INTERPOLACIÓN Y MODELADO DE CURVAS

Edison De Faria Campos
Universidad de Costa Rica. (Costa Rica)
edefaria@cariari.ucr.ac.cr

Campo de investigación: modelación matemática. Nivel educativo: superior
Palabras clave: interpolación polinomial, métodos numéricos, modelado de curvas, tecnologías digitales

Resumen

Desarrollamos distintas estrategias para interpolar o aproximar una serie de datos $\{(x_i, y_i)\}_{i=0}^{n}$ en el plano mediante polinomios: interpolación de Lagrange, de Newton en diferencias divididas, splines, interpolación segmentaria de Hermite, de Bessel y curvas paramétricas polinomiales, entre ellas las curvas Bézier y B-splines que son muy populares para modelar y hacer animación por computadora, para definir formas espaciales o trayectorias de un objeto en el plano o en el espacio tridimensional. Programas como el Autocad, ilustrador de Adobe, Corel Draw y Freehand utilizan estos tipos de curvas para modelar.

Los contenidos anteriores forman parte de un curso de análisis numéricos para estudiantes de ingeniería civil que impartimos en la Universidad de Costa Rica, utilizamos el software Mathematica para definir, calcular y graficar los polinomios mencionados, y el software de geometría dinámica Cabri para graficar curvas Bézier.

Introducción

El curso de introducción al análisis numérico para estudiantes de ingeniería civil de la Universidad de Costa Rica contiene, en sus contenidos, algunos algoritmos que permiten interpolar una serie de datos en el plano. La interpolación es fundamental en la modelación matemática, particularmente en el tratamiento de la búsqueda de curvas de mejor ajuste en dos y tres dimensiones.

En el segundo semestre del año 2005 y el primer del 2006 decidimos incorporar en los dos grupos del curso las curvas de Bézier, B-splines y Nurbs por la importancia que tienen en algunos programas CAD de diseño asistido por computadora (Computer Aided Design) que son muy utilizados por los ingenieros civiles, como por ejemplo el Autocad.

La metodología utilizada consistió en distribuir las cinco horas semanales del curso en tres horas de laboratorio con 20 computadoras para los estudiantes y un servidor con proyector para el profesor y 2 horas de lecciones expositivas. Parte de la evaluación consistió en pruebas cortas en el laboratorio y algunos proyectos, sobretodo de programación en el ambiente de Mathematica.

Interpolación versus aproximación

Dado un conjunto de puntos en el plano, queremos modelar una curva suave que pasa por ellos o bien que los aproxima. Las funciones más utilizadas para la modelación son los polinomios, las funciones racionales, trigonométricas o exponenciales.

Si la curva deseada es polinomial y pasa por los puntos dados, la denominamos curva de interpolación polinomial.

Polinomio interpolante de Lagrange

Dado un conjunto de puntos

$$\left(x_0,y_0\right),\left(x_1,y_1\right),\cdots,\left(x_n,y_n\right) \text{ con } x_i \neq x_j \text{ para } i \neq j$$

en el plano, existe un único polinomio P(x) de grado menor o igual a n tal que

$$P(x_i)=y_i,\ i=0,1,\cdots,n$$

Para dos puntos $A(x_0,y_0),B(x_1,y_1)$, Lagrange utilizó la siguiente notación para la recta correspondiente:

$$P_1(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1$$

Este polinomio cumple las condiciones $P_1(x_0)=y_0,\ P_1(x_1)=y_1$.

Para tres puntos $A(x_0,y_0),B(x_1,y_1),C(x_2,y_2)$

$$P_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2$$

satisface $P_2(x_0)=y_0,P_2(x_1)=y_1,P_2(x_2)=y_2$.

En general, para n + 1 puntos, $\left(x_0,y_0\right),\left(x_1,y_1\right),\cdots,\left(x_n,y_n\right)$

$$P_n(x)=\frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y_0+\cdots+\frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})}y_n$$

$$=\sum_{k=0}^{n}y_k\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}=\sum_{k=0}^{n}y_kL_k\left(x\right)$$

El comando *InterpolatingPolynomial[* $\left\{x_0,y_0\right\},\left\{x_1,y_1\right\},\cdots,\left\{x_n,y_n\right\}$] de Mathematica genera el Polinomio Interpolante de Lagrange que pasa por

$$\left(x_0,y_0\right),\left(x_1,y_1\right),\cdots,\left(x_n,y_n\right)$$

Con el comando *Plot* podemos graficar el polinomio anterior en el intervalo que contenga los puntos dados.

Si existen puntos dados con una misma abscisa y ordenadas distintas entonces podemos obtener una interpolación paramétrica para los puntos, introduciendo un nuevo parámetro t, y construyendo dos polinomios interpolantes de Lagrange:

X(t) = InterpolatingPolynomial[$\left\{t_0,x_0\right\}$, $\left\{t_1,x_1\right\}$, $\cdots$, $\left\{t_n,x_n\right\}$]

Y(t) = InterpolatingPolynomial[$\left\{t_0,y_0\right\}$, $\left\{t_1,y_1\right\}$, $\cdots$, $\left\{t_n,y_n\right\}$]

Graficamos la pareja de funciones anteriores con el comando *ParametricPlot.*

Otro método común para construir el polinomio de interpolación es el de interpolación de Newton en diferencias divididas (Burden y Faires, 2002, Davis, 1975, Powell, 1981) que hemos tratado en el curso de introducción al análisis numérico pero, por razones de espacio, no lo trataremos aquí. Los splines cúbicos también son muy importantes pues permiten un mayor control de los datos (De Boor, 1978, Schumaker, 1981).

Curvas de Bézier

Suponga que queremos aproximar una curva polinomial entre dos pontos P_0 y P_1 dados. La solución natural es un segmento de recta que pasa por P_0 y P_1,

$$P(t) = (1 - t)\,\mathrm{P}_0 + t\,P_1,\ 0 \le t \le 1$$

Una media ponderada entre P_0 y P_1.

Para generalizar para tres puntos P_0, P_1 y P_2 consideremos primeramente los segmentos de recta P_0P_1 y P_1P_2:

$$P_{01}(t) = (1 - t)\,\mathrm{P}_0 + t\,\mathrm{P}_1$$
$$P_{11}(t) = (1 - t)\,\mathrm{P}_1 + t\,\mathrm{P}_2$$

Posteriormente interpolamos entre $P_{01}(t)$ y $P_{11}(t)$:

$$P_{02}(t) = (1 - t)\,P_{01}(t) + t\,P_{11}(t) = (1 - t)^2\,P_0 + 2\,t\,(1 - t)\,P_1 + t^2\,P_2$$

La curva obtenida contiene como pesos en P_0, P_1 y P_2 tres funciones cuadráticas:

$$b_{02}(t) = (1 - t)^2,\ b_{12}(t) = 2\,t\,(1 - t),\ b_{22}(t) = t^2$$

Aplicando la misma idea, podemos definir una cúbica por 4 puntos P_0, P_1, P_2 y P_3:

$$P_{02}(t) = (1-t)^2\,P_0 + 2\,t\,(1-t)P_1 + t^2 P_2$$
$$P_{12}(t) = (1-\mathrm{t})^2\,P_1 + 2\,t\,(1-t)\,P_2 + t^2 P_3$$

$$P_{03}(t) = (1-t)\,P_{02}(t) + t\,P_{12}(t) = (1-t)^3 P_0 + 3\,t\,(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

La curva obtenida contiene como pesos en P_0, P_1, P_2 y P_3:

$$b_{03}(t) = (1-t)^3,\ b_{13}(t) = 3t(1-t)^2$$
$$b_{23}(t) = 3t^2(1-t),\ b_{33}(t) = t^3$$

Por lo general, una curva de grado *n* puede ser construida de esta forma y se expresa como $P_{0n}(t) = \sum_{j=0}^{n} b_{j\,n}(t)\ P_j$, para ciertos polinomios de grado n.

Las curvas construidas anteriormente son conocidas como *curvas de Bézier* y las funciones de peso se denominan *base Bézier* o *polinomios de Bernstein*. Los polinomios de Bernstein de grado *n* son de la forma general $b_{kn}(t) = c_k t^k (1-t)^{n-k}$.

Los coeficientes c_k son dados por $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Los puntos P_0 y P_n son respectivamente el punto inicial y el final. Los puntos $P_1, P_2, \cdots, P_{n-1}$ son los puntos de control. La curva de Bézier pasa por P_0 y P_1, y es tangente a los segmentos $\overline{P_0P_1}$ y $\overline{P_{n-1}P_n}$.

Graficando curvas cúbicas de Bézier con Cabri

Seleccione mostrar ejes. Construya el segmento OA en el intervalo [0,1] en el eje x. Construya un punto T arbitrario en el intervalo OA y despliegue las coordenadas de T. Supongamos que las coordenadas de T son (a, 0).

Ahora construya 4 puntos distintos en el plano: P_0, P_1, P_2 y P_3. Exhiba las coordenadas de cada uno de los 4 puntos. Sean $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ las coordenadas de P_0, P_1, P_2 y P_3 respectivamente.

Utilice la herramienta *calculadora* y efectúe las siguientes operaciones

$$(1-a)^3\, x_0 + 3(1-a)^2\, a\, x_1 + 3(1-a)\, a^2\, x_2 + a^3\, x_3$$
$$(1-a)^3\, y_0 + 3(1-a)^2\, a\, y_1 + 3(1-a)\, a^2\, y_2 + a^3\, y_3$$

Guarde el primer resultado en x(t) y el segundo en y(t).

Utilice la herramienta *transferencia de medidas* y transfiera el valor de x(t) en el eje x. Sea X el punto generado.

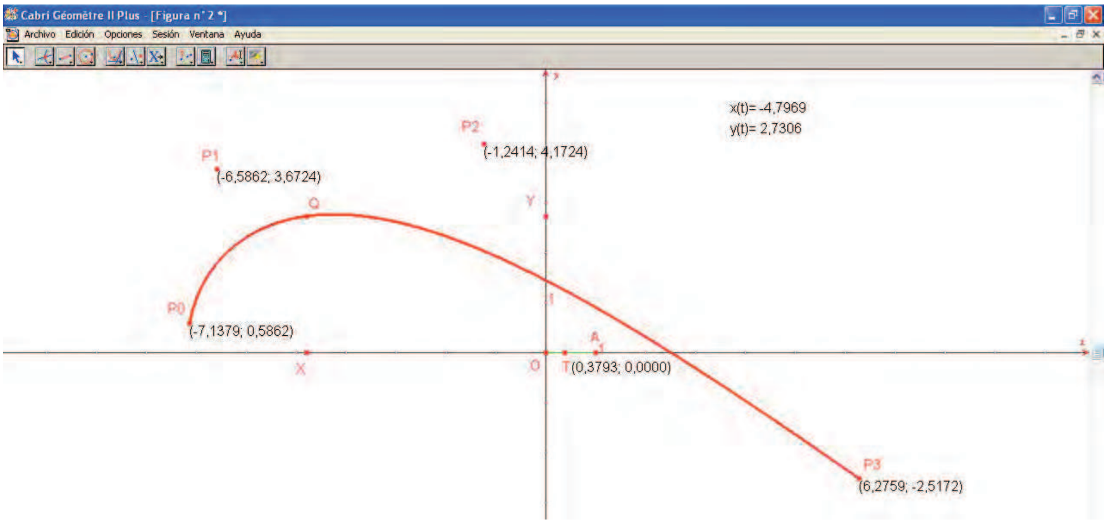
Igualmente transfiera el valor de y(t) en el eje y. Sea Y el punto generado.

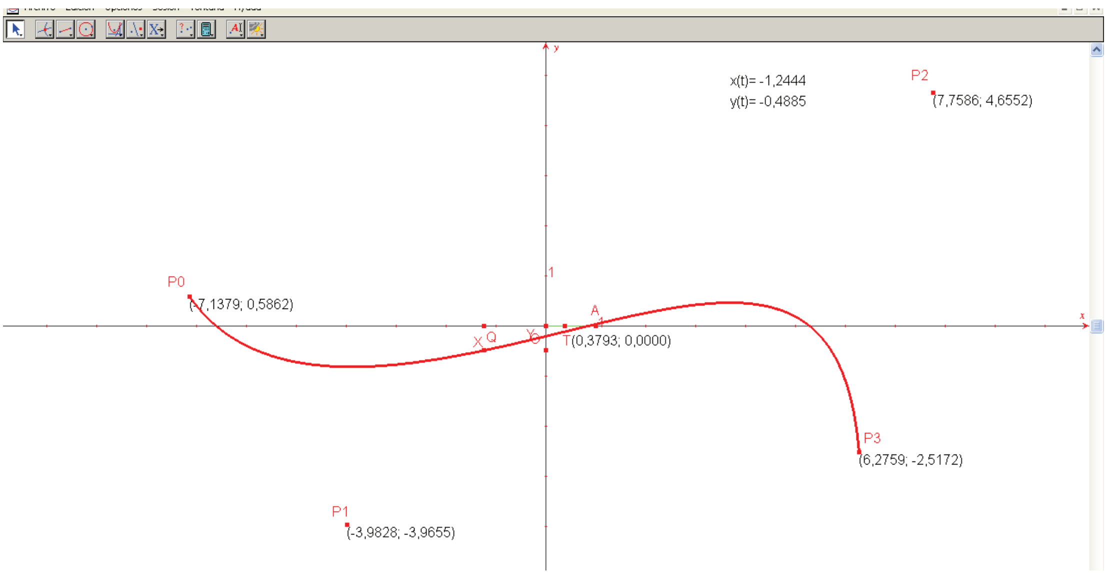
Construya una recta que pasa por X, perpendicular al eje x, y otra recta que pasa por Y, perpendicular al eje y. Sean Q el punto de intersección de las dos rectas.

Oculte las dos últimas rectas y utilice la herramienta *lugar* para determinar el lugar geométrico del punto Q cuando el punto T (correspondiente al valor del parámetro) recorre el intervalo OA.

La curva obtenida es la curva cúbica de Bézier para los 4 puntos dados. Observe que ella pasa por el punto inicial y el final.

Cambie la posición de los puntos de control P_1, P_2 para obtener nuevas curvas. Haga lo mismo con los puntos terminales P_0, P_3.





Las curvas Bézier son bastante útiles en el diseño gráfico por computadora pero presenta problemas para puntos muy alejados unos de los otros. Una mejor opción, que tratamos en el curso, son las denominadas curvas B-splines, que utiliza funciones base B-splines en lugar de los polinomios de Bernstein.

Conclusiones

La introducción de las curvas Bézier, B-Splines y Nurbs en el curso ha servido de motivación para los estudiantes quiénes asocian estos contenidos con el trabajo que realizan en los cursos de diseño asistido por computadora, graficación por computadora y con otras aplicaciones en ingeniería civil.

El uso del programa de geometría Cabri fue útil por razones didácticas y potenció la conexión de la geometría con el análisis numérico, mientras que la programación en Mathematica posibilitó la visualización de las superficies Bézier y B-Splines en el espacio tridimensional.

Referencias bibliográficas

Burden, R. L, & Faires, J. D. (2002). *Análisis numérico*. México: Thomson Learning.
Davis, P. J. (1975). *Interpolation and approximation*. Dover, New York.
De Boor, C. (1978). *A practical guide to splines*. New York: Springer-Verlag.
Powell, M. J. D. (1981). *Approximation theory and methods*. Cambridge: Cambridge University Press.
Schumaker, L. L. (1981). *Spline functions: basic theory*. New York: Wiley-Interscience.