



TAREA 3

Fecha de entrega: 29/09/2016 23:59 hrs

Problema 1

El objetivo de este problema es investigar cómo se comporta la interpolación con polinomios versus la interpolación spline en algunos casos interesantes.

Considere la siguiente función Gaussiana:

$$f(x) = e^{-x^2/0.05}$$

en el intervalo $[-1, 1]$. Divida el intervalo en 4 tramos equiespaciados (es decir, samplee 5 puntos en el intervalo $[-1, 1]$). Ahora interpole un polinomio (usando, por ejemplo, el método de Lagrange) a través de esos 5 puntos. Haga lo mismo usando una interpolación spline.

Ahora aumente secuencialmente el número de puntos y compruebe cómo se comportan los dos métodos (mejoran?, empeoran?, es lo que esperaba?).

Nota.

Puede programar su propio método de Lagrange y/o spline, o puede utilizar alguna librería que le parezca adecuada. Investigue, por ejemplo, el módulo de interpolación de `scipy`. Si decide usar una librería, asegúrese de estar usando Lagrange para los polinomios. También asegúrese de comprender los detalles de la implementación spline que esté utilizando (qué pasa en los extremos, por ejemplo?). Incluya esta información en el informe.

Problema 2

Una de las múltiples aplicaciones de los métodos de interpolación es la reconstrucción de imágenes dañadas (por fallas en el detector u otros motivos). Para esto, se requiere métodos tipo spline que funcionen en 2D.

Considere la siguiente imagen (archivo adjunto `galaxia.bmp`) que contiene pixeles dañados (el archivo `mask.bmp` le indica cuales son los pixeles dañados) y reconstrúyala. Para eso, reemplace los pixeles dañados por el valor obtenido de la interpolación 2D basada en un grupo de pixeles adyacentes no dañados.

Ayuda.

El módulo `scikit-image` le permite leer imágenes en formatos estándar (por ejemplo, `jpg`, `bmp`). Las imágenes a color son leídas en 3 arreglos 2D de tipo `numpy`, correspondientes a los colores rojo, verde y azul. El archivo `read_image.py` contiene algunas instrucciones que le pueden ser útiles.

No es necesario que escriba su propio algoritmo spline2D pero es importante que explique en detalle que hace el algoritmo que Ud. elija usar. Por ejemplo, `scipy.interpolate.SmoothBivariateSpline` es un wrapper de FITPACK. Si lo usa, explique qué es FITPACK, qué hace la función que está usando y qué parámetros utilizó al llamar a la función y por qué.

Para hacer esta parte de la tarea de manera efectiva es útil entender el concepto de *array slicing* en python y el uso de la función `numpy.where`. Estos conceptos no son exclusivos de python, sólo varían en su implementación en otros lenguajes (como `matlab`).

Instrucciones importantes.

- Utilice `git` durante el desarrollo de la tarea para mantener un historial de los cambios realizados. La siguiente [cheat sheet](#) le puede ser útil. **Esta vez revisaremos el uso apropiado de la herramienta y asignaremos una fracción del puntaje a este ítem.** Realice cambios pequeños y guarde su progreso (a través de *commits*) regularmente. No guarde código que no corre o compila (si lo hace por algún motivo deje un mensaje claro que lo indique). Escriba mensajes claros que permitan hacerse una idea de lo que se agregó de un `commit` al siguiente.
- También comenzaremos a revisar su uso correcto de python. Si define una función relativamente larga o con muchos parámetros, recuerde escribir el *docstring* que describa los parámetros que recibe la función, el output, y el detalle de qué es lo que hace la función. Recuerde que generalmente es mejor usar varias funciones cortas (que hagan una sola cosa bien) que una muy larga (que lo haga todo). Utilice nombres explicativos tanto para las funciones como para las variables de su código. El mejor nombre es aquel que permite entender qué hace la función sin tener que leer su implementación.
- La tarea se entrega subiendo su trabajo a github. Clone este repositorio (el que está en su propia cuenta privada), trabaje en el código y en el informe y cuando haya terminado asegúrese de hacer un último `commit` y luego un `push` para subir todo su trabajo a github.
- El informe debe ser entregado en formato `pdf`, este debe ser claro sin información de más ni de menos. Esto es importante, no escriba de más, esto no mejorará su nota sino que al contrario. Asegúrese de utilizar figuras efectivas y tablas para resumir sus resultados. Revise su ortografía.