Prueba evaluable de programación con Maxima

Criterios de evaluación

Cada uno de los ejercicios que componen esta prueba evaluable sobre la primera parte de la asignatura Física Computacional 1 se evaluará, de 0 a 10 puntos, de acuerdo a los siguientes criterios de evaluación:

- El código aportado realiza correctamente las tareas que se pedían en el enunciado, cálculos simbólicos y/o numéricos, representaciones gráficas, etc., (sin errores sintácticos): **5 puntos**
- El código está bien estructurado, se entiende claramente lo que se hace en cada parte del mismo, la estructura es lógica y está ordenada: **2 puntos**
- El código realiza las tareas que se piden de manera eficiente, las funciones pedidas están programadas correctamente, evitando el uso de variables globales: **1.5 puntos**
- El código está documentado con comentarios que facilitan entender qué es lo que se está haciendo en cada parte del mismo, incluyendo descripción del *input* y *output* y la finalidad del código: **1.5 puntos**

La calificación final de esta parte será la media aritmética de las calificaciones obtenidas en todos los ejercicios que forman esta prueba, siempre y cuando se haya obtenido una calificación mínima de 5 puntos en todos ellos. Si uno (o más) de los ejercicios propuestos no alcanzan la calificación mínima de 5 puntos la calificación global de la prueba será *suspenso*, y no se calculará la media.

Nota:

- Es muy importante darse cuenta de que lo que se pide en cada uno de estos ejercicios es la programación de una *función*, no la resolución de un problema concreto.
- En la medida de lo posible ajústese al input y output especificado en cada ejercicio (aunque puede introducir pequeñas modificaciones si lo considera preciso, en ese caso introduzca una breve frase explicando las modificaciones introducidas).
- Muy importante, evite la definición de *variables globales* fuera de estas funciones.

- También le recomendamos que, antes de realizar estos ejercicios, revise la colección de problemas resueltos que puede encontrar en la página de la asignatura, así como las soluciones de las pruebas evaluables anteriores.

La PEC de este curso se va a centrar en el análisis de datos numéricos, interpolaciones, ajustes, así como representaciones gráficas e input y output de resultados.

Ejercicio 1

Programe una función que realice las instrucciones siguientes:

1. Cargar un archivo de datos numéricos (con la misma estructura que los empleados como ejemplo en el tema 7) y asignar estos datos a una variable local.

2. Aplicar los métodos de interpolación *lineal*, *lagrangiana* y por medio de *splines* para generar las correspondientes interpolaciones de estos datos numéricos.

3. Mostrar la gráfica donde se vean los datos numéricos y las tres interpolaciones que hemos calculado, de forma que el rango de valores ($[x_{min}, x_{max}]$, $[y_{min}, y_{max}]$) de la gráfica esté determinado por los datos numéricos.
   Además de visualizar la gráfica, esta función deberá exportar la gráfica a un archivo de tipo `pdf` en nuestro directorio de trabajo, por medio de la opción `pdf_file` de `plot2d`.

4. Exportar a un archivo las tres interpolaciones realizadas, de forma que las podamos emplear posteriormente sin necesidad de volver a calcularlas.
   Para programar esta parte existen varias opciones y se dará por buena cualquiera que funcione correctamente. Una posibilidad sencilla en este sentido es emplar la instrucción `save( sconcat( filename, ".lisp"), ...)`, donde `filename` es una variable de tipo *string* que contiene el nombre del archivo donde guardaremos las interpolaciones. Haciéndolo de esta forma generamos un archivo con formato `lisp` (y también con extensión `.lisp`) que posteriormente puede ser cargado e interpretado directamente por medio de `load`.

input:
- Nombre del archivo que contiene los datos que vamos a cargar.
- Nombre del archivo donde guardaremos la gráfica de los datos y las interpolaciones.
- Nombre del archivo donde guardaremos las tres interpolaciones realizadas.

output: Lista con los siguientes dos elementos

- Gráfica de datos e interpolaciones visualizada por pantalla.
- Gráfica de datos e interpolaciones en archivo con nombre especificado en el input anterior.
- Archivo con las tres interpolaciones calculadas.

Ejercicio 2

Programe una función que cargue el archivo de datos numéricos del apartado anterior y realice un ajuste por mínimos cuadrados, por medio de la instrucción `lsquares_estimates` (ver tema 7), de estos datos a un cierto "modelo" matemático, suministrado como un argumento de esta función.

input:
- Nombre del archivo que contiene los datos que vamos a cargar.
- Expresión matemática a la que vamos a ajustar dichos datos.
- Nombre de la variable independiente en la anterior expresión.
- Lista de parámetros que intervienen en la anterior expresión.

output: Expresión matemática que mejor se ajusta a los datos de acuerdo al modelo matemático proporcionado.

Ejercicio 3

Una vez tenemos una aproximación funcional, podemos emplearla para hacer cálculos, como por ejemplo derivadas. En este ejercicio vamos a programar una función que realice la primera derivada de la aproximación funcional que queramos analizar (ya sea resultado de las interpolaciones del primer ejercicio o de la aproximación de mínimos cuadrados del segundo) y la compare con la primera derivada de los datos numéricos, calculada por medio del algoritmo de las *diferencias centradas*. Este algoritmo consiste en lo siguiente, dado el conjunto de N puntos sobre el plano xy

$$(x_i, y_i), \qquad i = 1, 2, \ldots, N$$

donde $y = f(x)$, los valores de la derivada de $f(x)$ se aproximan por medio de

$$f'(x_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}, \qquad i = 2, \ldots, N-1$$

Lógicamente este algoritmo no puede aplicarse para calcular los valores de la derivada en los extremos del intervalo (x_1 y x_N). Para esos puntos existen otros algoritmos que pueden usarse, pero en este momento nos conformaremos con calcular la derivada en el interior del intervalo, prescindiendo de los extremos.

Para el Ejercicio 3 programe una función que cargue el archivo de datos numéricos del apartado anterior, y que reciba como un argumento la expresión matemática que (en principio) aproxima dichos datos. A continuación calcule la primera derivada de los datos numéricos por medio del algoritmo de diferencias centradas, así como la primera derivada de la función (derivando la expresión correspondiente), y muestre por pantalla una gráfica que permita comparar estas derivadas.

input:
- Nombre del archivo que contiene los datos que vamos a cargar.

- Expresión matemática que ajusta dichos datos.
- Nombre de la variable independiente en la anterior expresión.

output: Gráfica con los valores de la primera derivada.

Ejercicio 4

Para este ejercicio haga la función análoga al ejercicio anterior pero con la segunda derivada, sabiendo que el algoritmo de diferencias centradas para la segunda derivada es

$$f''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{(x_{i+1} - x_i)^2}, \qquad i = 2, \ldots, N-1$$

(igual que antes, prescindimos de las derivadas correspondientes a los valores extremos x_1 y x_N.)

- **NOTA:**

  Para hacer pruebas pueden operar con el archivo de datos "datos.out", que encontrarán en el curso virtual. No es buena idea emplear los archivos de datos del Tema 7 para hacer pruebas, ya que por un lado son conjuntos de datos algo extensos y por otro tiene algo de ruido, de manera que no son muy adecuados para calcular derivadas.

  De todas formas, recuerde que lo que se pide en esta PEC no es operar con un conjunto de datos concretos, sino programar unas funciones que, en principio, puedan aplicarse a cualquier conjunto de datos numéricos (con la limitación, por supuesto, que si el conjunto es demasiado extenso los cálculos serán demasiado lentos, y eventualmente imposibles de realizar, y que si los datos son ruidosos el cálculo de derivadas no tiene sentido, ya que al hacer derivadas se amplifica dicho ruido).

- Realmente, tal y como están escritos, los algoritmos de diferencias centradas anteriores asumen que los valores x_i están *equiespaciados*, de lo contrario habría que introducir algunos factores de corrección, pero en este momento no nos vamos a preocupar por esto.