

####24-03-2020 FCI -> PEC Maxima (2/2). Curso 2019/2020

1) Ejercicio 1

- Análisis inputs/outputs
- Carga archivo de datos numéricos.
- Aplicar métodos de interpolación lineal, lagrangiana y splines
- Graficar
 - Problema de cálculo xmin, xmax, ymin, ymax
 - Figura datos numéricos e interpolaciones.
 - Salvar figura a pdf.
- Exportar interpolaciones

2) Ejercicio 2

- Análisis inputs/outputs.
- Selección de modelo.
- Carga archivo de datos numéricos.
- Ajuste mínimos cuadrados.

3) Ejercicio 3

- Análisis inputs/outputs.
- Generación de lista de valores de primera derivada / $i=2, \dots, N-1$
- Generación de primera derivada desde expresión:
- Gráfica con valores de la primera derivada de puntos anteriores

4) Ejercicio 4

- Análisis inputs/outputs.
- Generación de lista de valores de segunda derivada / $i=2, \dots, N-1$
- Generación de segunda derivada desde expresión:
- Gráfica con valores de la segunda derivada de puntos anteriores.

5) Sugerencias.

6) Cuestiones / Foro.

1) Ejercicio 1

➤ Análisis inputs/outputs.

Se solicita desarrollar una función con los siguientes inputs:

Ej: `interpol_plotter(data_filename, plot_filename, interpol_filename) := block (.....`

Salidas esperadas:

- X Gráfica mostrada por pantalla.
- X PDF de gráfica, de datos, interpolaciones, mostrada por pantalla.
- X Archivo .lisp de interpolaciones calculadas

➤ Carga archivos datos numéricos.

`data:read_nested_list(concat(Path,data_filename));`

Nota: Path: podemos incluirlo justificándolo en comentarios como argumento de función.

➤ Aplicar métodos de interpolación lineal,lagrange,splines.

llinear:linearinterpol(data)\$

llagran:lagrange(data)\$

lspline:cspline(data)\$

➤ Graficar

- **Problema de cálculo xmin, xmax, ymin, ymax. ¿Cómo lo harían?**

Solución (Evidente):

- Loop -> Ej. for i:1 thru length(data) step 1 do

```
/* Recorrer lista uno a uno para las x y para las y */  
Xvalue: data[i][1];  
  
/* ¿Es Xvalue el valor min por el momento? */  
  
/* Si -> Salvamos */  
  
/* De igual modo para xmax,ymin e ymax */
```

Solución (Convincente):

Cálculo de dominios, convertir listas en matrices y calcular los mínimos y máximos utilizando 'apply'. Ver UD7

- Figura datos numéricos e interpolaciones.

```
plot2d ([plot1, plot2, plot3, plot4], [x, xmin,xmax],[y,ymin,ymax],[style,points,linear])..
```

Sugerencia:

- X Jugar con diferentes opciones para mostrar los graficado con claridad.
- X Verificar que lo graficado este ajustado a los puntos. Por ello es más adecuado añadir los puntos como puntos y no como líneas.

- Salvar figura a pdf.

Utilizar en plot2d la opción [pdf_file, plot_filename]
Sugerencia. Añadir .pdf para que sea reconocido por el lector.
Cuidado. Esta opción inhabilita el visualizado por pantalla.

➤ Exportar interpolaciones

Solucionado en el enunciado:

```
save(concat(interpol_filename, ".lisp"), ilinear, ilagran, ispline)$
```

2) Ejercicio 2

➤ Análisis inputs/outputs.

Se solicita desarrollar una función con los siguientes inputs:

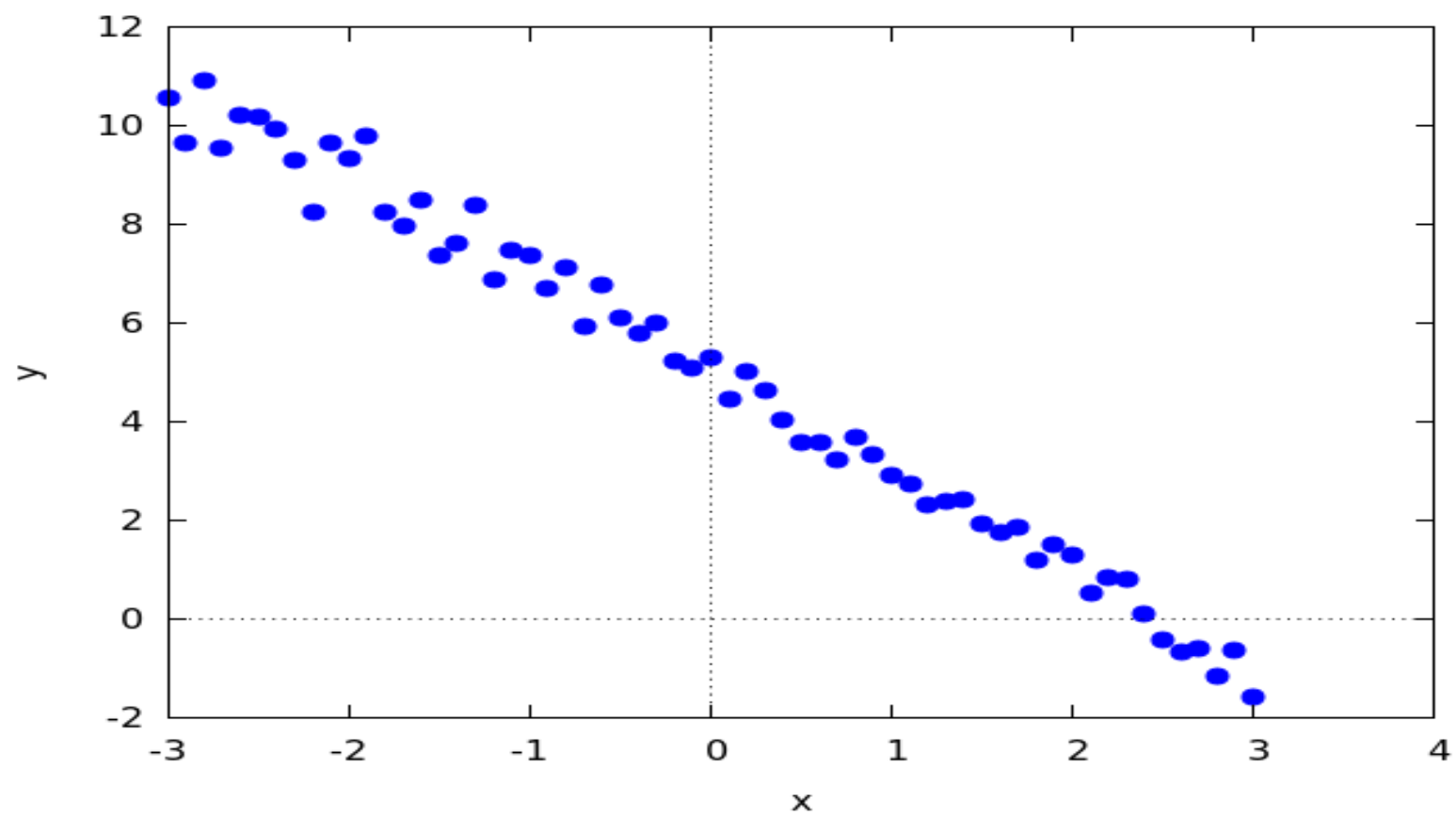
Ej: `sq_minimun(data_filename, [x,y], y=a*x+b, [a,b]) := block (.....`

Salidas esperadas:

X Expresión de ajuste.

➤ Selección del modelo. ¿Cómo seleccionamos el modelo?

Solución:



➤ Carga archivo de datos numéricos.

Utilización de lsquares_estimates. Es necesario transformar lista en matriz mediante el comando:

```
Mdata:apply(matrix, data);
```

```
float(lsquares_estimates(Mdata,[x,y],y=a*x+b,[a,b]))
```

Sugerencia utilización de float, facilita la visibilidad y el manejo.

3) Ejercicio 3

➤ Análisis inputs/outputs.

Se solicita desarrollar una función con los siguientes inputs:

Ej: `verify_1d(data_filename, expression_from_e1_or_e2, x) := block (.....`

Salidas esperadas

Gráfica comparando ambas derivadas.

➤ Generación de lista de valores de primera derivada / $i=2, \dots, N-1$

¿Qué les sugiere?

Solución (Evidente):

Loop -> for i:2 thru length(data) -1 step 1 do

/ Inicializar lista resultado */*

/ Calculo diferencia $Y_{i+1}-Y_{i-1}$ */*

/ Cálculo diferencia $X_{i+1}-X_{i-1}$ */*

/ Cálculo cociente de diferencias */*

/ Añadir en lista resultado */*

Solución (Convincente):

Utilizar instrucción makelist

➤ Generación de primera derivada desde expresión:

Utilización de instrucción diff (*expression_from_e1_or_e2* , x);

➤ Gráfica con valores de la primera derivada de puntos anteriores.

Calcular dominios. Ver ejemplo 1. Xmax,xmin,ymax e ymin

plot2d([discrete, listData],[discrete,listDerivate], [x,xmin,xmax],[y,ymin,ymax],[styles, point])

4) Ejercicio 4. (Muy similar al anterior)

➤ Análisis inputs/outputs.

Se solicita desarrollar una función con los siguientes inputs:

Ej: `verify_2d(data_filename, expression_from_e1_or_e2, x) := block (.....`

Salidas esperadas

Gráfica comparando ambas derivadas.

➤ Generación de lista de valores de primera derivada / $i=2, \dots, N-1$

Solución (Evidente):

Loop -> for $i:2$ thru $\text{length}(\text{data}) - 1$ step 1 do

`/* Inicializar lista resultado */`

`/* Calculo numerador $Y_{i+1} - 2Y_i + Y_{i-1}$ */`

`/* Cálculo denominador cuadrado de diferencia $(X_{i+1} - X_i)^2$ /`

`/* Cálculo cociente */`

`/* Añadir en lista resultado */`

Solución (Convincente):

Utilizar instrucción makelist

- Generación de la segunda derivada desde expresión:

Utilización de instrucción diff (*expression_from_e1_or_e2*, *x*, 2);

- Gráfica con valores de la primera derivada de puntos anteriores.

Calcular dominios. Ver ejemplo 1. Xmax,xmin,ymax e ymin

plot2d([discrete, listData],[discrete,listDerivate], [x,xmin,xmax],[y,ymin,ymax],[styles, point])

5) Sugerencias:

- Código solidario, muy comentado.
- Evitar cifras harcodeadas sin justificación.
- No olviden cargar librerías `load(interpol)`, `load(lsqares)`.
- Cuidado variables globales.
- Ordenen y prosigan con la estructura punto a punto propuesta para el ejercicio.
- Validen las funciones en sesiones limpias de máxima y wxmaxima.

6) Cuestiones / Foro.

¡Animo que lo tienen sencillo!