

# Tarea 2

## Algoritmos y Complejidad

*Interpolation in a nutshell*

Sassy Complexes

2018-10-7

### 1. Problemas

Sea  $g$  la siguiente función:

$$g(x) = x \cos(8x) + x \sin(x) \quad x \in [0, 20]$$

Se le pide:

1. Implementar en **Python 3** (dispone de toda la biblioteca científica) funciones para los métodos de interpolación vistos en clases, *Matriz de Vandermonde*, *Lagrange* y *Diferencias divididas de Newton*. Estas funciones deben recibir dos argumentos: 1. la función a interpolar (*callable*) y 2. los puntos de interpolación (*lista*). Deberán retornar un *callable* (una función, functor, etc) que corresponda a la función interpolada. Deberá nombrar las funciones de la siguiente manera:

- `my_vandermonde`
- `my_lagranje`
- `my_divided_differences`

Por ejemplo:

---

```
def my_lagranje (f, A):  
    #codigo aqui  
    return f_interp;
```

---

donde **A** es un arreglo con los puntos de interpolación  $x_i$ , **f** es *callable* de la función a interpolar, y **f\_interp** es el *callable* del polinomio interpolador.

2. Interpolar usando sus métodos anteriores y el método `interp1d` de **SciPy** la función  $g(x)$  ya definida para:
  - a) 10, 150 y 300 puntos equiespaciados entre 0 y 20.
  - b) 10, 150 y 300 puntos de *chebychev* entre 0 y 20.

Gráficar para cada cantidad de puntos un plot **del error** usando las 4 versiones de interpolación. Comente:

- Como cambia el error a medida que varía la cantidad de puntos a interpolar.
- Como el método escogido para definir los puntos (equiespaciados o Chebychev) influye en el error.

En total se deben crear 6 gráficos (1 por cada cantidad de puntos) los cuales deben ser agregados al Latex de su entrega.

## 2. Condiciones de Evaluación de código

Se evaluará:

- **Ejecución correcta:** que funcionen los casos de prueba y **NO** sea posible encontrar casos en que el programa entregue una respuesta equivocada.
- **Complejidad computacional adecuada:** Que el algoritmo implementado tenga una complejidad igual o mejor que la esperada, y que sea ad-hoc a la materia que se está evaluando (e.g. no utilizar programación dinámica si se pide programar un algoritmo voraz).
- **Calidad del programa:** uso adecuado de funciones, uso de estructuras de control, uso de estructuras de datos, código claro y simple.
- **Código ordenado:** nombres adecuados, indentación correcta, comentarios suficientes, ausencia de código comentado.

Para más información refiérase a las condiciones de evaluación del código publicadas en moodle

## 3. Condiciones de entrega

- La tarea se realizará *individualmente* (esto es grupos de una persona), sin excepciones.
- La entrega debe realizarse vía [Moodle](#) en un *tarball* en el área designada al efecto, bajo el formato `tarea-2-rol.tar.gz` (rol con dígito verificador y sin guión).

Dicho *tarball* debe contener las fuentes en LaTeX (al menos `tarea.tex`) de la parte escrita de su entrega, además de un archivo `tarea-2.pdf`, correspondiente a la compilación de esas fuentes.

- En caso de haber programas, su ejecutable *debe* llamarse `tarea-2`, de haber varias preguntas solicitando programas, estos deben llamarse `tarea-2-1`, `tarea-2-2`, etc. Si hay programas compilados, incluya una `Makefile` que efectúe las compilaciones correspondientes.

Los programas se evalúan según que tan claros (bien escritos) son, si se compilan y ejecutan sin errores o advertencias según corresponda. Parte del puntaje es por ejecución correcta con casos de prueba. Si el programa no se ciñe a los requerimientos de entrada y salida, la nota respectiva es cero.

- Además de esto, la parte escrita de la tarea debe en hojas de tamaño carta en Secretaría Docente de Informática (Piso 1, edificio F3).
- Tanto el *tarball* como la entrega física deben realizarse el día indicado en [Moodle](#). No entregar la parte escrita en papel o no entregar en formato electrónico tiene un descuento de 50 puntos.

Por cada día de atraso se descontarán 20 puntos. A partir del tercer día de atraso no se reciben más tareas, y la nota de la tarea es cero.

- Nos reservamos el derecho de llamar a interrogación sobre algunas de las tareas entregadas. En tal caso, la nota base (antes de descuentos por atraso y otros) es la de la interrogación. No presentarse a la interrogación sin justificación previa significa automáticamente nota cero.