Pontificia Universidad
JAVERIANA
Bogotá

Facultad de Ingeniería

Análisis numérico
Eddy Herrera

Reto mano interpolación

Rafael Salvador Frieri Cabrera
Daniel Hamilton-Smith Santa Cruz
Laura Juliana Mora Páez

Bogotá, 2019-01

Introducción:

Para afianzar los conocimientos adquiridos acerca de interpolación en la clase de Análisis Numérico se propuso una actividad a modo de reto para los estudiantes. Donde el reto consiste en el siguiente: “Construir un Interpolar (no necesariamente en forma polinómica) utilizando la menor cantidad de puntos puntos k y reproducir el dibujo completo de la mano (mejor exactitud) con la información dada en el script.”

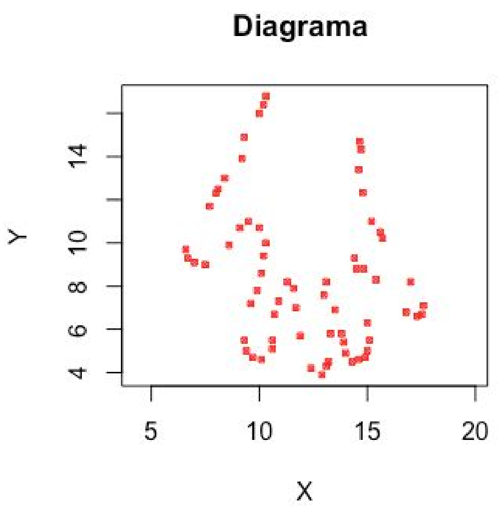
El script contaba con las coordenadas (x,y) de 67 puntos, los cuales al unirse forman la silueta de una mano; estas coordenadas vienen en 2 vectores (x,y) separadas por comas de la siguiente manera:

x=c(14.65, 14.71, 14.6, 14.8, 15.2, 15.6, 15.7, 17.0, 17.6, 17.52, 17.3, 16.8, 15.4, 14.83, 14.4, 14.5, 15.0, 15.1, 15.0, 14.9, 14.6, 14.3, 14.0, 13.9, 13.8, 13.5, 13.1, 13.0, 13.3, 13.2, 13.1, 12.9, 12.4, 11.9, 11.7, 11.6, 11.3, 10.9, 10.7, 10.6, 10.6, 10.1, 9.7, 9.4, 9.3, 9.6, 9.9, 10.1, 10.2, 10.3, 9.10, 8.6, 7.5, 7.0, 6.7, 6.6, 7.70, 8.00, 8.10, 8.40,9.20, 9.30, 10, 10.2, 10.3, 10.0, 9.50)
y=c(14.7, 14.33, 13.4, 12.33, 11.0, 10.5, 10.22, 8.20, 7.10, 6.70, 6.60, 6.80, 8.30, 8.80, 9.30, 8.80, 6.30, 5.50, 5.00, 4.70, 4.60, 4.50, 4.90, 5.40, 5.80, 6.90, 8.20, 7.60, 5.80, 4.50, 4.30, 3.90, 4.20, 5.70, 7.00, 7.90, 8.20, 7.30, 6.70, 5.50, 5.10, 4.60, 4.7, 5.0, 5.5, 7.2, 7.8, 8.60, 9.40, 10.0, 10.7, 9.9, 9.0, 9.1, 9.3, 9.7, 11.7, 12.3, 12.5, 13.0,13.91, 14.9, 16, 16.4, 16.8, 10.7, 11.0)

Tras estos vectores en el script se encuentran dos instrucciones donde una nos menciona la cantidad de puntos y la otra sirve para graficar en R:

```
length(x)
plot(x,y, pch=7, cex=0.5, col = "red", asp=1,xlab="X", ylab="Y", main="Diagrama ")
```

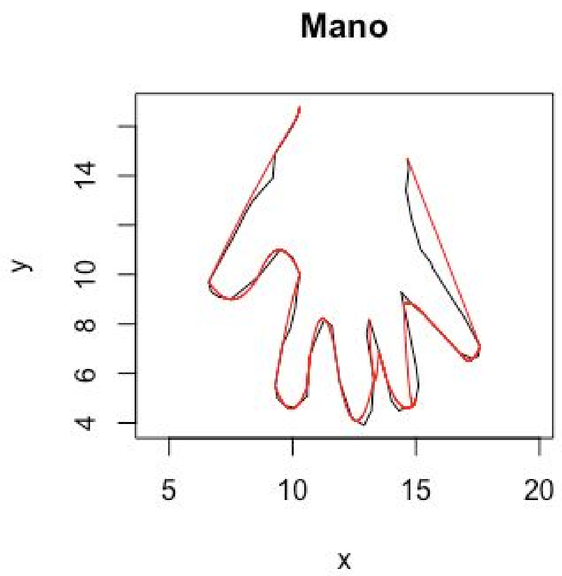
Luego de ejecutar el código del Script obtenemos la siguiente gráfica:



Descripción:
El algoritmo implementado tiene como objetivos encontrar K puntos para formar n cantidad de segmentos donde el error sea mínimo entre la función interpolada y los puntos de vector original, minimizando la cantidad de puntos utilizados, al igual que graficar estos segmentos para obtener la silueta de una mano.

Para cumplir con el objetivo se contemplaron varias opciones, como por ejemplo sacar el promedio entre cada par de puntos y así reducir la cantidad de puntos utilizados hasta la mitad, sin embargo esta posibilidad fue descartada debido al error apreciable que se presentaba en algunos segmentos de la silueta, por otro lado se consideró la posibilidad de dividir la mano en fragmentos para luego unirla. Otra opción contemplada se basó en invertir los puntos de la mano para obtener la interpolación de la mano rotada y luego devolverla a su estado inicial, sin embargo se descartó por la complejidad que presentaba este algoritmo, sin tener un efecto apreciable en la optimización del uso de puntos; finalmente se tomó la decisión de utilizar un algoritmo que tomará un punto del vector original y realizará segmentos hacia diferentes puntos del vector original, estos segmentos se calcularon bajo el funcionamiento de interpolación por spline, con base en esos segmentos calculará el error respecto al segmento original y finalmente se quedará con el segmento que presenta el error mínimo entre los segmentos obtenidos; tras obtener un segmento el punto de partida para la búsqueda de los próximos segmentos es el punto de cierre del segmento inmediatamente anterior.

Finalmente se obtuvo la siguiente silueta de una mano interpolada (mano en color rojo):



¿El origen se puede modificar? Si tenemos nueva información (nodos) ¿como podemos implementar esa información en el algoritmo de interpolación?
Debido a que el método utilizado se basa en ir verificando errores entre los diferentes puntos que se encuentran en el vector inicial y el spline calculado, es decir no depende del origen, al cambiar el valor inicial o el origen el sistema no se verá afectado debido a que

seguirá verificando errores con dicha función hasta encontrar el menor error y seguirá utilizando los segmentos de menor error con los puntos que estos establezcan.

¿Su método es robusto, en el sentido que si se tienen más puntos la exactitud no disminuye?

Mientras los puntos que se presentan no sean considerados como ruido o datos inexactos, es decir sean puntos efectivamente que hacen parte de la mano, la exactitud del algoritmo es capaz de al menos mantenerse e inclusive, debido a los segmentos más cortos, llegará a ser más preciso.

Si la información adicional o suponga tiene la información de otra mano con más cifras significativas ¿como se comporta su algoritmo ? ¿la exactitud decae?

Al haber mayor cantidad de cifras significativas, debido a que la aproximación hecha por el método utilizado a través de spline, ésta información de menor cantidad de cifras al ser comparada con valores de gran precisión podría no ser tan exacta, luego es posible que exista mayor error entre los puntos dados y los graficados con la aproximación del método.

Bibliografía:

Para consultar el funcionamiento de spline se utilizó uno de los libros proporcionados por la profesora del curso, el cual fue: Introducción a los métodos numéricos por Walter Mora F.