

Bordando Imágenes

Josefina Alvarez
Department of Mathematics
New Mexico State University
Las Cruces, NM 88003-8001, USA
Email: jalvarez@nmsu.edu

La palabra fotografía proviene del griego *phōs* (luz) y *graphos* (escrito). Literalmente quiere decir “escrito por la luz”. Este nombre tiene sentido cuando pensamos en hacer una fotografía con una cámara usual. Pero ¿cuál es el proceso que permite mostrar una imagen en la pantalla de un ordenador? Puesto que un ordenador sólo entiende de números, la imagen que vemos en la pantalla debe de ser el resultado de la interacción entre luz y números. Con el fin de motivar lo más básico de esta interacción vamos a usar la idea del bordado a punto cruz, una técnica que existe hace ya muchos siglos.

En esta técnica se comienza con una página cuadriculada donde se tiene el patrón del diseño que se quiere bordar. Cada uno de los colores a usar en el bordado se indica con un trazo diferente, el cual se corresponde con el nombre del color y con un número. En realidad, los fabricantes de hilos de bordar sólo usan números para describir los colores; los nombres se los dan los diseñadores y las editoriales para facilitar la interpretación del patrón.

Por ejemplo, la Figura 1(b) muestra el patrón del símbolo japonés que se ve en la Figura 1(a), el cual representa la palabra *este*. El tamaño del patrón se da en puntadas. Por ejemplo, el patrón en la Figura 1(b) tiene 37 puntadas de largo, que es la medida horizontal, por 36 puntadas de ancho, que es la medida vertical. Cada puntada corresponde a un cuadradito en la rejilla. En la Figura 1(c) puede verse la clave para reproducir el diseño. Lo único que no está especificado es el tamaño de la cuadrícula.



Figura 1(a)

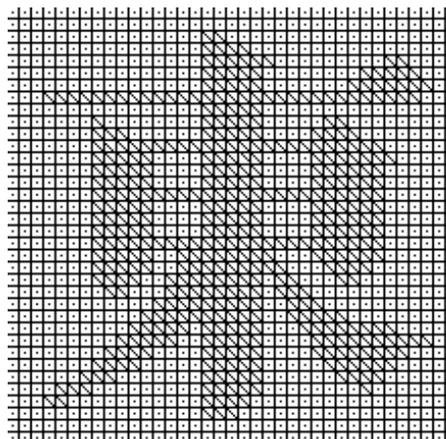


Figura 1(b)

37 puntos por 36 puntos		
\	666	Rojo Brillante
•	742	Naranja Claro

Figura 1(c)

Todo esto está muy bien, pero se estarán preguntando: ¿qué tiene que ver el punto cruz con un ordenador? En realidad tiene mucho que ver. Si miramos de cerca la pantalla del monitor veremos una rejilla, y a medida que apretamos las teclas, las letras o los símbolos que aparecen en la pantalla realmente se asemejan a un bordado a punto cruz. El número de cuadrados minúsculos que forman la rejilla es la resolución de la pantalla. Esta resolución se expresa en una unidad llamada *pixel*.

El pixel no es realmente una unidad de medida como el centímetro o el metro, sino es una manera de describir una rejilla. Por ejemplo, la resolución de una cierta pantalla podría ser 640×480 , lo cual querría decir que esta rejilla tiene 640 pixels, o segmentos, en la dirección horizontal y 480 pixels, o segmentos, en la dirección vertical. ¡También podría interpretarse como que la rejilla tiene 307.200 pixels, o cuadrados! ¿Qué pasa con lo que vemos en la pantalla si cambiamos la resolución? Con el fin de contestar a esta pregunta, volvamos a usar la metáfora del punto cruz. En la Figura 2(a) podemos ver el símbolo “bordado”. La irregularidad de los bordes indica que el bordado es sólo una aproximación del símbolo. Si incrementamos el número de cuadrados en la rejilla, el bordado será más pequeño y la irregularidad de los bordes se notará menos. La Figura 2(b) muestra este efecto, que se obtuvo dividiendo en cuatro cada uno de los cuadrados de la rejilla original.

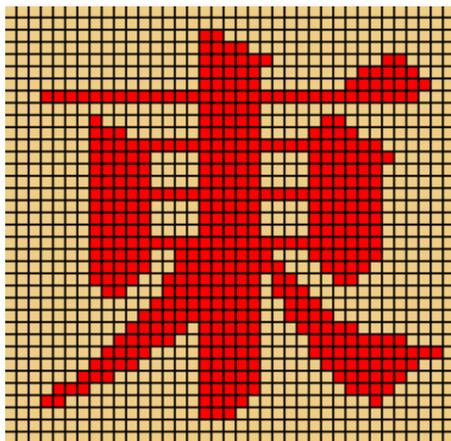


Figura 2(a)

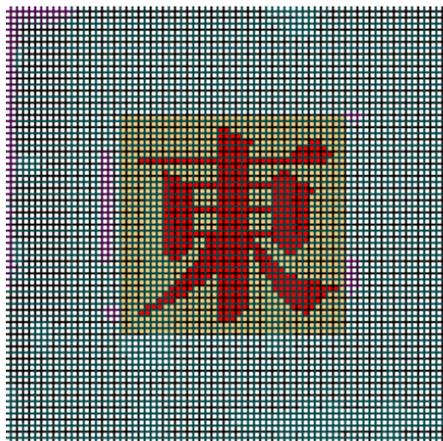


Figura 2(b)

Voyendo a la pantalla del monitor, podemos deducir que un incremento en la resolución producirá una imagen que muestra mejor los detalles más finos. El precio que pagamos es el tener una imagen más pequeña, que quizá sea difícil de ver cuando la resolución es muy grande.

Como ya hemos dicho, los colores en nuestro bordado son codificados usando números. Por ejemplo, en la Figura 3(a) hemos destacado una porción del símbolo en rojo brillante con su fondo naranja claro. Esta porción del bordado está representada en la Figura 3(b) por medio de $8 \times 8 = 64$ números, que son los números asignados a los colores en la Figura 1(c).

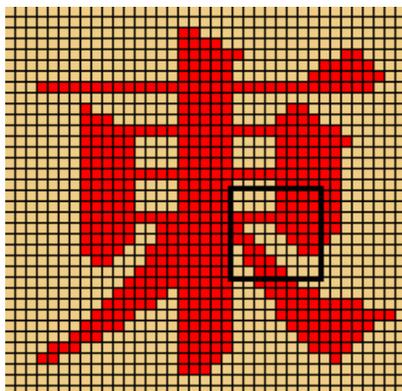


Figura 3(a)

742	742	742	742	666	666	666	666
742	742	742	742	666	666	666	666
666	666	666	666	666	666	666	666
666	742	742	742	666	666	666	666
666	666	742	742	742	666	666	666
742	666	666	742	742	742	666	666
742	666	666	666	742	742	742	742
742	742	666	666	666	742	742	742

Figura 3(b)

Necesitaríamos un total de $37 \times 36 = 1.332$ números para representar por completo el símbolo y su fondo. Estos números son en realidad todo lo que necesitamos para reproducir los diseños de las Figuras 2(a) y 2(b). Pensándolo mejor, podríamos haber usado directamente números en la Figura 1(b). Matemáticamente hablando, estas formaciones de números como la que se ve en la Figura 3(b) se llaman *matrices*. Lo habitual es indicirlas dando el número de filas por el número de columnas, o en el lenguaje del punto cruz, como el número de puntadas verticales por el número de puntadas horizontales. Pero ¡un momento!, esto es precisamente lo contrario de lo que tenemos en la Figura 1(c), donde dice que el diseño tiene 37 puntadas de largo por 36 puntadas de ancho. En efecto, si aplicamos esta notación a la matriz de la Figura 3(b), el largo es el número de columnas, mientras que el ancho es el número de filas. ¿Qué pasa aquí? Lo que estamos observando es que bordadores y matemáticos no indican sus formaciones de números de la misma manera. ¿Es esto muy importante? ¡Claro que no! Las dos maneras funcionan perfectamente bien, cada una en su propio terreno. Ahora que estamos hablando de esto, ¿por qué será que los matemáticos cuentan primero las filas y luego las columnas? Quizá será porque las matrices son usadas para representar sistemas de ecuaciones. Por ejemplo, la matriz

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & 3 \end{bmatrix}$$

contiene toda la información que se necesita para reconstruir la parte destacada en el sistema

$$\begin{cases} 2x + y + z = 1 \\ x - y + 3z = 2 \end{cases}$$

de dos ecuaciones con tres incógnitas. ¿Ven como decimos dos ecuaciones, o filas en la matriz, con tres incógnitas, o columnas en la matriz? Sin embargo, aún dentro de las matemáticas, éste no es siempre el caso. En efecto, si alguien nos pidiera que describiéramos el rectángulo de la Figura 4,

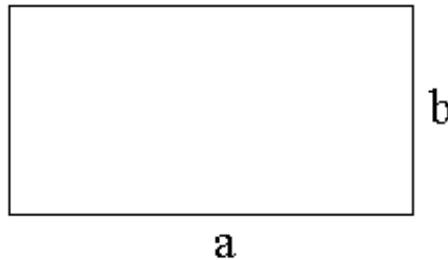


Figura 4

apuesto a que entonces pensaríamos como los bordadores, dando primero el largo a y luego el ancho b . Por otra parte, cuando tuve que fijar los lados de este rectángulo, el sistema operativo de mi ordenador me pidió que seleccionara primero el ancho a y luego la altura b . Así vemos que raramente los nombres y las notaciones son universales. Su elección depende del contexto en que se usen, con lo cual no hay ningún problema, siempre y cuando tengamos claro ese contexto.

Volviendo a nuestro bordado, quizá se estén preguntado para qué insistir en usar todos esos números. Imagínense que tuvieran que describir el símbolo de la Figura 1(a) por teléfono. Sería bastante difícil el describir la forma del símbolo correctamente, ¿verdad? Sin embargo resulta trivial, en principio, el describirlo al menos aproximadamente, cuando la forma del símbolo ha sido convertida en un manojito de números. Digo con precaución en principio, porque si el manojito de números fuera enormemente grande, quizá tomaría mucho tiempo el transmitirlo, incluso para un ordenador. Pero no nos vamos a preocupar aquí por este detalle, aunque ya podemos ver que nuestro ejemplo sencillísimo ha resultado en la necesidad de usar muchos números. Y aún no hemos visto toda la historia. Eso de describir cada color con un solo número, no es realmente cómo funcionan las cosas.

La idea del color es en efecto una idea muy compleja, bastante difícil de describir en palabras o de cuantificar. Un método que los pintores usan, es el describir el color usando tres atributos: matiz, saturación y valor. Básicamente, el matiz de un color indica dónde ubicarlo, entre los rojos, o los verdes, o los amarillos, etcétera. La saturación indica cuán diluido es el matiz. El tercer atributo, el valor, describe si el color es oscuro o claro.

Otra manera en la cual podemos imaginarnos un color, es como una mezcla de colores básicos. El sistema llamado *RGB*, por los nombres ingleses de

los colores rojo, verde y azul, usa distintas proporciones de estos tres colores para obtener casi todos los colores. En un modelo comúnmente usado en ordenadores, la cantidad de cada uno de los colores rojo, verde y azul se indica con un número de 0 a 255. ¡Por lo tanto, el número total de colores disponibles es $255 \times 255 \times 255 = 16.777.216!$ Lo más probable es que el sistema operativo de cualquier ordenador tenga una manera de manejar el color, la cual nos permite experimentar con esta idea de mezclar colores, y describir los atributos resultantes usando números. Para mantener las cosas bajo control, evitaremos cualquier referencia a los procesos físicos y químicos por los cuales los números se convierten en colores visibles en la pantalla. Los libros *Colour: Why the world isn't gray* ([11]) y *El arte del color* ([7]), analizan la idea del color desde muchos puntos de vista y son una excelente introducción a estos temas.

Después de todos estos comentarios, debería estar claro que nuestro intento inicial de asignar un solo número a cada color fue una simplificación excesiva. En efecto, cada pixel en la pantalla deberá ser “pintado” con tres números. Podemos pensar que cada una de estas “manos de pintura” es una función que asigna a cada punto, o pixel, en la pantalla, un número de 0 a 255. Así nos encontramos con números mucho más grandes que los que hemos visto hasta ahora. En efecto, una resolución de 640×480 es en realidad bastante mala. Mejor sería el tener una resolución de, digamos, 1024×768 . En una pantalla con esta resolución, necesitaremos millones de números para representar los colores.

Quizá se estén preguntando de dónde viene esto de usar una escala de 0 a 255 en el sistema *RGB*. Aquí va una explicación. La verdad es que los ordenadores son muy remilgados. No sólo piden comer nada más que números, sino que también exigen que los números sean preparados de una manera especial. Un ordenador del tipo que usamos en la actualidad, sólo entiende la acción de apagar o encender una corriente eléctrica. Por ello, tal ordenador sólo se alimenta de números representados con dos símbolos, o *bits*, 0 y 1, que corresponden al apagado y encendido de la corriente, respectivamente. La representación en este sistema, llamado *binario*, se hace escribiendo los números en potencias de 2, en lugar de las habituales potencias de 10.

Por ejemplo, $37 = 32 + 4 + 1 = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$. Es decir que la representación del número 37 en potencias de 2, o en base 2, es 100101. Podríamos haber escrito 0100101, 00100101, 000100101, etcétera. Lo importante es que 2^5 es la mayor potencia de 2 que es menor o igual que 37.

Puesto que $2^7 = 128$ y $2^8 = 256$, podemos ver que $255 = 128 + 127 = 128 + 64 + 63 = \dots$, por lo cual $255 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$. Es decir que la representación de 255 en base 2 está dada por la sucesión 11111111, que consta de ocho unos. Como consecuencia de esto, las representaciones binarias de todos los números de 0 a 255, son precisamente todas las sucesiones de ocho ceros o unos desde 00000000 a 11111111. Una sucesión de ocho ceros o unos, o un *byte*, es una unidad de almacenaje comúnmente usada en ordenadores. Esto explica el uso de la escala de 0 a 255. Entonces, en el sistema *RGB*, cada color está representado por tres bytes o, en otras palabras, por 24 bits. No siempre un monitor tendrá la capacidad de asignar los 24 bits a cada pixel. En ese caso se usará una

aproximación al verdadero color, usando un número menor de bits. El “pintado” de la rejilla se hace usualmente bit por bit. Podemos pensar que cada pixel recibe a lo más 24 manos de pintura, y que cada mano tiene un bit de “espesor”. Mirando a la rejilla entera, cada una de las manos es en realidad una matriz de ceros y unos.

Pensándolo bien, esto de reemplazar los nombres de los colores con números suena bastante prosaico. Pero como una consolación podemos recordar las palabras del poeta y matemático aplicado Philip Holmes cuando dice que “nuestros modelos son nuestras metáforas” ([13]).

Espero que ahora esté claro el papel central que la noción de rejilla juega tanto en el bordado a punto cruz como en los ordenadores. Este concepto aparece también en otros contextos, por ejemplo en el trabajo del pintor Chuck Close, quien torna fotografías en pinturas enrejilladas gigantescas. Close explica sus retratos enrejillados con estas palabras ([6], p. 10): “... La idea de pintar una cara enorme me abruma. No estoy seguro de cómo puedo producir una nariz a esa escala. Pero si divido la imagen en pequeños trozos, entonces puedo transformar cada decisión que tengo que tomar en muchas, pequeñas, decisiones...”. Chuck Close emplea frecuentemente lienzos enormes, que llena de trazos coloridos. Para enfatizar algún rasgo especial, Close usa a veces rejillas oblicuas o circulares, quizá con pixels de tamaños diferentes. En el libro que acabo de citar, escrito por J. Greenberg y S. Jordan ([6]), se pueden ver reproducciones de muchas de las pinturas de Close, con explicaciones de las técnicas usadas por el artista.

Volviendo al símbolo japonés bordado a punto cruz, una vez que hemos aproximado su forma y su color, estamos listos para emprender el siguiente paso, que es el suavizar los bordes. ¿Cómo se puede hacer esto? En el bordado a punto cruz, se usan puntadas lineales a todo lo largo del contorno. En diseño y en las artes visuales, reglas y compases fueron en la antigüedad las únicas herramientas disponibles, produciendo resultados no muy satisfactorios con segmentos y arcos de circunferencia. Posteriormente, fue posible reproducir curvas más complicadas usando plantillas de madera o de plástico llamadas *curvas francesas*, las cuales aún se pueden comprar en cualquier papelería. En la Figura 5(a) se muestra una de estas plantillas, mientras que en la Figura 5(b) se puede ver el resultado de usar la plantilla en nuestro símbolo japonés.

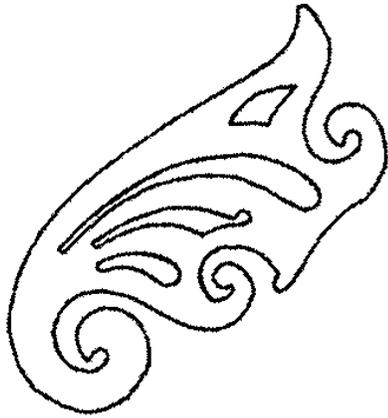


Figura 5(a)

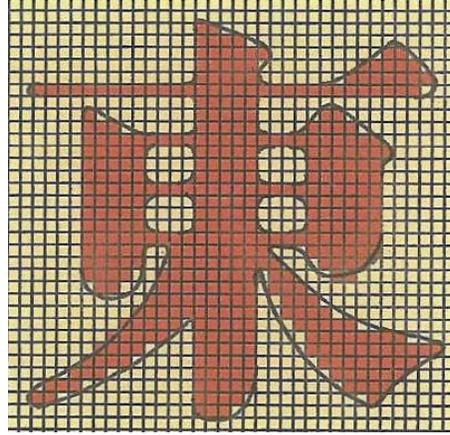


Figura 5(b)

En los años 1930 y 1940, el auge del diseño industrial produjo la necesidad de reproducir curvas y superficies muy complicadas. Hacia fines de la década del 50 y comienzos de la del 60, tuvo lugar un avance monumental cuando el matemático Paul de Faget de Casteljaou (nacido en 1933) y el ingeniero mecánico y eléctrico Pierre Bézier (1910-1999), produjeron independientemente una familia de curvas muy versátiles, generadas matemáticamente. En ese entonces de Casteljaou trabajaba para la fábrica de automóviles Citroën y Bézier trabajaba para Renault. Ya en 1966 Bézier estaba publicando sus resultados mientras trabajaba para Renault, y es por ese motivo que las curvas se conocen como curvas de Bézier. Estas curvas se convirtieron rápidamente en una herramienta fundamental en el diseño asistido por ordenador, o CAGD (*computer aided geometric design*), como se empezó a conocer hacia 1975. El trabajo de de Casteljaou para Citroën fue citado públicamente por primera vez en 1971.

Ya hemos mencionado que las curvas de Bézier se obtienen matemáticamente. En realidad, uno de sus mayores atractivos, como veremos enseguida, es la simplicidad y la belleza de las matemáticas que las generan. Para comenzar, pensemos en una curva como el camino recorrido por un punto que se mueve sobre un plano, de acuerdo a una función $t \rightarrow P(t)$ que manda valores del parámetro t a puntos $P(t)$ en el plano. Como podemos ver en la Figura 6, la función $P(t)$ realmente consiste en dos funciones, $x(t)$ e $y(t)$, las cuales dan las coordenadas horizontal y vertical de cada punto $P(t)$.

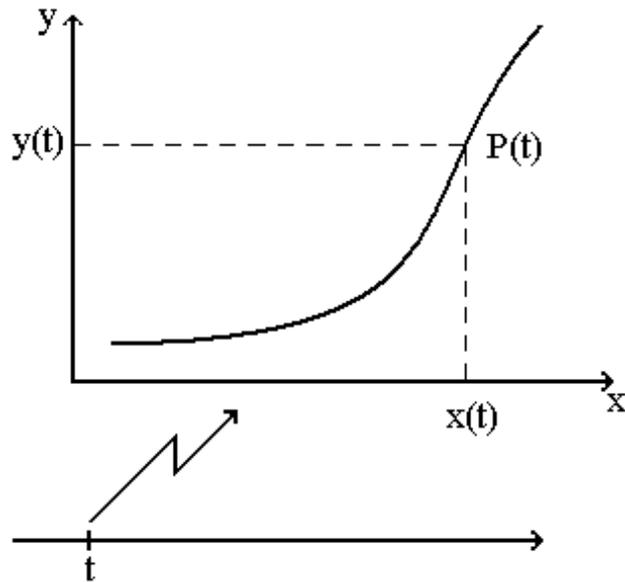


Figura 6

La función $t \rightarrow P(t)$ es lo que se llama una *ecuación paramétrica de la curva*, que entonces se dice que está dada paramétricamente. Por supuesto, el que una curva sea o no una curva de Bézier va a depender de cómo se elige la función $P(t)$. Para explicar esta elección en los términos más sencillos posibles, comenzamos por escribir la identidad

$$1 = 1 - t + t,$$

con la cual representamos al número 1 como la suma de dos polinomios de primer grado en la variable t . Dados dos puntos cualesquiera P_0 y P_1 en el plano, vamos a usar estos dos polinomios para armar la función

$$P(t) = (1 - t)P_0 + tP_1,$$

que nos da nuestro primer ejemplo de curvas de Bézier, las curvas de Bézier de primer orden. Observemos que $P(0) = P_0$ y $P(1) = P_1$, y que cuando el parámetro t se mueve de 0 a 1, la función $P(t)$ genera el segmento de extremos P_0 y P_1 , como podemos ver en la Figura 7.

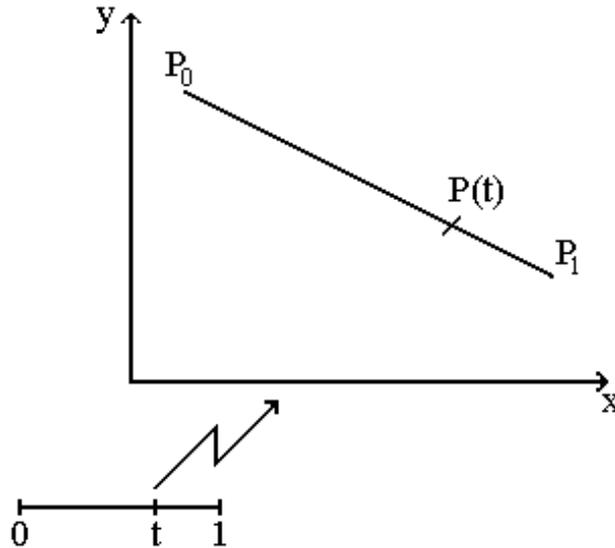


Figura 7

Si indicamos $P_0 = (x_0, y_0)$ y $P_1 = (x_1, y_1)$, entonces el punto $P(t)$ tendrá las coordenadas $x(t) = (1-t)x_0 + tx_1$ y $y(t) = (1-t)y_0 + ty_1$. Apuesto a que están pensando que estas curvas de Bézier no parecen por ahora muy interesantes. Quizá tengan razón, pero yo sigo en lo mío, escribiendo la segunda identidad

$$1 = 1^2 = ((1-t) + t)^2 = (1-t)^2 + 2(1-t)t + t^2.$$

Lo que estamos haciendo aquí es representar al número 1 como la suma de tres polinomios de grado dos en la variable t . Si ahora elegimos tres puntos P_0, P_1, P_2 , podemos usar estos tres polinomios para escribir la función

$$P(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, \quad (1)$$

la cual nos da las curvas de Bézier de segundo orden. Esta vez $P(0) = P_0$ y $P(1) = P_2$, o en otras palabras, P_0 y P_2 son los extremos de la curva. ¿Cuál es entonces el rol que juega P_1 ? Este punto P_1 se llama *control* y para entender qué efecto tiene en la forma de la curva, podemos examinar los dos ejemplos de la Figura 8. En el ejemplo de la izquierda, la curva se dirige desde $P_0 = (0, 0)$ en la dirección de la recta que pasa por los puntos P_0 y $P_1 = (0, 1)$, y se acerca al punto $P_2 = (1, 0)$ siguiendo la dirección de la recta que pasa por P_1 y P_2 . En el ejemplo de la derecha, la curva se dirige desde $P_0 = (0, 0)$ en la dirección de la recta que pasa por los puntos P_0 y $P_1 = (1, 0)$, y se acerca al punto $P_2 = (0, 1)$ siguiendo la dirección de la recta que pasa por P_1 y P_2 .

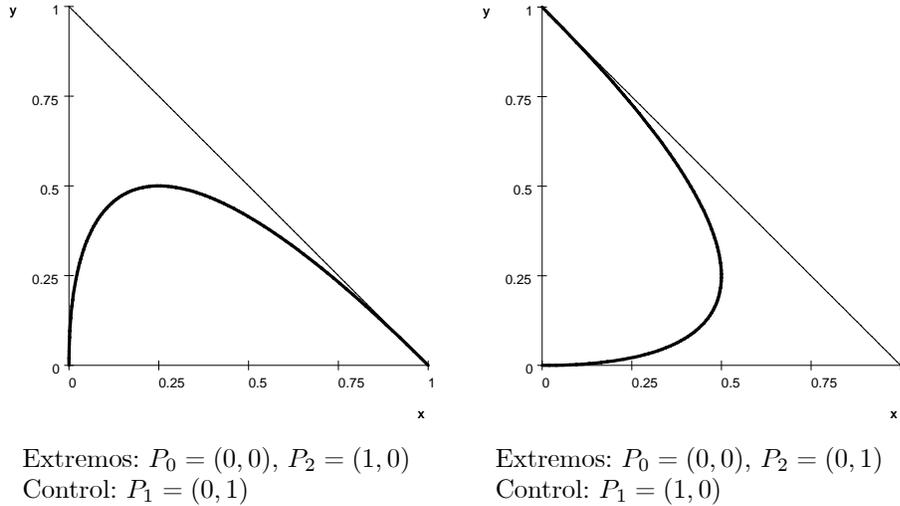


Figura 8

Por lo tanto, uno puede pensar que P_0 y P_2 son las “anclas” de la curva, mientras que P_1 es el “conductor”. En términos matemáticos, los ejemplos de la Figura 8 sugieren que la recta que pasa por los puntos P_0 y P_1 es tangente a la curva en el punto P_0 , mientras que la recta que pasa por los puntos P_1 y P_2 es tangente a la curva en el punto P_2 .

Continuando con nuestra construcción de las curvas de Bézier, vamos a usar ahora cuatro puntos y cuatro polinomios de tercer grado. El procedimiento es el mismo. Comenzamos con la identidad

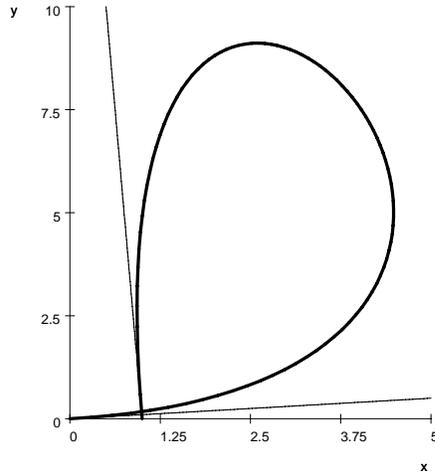
$$1 = 1^3 = ((1-t) + t)^3 = (1-t)^3 + 3(1-t)^2 t + 3(1-t)t^2 + t^3.$$

A continuación elegimos cuatro puntos, P_0, P_1, P_2 y P_3 , y escribimos la función

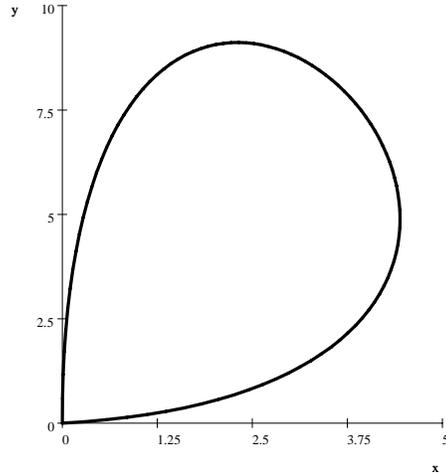
$$P(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \quad (2)$$

la cual nos da las curvas de Bézier de tercer orden, que son las curvas de Bézier más usadas. Podemos comprobar fácilmente que $P(0) = P_0$ y $P(1) = P_3$, es decir que los puntos P_0 y P_3 son los extremos de la curva. Los puntos P_1 y P_2 se llaman los controles de la curva. Podemos ver en los ejemplos de la Figura 9 qué efecto tienen estos controles en la forma de la curva de Bézier. En efecto, en cada uno de los ejemplos, la curva se dirige desde P_0 en la dirección de la recta que pasa por P_0 y P_1 , y se acerca a P_3 en la dirección de la recta que pasa por

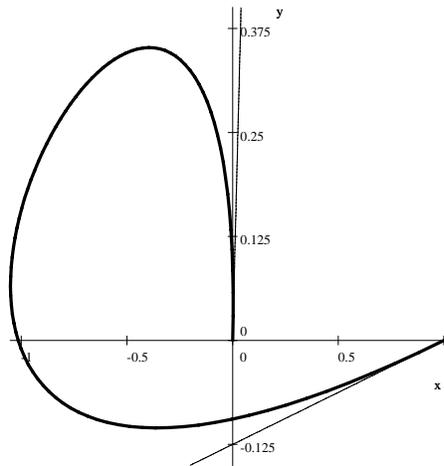
P_2 y P_3 .



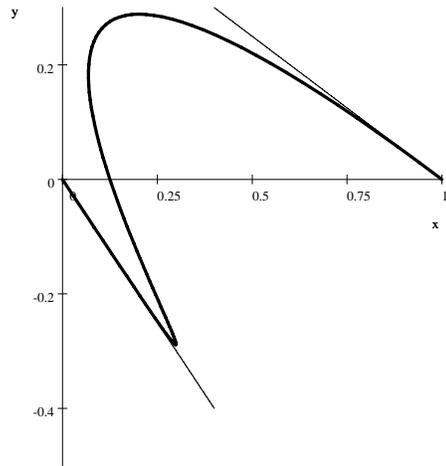
Extremos: $P_0 = (0, 0)$, $P_3 = (1, 0)$
 Controles: $P_1 = (10, 1)$, $P_2 = (0, 20)$



Extremos: $P_0 = P_3 = (0, 0)$
 Controles: $P_1 = (10, 1)$, $P_2 = (0, 20)$



Extremos: $P_0 = (0, 0)$, $P_3 = (1, 0)$
 Controles: $P_1 = (,1, 1)$, $P_2 = (-3, -,5)$



Extremos: $P_0 = (0, 0)$, $P_3 = (1, 0)$
 Controles: $P_1 = (1, -1)$, $P_2 = (-1, 1)$

Figura 9

Es decir que esta vez la curva tiene dos anclas, P_0 y P_3 , y dos conductores, P_1 y P_2 . Cambiando la posición de los puntos P_1 y P_2 , podemos “tirar” de la curva en dos direcciones diferentes. Esto permite una gran variedad de formas, como se ve en los ejemplos de la Figura 9. Podemos expresar en términos matemáticos lo que sugieren estos ejemplos. La recta que pasa por los puntos P_0 y P_1 es tangente a la curva en el punto P_0 , mientras que la recta que pasa por los puntos P_2 y P_3 es tangente a la curva en el punto P_3 .

Está claro que podríamos continuar con este proceso indefinidamente, agre-

gando más controles y usando polinomios de mayor grado. Sin embargo, vamos a parar en el caso de tercer orden, porque en general, en las aplicaciones, la manera de obtener una forma complicada es el dividirla en secciones y aproximar cada sección con curvas de Bézier de segundo o tercer orden. Uno de los aspectos interesantes de las curvas de Bézier es que quedan definidas en parte usando puntos que están fuera de la curva.

Los ejemplos de las Figuras 8 y 9 muestran que en la mayoría de los casos las curvas de Bézier no son las gráficas de funciones $x \rightarrow y$. En otras palabras, no se puede despejar t en las fórmulas $x = x(t)$ e $y = y(t)$, para todos los valores de t entre 0 y 1. Esta es precisamente una de las razones por las que consideramos curvas dadas paramétricamente. Estas curvas permiten mucha mayor variedad de formas que las gráficas de funciones $x \rightarrow y$.

Si quisieran experimentar con estas curvas de Bézier y ver qué flexibles y versátiles son, prácticamente cualquier programa para dibujar en un ordenador les permitirá diseñar sus propias curvas, anclando y tirando. Recuerden que para cada curva hay una función $t \rightarrow P(t)$ que la describe. Como ejemplo, el primer lazo en la Figura 9 consiste en todos los puntos $P(t) = (x(t), y(t))$ que satisfacen

$$\begin{cases} x(t) &= 30t - 60t^2 + 31t^3 \\ y(t) &= 3t + 54t^2 - 57t^3 \end{cases}$$

para $0 \leq t \leq 1$. Pueden obtener estas fórmulas simplemente sustituyendo en (2) los valores correspondientes de los extremos y los controles, y haciendo algunas manipulaciones algebraicas sencillas. ¿Quisieran intentar hacer lo mismo con algunas de las otras curvas? Ahora que vemos estas fórmulas, es natural el pensar si no habría sido más fácil el proponer directamente un par de polinomios de tercer grado para $x(t)$ e $y(t)$ y a continuación elegir los coeficientes dependiendo de la forma que uno quiera reproducir. La respuesta a esta pregunta es un rotundo ¡no! Si escribiéramos $x(t)$ e $y(t)$ como dos polinomios habituales, sería muy difícil el vincular los coeficientes de los polinomios con la forma de la curva. Precisamente, lo que hemos hecho es evitar ese problema usando una representación de los polinomios de tercer grado en términos de polinomios muy especiales, $(1-t)^3$, $3(1-t)^2t$, $3(1-t)t^2$ y t^3 . Estos polinomios, llamados *polinomios de Bernstein de tercer grado*, ya tenían una fecunda historia dentro de las matemáticas mucho antes del tiempo de de Casteljaou y Bézier, pero fueron ellos definitivamente quienes les dieron una vida mucho más pública dentro del mundo del diseño. Por supuesto, pueden imaginarse que para cada n uno podría escribir los $n+1$ polinomios de Bernstein de grado n . En efecto, en la misma manera que lo hicimos para $n=1, 2, 3$, esos polinomios aparecerán en el desarrollo del binomio $((1-t) + t)^n$. Observemos que el caso $n=0$ no es de importancia, porque nos daría la curva $P(t) = P_0$, la cual se reduce a un solo punto. Para ver más, mucho más sobre los polinomios de Bernstein, les recomiendo el libro escrito por G. G. Lorentz ([8]). Pero les advierto que las matemáticas de este libro, aunque extremadamente interesantes, se ponen bastante espesas. Los polinomios de Bernstein toman su nombre del matemático ruso Sergei Natanovich Bernstein (1880-1968). Bernstein estudió en París y se

graduó en la misma escuela de ingeniería donde Bézier estudió treinta años más tarde.

Lo que hemos visto es sólo el comienzo de cómo definir y usar curvas de Bézier y sus modificaciones. Si quieren ver la historia completa y más, pueden ver el libro escrito por R. H. Bartels, J. C. Beatty and B. A. Barsky ([1]). En este libro, los autores presentan el desarrollo matemático de las curvas de Bézier y otros objetos relacionados, de una manera muy intuitiva, evitando las demostraciones formales y reemplazándolas con explicaciones convincentes. Este libro también muestra que las matrices tienen un rol importante en la trama. Como una atracción adicional, el libro incluye un prefacio escrito con gran estilo por el mismo Pierre Bézier.

Antes de que nos olvidemos del porqué de habernos metido en todas estas curvas, la Figura 10 muestra cómo se ve el símbolo japonés después de haber sido suavizado con curvas de Bézier de tercer orden. Lo más importante es observar que en lugar de usar un instrumento mecánico como las curvas francesas, el símbolo ha sido suavizado por medio de fórmulas, las cuales constituyen un método mucho más sistemático y fácil de repetir.

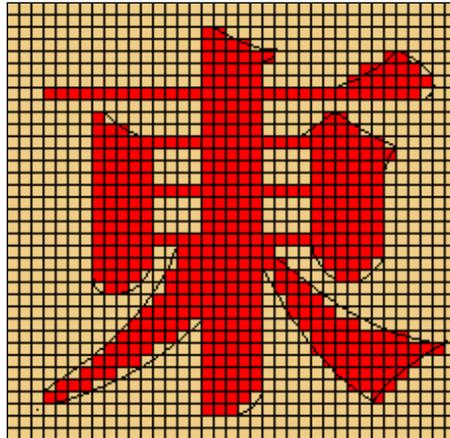


Figura 10

No solamente las curvas de Bézier son una componente importante de muchos programas de dibujo, sino que son también uno de los elementos básicos en el diseño de fuentes. ¿Fuentes? No, no me refiero a las fuentes que adornan plazas y jardines, sino a las colecciones de símbolos hechos en un estilo particular. Por ejemplo, el símbolo que aparece en la Figura 1(a) forma parte de una fuente para el idioma japonés. En un ordenador, las fuentes se diseñan usando rejillas, rectas y curvas de Bézier. Pero eso no es todo. También se incluyen efectos especiales para asegurar que los símbolos se vean igualmente atractivos a cualquier resolución, sea en la pantalla o en una página impresa. ¡Con todos estos detalles por considerar, no es ninguna sorpresa que los manuales de especificación de fuentes sean unos libretos de cientos de páginas!

Después de haber insistido en lo versátiles que son las curvas de Bézier, uno puede preguntarse si en efecto pueden reproducir exactamente cualquier forma de curva. La respuesta es no. Por ejemplo, no es posible obtener una circunferencia exactamente, aunque desde luego las curvas de Bézier permitirán lograr una aproximación satisfactoria para cualquier aplicación. ¿Cuán malo es esto de no poder hacer circunferencias? Después de todo cualquier programa para dibujar las hace. Sí, pero si uno está en el negocio de diseñar fuentes, es posible que las herramientas disponibles sean rectas y curvas de Bézier de segundo y tercer orden. Entonces el cómo aproximar circunferencias usando curvas de Bézier se convierte en una pregunta interesante. ¿Les gustaría trabajar en ella? Les sugiero que empiecen con una mitad de circunferencia centrada en 0 y con radio 1, la cual puede describirse paramétricamente con las ecuaciones

$$\begin{cases} x(t) &= \cos \pi t \\ y(t) &= \sin \pi t \end{cases}$$

para $0 \leq t \leq 1$. Desde aquí los dejo seguir por su cuenta.

En el artículo de G. Farin ([5]) y en el libro de P. B. Meggs ([9]), pueden leer la historia de cómo curvas y superficies han sido usadas en el diseño industrial. Uno de los primeros usos documentados es el de los romanos, quienes fueron maestros en el diseño y construcción de barcos. Los libros de P. B. Meggs ([9]) y E. Satue ([12]) examinan la historia del diseño desde muchos puntos de vista. Para leer una somera introducción al diseño asistido por un ordenador (CAGD), les recomiendo otro artículo de G. Farin, *Shape* ([4]) y el libro de E. Cabello Pardos ([2]). Si quieren aprender en más detalle cómo diseñar formas matemáticamente, una buena referencia es el libro de G. Farin y D. Hansford ([3]).

Volviendo a nuestro punto de partida, si todos los comentarios que he hecho sobre el punto cruz han despertado su curiosidad, la página ([10]), preparada por nuestra matemática accidental Angela Otero, tiene mucha información muy bien explicada. Un interesante ejercicio que pueden hacer leyendo esta página, es el traducir al contexto de los ordenadores, los términos y las explicaciones que se dan en la página en el contexto del punto cruz. Como se menciona en esta página, hay programas que permiten transformar fotografías en patrones de punto cruz. Esta transferencia es otra manera de cerrar el círculo entre una imagen y sus aproximaciones.

Referencias

- [1] R. H. Bartels, J. C. Beatty, B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann 1987.
- [2] E. Cabello Pardos, *Introducción al diseño por ordenador*, Universidad de Salamanca 1992.
- [3] G. Farin, D. Hansford, *The Essentials of CAGD*, AK Peters 2000.

- [4] G. Farin, Shape, *Mathematics Unlimited-2001 and Beyond*, 463-477, Springer-Verlag 2001.
- [5] G. Farin, A History of Curves and Surfaces in CAGD, *Handbook of Computer Aided Geometric Design*, 1-21, North-Holland 2002.
- [6] J. Greenberg, S. Jordan, *Chuck Close, Up Close*, DK Publishing, Inc. 1998.
- [7] J. Itten, *El arte del color*, Limusa 1992.
- [8] G. G. Lorentz, *Bernstein polynomials*, University of Toronto Press 1953.
- [9] P. B. Meggs, *Historia del diseño gráfico*, Trillas 1997.
- [10] A. Otero, *El rincón del punto cruz*,
<http://www.geocities.com/rinconpuntocruz>.
- [11] H. Rossotti, *Colour: Why the world isn't gray*, Princeton University Press 1985.
- [12] E. Satue, *El diseño gráfico desde los orígenes hasta nuestros días*, Alianza 1988.
- [13] I. Stakgold, Our Models are our Metaphors, *SIAM News*, Volume 35, Number 5, June 2002.