



Universidad de Chile
Facultad de Cs. Físicas y Matemáticas
Departamento de Física
FI3104-1: Métodos Numéricos para la Ciencia e Ingeniería.

Tarea N°3

Interpolación de Polinomios.

José Ignacio Vines.
Profesor: Valentino González.
Auxiliares: Mario Aguilar.
Ignacio Armijo.
María Constanza Flores.
Fecha: 28 de septiembre de 2016



Índice

1. Introducción	2
2. Procedimiento	2
2.1. Comparación de Métodos	2
2.2. Reparación de Imagen	3
3. Resultados	3
4. Conclusiones	3

Índice de figuras

1. Introducción

El objetivo del presente informe es investigar cómo se comportan la interpolación de polinomios contra la de Spline, y utilizar el método de interpolación de Spline en 2 dimensiones para reparar una imagen dañada de una galaxia.

La interpolación de polinomios consiste en, dado un conjunto de puntos, encontrar un polinomio que pase por estos. Por otro lado, un Spline es una función definida por tramos de polinomios, con un grado de suavidad en los puntos, o nodos, donde se unen estos.

2. Procedimiento

2.1. Comparación de Métodos

Para comparar como funcionan los métodos, estos se aplicarán a una función Gaussiana definida como sigue:

$$f(x) = e^{-x^2/0,05} \quad (1)$$

En primera instancia se divide el intervalo $[-1, 1]$ en 4 tramos equiespaciados con la función **linspace**, dejando un total de 5 puntos para llevar a cabo la interpolación, luego, haciendo uso del módulo **scipy.interpolate** y de la función **lagrange** y clase **UnivariateSpline** se hace la interpolación, aumentando en 5 el número de puntos para la interpolación hasta llegar a 20, siempre manteniendo el equiespaciado del intervalo.

La función **lagrange** recibe como argumentos dos vectores, x e $y = f(x)$: un vector con el intervalo de puntos a samplear y uno con la función evaluada en dichos puntos respectivamente, y retorna un polinomio de Lagrange que pasa por todos los puntos de x .

La clase **UnivariateSpline** recibe como argumentos dos vectores, x e $y = f(x)$: un vector con el intervalo de puntos a samplear y uno con la función evaluada en dichos puntos respectivamente, y como parámetros adicionales recibe s , un factor de suavidad utilizado para elegir el número de nudos del Spline, y k , el orden del Spline. Se elige $s = 0$ para que pase por todos los puntos del intervalo y $k = 3$ para que sea un Spline cúbico. Esta clase instancia un objeto de tipo **UnivariateSpline** que representa un Spline que pasa por todos los puntos de x .

La clase **UnivariateSpline** es un wrapper de FITPACK, un paquete de subrutinas escrito en FORTRAN utilizado para calcular Splines suaves para distintos tipos de datos y geometrías¹; en particular **UnivariateSpline** hace uso de la subrutina **fpcurf0**, que a su vez utiliza la subrutina **fpcurf**² para calcular el Spline. La condición de suavidad que se debe cumplir en los nodos es que la k -ésima derivada del Spline sea 0. La documentación de **fpcurf** no contiene información de cómo maneja la implementación los extremos del intervalo, sin embargo la implementación clásica del método de Spline es imponer que la segunda derivada del Spline en los extremos sea 0, así el Spline sigue una línea recta desde los extremos.

¹<http://www.netlib.org/dierckx/>

²El archivo **fpcurf.f** está incluido en la carpeta **for_routines** en el repositorio.



2.2. Reparación de Imagen

Para reparar la imagen primero se toma una sección de ella sobre la cual se interpolará un Spline de dos dimensiones

3. Resultados

4. Conclusiones