componentes del digitalizador se describen a continuación:

1) Unidad de control: contiene un microprocesador que coordina la operación interna del digitalizador y realiza la conversión de señales analógicas (posición física de un punto) a datos digitales (par de coordenadas correspondiente, etc.); además tiene un interruptor de encendido y apagado, un indicador de encendido, una perilla de velocidad de conversión y varios conectores de cables de interfaces y de energía.

2) Tabla de digitalización: tiene una área activa de digitalización (incluye el área de trabajo y el margen) y una tableta de control ubicada en la esquina inferior izquierda de la

tabla de digitalización como se muestra en la Fig. 3.2.2

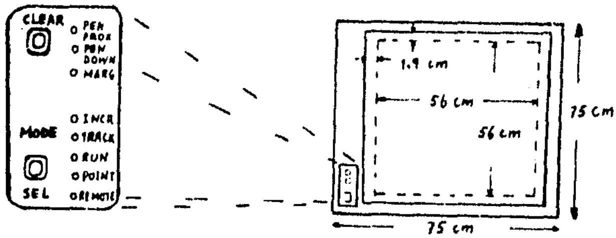


Fig.3.2.2 Tabla de digitalizador

Dentro del cuadrado marcado con líneas intermitentes se encuentra el área de trabajo; asociado a ésta hay un sistema cartesiano de coordenadas comúnmente denominado sistema coordenado del digitalizador, cuyo origen se localiza en la esquina inferior izquierda y los ejes de abscisas y de ordenadas coinciden, respectivamente, con las líneas del margen inferior y del izquierdo del área de trabajo.

En la tableta de control existen dos botones de control manual y ocho indicadores, cuyas funciones se explicarán posteriormente.

3) Cursor: sobre éste hay doce botones con diferentes símbolos, a saber, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, *, #, un indicador y una lupa a través de la cual se observa una cruz formada por dos hilos metálicos como se ve en la Fig. 3.2.3:

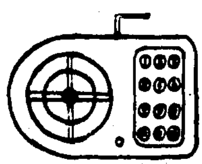


Fig. 3.2.3 El cursor del digitalizador

4) Cables de interfaces: La tabla de digitalización y el cursor están conectados a la unidad de control a través de los cables de interfaces internos; por otro lado, la unidad de control y la computadora (o cualquier sistema digital de recepción) se conectan

por los cables de interfaces externos.

La salida del digitalizador es una serie de símbolos y dígitos codificados en ASCII y es transmitida en la secuencia dada a continuación [18]:

B-espacio,
M-modo de operación,
Tecla-símbolo del botón usado del cursor,
X_1,
X_2,
X_3,
X_4,
Y_1,
Y_2,
Y_3,
Y_4,
CR-carácter de control.

donde X_4 y Y_4 son los dígitos más significativos de la abscisa y de la ordenada, respectivamente, del punto digitalizado, y éste está localizado por la cruz del cursor. Las abscisas X y las ordenadas Y, en diezmilímetros (0.1 mm), son números enteros no negativos y menores que 5600. La exactitud y la resolución del digitalizador son +0.025 centímetros y 100 puntos por centímetro respectivamente.

Existen cinco diferentes modos de operación:

1) Modo "POINT": Sólo una digitalización de un punto al oprimir una vez un botón del cursor; es decir, se obtiene una salida cuando se oprime y se suelta algún botón del cursor.

2) Modo "TRACK": Hay conversión continua mientras algún botón está oprimido.

3) Modo "RUN": Hay conversión continua mientras el cursor se aproxima a la superficie de la tabla de digitalización.

4) Modo "INCREMENT": Hay conversión cuando el cursor se mueve sobre la superficie de la tabla; en caso de que no haya movimiento del cursor, no habrá datos de salida.

5) Modo "REMOTE": Hay conversión cuando recibe señales externas.

Algunos controles manuales son:

1) "MODE SEL": Es un botón en la tableta de control, que sirve para selecionar modos de operación.

2) "CLEAR": Botón en la tableta de control, que sirve para establecer el par coordenado nulo (origen del sistema coordenado del digitalizador).

3) "RATE": Perilla en la unidad de control, sirve para ajustar la velocidad de conversión. La máxima velocidad de conversión que se puede obtener es 100 conversiones por segundo (se recomienda usar la máxima).

4) "ON": Interruptor de encendido y apagado, que se encuentra en la unidad de control.

Los indicadores tienen las siguientes funciones:

1) "PEN PROX": Es un diodo emisor de luz (LED) en la tableta de control, e indica que el cursor está dentro del área de operación de la superficie; es decir, que la separación entre el cursor y la superficie es menor que 12 milímetros.

2) "PEN DOWN": Indica que algún botón del cursor está activado.

3) "MARGIN": LED en la tableta de control; se prende cuando el cursor se encuentra fuera del área activa de la tabla de digitalización.

4) "POWER": LED en la unidad de control, indica que el equipo está energizado.

La operación y las aplicaciones de este equipo de digitalización se presentarán en los capítulos posteriores.

3.3 El graficador CALCOMP-907/1039

El graficador CALCOMP-907/1039 es un equipo de graficación electromecánico que produce caracteres, gráficas y mapas en cualquier medio común de operación (plumas de varios tipos, colores y tamaños, y en el papel preferido) [18]. Es importante mencionar que este equipo se maneja en forma fuera de línea ("off-line") en este trabajo.

El sistema de graficación consiste en "hardware" y "software". La Fig.3.3.1 muestra las diferentes componentes de este sistema.

Sistema de graficación
- "hardware"
  - graficador Calcomp
  - controlador
  - dispositivo de entrada
- "software"
  - programa de aplicación
  - "software" funcional
  - "software" básico

Fig. 3.3.1 Estructura de sistema de graficación Calcomp

Los términos importantes para este trabajo se mencionan a continuación.

a) Graficador Calcomp:

Este graficador se muestra en la Fig.3.3.2

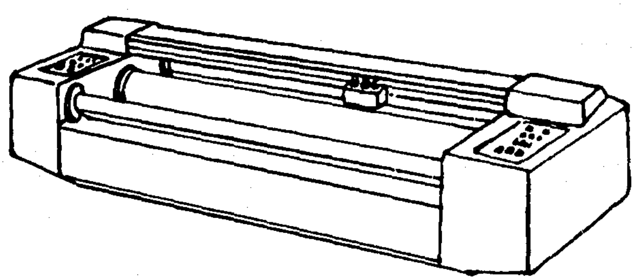


Fig.3.3.2 Graficador Calcomp 907/1039

El graficador puede estar en uno de dos modos, a saber, modo gráfico ("PLOT") o modo reposo ("STANDBY"), el cual selecciona el operador.

En el primer modo de operación el graficador traza de acuerdo con los comandos recibidos desde el controlador o desde la unidad de disco donde guarda los códigos. Por cada comando se traza un segmento de 0.05 mm de línea en una dirección especificada, Las direcciones básicas de la gráfica se muestran en la Fig. 3.3.3:

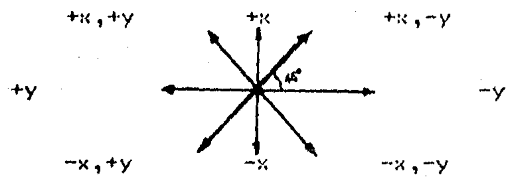


Fig. 3.3.3 Ocho direcciones básicas de la gráfica.

Como los segmentos de línea que se trazan son relativamente cortos, las curvas y líneas generadas son suficientemente suaves.

El modo reposo generalmente se usa para ajustar manualmente el graficador; por ejemplo, al mover la pluma a una posición preferida, al inicializar el graficador, al cambiar el papel, etc.

b) El controlador:

El controlador puede ser de uno de los siguientes tipos: unidad de disco, "hardware" y programable.

En este trabajo se utiliza el controlador de tipo de unidad de disco. Es decir, los comandos y datos pasan "directamente" desde una unidad de disco al graficador. Por eso, los comandos de graficación se deben generar en la computadora ("Host-Computer"), y se escriben en un archivo en disco.

c) Programa de aplicación:

El programa de aplicación ocupa el nivel más alto dentro del "software" de graficación. Como no se usa en este trabajo no se describe detalladamente.

d) "Software" funcional:

El "Software" funcional es el nivel intermedio del "software" de graficación, el cual permite a los usuarios realizar varias funciones de graficación. El "software" funcional está escrito en lenguaje FORTRAN y agrupado en paquetes para usos específicos.

Los paquetes principales son:

. Comercial ("BUSINESS")
. Dibujo ("DRAFTING")
. Científico ("SCIENTIFIC")
. General ("GENERAL")

Cada paquete consiste de un conjunto de rutinas para realizar algunas operaciones especificas: por ejemplo, la rutina CIRCL en el paquete general permite al usuario graficar un círculo o un arco circular; el usuario proporciona los parámetros de cada rutina.

e). "Software" básico:

El nivel más bajo en el "Software" de graficación es el "software" basico. Este software consiste en varias rutinas que permiten al programador realizar algunas operaciones elementales de gráficas; por ejemplo, definir un nuevo origen o trazar una recta o símbolos. También sirve para seleccionar plumas, escalar la gráfica, etc.

El "software" básico genera los comandos necesarios en la unidad de disco.

Hasta aquí se ha descrito brevemente el "hardware" y el "software" del sistema de graficación Calcomp. Las aplicaciones de este sistema se verán posteriormente en el capítulo 5, especialmente la reproducción de las gráficas planas digitalizadas.

CAPITULO 4

DIGITALIZACION DE CURVAS PLANAS

En el capítulo anterior se han descrito el funcionamiento y los componentes principales del digitalizador. En el presente capítulo se presenta el "software" diseñado para utilizar este equipo de digitalización y el procedimiento para obtener información gráfica de curvas planas utilizando el programa. Para empezar la descripción del funcionamiento del sistema de digitalización, es necesario definir la información gráfica de curvas planas que resulta de interés para este trabajo.

4.1 Información gráfica de curvas planas

La información gráfica contenida en una curva plana se define en este trabajo como un conjunto $\{(x_k,y_k)\}$, k=1,...,n que representa n pares de coordenadas cartesianas de sendos puntos contenidos en la curva. Este conjunto de puntos no es arbitrario, sino que debe ser lo suficientemente representativo como para proporcionar información lo más fiel posible sobre la curva. Por ejemplo, si se conocen el centro y el radio de una circunferencia, se puede trazar ésta sin requerir información adicional, puesto que la expresión matemática y los parámetros de la curva son conocidos. En caso de un segmento rectilíneo, una descripción sencilla y completa puede ser las coordenadas de los dos puntos extremos de éste. Es claro que el segmento rectilíneo puede ser reconstruido perfectamente a partir de esta descripción geométrica.

Sin embargo, la mayoría de las curvas planas que se encuentran en el mundo real no son tan simples como los segmentos rectilíneos o las circunferencias, pudiendo ser de forma arbitraria y sumamente complicadas de tal manera que no hay posibilidad de obtener descripciones sencillas, y a la vez completas, de ellas. Una estrategia para tratar de simplificar la recopilación de información de curvas complejas es dividirlas en un número razorable de segmentos más o menos simples y obtener una descripción geométrica sencilla y completa para cada uno de ellos. Para fines prácticos, se pueden clasificar todos los segmentos simples en tres tipos, a saber, rectilíneo, arco circular (incluyendo el círculo completo), y curva "suave" arbitraria. En este punto hay que observar que las curvas comúnmente usadas en las diversas aplicaciones de la ingeniería están compuestas por los segmentos de los tres tipos mencionados y por lo tanto la clasificación es válida para tales aplicaciones.

La información gráfica de los segmentos rectilíneos y de los círculos ya se mencionó anteriormente. Para los arcos circulares, la información gráfica consiste tan sólo de coordenadas de tres puntos sobre éstos, dos de los cuales son extremos. Finalmente, para las curvas suaves de forma arbitraria, la información gráfica no puede ser completa debido al desconocimiento de la expresión matemática de tales curvas. Una descripción aproximada puede ser una serie finita de propiedades geométricas tales como las coordenadas y pendientes de la curva, en un conjunto finito de puntos discretos (puntos de apoyo) sobre la curva. Es claro que mientras el número de puntos de apoyo sea mayor, más precisa será la información gráfica. En el límite, si se conocieran las coordenadas de todos los puntos sobre la curva, se podría reproducir perfectamente ésta. Esto implicaría el uso de una memoria de capacidad infinita, lo cual es materialmente imposible de obtener. Gracias a los métodos de interpolación y a las funciones "spline", tanto paramétricas como no paramétricas, todas las curvas suaves pueden aproximarse con cierta precisión a partir de las coordenadas de un conjunto finito de puntos de apoyo. Las funciones "spline" no son las únicas funciones de interpolación posibles, pero en este trabajo sólo se consideran éstas por las razones explicadas en el capítulo 2. De aquí en adelante se dirá que las curvas suaves de forma arbitraria son del tipo "spline", aunque en realidad se quiere decir que éstas son aproximadas por funciones "spline". En la siguiente tabla se muestra un resumen de los tres tipos de curvas.

Tipo de segmento	Información necesaria y suficiente para definir el segmento
Segmento de recta	Las coordenadas de los dos puntos extremos
Arco circular	Las coordenadas de tres puntos, dos de los cuales son extremos

Círculo	Las coordenadas de tres puntos cualesquiera sobre el círculo
Splines	Las coordenadas de un conjunto de puntos de apoyo

Las coordenadas se obtienen mediante el digitalizador. La información aún no está completa debido a que para poder hablar de las coordenadas de puntos sobre una curva plana, es necesario dar primero una descripción del sistema de coordenadas (en este trabajo se usa únicamente el sistema cartesiano). El sistema usado puede ser el mismo que el sistema del digitalizador, la descripción del cual se encuentra en el capítulo anterior, o puede obtenerse a partir de este último mediante las siguientes transformaciones :

1) Traslación del origen del sistema
2) Rotación de los ejes del sistema
3) Cambio de escala del sistema
4) Combinación de las transformaciones anteriores

Entonces, la descripción del sistema coordenado debe especificar el origen, el ángulo de rotación, y el factor de escala del nuevo sistema, obtenido a partir del sistema del digitalizador.

Hay dos detalles prácticos que deben tomarse en consideración, a saber: Primero, se pueden encontrar líneas de puntos en diversos dibujos de ingeniería; una forma de describir una línea intermitente es mediante la descripción de las partes sólidas de ésta sin considerar las partes en blanco; pero este método es tedioso y muy ineficiente. Esta dificultad se supera definiendo un nuevo tipo de curva que corresponde a cada uno de los tipos de curva ya definidos. Por ejemplo: segmento rectilíneo intermitente, "spline" periódica intermitente, etc. Segundo, se pueden encontrar dibujos demasiado grandes que no quepan en el área del digitalizador. Es necesario, en este caso, digitalizar las curvas en tales dibujos por partes, es decir, se divide un segmento de curva grande en varias partes que puedan caber completamente en el área del digitalizador. Se coloca la primera parte dentro del área del digitalizador y se digitaliza; después se quita el dibujo para colocar la segunda parte dentro de la misma área y se digitaliza, y así sucesivamente, para todas las partes, hasta que toda la curva se haya digitalizado. Es indispensable que la información gráfica incluya también una descripción del mecanismo para interconectar estas partes digitalizadas separadamente, lo cual se hace definiendo un nuevo sistema coordenado para cada parte con respecto a la anterior. El mecanismo se describe en el capítulo 5.

La información gráfica de las curvas planas debe obtenerse por medio del digitalizador, con ayuda de la terminal y la computadora. En la sección siguiente se tratará la comunicación entre el digitalizador y la computadora, ya que los datos digitalizados deben transmitirse a la computadora y almacenarse en un disco.

4.2 Comunicación básica entre el digitalizador y la computadora

Cada vez que se oprime un botón del cursor del digitalizador sobre el área, la unidad de control del digitalizador transmite una serie de símbolos y dígitos hacia la computadora a la cual está conectado. Estos símbolos y dígitos se decodifican en la siguiente información:

1) El modo de operación del digitalizador:

Modo	Código
"RUN" / "INCR"	0
"POINT" / "TRACK"	1

2) Los caracteres de los botones del cursor:

Caracteres	1	2	3	4	5	6	7	8	9	0	*	#
Códigos	1	2	3	4	5	6	7	8	9	0	10	11

3) La abscisa y la ordenada del punto digitalizado (punto localizado por la cruz del cursor). El par coordenado (en diezmilímetros) está constituido por dos números enteros positivos (X,Y), tales que 0=<X=<5589, 0=<Y=<5590 [17].

Se ha diseñado y desarrollado una rutina llamada COM que toma los datos enviados por el digitalizador y realiza la decodificación de la información acerca del modo de operación del digitalizador, los botones del cursor oprimidos y los pares de coordenadas de los puntos digitlizados. El listado de la rutina COM se encuentra en el apéndice A.1.

4.3 Diseño del menú para el digitalizador

Hay que observar que la comunicación básica descrita en la sección anterior es de tipo "simplex", es decir, la computadora recibe solamente la información transmitida por el digitalizador y no tiene ninguna acción directa sobre la operación del mismo. En realidad, el usuario es el operador del digitalizador y lo maneja de manera independiente de la computadora. Para que la computadora reconozca e interprete perfectamente la información recibida, es indispensable que el usuario obedezca un conjunto de instrucciones preestablecidas (menú del digitalizador).

El menú consiste de una definición de las funciones asignadas a cada botón del cursor del digitalizador y algunas reglas para su manejo, para que la operación del mismo sea sencilla, eficiente y comprensible para la computadora.

En este menú, la función del botón 1 es definir el origen del sistema coordenado de un nuevo segmento con respecto al sistema coordenado del digitalizador e indica también que el segmento debe ser conectado al segmento anterior siempre y cuando éste haya sido el primer botón oprimido para crear el segmento nuevo.

El botón 2 indica a la computadora que debe leer un factor de escala para un segmento introducido por el usuario a través de la terminal. El factor de escala debe ser un número real positivo.

El botón 3 define el ángulo de rotación: Si se oprime una sola vez, la computadora calcula el ángulo formado por la línea que une el origen del sistema y el punto digitalizado, y el eje horizontal del sistema de coordenadas del digitalizador; y si éste se oprime dos veces, el ángulo es el formado entre la línea definida por estos dos puntos y el eje horizontal del sistema coordenado del digitalizador.

El botón 4 se usa para indicar a la computadora que el tipo de segmento de una curva es rectilíneo.

El botón 5 sirve para definir el tipo de curva de un segmento como arco circular o círculo de la manera siguiente: si se oprime una vez, es considerado como el tipo del segmento en forma de arco circular; y si se oprime dos veces, el tipo del segmento a considerar es un círculo.

usuario estará listo de digitalizar la curva; por ejemplo:

Acciones del usuario	Respuesta correspondiente de la computadora
Oprime el botón *	Punto 1 = (3169. 1380.)
Oprime el botón *	Punto 2 = (3059. 1555.)
Oprime el botón *	Punto 3 = (2721. 1574.)
Oprime el botón 0	Punto 3 = (2721. 1574.) es eliminado.
Oprime el botón 0	Punto 2 = (3059. 1555.) es eliminado.
Oprime el botón *	Punto 2 = (3150. 1467.)
Oprime el botón *	Punto 3 = (2785. 1562.)
Oprime el botón *	Punto 4 = (2547. 1442.)
Oprime el botón *	Punto 5 = (2452. 1268.)
Oprime el botón *	Punto 6 = (2538. 1013.)
Oprime el botón *	Punto 7 = (2874. 972.)
Oprime el botón *	Punto 8 = (3125. 1244.)
Oprime el botón *	Punto 9 = (3168. 1379.)
Oprime el botón 9	Ajuste punto 9 por la terminal (x,y):
Escribe 3169,1380	Punto ajustado = (3169. 1380.)

En este ejemplo, se usó el botón 9 para ajustar el último punto e igualarlo al primero, porque la curva digitalizada fue una curva plana cerrada de tipo "spline" paramétrica periódica.

De manera similar se puede crear el segundo segmento definiendo sus parámetros y digitalizando los puntos de éste. Así, sucesivamente, para todos los segmentos de la curva. En el ejemplo se supone que sólo existe un segmento.

Para la digitalización, se procede de manera similar al siguiente ejemplo:

Acciones del usuario	Respuesta correspondiente de la computadora
Oprime el botón #	Escriba el nombre del archivo de datos digitalizados.
Escribe DK1:CURVAP.DAT ("RETURN")	Escoja el tipo de archivo de datos digitalizados: ?No-Formateado o Formateado? (N,F)
Escribe F ("RETURN")	?Desea un resumen de los datos digitalizados?(S,N)
Escribe S ("RETURN")	

El número de segmentos = 1. El número de puntos = 9.

El botón 6 se usa en forma similar a la del botón 5 y define el tipo de curva de un segmento como función "spline" natural o función "spline" periódica, en lugar de arco circular o de círculo, respectivamente.

El botón 7 tiene una función semejante a las de los dos botones anteriores. Se utiliza para definir el tipo de función "spline" paramétrica natural o de función "spline" paramétrica periódica, respectivamente.

El botón 8 se usa en combinación con alguno de los cuatro botones descritos anteriormente (botones 4-7) para seleccionar uno de los dos tipos de segmento: intermitente o continuo. Si se oprime una vez, el segmento será del tipo intermitente; si dos, será del tipo continuo.

El botón 9 se utiliza en caso de que el usuario cometa un error al digitalizar un punto sobre una curva, o si por alguna otra razón quiere ajustarlo introduciendo un par coordenado deseado a través de la terminal de la computadora sin usar el digitalizador. Hay que recordar que el botón 9 ajusta únicamente el último punto digitalizado en el momento de digitalización y no tiene ningún efecto sobre los puntos digitalizados anteriores del último.

El botón 0 está diseñado especialmente para eliminar sucesivamente los puntos digitalizados de una estructura de datos (ver la sección 4.4) del último punto hacia atrás. Los puntos eliminados son así soslayados por la computadora como si no hubieran sido digitalizados.

El botón * se usa para digitalizar un punto cuando el usuario lo oprime; es decir, con este botón se obtendrán los pares coordenados correspondientes a los puntos digitalizados.

Finalmente, el botón # indica a la computadora el fin del proceso de digitalización y que almacene toda la información gráfica obtenida durante el proceso de digitalización en un archivo creado en un disco de la computadora.

En la siguiente tabla se muestra un resumen de la asignación de funciones a cada botón del cursor:

Botones	Funciones
1	Origen del sistema coordenado

2	Factor de escala
3	Angulo de rotación
4	Línea recta
5	Arco circular
5,5	Círculo
6	"Spline" natural
6,6	"Spline" periódica
7	"Spline" paramétrica natural
7,7	"Spline" paramétrica periódica
8	Línea intermitente
8,8	Línea continua
9	Ajuste del punto recién digitalizado
0	Eliminación de puntos digitalizados hacia atrás
*	Digitaliza puntos
#	Almacena datos y termina el proceso

Algunas reglas de manejo del sistema de digitalización que debe seguir el usuario son las siguientes:

1) Para cada segmento de una curva plana se deben definir los parámetros del sistema coordenado asociado. Por convención, el sistema coordenado del segmento anterior es considerado como válido para el siguiente segmento, y el sistema coordenado del primer segmento coincide con el del digitalizador, es decir, el origen del sistema está ubicado en el punto (0,0), con ángulo de rotación nulo y factor de escala unitario.

2) Para cada segmento de curva, se asocia uno de los siguientes tipos de curva y solo uno:

a) Línea recta
b) Línea recta intermitente
c) Arco circular
d) Arco circular intermitente
e) Círculo
f) Círculo intermitente
g) "Spline" natural
h) "Spline" natural intermitente
i) "Spline" periódica
j) "Spline" periódica intermitente

k) "Spline" paramétrica natural

l) "Spline" paramétrica natural intermitente

m) "Spline" paramétrica periódica

n) "Spline" paramétrica periódica intermitente

Por convención se supone que el tipo de curva del primer segmento es línea recta continua.

3) Los parámetros asociados a cada segmento de una curva plana, tales como el tipo de curva de éste o los parámetros del sistema coordenado correspondiente, pueden definirse varias veces antes de empezar a digitalizar el segmento. Solo los parámetros definidos últimamente son considerados como válidos.

4) El botón 0 puede usarse tantas veces como se quiera, hasta que se eliminen todos los datos erróneos.

5) Para ajustar un punto anterior es necesario eliminar todos los puntos digitalizados después de éste usando el botón 9 en combinación con el botón 0.

6) Para crear un nuevo segmento de una curva plana se puede usar cualquiera de los botones 1-8.

7) Siempre es necesario usar el modo de operación "POINT" del digitalizador; en caso contrario, la computadora entra a un estado de espera hasta que se habilite el modo "POINT", y el usuario debe oprimir un "RETURN" para continuar.

8) El usuario puede reiniciar el proceso de digitalización después de almacenar un archivo de datos digitalizados; así, se pueden obtener varios archivos en una sola ejecución del programa, pero el número máximo de archivos que puede obtenerse no excede de nueve [15, 19].

No es necesario que el usuario memorice todas las reglas anteriores y la asignación de funciones de los botones. En efecto, la computadora informa al usuario del estado en que se encuentra el proceso de digitalización y los datos digitalizados para que éste pueda proceder. La computadora despliega en la teleimpresora o en la pantalla una tabla de asignación de funciones de los botones cuando se inicia el proceso de digitalización. En la última sección de éste capítulo se aclaran estas reglas de operación mediante la descripción de un procedimiento concreto de digitalización.

4.4 Estructura de datos y archivos usados para datos digitalizados

Durante el proceso de digitalización de curvas planss los datos digitalizados se registran dinámicamente haciendo uso de una estructura de datos y finalmente la información gráfica de un dibujo digitalizado se transfiere de la estructura de datos a un archivo de datos digitalizados. El archivo que contiene la información gráfica obtenida durante la digitalización es el productc que se desea, está almacenado en un disco y será usada posteriormente para efectuar su procesamiento, su cálculo o su reproducción.

En esta sección se describe en forma sucinta la estructura de datos y los archivos usados. Los aspectos prácticos acerca del uso de los datos digitalizados se presentarán en los capítulos subsecuentes.

La estructura de datos usada está formada principalmente por dos tablas de datos. La primera se usa para almacenar la descripción de diferentes segmentos de una curva plana, correspondiendo cada renglón de ésta a un segmento, y contiene los siguientes datos:

1) Un apuntador que indica la dirección de los puntos digitalizados del segmento contenido en una tabla de puntos digitalizados;
2) La longitud del segmento, es decir, el número de puntos que tiene éste;
3) El origen del sistema coordenado del segmento;
4) El ángulo de inclinación del eje horizontal del sistema coordenado con respecto al del sistema coordenado del digitalizador;
5) El factor de escala;
6) El tipo de curva del segmento;
7) Su relación con el segmento anterior (conectado o desconectado).

La tabla de segmentos se realiza por un arreglo estático bidimensional. Se usa un índice de segmentos para indicar el último segmento digitalizado en el momento de digitalización, éste incrementa en uno apuntando al siguiente renglón de la tabla, que corresponde a un nuevo segmento recientemente creado por el usuario. El contenido de este renglón se obtiene oprimiendo los botones correspondientes, de acuerdo con las reglas de operación establecidas en la sección anterior. También se puede quitar éste simplemente eliminando todos los puntos que lo constituyen y el índice de los segmentos se decrece en uno.

La segunda tabla almacena todos los puntos digitalizados de los diferentes segmentos y se realiza por medio de una lista de pares coordenados (arreglo estático bidimensional). Se usa un índice de puntos para indicar el último punto digitalizado durante el proceso de digitalización, y se incrementa en uno cuando el usario toma un punto nuevo de la curva usando el botón *, disminuyéndose en uno si el usuario elimina el punto digitalizado con el botón 0.

La tabla de puntos y la tabla de segmentos están relacionadas por medio del apuntador de direcciones de puntos digitalizados, el cual está contenido en cada renglón de la tabla de segmentos (se observa que el acceso a las dos tablas se realiza en forma de "pila").

Principalmente se dispone de dos tipos de archivos, a saber, archivo formateado y archivo no formateado. Cuando se usa un archivo formateado la computadora realiza una traducción entre las representaciones binaria y ASCII; por lo tanto los archivos formateados pueden ser convenientemente leídos y modificados mediante el editor del sistema operativo de la computadora. En cambio los archivos no formateados usan únicamente la representación binaria y consecuentemente no pueden ser leídos ni modificados directamente por el usuario. Sin embargo, los archivos no formateados son económicos en cuanto al espacio de almacenamiento y pueden tener tamaño prácticamente ilimitado, en contraste con los archivos formateados, que tienen un tamaño máximo de 764 registros [19]. El usuario del sistema de digitalización debe escoger una de las dos opciones para su archivo de datos digitalizados de acuerdo con su compromiso de comodidad y economía.

Un archivo de datos digitalizados está formado por la combinación de las dos tablas alternadas, es decir, cada renglón de la tabla de segmentos es seguida por una serie de puntos que lo constituye (ver la Fig.4.1).

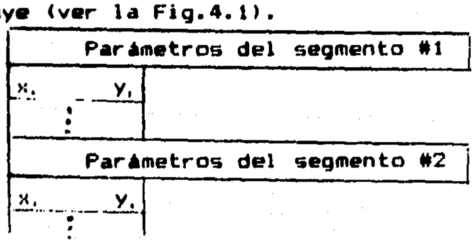


Fig.4.1 Archivo de datos digitalizados

4.5 Procedimiento de digitalización de curvas planas

Esta sección está dedicada a aquellos lectores que deseen usar el programa para el digtalizador en el Laboratorio de CAD de la DEPFI. Se describe en detalle el procedimiento de digitalización.

Antes de la ejecución del programa, es necesario que el usuario siga los siguientes pasos preparativos:

1) Encender la computadora PDP-11/40 y colocar el disco que contiene el programa en una unidad de disco de la computadora.

2) Encender el digitalizador y conectarlo a la computadora; habilitar el modo "POINT" y limpiar la memoria del controlador del mismo.

3) Fijar el dibujo que se va a digitalizar dentro del área del digitalizador; dividir adecuadamente la figura en segmentos, y escoger un juego de puntos discretos para cada segmento de las curvas planas en el dibujo.

Para ejecutar el programa el usuario teclea en la terminal:

```
RUN DK:DIGTLZ.SAV
```

sin dejar de oprimir la tecla "RETURN". La computadora responde en la pantalla el siguiente mensaje:

```
BIENVENIDO!
Verifique el modo de operación del digitalizador, y habilite el modo POINT. Quiere ver las instrucciones para el manejo del digitalizador? (S,N)
```

Si el usuario escribe S, la computadora despliega una tabla en la pantalla como sigue:

Instrucciones para el manejo del digitalizador

BOTON	FUNCION
1	ORIGEN DEL SISTEMA COORDENADO
2	FACTOR DE ESCALA
3	ANGULO DE ROTACION
4	LINEA RECTA
5	ARCO CIRCULAR O CIRCULO COMPLETO
6	SPLINE NATURAL O PERIODICA

7 SPLINE PARAMETRICA NATURAL O PERIODICA
8 TIPO DE CURVA INTERMITENTE
9 AJUSTE DE PUNTOS DIGITALIZADOS
0 ELIMINACION DE PUNTOS DIGITALIZADOS
* TOMA DE MUESTRAS DE PUNTOS
# ALMACENAMIENTO Y TERMINACION

Defina los parámetros y digitalice la gráfica. Adelante!

Si el usuario escribe N, la computadora sólo despliega la última frase.

El modo de operación debe ser el modo "POINT"; en caso contrario, la computadora responde con el siguiente mensaje:

El modo de operación del digitalizador debe ser POINT.
Use RETURN para continuar.
-PAUSE

En seguida, el usuario define los parámetros deseados usando los botones correspondientes. Por ejemplo, una típica secuencia de interacciones entre el usuario y la computadora es la siguiente:

Acciones del usuario	Respuesta correspondiente de la computadora
Oprime el botón 1	Se crea el segmento 1. El origen del sistema coordenado=(2631. 1159.)
Oprime el botón 2	Escriba el factor de escala(número positivo):
Escribe 2.5 ("RETURN")	El factor de escala = 2.5
Oprime el botón 3	El ángulo de rotación = 22.4 grados.
Oprime el botón 7	El tipo de curva es Spline paramétrica natural
Oprime el botón 7	El tipo de curva es Spline paramétrica periódica

Después de definir los parámetros del primer segmento, el usuario estará listo de digitalizar la curva; por ejemplo:

Acciones del usuario	Respuesta correspondiente de la computadora
Oprime el botón *	Punto 1 = (3169. 1380.)

Oprime el botón *	Punto 2 = (3059. 1555.)
Oprime el botón *	Punto 3 = (2721. 1574.)
Oprime el botón 0	Punto 3 = (2721. 1574.) es eliminado.
Oprime el botón 0	Punto 2 = (3059. 1555.) es eliminado.
Oprime el botón *	Punto 2 = (3150. 1467.)
Oprime el botón *	Punto 3 = (2785. 1562.)
Oprime el botón *	Punto 4 = (2547. 1442.)
Oprime el botón *	Punto 5 = (2452. 1268.)
Oprime el botón *	Punto 6 = (2538. 1013.)
Oprime el botón *	Punto 7 = (2874. 972.)
Oprime el botón *	Punto 8 = (3125. 1244.)
Oprime el botón *	Punto 9 = (3168. 1379.)
Oprime el botón 9	Ajuste punto 9 por la terminal (x,y):
Escribe 3169,1380	Punto ajustado = (3169. 1380.)

En este ejemplo, se usó el botón 9 para ajustar el último punto e igualarlo al primero, porque la curva digitalizada fue una curva plana cerrada de tipo "spline" paramétrica periódica.

De manera similar se puede crear el segundo segmento definiendo sus parámetros y digitalizando los puntos de éste. Así sucesivamente, para todos los segmentos de la curva. En el ejemplo se supone que sólo existe un segmento.

Para la digitalización, se procede de manera similar al siguiente ejemplo:

Acciones del usuario	Respuesta correspondiente de la computadora
Oprime el botón #	Escriba el nombre del archivo de datos digitalizados.
Escribe DK1:CURVAP.DAT ("RETURN")	Escoja el tipo de archivo de datos digitalizados: ?No-Formateado o Formateado? (N,F)
Escribe F ("RETURN")	?Desea un resumen de los datos digitalizados?(S,N)
Escribe S ("RETURN")	

```
El número de segmentos = 1. El número de puntos = 9.
Segmento,Indice,Longitud,Origen(x,y),Escala,Angulo,Curva,Conexión
   1        1.       9.   2631. 1159.   2.5   0.39    7.      0.
Punto      (x,y)
  1     (3169. 1380.)
  2     (3150. 1467.)
  3     (2785. 1562.)
  4     (2547. 1442.)
  - - - - - - - - -
  9     (3169. 1380.)
?Quiere continuar? (S,N)
Escribe  N ("RETURN")                    .(Fin)
```

Se puede volver a inicializar el proceso de digitalización y obtener un archivo nuevo de datos digitalizados, si se contesta la última pregunta tecleando S. Hay que recordar que el número máximo de archivos de datos digitalizados que se pueden crear es nueve, incurriéndose un error de ejecución si el usuario intenta rebasar este límite.

El programa DIGTLZ se encuentra en el Apéndice B.1.

CAPITULO 5

PROCESAMIENTO DE DATOS DIGITALIZADOS Y REPRODUCCION DE CURVAS PLANAS

En el presente capítulo, se describen las técnicas y el procedimiento para reproducir las curvas planas a partir de la información gráfica digitalizada. Obviamente, el procesamiento y la síntesis de las curvas planas son los pasos importantes para la reproducción. En la Fig.5.1 se muestra el procedimiento para la reproducción de las curvas planas.

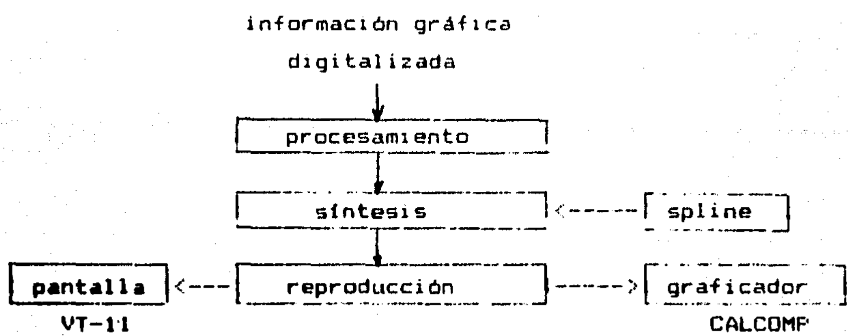


Fig 5.1 Procedimiento de la reproducción de curvas planas.

La importancia de esta técnica es mostrar diversas utilizaciones de la información gráfica digitalizada. Esta representa un gran futuro en muchos campos de la ingeniería y de otras disciplinas.

En este trabajo, los medios de la reproducción son el graficador y la pantalla.

5.1 Procesamiento de los datos digitalizados

El procesamiento de cualquier tipo de información se define como un proceso de transformación de los datos originales para lograr algún objetivo específico. En la sección 4.4 se describió la estructura de datos y el archivo de los datos digitalizados de las curvas planas. La información principal consiste en :

a). Los parámetros de cada segmento (número de puntos digitalizados, origen, factor de escala, ángulo de rotación, tipo de curvas, conexión).

b). Las coordenadas de los puntos (x_k, y_k). $(k=1,...,n)$

El procesamiento de los parámetros es simplemente un proceso de lectura y sus cambios, los cuales se realizan en forma interactiva.

Los puntos digitalizados tienen originalmente las coordenadas con respecto al sistema coordenado del digitalizador con el origen (0,0) y el ángulo de rotación nulo. Para algún uso especial, en la práctica hay necesidad de cambiar estos parámetros (origen, ángulo); por este motivo, hay que realizar una traslación y una rotación de los sistemas coordenados. Es decir, el procesamiento de las coordenadas de los puntos consiste en transformar éstas mediante algoritmos adecuados.

Sea el sistema de coordenadas X-Y para el digitalizador, y (X_1,Y_1) la coordenada de un punto cualquiera P en este sistema. Dado otro sistema de coordenadas deseado x-y con su origen (ox,oy) y el ángulo de inclinación θ con respecto a X-Y, para calcular la coordenada (x_1,y_1) del punto P con respecto al sistema nuevo(Fig 5.2), se tiene la siguiente transformación:

$$x_1 = (X_1-ox)\cos\theta+(Y_1-oy)\operatorname{sen}\theta$$
$$y_1 = -(X_1-ox)\operatorname{sen}\theta+(Y_1-oy)\cos\theta$$

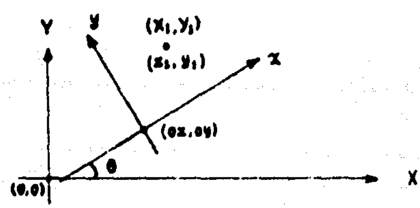


Fig 5.2 Sistemas coordenados y sus relaciones, donde (X_1,Y_1) es la coordenada original y (x_1,y_1) es la coordenada nueva

En caso de que haya conexión entre dos o más segmentos (ver la sección 4.3), también se puede aplicar esta técnica para que los segmentos tengan el mismo sistema de coordenadas, se conecten de forma deseada. Esto sucede cuando la curva que se digitaliza rebasa el área del digitalizador, y se necesita digitalizar toda la curva. Para ello, si se continúa con la parte faltante a digitalizar, se cambia el sistema de coordenadas. Por eso se procede al cambio de coordenadas (Fig 5.3), de la siguiente forma:

a) Toma el último punto (X_R, Y_R) de la primera parte de la curva como el punto de referencia.

b) Mueve la parte faltante de la curva al área del digitalizador, y se digitaliza esta parte incluyendo el punto de referencia.

c) Cambian las coordenadas de los puntos de la parte faltante, con respecto al origen (x_r, y_r) y al ángulo θ.

d) Traslada las coordenadas calculadas en el inciso anterior con respecto al punto de referencia (X_R, Y_R).

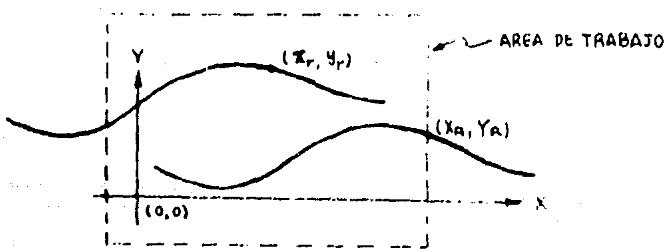


Fig 5.3 Conexión entre dos segmentos de una curva

Así se obtienen las coordenadas de los puntos de los diferentes segmentos, que se desean conectar.

Ahora bien, debido a los diferentes sistemas de medidas que usan el digitalizador y el graficador, es necesario hacer una conversión de unidades de diezmilímetros a pulgadas (254 dmm = 1 pulgada) [18]. El objetivo de ésto es lograr el mismo tamaño de la gráfica.

A continuación, se estudia la reproducción de curvas planas utilizando los datos procesados.

5.2 Generación de datos para la reproducción de curvas planas

Después de la digitalización de las curvas planas, los puntos discretos son las representaciones de estas curvas. Para reproducir estas curvas, se debe tener una descripción global de ellas. En este trabajo, cada tipo de curvas utiliza diferentes métodos para su reproducción.

1) Círculo o arco circular:

En este trabajo, la información necesaria para representar un círculo o un arco circular consiste en tres puntos sobre la circunferencia (para el caso de un arco circular, deben incluirse dos puntos extremos). Para reproducir el círculo y el arco circular, por medio de los tres puntos, se usan los siguientes cálculos, pudiendo obtenerse de ellos el radio, el centro y los ángulos extremos (especialmente para arco circular).

La ecuación de un círculo que pasa por tres puntos no-colineales, $P_1(x_1,y_1)$, $P_2(x_2,y_2)$ y $P_3(x_3,y_3)$, está dada por:

$$\begin{vmatrix} x^2+y^2 & x & y & 1 \\ x_1^2+y_1^2 & x_1 & y_1 & 1 \\ x_2^2+y_2^2 & x_2 & y_2 & 1 \\ x_3^2+y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0$$

La ecuación 5.1 se puede convertir en la siguiente forma:

$$x^2+y^2+Dx+Ey+F=0 \qquad (5.2)$$

El radio R está dado por $1/2(D^2+E^2-4F)^{1/2}$, y el centro C (x_c,y_c) está dado por $(-D/2,-E/2)$, donde:

$$D=[(x_3-y_2)(x_1^2+y_1^2)-y_1(x_2^2+y_2^2)-(x_3^2+y_3^2)+x_3(x_2^2+y_2^2)-y_2(x_3^2+y_3^2)]/[x_1(y_2-y_3)-y_1(x_2-x_3)+(x_2y_3-x_3y_2)]$$

$$E=[(x_3-x_2)(x_1^2+y_1^2)-x_1(x_2^2+y_2^2)-(x_3^2+y_3^2)+x_3(x_2^2+y_2^2)-x_2(x_3^2+y_3^2)]/[x_1(y_2-y_3)-y_1(x_2-x_3)+(x_2y_3-x_3y_2)]$$

$$F=[(x_2^2+y_2^2)(x_3y_2-x_2y_3)+x_1(y_2(x_3^2+y_3^2)-y_1(x_2^2+y_2^2))+y_1(x_2(x_2^2+y_2^2)-x_3(x_3^2+y_3^2))]/[x_1(y_2-y_3)-y_1(x_2-x_3)+(x_2y_3-x_3y_2)]$$

y los ángulos extremos (ver Fig 5.4) para el arco circular se calculan por:

ángulo inicial $\theta_1 = \tan^{-1}[(y_1-y_c)/(x_1-x_c)]$;

ángulo final $\theta_2 = \tan^{-1}[(y_3-y_c)/(x_3-x_c)]$;

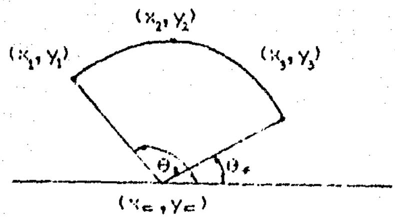


Fig 5.4 Angulos extremos para arcos circulares.

Con los parámetros calculados anteriormente, se sintetizan los círculos y los arcos circulares respectivamente. En la sección siguiente se verá que estos parámetros se aplican fácilmente en las rutinas del Calcomp para el graficador, con el objeto de reproducir los círculos y los arcos circulares.

Para la pantalla, como no existe una rutina que genere un círculo, se calculan los puntos de éste usando el parámetro R y los parámetros X_c, Y_c calculados anteriormente; para un incremento pequeño(dθ) de θ, se tiene:

$$X_{n+1} = X_c + R\cos(\theta + d\theta)$$

$$Y_{n+1} = Y_c + R\,\mathrm{sen}(\theta + d\theta)$$

Para calcular los puntos de un arco circular hay que considerar el sentido en que se digitaliza el arco circular.

Los sentidos a considerar son :

a) sentido antihorario (0-360 grados)

b) sentido horario (360-0 grados)

El cálculo para determinar este sentido se basa en los ángulos que forman los segmentos Y_1y_c, Y_2y_c, Y_3y_c con respecto al eje X del sistema coordenado del digitalizador y es el siguiente:

$\theta_1 = \tan^{-1}(Y_1 - Y_c / X_1 - X_c)$

$\theta_2 = \tan^{-1}(Y_2 - Y_c / X_2 - X_c)$

$\theta_3 = \tan^{-1}(Y_3 - Y_c / X_3 - X_c)$

Con el cálculo del sentido de digitalización se sabe cuál es el punto de cada extremo del arco circular(punto inicial P1 y punto final P3).

La abertura del ángulo formado por los segmentos P1C, P3C y el punto común C dan la longitud del arco circular a calcular, donde P1 es (X_1, Y_1) y P3 es (X_3, Y_3).

Los parámetros anteriores (sentido, longitud) son la base para calcular los puntos de cualquier arco circular, al evaluar éstos en la fórmula que calcula los puntos de un círculo.

Nótese que los círculos y los arcos circulares se usan frecuentemente en el dibujo mecánico.

2). Curvas arbitrarias:

En la práctica, las curvas generalmente son arbitrarias. Para este tipo de curvas, se aplican las funciones "spline" para interpolar debido a muchas razones; la principal es su exactitud y su versatilidad, o sea, las funciones "spline" paramétricas se pueden aplicar aunque las curvas planas no sean funciones del tipo $y = f(x)$, sino que presente puntos donde la curva se cruce a sí misma, por ejemplo.

En el capítulo 2 se han introducido los aspectos teóricos de las funciones "spline". Para la reproducción de curvas arbitrarias, se obtienen los coeficientes A_k, B_k, C_k, D_k del polinomio cúbico en cada intervalo entre dos puntos consecutivos (x_k, y_k) y (x_{k+1}, y_{k+1}), $k=1,...,n-1$.

Cuando la curva no es una función del tipo $y = f(x)$, se aplican las funciones "spline" paramétricas con un parámetro t, siendo sus coeficientes (Ax_k, Bx_k, Cx_k, Dx_k) y (Ay_k, By_k, Cy_k, Dy_k), respectivamente.

Una vez obtenidos los coeficientes de la función "spline", se pueden calcular los puntos intermedios en cada intervalo de acuerdo con la función "spline" obtenida, completándose así la reproducción de esta curva en forma discreta.

Las fórmulas para obtener los coeficientes de la función "spline" se encuentran descritas en el capítulo 2. Las subrutinas para la síntesis de curvas arbitrarias son las siguientes [7] y se encuentran en el Apéndice B.7:

a) CUBIC1: Calcula la segunda derivada para una función "spline" natural.

b) CUBICP: Calcula la segunda derivada para una función "spline" periódica.

c) CUBPAR: Calcula la segunda derivada para una función "spline" paramétrica natural.

d) CYCLIC: Calcula la segunda derivada para una función "spline" paramétrica periódica.

e) COEF: Calcula los coeficientes A_i, B_i, C_i, D_i, $i = 1,\ldots,n-1$, a partir de su segunda derivada y de los puntos dados.

f) EVAL: Calcula los puntos intermedios de cualquier función "spline" en cada intervalo (el número de puntos intermedios es un parámetro que el usuario puede definir segun su necesidad).

5.3 Reproducción de curvas planas con el graficador

La reproducción de las curvas planas, después del procesamiento y de la síntesis, es simplemente un proceso de graficación.

En base a las rutinas de CALCOMP para el graficador, e introduciendo los parámetros requeridos, se puede graficar diversos tipos de curvas planas. Para entender mejor este proceso, es necesario declarar las rutinas de CALCOMP (sólo las que se usan en este trabajo) que se mencionan a continuación [18]:

Rutina	Descripción
PLOTS(0,0,LDEV)	Inicializa la graficación y prepara un archivo para almacenar la información de las gráficas, con una unidad lógica especificada en LDEV.
PLOT(x,y,k)	Se mueve la pluma del graficador de la posición actual a la posición (x,y), y de acuerdo con el valor de k. Esta rutina tiene varias funciones.
FACTOR(fact)	Se reduce o amplifica la gráfica de acuerdo con el factor de escala que se proporciona.
LINE(x,y,np,inc,lt,inteq)	Grafica una línea recta por cada par de puntos del arreglo (x,y).
DASHL(x,y,np,inc)	Traza una línea recta intermitente por cada par de puntos de un arreglo (x,y).
CIRCL(x1,y1,th0,thf,r0,rf,di)	Genera un cículo o un arco circular de acuerdo con los parámetros especificados.
SYMBOL(x,y,altura,ibcd,ang,inteq)	Grafica n símbolos de acuerdo con el código(=inteq), el ángulo(=ang) y la altura(=altura) del símbolo, en la posición(x,y).

El programa principal de la reproduccion con el graficador CALCOMP es:

GRAFI: Este programa incluye todas las funciones de procesamiento, síntesis y reproducción de cualquier curva plana digitalizada, y usa las rutinas que se mencionan anteriormente para cada tipo de curvas.

El procedimiento de la ejecución del programa GRAFI para la reproducción de las gráficas es interactivo. Los pasos principales son:

1) RUN DK1: GRAFI

DK1 es la unidad del disco donde está el programa GRAFI. Inmediatamente se despliega en la pantalla lo siguiente:

2) ?PROCEDIMIENTO NORMAL O INTERACTIVO? (N o I)

?Quiere reproducir una gráfica con el procedimiento normal (predefinido) o con el procedimiento interactivo (define los parámetros durante la ejecucion del programa)? Los parámetros del procedimiento normal son predefinidos como:

a). número de puntos intermedios para la función "spline"=9

b). factor de escala para la reproducción= factor de escala original que se define en la digitalización

c). no habrá un símbolo en los puntos digitalizados

d). el formato de los archivos es formateado (observable).

Luego, se despliega

3) DEME EL ARCHIVO DE LOS DATOS DIGITALIZADOS:

Aquí pide el archivo donde guarda los datos digitalizados de la gráfica deseada. Por ejemplo, puede ser DK1: PERFIL.DAT. Si el archivo que se escribe no existe, se manda un error de entrada/salida, y se tiene que reinicializar la ejecución. Ahora, en la pantalla aparece

4) DEME UN ARCHIVO PARA EL GRAFICADOR:

Este archivo sirve para guardar la información de graficación que se genera durante la ejecución del programa. Por ejemplo, puede ser DK1: P.DAT. En este momento, si el procedimiento es normal, se ejecuta el programa hasta que termine el proceso. En caso contrario, se depliega una secuencia de preguntas durante la ejecución del programa, que es:

5) NUMERO DE PUNTOS INTERMEDIOS PARA LA "SPLINE":

Hay que dar un número natural. El efecto de números intermedios es la suavidad de las curvas arbitrarias.

Después, se despliega:

6) ?ARCHIVO FORMATEADO O NO FORMATEADO? (F o N)

Los archivos formateados tienen la ventaja de que los datos en el archivo son observables; pero los archivos no formateados tienen la ventaja de que ocupan menos espacio de memoria que los formateados.

Ahora bien, para cada segmento de la gráfica, se tienen los siguientes desplegados:

7) FACTOR DE ESCALA= **.**

DEME EL MULTIPLO DE LA ESCALA:

El número **.** es el factor de escala original que se define en la digitalización, si se desea cambiarlo, darle un múltiplo de la escala.

8) ?QUIERE UN SIMBOLO EN LOS PUNTOS DIGITALIZADOS? (S o N)

Esto sirve para verificar las posiciones de los puntos digitalizados, o también, para poner un símbolo especial en algún punto como flechas, etc..

En caso de que se quiera un símbolo en los puntos (S), se despliega:

9) DEME EL CODIGO Y LA ALTURA DEL SIMBOLO:

El código de los símbolos se encuentra en el manual de Calcomp(existen 14 diferentes símbolos). La altura del símbolo tiene que ser en pulgadas, y la anchura es igual a su altura [18].

Los desplegados 7), 8) y 9) se repiten en la ejecución de cada segmento de la gráfica hasta que se termine el proceso.

Cuando se termina la ejecución del programa GRAFI, para actualizar la graficación, se tiene que ejecutar una rutina MANDA, que manda la información de graficación al graficador, estando la información de graficación en el archivo que se especificó anteriormente. El procedimiento es simplemente como sigue:

10) RUN DK:MANDA

Después de esto, se pone inmediatamente el nombre del archivo que guarda la información para graficador; por ejemplo: DK1: P.DAT y después oprime la tecla RETURN. En este momento, el graficador realiza automáticamente la graficación que se desea.

Nótese que el procedimiento normal es más fácil y más rápido, pero con el procedimiento interactivo, se puede tener más opciones para la reproducción de las gráficas.

5.4 Reproducción de curvas planas con la pantalla

El avance de la tecnología ha cambiado muchas cosas; por ejemplo, anteriormente las pantallas (simple televisor convencional para que el receptor de TV actúe como pantalla de visualización) servían únicamente para introducir información por medio de un teclado y el resultado de procesar la información en la computadora se desplegaba en la pantalla. En la actualidadd se han desarrollado más usos para éstas, como el de obtener dibujos sobre la pantalla en forma interactiva o, más aún, en línea (on-line); por ejemplo, con una pluma (anexada a la pantalla para trazar líneas sobre ésta) se realizan dibujos y se modifican en tiempo real automáticamente.

Con el fin de usar todos los recursos del equipo que se encuentra en el Laboratorio de CAD (Cálculo Automatizado para Diseño) de la DEPFI (División de Estudios de Posgrado de la Facultad de Ingeniería), se considera graficar en la pantalla las curvas planas digitalizadas.

La reproducción de las gráficas en pantalla requiere de rutinas especiales, para lo cual hay que considerar que entre más grande sea la gráfica, más memoria requerirá.

Las rutinas en FORTRAN-IV que se encuentran en el manual RT-11 para pantalla [16] utilizadas en este trabajo son:

Rutina	Descripción
INIT ((n))	Limpia la pantalla, inicializa el desplegado de un archivo a usar(n palabras del área COMMON/ DFILE) y habilita los parámetros iniciales.
DISPLY (n)	Crea los archivos de desplegado rápidamente, suprimiendo el desplegado de la primitiva (ref) insertada.
APNT (x,y(,l,i,f,t))	Despliega un punto en la posición absoluta (x,y) con los parámetros opcionales l,i,f,y,t.
VECT (x,y(,l,i,f,t))	Dibuja un vector de la posición actual a la posición relativa representada por (x,y).

Las rutinas que se mencionan a continuación fueron realizadas en este trabajo:

Rutina	Descripción
WIND (WXL,WYB,WXR,WYT)	Verifica si los datos digitalizados se encuentren en el área standard.
VPORT (wxl,wyb,wxr,wyt)	Verifica si los datos usados para graficar en pantalla están en el área permitida.
CIRTRE (x1,y1,x2,y2,x3,y3,h,k,r)	Cálcula el radio y el centro de un círculo a partir de tres puntos dados.
ANGULO (x1,y1,x3,y3,h,k)	Calcula los ángulos iniciales y finales (abertura que hay en los segmentos P1C y P3C con respecto al eje x del sistema coordenado del digitalizador,donde P1 es el punto (x1,y1) ,C es el centro (h,k) y P3 es el punto (x3,y3)).
SC (x1,y1,x2,y2,x3,y3,h,k)	Calcula el sentido (horario o antihorario) en que se digitalizan los puntos de un arco circular.
CIR (x1,y1,x2,y2,x3,y3,h,k)	Cálcula los puntos de un círculo.
CIRPTO (x1,y1,x2,y2,x3,y3,h,k,xa,ya,n)	Programa principal para calcular los puntos de un círculo o de un arco circular.

Las rutinas que realizan el cálculo de las funciones "spline" se mencionaron en la sección 5.2.

Los programas principales son:

Rutina	Descripción
CALGRA	Calcula los puntos discretos de una curva y los almacena.
DIBGRA	Construye y grafica la curva en pantalla, permaneciendo en forma breve.

Para la descripción del procedimiento de ejecución de estos programas es importante señalar que:

DK1: indica el lugar del "drive" donde se encuentra el disco; en este caso el disco a usar se encuentra en el drive número 1.

FILENAME: indica el nombre del archivo a emplear; para ello se requiere escribir el nombre del archivo que se necesite, considerando que éste no tenga más de seis letras.

(R): Oprimir la tecla "RETURN".

La ejecución de los programas principales se lleva a cabo teniendo un programa generado para ejecución, o sea :

CALGRA.SAV y DIBGRA.SAV

A continuación se dan los pasos a seguir para la ejecución de estos programas:

1. a) Escribir RUN DK1:CALGRA (R)
   b) Escribir el nombre del archivo a grabar.
      DK1:FILENAME.DAT (R)
   c) Escribir el nombre del archivo a leer.
      DK1:FILENAME.DAT (R)
   d) Leer los datos digitalizados.
   e) Almacenar la información y terminar el proceso de ejecución.
2. a) Escribir RUN DK1:DIBGRA (R)
   b) Escribir el nombre del archivo a leer.
      DK1:FILENAME.DAT (R)
   c) Construye la gráfica y, al terminar, se despliega en la pantalla. Para quitarla basta con oprimir "RETURN".

Los programas de GRAFI y PANTAL se encuentran en los Apéndices B.2 y B.3 respectivamente.

CAPITULO 6

EVALUACION DE ALGUNAS PROPIEDADES GEOMETRICAS GLOBALES DE CURVAS CERRADAS Y SOLIDOS DE REVOLUCION

6.1 Definición del problema

En muchas ramas de la ingeniería, sobre todo en diseño mecánico, se requiere evaluar cantidades globales asociadas a contornos planos y a sólidos de revolución [20, 21].

Entre las diversas cantidades globales, las siguientes propiedades geométricas son de interés común y se consideran en este capítulo: el área, el volumen, el primer momento, el centroide, y los momentos y ejes principales de inercia. Matemáticamente, las propiedades mencionadas se definen por medio de integrales dobles o triples sobre el contorno o el sólido de revolución, y por lo general su cálculo preciso es muy elaborado. El objetivo del presente capítulo es presentar un método computacionalmente eficiente para calcular dichas propiedades geométricas globales asociadas a contornos planos y a sólidos de revolución. Este método es una aplicación directa de las técnicas de digitalización y síntesis de curvas planas descritas en los capítulos 4 y 5.

Aplicando el teorema de la divergencia debido a Gauss [22], las integrales múltiples se convierten en integrales sobre las fronteras de contorno o sobre la curva generatriz del sólido de revolución [20]. Es decir, conociendo las ecuaciones que describen la frontera de contornos planos o la curva generatriz de sólidos de revolución, teóricamente, las propiedades geométricas pueden calcularse mediante integrales sobre ellas.

Sin embargo, dado que en la práctica las expresiones matemáticas de las fronteras no siempre están disponibles (aunque estén disponibles, la evaluación directa de las integrales utilizando estas expresiones de frontera puede ser difícil o computacionalmente ineficiente), es conveniente obtener una aproximación de las fronteras utilizando las técnicas de digitalización de curvas planas, representando las fronteras mediante funciones "spline", con lo que las integrales se simplifican considerablemente.

En las siguientes secciones se dan las fórmulas utilizadas y se describe su realización en la computadora.

6.2 Fórmulas utilizadas

En esta sección se presentan las fórmulas desarrolladas para evaluar las propiedades geométricas globales de curvas cerradas y de sólidos de revolución.

Las deducciones de estas fórmulas se encuentran en [20].

a) Contornos planos

Sea C una curva plana cerrada, que no se interseca a sí misma, cuyas ecuaciones paramétricas son:

$$x = x(t), \quad y = y(t), \quad t = [t_o, t_n]$$

Se llama contorno plano S(C) a la región encerrada por la frontera C (ver Fig.6.1).

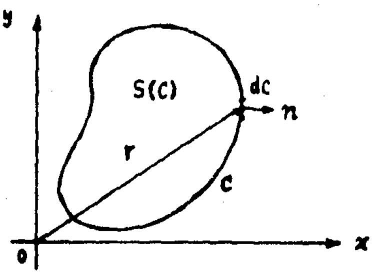


Fig.6.1 Contorno plano y su frontera,
r = vector de posición de los puntos sobre la frontera C,
n = vector normal unitario de la frontera C

1) El área del contorno S(C), denominada A, está dada por la siguiente integral sobre la frontera:

$$A = (1/2) \oint_C r^T n \, dC \qquad (6.1a)$$

donde

$r=[x,y]^T$, $n=[y',x']^T/(x'^2+y'^2)^{1/2}$,

$dC=(x'^2+y'^2)^{1/2}dt$, $x'=dx/dt$, $y'=dy/dt$, y T es la operación de transposición. Simplificando la expresión anterior se obtiene

$$A(S) = (1/2) \int_{t_o}^{t_n} (xy'-x'y)\ dt \qquad (6.1b)$$

2) El primer momento del contorno S(C) se calcula como la integral

$$Q(S) = (1/3) \oint_C (r^T n) r\ dC \qquad (6.2a)$$

sustituyendo las expresiones de r, n, y dC en (6.2a) y simplificando, se obtiene

$$Q(S) = (1/3) \int_{t_o}^{t_n} (x^2+y^2) [y', -x']^T\ dt \qquad (6.2b)$$

nótese que (6.2b) es una expresión vectorial.

3) El centroide del contorno S(C) está dado simplemente por

$$c(S) = Q(S)/A(S) \qquad (6.3)$$

4) Los momentos de inercia del contorno S(C) con respecto a un punto O están dados por la siguiente expresión matricial:

$$I^o = (1/2) \oint_C (r^T r)[(3/4)(r^T n)I - r n^T] dC \qquad (6.4a)$$

donde I es la matriz identidad de dimensión 2 y r es el vector de posición del punto, desde O. La versión simplificada es

$$I^o = (1/8) \int_{t_o}^{t_n} (x^2+y^2) \begin{bmatrix} -xy'-3x'y & 4xx' \\ -4yy' & 3xy'+x'y \end{bmatrix} dt \qquad (6.4b)$$

Los momentos de inercia con respecto al centroide se obtienen usando el teorema de los ejes paralelos [23]:

$$I^c = I^o - A \begin{bmatrix} y_c^2 & x_c y_c \\ x_c y_c & x_c^2 \end{bmatrix} \qquad (6.4c)$$

5) Los momentos principales de inercia y sus ejes correspondientes son los valores característicos y la dirección de los vectores característicos de I^o, cuyo cálculo se realiza mediante las siguientes fórmulas:

$$I_1 = (a+b)/2 + [(a-b)^2 + (2c)^2]^{1/2} \qquad (6.5a)$$

$$I_2 = (a+b)/2 - [(a-b)^2 + (2c)^2]^{1/2} \qquad (6.5b)$$

$\theta_1=\tan^{-1}[(I_1-a)/c]=\tan^{-1}[c/(I_1-b)]$ (6.5c)

$\theta_2=\tan^{-1}[(I_2-a)/c]=\tan^{-1}[c/(I_2-b)]$ (6.5d)

donde:

$$I^o=\begin{bmatrix} a & c \\ c & b \end{bmatrix}$$

b) Sólidos de revolución

Sea C una curva plana (no necesariamente cerrada) que no se interseca a sí misma y está ubicada a un lado de un eje denominado eje de revolución (convencionalmente es el eje de las abscisas OX). Al girar la curva C alrededor del eje OX, se genera un sólido de revolución V(C) cuya curva generatriz es C.

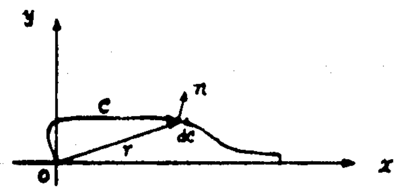


Fig.6.2 Sólido de revolución generado por una curva plana que gira alrededor de un eje.

Usando la misma nomenclatura que la parte anterior de esta sección, se tienen las siguientes fórmulas [20] y sus versiones simplificadas:

1) El volumen del sólido de revolución V(C) se calcula como sigue:

$$V = (2\pi/3)\int_C y(r^T n)\ dC \qquad (6.6a)$$

o simplemente como

$$V = (2\pi/3)\int_{t_o}^{t_n} y(xy'-x'y)\ dt \qquad (6.6b)$$

2) El primer momento del sólido de revolución V(C) está dado por

$$Q(V) = -\pi\int_{t_o}^{t_n} yy'(x^2+y^2)\ dt \qquad (6.7)$$

3) El centroide de V(C) está ubicado sobre el eje de revolución y la abscisa está dada por

$$c_x(V) = Q/V \qquad (6.8)$$

4) Los momentos de inercia de V(C) con respecto a los ejes OX y OY se obtienen de la siguiente fórmula:

$$[I_x, I_y]^T = \pi/10 \int_{t_0}^{t_n} y(x^2+y^2)[4xy'+6x'y, x'y-6xy']dt \qquad (6.9)$$

6.3 Algoritmos y su realización en la computadora

Se debe notar que todas las fórmulas presentadas en la sección 6.2 no pueden ser evaluadas si no se conocen analíticamente las ecuaciones paramétricas de la curva plana C (frontera de contorno plano o curva generatriz del sólido de revolución). El método que se usa en este trabajo para obtener una aproximación de las ecuaciones paramétricas de la curva plana C es precisamente una combinación de las técnicas de digitalización y de síntesis de curvas mediante funciones "spline" paramétricas. Como las "splines" son polinomios cúbicos definidos seccionalmente, una vez obtenidas esas ecuaciones paramétricas, al sustituirlas en las fórmulas (6.1)-(6.9), se obtiene un conjunto de fórmulas puramente algebraicas que pueden codificarse fácilmente en cualquier lenguaje de programación de una computadara digital; consecuentemente, el cálculo de esas integrales se vuelve considerablemente sencillo y no se necesita recurrir a ningún método numérico. En los Apéndices A, B.4 y B.5, que se incluyen al final de este trabajo, los interesados pueden encontrar las fórmulas desarrolladas para la computación, y los programas correspondientes en FORTRAN IV, respectivamente.

El procedimiento para el cálculo de las propiedades geométricas globales de contorno plano o de sólido de revolución es el siguiente:

1) Digitalizar la curva plana C en sentido antihorario (recomendado, aunque no necesariamente), utilizando el sistema de digitalización descrito en el capítulo 4 para obtener la información gráfica de C (en el caso de que C sea la frontera de un contorno plano cerrado,

el tipo de curva debe ser "spline" paramétrica periódica y el primer punto digitalizado debe coincidir exactamente con el último).

2) Sintetizar la curva C mediante las funciones "spline" paramétricas y realizar los cálculos de las propiedades geométricas globales utilizando los programas FRONT y SOLIDO, para contornos planos y sólidos de revolución, respectivamente.

CAPITULO 7

CALCULO Y REPRESENTACION GRAFICA DE ALGUNAS PROPIEDADES GEOMETRICAS LOCALES DE CURVAS PLANAS

Una clase de problemas técnicos en diversas áreas de la ingeniería involucra la determinación de las propiedades geométricas locales de curvas planas, o sea, su pendiente y su curvatura [24]. Tradicionalmente, estos problemas se han resuelto cuando la curva es una función descrita por una fórmula algebraica; pero esta clase de curvas es limitada. Generalmente, las curvas que se manejan en ingeniería son arbitrarias y no se pueden expresar con fórmulas algebraicas sencillas.

Introduciendo las funciones "spline" (ver capítulo 2), este problema se resuelve fácilmente con un conjunto finito de puntos sobre la curva. Es decir, se puede sintetizar la curva deseada con n puntos de apoyo, con lo que el problema se reduce a resolver un sistema de ecuaciones algebraicas [24].

El objetivo de este capítulo es obtener las propiedades geométricas locales de curvas planas y representarlas en forma gráfica. En este capítulo se analizan dos tipos de curvas: funciones $y=f(x)$ y curvas arbitrarias que se pueden sintetizar con las funciones "spline" paramétricas.

7.1 Algunas propiedades geométricas locales de las curvas planas del tipo $y=f(x)$

Dado un conjunto de puntos de apoyo (x_k,y_k), $k=1,\ldots,n$, de una curva plana, aplicando las técnicas de interpolación con funciones "spline", se puede obtener una ecuación:

$$S_k(x)=A_k(x-x_k)^3+B_k(x-x_k)^2+C_k(x-x_k)+D_k$$

en cada intervalo $[x_k,x_{k+1}]$, $k=1,\ldots,n-1$.

Algunas propiedades geométricas locales de este tipo de curvas pueden ser la primera derivada $S'(x)$ (pendiente) y la segunda derivada $S''(x)$, o sea:

$$S_k'(x)=3A_k(x-x_k)^2+2B_k(x-x_k)+C_k \quad (7.1)$$

$$S_k''(X)=6A_k(X-X_k)+2B_k \quad k=1,2,\ldots,n-1 \quad (7.2)$$

La representación gráfica de estas propiedades se logra utilizando las rutinas de Calcomp [18]. Los subprogramas para obtener estas propiedades se describen a continuación:

1) CUBIC1,CUBICP,COEF,EVAL (ver capítulo 5)
2) EVAL1: Calcula los valores de la primera derivada
3) EVAL2: Calcula los valores de la segunda derivada

7.2 Algunas propiedades geométricas locales de curvas arbitrarias

El problema abordado ahora es determinar las propiedades geométricas locales de curvas arbitrarias, que pueden ser pendiente y curvatura.

Ahora, dado un conjunto de puntos (x_k,y_k), $k=1,2,..,n$, de esta curva, aplicando las funciones "spline" paramétricas, se obtiene una aproximación de esta curva con dos funciones paramétricas $x(t),y(t)$, donde t es un parámetro de la curva, es decir:

$$x(t)=A_{xk}(t-t_k)^3+B_{xk}(t-t_k)^2+C_{xk}(t-t_k)+D_{xk} \quad (7.3)$$

$$y(t)=A_{yk}(t-t_k)^3+B_{yk}(t-t_k)^2+C_{yk}(t-t_k)+D_{yk} \quad (7.4)$$

siendo: $t_k=t_{k-1}+\Delta t_{k-1}$,

$\Delta t_k=(\Delta X_k^2+\Delta Y_k^2)^{1/2}$,

$\Delta X_k=X_{k+1} - X_k$,

$\Delta Y_k=Y_{k+1} - Y_k$, $k=1,2,...,n-1$.

La primera derivada(pendiente) de esta curva, se expresa con dos funciones paramétricas $y'(t)$, $y'(t)$, donde:

$$x'(t)=3A_{xk}(t-t_k)^2+2B_{xk}(t-t_k)+C_{xk} \quad (7.5)$$

$$y'(t)=3A_{yk}(t-t_k)^2+2B_{yk}(t-t_k)+C_{yk} \quad (7.6)$$

La segunda derivada se calcula de la siguiente manera:

$$x''(t)=6A_{xk}(t-t_k) +2B_{xk} \quad (7.7)$$

$$y''(t)=6A_{yk}(t-t_k) +2B_{yk} \quad (7.8)$$

La curvatura $f(t)$ de esta curva se obtiene con la combinación de las derivadas primera y segunda, es decir:

$$f(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'^2(t) + y'^2(t))^{3/2}} \quad (7.9)$$

Para visualizar las propiedades geométricas locales de este tipo de curva, se grafica en dos formas: en función de t(longitud) y en función de θ, el ángulo que forma el segmento punto-origen con respecto al eje horizontal. Los subprogramas que realizan este cálculo son:

1) CUBPAR, CYCLIC, COEF, EVAL (ver capítulo 5)
2) EVAL1: Calcula los valores de las primeras derivadas de $x(t)$ y $y(t)$.
3) EVAL2: Calcula los valores de las segundas derivadas de $x(t)$ y $y(t)$.
4) CURV: Calcula los valores de curvatura.

El programa principal que realiza los cálculos y las representaciones gráficas de las propiedades geométricas locales de cualquier tipo de curvas planas se denomina LOCAL.

El procedimiento a seguir para calcular las propiedades geométricas locales de curvas planas es el siguiente:

a) Digitalizar la curva deseada.
b) Ejecutar el programa LOCAL escribiendo RUN DK1:LOCAL
c) Escribir el nombre del archivo de los datos digitalizados.
d) Escribir el nombre del archivo para la gráfica
e) Escribir un factor de escala deseado para la gráfica.
f) Escribir la longitud deseada para los ejes coordenados de la gráfica.

Si el tipo de curva es una función $y=f(x)$ pasa a h).
Si el tipo de curva es arbitraria,

g) Selecciona una opción (t o θ).
h) Termina el proceso de cálculo.

Para graficar las propiedades geométricas locales, se realiza lo siguiente:

1) Escribir RUN DK:MANDA
2) Escribir el nombre del archivo para gráfica y se obtiene la gráfica deseada.

Los programas para calcular las propiedades geométricas locales se encuentran en el apéndice B.6.

La determinación de las propiedades geométricas locales de curvas planas tiene muchas aplicaciones en la ingeniería mecánica; por ejemplo, en el análisis y diseño de mecanismos con seguidores de levas [25, 26, 27].

CAPITULO 8

RESULTADOS EXPERIMENTALES

En este capítulo se muestran algunos resultados obtenidos mediante el "software" desarrollado; primero se muestran las gráficas originales y sus reproducciones; después, los resultados de cálculos de las propiedades geométricas globales de contornos planos y sólidos de revolución, y finalmente, los resultados y representaciones gráficas de las propiedades geométricas locales de curvas planas.

Las Fig 8.1 y Fig 8.2 son dos dibujos artísticos originales de Picasso, cuyas reproducciones reducidas se encuentran en las Fig 8.3 y Fig 8.4 respectivamente. La razón por que se reprodujeron los dibujos artísticos estriba en que en tales dibujos se encuentra una gran variedad de curvas planas, las cuales no son sólo dificiles de obtener matemáticamente, sino también útiles para evaluar el desempeño del método de síntesis utilizando las "splines" estudiadas en este trabajo.

La Fig 8.5 muestra un dibujo mecánico, en el cual se encuentran todos los tipos de curvas considerados en el trabajo, es decir, el dibujo contiene líneas rectas, círculos o arcos circulares, curvas arbitrarias, y líneas intermitentes. Por lo tanto, éste es apropiado para la evaluación del desempeño del sistema de digitalización y reproducción. Una reproducción de la Fig 8.5 se muestra en la Fig 8.6.

La Fig 8.7 es un dibujo original de una botella de cerveza cuyos resultados del cálculo de las propiedades geométricas globales, a saber, el volumen, el primer momento, el centroide, y los mementos de inercia, se obtuvieron experimentalmente [28] y se encuentran consignados en la página siguiente de la Fig 8.7. Este dibujo se seleccionó con el objeto de comparar los resultados de este "software" con los obtenidos experimentalmente.

Finalmente, las Fig 8.8 y Fig 8.9 representan a una leva, y son sintetizadas a partir de un conjunto de puntos de apoyo obtenidos en [25] con factor de escala iqual a 1 y a 0.5, respectivamente. Los resultados del cálculo de las propiedades geométricas locales, a saber, la pendiente y la curvatura, se encuentran representadas en forma gráfica, en las Fig 8.10 y Fig 8.11. En seguida se dan también los resultados del cálculo de las propiedades geométricas globales de la leva (el área, el centroide, los momentos y ejes principales de inercia, etc.). Tanto las propiedades locales como las globales pueden ser de gran utilidad para el diseño de las levas y las piezas mecánicas en general.

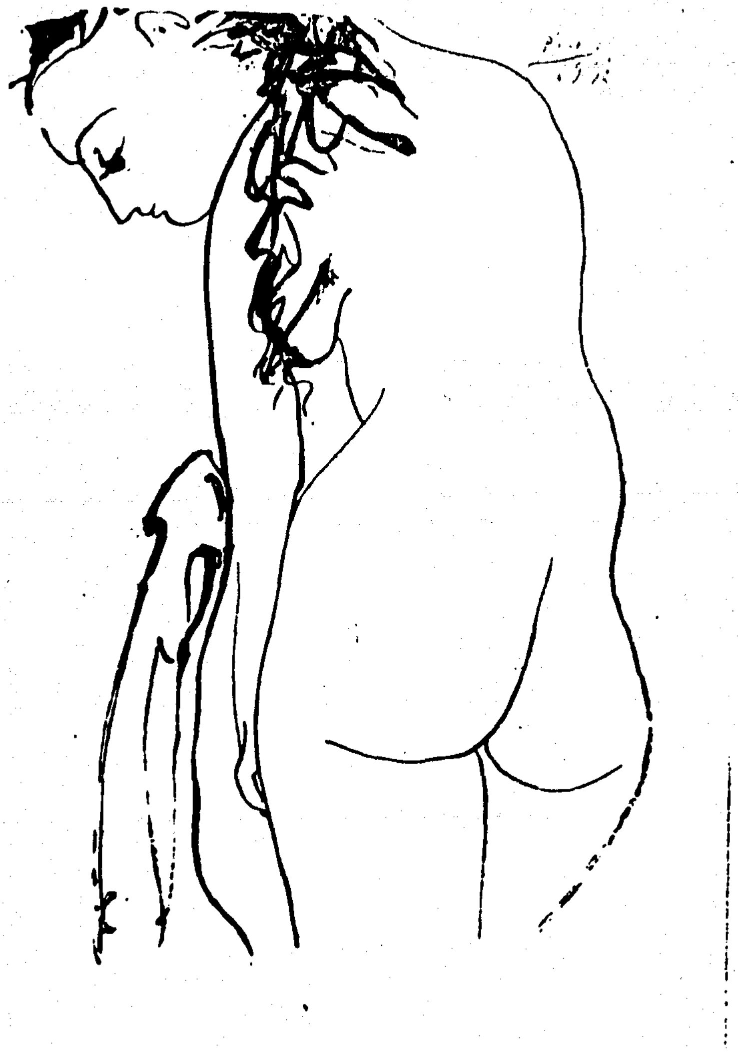


Fig 8.1 Desnudo de Espalda (por Picasso,1958)

Fig 8.2   Desnudo Lateral (por Picasso,1958)

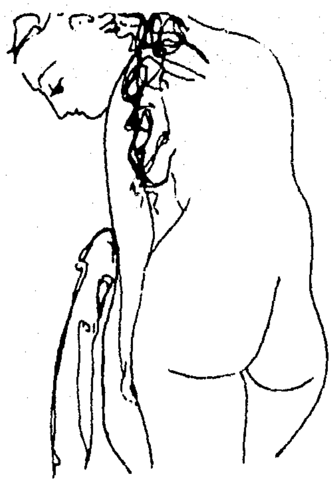


Fig 8.3 Reproducción de la Fig 8.1 con "splines" paramétricas naturales y un factor de escala iqual a 0.5

Fig 8.4 Reproducción de la Fig 8.2 con "splines" paramétricas naturales y un factor de escala iqual a 0.5

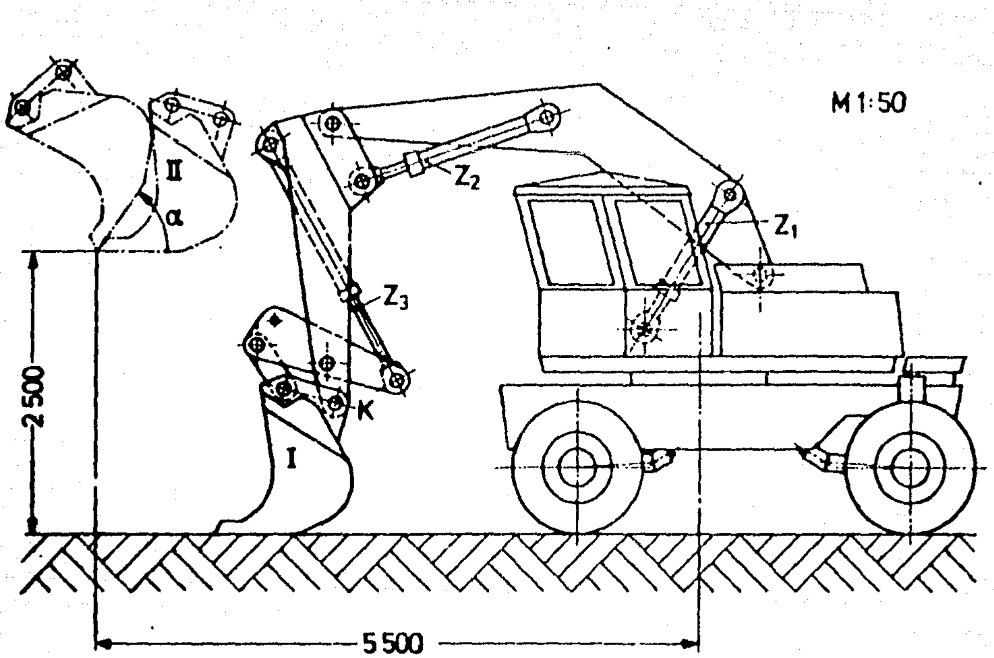


Fig 8.5 Dibujo Mecanico

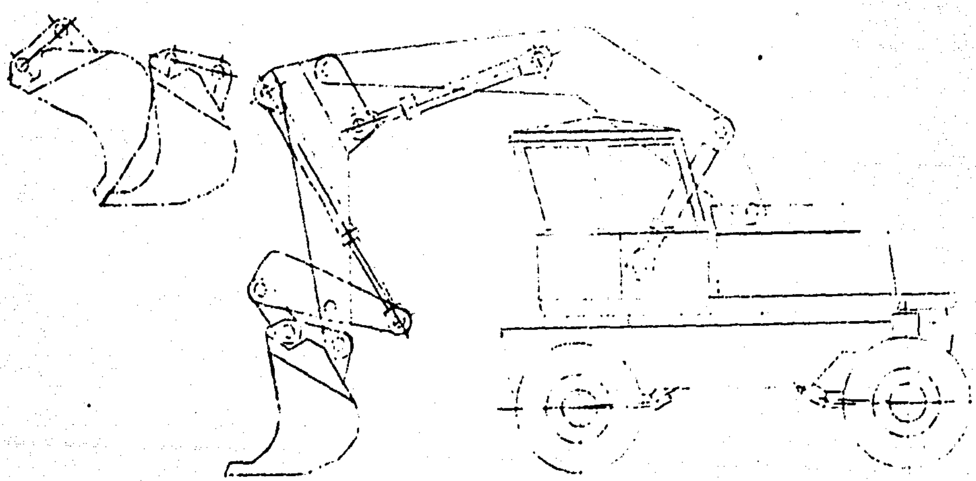


Fig 8.6 Reproducción de la Fig 8.5 con líneas rectas, círculos, arcos circulares, y "splines" paramétricas naturales y un factor de escala igual a 1

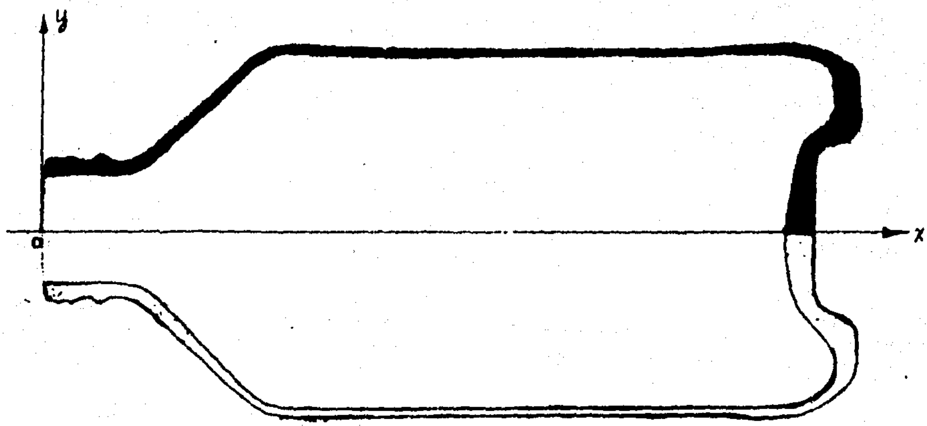


Fig B.7   Dibujo de una botella de cerveza "Quita-Pon"

Resultados del cálculo de las propiedades geométricas globales de la botella de cerveza mostrada en la Fig 8.7

(el volumen contenido de la botella):

```
VOLUMEN =      347.2674          (CM**3)
PRIMER MOMENTO QX =     [illegible]          (CM**4)
EL CENTROIDE RX =    8.[illegible]19          (CM) AL ORIGEN
EL MOMENTO DE INERCIA IX =      1570.[illegible]          (CM**5)
EL MOMENTO DE INERCIA IY =      30633.25          (CM**5)
MOMENTO DE INERCIA CENTROIDAL=    501[illegible].262          (CM**5)
************************************************
```

(el volumen de vidrio de la botella):

```
VOLUMEN =     70.81271          (CM**3)
PRIMER MOMENTO QX =    679.2192          (CM**4)
EL CENTROIDE RX =    9.591756          (CM) AL ORIGEN
EL MOMENTO DE INERCIA IX =     641.0128          (CM**5)
EL MOMENTO DE INERCIA IY =     8439.234          (CM**5)
MOMENTO DE INERCIA CENTROIDAL=    1924.339          (CM**5)
************************************************
```

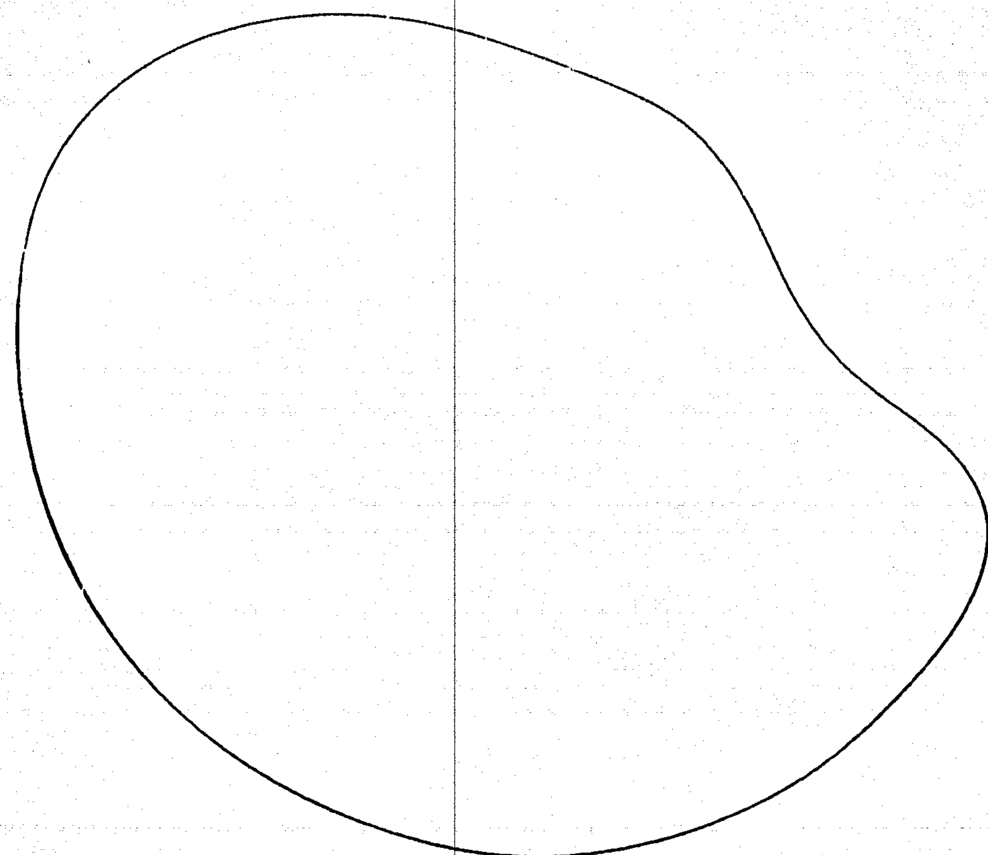


Fig 8.8 Dibujo de una leva, sintetizado con "spline" paramétrica periódica y su escala original

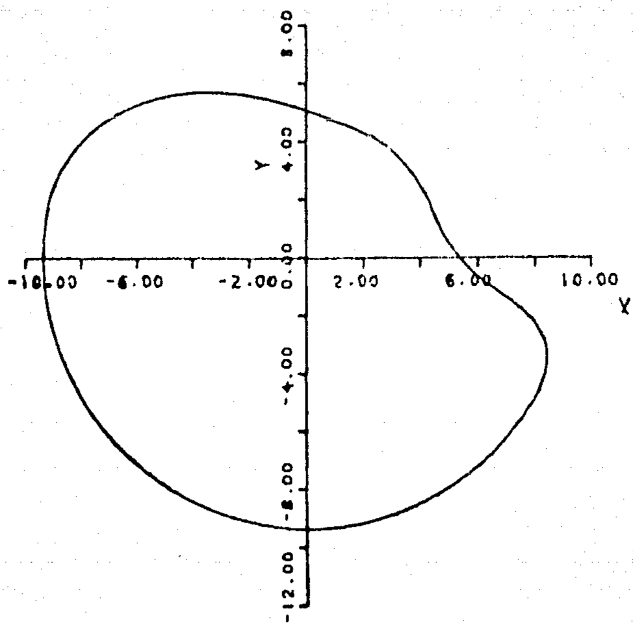
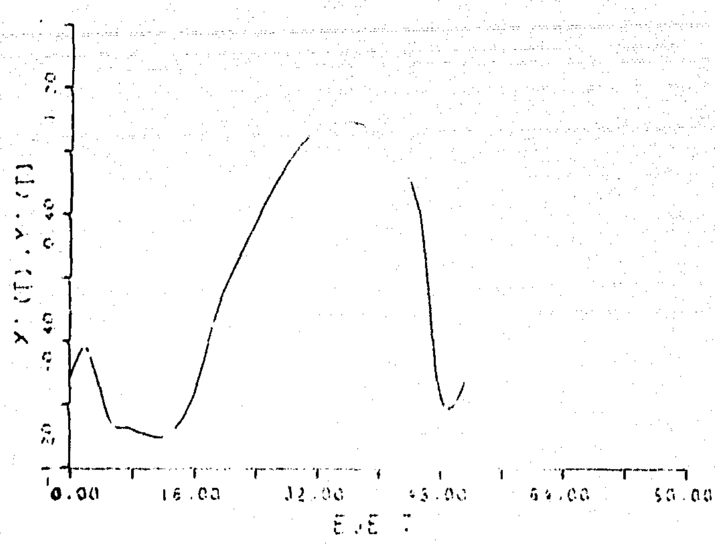


Fig 8.9 Dibujo de una leva, sintetizado con "spline" paramétrica periódica y un factor de escala iqual a 0.5



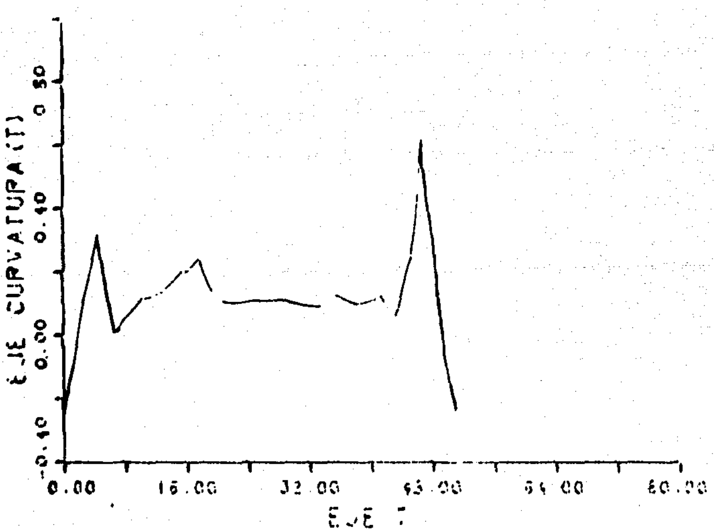
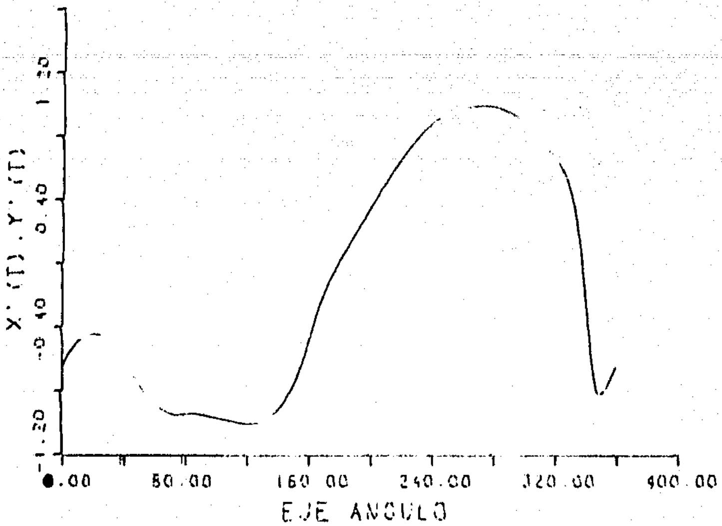


Fig 8.10 Representación gráfica de las propiedades geométricas locales de la leva en la Fig 8.9 (en función de longitud de la curva)



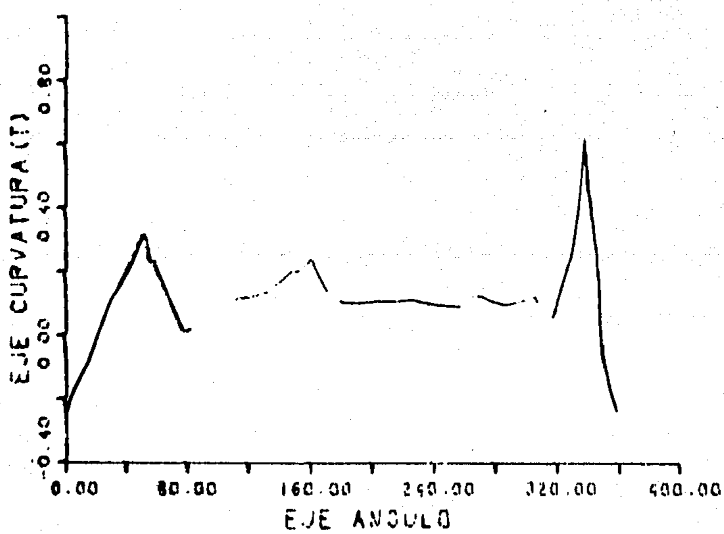


Fig 8.11 Representación gráfica de las propiedades geométricas locales de la leva en la Fig 8.9 (en función de ángulo)

Resultados del cálculo de las propiedades geométricas globales de la leva mostrada en la Fig.8.8:

```
AREA =          1[illegible] 3742  CM**2

CENTROIDE = (          [illegible]          [illegible] ) CM

MOMENTO DE INERCIA CON RESPECTO AL ORIGEN  CM**4

                              7613.1699       -404.2178
                              -404.2178       3861.3462


MOMENTO DE INERCIA CON RESPECTO AL CENTROIDE =

                              2905.9258       -828.2965
                              -828.2965       3606.4604

EJE.        ANGULO:        MOMENTO:

 1          -53.5 GRAD.         3314.4229
 2         -143.5 GRAD.         4168.0942
```

CAPITULO 9

CONCLUSIONES Y RECOMENDACIONES PARA TRABAJO FUTURO

En los capítulos anteriores se ha descrito un sistema de "software" utilizando tanto para digitalizar, sintetizar y reproducir curvas planas, como para calcular algunas de sus propiedades geometricas globales y locales mediante una computadora y sus periféricos (digitalizador y graficador); asimismo, se incluyen algunos resultados interesantes obtenidos usando este sistema. Una conclusión inmediata es que se ha logrado el objetivo que se mencionó en el capítulo 1.

El sistema es práctico y da resultados satisfactorios para la mayoría de las aplicaciones. Los resultados experimentales presentados en el capítulo anterior muestran la utilidad y confiabilidad de las funciones "spline"; es decir, las "splines" resultan ser una herramienta adecuada para la solución de problemas prácticos de interpolación en el diseño auxiliado por computadoras [13].

Como todos los trabajos hechos, éste tiene sus alcances y sus limitaciones. El diseño del sistema es general, es decir, teóricamente el sistema es capaz de digitalizar y reproducir figuras formadas por curvas de cualquier tamaño y complejidad, así como de construir bases de datos suficientemente grandes para representar fácilmente la información gráfica de las curvas planas. No obstante, existen en realidad severas limitaciones que restringen la utilidad del sistema. Primero, el error humano introducido en el proceso de digitalización y las imprecisiones del equipo deterioran la fidelidad de la información gráfica digitalizada, así como la de la reproducción de las curvas planas. Segundo, debido a que el proceso de digitalización es semiautomático, éste se vuelve complicado y tedioso, cuando el tamaño y la complejidad de la figura plana aumenta. Tercero, hay limitacion de espacio de la memoria principal de la computadora, es decir, el número de puntos digitalizados permisible es limitado(764 puntos). Cuarto, los cálculos de las propiedades geométricas pueden ser imprecisos debido a la manera inadecuada de la digitalización (por ejemplo, insuficiencia de número de puntos de apoyo o posiciones inadecuadas de éstos). Finalmente, hay que enfatizar que este sistema fue diseñado únicamente para procesar figuras planas formadas por curvas planas. Conceptos más refinados tales como perspectiva, contraste, etc., están fuera de consideración.

Este trabajo es original; pero todavía primitivo. A continuación se mencionan algunas recomendaciones para trabajo futuro:

1) Se ha observado que el ruido introducido en la digitalización resulta fuertemente amplificado al calcular derivadas, esto es, al calcular las propiedades geométricas locales. Se recomienda, en este caso, filtrar la señal antes de procesarla. Por ejemplo, en [29] se propuso un filtro de programación dinámica, para estimar las derivadas primera y segunda de los datos empíricos con ruido aleatorio aditivo. Sin embargo, el "software" para este cálculo resulta útil cuando se aplica a curvas obtenidas mediante algún proceso de síntesis, sin intervención del digitalizador. Efectivamente, se ha probado con éxito en el análisis de contornos de levas [26].

2) Explotar el mayor uso del digitalizador, es decir, diseñar "software" que maneje todos los modos de operación de este equipo, o bien, implantar programas particulares para algunas aplicaciones especiales.

3) Incluir más rutinas de usos especiales del "software" del graficador [18].

4) Aumentar la capacidad de almacenamiento de la memoria principal de la computadora para aumentar el poder del sistema de "software".

5) Investigar algoritmos más eficientes para la síntesis de curvas y para cálculos de las propiedades geométricas tanto globales como locales; por ejemplo, utilizar "splines" de grado mayor.

RECONOCIMIENTOS

Los autores desean expresar su sincero agradecimiento al doctor Jorge Angeles Alvarez y a todo el personal del Laboratorio de Cálculo Automatizado para el Diseño, de la DEFFI, por su apoyo y cooperación continuos durante el desarrollo de este trabajo. Los autores también desean agradecer a la señora Ruth Ramírez Bermúdez por la excelente mecanografía del Apéndice A de este trabajo.

REFERENCIAS

1 Faus, I.D. y Pratt, M.J., Computational Geometry for Design and Manufacture, Horwood, Chichester, England, 1979.

2 Pavlidis, T., Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, Md, 1982.

3 Henrici, P., Elements of Numerical Analysis, John Wiley & Sons, Inc., 1964. pp 183-213.

4 Issacson, E. y Keller, H.B., Analysis of Numerical Methods, John Wiley & Sons, Inc., 1969. pp 187-189, 229-237.

5 Froberg, C.-E., Introduction to Numerical Analysis, Addison-Wesley, Inc., 1965. pp 142-148.

6 Dahlquist, G. y Björck, Å., Numerical Methods, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1974. pp 81-136, 277-288, 131-134.

7 Späth, H. Spline Algorithmen zur Konstruktion Glatter Kurven und Flächen, 2nd edición, R. Olderburg, Munich, 1978.

8 Böhmer, K., Spline-Funktionen, B.G.Teuhner, Stuttgart, 1974.

9 Schultz, M.H., Spline Analysis, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.

10 Forsythe, G., Malcolm, M.A. y Moler, C.B., Computer Methods for Mathematical Computations, Prentice-Hall,Inc., Englewood Cliffs, N.J., 1979. pp 63-79.

11 De Boor, C., A Practical Guide to Splines, Springer-verlag, Nueva York, 1978.

12 Angeles, J., "Síntesis de curvas planas cerradas usando funciones spline paramétricas y periódicas", Revista de la Academia Nacional de Ingeniería, A.C., Vol.II No.1, 1983.

13 Björkenstam, U.C. y Kjellander, J.A.P., "Cubic curve fitting using variable segment stiffness for computer-aided design", Computers in Mechanical Engineering, Noviembre, 1983.

14 PDP-11/40,-11/35 System Manual, Digital Equipment Corporation, Maynard, Massachusetts, 1975.

15 RT-11 System reference Manual, Digital Equipment Corporation, Maynard, Massachusetts, 1976.

16 DECgraphic-11, FORTRAN Programming Manual, Digital equipment Corporation, Maynard, Massachusetts, 1978.

17 Cybergraph System Instruction Manual, Talos System, Inc.,

Arizona, S.L., 1977.
18 CalComp User's Manual. California Computer Products, Inc., 1969.
19 RT-11, RSTS/E FORTRAN IV User's Guide, Digital Equipment Corporation, Maynard,Massachusetts, 1980.
20 Angeles, J., "Cálculo de cantidades físicas globales asociadas a volúmenes acotados por superficies cerradas mediante integración en la frontera", Revista Ingeniería, No.1, Cd. de México, 1983.
21 Lee, Y.T. y Requicha, A.A.G., "Algorithms for computing the volume and other integral properties of solids", Communication of the ACM, Vol.25 No.9, Septiembre, 1982.
22 Brand, L., Advanced Calculus, John Wiley & Sons, Inc., Nueva York, 1955. pp 387-389.
23 Meriam, J.L., Statics SI-Version, 2nd Edición, John Wiley & Sons, Inc., 1975. pp 322.
24 Angeles, J., "Synthesis of plane curves with prescribed local geometric properties using periodic splines", Computer-Aided Design, Vol.15 No.3, Mayo, 1983.
25 Angeles, J., "Diseño automatizado de mecanismo de leva de disco con seguidor traslacional de carga plana", Memorias del IX Congreso de la Academia Nacional de Ingeniería, A. C., Septiembre de 1983.
26 Angeles, J. y López, C.C., "Optimal synthesis of translating-follower cam mechanisms with prescribed functional constraints", aceptado para su presentación en: International Symposium on Design and Synthesis, a celebrarse el 12 de Julio de 1984, en Tokio(Japón).
27 Angeles, J. y López, C.C., "Optimal synthesis of oscillating roller-follower cam mechanisms with prescribed functional constraints", aceptado para su presentación en: 1984 ASME International Computers in Engineering Conference and Exhibit, a celebrarse del 12 al 16 de agosto de 1984, en Las Vegas, Nevada(EUA).
28 Alonso, A., Navarrete, M. y Chicurel, R.,"Empujador de envases", Informe final, Primera etapa, Instituto de Ingeniería, UNAM, febrero, 1980. pp 4-13.
29 Trujillo, D.M. y Busby, H.R., "Investigation of a technique for the differentiation of empirical data", Transactions of

the ASME, Vol.105, Septiembre, 1983.

APENDICE A

DESARROLLO DETALLADO DE LAS FORMULAS PARA EL CALCULO DE LAS PROPIEDADES GEOMETRICAS GLOBALES DE CURVAS CERRADAS Y DE SOLIDOS DE REVOLUCION

A.1 Fómulas para calcular las propiedades geométricas globales de curvas cerradas.

Sea C una curva plana cerrada, descrita por la siguiente función "spline" paramétrica periódica:

$$\begin{aligned} x_k(t) &= A_{xk}(t-t_k)^3+B_{xk}(t-t_k)^2+C_{xk}(t-t_k)+D_{xk} \\ y_k(t) &= A_{yk}(t-t_k)^3+B_{yk}(t-t_k)^2+C_{yk}(t-t_k)+D_{yk} \end{aligned} \qquad (A.1.1)$$

$$t_k \le t \le t_{k+1}, \quad \Delta t_k = t_{k+1} - t_k, \quad K = 0, 1, 2, \ldots, n-1$$

Cuando no hay confusión, se omite el subíndice k para tener mayor simplicidad.

1) El área encerrada por C se calcula por las siguientes fórmulas:

$$A = \frac{1}{2}\int_{t_0}^{t_n} (x\dot{y}-\dot{x}y)\, dt = \sum_{k=0}^{n-1} \frac{1}{2}\int_{t_k}^{t_{k+1}} (x_k\dot{y}_k-\dot{x}_k y_k)\, dt \qquad (A.1.2)$$

Defínase $$A_k = \frac{1}{2}\int_{t_k}^{t_{k+1}} [x_k(t)\dot{y}_k(t) - \dot{x}_k(t)y_k(t)]\, dt \qquad (A.1.3)$$

Sustituyendo (A.1.1) en (A.1.3) y omitiendo el subíndice k, resulta una fórmula algebraica para calcular A_k, $k = 0,1,2,\ldots,n-1$:

$$\begin{aligned} A_k = {} & \frac{1}{10}(A_yB_x-A_xB_y)\,\Delta t^5 + \frac{1}{4}(A_xC_x-A_xC_y)\,\Delta t^4 + \\ & + \frac{1}{6}(3A_yD_x-3A_xD_y + B_yC_x-B_xC_y)\,\Delta t^3 + \\ & + \frac{1}{2}(B_yD_x-B_xD_y)\,\Delta t^2 + \frac{1}{2}(C_xD_x-C_xD_y)\,\Delta t \end{aligned} \qquad (A.1.4)$$

Finalmente,

$$A = \sum_{k=0}^{n-1} A_k \tag{A.1.5}$$

2) El primer momento se calcula con las siguientes fórmulas:

$$Q = \frac{1}{2}\int_{t_0}^{t_{n-1}} (x^2+y^2)\,[\dot{y},\ -\dot{x}]^T\,dt = \sum_{k=0}^{n-1} \frac{1}{2}\int_{t_k}^{t_{k+1}} (x_k^2+y_k^2)\,[\dot{y}_k,\ -\dot{x}_k]^T\,dt \tag{A.1.6}$$

Defínase $$Q_k^y = \frac{1}{2}\int_{t_k}^{t_{k+1}} \left[x_k^2(t)+y_k^2(t)\right]\,\dot{x}_k(t)\,dt \tag{A.1.7}$$

$$Q_k^x = \frac{1}{2}\int_{t_k}^{t_{k+1}} \left[x_k^2(t)+y_k^2(t)\right]\,\dot{y}_k(t)\,dt \tag{A.1.8}$$

Sustituyendo (A.1.1) en (A.1.7) y (A.1.8), se obtienen las siguientes fórmulas para evaluar Q_k^y y Q_k^x, k=0,1,2,...,n-1, respectivamente. (Nótese que el subíndice **k** se omite):

$$\begin{aligned} Q_k^y = {} & \frac{1}{3} A_xT_1\Delta t^9 + \frac{1}{4}(B_xT_1+3A_xT_2)\,\Delta t^8 + \\ & + \frac{1}{7}(C_xT_1 + 4B_xT_2 + 3A_xT_3)\,\Delta t^7 + \\ & + \frac{1}{3}(C_xT_2 + B_xT_3 + 3A_xT_4)\,\Delta t^6 + \\ & + \frac{1}{5}(C_xT_3 + 4B_xT_4 + 3A_xT_5)\,\Delta t^5 + \\ & + \frac{1}{2}(C_xT_4 + B_xT_5 + 3A_xT_6)\,\Delta t^4 + \\ & + \frac{1}{3}(C_xT_5 + 4B_xT_6 + 3A_xT_7)\,\Delta t^3 + \\ & + (C_xT_6 + B_xT_7)\,\Delta t^2 + C_xT_7\,\Delta t \end{aligned} \tag{A.1.9}$$

$$Q_k^x = \frac{1}{3} A_y T_1 \Delta t^9 + \frac{1}{4} (B_y T_1 + 3A_y T_2) \Delta t^8 +$$

$$+ \frac{1}{7} (C_y T_1 + 4B_y T_2 + 3A_y T_3) \Delta t^7 +$$

$$+ \frac{1}{3} (C_y T_2 + B_y T_3 + 3A_y T_4) \Delta t^6 +$$

$$+ \frac{1}{5} (C_y T_3 + 4B_y T_4 + 3A_y T_5) \Delta t^5 + \qquad (A.1.10)$$

$$+ \frac{1}{2} (C_y T_4 + B_y T_5 + 3A_y T_6) \Delta t^4 +$$

$$+ \frac{1}{3} (C_y T_5 + 4B_y T_6 + 3A_7 T_7) \Delta t^3 +$$

$$+ (C_y T_6 + B_y T_7) \Delta t^2 + C_y T_7 \Delta t$$

donde $T_1 = A_x^2 + A_y^2$, $T_2 = A_x B_x + A_y B_y$, $T_3 = B_x^2 + B_y^2 + 2(A_x C_x + A_y C_y)$,

$T_4 = A_x D_x + A_y D_y + B_x C_x + B_y C_y$, $T_5 = C_x^2 + C_y^2 + 2(B_x D_x + B_y D_y)$

$T_6 = C_x D_x + C_x D_x$, $T_7 = D_x^2 + D_y^2$

son términos comunes y simétricos (en cuanto a x,y), los cuales se deben calcular primeramente.

Se puede notar que (A.1.9) y (A.1.10) son semejantes. De hecho se puede obtener una de otra haciendo un intercambio de subíndices:

$$x \longleftrightarrow y$$

Finalmente,

$$Q = \left[\sum_{k=0}^{n-1} Q_k^x, \; - \sum_{k=0}^{n-1} Q_k^y \right]^T \qquad (A.1.11)$$

3) Los momentos de inercia se calculan mediante las siguientes fórmulas:

$$I = \frac{1}{8}\int_{t_0}^{t_n} (\dot{x}+\dot{y}) \begin{bmatrix} -x\dot{y}-3\dot{x}y & 4x\dot{x} \\ -4y\dot{y} & 3x\dot{y}+\dot{x}y \end{bmatrix} dt \qquad (A.1.12)$$

Definase $I_k^1 = \frac{1}{8}\int_{t_k}^{t_{k+1}} [x_k^2(t)+y_k^2(t)]\; x_k(t)\; \dot{x}_k(t)\; dt$ (A.1.13)

$$I_k^2 = \frac{1}{8}\int_{t_k}^{t_{k+1}} [x_k^2(t) + y_k^2(t)]\; x_k(t)\; \dot{y}_k(t)\; dt \qquad (A.1.14)$$

$$I_k^3 = \frac{1}{8}\int_{t_k}^{t_{k+1}} [x_k^2(t) + y_k^2(t)]\; \dot{x}_k(t)\; y_k(t)\; dt \qquad (A.1.15)$$

$$I_k^4 = \frac{1}{8}\int_{t_k}^{t_{k+1}} [x_k^2(t) + y_k^2(t)]\; \dot{y}_k(t)\; y_k(t)\; dt \qquad (A.1.16)$$

sustituyendo (A.1.1) en (A.1.13) y (A.1.15) se obtienen:

$$I_k^1 = \frac{1}{4} A_x^2 T_1 \Delta t^{12} + \frac{1}{11}(5A_xB_xT_1+6A_x^2T_2)\,\Delta t^{11} +$$

$$+ \frac{1}{10}(2P_xT_1+10A_xB_xT_2+3A_x^2T_3)\,\Delta t^{10} +$$

$$+ \frac{1}{9}(3B_xC_xT_1+4P_xT_2+5A_xB_xT_3+6A_x^2T_4)\,\Delta t^{9} +$$

$$+ \frac{1}{8}(Q_xT_1+6B_xC_xT_2+2P_xT_3+10A_xB_xT_4+3A_x^2T_5)\,\Delta t^{8} +$$

$$+ \frac{1}{7}(C_xT_1+2Q_xT_2+3B_xC_xT_3+4P_xT_4+5A_xB_xT_5+6A_x^2T_6)\,\Delta t^{7} +$$

$$- \frac{1}{6}(2C_xT_2+Q_xT_3+6B_xC_xT_4+2P_xT_5+10A_xB_xT_6+3A_x^2T_7)\,\Delta t^{6}$$

$$+ \frac{1}{5} (C_x T_3 + 2Q_x T_4 + 3B_x C_x T_5 + 4P_x T_6 + 5A_x B_x T_7) \Delta t^5 + \qquad (A.1.17)$$

$$+ \frac{1}{4} (2C_x T_4 + Q_x T_5 + 6B_x C_x T_6 + 2P_x T_7) \Delta t^4 +$$

$$+ \frac{1}{3} (C_x T_5 + 2Q_x T_6 + 3B_x C_x T_7) \Delta t^3 +$$

$$+ (2C_x T_6 + Q_x T_4) \Delta t^2 + C_x D_x T_7 \Delta t$$

$$I_k^3 = \frac{1}{4} A_x A_y T_1 \Delta t^{12} + \frac{1}{11} (U_x T_1 + 6A_x A_y T_2) \Delta t^{11} +$$

$$+ \frac{1}{10} (V_x T_1 + 2U_x T_2 + 3A_x A_y T_3) \Delta t^{10} +$$

$$+ \frac{1}{9} (W_x T_1 + 2V_x T_2 + U_x T_3 + 6A_x A_y T_4) \Delta t^9 +$$

$$+ \frac{1}{8} (Z_x T_1 + 2W_x T_2 + V_x T_3 + 2U_x T_4 + 3A_x A_y T_5) \Delta t^8 +$$

$$+ \frac{1}{7} (C_x D_y T_1 + 2Z_x T_2 + W_x T_3 + 2V_x T_4 + U_x T_5 + 6A_x A_y T_6) \Delta t^7 +$$

$$+ \frac{1}{6} (2C_x D_y T_2 + Z_x T_3 + 2W_x T_4 + V_x T_5 + 2U_x T_6 + 3A_x A_y T_7) \Delta t^6 + \qquad (A.1.18)$$

$$+ \frac{1}{5} (C_x D_y T_3 + 2Z_x T_4 + W_x T_5 + 2V_x T_6 + U_x T_7) \Delta t^5 +$$

$$+ \frac{1}{4} (2C_x D_y T_4 + Z_x T_5 + 2W_x T_6 + V_x T_7) \Delta t^4 +$$

$$+ \frac{1}{3} (C_x D_y T_5 + 2Z_x T_6 + W_x T_7) \Delta t^3 +$$

$$+ \frac{1}{2} (2C_x D_y T_6 + Z_x T_7) \Delta t^2 + C_x D_y T_7 \Delta t$$

donde

$$P_x = B_x^2 + 2A_x C_x, \quad Q_x = C_x^2 + 2B_x D_x$$

$$U_x = 2A_y B_x + 3A_x B_y, \quad V_x = A_y C_x + 2B_x B_y + 3A_x C_y,$$

$$W_x = B_yC_x + 2B_xC_y, \quad Z_x = C_xC_y + 2B_xD_y,$$

siendo $T_1, \ldots, T_7$, las mismas cantidades definidas anteriormente en (A.1.9) y (A.1.10).

Las fórmulas para I_k^2 y I_k^4 se obtienen a partir de (A.1.17) y (A.1.18), respectivamente, haciendo un intercambio de subíndices, $x \leftrightarrow y$.

Finalmente,

$$I = \begin{bmatrix} -\sum_{k=0}^{n-1} (I_k^2+3I_k^3) & 4\sum_{k=0}^{n-1} I_k^1 \\ -4\sum_{k=0}^{n-1} I_k^4 & \sum_{k=0}^{n-1} (3I_k^2+I_k^3) \end{bmatrix} \qquad \text{(A.1.19)}$$

A.2 Fómulas para el cálculo de las propiedades geométricas globales asociadas a un sólido de revolución.

Se define:

C : Curva generatriz del sólido de revolución.

$$\Delta t_k = (t_{k+1} - t_k)$$

$$x_k(t) = A_{xk}(t-t_k)^3 + B_{xk}(t-t_k)^2 + C_{xk}(t-t_k) + D_{xk}$$

$$x_k(t) = A_{yk}(t-t_k)^3 + B_{yk}(t-t_k)^2 + C_{yk}(t-t_k) + D_{yk}, \quad \text{para } K=0,1,\ldots,n-1$$

Para simplificar las fórmulas desarrolladas se usa las variables intermedias, y en caso de que no haya confusión, el subíndice K se omite.

1) Volumen

$$V = 2\pi/3 \int_{t_o}^{t_n} y(x\dot{y} - \dot{x}y)\, dt = \sum_{k=0}^{n-1} V_k$$

donde $V_k = 2\pi/3 \int_{t_k}^{t_{k+1}} y_k(x_k\dot{y}_k-\dot{x}_k y_k)\, dt, \quad k=0,1,\ldots,n-1$

$$(x\dot{y}-\dot{x}y) = (B_xA_y-A_xB_y)\,\Delta t^4 + 2(C_xA_y-A_xC_y)\,\Delta t^3$$

$$+ (C_xB_y-B_xC_y+3D_xA_y-3A_xD_y)\,\Delta t^2 + 2(D_xB_y-B_xD_y)\,\Delta t$$

$$+ (D_xC_y-C_xD_y) = \sum_{i=0}^{4} Z_i\,\Delta t^i \qquad \text{(A.2.1)}$$

$$\therefore\ V_k = 2\pi/3 \sum_{i=1}^{8} \frac{1}{i} W_i\,\Delta t_k^i$$

donde
$$W_1 = D_yZ_0, \qquad W_5 = A_yZ_1+B_yZ_2+C_yZ_3+D_yZ_4$$
$$W_2 = C_yZ_0+D_yZ_1 \qquad W_6 = A_yZ_2+B_yZ_3+C_yZ_4$$
$$W_3 = B_yZ_0+C_yZ_1+D_yZ_2 \qquad W_7 = A_yZ_3+B_yZ_4$$
$$W_4 = A_yZ_0+B_yZ_1+C_yZ_2+D_yZ_3 \qquad W_8 = A_yZ_4$$

2) Primer momento.

$$Q_x = -\pi \int_{t_o}^{t_n} y\dot{y}(x^2+y^2)\,dt = \sum_{k=0}^{n-1} Q_{xk}$$

donde: $$Q_{xk} = -\pi \int_{t_k}^{t_{k+1}} y_k\dot{y}_k(x_k^2+y_k^2)\,dt \qquad \text{(A.2.2)}$$

$$(x^2+y^2) = \sum_{i=0}^{6} f_i\,\Delta t^i \qquad \text{(A.2.3)}$$

Las f_i están dadas por:

$$f_0 = D_x^2+D_y^2$$

$$f_1 = 2(C_xC_x+C_yD_y)$$

$$f_2 = 2B_xD_x+C_x^2+2B_yD_y+C_y^2$$

$$f_3 = 2(A_xD_x+B_xC_x+A_yD_y+B_yC_y)$$

$$f_4 = 2A_xC_x+B_x^2+2A_yC_y+B_y^2$$

$$f_5 = 2(A_xB_x+A_yB_y)$$

$$f_6 = A_x^2+A_y^2$$

(A.2.4)

$$y\dot{y} = \sum_{i=0}^{5} g_i\,\Delta t^i$$

donde las g_i están dadas por:

$$g_0 = C_yD_y$$

$$g_1 = 2B_yD_y+C_y^2$$

$$g_2 = 3B_yC_y+3A_yD_y$$

$$g_3 = 4A_yC_y+2B_y^2$$

$$g_4 = 5A_yB_y$$

$$g_5 = 3A_y^2$$

Sustituyendo (A.2.3) y (A.2.4) en la ecuación (A.2.2), se obtiene:

$$Q_{xk} = -\pi \sum_{i=1}^{12} \frac{1}{i} T_i\,\Delta t_k^i,$$ donde T_i están dadas por:

$$T_1 = g_0f_0$$

$$T_2 = g_1f_0+g_0f_1$$

$$T_3 = g_2f_0+g_1f_1+g_0f_2$$

$$T_4 = g_3f_0+g_2f_1+g_1f_2+g_0f_3$$

$$T_5 = g_4f_0+g_3f_1+g_2f_2+g_1f_3+g_0f_4$$

$$T_6 = g_5f_0+g_4f_1+g_3f_2+g_2f_3+g_1f_4+g_0f_5$$

$$T_7 = g_5f_1+g_4f_2+g_3f_3+g_2f_4+g_1f_5+g_0f_6$$

$$T_8 = g_5f_2+g_4f_3+g_3f_4+g_2f_5+g_1f_6$$

$$T_9 = g_5f_3+g_4f_4+g_3f_5+g_1f_6$$

$$T_{10} = g_5f_4+g_4f_5+g_3f_6$$

$$T_{11} = g_5f_5+g_4f_6$$

$$T_{12} = g_5f_6$$

3) Centroide

$$r_x = Q_x/V$$

4) Momentos de Inercia

$$I^0 = \left[I_x^0,\ I_y^0\right]'$$

$$I_x^0 = \pi/5 \int_{t_o}^{t_n} y(x^2+y^2)(2x\dot{y}+3\dot{x}y)\ dt = \sum_{x=0}^{n-1} I_{xk}^0$$

donde $$I_{xk}^0 = \pi/5 \int_{t_k}^{t_{k+1}} y_k(x_k^2+y_k^2)(2x_k\dot{y}_k+3\dot{x}_k y_k)\ dt \qquad (A.2.5)$$

y $$I_y^0 = -\ \pi/10 \int_{t_o}^{t_n} y(x^2+y^2)(6x\dot{y}-\dot{x}y)\ dt = \sum_{k=0}^{n-1} I_{yk}^0$$

donde $$I_{yk}^0 = -\ \pi/10 \int_{t_k}^{t_{k+1}} y_k(x_k^2+y_k^2)(6x_k\dot{y}_k-\dot{x}_k y_k)\ dt \qquad (A.2.6)$$

utilizando la ecuación (A.2.3), se tiene:

$$y(x^2+y^2) = \sum_{i=0}^{9} e_i\ \Delta t^i \qquad (A.2.7)$$

donde las e_i están dadas por:

$e_0 = D_y f_0$

$e_1 = C_y f_0 + D_y f_1$

$e_2 = B_y f_0 + C_y f_1 + D_y f_2$

$e_3 = A_y f_0 + B_y f_1 + C_y f_2 + D_y f_3$

$e_4 = A_y f_1 + B_y f_2 + C_y f_3 + D_y f_4$

$e_5 = A_y f_2 + B_y f_3 + C_y f_4 + D_y f_5$

$e_6 = A_y f_3 + B_y f_4 + C_y f_5 + D_y f_6$

$e_7 = A_y f_4 + B_y f_5 + C_y f_6$

$e_8 = A_y f_5 + B_y f_6$

$e_9 = A_y f_6$

$$\therefore \quad (2x\dot{y} + 3\dot{x}y) = \sum_{i=0}^{5} R_i \, \Delta t^i \qquad \text{(A.2.8)}$$

donde R_i están dadas por:

$R_0 = 2D_x C_y + 3C_x D_y$

$R_1 = 5C_x C_y + 4D_x B_y + 6D_y B_x$

$R_2 = 8C_y B_x + 7C_x B_y + 6A_y D_x + 9A_x D_y$

$R_3 = 9C_x A_y + 10B_x B_y + 11C_y A_x$

$R_4 = 12A_y B_x + 13A_x B_y$

$R_5 = 15A_x A_y$

Sustituyendo (A.2.7) y (A.2.8) en la ecuación (A.2.5), se obtiene:

$I^0_{xk} = \pi/5 \sum_{i=1}^{15} \frac{1}{i} P_i \, \Delta t^i_k$, donde las P_i están dadas por:

$P_1 = e_0 R_0$

$P_2 = e_0 R_1 + e_1 R_0$

$P_3 = e_0 R_2 + e_1 R_1 + e_2 R_0$

$P_4 = e_0 R_3 + e_1 R_2 + e_2 R_1 + e_3 R_0$

$$
\begin{aligned}
P_5 &= e_0R_4+e_1R_3+e_2R_2+e_3R_1+e_4R_0 \\
P_6 &= e_0R_5+e_1R_2+e_2R_3+e_3R_2+e_4R_1+e_5R_0 \\
P_7 &= e_1R_5+e_2R_4+e_3R_3+e_4R_2+e_5R_1+e_6R_0 \\
P_8 &= e_2R_5+e_3R_4+e_4R_3+e_5R_2+e_6R_1+e_7R_0 \\
P_9 &= e_3R_5+e_4R_4+e_5R_3+e_6R_2+e_7R_1+e_8R_0 \\
P_{10} &= e_4R_5+e_5R_4+e_6R_3+e_7R_2+e_8R_1+e_9R_0 \\
P_{11} &= e_5R_5+e_6R_4+e_7R_3+e_8R_2+e_9R_1 \\
P_{12} &= e_6R_5+e_7R_4+e_8R_3+e_9R_2 \\
P_{13} &= e_7R_5+e_8R_4+e_9R_3 \\
P_{14} &= e_8R_5+e_9R_4 \\
P_{15} &= e_9R_5
\end{aligned}
\qquad (A.2.9)
$$

Para calcular I_{yk}^{0}, se tiene:

$$(6x\dot{y}-\dot{x}y) = \sum_{i=0}^{5} r_i \, \Delta t^i \qquad (A.2.10)$$

donde las r_i están dadas por:

$$r_0 = 6D_xC_y-C_xD_y \qquad r_3 = 3A_xC_y+10B_xB_y+17C_xA_y$$

$$r_1 = 5C_xC_y+12D_xB_y-2D_yB_x \qquad r_4 = 9A_xB_y+16A_yB_x$$

$$r_2 = 4B_xC_y+11C_xB_y+18D_xA_y-3D_yA_x, \qquad r_5 = 15A_xA_y$$

Las otras operaciones son similares a las de la ec (A.2.9), solamente que en lugar de R_i cambian a r_i.

APENDICE B

B.1 DIGTLZ:

```
C     ------------------------------------------------------------------
C       Este programa es disenado e implementado para digitalizar
C     la imformacion grafica de curvas plana  utilizando un digitalizador
C     el cual esta conectado a la computadora PDP-11/40 en el
C     laboratorio de Calculo Automatizado para el Diseno de la
C     DEPFI,UNAM.
C       La funcion del programa es establecer la comunicacion del
C     digitalizador a la computadora, recibir e interpretar
C     adecuadamente los datos enviados en forma automatica e
C     interactiva de acuerdo a un conjunto de instrucciones
C     preestablecido y almacenar la informacion grafica digitalizada
C     en un archivo de datos digitalizados que puede ser utilizada
C     posteriomente.
C     ------------------------------------------------------------------
      REAL S(80,8),P(700,2)
      LOGICAL COND
      INTEGER TIMBRE,MODO,BOTON,LDEV,D,X,Y,X1,Y1
C
      TIMBRE=7
      WRITE(7,1) TIMBRE
1     FORMAT(1X,A1)
      WRITE(7,2)
2     FORMAT(3X,'BIENVENIDO!',//,3X,'Verifique el modo de
     *operacion del digitalizador, y habilite el modo POINT.'/)
      WRITE(7,5)
5     FORMAT(3X,'?Quiere ver las instrucciones para el manejo del
     * digitalizador? (S,N)',/)
      READ(5,6) D
6     FORMAT(A1)
      IF(D.EQ.'N') GO TO 8
      WRITE(7,12)
12    FORMAT(//,10X,'Instrucciones para el manejo del digitalizador',//,
     * 10X,'BOTON',10X,'FUNCION',//,
     * 13X,'1',5X,'ORIGEN DEL SISTEMA COORDENADO',/,
     * 13X,'2',5X,'FACTOR DE ESCALA',/,
     * 13X,'3',5X,'ANGULO DE ROTACION',/,
     * 13X,'4',5X,'LINEA RECTA',/,
     * 13X,'5',5X,'ARCO CIRCULAR O CIRCULO COMPLETO',/,
     * 13X,'6',5X,'SPLINE NATURAL O PERIODICA',/,
     * 13X,'7',5X,'SPLINE PARAMETRICA NATURAL O PERIODICA',/,
     * 13X,'8',5X,'TIPO DE CURVA INTERMITENTE',/,
     * 13X,'9',5X,'AJUSTE DE PUNTOS DIGITALIZADOS',/,
     * 13X,'0',5X,'ELIMINACION DE PUNTOS DIGITALIZADOS',/,
     * 13X,'*',5X,'TOMA DE MUESTRAS DE PUNTOS',/,
     * 13X,'#',5X,'ALMACENAMIENTO Y TERMINACION',/)
8     LDEV=0
10    LDEV=LDEV+1
C
13    NS=1
      NP=0
      S(1,1)=1.
      S(1,2)=0.
      S(1,3)=0.
      S(1,4)=0.
      S(1,5)=1.
      S(1,6)=0.
      S(1,7)=1.
      S(1,8)=0.
      COND=.TRUE.
```

```
14    WRITE(7,15)
15    FORMAT(3X,'Defina los parametros y digitalice la grafica.
     *Adelante!',//)
20    CALL COM(MODO,BOTON,X,Y)
      IF(MODO-1) 25,30,25
25    WRITE(7,1) (TIMBRE,I=1,3)
      WRITE(7,27)
27    FORMAT(3X,'El modo de operacion del digitalizador debe ser
     * POINT.',/)
      PAUSE '   Use  RETURN  para continuar.'
      GO TO 14
30    IF(.NOT.COND) GO TO 40
      IF(NS.EQ.1.AND.NP.EQ.0) GO TO 36
      DO 35 I=3,8
35    S(NS+1,I)=S(NS,I)
      S(NS,2)=NP-S(NS,1)+1
      NS=NS+1
      S(NS,1)=NP+1
      IF(BOTON.EQ.1) S(NS-1,8)=1.
36    COND=.FALSE.
      K=0
      L1=0
      L2=0
      L3=0
      WRITE(7,38) NS
38    FORMAT(3X,'Se crea el segmento',I3,/)
40    GO TO(180,100,110,120,130,140,150,160,170,190,200,210) BOTON+1
C >>>El origen del sistema coordenado<<<
100   S(NS,3)=X
      S(NS,4)=Y
      WRITE(7,105) S(NS,3),S(NS,4)
105   FORMAT(3X,'El origen del sistema coordenado = (',2F6.0,' )',/)
      GO TO 20
C >>>Factor de escala<<<
110   WRITE(7,1) (TIMBRE,I=1,2)
      WRITE(7,115)
115   FORMAT(3X,'Escriba el factor de escala( numero positivo):',/)
      READ(5,*) S(NS,5)
      WRITE(7,117) S(NS,5)
117   FORMAT(3X,'El factor de escala = ',F8.3,/)
      GO TO 20
C >>>Angulo de rotacion<<<
120   IF(K-1) 121,125,125
121   ZY=Y-S(NS,4)
      ZX=X-S(NS,3)
      S(NS,6)=ATAN2(ZY,ZX)
      K=1
      GO TO 127
125   ZY=Y-Y1
      ZX=X-X1
      S(NS,6) = ATAN2(ZY,ZX)
127   X1=X
      Y1=Y
      WRITE(7,128) S(NS,6)*180./3.1415926
128   FORMAT(3X,'El angulo de rotacion =',F5.1,' grados.',/)
      GO TO 20
C >>>Linea recta (1)<<<
130   S(NS,7)=1.
      GO TO 171
```

```
C >>>Arco circular (2) o circulo completo (3)<<<
140     L1=L1+1
        IF(L1-2) 145,147,147
145     S(NS,7)=2.
        GO TO 171
147     S(NS,7)=3.
        L1=0
        GO TO 171
C >>>Spline natural (4) o periodica (5)<<<
150     L2=L2+1
        IF(L2-2) 155,157,157
155     S(NS,7)=4.
        GO TO 171
157     S(NS,7)=5.
        L2=0
        GO TO 171
C >>>Spline parametrica natural (6) o periodica (7)<<<
160     L3=L3+1
        IF(L3-2) 165,167,167
165     S(NS,7)=6.
        GO TO 171
167     S(NS,7)=7.
        L3=0
        GO TO 171
C >>>Curva de tipo solido (+) o intermitente (-)<<<
170     S(NS,7)=-S(NS,7)
171     WRITE(7,172)
172     FORMAT(3X,'El tipo de curva es ')
        I=ABS(S(NS,7))
        GO TO (1717,1720,1730,1740,1750,1760,1770) I
1717    WRITE(7,1711)
1711    FORMAT(23X,'Linea recta')
        GO TO 173
1720    WRITE(7,1721)
1721    FORMAT(23X,'Arco circular')
        GO TO 173
1730    WRITE(7,1731)
1731    FORMAT(23X,'Circulo completo')
        GO TO 173
1740    WRITE(7,1741)
1741    FORMAT(23X,'Spline natural')
        GO TO 173
1750    WRITE(7,1751)
1751    FORMAT(23X,'Spline periodica')
        GO TO 173
1760    WRITE(7,1761)
1761    FORMAT(23X,'Spline parametrica natural')
        GO TO 173
1770    WRITE(7,1771)
1771    FORMAT(23X,'Spline parametrica periodica')
173     IF(S(NS,7).LT.0.) WRITE(7,175)
175     FORMAT(23X,'Intermitente')
        GO TO 20
C >>>Eliminacion de puntos digitalizados<<<
180     IF(NP-1) 181,185,185
181     WRITE(7,1) (TIMBRE,I=1,2)
        WRITE(7,182)
182     FORMAT(3X,'No hay datos digitalizados.',/)
        GO TO 20
```

```
185     WRITE(7,184) NP,P(NP,1),P(NP,2)
184     FORMAT(3X,'Punto',I3,' = (',2F6.0,') es eliminado.',/)
        NP=NP-1
        IF(NP.GE.S(NS,1)) GO TO 202
        NS=NS-1
        COND=.TRUE.
        WRITE(7,187) NS+1
187     FORMAT(3X,'Segmento',I3,' es eliminado.',/)
        IF(NS.LE.0) GO TO 13
        GO TO 20
C >>>Ajuste de puntos digitalizados<<<
190     WRITE(7,1) (TIMBRE,I=1,2)
        WRITE(7,191) NP
191     FORMAT(3X,'Ajuste punto',I3,' por la terminal (x,y):',/)
        READ(5,*) P(NP,1),P(NP,2)
        WRITE(7,195) P(NP,1),P(NP,2)
195     FORMAT(3X,'Punto ajustado = (',2F6.0,').',/)
        GO TO 202
C >>>Toma de puntos de la grafica<<<
200     NP=NP+1
        P(NP,1)=X
        P(NP,2)=Y
        COND=.TRUE.
        K=0
        L1=0
        L2=0
        L3=0
        WRITE(7,201) NP,P(NP,1),P(NP,2)
201     FORMAT(3X,'Punto',I3,' = (',2F6.0,').',/)
202     CALL COM(MODO,BOTON,X,Y)
        IF(BOTON.EQ.0) GO TO 180
        IF(BOTON.GE.9) GO TO(190,200,210) BOTON-8
        GC TO 30
C >>>Almacenamiento de datos digitalizados en un archivo<<<
210     S(NS,2)=NP-S(NS,1)+1
        WRITE(7,1) (TIMBRE,I=1,2)
        WRITE(7,11)
11      FORMAT(3X,'Escriba el nombre de archivo de datos digitalizados:'
     *  ,/)
        CALL ASSIGN(LDEV,' ',-1)
        WRITE(7,211)
211     FORMAT(3X,'Escoja el tipo de archivo de datos digitalizados:',/,
     *  '    ?No-Formateadc o Formateado? (N,F).',/)
        READ(5,6) D
        IF(D.EQ.'F') GO TO 215
C >>>No-Formateado<<<
212     DO 213 I=1,NS
        WRITE(LDEV) (S(NS,K),K=2,8)
        N1=S(I,1)
        N2=N1+S(I,2)
        WRITE(LDEV) ((P(J,K),K=1,2),J=N1,N2-1)
213     CONTINUE
        GO TO 219
C >>>Formateado<<<
215     DO 219 I=1,NS
        WRITE(LDEV,217) (S(I,J),J=2,8)
217     FORMAT(7F9.3)
        N1=S(I,1)
        N2=N1+S(I,2)
        WRITE(LDEV,218) ((P(J,K),K=1,2),J=N1,N2-1)
218     FORMAT(2F7.1)
219     CONTINUE
```

```
      WRITE(7,1) (TIMBRE,I=1,2)
      WRITE(7,220)
220   FORMAT(3X,'?Desea un resumen de los datos disitalizados?  (S,N)',/
      READ(5,6) D
      IF(D.EQ.'N') GO TO 228
      WRITE(7,223) NS,NP
223   FORMAT(3X,'El numero de sesmentos =',I3,'   El numero de
     *puntos =',I3,/)
      WRITE(7,224)
224   FORMAT('  Sesmento  Indice  Londitud  Orisen (x,y)   Escala  Ansulo
     *  Curva  Conexion',/)
      WRITE(7,225) (I,(S(I,J),J=1,8),I=1,NS)
225   FORMAT(3X,I6,2F8.2,1X,6F8.2,/)
      WRITE(7,227)
227   FORMAT(11X,'Punto    (x,y)',/)
      WRITE(7,226) (I,(P(I,J),J=1,2),I=1,NP)
226   FORMAT(8X,I4,,' ( ',2F7.1,' )',/)
228   IF(LDEV.EQ.9) GO TO 300
      WRITE(7,1) (TIMBRE,I=1,2)
      WRITE(7,229)
229   FORMAT(3X,'?Quiere continuar? (S,N)',/)
      READ(5,6) D
      IF(D.EQ.'S') GO TO 10
300   WRITE(7,1) TIMBRE
      CALL EXIT
      END
```

SUBRUTINA COM:

```
C       Programa que da los datos transmitidos por el digitalizador
C
        SUBROUTINE COM(M,TECLA,X,Y)
        INTEGER IBUFF(20)
        INTEGER X,Y,TECLA
        Y = 0
        X = 0
        I = 1
C       Programa que lee el dato transmitido por el digitalizador.
10      CALL REDPLT(IBUFF(I))
        I = I + 1
        IF (IBUFF(I-1) .NE. 13) GO TO 10
C       M = 1 Modo P/T          M = 0 Modo R/1
        IF (IBUFF(2) .EQ. 62) M = 1
        IF (IBUFF(2) .EQ. 63) M = 0
        TECLA = ICOD(IBUFF)
        K = 4
        L = 8
        X = EJEX(K,L,IBUFF)
        K = 9
        L = 13
        Y = EJEX(K,L,IBUFF)
        RETURN
        END

C       Programa que obtiene el codigo y numero del boton del cursor
C
        FUNCTION ICOD(M)
        INTEGER M(20),A(12)
        DATA A/52,62,61,60,59,58,57,56,55,54,53,51/
        DO 7 J = 1,12
        IF (M(3) .EQ. A(J)) GO TO 10
        GO TO 7
10      ICOD = J - 1
7       CONTINUE
        RETURN
        END


C       Obtener   X y  Y para el punto enviado
C
        FUNCTION EJEX(K,L,IA)
        INTEGER IA(20)
        NASCII = 0
        NUM = 0
        DO 12 J = K,L
          NASCII = IA(J)
          NASCII = NASCII - 48
          NUM = NUM * 10 + NASCII
12      CONTINUE
        EJEX = NUM
        RETURN
        END
```

B.2 GRAFI:

```
C
C  este programa se procesa los datos digitalizados
C  de una curva plana y reproducirla en forma deseada
C
       DIMENSION X(30),Y(30),A(30),B(30),C(30)
       DIMENSION XT(30),YT(30),S(7)
       LOGICAl  COND
       DIMENSION X2(30),H(30),T(30),XM(302),YM(302),TM(302)
       INTEGER NAME,LDFV,LTYPE,NP,INTEQ,TIPO,CONEX,CONT
C
C  asigna los archivos para el graficador y los datos digitalizados.
C
       COND=.TRUE.
       TYPE *,'PROCESO Y PARAMETROS NORMALES(S O N)'
       READ(5,1) LOG
1      FORMAT(A1)
       IF (LOG.EQ.'N') COND=.FALSE.
       XMUL=1.
       M=9
       IS='N'
       NAMF=1
       LDEV=2
       TYPE *,'DAME ARCHIVO DE LOS DATOS DIGITALIZADOS'
       TYPE *,' '
       CALL ASSIGN(NAME,'DK1:F.DAT',-1)
       TYPE *,'DAME ARCHIVO PARA GRAFICADOR'
       TYPE *,' '
       CALL ASSIGN(LDFV,' ',-1)
       IF (COND) GOTO 7
       TYPE *,' # DE PUNTOS INTERMEDIOS PARA SPLINE?'
       READ(5,*) M
7      CONT=0
       INTEQ=3
       CALL PLOTS(0,0,LDFV)
       CALL PLOT(0.,0.,3)
5      TYPE *,' ARCHIVOS FORMATEADOS(S  O  N)?'
       READ(5,1) IFOR
C
C  lee los parametros de la grafica(origen,factor,area,angulo,
C                                    tipo de curva,etc)
C
10     IF (IFOR.EQ.'N') GOTO 30
       READ(NAME,20,END=500,ERR=5) (S(I),I=1,7)
20     FORMAT(7F9.3)
       GOTO 31
30     READ(NAME,END=500,ERR=5) (S(I),I=1,7)
31     NP=S(1)
       LT=0
       LTYPE=S(6)
```

```
      CONEX=S(7)
      FSIN=SIN(S(5))
      FCOS=COS(S(5))
      IF (COND) GOTO 40
      WRITE(7,32) S(4)
32    FORMAT(' FACTOR DE ESCALA= ',F9.3,'DAME EL MULTIPLO DE ESCALA')
      READ(5,*) XMUL
      TYPE *,'QUIERES QUE TE PONGA UN SIMBOLO EN CADA PUNTO(S O N)?'
      READ (5,1) IS
      IF(IS.EQ.'N') GOTO 40
      TYPE *,'DAME CODIGO Y ALTURA?'
      READ(5,*) INTEQ,TAM
      LT=1
40    FACT=S(4)*XMUL
C  lee las informaciones de la grafica y ajusta las medidas
C  (cm-pulgada),gira los ejes si se desea.
C
      DO 70 I=1,NP
      IF (IFOR.EQ.'N') GOTO 65
      READ(NAME,60,END=500,ERR=500) XT(I),YT(I)
60    FORMAT(2F7.1)
      GOTO 68
65    READ(NAME,END=500,ERR=500) XT(I),YT(I)
68    X(I)=((XT(I)-S(2))/127.)*FCOS+((YT(I)-S(3))/127.)*FSIN
70    Y(I)=-((XT(I)-S(2))/127.)*FSIN+((YT(I)-S(3))/127.)*FCOS
      IF((CONEX.EQ.0).AND.(CONT.EQ.0)) GOTO 95
      IF(CONT.EQ.1) GOTO 90
80    ANG0=S(5)
      ORX=X(NP)
      ORY=Y(NP)
      CONT=CONT+1
      GOTO 95
90    ANG1=ANG0-S(5)
      FSIN=SIN(ANG1)
      FCOS=COS(ANG1)
      DO 93 I=1,NP
      XT(I)=X(I)*FCOS+Y(I)*FSIN
      YT(I)=-X(I)*FSIN+Y(I)*FCOS
      X(I)=XT(I)
      Y(I)=YT(I)
93    CONTINUE
      CALL PLOT(ORX,ORY,-3)
      CONT=0
      IF(CONEX.EQ.1) GOTO 80
95    CALL FACTOR(FACT*0.25)
      GOTO (100,400,400,310,320,330,340) IABS(LTYPE)
C
C
C  tipo de curva: linea recta.
C
100   X(NP+1)=0.
```

```
      X(NP+2)=1.0
      Y(NP+1)=0.
      Y(NP+2)=1.0
      IF (ITYPE.LT.0) GOTO 110
      CALL LINE(X,Y,NP,1,LT,INTEQ)
      GOTO 10
C
C  tipo de curva: linea recta punteada.
C
110   CALL DASHL(X,Y,NP,1)
      GOTO 10
C
C tipo de curva: spline(natural,periodica,parametrica natural,
C  parametrica periodica).
C
310   CALL CUBIC1(NP,X,Y,Y2)
      GOTO 325
320   CALL CUBICP(NP,X,Y,Y2)
325   CALL COEF(NP,X,Y,Y2,A,B,C)
      CALL FVAL(NP,M,A,B,C,X,Y,XM,YM)
      GOTO 350
330   CALL CUBPAR(NP,1,T,X,X2,Y,Y2)
335   CALL COEF(NP,T,X,X2,A,B,C)
      CALL EVAL(NP,M,A,B,C,T,X,TM,XM)
      CALL COEF(NP,T,Y,Y2,A,B,C)
      CALL EVAL(NP,M,A,B,C,T,Y,TM,YM)
      GOTO 350
340   CALL CYCLIC(NP,1,T,X,X2,Y,Y2)
      GOTO 335
350   NP1=(NP-1)*M+1
      IF (IS.EQ.'N') GOTO 370
      DO 360 I=1,NP
360   CALL SYMBOL(X(I),Y(I),TAM,INTEQ,0.,-1)
370   IF (LTYPE.LT.0) GOTO 380
      XM(NP1+1)=0.
      XM(NP1+2)=1.
      YM(NP1+1)=0.
      YM(NP1+2)=1.
      CALL LINE(XM,YM,NP1,1,0,INTEQ)
      GOTO 10
C
C  spline punteada
C
380   INI=1
      IPAR=NP/2
      DO 395 I=1,IPAR
      DO 390  J=1,M+1
      XT(J)=XM(J+M*(I-1)*2)
390   YT(J)=YM(J+M*(I-1)*2)
      XT(M+2)=0.
      XT(M+3)=1.
```

```
      YT(M+2)=0.
      YT(M+3)=1.
      CALL LINE(XT,YT,M+1,1,0,INTEQ)
395   CONTINUE
      GOTO 10
C
C  tipo de curva: circulo o arco circular.
C
400   G0=X(1)*Y(2)+X(2)*Y(3)+X(3)*Y(1)-X(1)*Y(3)-X(2)*Y(1)-X(3)*Y(2)
      A0=X(1)*X(1)+Y(1)*Y(1)
      B0=X(2)*X(2)+Y(2)*Y(2)
      C0=X(3)*X(3)+Y(3)*Y(3)
      D=(-A0*Y(2)-B0*Y(3)-C0*Y(1)+A0*Y(3)+B0*Y(1)+C0*Y(2))/G0
      E=(A0*X(2)+B0*X(3)+C0*X(1)-A0*X(3)-B0*X(1)-C0*X(2))/G0
      F0=(-A0*X(2)*Y(3)-B0*X(3)*Y(1)-C0*X(1)*Y(2)+A0*X(3)*Y(2)
     * +B0*X(1)*Y(3)+C0*X(2)*Y(1))/G0
      RO=0.5*SQRT(D*D+E*E-4*F0)
      RF=RO
410   XC=-0.5*D
      YC=-0.5*E
      THO=ATAN2(Y(1)-YC,X(1)-XC)*57.295779
      IF(IABS(LTYPE).EQ.2) GOTO 415
      THF=THO+360.
      GOTO 420
415   THM=ATAN2(Y(2)-YC,X(2)-XC)*57.295779
      THF=ATAN2(Y(3)-YC,X(3)-XC)*57.295779
      IF(THO.LT.THF.AND.THF.LT.THM) THO=THO+360.
      IF(THM.LT.THO.AND.THO.LT.THF) THF=THF-360.
      IF(THM.LT.THF.AND.THF.LT.THO) THO=THO-360.
      IF(THF.LT.THO.AND.THO.LT.THM) THF=THF+360.
420   DI=0.
      IF (LTYPE.LT.0) DI=0.5
      CALL CIRCL(X(1),Y(1),THO,THF,RO,RF,DI)
      GOTO 10
C
C  termina el proceso y cierra el archivo.
C
500   CALL PLOT(0.0,0.0,999)
      CALL EXIT
      END
$
```

B.3 PANTAL:

```
C       Este programa calcula los puntos para funciones "spline"
C       de tipo:NATURAL ,PERIODICA ,PARAMETRICA NATURAL Y ,
C       PARAMETRICA PERIODICA y los puntos de un circulo o de un
C       arco circular.
        REAL X(30),Y(30),A(30),B(30),C(30)
        REAL T(30),X2(30),Y2(30),S(7),AW(1,16)
        REAL XS(370),YS(370),TS(370)
        INTEGER OPCION
        LGR = 1
        LSCR = 2
        NINT = 9
        TYPE *,'ESCRIBE EL NOMBRE DEL ARCHIVO A GRABAR'
        TYPE *,' '
        CALL ASSIGN(LGR,' ',-1)
        TYPE *,'ESCRIBE EL NOMBRE DEL ARCHIVO A LEER'
        TYPE *,' '
        CALL ASSIGN(LSCR,' ',-1)
C       longitud,origen(x,y),factor,angulo,tipo de curva,coneccion
5       READ(LSCR,10,END=1000)(S(J),J=1,7)
10      FORMAT(7F9.3)
        I= 0
20      J = 0
30      J = J+1
        I = I+1
        READ(LSCR,40) SX,SY
40      FORMAT(2F7.1)
        CALL WIND(0.,0.,SX,SY)
        X(J)=(1023./5589.)*SX
        Y(J)=(1023./5590.)*SY
        CALL VPORT(0.,0.,X(J),Y(J))
        IF (S(1) .EQ. I) GO TO 50
        GO TO 30
50      N = J - 1
        VALOR = ABS(S(6))
        IF (N .GT. 0) GO TO (60,70,80,110,120,130,140) VALOR
        GO TO 5
60      SX = -10.0
        SY = 2.0
        DO 90 M = 1, N
        XS(M) = X(M)
        YS(M) = Y(M)
90      CONTINUE
        GO TO 2000
70      N = 0
        GO TO 2
80      N = 1
2       CALL CIRPTO(X(1),Y(1),X(2),Y(2),X(3),Y(3),H1,K,XS,YS,N)
        GO TO 2000
C       Calcula spline natural
110     CALL CUBIC1(N,X,Y,Y2)
150     CALL COEF(N,X,Y,Y2,A,B,C)
        CALL EVAL(N,NINT,A,B,C,X,Y,XS,YS)
        N = NINT*(N-1)+1
        GO TO 2000
```

```
C       Calcula spline periodica.
120     CALL CUBICP(N,X,Y,Y2)
        GO TO 150
C       Calcula spline periodica natural.
130     CALL CUBPAR(N,1,T,X,X2,Y,Y2)
160     CALL COEF(N,T,X,X2,A,B,C)
        CALL EVAL(N,NINT,A,B,C,T,X,TS,XS)
        CALL COEF(N,T,Y,Y2,A,B,C)
        CALL EVAL(N,NINT,A,B,C,T,Y,TS,YS)
        N = NINT*(N-1)+1
        GO TO 2000
C       Calcula spline parametrica periodica.
140     CALL CYCLIC(N,1,T,X,X2,Y,Y2)
        GO TO 160
C       Almacena los puntos calculados.
2000    IR = 1
        IC = 2
        IF (SX .NE. -10.0) GO TO 4
          AW(1,IR)=SX
          AW(1,IC)=SY
          IR=IR+2
          IC=IC+2
4       DO 810 M= 1,N
        IF (IK .EQ. 0) GO TO 35
        IR = IK + 1
        IC = IK + 2
        IK = 0
35      AW(1,IR) = XS(M)
        AW(1,IC) = YS(M)
        IF (IC .LT. 16) GO TO 36
        WRITE(LGR,800) (AW(1,IM),IM=1,16)
800     FORMAT(16F7.1)
        DO 37 IM = 1,16
        AW(1,IM) = 0
37      CONTINUE
        IR = -1
        IC = 0
36      IC = IC + 2
        IR = IR + 2
810     CONTINUE
        IF (IC .GT. 16) GO TO 47
        GO TO 38
47      IR = 1
        IC = 2
38      AW(1,IR) = -1.0
        AW(1,IC) = -1.0
        IK = IC
        IF (IC .EQ. 16) GO TO 39
        GO TO 5
39      WRITE(LGR,800) (AW(1,IM),IM=1,16)
        DO 41 IM = 1,16
        AW(1,IM)= 0
41      CONTINUE
        GO TO 5
1000    IF (IC .EQ. 16) GO TO 999
        WRITE(LGR,800)(AW(1,IM),IM=1,IC)
999     CALL EXIT
        END
```

```
      SUBROUTINE VPORT(WXL,WYB,WXR,WYT)
      IF (WXL.LT.0) GO TO 25
      IF(WYB.LT.0) GO TO 25
      IF(WXR.GE.1023.) GO TO 25
      IF(WYT.GE.1023.) GO TO 25
      GO TO 70
25    TYPE 5
5     FORMAT(X,'ERROR',2X,'TECLEA RETURN')
      ACCEPT 60,A
60    FORMAT(A1)
70    RETURN
      END
```

```
      SUBROUTINE WIND(WXL,WYB,WXR,WYT)
      IF (WXL.LT.0) GO TO 25
      IF(WYB.LT.0) GO TO 25
      IF(WXR.GE.5589.) GO TO 25
      IF(WYT.GE.5590.) GO TO 25
      RETURN
25    TYPE 5
5     FORMAT(X,'ERROR',2X,'TECLEA RETURN')
      ACCEPT 60,A
60    FORMAT(A1)
      RETURN
      END
```

```
C       Este programa lee los datos calculados para curvas planas
C       y los aplica a una serie de rutinas de pantalla
C       para que construyan la grafica.
C
        COMMON/DFILE/IBUFF(2000)
        DIMENSION AW(1,16)
        IL = 1
        LGR = 1
        TYPE *,'ESCRIBE EL NOMBRE DEL ARCHIVO A LEER'
        TYPE *,' '
        CALL ASSIGN(LGR,' ',-1)
        CALL INIT(2000)
        CALL DISPLY(-1)
        DO 3 I=1,31
        TYPE 2
2       FORMAT(X,73(' '))
3       CONTINUE
90      IR = -1
        IC = 0
        READ(LGR,20,END=2000) (AW(1,IM),IM=1,16)
20      FORMAT(16F7.1)
        DO 70 M = 1,8
        IR = IR + 2
        IC = IC + 2
        XS = AW(1,IR)
        YS = AW(1,IC)
        IF(XS .EQ. -1.0 .AND. YS .EQ. -1.0) GO TO 40
        IF (XS .EQ. -10.0 .AND. YS .EQ. 2.0) GO TO 40
        IF (IL .EQ. 1) GO TO 30
        GO TO 50
30      CALL APNT(XS,YS)
        X = XS
        Y = YS
        IL = 0
        GO TO 70
50      CALL VECT(XS-X,YS-Y)
        X = XS
        Y = YS
        GO TO 70
40      IL = 1
70      CONTINUE
        GO TO 90
2000    CALL DISPLY(1)
        CALL EXIT
        END
```

```
      SUBROUTINE CIRPTO(X1,Y1,X2,Y2,X3,Y3,H,K,XA,YA,N)
C     Esta rutina reune a las rutinas CIRTRE,SC,ANGULO,CIR
C     y regresa los puntos completos o partes de un circulo.

      REAL XA(N),YA(N),X1,Y1,X2,Y2,X3,Y3,H,K,R,BETA,FI
      INTEGER N,DIREC
      CALL CIRTRE(X1,Y1,X2,Y2,X3,Y3,H,K,R)
      IF (N .EQ. 1) GO TO 10
      DIREC = SC(X1,Y1,X2,Y2,X3,Y3,H,K)
      IF (DIREC .EQ. 0) GO TO 100
      FI = ATAN2(Y1-K,X1-H)
      BETA = ANGULO(X1,Y1,X3,Y3,H,K)
      GO TO 15
100   BETA = 360 - ANGULO(X1,Y1,X3,Y3,H,K)
      FI = ATAN2(Y3-K,X3-H)
      GO TO 15
10    FI = 0.0
      BETA = 360.0
15    N = 25
      CALL CIR(H,K,XA,YA,N,R,BETA,FI)
      RETURN
      END
```

```
      FUNCTION SC(X1,Y1,X2,Y2,X3,Y3,H,K)
C     Esta rutina determina la direccion en que fue digita
C     zado el circulo.
C     SC = 0 Sentido horario
C     SC = 1 Sentido antihorario

      REAL H,K
      THETA1 = ATAN2(Y1-K,X1-H)*180/3.1416
      THETA2 = ATAN2(Y2-K,X2-H)*180/3.1416
      THETA3 = ATAN2(Y3-K,X3-H)*180/3.1416
      IF(THETA1 .GT. THETA3) GO TO 100
      IF(THETA1 .GT. THETA2) GO TO 200
      IF(THETA2 .GT. THETA3) GO TO 300
      SC = 1
      RETURN
300   SC = 0
      RETURN
100   IF(THETA1 .GT. THETA2) GO TO 500
      IF(THETA2 .GT. THETA3) GO TO 600
      SC = 0
      RETURN
600   SC = 1
      RETURN
500   IF (THETA2 .GT. THETA3) GO TO 700
      SC = 1
      RETURN
700   SC = 0
      RETURN
200   SC = 0
400   RETURN
      END
```

```
      SUBROUTINE CIRTRE(X1,Y1,X2,Y2,X3,Y3,H,K,R)
C      Calculo del centro(H,K) y radio R de un circulo
C      dados tres puntos diferentes cualquiera
C
      REAL K,X1,Y1,X2,Y2,X3,Y3,H,R,A,B,C,P1,P2,P3,P4,P5,P6
      REAL C11,C12,C13,C14,D,E,F
      A = (X1*X1)+(Y1*Y1)
      B = (X2*X2)+(Y2*Y2)
      C = (X3*X3)+(Y3*Y3)
      P1 = Y2 - Y3
      P2 = X2 - X3
      P3 = B - C
      P4 = X2*Y3 - Y2*X3
      P5 = B*Y3 - C*Y2
      P6 = B*X3 - X2*C
      C11 = X1*P1 - Y1*P2 + P4
      C12 = A*P1 - Y1*P3 + P5
      C13 = A*P2 - X1*P3 + P6
      C14 = A*P4 - X1*P5 + Y1*P6
      IF (C11 .EQ. 0.0) GO TO 30
      D = -C12/C11
      E = C13/C11
      F = -C14/C11
      H = -D*0.5
      K = -F*0.5
      R = (D*D) +(E*E)-4.0*F
      IF (R .LE. 0.0) GO TO 30
      R = SQRT( R )
      R = R*0.5
30    RETURN
      END
```

```
      FUNCTION ANGULO(X1,Y1,X3,Y3,H,K)
C     Se obtiene el angulo entre la abertura de dos rectas.

      REAL H,K
      THETA1 = ATAN2(Y1-K,X1-H)*180/3.1416
      THETA2 = ATAN2(Y3-K,X3-H)*180/3.1416
      IF((THETA1 + THETA2) .GT. 360) GO TO 100
      IF(THETA1 .GT. THETA2) GO TO 200
      ANGULO = THETA2 - THETA1
      RETURN
200   ANGULO = 360 - (THETA1 - THETA2)
      RETURN
100   IF(THETA1 .GT. THETA2) GO TO 400
      ANGULO = THETA2 - THETA1
      RETURN
400   ANGULO = 360 - (THETA1 - THETA2)
      RETURN
      END


      SUBROUTINE CIR(H,K,XA,YA,N,R,BETA,FI)
C     Esta rutina calcula los puntos de un circulo

      REAL RAD,H,K,R,XA(N),YA(N),FI,BETA,ANG
      INTEGER N
      IF (FI .EQ. 0.0) GO TO 1000
      N = BETA *N/360.0
1000  ANG = BETA/(N-1)
      DO 7 M=1,N
      TETHA = ANG * (M-1)
      RAD = TETHA/180.0 * 3.1416 + FI
      XA(M)=H+R*COS(RAD)
      YA(M)=K+R*SIN(RAD)
7     CONTINUE
      RETURN
      END
```

B.4 FRONT:

```
C       ESTE FROGRAMA CALCULA EL AREA,CENTROIDE,Y EL MOMENTO
C       DE INERCIA CENTRAL DE UNA AREA ENCERRADA POR UNA
C       CURVA PLANA CERRADA SIMPLE.
C
        REAL AX(50),BX(50),CX(50),X(50),X2(50)
        REAL AY(50),BY(50),CY(50),Y(50),Y2(50)
        REAL T(50),DT(50),S(7)
        REAL XO,YO,IXX,IXY,IYX,IYY,AR,QX,QY,FACT,MULT,FA
        REAL V1,V2,T1,T2
        INTEGER NAME,IS
        NAME=1
5       WRITE(7,6)
6       FORMAT(3X,' NOMBRE DEL ARCHIVO DE LOS DATOS DIGITALIZADOS?',/)
        CALL ASSIGN(NAME,' ',-1)
        WRITE(7,8)
8       FORMAT(3X,'ARCHIVO FORMATEADO O NO? (F,N)',/)
        READ(5,7) IS
7       FORMAT(A1)
9       IF(IS.EQ.'N') GO TO 15
        READ(NAME,10,END=500) (S(I),I=1,7)
10      FORMAT(7F9.3)
        GO TO 16
15      READ(NAME,END=500) (S(I),I=1,7)
16      N=S(1)
        XO=S(2)
        YO=S(3)
        FACT=S(4)
        COSA=COS(S(5))
        SINA=SIN(S(5))
        CURVA=S(6)
        IF(ABS(CURVA).NE.7.0) GO TO 5
        IF(IS.EQ.'N') GO TO 25
        READ(NAME,20,END=500) (X(I),Y(I),I=1,N)
20      FORMAT(2F7.1)
        GO TO 30
25      READ(NAME,END=500) (X(I),Y(I),I=1,N)
30      WRITE(7,40) FACT
40      FORMAT(3X,'EL FACTOR DE ESCALA = ',F7.3,3X,' MULTIPLO=?',/)
        READ(5,*) MULT
        FACT=FACT*MULT
        DO 50 I=1,N
        X(I)=0.01*FACT*(X(I)-XO)
        Y(I)=0.01*FACT*(Y(I)-YO)
        Z=X(I)
        X(I)=X(I)*COSA-Y(I)*SINA
        Y(I)=Z*SINA+Y(I)*COSA
50      CONTINUE
        CALL CICLIC(N,T,DT,X,X2,Y,Y2)
        CALL COEF(N,T,X,X2,AX,BX,CX)
        CALL COEF(N,T,Y,Y2,AY,BY,CY)
        CALL AREA(N,DT,AX,BX,CX,X,AY,BY,CY,Y,AR)
        CALL MOMENT(N,DT,AX,BX,CX,X,AY,BY,CY,Y,QX,QY)
```

```
        CALL MIC(N,DT,AX,BX,CX,X,AY,BY,CY,Y,IXX,IXY,IYX,IYY)
        CALL FJFP(IXX,IYY,IXY,V1,V2,T1,T2)
        WRITE(7,70) AR
70      FORMAT(//,10X,'AREA = ',F14.4,'  CM**2')
        QX=QX/AR
        QY=QY/AR
        WRITE(7,80) QX,QY
80      FORMAT(//,10X,'CENTROIDE = (',2F14.4,' ) CM')
        WRITE(7,85)
85      FORMAT(//,10X,'MOMENTO DE INERCIA CON RESPECTO AL ORIGEN =CM**4')
        WRITE(7,90) IXX,IXY,IXY,IYY
        IXX=IXX-AR*QY**2
        IXY=IXY-AR*QX*QY
        IYY=IYY-AR*QX**2
        WRITE(7,88)
88      FORMAT(//,10X,'MOMENTO DE INERCIA CON RESPECTO AL CENTROIDE =')
        WRITE(7,90) IXX,IXY,IXY,IYY
90      FORMAT(/,30X,2F15.4,/,30X,2F15.4,/)
        WRITE(7,95)
95      FORMAT(/,10X,'EJE:',6X,'ANGULO:',6X,'MOMENTO:',//)
        WRITE(7,96) T1*180./3.1415926,V1
96      FORMAT(10X,' 1',7X,F7.1,' GRAD.',3X,F15.4)
        WRITE(7,97) T2*180./3.1415926,V2
97      FORMAT(10X,' 2',7X,F7.1,' GRAD.',3X,F15.4/)
        GO TO 9
        WRITE(7,100)
100     FORMAT(//,3X,'OTRO ARCHIVO ? (S,N)',/)
        READ(5,*) IS
        IF(IS.EQ.'S') GOTO 5
500     CALL EXIT
        END
```

```
C         ESTA SUBRUTINA CALCULA EL AREA ENCERRADA POR UNA CURV
C         PLANA SIMPLE UTILIZANDO LAS FUNCIONES SPLINES CICLICAS
C         E INTEGRANDO SOBRE LA FRONTERA. EL SIGNO DEL AREA SERA
C         POSITIVO SI LA CURVA ES DIGITALIZADA EN SENTIDO ANTIHORARIO.
C
          SUBROUTINE AREA(N,DT,AX,BX,CX,DX,AY,BY,CY,DY,S)
          REAL AX(2),BX(2),CX(2),DX(2)
          REAL AY(2),BY(2),CY(2),DY(2)
          REAL DT(2),P,S
          N1=N-1
          S=0.
          DO 1 K=1,N1
          P=0.2*(AY(K)*BX(K)-AX(K)*BY(K))*DT(K)
          P=(P+0.5*(AY(K)*CX(K)-AX(K)*CY(K)))*DT(K)
          P=(P+AY(K)*DX(K)-AX(K)*DY(K)+(BY(K)*CX(K)-BX(K)*CY(K))/3.)*DT(K)
          P=(P+BY(K)*DX(K)-BX(K)*DY(K))*DT(K)
          P=(P+CY(K)*DX(K)-CX(K)*DY(K))*DT(K)
          S=S+P
1         CONTINUE
          S=0.5*S
          RETURN
          END


C         SUBRUTINA PARA CALCULAR EJES PRINCIPALES Y MOMENTOS
C         PRINCIPALES DE LA MATRIZ DE MOMENTOS DE INERCIA
C
          SUBROUTINE EJEP(A,B,C,V1,V2,T1,T2)
          X=(A+B)/2.
          Y=(A-B)/2.
          Z=SQRT(Y**2+C**2)
          V1=X-Z
          V2=X+Z
          T1=ATAN2(C,Y+Z)
          T2=ATAN2(C,Y-Z)
          RETURN
          END
```

```
C         ESTA SUBRUTINA CALCULA EL PRIMER MOMENTO DEL AREA ENCERRADA
C         POR UNA CURVA PLANA SIMPLE CON FUNCIONES SPLINES CICLICAS E
C         INTEGRANDO SOBRE LA FRONTERA. EL SENTIDO DE DIGITALIZACION
C         DEBE SER ANTIHORARIO(CCW).
C
          SUBROUTINE MOMENT(N,DT,AX,BX,CX,DX,AY,BY,CY,DY,QX,QY)
          REAL AX(2),BX(2),CX(2),DX(2)
          REAL AY(2),BY(2),CY(2),DY(2)
          REAL DT(2),AA,AB,BAC,ADBC,CBD,CD,DD,QX,QY
          N1=N-1
          QX=0.
          QY=0.
          DO 1 K=1,N1
          AA=AX(K)**2+AY(K)**2
          AB=AX(K)*BX(K)+AY(K)*BY(K)
          BAC=BX(K)**2+BY(K)**2+2.*(AX(K)*CX(K)+AY(K)*CY(K))
          ADBC=AX(K)*DX(K)+AY(K)*DY(K)+BX(K)*CX(K)+BY(K)*CY(K)
          CBD=CX(K)**2+CY(K)**2+2.*(BX(K)*DX(K)+BY(K)*DY(K))
          CD=CX(K)*DX(K)+CY(K)*DY(K)
          DD=DX(K)**2+DY(K)**2
C
          QY=QY+Q1(AA,AB,BAC,ADBC,CBD,CD,DD,AX(K),BX(K),CX(K),DX(K),DT(K))
          QX=QX+Q1(AA,AB,BAC,ADBC,CBD,CD,DD,AY(K),BY(K),CY(K),DY(K),DT(K))
1         CONTINUE
          QX=QX/2.
          QY=-QY/2.
          RETURN
          END
C
          FUNCTION Q1(X1,X2,X3,X4,X5,X6,X7,A,B,C,D,T)
          REAL Z,X1,X2,X3,X4,X5,X6,X7,A,B,C,D,T
          Z=(A*X1*T/3.+(B*X1+3.*A*X2)/4.)*T
          Z=(Z+(C*X1+4.*B*X2+3.*A*X3)/7.)*T
          Z=(Z+(C*X2+B*X3)/3.+A*X4)*T
          Z=(Z+(C*X3+4.*B*X4+3.*A*X5)/5.)*T
          Z=(Z+(C*X4+B*X5+3.*A*X6)/2.)*T
          Z=(Z+(C*X5+4.*B*X6)/3.+A*X7)*T
          Z=(Z+C*X6+B*X7)*T
          Q1=(Z+C*X7)*T
          RETURN
          END
```

```
C         ESTA SUBRUTINA CALCULA EL MOMENTO DE INERCIA CENTRAL DE
C         UN AREA ENCERRADA POR UNA CURVA PLANA.EL SENTIDO DE
C         DIGITALIZAR LA CURVA DEBE SER ANTIHORARIO.
C
          SUBROUTINE MIC(N,DT,AX,BX,CX,DX,AY,BY,CY,DY,IXX,IXY,IYX,IYY)
          REAL AX(2),BX(2),CX(2),DX(2)
          REAL AY(2),BY(2),CY(2),DY(2)
          REAL DT(2),F(7)
          REAL I1,I2,I3,I4,IXX,IXY,IYX,IYY
          N1=N-1
          I1=0.
          I2=0.
          I3=0.
          I4=0.
          DO 1 K=1,N1
C
          F(1)=AX(K)**2+AY(K)**2
          F(2)=AX(K)*BX(K)+AY(K)*BY(K)
          F(3)=BX(K)**2+BY(K)**2+2.*(AX(K)*CX(K)+AY(K)*CY(K))
          F(4)=AX(K)*DX(K)+AY(K)*DY(K)+BX(K)*CX(K)+BY(K)*CY(K)
          F(5)=CX(K)**2+CY(K)**2+2.*(BX(K)*DX(K)+BY(K)*DY(K))
          F(6)=CX(K)*DX(K)+CY(K)*DY(K)
          F(7)=DX(K)**2+DY(K)**2
C
          I1=I1+P(F,AX(K),BX(K),CX(K),DX(K),DT(K))
          I2=I2+P(F,AY(K),BY(K),CY(K),DY(K),DT(K))
          I3=I3+Q(F,AX(K),BX(K),CX(K),DX(K),AY(K),BY(K),CY(K),DY(K),DT(K))
          I4=I4+Q(F,AY(K),BY(K),CY(K),DY(K),AX(K),BX(K),CX(K),DX(K),DT(K))
1         CONTINUE
          IXX=(3.*I3+I4)/(-8.)
          IXY=0.25*(I2-I1)
          IYX=IXY
          IYY=(3.*I4+I3)/8.
          RETURN
          END
C
          FUNCTION P(F,A,B,C,D,T)
          REAL F(2),G1,G2,G3,G4,G5,G6,A,B,C,D,T,Z
          G1=A**2
          G2=A*B
          G3=B**2+2.*A*C
          G4=B*C+A*D
          G5=C**2+2.*B*D
          G6=C*D
          Z=(G1*F(1)*T/4.+(5.*G2*F(1)+6.*G1*F(2))/11.)*T
          Z=(Z+0.2*G3*F(1)+G2*F(2)+0.3*G1*F(3))*T
          Z=(Z+(3.*G4*F(1)+4.*G3*F(2)+5.*G2*F(3)+6.*G1*F(4))/9.)*T
          Z=(Z+(G5*F(1)+6.*G4*F(2)+2.*G3*F(3)+10.*G2*F(4)+3.*G1*F(5))/8.)*
     *    T
```

```
      Z=(Z+(G6*F(1)+2.*G5*F(2)+3.*G4*F(3)+4.*G3*F(4)+5.*G2*F(5)+
     * 6.*G1*F(6))/7.)*T
      Z=(Z+(2.*G6*F(2)+G5*F(3)+6.*G4*F(4)+2.*G3*F(5)+
     * 10.*G2*F(6)+3.*G1*F(7))/6.)*T
      Z=(Z+(G6*F(3)+2.*G5*F(4)+3.*G4*F(5)+4.*G3*F(6))/5.+G2*F(7))*T
      Z=(Z+(G6*F(4)+0.5*G5*F(5)+3.*G4*F(6)+G3*F(7))/2.)*T
      Z=(Z+(G6*F(5)+2.*G5*F(6))/3.+G4*F(7))*T
      P=((Z+G6*F(6)+G5*F(7)/2.)*T+G6*F(7))*T
      RETURN
      END
C
      FUNCTION Q(F,AX,BX,CX,DX,AY,BY,CY,DY,T)
      REAL F(2),H1,H2,H3,H4,H5,H6,T,W
      REAL AX,BX,CX,DX,AY,BY,CY,DY
      H1=AX*AY
      H2=2.*AY*BX+3.*AX*BY
      H3=AY*CX+2.*BX*BY+3.*AX*CY
      H4=BY*CX+2.*BX*CY+3.*AX*DY
      H5=CX*CY+2.*BX*DY
      H6=CX*DY
      W=(H1*F(1)/4.*T+(H2*F(1)+6.*H1*F(2))/11.)*T
      W=(W+(H3*F(1)+2.*H2*F(2)+3.*H1*F(3))/10.)*T
      W=(W+(H4*F(1)+2.*H3*F(2)+H2*F(3)+6.*H1*F(4))/9.)*T
      W=(W+(H5*F(1)+2.*H4*F(2)+H3*F(3)+2.*H2*F(4)+3.*H1*F(5))/8.)*T
      W=(W+(H6*F(1)+2.*H5*F(2)+H4*F(3)+2.*H3*F(4)+
     * H2*F(5)+6.*H1*F(6))/7.)*T
      W=(W+(2.*H6*F(2)+H5*F(3)+2.*H4*F(4)+H3*F(5)+2.*H2*F(6)+
     * 3.*H1*F(7))/6.)*T
      W=(W+(H6*F(3)+2.*H5*F(4)+H4*F(5)+2.*H3*F(6)+H2*F(7))/5.)*T
      W=(W+(2.*H6*F(4)+H5*F(5)+2.*H4*F(6)+H3*F(7))/4.)*T
      W=(W+(H6*F(5)+2.*H5*F(6)+H4*F(7))/3.)*T
      Q=(W+(H6*F(6)+0.5*H5*F(7))*T+H6*F(7))*T
      RETURN
      END
```

```
C         ESTA SUBRUTINA CALCULA LAS SEGUNDA DERIVADAS DE X,Y,
C         Y LOS INCREMENTOS DEL PARAMETRO T.
C         UTILIZA LA SUBRUTINA CUBICP.
C
          SUBROUTINE CICLIC(N,T,DT,X,X2,Y,Y2)
          REAL X(2),X2(2),Y(2),Y2(2)
          REAL T(2),DT(2)
          REAL U,V
          T(1)=0.
          I1=1
          DO 1 K=2,N
                   U=(X(K)-X(I1))**2
                   V=(Y(K)-Y(I1))**2
                   DT(I1)=DSQRT(U+V)
                   T(K)=T(I1)+DT(I1)
                   I1=K
1         CONTINUE
          CALL CUBICP(N,T,X,X2)
          CALL CUBICP(N,T,Y,Y2)
          RETURN
          END


          SUBROUTINE COEF(N,X,Y,Y2,A,B,C)
          REAL X(2),Y(2),Y2(2),A(2),B(2),C(2)
          DO 10 K=1,N-1
          A(K)=(Y2(K+1)-Y2(K))/(6.*(X(K+1)-X(K)))
          B(K)=0.5*Y2(K)
          C(K)=(Y(K+1)-Y(K))/(X(K+1)-X(K))
          C(K) = C(K) - (X(K+1)-X(K))*(Y2(K+1)+2.*Y2(K))/6.
10        CONTINUE
          RETURN
          END
```

■.5 SOLIDO:

```
C
C   este programa es para calcular las propiedades geometrica
C   globales de un solido de revolucion,las propiedades son:
C   volumen, primeros momentos, centroide, los momentos de inercia.
C
        DIMENSION X(50),Y(50),XT(50),YT(50),T(50)
        DIMENSION X2(50),Y2(50),AX(50),BX(50),CX(50)
        DIMENSION AY(50),BY(50),CY(50),S(7)
        INTEGER   TIPO
        REAL      VOL,MOM,INE1,INE2,INE3
        LD1=1
        TYPE *,' UN ARCHIVO DE LOS DATOS DIGITALIZADOS:'
        TYPE *,' '
        CALL ASSIGN(LD1,'DK1:ARCH.DAT',-1)
5       READ(LD1,10,END=200,ERR=200) (S(I),I=1,7)
10      FORMAT(7F9.3)
        TIPO=S(6)
        N=S(1)
        OX=S(2)
        OY=S(3)
        ANG=S(5)
        READ(LD1,20) ((X(I),Y(I)),I=1,N)
20      FORMAT(2F7.1)
        IF(TIPO.NE.1) GOTO 100
        IF(N.NE.4) GOTO 30
        X(1)=(X(1)+X(2))/2.
        X(2)=(X(3)+X(4))/2.
        Y(1)=(Y(1)+Y(2))/2.
        Y(2)=(Y(3)+Y(4))/2.
        ANG=ATAN2(Y(2)-Y(1),X(2)-X(1))
        OX=X(1)
        OY=Y(1)
        GOTO 5
30      READ(LD1,10,END=200,ERR=200) (S(I),I=1,7)
        TIPO=S(6)
        IF(TIPO.LT.4) GOTO 200
100     FSIN=SIN(ANG)
```

```
      FCOS=COS(ANG)
      DO 120 J=1,N
      XT(J)=(X(J)-OX)/100.
      YT(J)=(Y(J)-OY)/100.
      X(J)=XT(J)*FCOS+YT(J)*FSIN
      Y(J)=-XT(J)*FSIN+YT(J)*FCOS
120   CONTINUE
      CALL CUBPAR(N,1,T,X,X2,Y,Y2)
      CALL COFF(N,T,X,X2,AX,BX,CX)
      CALL COFF(N,T,Y,Y2,AY,BY,CY)
      CALL VOLPAR(N,AX,BX,CX,X,AY,BY,CY,Y,T,VOL,TIPO)
      CALL MOMPAR(N,AX,BX,CX,X,AY,BY,CY,Y,T,MOM)
      CALL INFPAR(N,AX,BX,CX,X,AY,BY,CY,Y,T,INE1,INE2)
      VOL=VOL*(S(4))**3
      TYPE *,' VOLUMEN = ',VOL,'  (CM**3)'
      TYPE *,' '
      MOM=MOM*(S(4))**4
      TYPE *,' PRIMER MOMENTO QXX =',MOM,'  (CM**4)'
      TYPE *,' '
      RX=MOM/VOL
      TYPE *,' EL CENTROIDE RXX =',RX,'  (CM) AL ORIGEN '
      TYPE *,' '
      INE1=INE1*(S(4))**5
      INE2=INE2*(S(4))**5
      INE3=INE2-RX**2*VOL
      TYPE *,' EL MOMENTO DE INERCIA IXX = ',INE1,'  (CM**5)'
      TYPE *,' '
      TYPE *,' EL MOMENTO DE INERCIA IYY = ',INE2,'  (CM**5)'
      TYPE *,' '
      TYPE *,' MOMENTO DE INERECIA CENTROIDAL=',INE3,'  (CM**5)'
      TYPE *,' '
      TYPE *,' ***************************************************'
      TYPE *,' '
200   CALL EXIT
      END
```

```
C
C  esta subrutina calcula el volumen del solido de revolucion aplicando
C  las funciones spline parametricas
C
      SUBROUTINE VOLPAR(N,AX,BX,CX,DX,AY,BY,CY,DY,T,VOL,TIPO)
      DIMENSION AX(2),BX(2),CX(2),AY(2),BY(2),CY(2),T(2)
      DIMENSION DX(50),DY(50),Z(10)
      INTEGER TIPO
      VOL=0.
      DO 20 K=1,N-1
      Z(1)=(AY(K)*(BX(K)*AY(K)-AX(K)*BY(K)))/8.
      Z(2)=(BY(K)*(BX(K)*AY(K)-AX(K)*BY(K))+2.*AY(K)*(CX(K)*AY(K)-AX(K)
     * *CY(K)))/7.
      W=CX(K)*BY(K)-BX(K)*CY(K)+3.*DX(K)*AY(K)-3.*AX(K)*DY(K)
      Z(3)=(CY(K)*(BX(K)*AY(K)-AX(K)*BY(K))+2.*BY(K)*(CX(K)*AY(K)-AX(K)
     * *CY(K))+AY(K)*W)/6.
      Z(4)=(DY(K)*(BX(K)*AY(K)-AX(K)*BY(K))+2.*CY(K)*(CX(K)*AY(K)-AX(K)
     * *CY(K))+BY(K)*W+2.*AY(K)*(
     * DX(K)*BY(K)-BX(K)*DY(K)))/5.
      Z(5)=(2.*DY(K)*(CX(K)*AY(K)-AX(K)*CY(K))+CY(K)*W+2.*BY(K)*(DX(K)*
     * BY(K)-BX(K)*DY(K))+AY(K)*(
     * DX(K)*CY(K)-CX(K)*DY(K)))/4.
      Z(6)=(DY(K)*W+2.*CY(K)*(DX(K)*BY(K)-BX(K)*DY(K))+BY(K)*(DX(K)*CY(
     * K)-CX(K)*DY(K)))/3.
      Z(7)=(2.*DY(K)*(DX(K)*BY(K)-BX(K)*DY(K))+CY(K)*(DX(K)*CY(K)-CX(K)
     * *DY(K)))/2.
      Z(8)=DY(K)*(DX(K)*CY(K)-CX(K)*DY(K))
      DT=T(K+1)-T(K)
9     W=Z(1)*DT
      DO 10 I=2,8
10    W=(W+Z(I))*DT
15    VOL=VOL+W
20    CONTINUE
      VOL=2.094395102*ABS(VOL)
      RETURN
      END
```

```
C  esta subrutina calcula los primeros momentos con respecto al
C  origen del solido de revolucion usando spline parametrica
C
      SUBROUTINE MOMPAR(N,AX,BX,CX,DX,AY,BY,CY,DY,T,MOM)
      DIMENSION AX(2),BX(2),CX(2),DX(50)
      DIMENSION AY(2),BY(2),CY(2),DY(50),T(2),Z(14)
      REAL MOM
      MOM=0.
      DO 30 K=1,N-1
      A6=AX(K)**2+AY(K)**2
      A5=2.*(AX(K)*BX(K)+AY(K)*BY(K))
      A4=2.*AX(K)*CX(K)+BX(K)**2+2.*AY(K)*CY(K)+BY(K)**2
      A3=2.*(AX(K)*DX(K)+BX(K)*CX(K)+AY(K)*DY(K)+BY(K)*CY(K))
      A2=2.*BX(K)*DX(K)+CX(K)**2+2.*BY(K)*DY(K)+CY(K)**2
      A1=2.*(CX(K)*DX(K)+CY(K)*DY(K))
      A0=DX(K)**2+DY(K)**2
      B5=3.*AY(K)**2
      B4=5.*AY(K)*BY(K)
      B3=4.*AY(K)*CY(K)+2.*BY(K)**2
      B2=3.*BY(K)*CY(K)+3.*AY(K)*DY(K)
      B1=2.*BY(K)*DY(K)+CY(K)**2
      B0=CY(K)*DY(K)
      Z(1)=A6*B5/12.
      Z(2)=(B5*A5+B4*A6)/11.
      Z(3)=(A5*B4+B4*A5+B3*A6)/10.
      Z(4)=(B5*A3+B4*A4+B3*A5+B2*A6)/9.
      Z(5)=(B5*A2+B4*A3+B3*A4+B2*A5+B1*A6)/8.
      Z(6)=(B5*A1+B4*A2+B3*A3+B2*A4+B1*A5+B0*A6)/7.
      Z(7)=(B5*A0+B4*A1+B3*A2+B2*A3+B1*A4+B0*A5)/6.
      Z(8)=(B4*A0+B3*A1+B2*A2+B1*A3+B0*A4)/5.
      Z(9)=(B3*A0+B2*A1+B1*A2+B0*A3)/4.
      Z(10)=(B2*A0+B1*A1+B0*A2)/3.
      Z(11)=(B1*A0+B0*A1)/2.
      Z(12)=B0*A0
      DT=T(K+1)-T(K)
      W=Z(1)*DT
      DO 10 I=2,12
10    W=(W+Z(I))*DT
      MOM=MOM+W
30    CONTINUE
      MOM=-3.141592653579793*MOM
      RETURN
      END
```

```
C
C  esta subrutina calcula los momentos de inercia del solido
C  de revolucion usando spline parametrica
C
      SUBROUTINE INEPAR(N,AX,BX,CX,DX,AY,BY,CY,DY,T,INE1,INE2)
      DIMENSION AX(2),BX(2),CX(2),DX(50)
      DIMENSION AY(2),BY(2),CY(2),DY(50),T(2),Z(15)
      REAL INE1,INE2,INE
      IN=0
1     INE=0.
      DO 30 K=1,N-1
      A6=AX(K)**2+AY(K)**2
      A5=2.*(AX(K)*BX(K)+AY(K)*BY(K))
      A4=2.*AX(K)*CX(K)+BX(K)**2+2.*AY(K)*CY(K)+BY(K)**2
      A3=2.*(AX(K)*DX(K)+BX(K)*CX(K)+AY(K)*DY(K)+BY(K)*CY(K))
      A2=2.*BX(K)*DX(K)+CX(K)**2+2.*BY(K)*DY(K)+CY(K)**2
      A1=2.*(CX(K)*DX(K)+CY(K)*DY(K))
      A0=DX(K)**2+DY(K)**2
      IF (IN.EQ.1) GOTO 3
      R5=15.*AX(K)*AY(K)
      R4=12.*AY(K)*BX(K)+13.*AX(K)*BY(K)
      R3=9.*CX(K)*AY(K)+10.*BX(K)*BY(K)+11.*CY(K)*AX(K)
      R2=8.*CY(K)*BX(K)+7.*CX(K)*BY(K)+6.*AY(K)*DX(K)+9.*AX(K)*DY(K)
      R1=5.*CX(K)*CY(K)+4.*DX(K)*BY(K)+6.*DY(K)*BX(K)
      R0=2.*DX(K)*CY(K)+3.*CX(K)*DY(K)
      GOTO 2
3     R5=15.*AX(K)*AY(K)
      R4=9.*AX(K)*BY(K)+16.*AY(K)*BX(K)
      R3=3.*AX(K)*CY(K)+10.*BX(K)*BY(K)+17.*CX(K)*AY(K)
      R2=4.*BX(K)*CY(K)+11.*CX(K)*BY(K)+18.*DX(K)*AY(K)-3.*DY(K)*AX(K)
      R1=5.*CX(K)*CY(K)+12.*DX(K)*BY(K)-2.*DY(K)*BX(K)
      R0=6.*DX(K)*CY(K)-CX(K)*DY(K)
2     X11=R5*A6
      X10=R5*A5+R4*A6
      X9=R5*A4+R4*A5+R3*A6
      X8=R5*A3+R4*A4+R3*A5+R2*A6
      X7=R5*A2+R4*A3+R3*A4+R2*A5*R1*A6
      X6=R5*A1+R4*A2+R3*A3+R2*A4+R1*A5+R0*A6
      X5=R5*A0+R4*A1+R3*A2+R2*A3+R1*A4+R0*A5
```

```
      X4=R4*A0+R3*A1+R2*A2+R1*A3+R0*A4
      X3=R3*A0+R2*A1+R1*A2+R0*A3
      X2=R2*A0+R1*A1+R0*A2
      X1=R1*A0+R0*A1
      X0=R0*A0
      Z(1)=AY(K)*X11/15.
      Z(2)=(AY(K)*X10+BY(K)*X11)/14.
      Z(3)=(AY(K)*X9+BY(K)*X10+CY(K)*X11)/13.
      Z(4)=(AY(K)*X8+BY(K)*X9+CY(K)*X10+DY(K)*X11)/12.
      Z(5)=(AY(K)*X7+BY(K)*X8+CY(K)*X9+DY(K)*X10)/11.
      Z(6)=(AY(K)*X6+BY(K)*X7+CY(K)*X8+DY(K)*X9)/10.
      Z(7)=(AY(K)*X5+BY(K)*X6+CY(K)*X7+DY(K)*X8)/9.
      Z(8)=(AY(K)*X4+BY(K)*X5+CY(K)*X6+DY(K)*X7)/8.
      Z(9)=(AY(K)*X3+BY(K)*X4+CY(K)*X5+DY(K)*X6)/7.
      Z(10)=(AY(K)*X2+BY(K)*X3+CY(K)*X4+DY(K)*X5)/6.
      Z(11)=(AY(K)*X1+BY(K)*X2+CY(K)*X3+DY(K)*X4)/5.
      Z(12)=(AY(K)*X0+BY(K)*X1+CY(K)*X2+DY(K)*X3)/4.
      Z(13)=(BY(K)*X0+CY(K)*X1+DY(K)*X2)/3.
      Z(14)=(CY(K)*X0+DY(K)*X1)/2.
      Z(15)=DY(K)*X0
      DT=T(K+1)-T(K)
      W=Z(1)*DT
      DO 10 I=2,15
10    W=(W+Z(I))*DT
      INE=INE+W
30    CONTINUE
      IF (IN.EQ.1) GOTO 20
      INE1=ABS(3.141592653579793*INE/5.)
      IN=1
      GOTO 1
20    INE2=ABS(3.141592653579793*INE/10.)
      RETURN
      END
```

B.6 LOCAL:

```
      $
C
C  Este programa es para calcular las propiedades geometricas
C  locales de una curva plana y representar en forma grafica
C
      DIMENSION X(30),Y(30),A(30),B(30),C(30)
      DIMENSION X2(30),Y2(30),T(30),S(7),XM(212),YM(212)
      DIMENSION XM1(212),YM1(212),XM2(212),TM(212)
      INTEGER TIPO
      M = 7
      NAME = 1
      LDEV = 2
      TYPE *,'DAME ARCHIVO DE LOS DATOS DIGITALIZADOS'
      TYPE *,' '
      CALL ASSIGN(NAME,' ',-1)
      TYPE *,'DAME ARCHIVO PARA GRAFICADOR'
      TYPE *,' '
      CALL ASSIGN(LDEV,' ',-1)
      CALL PLOTS(0,0,LDEV)
      CALL PLOT(0.,0.,3)
      READ(NAME,20,END=500)(S(I),I=1,7)
20    FORMAT(7F9.3)
      NP = S(1)
      NP1 = (NP-1)*M+1
      K1 = NP1+1
      K2= NP1+2
      LTYPE = S(6)
      FSIN =S(4)*SIN(S(5))
      FCOS =S(4)*COS(S(5))
      WRITE(7,32)S(4)
32    FORMAT(' FACTOR DE ESCALA= ',F9.3,' DAME EL FACTOR PARA GRAFICA')
      READ(5,*)FACT
      DO 70 I = 1,NP
      READ(NAME,60,END=500)XT,YT
60    FORMAT(2F7.1)
      X(I)=((XT-S(2))/100.)*FCOS+((YT-S(3))/100.)*FSIN
70    Y(I)=-((XT-S(2))/100.)*FSIN+((YT-S(3))/100.)*FCOS
      CALL FACTOR(FACT*0.25)
      TYPE *,'DAME LONGITUD DE EJE X,Y (SI FACT=1,LONG=10)'
      READ(5,*) FLONX,FLONY
      GO TO (310,320,330,340) IABS(LTYPE)-3
310   CALL CUBIC1(NP,X,Y,Y2)
      GO TO 325
C
C  spline natural o periodica
C
320   CALL CUBICP(NP,X,Y,Y2)
325   CALL COEF(NP,X,Y,Y2,A,B,C)
      CALL EVAL(NP,M,A,B,C,X,Y,XM,YM)
      CALL EVAL1(NP,M,A,B,C,X,Y,XM,YM1)
      CALL EVAL2(NP,M,A,B,C,X,Y,XM,TM)
      CALL SCALE(XM,FLONX,NP1,1)
      CALL SCALE(YM,FLONY,NP1,1)
      CALL SCALE(YM1,FLONY,NP1,1)
      CALL SCALE(TM,FLONY,NP1,1)
      FIRSTX = XM(K1)
      FIRSTY=YM(K1)
```

```
      IF (FIRSTY.GE.YM1(K1)) FIRSTY=YM1(K1)
      IF (FIRSTY.GE.TM(K1)) FIRSTY=TM(K1)
      DELTX=XM(K2)
      DELTY=YM(X2)
      IF (DFLTY.LT.YM1(K2)) DELTY = YM1(K2)
      IF (DFLTY.LT.TM(K2)) DFLTY = TM(K2)
      YM(K1)=FIRSTY
      YM1(K1)=FIRSTY
      TM(K1)=FIRSTY
      YM(K2)=DELTY
      TM(K2)=DELTY
      YM1(K2)=DELTY
      CALL AXIS(0.,0.,5HEJE X,-5,FLONX,0.,FIRSTX,DELTX)
      CALL AXIS(0.,0.,20HEJE Y(X),Y'(X),Y"(X),20,FLONY,
     * 90.0,FIRSTY,DFLTY)
      CALL LINE(XM,YM,NP1,1,0,0)
      CALL LINE(XM,YM1,NP1,1,0,0)
      CALL NEWPEN(2)
      CALL LINE(XM,TM,NP1,1,0,0)
      GO TO 500
C
C  spline parametrica natural o periodica
C
330   CALL CUBPAR(NP,1,T,X,X2,Y,Y2)
335   CALL COEF(NP,T,X,X2,A,B,C)
      CALL FVAL(NP,M,A,B,C,T,X,TM,XM)
      CALL FVAL1(NP,M,A,B,C,T,X,TM,XM1)
      CALL FVAL2(NP,M,A,B,C,T,X,TM,XM2)
      CALL COEF(NP,T,Y,Y2,A,B,C)
      TYPE *,'CUAL PARAMETRO PREFIERE(ANGULO=0,LONGITUD=1)?'
      READ(5,*) IP
      IF (IP.EQ.1) GOTO 339
      CALL FVAL(NP,M,A,B,C,T,Y,TM,YM)
      DO 337 I=1,NP1
      XM(I)=ATAN2(YM(I),XM(I))*180/3.1415926
      IF(XM(I).LT.0.0) XM(I)=XM(I)+360
337   CONTINUE
      IF(LTYPE.EQ.7) XM(NP1)=360.
339   CALL FVAL1(NP,M,A,B,C,T,Y,TM,YM1)
      CALL FVAL2(NP,M,A,B,C,T,Y,TM,YM)
      GO TO 345
340   CALL CYCLIC(NP,1,T,X,X2,Y,Y2)
      GO TO 335
345   IF(IP.EQ.1) GOTO 346
      CALL SCALE(XM,FLONX,NP1,1)
      GOTO 347
```

```
346     CALL SCALE(TM,FLONX,NP1,1)
347     CALL SCALE(XM1,FLONY,NP1,1)
        CALL SCALE(YM1,FLONY,NP1,1)
        IF (XM1(K1).LT.YM1(K1)) YM1(K1)=XM1(K1)
        IF (XM1(K2).GT.YM1(K2)) YM1(K2)=XM1(K2)
        XM1(K1)=YM1(K1)
        XM1(K2)=YM1(K2)
        IF(IP.EQ.0) GOTO 350
        CALL AXIS(0.,0.,5HEJE T,-5,FLONX,0.,TM(K1),TM(K2))
        GOTO 355
350     CALL AXIS(0.,0.,10HEJE ANGULO,-10,FLONX,0.,XM(K1),XM(K2))
355     CALL AXIS(0.,0.,11HX'(T),Y'(T),11,FLONY,90.,YM1(K1),YM1(K2))
        IF(IP.EQ.1) GOTO 360
        CALL LINE(XM,XM1,NP1,1,0,0)
        GOTO 370
360     CALL LINE(TM,XM1,NP1,1,0,0)
370     CALL NEWPEN(2)
        IF(IP.EQ.1) GOTO 380
        CALL LINE(XM,YM1,NP1,1,0,0)
        GOTO 385
380     CALL LINE(TM,YM1,NP1,1,0,0)
385     CALL PLOT(FLONX+2.0,0.,-3)
        CALL CURV(NP1,XM1,XM2,YM1,YM)
        CALL NEWPEN(1)
        CALL SCALE(YM,FLONY,NP1,1)
        IF(IP.EQ.1) GOTO 390
        CALL AXIS(0.,0.,10HEJE ANGULO,-10,FLONX,0.,XM(K1),XM(K2))
        GOTO 395
390     CALL AXIS(0.,0.,5HEJE T,-5,FLONX,0.,TM(K1),TM(K2))
395     CALL AXIS(0.0,0.,16HEJE CURVATURA(T),16,FLONY,90.0,YM(K1),YM(K2))
        IF(IP.EQ.1) GOTO 410
        CALL LINE(XM,YM,NP1,1,0,0)
        GOTO 500
410     CALL LINE(TM,YM,NP1,1,0,0)
500     CALL PLOT(0.,0.,999)
        CALL EXIT
        END
*
```

```
C
C   subrutina para calcular los valores de la curvatura
C
       SUBROUTINE CURV(NP1,XM1,XM2,YM1,YM)
       DIMENSION XM1(2),XM2(2),YM1(2),YM(2)
       DO 400 I =1,NP1
       FNUM=XM1(I)*YM(I)-XM2(I)*YM1(I)
       RAIZ=SQRT(XM1(I)**2 + YM1(I)**2)
       DEN = RAIZ*RAIZ*RAIZ
       YM(I) = FNUM/DEN
400    CONTINUE
       RETURN
       END
```

SUBRUTINA EVAL1:

```
    1
C
C   Subrutina para evaluar las primeras derivadas de una
C   curva plana utilizando las funciones spline
C
      SUBROUTINE EVAL1(N,M,A,B,C,X,Y,XM,YM)
      DIMENSION A(2),B(2),C(2),X(2),Y(2),XM(2),YM(2)
      DO 3 I=1,N-1
      DO 3 J=1,M
      K=(I-1)*M+J
      Z=XM(K)-X(I)
      YM(K) = 3*A(I)*Z**2+2*B(I)*Z+C(I)
3     CONTINUE
      XM(K+1)=X(N)
      YM(K+1)=3*A(N-1)*(X(N)-X(N-1))**2+2*B(N-1)*(X(N)-X(N-1))
     -+C(N-1)
      RETURN
      END
*
```

SUBRUTINA EVAL2:

```
$
C
C  Subrutina para evaluar las segundas derivadas de una curva
C
      SUBROUTINE EVAL2(N,M,A,B,C,X,Y,XM,YM)
      DIMENSION A(2),B(2),C(2),X(2),Y(2),XM(2),YM(2)
      DO 3 I=1,N-1
      DO 3 J=1,M
      K=(I-1)*M+J
      Z=XM(K)-X(I)
      YM(K) = 6*A(I)*Z + 2*B(I)
3     CONTINUE
      XM(K+1)=X(N)
      YM(K+1)=6*A(N-1)*(X(N)-X(N-1))+2*B(N-1)
      RETURN
      END
$
```

B.7 SUBRUTINA DE "SPLINES":

CUBIC1:

```
C
C  esta subrutina es para calcular los segundas derivadas
C  en los puntos de apoyo de una curva plana con spline natural
C
       SUBROUTINE CUBIC1(N,X,Y,Y2)
       DIMENSION X(2),Y(2),Y2(2)
       DIMENSION F(30),G(30)
       N1=N-1
       G(1)=0.
       F(1)=0.
       Y2(1)=0.
       Y2(N)=0.
       DO 2 K=1,N1
              J2=K+1
              H2=X(J2)-X(K)
              R2=(Y(J2)-Y(K))/H2
              IF (K.EQ.1) GOTO 1
              Z=1./(2.*(H1+H2)-H1*G(J1))
              G(K)=Z*H2
              H=6.*(R2-R1)
              IF (K.EQ.2) H=H-H1*Y2(1)
              IF (K.EQ.N1) H=H-H2*Y2(N)
              F(K)=Z*(H-H1*F(J1))
1             J1=K
              H1=H2
              R1=R2
2      CONTINUE
       Y2(N1)=F(N1)
       IF (N1.LE.2) RETURN
       N2=N1-1
       DO 3 J1=2,N2
              K=N-J1
              Y2(K)=F(K)-G(K)*Y2(K+1)
3      CONTINUE
       RETURN
       END
```

CUBICP:

```
C
C  esta subrutina calcula las segundas derivadas en los puntos
C  de apoyo de una curva plana con spline periodica
C
      SUBROUTINE CUBICP(N,X,Y,Y2)
      DIMENSION X(2),Y(2),Y2(2)
      DIMENSION F(30),H(30),G(30)
      N1=N-1
      N2=N-2
      J1=1
      G(1)=0.
      F(1)=0.
      H(1)=-1.
      H1=X(N)-X(N1)
      W=H1
      H2=X(N1)-X(N2)
      U=2.*(H1+H2)
      R1=(Y(N)-Y(N1))/H1
      R2=(Y(N1)-Y(N2))/H2
      V=6.*(R1-R2)
      DO 2 K=1,N2
            J2=K+1
            H2=X(J2)-X(K)
            R2=(Y(J2)-Y(K))/H2
            IF(K.EQ.1) GOTO 1
            U=U-W*H(J1)
            V=V-W*F(J1)
            W=-G(J1)*W
1           Z=1./(2.*(H1+H2)-H1*G(J1))
            G(K)=Z*H2
            H(K)=-Z*H(J1)*H1
            F(K)=Z*(6.*(R2-R1)-H1*F(J1))
            J1=K
            H1=H2
            R1=R2
2     CONTINUE
      H2=W+H1
      H1=(V-H2*F(N2))/(U-H2*(G(N2)+H(N2)))
      Y2(N1)=H1
      DO 3 J1=2,N1
            K=N-J1
            Y2(K)=F(K)-G(K)*Y2(K+1)-H(K)*H1
3     CONTINUE
      Y2(N)=Y2(1)
      RETURN
      END
```

CUBPAR:

```
C
C   esta subrutina calcula las segundas derivadas en los puntos
C   de apoyo de una curva plana con spline parametrica natural
C
      SUBROUTINE CUBPAR(N,NPAR,T,X,X2,Y,Y2)
      DIMENSION T(2),X(2),X2(2),Y(2),Y2(2)
      DIMENSION F(30),G(30)
      T(1)=0.
      J1=1
      DO 3 K=2,N
               U=X(K)-X(J1)
               V=Y(K)-Y(J1)
               IF(NPAR.NE.3) GOTO 1
               D=ABS(U)+ABS(V)
               GOTO 2
1              D=U*U+V*V
               IF(NPAR.EQ.1) D=SQRT(D)
2              T(K)=T(J1)+D
               J1=K
3     CONTINUF
      CALL CUBIC1(N,T,X,X2)
      CALL CUBIC1(N,T,Y,Y2)
      RETURN
      END
```

CYCLIC:

```
C
C   esta subrutina calcula las segundas derivadas en los puntos
C   de apoyo de una curva plana con spline parametrica periodica
C
      SUBROUTINE CYCLIC(N,NPAR,T,X,X2,Y,Y2)
      DIMENSION T(2),X(2),X2(2),Y(2),Y2(2)
      DIMENSION F(30),G(30),H(30)
      T(1)=0.
      J1=1
      DO 3 K=2,N
               U=X(K)-X(J1)
               V=Y(K)-Y(J1)
               IF(NPAR.NE.3) GOTO 1
               D=ABS(V)+ABS(U)
               GOTO 2
1              D=U*U+V*V
               IF(NPAR.EQ.1) D=SQRT(D)
2              T(K)=T(J1)+D
               J1=K
3     CONTINUF
      CALL CUBICP(N,T,X,X2)
      CALL CUBICP(N,T,Y,Y2)
      RETURN
      END
```

EVAL:

```
C
C       Esta subrutina evalua las funciones spline
C       de cualquier tipo en los puntos de interpo
C       lacion con un intervalo suficientemente
C       pequeno.
C
        SUBROUTINE EVAL(N,M,A,B,C,X,Y,XM,YM)
        DIMENSION A(2),B(2),C(2),X(2),Y(2),XM(2),YM(2)
        DO 3 I=1,N-1
        DX=(X(I+1)-X(I))/M
        DO 3 J=1,M
        K=(I-1)*M+J
        XM(K)=X(I)+DX*(J-1)
        Z=XM(K)-X(I)
        YM(K)=A(I)*Z**3+B(I)*Z**2+C(I)*Z+Y(I)
3       CONTINUE
        XM(K+1)=X(N)
        YM(K+1)=Y(N)
        RETURN
        END
```

COEF:

```
C
C       Esta subrutina es para calcular los coeficientes de una
C       funcion spline de cualquier tipo.
C
        SUBROUTINE COEF(N,X,Y,Y2,A,B,C)
        DIMENSION X(2),Y(2),Y2(2),A(2),B(2),C(2)
        DO 10 K=1,N-1
        A(K)=(Y2(K+1)-Y2(K))/(6.*(X(K+1)-X(K)))
        B(K)=0.5*Y2(K)
        C(K)=(Y(K+1)-Y(K))/(X(K+1)-X(K))-
     *  (X(K+1)-X(K))*(Y2(K+1)+2.*Y2(K))/6.
10      CONTINUE
        RETURN
        END
```