Caracas, 2 de enero de 2019

Universidad Simón Bolívar

CO3211 Cálculo Numérico

Trimestre septiembre-diciembre 2018

Proyecto 2

Daniel Francis

12-10863

0. Contenidos

1. Introducción

El presente informe pretende servir como justificación y representación de los datos y métodos utilizados para desarrollar los objetivos correspondientes al Proyecto 2 de la materia CO3211, Cálculo Numérico, de la Universidad Simón Bolívar, para el trimestre septiembre-diciembre 2018.

A continuación, se introducen los contenidos del informe, el cual consiste del planteamiento de los objetivos, tal cual se describen en el enunciado del proyecto. Seguidamente, se presenta una descripción del trabajo realizado, las técnicas utilizadas y las decisiones tomadas para cumplir con los objetivos.

Finalmente, se disponen las conclusiones de la ejecución de la actividad, seguido de las referencias bibliográficas y virtuales que permitieron lograr, verificar y mejorar la obtención de resultados.

2. Objetivos

En el enunciado del proyecto se plantea el siguiente objetivo principal:

Hallar una aproximación de la longitud de la línea de costa de la Península de Paraguaná usando splines cúbicos.

Se dispone de una imagen de la península en mapa de bits, con escala de 66 pixeles que equivalen a 10 kilómetros (medida verificada por el autor de este informe).

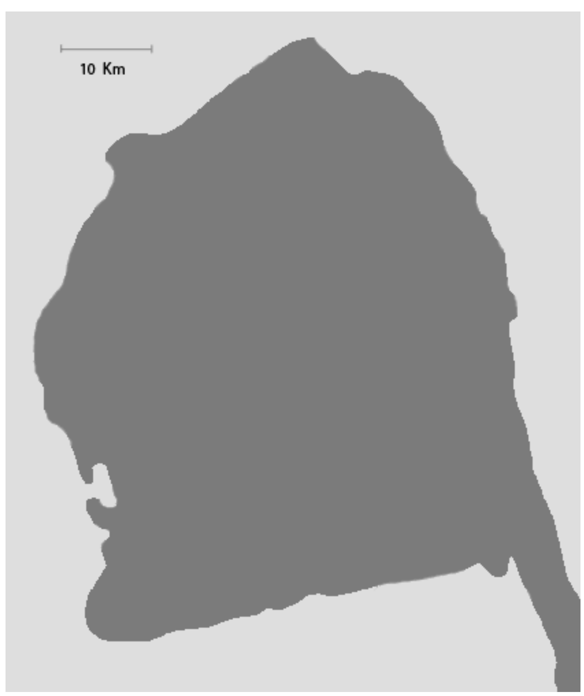

Figura 1. paraguana.bmp

3. Desarrollo

Como primera actividad, se plantea programar una función que, dados unos puntos en un plano, permita calcular el spline cúbico que los relaciona. Se pide que, de suministrarse el valor de las derivadas en los extremos del intervalo de interés, se pueda generar el spline amarrado. De lo contrario, se espera que se calcule el spline natural para los puntos.

Para implementar esta función, se tomó la explicación matricial en el libro de texto del curso y se desarrolló un algoritmo para encontrar los coeficientes del spline solicitado.

Como siguiente actividad, se propone implementar una función que calcule la longitud de una curva, específicamente una descrita por un polinomio de tercer grado.

La longitud de una curva se calculó haciendo uso de la siguiente fórmula cerrada:

$$s = \int_a^b \sqrt{1 + [f'(x)]^2}\, dx$$

Figura 2. Fórmula cerrada para hallar la longitud de un arco.

Donde *a* y *b* corresponden a los límites inferior y superior, respectivamente, del intervalo en el cual se define la curva *f(x)*. Eventualmente, se sugirió hacer uso de la distancia euclidiana como medio de asegurar que el resultado fuese una buena aproximación.

$$d_E(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

Figura 3. Fórmula de la distancia euclidiana entre dos puntos.

Luego, se pide implementar el método de Horner para evaluar un polinomio. En este caso, no se implementó el algoritmo de la manera habitual, tratándose siempre de un polinomio de grado 3, además de poseer traslación en el eje de las abscisas. Se prefirió implementar una fórmula cerrada que evaluase el polinomio en un punto dado, con un desplazamiento dado. Esta implementación permite una mejor depuración de errores.

Seguidamente, se propone hacer uso del código presente en "captura_puntos.m" para obtener los puntos de interpolación para la línea de la costa de la Península de Paraguaná. Para esto, fue necesario editar "paraguana.bmp", dibujando puntos de 1 pixel y de color rojo, código RGB (255,0,0).

Los puntos fueron seleccionados de tal forma de que no hubiese dos puntos con la misma coordenada del eje de las abscisas, ya que, de haberlo, habría problemas en el cálculo del spline, con una división entre cero. Tomando en cuenta el método del libro de texto, fue evidente la necesidad de cortar la imagen en dos porciones, una superior y una inferior, además de voltearla 90 grados hacia la derecha.

Figura 4. Puntos superiores de la península.

Puede notarse que, en el segmento de la izquierda, hay una clara discrepancia entre la línea de la costa y los puntos escogidos. Esto se debe a que, para cada punto, la coordenada de las abscisas debía ser única, además de que la función en "captura_puntos.m" encuentra los puntos rojos de forma estricta, de izquierda a derecha. Este último hecho hacía imposible que se pudiese calcular una buena aproximación de la longitud de la costa sin separar la imagen en trozos.

Figura 5. Puntos inferiores de la península.

Para una aproximación más precisa, se propone la siguiente división del mapa de bits:

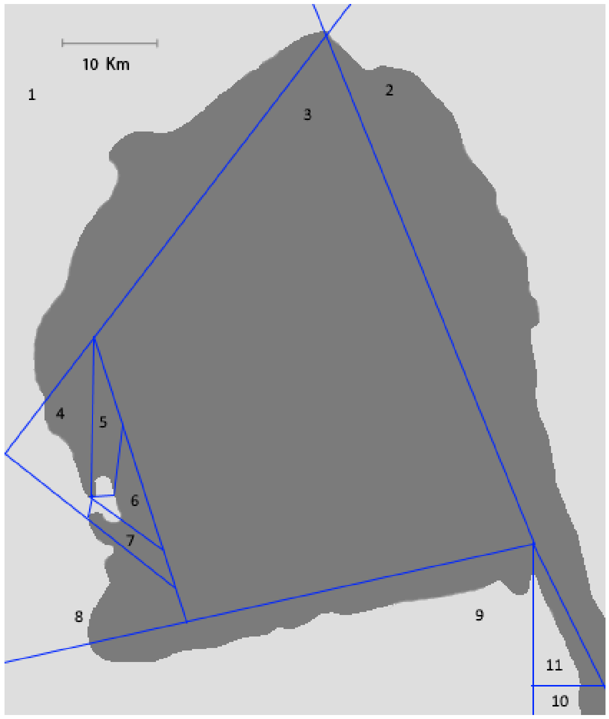

Figura 6. División en 11 partes para el mapa de bits.

Sin embargo, más adelante veremos que no fue estrictamente necesario utilizar la delimitación de la figura 6 para encontrar una buena aproximación del objetivo en cuestión. Con dividir el mapa de bits en dos secciones y escoger los puntos de manera cuidadosa fue suficiente para conseguir un resultado satisfactorio. Además, la división en dos partes permitía representar la aproximación de forma intuitiva y agradable a la vista en una gráfica de Octave, descrita a continuación.

Después de escoger los puntos, se solicita calcular los splines cúbicos a partir de los puntos obtenidos para luego graficar las funciones spline resultado.

En esta sección, fue de especial importancia recordar que cada trozo del spline está trasladado en el plano cartesiano por una distancia igual al límite inferior del intervalo correspondiente al trozo en cuestión. Se utilizaron splines naturales.

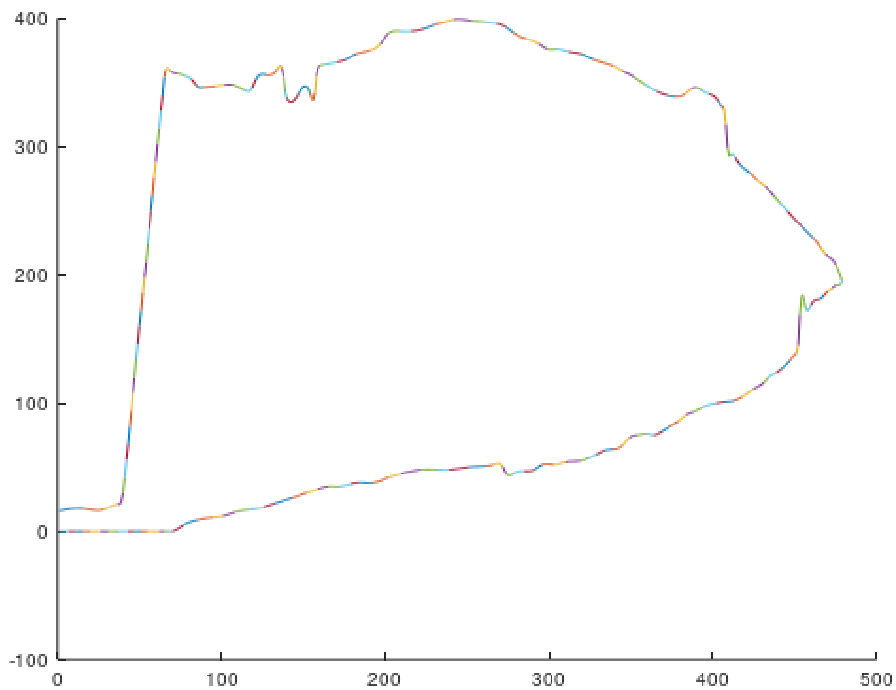

Figura 7. Gráfica de todos los spline obtenidos.

Recordando las figuras 4 y 5, se puede ver que la gráfica de los spline, cada uno representado en un color automático alternante, es bastante similar al contorno del mapa de bits original. Se nota una línea recta en la parte izquierda, que podría pensarse que resta precisión a la aproximación, pero se considera que, por asemejarse esta recta a la diagonal de un paralelogramo, siendo la línea de la costa original la otra diagonal, hay una compensación aceptable. Esto se evidencia al estudiar el resultado final.

Es importante notar que no se muestran en el informe o en el código los valores de los coeficientes de los spline, dado a que la interpretación gráfica de los ejercicios se considera que posee mayor valor a la hora de comunicar resultados y métodos.

Finalmente, se solicita programar un procedimiento que calcule la longitud de la línea de la costa de la península. Para ello, se utilizó un método iterativo que, para cada trozo del spline superior e inferior, calculase la longitud de arco, usando la función ya implementada. El resultado obtenido fue de:

1595.1387075710722 pixeles ó 241.6876829653139 kilómetros

No fue posible, dadas las restricciones de tiempo (ya que hubiera sido adecuado obtener información de algún ente del estado) conseguir una medida confiable de la longitud de la costa de la Península de Paraguaná. Sin embargo, haciendo uso de imágenes satelitales a través de Google Earth, se pudo obtener lo siguiente:

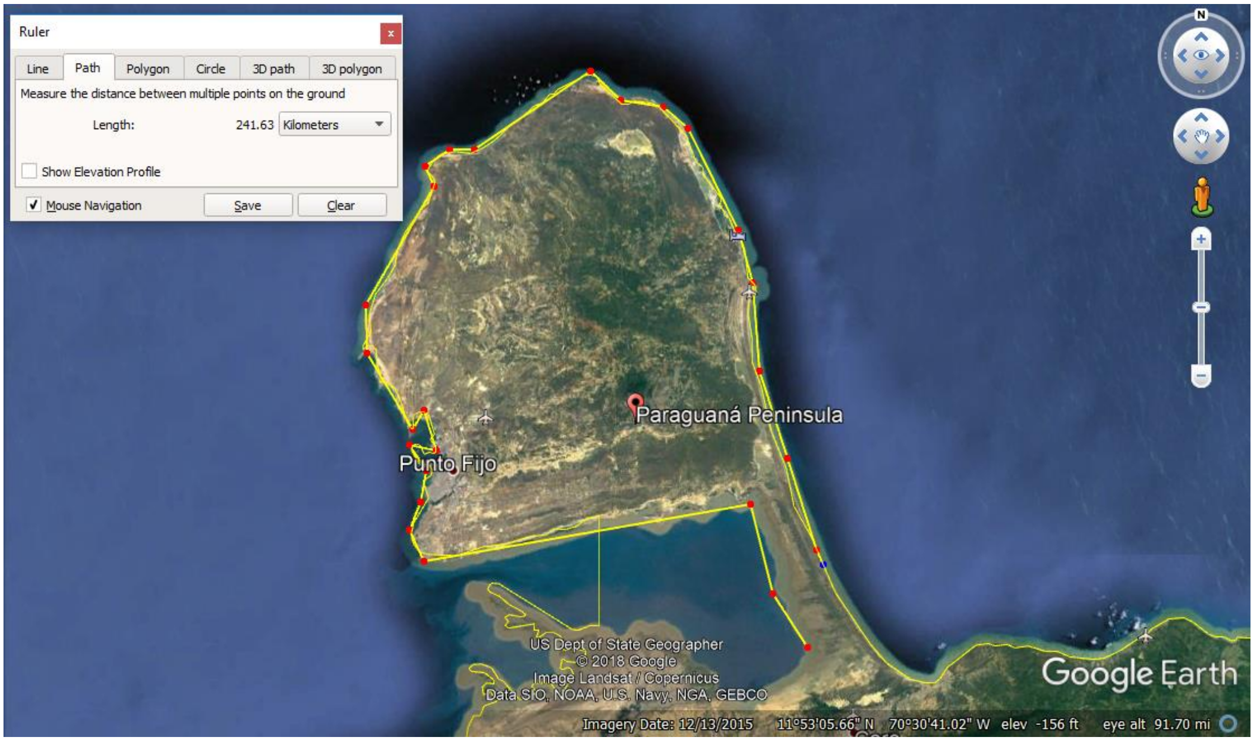

Figura 8. Imagen satelital de la Península de Paraguaná desde Google Earth.

Utilizando la herramienta de medición de Google Earth, se pudo encontrar una buena aproximación de la longitud de la costa de la Península de Paraguaná, marcada en la figura 8 por puntos rojos. Se obtuvo una distancia de **241.63 kilómetros,** una medida bastante cercana al resultado obtenido de **241.68 kilómetros.**

4. Conclusiones

Fue evidente que el uso de splines cúbicos permite obtener buenas aproximaciones de curvas irregulares, particularmente para traducirlas a una forma polinómica y manipularlas matemáticamente.

Se puede suponer que esta metodología puede ser de especial importancia para encontrar máximos y mínimos, y otras funcionalidades que brindan las aplicaciones del cálculo numérico.

Sin embargo, se presentan dudas en las diferencias entre la aproximación a través de splines y la medición de distancias a nivel de pixeles usando distancia euclidiana con muchos puntos, ya que, tendiendo a una cantidad de puntos infinita, ambos métodos deberían ser equivalentes, tomando en cuenta que, en un computador, las mediciones son discretas y no continuas. Se debe estudiar cuál método utilizar en qué caso, además de la cantidad y disposición de puntos a graficar.

Puede suponerse que el método de splines es más poderoso cuando no se dispone de un elemento base fácil de medir, como es un mapa de bits (el cual actúa como un papel cuadriculado). También existe la debilidad del método que no permite que dos puntos posean la misma coordenada en el eje de las abscisas.

Por otro lado, el proceso de separar la imagen en cuadrantes podría automatizarse. Se puede implementar un método que tome una imagen con puntos, que luego intente generar un spline, y si falla por la disposición de los puntos, que separe la imagen en cuadrantes más pequeños e intente rotarlos hasta que se logre el objetivo. Evidentemente, este método estaría abierto a mejoras, pero se puede pensar en la utilidad de una aplicación que calcule perímetros dados unos puntos en un mapa de bits.

Se considera que los resultados son satisfactorios, resultando en un error relativo de 0,0002 kilómetros si se toma la aproximación de Google Earth como verídica.

5. Referencias

Enunciado del Proyecto 2 de CO3211 – *Prof. Saúl Buitrago*. Consultado en diciembre de 2018. Disponible en línea a través de la web del curso.

Numerical Analysis, 9na edición - *Richard L. Burden, J. Douglas Faires*. Consultado en diciembre de 2018. Disponible en línea a través de la Kansas State University.

Longitud de arco – *Wikipedia. La enciclopedia libre*. Consultado en diciembre de 2018. Disponible en línea.

Ejemplo para evaluar una integral en Octave - *Universidad de Illinois*. Consultado en diciembre de 2018. Disponible en línea a través de la Universidad de Illinois.