Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Métodos Numéricos

Trabajo Práctico Nº3

Más que Splines

Nombre	LU	Mail
Carla Livorno	424/08	carlalivorno@hotmail.com
Mariano De Sousa Bispo	389/08	marian_sabianaa@hotmail.com

Abstract

El siguiente trabajo se propone la implementación de una curva paramétrica mediantes *splines* naturales a partir de un conjunto de puntos.
Además, la curva provee la funcionalidad de seleccionar un punto cualquiera de la misma (generalmente este punto se encuentra meramente *cerca* de la curva) y moverlo a una nueva posición modificando la curva.

Spline cúbico natural **Curva paramétrica** **Polinomios**

Índice

1. Introducción teórica

En este trabajo se utilizan splines cúbicos naturales como método de interpolación para generar una curva.

Un spline es una curva definida en porciones mediante polinomios (en este caso de grado 3). La idea central es que en vez de usar un único polinomio para interpolar todos los datos, se usan segmentos de polinomios entre puntos de control (pares coordenados) y se une cada uno de ellos adecuadamente para ajustar los datos. Los polinomios que definen la curva satisfacen ciertas condiciones específicas de continuidad en la frontera de cada intervalo para asegurar una transición suave.

Se utiliza a menudo la interpolación mediante splines porque da lugar a buenos resultados requiriendo solamente el uso de polinomios de bajo grado, evitando así las oscilaciones, indeseables en la mayoría de las aplicaciones, encontradas al interpolar mediante polinomios de grado elevado.

En este trabajo se utilizan curvas paramétricas en $\mathbb{R}^2$ que dadas las coordenadas de los puntos de control definen la parametrización de la siguiente manera:

Uniforme: la variación del parámetro es igual entre cualquier par de puntos de control consecutivos;

***Chord-length*:** la variación del parámetro entre dos puntos de control consecutivos es proporcional a la distancia entre los mismos;

Centrípeta: la variación del parámetro es proporcional a la raíz cuadrada de la distancia entre los puntos de control.

2. Desarrollo

2.1. Explicación

Para construir la curva parámetrica se puede utilizar cualquiera de las *tres* parametrizaciones (uniforme, chord-length y centripeta) en el intervalo $[0, 1]$. La parametrización t se elige a partir de los puntos de control (x, y) recibidos en la entrada del programa. Una vez elegida la parametrización se generan dos splines, el primero S_x a partir del par (t, x) y el otro S_y a partir de (t, y). De esta manera, queda definida una curva $C \in \mathbb{R}^2$ donde $C(t) = (S_x(t), S_y(t),)$.

Para mover un punto (x, y) cercano a la curva a una nueva posición $(x*, y*)$, calculamos primero el punto de la curva más próximo a (x, y) y luego construimos una nueva curva de manera tal que ambas splines (S_x y S_y) ahora pasen también por la nueva posición $(x*, y*)$.

Para calcular el punto de la curva más próximo a (x, y) minimizamos (derivamos y buscamos donde se anula distinguiendo entre máximos y mínimos) la función distancia[1] de la curva al punto (x, y) en cada intervalo $[t_i, t_{i+1}] \in [0, 1]$ (polinomio que la conforma) y luego seleccionamos el mínimo en $[0, 1]$.

A continuación se detalla la busqueda del punto más cercano a la curva dado (x, y).

Sea n la cantidad de puntos de control y S_x^i, S_y^i el $i - esimo$ polinomio de cada spline de la curva.

$$t \; / \; min \; \sqrt{(S_x(t) - x)^2 + (S_y(t) - y)^2} \left\{ \begin{array}{c} t_1 \; / \; min \; \sqrt{(S_x^{(1)}(t) - x)^2 + (S_y^{(1)}(t) - y)^2} \\ t_2 \; / \; min \; \sqrt{(S_x^{(2)}(t) - x)^2 + (S_y^{(2)}(t) - y)^2} \\ \vdots \\ t_{n-1} \; / \; min \; \sqrt{(S_x^{(n-1)}(t) - x)^2 + (S_y^{(n-1)}(t) - y)^2} \end{array} \right.$$

[1]Distancia de $(S_x(t), S_y(t))$ a (x, y): $\sqrt{(S_x(t) - x)^2 + (S_y(t) - y)^2}$

2.2. Implementación

La implementación esta divida en módulos que realizan tareas especificas, a continuación detallaremos cada uno de ellos.

- Módulo Parametrización:
  Este módulo implementa las *tres* parametrizaciones (uniforme, chord-length, centripeta) en el $[0, 1]$ dado un conjunto de puntos de control.

- Módulo Polinomio:
  Escribimos el módulo `Polinomio` que implementa un polinomio de grado n con las siguientes operaciones:

`Evaluar`	Evalua el polinomio en un valor recibido por parámetro.
`Derivar`	Realiza la derivada primera del polinomio.
`Ceros`	Busca una raíz del polinomio usando bisección y el método de Newton.

  EXPLICAR CON MAYOR DETALLES LAS OPERACIONES (COMO NEWTON)!!!!!!

- Módulo Spline:
  Escribimos el módulo `Spline` que implementa un spline cúbico natural con las siguientes operaciones:

`Evaluar`	Evalua la spline (el polinomio correspondiente) en un valor recibido como parámetro como parámetro.
`Polinomio`	Devuelve el polinomio requerido.

- Módulo Curva:
  El módulo `Curva` implementa una curva parámetrica con las siguientes operaciones:

`Punto cercano` Dadas las coordenadas de un punto cercano a la curva, calcula el punto de la curva más próximo.

`Mover punto` Dadas las coordenadas de un punto cercano a la curva, calcula el punto de la curva más próximo y construye una nueva spline resultante de modificar la spline original de manera que ahora pase por la nueva posición del punto seleccionado.

`Muestreo` Devuelve un muestreo de la curva.

Esta clase cuenta con un método *private* que busca el $t \in [0, 1]$ tal que al evaluar la curva en t se obtiene el punto más cercano a la misma respecto de un punto dado (x, y).

Se busca el t que minimiza la función distancia[1] del punto (x, y) a cada polinomio de la curva en el intervalo correspondiente ($[t_i, t_{i+1}]$) y luego se calcula la distancia de evaluar la curva en cada t obtenido (que minimiza la distancia a cada polinomio) a (x, y) y se selecciona el mínimo.

Llamamos $d(t)$ a la distancia de $(S_x(t), S_y(t))$ a (x, y). Para minimizar $d(t)$ obtenemos la derivada y buscamos los t donde se anula. Estos t que son los puntos criticos de la función distancia corresponden a máximos o mínimos.

3. Resultados

4. Discusión

5. Conclusiones

6. Modo de compilación y uso

Para compilar se hace uso de la herramienta Makefile.

Abrir una terminal dentro la carpeta *code* entregada y escribir el comando "make".

Para ejecutar el programa: `./tp3 input output` (donde el archivo input cumple las condiciones del enunciado).

7. Apéndices

7.1. Apéndice A: Enunciado

Laboratorio de Métodos Numéricos - Primer cuatrimestre de 2011
Trabajo Práctico Número 3: CAD - Más que splines

Los programas de diseño asistido por computadora (CAD) son herramientas fundamentales para ingenieros, arquitectos, diseñadores, artistas y animadores. Sus interfaces gráficas esconden un sinnúmero de complicadas operaciones. Un ejemplo básico de tales operaciones es la tarea de seleccionar un punto cualquiera de una curva y moverlo a una nueva posición deformando la curva. El punto seleccionado se ingresa usualmente mediante un dispositivo apuntador (*mouse* o tableta digitalizadora) interactuando con la interfaz, y en general el punto ingresado está meramente *cerca* de la curva.

En este trabajo práctico se deberán diseñar algoritmos e implementar un programa que, dadas las coordenadas $(\bar{x}_i, \bar{y}_i) \in \mathbb{R}^2$ de una serie de puntos de control ($i = 1..n$), construya una spline natural[2] paramétrica que pase por los puntos en el orden dado. Además, dadas las coordenadas $(x^*, y^*) \in \mathbb{R}^2$ de un punto cercano a la curva, calcule el punto de la curva más próximo y construya una nueva spline natural resultante de modificar la spline original de forma que ahora además pase por la nueva posición $(\bar{x}^*, \bar{y}^*) \in \mathbb{R}^2$ del punto seleccionado. El procedimiento descripto se muestra en la figura con la spline original dibujada en línea de trazos y la spline deformada en línea continua.

El programa deberá trabajar las splines como curvas paramétricas en $\mathbb{R}^2$. Dadas las coordenadas de los puntos de control existen varias estrategias para definir la parametrización. Algunas de las parametrizaciones comúnmente utilizadas son:

Uniforme: la variación del parámetro es igual entre cualquier par de puntos de control consecutivos;

---

[2]Esto es suficiente para nuestro TP, pero en realidad los sistemas de CAD utilizan más frecuentemente otros mecanismos para obtener, describir y manipular curvas y superficies.

***Chord-length*:** la variación del parámetro entre dos puntos de control consecutivos es proporcional a la distancia entre los mismos;

Centrípeta: la variación del parámetro es proporcional a la raíz cuadrada de la distancia entre los puntos de control[3].

En este trabajo práctico deberán utilizar alguna de estas parametrizaciones. Opcionalmente podrán implementar las restantes y comparar los resultados obtenidos con las tres variantes.

Además, el programa deberá conservar el valor del parámetro que le corresponde a cada punto de control y al punto seleccionado, antes y después de moverlo.

Preguntas:

1. ¿Depende la forma de la curva de la elección de la parametrización?
2. ¿Cambia la forma de la curva si en lugar de deformar la curva conservando la parametrización el programa la recalcula al mover el punto?
3. (Opcional) ¿Cómo cambia la forma de la curva según la condición de borde usada (natural, sujeto, *not-a-knot*, etc.)?
4. (Opcional) Si se quiere redibujar contínuamente la curva mientras el usuario mueve el punto seleccionado, ¿cómo se puede calcular esto más eficientemente?
5. (Opcional) Luego de mover el punto seleccionado, ¿cambia toda la curva (*control global*) o solamente una parte (*control local*)? ¿Qué consecuencias puede tener esto?
6. (Opcional) Si el intervalo del parámetro se muestrea uniformemente, ¿los puntos resultantes quedan espaciados uniformemente? ¿Qué otras alternativas de muestreo serían apropiadas?
7. (Opcional) Si se necesitara que las longitudes de curva entre puntos consecutivos sean todas iguales, ¿cómo debería muestrearse?

Archivos de entrada / salida

La entrada de datos se realizará mediante un archivo de texto con el siguiente formato:

[3]Este método fue propuesto por Eugene Lee en *Choosing nodes in parametric curve interpolation*, Computer-Aided Design 21, 1989.

- En la primera línea figurará el número n de puntos de control utilizados para definir la spline y, separado por espacio, el número m de puntos de muestreo de la spline.
- En las siguientes n líneas figurarán las coordenadas $\bar{x}$ e $\bar{y}$ de cada punto de control separadas por espacio.
- Una línea en blanco
- Una línea con las coordenadas x^* e y^* del punto próximo a la curva, separadas por espacio.
- Una línea en blanco
- Una línea con las coordenadas $\bar{x}^*$ e $\bar{y}^*$ de la nueva posición del punto, separadas por espacio.

La salida de datos estará dada por un archivo de texto con el siguiente formato:

- En la primera línea figurará el número m de puntos muestreados.
- En las siguientes m líneas figurarán las coordenadas x e y de cada punto muestreado en la spline original, separadas por espacio. Estos puntos corresponderán a un muestreo uniforme del rango del parámetro e incluirán los extremos. De esta forma, probablemente este conjunto de puntos no incluya los puntos de control originales.
- Una línea en blanco
- Una línea con las coordenadas del punto en la curva original más próximo al punto ingresado, separadas por espacio.
- Una línea en blanco
- En las siguientes m líneas figurarán las coordenadas x e y de cada punto muestreado en la spline deformada, separadas por espacio.

8. Referencias