C O M P U T A C I O N G R A F I C A:

METODOS DE APROXIMACION DE CURVAS Y SUPERFICIES

Por: Daniel Formica

Claudio Delrieux

Daniel Masni

Laboratorio de Sistemas Digitales
Depto. Ingeniería Eléctrica
Universidad Nacional del Sur
ARGENTINA

Palabras Clave: Paquetes Gráficos Interactivos; Algoritmos de Aproximación e interpolación de Curvas y Superficies; Modelado por Puntos de Control; Representación Gráfica Tridimensional.

I - INTRODUCCION

En las representaciones de gráficos por computadoras, gran cantidad de casos involucran figuras cuya definición no es necesariamente funcional (no podemos evaluar las posiciones del gráfico en base a variables), sino más bien proviene de un conjunto de datos encontrados experimentalmente (representación de objetos reales, funciones no conocidas pero medibles, etc.). En dichos casos se puede modelar la figura a representar según dos técnicas opuestas: se reconstruye la figura obteniendo una gran base de datos experimentalmente de modo que los programas clásicos graficadores de funciones nos sigan siendo de utilidad, o bien utilizamos algún método en nuestro programa que nos permita reconstruir la figura aproximandola a partir de pocos puntos experimentales.

Este último método, llamado método de los PUNTOS DE CONTROL, es unánimemente utilizado para el modelado de curvas y superficies no funcionales, ya que la otra alternativa, además de requerir bases de datos de tamaños imprácticos, produce en general problemas de actualización y de resultados gráficos que la hacen inelastica y ensorrosa.

Dada la necesidad de implementar una rutina de aproximación de curvas y superficies para un paquete gráfico interactivo tridimensional, se investigaron, ensayaron y compararon cuatro técnicas, basadas todas en el modelado por puntos de control, que fueron los métodos de Hermite, Bezier, B-Spline y Beta-Spline.

Los requerimientos para poder incluir dicha rutina en el paquete gráfico existían la capacidad de producir gráficos tridimensionales interactivos de gran calidad y realismo, con la mayor rapidez y versatilidad posible, sin depender del Hardware utilizado. El criterio de selección estuvo orientado por las siguientes características [2]:

1. INDEPENDENCIA RESPECTO DE LOS EJES: La forma resultante de un objeto no debe cambiar si los puntos de control se miden con distintos sistemas de coordenadas (invariancia a rotación, traslación y cambios de escala)

2. VALORES MULTIPLES: Debe ser permitida la representación de funciones multivaluadas respecto de cualquiera de los ejes.

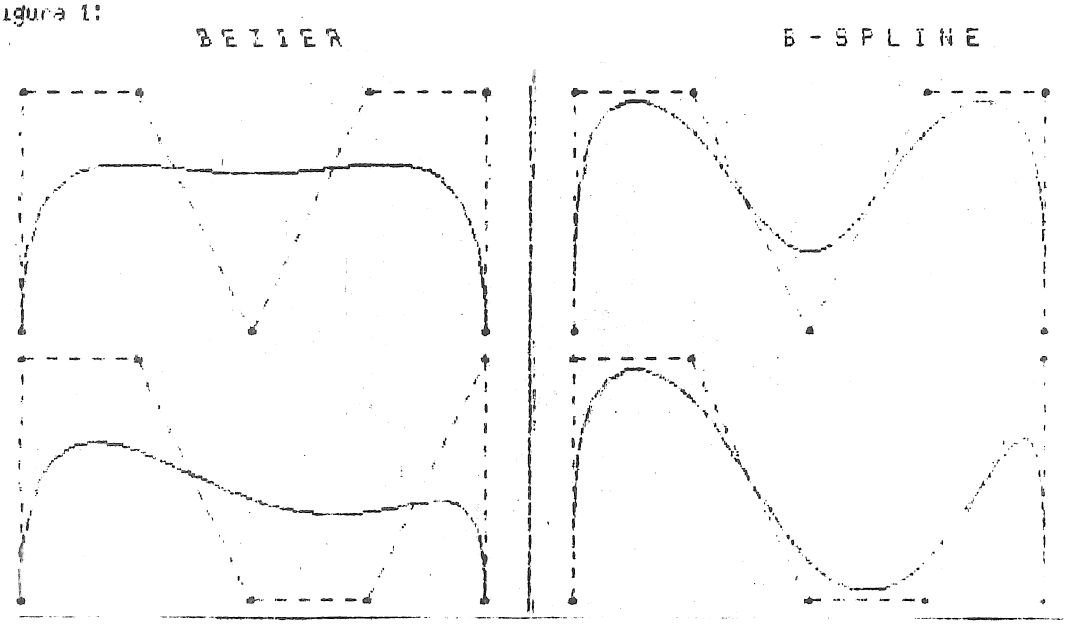
3. PROPIEDAD DE DISMINUCION DE VARIACIONES: La representación debe suavizar y nunca amplificar las pequeñas irregularidades delineadas por los puntos de control.

4. CONTROL LOCAL O CONTROL GLOBAL: Si al alterar un punto de control se modifica solo un entorno de la curva original, denominaremos CONTROL LOCAL a la característica del método. En oposición un algoritmo con CONTROL GLOBAL modificará toda la curva al alterarse un punto de control cualquiera (ver Figura 1). Debe ser conocida la característica de control del método utilizado para poder seleccionarlo respecto de la aplicación.

5. VERSALTILIDAD: La propiedad del método de poder modificar el resultado gráfico a voluntad sin introducir cambios en la base de datos.

6. ORDEN DE CONTINUIDAD: Sea f(x) considerada como la derivada de orden cero de si misma. El orden de continuidad "i" de la curva es aquella f'(x) del grado máximo que sea derivable. El orden de continuidad de la curva provista por el método debe ser conocido y en lo posible podrá modificarse.

Figura 1:



II - HERMITE

El desarrollo de **Hermite** [3] parte de las ecuaciones polinómicas resultantes de suponer continuidad de orden cero y de orden uno. El método consiste en recorrer los puntos de control tomándolos de a pares junto con una estimación de las derivadas de la curva en dichos puntos. Como resultado el algoritmo entrega una curva que pasa por los puntos de control con las derivadas especificadas interpolando las posiciones y derivadas intermedias, cuyos valores quedan determinados por los polinomios mencionados.

Las condiciones establecidas son:

(1) $U(0) = P_{0,u}$
$U(1) = P_{1,u}$

(2) $U'(0) = P_{0,u}$
$U'(1) = P_{1,u}$

donde: U denota cada uno de los ejes coordenados (x,y,z)
U(i) es el polinomio que realiza la interpolación entre las coordenadas $P_{0,u}$ y $P_{1,u}$.

La condición (1) asegura continuidad de orden cero y la condición (2) asegura continuidad de orden uno. Planteando dichas condiciones la ecuación de una curva que interpola un conjunto de N puntos es:

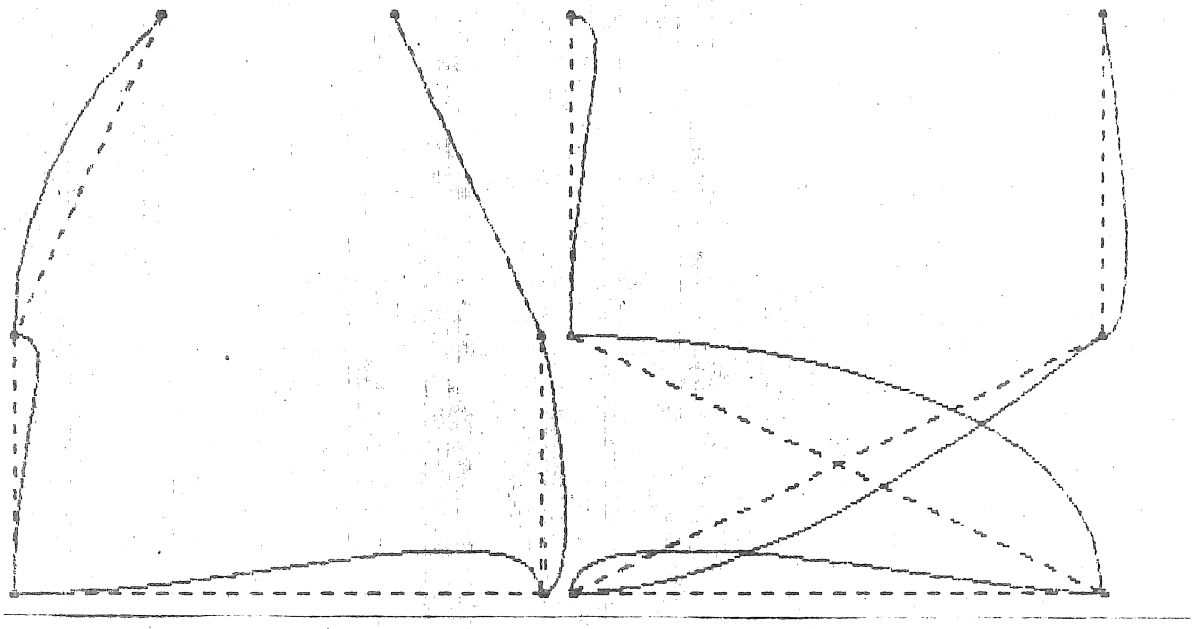
$$P(u) = \sum_{i=1}^{N} \sum_{j=0}^{1} \sum_{k=0}^{1} s_{j\,k}(u) \cdot P^{j}_{i-1+k}$$

donde $s_{jk}(u)$ son los polinomios cúbicos de Hermite:

$$s_{oo}(u) = 2.u^3 - 3.u^2 + 1$$
$$s_{o1}(u) = -2.u^3 + 3.u^2$$
$$s_{1o}(u) = u^3 - 2.u^2 + u$$
$$s_{11}(u) = u^3 - u^2$$

En la figura 2 vemos el comportamiento de este método (graficado en trazo lleno) para dos poligonos de control dados (graficados en líneas punteadas).

Figura 2: Hermite



III - BEZIER

El algoritmo de Bezier [1] construye polinomios de orden de continuidad "n", siendo "n" el número de puntos de contol. Utilizando una función estadistica denominada "Familia de Polinomios de Berstein" [4] obtenidos a partir de la distribución binomial construye las llamadas FUNCIONES DE MEZCLA de modo que la posición ocupada por un punto determinado de la curva es el resultado de "mezclar" la posición de TODOS los puntos de control en proporciones indicadas por las funciones de mezcla. Por ello su caracteristica es de ser control global. La ecuación de una curva representada de este modo es:

$$P(u) = \sum_{i=0}^{n} B_{i,n}(u).P_i$$

donde u es el parámetro tal que:

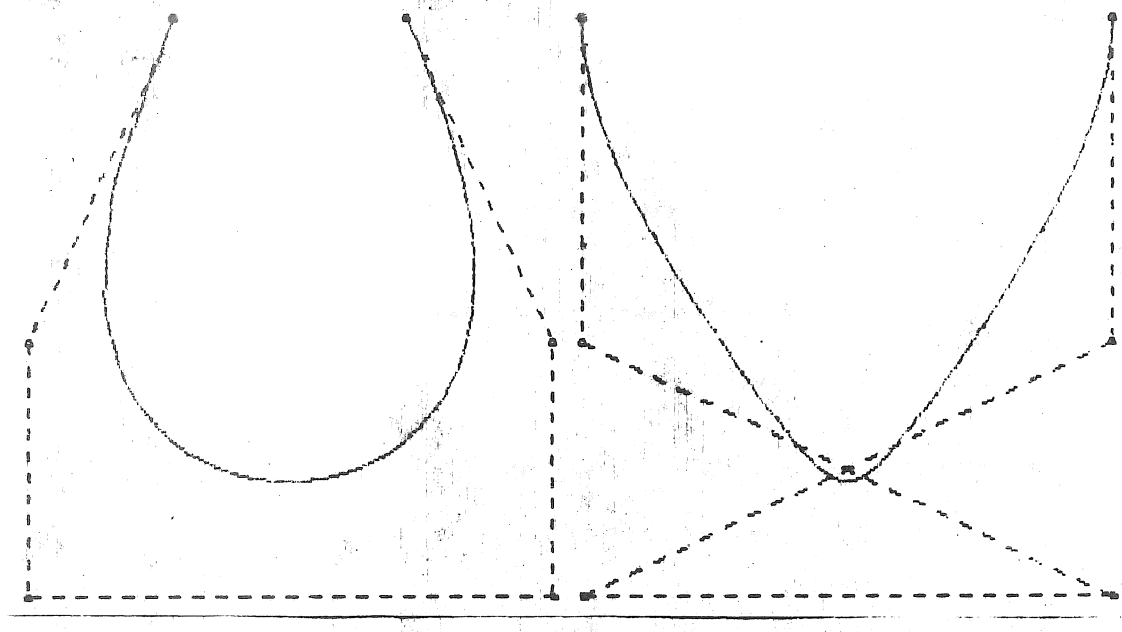
$$0 \leq u \leq 1$$

y la Función de Mezcla es:

$$B_{i,n}(u) = \frac{n!}{i! \cdot (n-i)!} \cdot u^i \cdot (1-u)^{n-i}$$

En la figura 3 se observa el resultado obtenido con los polígonos de control anteriores aproximados por este método.

Figura 3: Bezier



IV - B-SPLINE

El B-Spline [1], [2] fue el próximo método investigado. Las curvas Spline se obtienen con una filosofía similar a las curvas de Hermite, pero con ciertas variaciones en el planteo de las condiciones de borde. Un "Spline" se define como la suma de un conjunto de polinomios que describen la curva entre los puntos 'i-1' e 'i' cumpliendo con las restricciones:

(A) $P_i(u) = P_{i+1}(u)$

$$u_i \leq u \leq u_{i+1}$$

$$i = 0, 1, 2, \ldots, k-1$$

(B) $P_i^j(u_i) = P_{i+1}^j(u_i)$

$$J = 0, 1, 2, \ldots, r-1$$

$$i = 0, 1, 2, \ldots, k-1$$

donde k: Número de puntos de control
J: Orden de continuidad del polinomio
r: Número de restricciones

La condición (A) impone que el polinomio 'i' tiene valor cero excepto en el intervalo entre los puntos de control 'i-1' e 'i'. La condición (B) establece que en la unión entre los polinomios se conserva un orden de continuidad 'J'.

Con dichas restricciones la ecuación de una curva que pasa por N puntos de control es:

$$P(u) = \sum_{i=0}^{n} N_{i,k}(u) \cdot P_i$$

Donde $N_{i,k}(u)$ es una función de mezcla para el orden de continuidad k que se expresa en forma recursiva:

$$N_{i,k}(u) = \frac{(u - P_i) \cdot N_{i,K-1}(u)}{P_{i+K-1} + P_i} + \frac{(P_{i+K} - u) \cdot N_{i+1,K-1}(u)}{P_{i+K} - P_{i+1}}$$

donde $N_{i,1} = 1 \quad$ si $P_i \leq u \leq P_{i+1}$

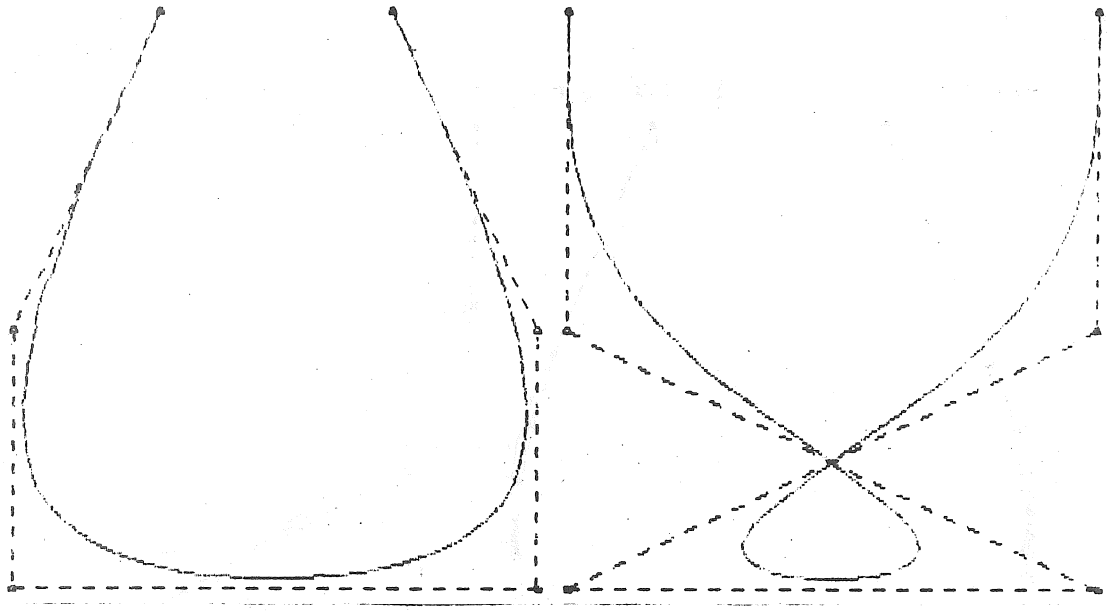
$= 0 \quad$ de otro modo

Estas funciones de mezcla tienen la misma forma, independientemente del valor de i. Por lo tanto, para que la curva pase por el primer y por el último punto de control, se repiten k veces dichos puntos asegurando continuidad de orden cero en los mismos.

Debido a que el ojo humano no detecta fácilmente órdenes de continuidad mayores que dos, se eligió este valor para la implementación del método. Con dicho orden de continuidad solo es necesario evaluar cuatro puntos de control para encontrar la posición de cualquier punto de la curva. Esto indica que, a diferencia del método de Bezier, su característica es de control local (al variar un punto de control cambia solamente un entorno de la curva).

En la figura 4 se observa el resultado obtenido al graficar los polígonos de control anteriores con este ultimo método y k=4 (orden de continuidad dos), siendo observable cómo la curva se aproxima mejor al polígono de control

Figura 4: B - S p l i n e



V - BETA-SPLINE

El Beta-Spline fue el último método investigado [6], [8], [9]. Las curvas Beta-Spline se obtienen, al igual que en el B-Spline, como suma de polinomios, pero planteando las siguientes condiciones de borde [9] en la intersección de cada intervalo:

(D) $$P_{i-1}(u_i) = P_i(u_i)$$

(E) $$B_1 \cdot P'_{i-1}(u_i) = P'_i(u_i)$$

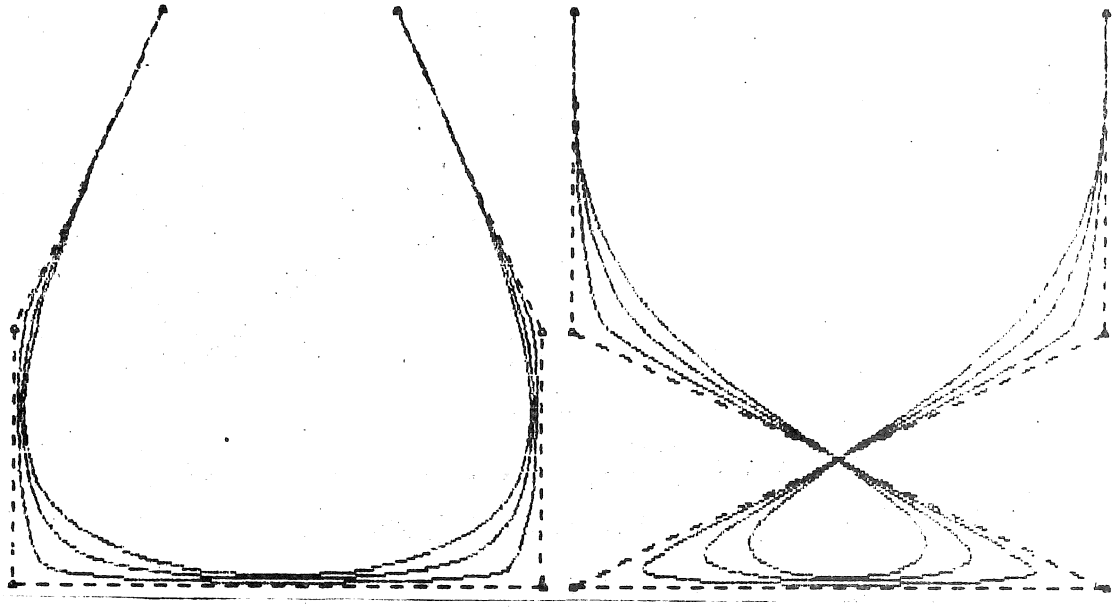
(F) $$B_1^2 \cdot P''_{i-1}(u_i) + B_2 \cdot P'_{i-1}(u_i) = P''_i(u_i)$$

La condición (D) establece continuidad en la posición. La condición (E) permite vectores tangentes colineales pero con una discontinuidad en su magnitud (B1) llamado 'Sesgo' (Bias). Físicamente ésto representa un cambio instantáneo de la velocidad de la curva en la misma dirección en la unión entre dos segmentos. La condición (F) preserva la continuidad del vector curvatura en la unión entre dos segmentos, pero con el agregado de un factor B2 llamado 'Tensión' que introduce una componente extra de aceleración en la dirección del vector tangente. Hay también un término de aceleración colineal al vector tangente inevitable debido al B1.

Como puede verse, de esta definición se obtienen dos grados más de libertad cuyo efecto es el siguiente: Si el sesgo (bias) es uno y la tensión es cero, las ecuaciones (D), (E) y (F) se igualan a las (A) y (B) y el método se comporta como el B-Spline. Luego el

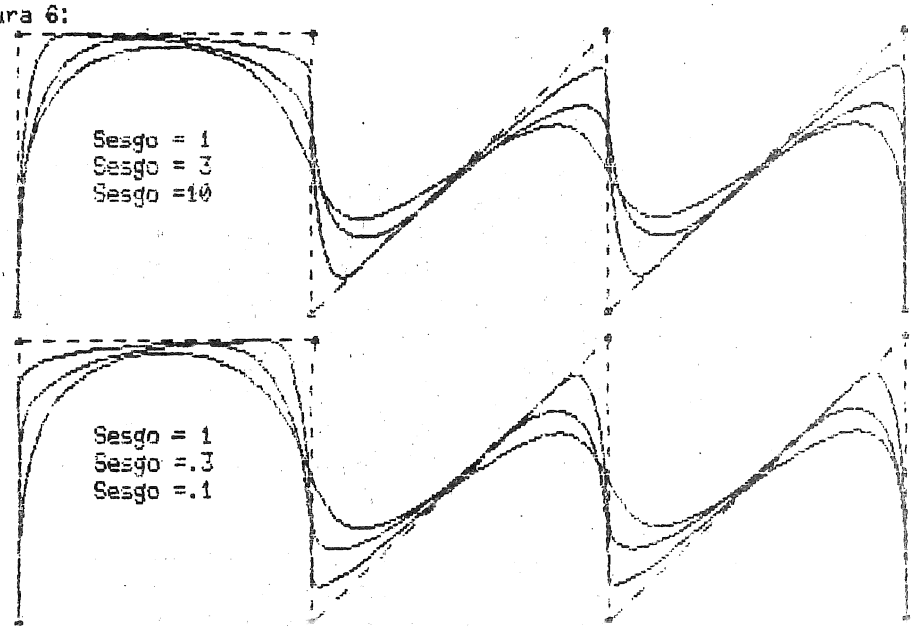
B-Spline es un caso particular del Beta-Spline. En la figura 5 podemos apreciar el comportamiento del Beta-Spline con los polígonos de control anteriores y para tres valores de sesgo y tensión distintos.

Figura 5: Beta-Spline



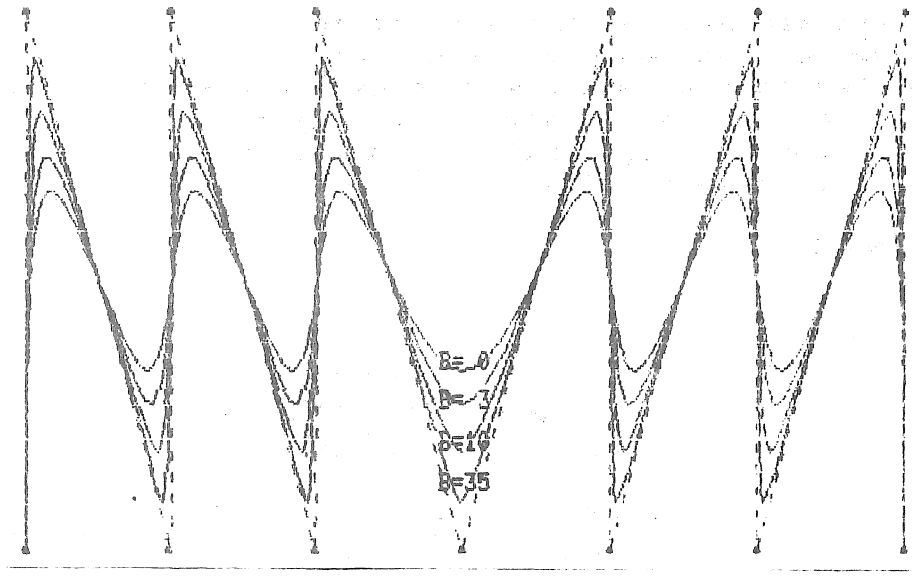
Es de notar cómo los resultados producidos son aún mejores que aquellos obtenidos con el B-Spline. Una explicación más detallada del funcionamiento de los parámetros muestra lo siguiente: en la figura 6 observamos el efecto de variar el sesgo (B1) manteniendo la tensión (B2) en cero.

Figura 6:



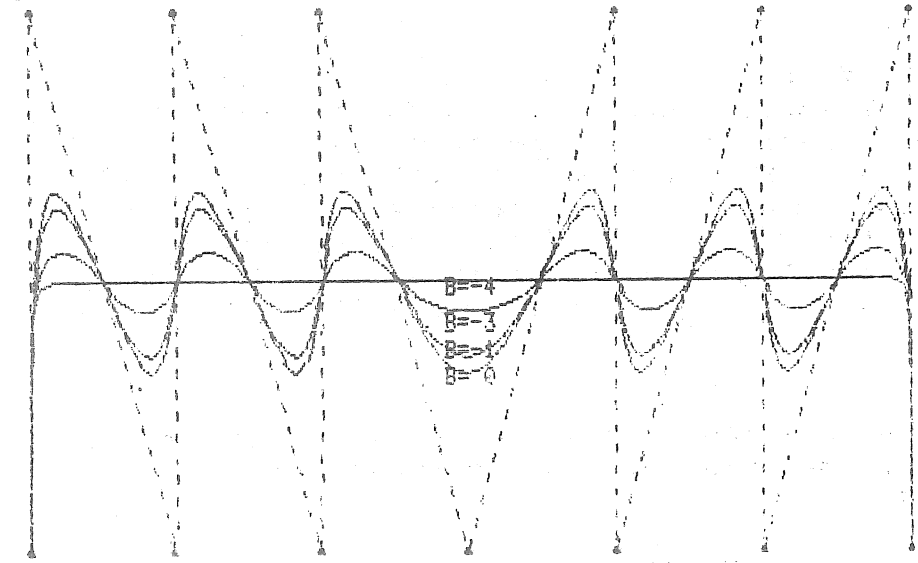
Para valores mayores que uno, la curva se desplaza hacia la izquierda aproximándose más y más al polígono de control. Para los valores recíprocos respecto de la unidad la curva se desplaza hacia la derecha con un comportamiento similar. En la figura 7 se muestra cómo se comporta la curva para distintos valores positivos de la tensión pero manteniendo el sesgo en su valor unitario.

Figura 7:



A medida que aumenta dicha tensión la curva se junta más y más con el polígono de control. En la figura 8 vemos el resultado de aplicar tensiones negativas. Se puede apreciar cómo la curva va separándose del polígono de control hasta prácticamente convertirse en una línea recta.

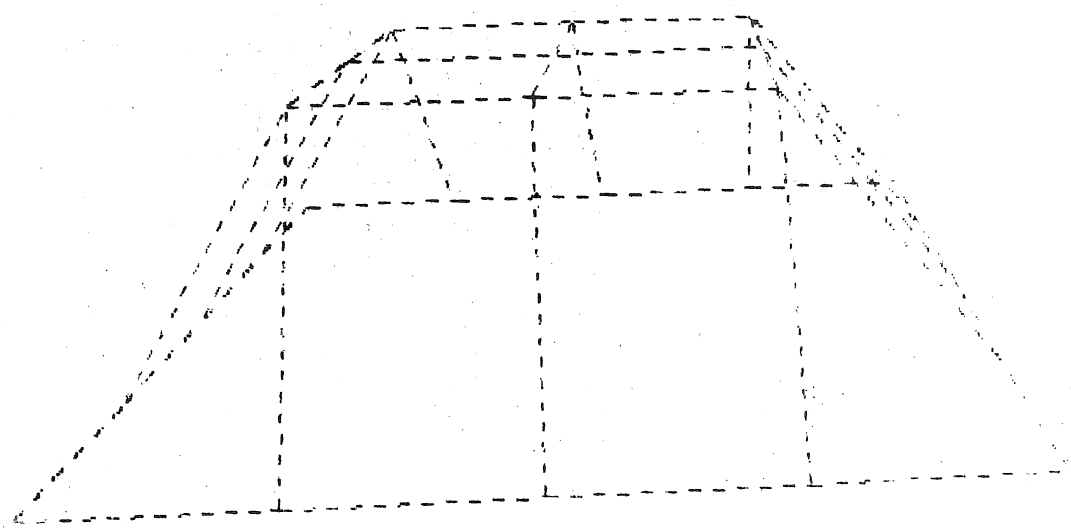
Figura 8:



VI - EXTENSION DEL METODO A LA APROXIMACION DE SUPERFICIES

Para representar una superficie a través de puntos de control se trabajó en un modelo en el cual se han obtenido las alturas Z además de las coordenadas X e Y de cada punto de control. La unión de dichos puntos de control con líneas rectas a X constante e Y constante conforma el polígono de control de la superficie a interpolar, como se muestra en la figura 9.

Figura 9:



La función que aproxima dicha superficie debe ser una función de dos variables que para cualquier coordenada constante traze una curva que cumpla con las propiedades de aproximación de curvas ya explicadas. El método matemático que permite obtener dichas propiedades a partir de los métodos de aproximación de curvas se denomina PRODUCTO CARTESIANO de las funciones. Utilizando dicha herramienta matemática es posible extender la aplicación de los métodos ya vistos al caso general de los puntos de control en el espacio.

La ecuación de una superficie aproximada por el producto cartesiano de uno de los métodos, por ejemplo el Bezier, queda, entonces:

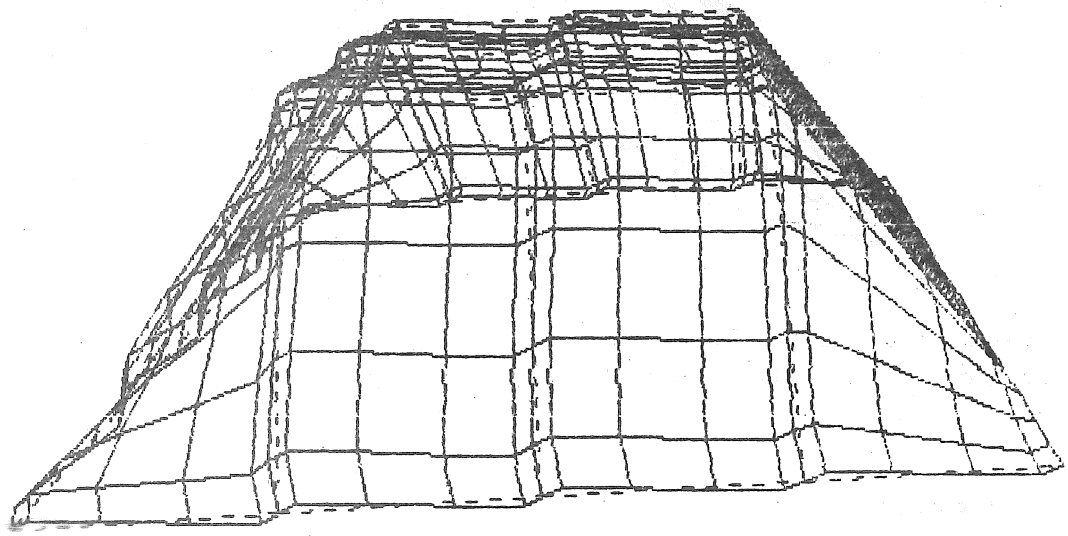
$$P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,n}(u) \cdot B_{j,m}(v) \cdot P_{i,j}$$

donde N: Cantidad de puntos de control en la coordenada X.
M: Cantidad de puntos de control en la coordenada Y.

u: Parámetro de la superficie, recorrida según el eje X.
v: Parámetro de la superficie, recorrida según el eje Y.
$P_{i,j}$: Valor de la coordenadas (x,y,z) del punto de control i,j.
$B_{i,n}(u)$: Función de mezcla del parámetro u.
$B_{j,m}(v)$: Función de mezcla del parámetro v.
(tal como fueron definidas en el capítulo III)

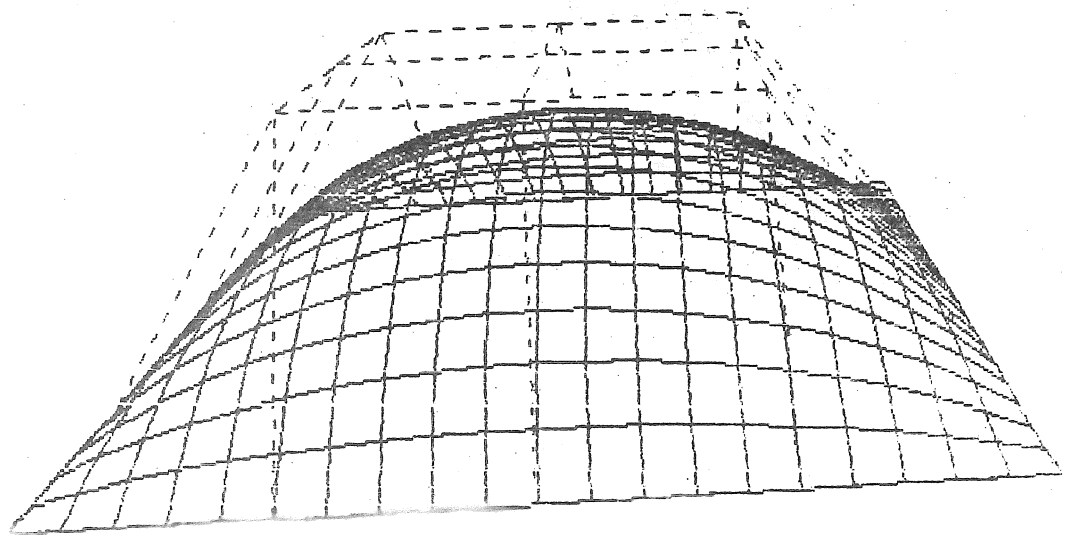
El tratamiento para los otros métodos es enteramente similar. En la fisura 10 se observa el comportamiento del algoritmo Hermite para el grafo de control de la fisura 9.

Figura 10:



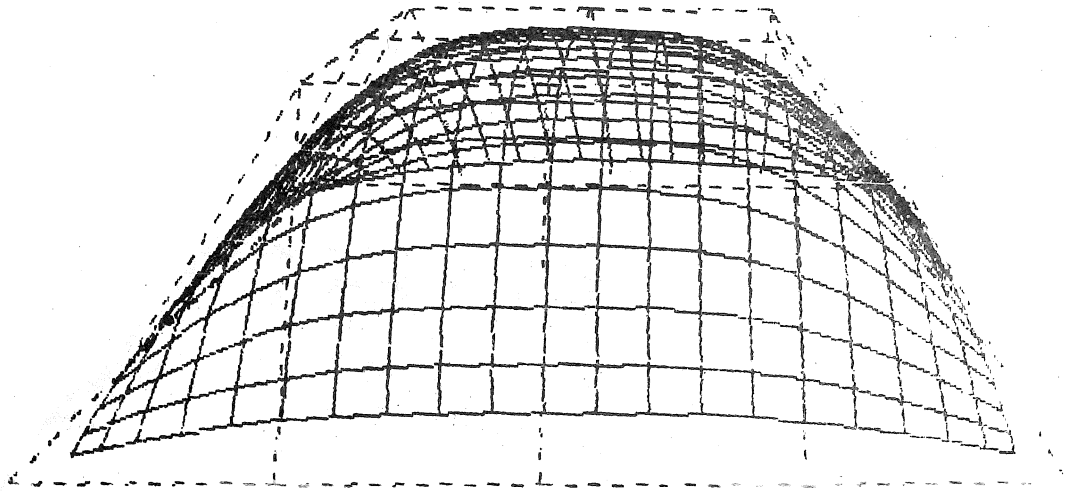
En la fisura 11 vemos el mismo grafo aproximado por Bezier.

Figura 11:



La aproximación realizada por el método B-Spline se observa en la fisura 12.

Figura 12:



Por último se ve el resultado producido para tres valores distintos de sesgo y tensión con el algoritmo Beta-Spline en las figuras 13 y 14

Figura 13:

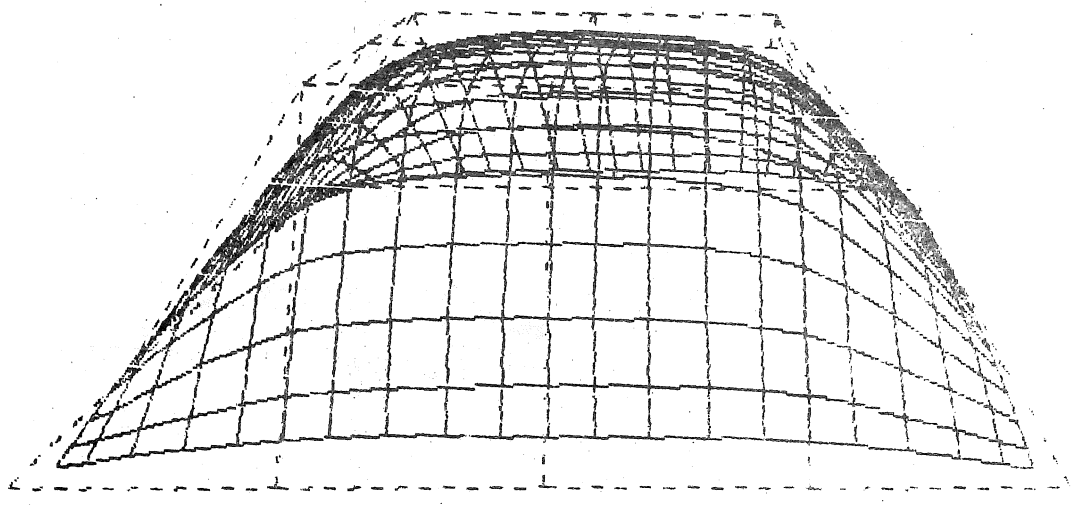
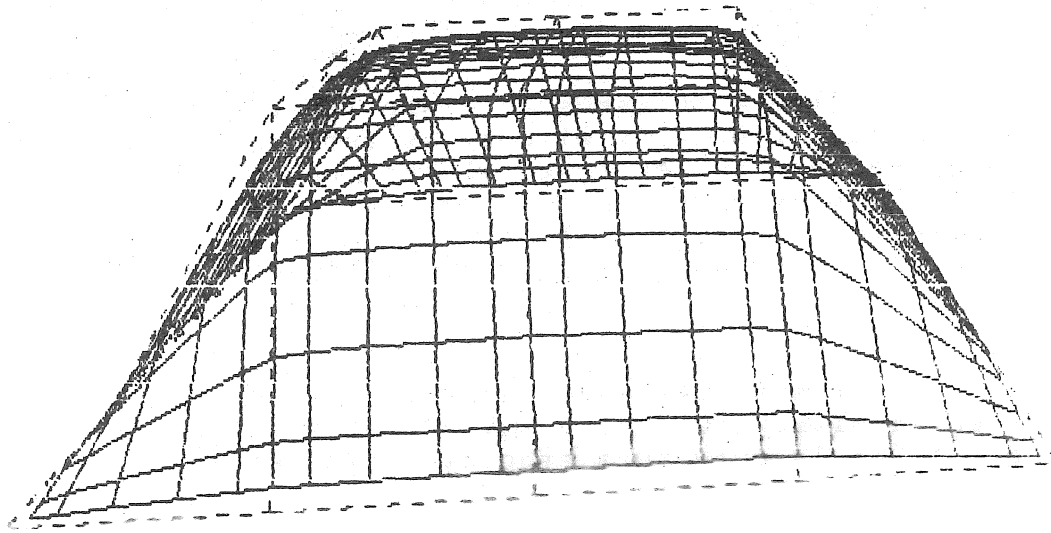


Figura 14:



VII - COMPARACION DE LOS METODOS Y CONCLUSIONES

De la evaluación del comportamiento de los métodos analizados y sus respuestas al conjunto de polígonos de control en el plano y en el espacio surgen las siguientes consideraciones:

De los gráficos de la figura 2 producidos por el método de Hermite se ve cómo, si bien se respeta la forma aproximada del polígono de control, existe sin embargo una discrepancia que acentúa exageradamente las discontinuidades del mismo, siendo el resultado general bastante poco exacto.

En la figura 3 el resultado producido con el algoritmo de Bezier para los mismos polígonos de control muestra que el comportamiento en general es más satisfactorio, aunque se ve cómo para el segundo polígono el método es incapaz de seguir la 'vuelta' dada por el mismo.

Un mejor comportamiento en este caso se observa en la figura 4 en las curvas producidas con el B-Spline. El seguimiento de los polígonos de control es mucho mejor, aún en la 'vuelta' dada por el segundo de los mismos.

El resultado es muy superior con el algoritmo Beta-Spline mostrado en la figura 8. Se produjeron tres de las curvas que es posible obtener con el método gracias a sus grados de libertad adicionales (sesgo y tensión). Es notable cómo gracias a ello podemos hacer parecerse las curvas trazadas al polígono de control tanto como se desee.

En la figura 10 vemos en trazo discontinuo el polígono de control de la figura 9 el cual es aproximado con el método de los polinomios de Hermite (en trazo lleno) con la extensión para figuras en tres dimensiones explicado en VI. Se observa que el comportamiento es similar al de la figura 2 en el sentido de presentar 'curvaturas' ajenas al polígono de control que lo alejan del realismo.

En la figura 11, en el gráfico producido por el método de Bezier en el mismo caso, el resultado es más armónico que el anterior aunque tal vez la curvatura producida siga escasamente al polígono de control.

En la figura 12, en cambio, observamos que esta curvatura es seguida con más fidelidad por el algoritmo B-Spline. El mejor comportamiento se observa al notar que la curva alcanza a tocar el polígono de control en la parte superior del mismo, cosa que no sucedía con el método de Bezier.

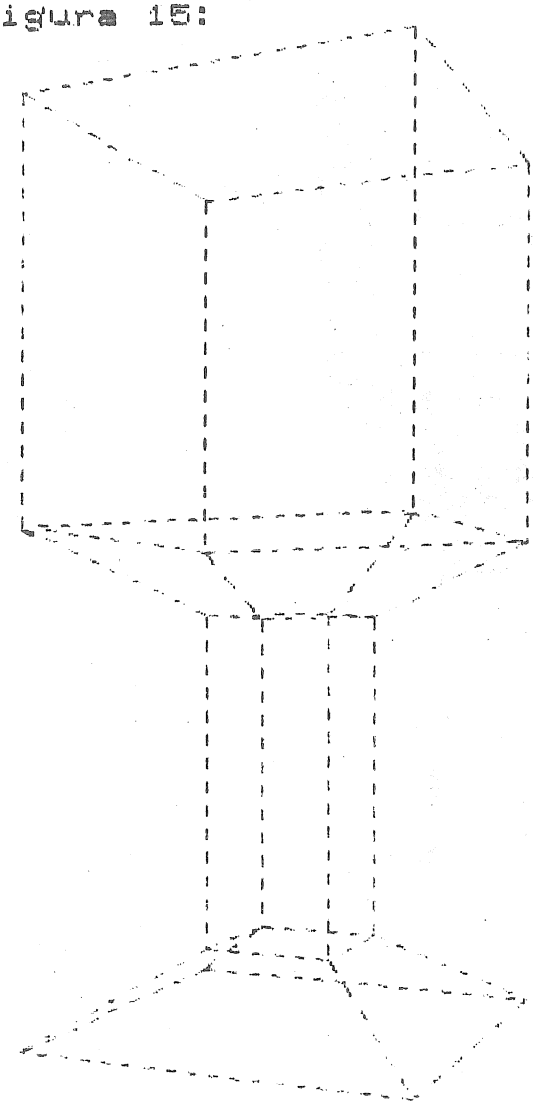
En la figura 13 vemos una de las posibilidades que nos brinda el Beta-Spline: se tensionó el gráfico de modo que se ve cómo se ha acercado más aun al polígono de control que en el B-Spline. Por último en la figura 14 vemos otro caso donde se llevó el gráfico más cerca aún del polígono de control.

De todos los resultados mostrados surge claramente que el método Beta-Spline es al mismo tiempo el más versatil y el que produce los resultados gráficos más realistas. Su rapidez es comparable a la de los otros algoritmos siendo superior la calidad gráfica producida, independientemente del dispositivo utilizado y aún en terminales gráficos de baja resolución. Por dichas razones se adoptó este método como el utilizado para realizar la parte siguiente del presente trabajo.

VIII - TRABAJO DE APLICACION

Como trabajo de aplicación se elisió el modelo de una copa cuyo polígono de control, mostrado en la figura 15, está construído con una base de datos de tan solo 20 puntos.

Figura 15:



En la figura 16 se observa la copa dibujada con línea llena a partir de los puntos de control (dibujados con línea de trazos). La mencionada versatilidad del método se aprecia entre las diferencias entre esta figura y la figura 17 en donde se han variado localmente el sesgo y la tensión en distintos lugares de la copa, produciéndose formatos totalmente distintos sin alterar la base de datos.

Figura 16:

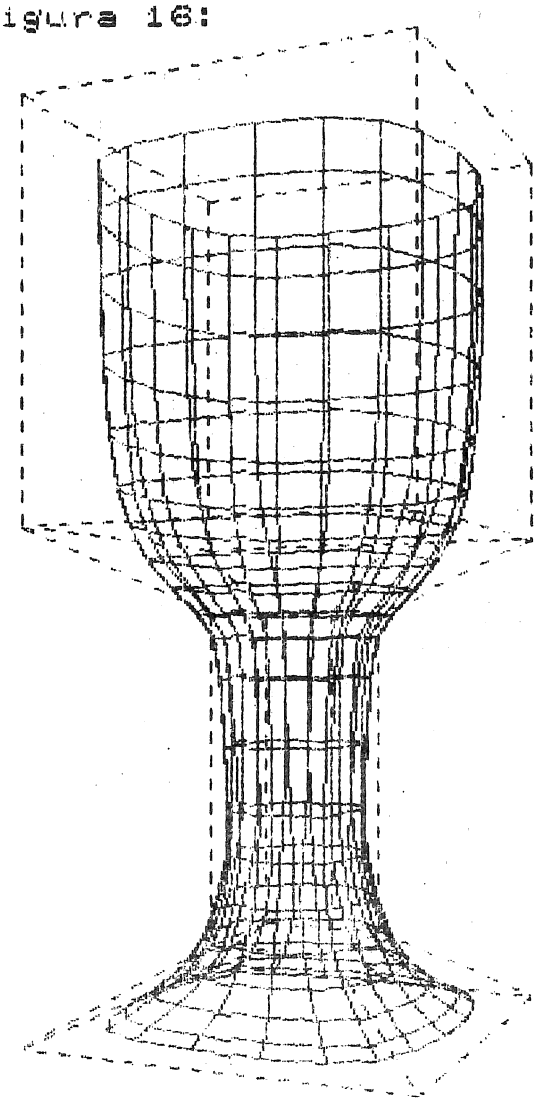
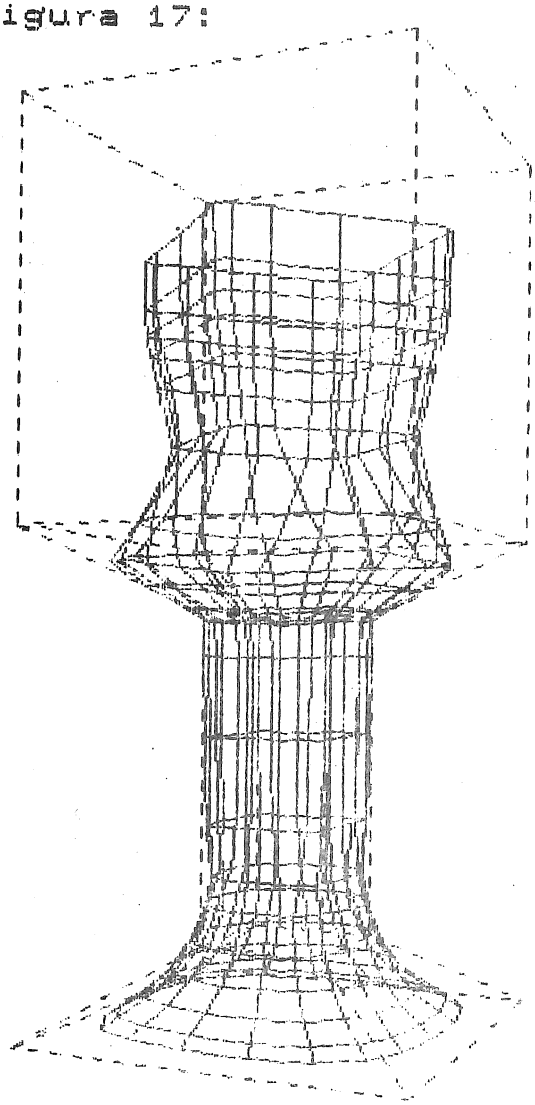


Figura 17:



Para producir resultados de mayor realismo se buscó implementar una rutina de eliminación de líneas ocultas. Dicho algoritmo produce resultados de calidad aceptable y actualmente es objeto de estudio e investigación en este Laboratorio. Con el agregado de dicho algoritmo se produjeron las figuras 18 y 19, donde se aprecian sus resultados combinados con transformaciones tridimensionales.

Figura 18:

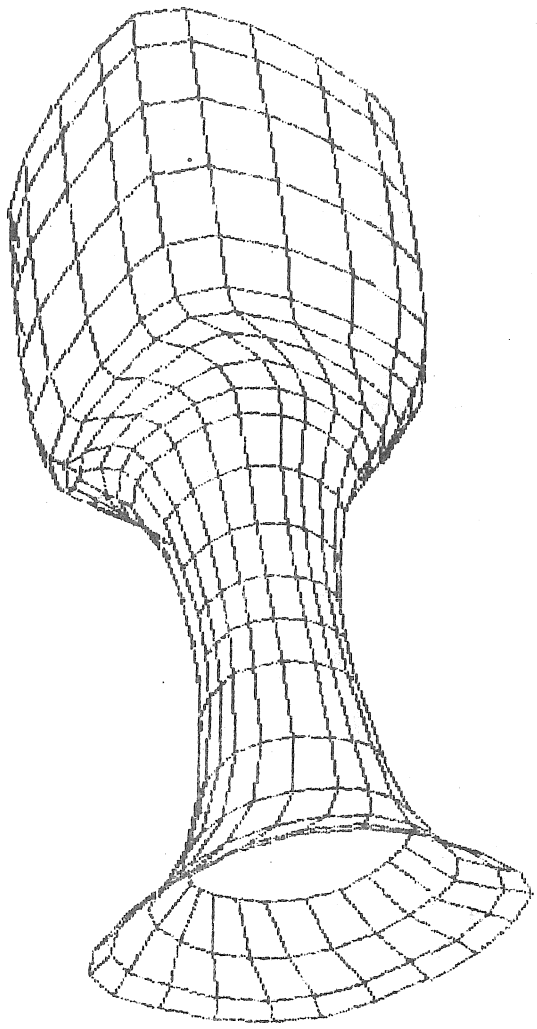
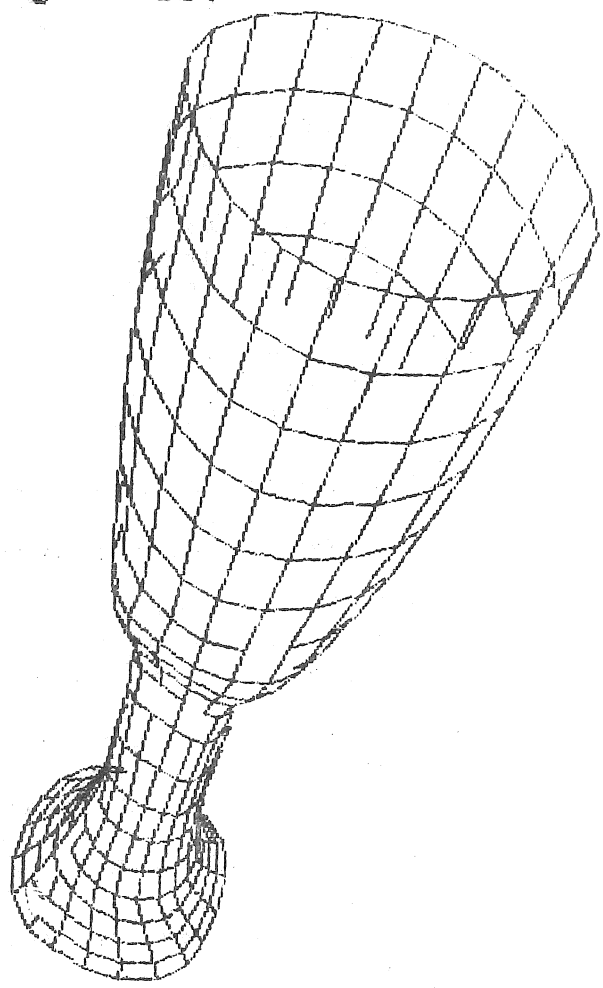


Figura 19:



IX - BIBLIOGRAFIA

[1]: Newmann - Sproull
Principles of Interactive Computer Graphics
Mc Graw-Hill, 1979

[2]: Giloi, W.
Interactive Computer Graphics
Prentice-Hall, 1978

[3]: Foley-Van Dam
Fundamentals of Interactive Computer Graphics
Addison-Wesley, 1982

[4]: Algorithms for Graphics & Image Processing
Theo Pavlidis
Computer Science Press, 1982

[5]: A two-space solution to the hidden-line problem
Thomas Wright
IEEE Transactions on Computers, Jan 1973

[6]: A description and evaluation of various 3D models
Brian Barsky
IEEE Computer Graphics, Jan 1981

[7]: Perspective representation of functions of two variable
Kubert, Szabo & Giulieri
Journal of the ACM vol.15 n.2, Apr 1968

[8]: Local control of Bias and Tension in Beta-Splines
J. Barsky-J.Beatty
Computer Graphics of the ACM vol. 17 n. 3, Jul 1983

[9]: Urn Models & Beta-Splines
Ronald Goldmen
IEEE Computer Graphics, Feb 1986

[10]: Implemntación de Rutinas Gráficas Tridimensionales
Claudio Delrieux
XV JAIIO
Sociedad Argentina de Informática e Investigación Operativa
Universidad Nacional del Sur, 1985