PRÁCTICA 4

Ejercicio 1: Se desea implementar el método de Lagrange para interpolar un conjunto de puntos, usando la fórmula:

$$P_n(x) = f(x_0)\,l_0(x) + f(x_1)\,l_1(x) + \cdots + f(x_n)\,l_n(x)$$

$$l_i(x) = \frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \qquad \text{con } i = 0,1,\cdots,n.$$

El programa debe preguntarnos cuántos puntos vamos a utilizar, a continuación nos debe ir solicitando las coordenadas de los puntos, y, por último, debe preguntarnos el punto donde queremos estimar la solución. El programa debe sacar en pantalla el valor de dicha estimación.
Comprueba el correcto funcionamiento del programa viendo que si le damos los 4 puntos de la tabla:

x_i	-1	0	2	3
y_i	-5	1	7	19

y pedimos la estimación en $x = 0.5$ el programa debe obtener 2.125

Ejercicio 2: Hay que hacer un programa que nos permita averiguar el polinomio interpolador de Lagrange usando el método de las diferencias divididas de Newton. Dada la función $f(x) = x + \ln(x)$ y los nodos: 1, 3, 5, 10, averiguar la tabla de las diferencias divididas. Averiguar el valor del polinomio interpolador en 5.3.

Solución:

$$\begin{pmatrix} 1 & 1.54931 & -0.0734733 & 0.00631 \\ 4.09861 & 1.25541 & -0.0166833 & - \\ 6.60944 & 1.13863 & - & - \\ 12.3026 & - & - & - \end{pmatrix}$$

P(5.3)=6.954087

Ejercicio 3: Ahora puedes coger el programa del ejercicio anterior y con breves modificaciones hacer este nuevo ejercicio.
En la gráfica siguiente se representan la función $\mathrm{f(x)} = 10\,|\mathrm{x}| = 10 * \mathrm{fabs(x)}$ y el polinomio interpolador correspondiente después de dividir el intervalo [-1,1] en 10 partes iguales. Puedes observar como cerca de los bordes del intervalo se producen grandes oscilaciones.

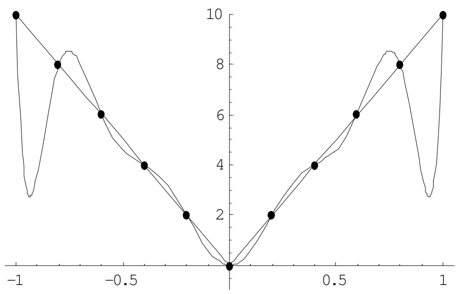

Figura 1

Estas oscilaciones se acentúan a medida que aumenta el orden del polinomio, es decir, a medida que aumenta el número de nodos de interpolación. Este fenómeno se conoce con el nombre de *efectos de borde o efecto de Runge*. Para cuantificar este hecho comprueba, con un programa en C, que $f(0.91) = 9.1$ mientras que:

a) Si dividimos el intervalo [-1, 1] en 10 partes iguales y hallamos el polinomio interpolador $P_{10}(x)$ resulta que la estimación proporcionada por el polinomio interpolador es $P_{10}(0.91) = 3.27522$.

b) Si dividimos el intervalo [-1, 1] en 20 partes iguales y hallamos el polinomio interpolador $P_{20}(x)$ resulta que la estimación proporcionada por el polinomio interpolador es $P_{20}(0.91) = 76.6756$.

Ejercicio 4: Representación gráfica con GNUPLOT

Supongamos que estamos interesados en una representación gráfica de la interpolación polinómica usando GNUPLOT.

Supongamos que la función es $f(x) = 10|x|$ en el intervalo [-1, 1]. Dividimos el intervalo en 10 partes iguales y vamos a representar el polinomio interpolador de grado 10 que pasa por esos once puntos y la función original, es decir, se desea obtener el gráfico de la fig. 1. Se deben seguir los pasos siguientes.

1) Mediante un programa adecuado en C puedes conseguir un fichero ASCII con dos columnas, con las coordenadas de los puntos de interpolación:

```
-1.00  10.00
-0.80  8.00
-0.60 6.00
-0.40  4.00
-0.20  2.00
0.00  0.00
0.20  2.00
0.40  4.00
0.60  6.00
0.80  8.00
1.00  10.00
```

Se puede conseguir mediante:

```
n = 10;
for(i = 0 ; i < = n; i ++)
{
 x = -1 + (2.0/n) * i ;
printf (" %f    %f  \n ",  x, 10*fabs(x)) ;
 }
```

Si el ejecutable se llama a.exe, por ejemplo, entonces volcando la salida en un fichero, conseguiremos el fichero ASCII anterior. Escribiendo en una ventana MS-DOS a.exe>ejercicio4.dat, conseguiremos dicho fichero.

2) Por otro lado para representar la función original $f(x)$ con GNUPLOT, evaluaremos la función $f(x)=10|x|$ en muchos puntos del intervalo [-1, 1], concretamente comenzaremos por –1 y terminaremos en 1 con incrementos de, por ejemplo, 0.01. El fichero ASCII del apartado 1, además de los 11 puntos de interpolación, contendrá los 201 puntos donde se ha evaluado la función.

3) Por último, averiguaremos el polinomio de interpolación $P_{10}(x)$ con el programa del ejercicio 3 y, al igual que antes, lo evaluaremos en muchos puntos, comenzaremos por –1 y terminaremos en 1 con incrementos de 0.01. El fichero ASCII contendrá también los 201 puntos donde se ha evaluado el polinomio interpolador $P_{10}(x)$.

Debemos separar cada uno de los tres conjuntos de puntos dentro del fichero de datos con dos líneas en blanco.

Debes obtener un fichero ASCII llamado “ejercicio4.dat” como el que puedes ver en la página web de la asignatura.

Usaremos el siguiente script de GNUPLOT :

```
set xrange[-1.1 : 1.1]
set yrange [0: 13]
plot   "ejercicio4.dat"  index 1 with lines,  "ejercicio4.dat" index 2 with lines,\
          "ejercicio4.dat"   index 0 with points 8
```

donde el significado de cada línea es sencillo: especificamos el rango de valores de x e y que vamos a emplear (aunque esto no es necesario, porque la salida por defecto que hace GNUPLOT también es adecuada) y el comando **plot** hará la representación gráfica del conjunto de puntos cuyas coordenadas vienen en el fichero ASCII llamado “ejercicio4.dat” (a dos columnas, donde la primera es la coordenada x y la segunda es la coordenada y) con la opción **lines** (que une dichos puntos con líneas) o con la opción **points 8** que dibuja puntos de tamaño 8. Recordar que el conjunto de datos estaba partido en tres partes (separadas por dos líneas en blanco cada parte). La numeración de cada parte del fichero es 0, 1 y 2 respectivamente. Con **index 0** se le pide que dibuje esos datos con puntos (correspondientes a los nodos), con **index 1** y **index 2** le pedimos que represente las otras dos porciones de datos (correspondientes a los puntos de la función $f(x)$ y a los puntos del polinomio interpolador $P_{10}(x)$).

El programa GNUPLOT nos proporciona entonces la siguiente representación gráfica:

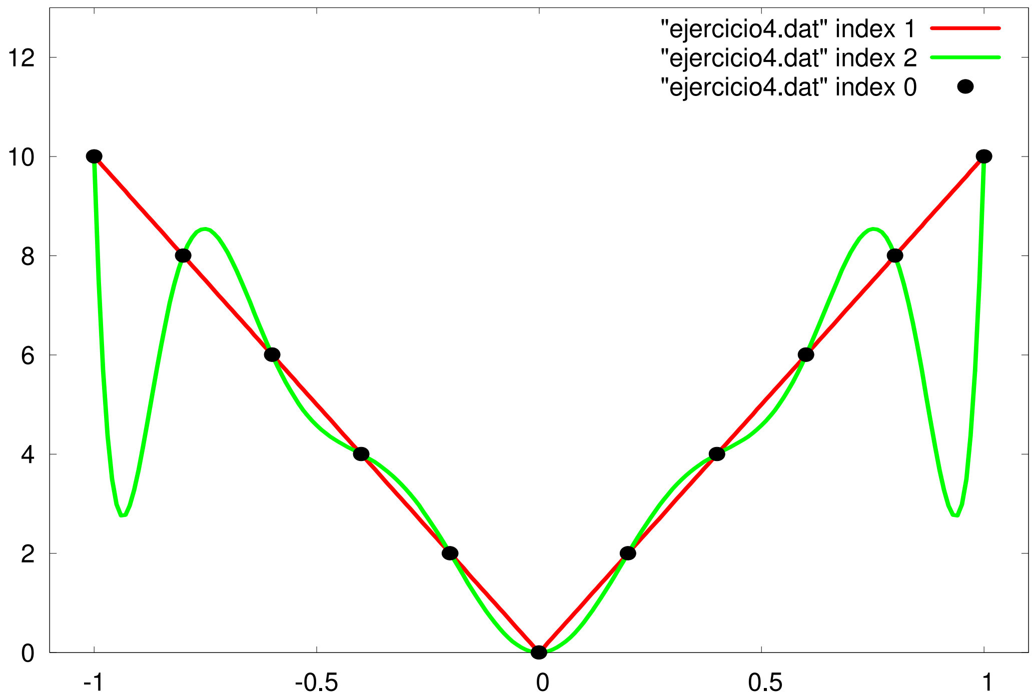

Figura 1

En resumen, debes hacer lo siguiente:

- primero: un programa en C que obtenga el fichero ASCII exactamente igual a ejercicio4.dat. Imprime el código del programa.

- segundo (opcional): con el programa GNUPLOT y usando los datos obtenidos, representar gráficamente estos datos y obtener la gráfica de la figura 1, donde se dibujan: la función, los puntos de interpolación y el polinomio interpolador. Imprime la gráfica también. Dependiendo de la versión que se utilice de GNUPLOT pueden variar los colores y/o la forma de los puntos (pueden ser triángulos, en lugar de círculos).

<u>Ejercicio 5:</u> Se considera la función $f(x) = e^{x/2}$. Averiguar la estimación que se obtiene de $f(1)$ usando el polinomio interpolador de Mac-Laurin de grado 10 y calcular el error cometido por la estimación.

Solución: $\hat{f}(1) = 1.64872...$ $\quad |error| = 1.276 \times 10^{-11}$

<u>Ejercicio 6:</u> Consideremos la función $f(x) = \dfrac{1}{25x^2 + 1}$ en el intervalo [-1, 1]. Divide en 10 partes iguales y averigua el polinomio interpolador que pasa por ellos. Comprueba que se obtiene la representación gráfica con GNUPLOT que se indica en la figura 2. Podemos observar el llamado efecto de Runge o efecto de bordes. En este caso los nodos están equiespaciados: $x_i = -1 + \frac{2}{n} i$ con $i = 0, 1, 2, \cdots, n$ con $n = 10$

A continuación se consideran 11 nodos también, pero ahora los nodos no van a estar equiespaciados sino que están definidos por: $x_i = \cos(\frac{\pi\ i}{n})$ $i = 0, 1, \ldots, n$ con $n = 10$. Comprobar como el polinomio interpolador $P_{10}(x)$ proporciona ahora mejores resultados obteniendo la gráfica de la Figura 3.

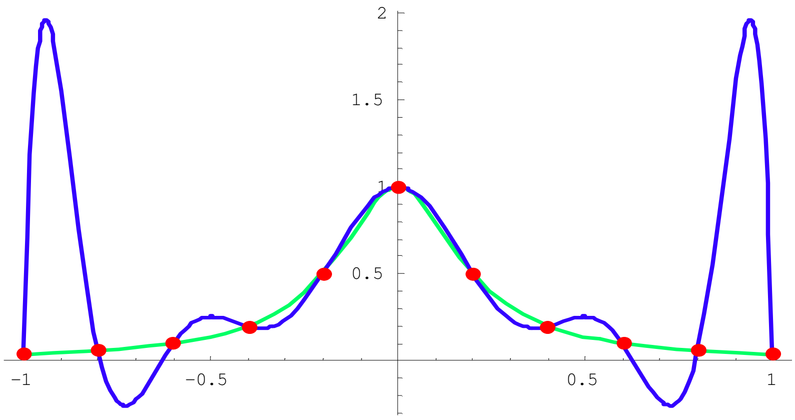

Figura 2

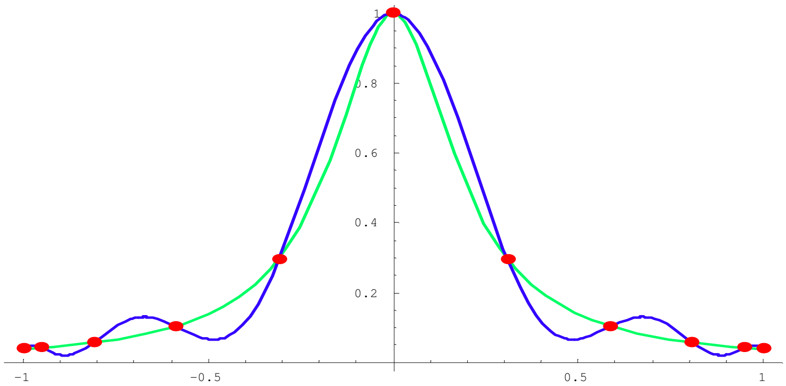

Figura 3