Guión Práctica 6

José Díaz

4 de abril de 2007

1. Introducción

En esta práctica estudiaremos los diferentes algoritmos de interpolación. Leed atentamente el contenido del sexto tema de Prácticas de Cálculo Numérico en C++, `practica6.pdf`.

2. Creación del directorio de trabajo

Conectarnos en un ordenador en Linux. Creamos un directorio de trabajo con nuestro nombre, si no existe el que creásteis en prácticas anteriores. Cuando entráis en Linux estáis en el directorio `/home/calnum`. Si cambiáis de directorio por alguna razón y queréis volver, podéis llegar al mismo con el comando

```
cd ~
```

En el terminal alfanumérico (pantalla xterm) creamos un directorio con nuestro nombre (ej. raquel) escribiendo el siguiente comando:

```
mkdir raquel
```

El comando `mkdir` crea un directorio con el nombre que sigue. Este directorio existirá hasta que se borre el disco del ordenador.

Verificáis que se ha creado el directorio con el comando

```
ls
```

que os lista todos los ficheros y directorios contenidos en el directorio donde se realiza el comando. El camino absoluto de este directorio es `/home/calnum/raquel` como podéis verificar escribiendo

```
pwd
```

en el terminal alfanumérico.

3. Descarga y descompresión de la práctica

Con el navegador Mozilla Web Browser váis a la página de cálculo numérico

`www.uv.es/~diazj/`

y os bajáis

`P6.tgz`

Hacer una actualización (reload) antes de bajarlo para asegurarnos que bajáis la última versión y no el contenido del caché del navegador.

Colocáis este fichero en `/home/calnum/raquel` con el comando

```
mv P6.tgz /home/calnum/raquel
```

Bajáis al directorio `/home/calnum/raquel`:

```
cd /home/calnum/raquel
```

y descomprimís el fichero `P6.tgz` con el comando:

```
tar -zxvf P6.tgz
```

Se creará el directorio

```
/home/calnum/raquel/P6
```

Bajáis al directorio `/home/calnum/raquel/P6`

```
cd P6
```

Ya estáis en el directorio donde váis a realizar la práctica.

4. compilación y ejecución de los programas

Esta práctica es muy similar a la anterior. En la cabecera `interpolacion.h` estan programados los algoritmos de interpolacion. Como se utilizan vectores, también hay que compilar con la librería TNT. Además muchos de los programas utilizan la clase `gnuplot_i++` para dibujar. Los programas se compilan en el caso más complicado con

```
g++ -I. -I./templates -o interpol interpol.cpp gnuplot_i_calnum.cpp
```

El resto de los programas se compilan incluyendo el directorio `./templates`

```
g++ -I./templates -o splines splines.cpp gnuplot_i_calnum.cpp
```

Para utilizar las funciones de la librería TNT hay que utilizar el namespace `CALNUM`, incluyendo la sentencia:

```
using namespace CALNUM;
```

y hay que incluir la cabecera `algebralineal.h`.

Estudiad el contenido de cada uno de los programas y realizad los ejercicios propuestos en el manual de la práctica 6, `practica6.pdf` incluido en el directorio `P6`.

Ya estáis en condiciones de realizar la memoria.

5. Realización de la memoria

Para cada uno de los ejercicios pedidos abajo cread el fichero fuente, que llamaréis `P6ej1.cpp` en el caso del programa del primer ejercicio, y los ficheros de datos en su caso, `ej1.dat` por ejemplo. Compiláis el programa

```
g++ -o P6ej1 P6ej1.cpp
```

y ejecutáis el programa en un fichero de salida `P6ej1.res`

```
./P6ej1<P6ej1.dat>P6ej1.res
```

Hacéis esto para todos los ficheros.

Para confeccionar la memoria utilizáis `emacs`, u otro editor ASCII, pero no MS-Word o editores de RichText. Creáis un fichero llamado `P6_nombre_apellido1_apellido2.txt`, y con Emacs escribís vuestro nombre y grupo de prácticas (José Pérez García. Grupo BL2) en la primera línea, e incluís en él todos los ficheros por orden correlativo: `P6ej1.cpp`, `P6ej1.dat`, `P6ej1.res`, `P6ej2.cpp`, etc. Escribís cualquier comentario a final de los ficheros de resultados del ejercicio correspondiente. Separáis cada ejercicio con una línea de caracteres, como por ejemplo:

```
=====
```

Imprimís este fichero, a ser posible a doble cara. Esta es la memoria que debéis presentar.

6. Antes de abandonar la sesión de prácticas

Os colocáis en `/home/calnum/raquel` con

```
cd ..
```

Hacéis un archivo comprimido de P6, con vuestro nombre, para que no se confunda con P6.tgz original.

```
tar -zcvf P6raquel.tgz P6
```

Podéis verificar que están todos los ficheros, listando el archivo comprimido

```
tar -ztvf P6raquel.tgz
```

Poneís este fichero en vuestro espacio web de la Universitat y lo salváis en disco o memoria USB. Tomad al menos dos medidas de precaución, para evitar pérdidas de datos. Las unidades de diskette y USB se montan haciendo doble click sobre los iconos correspondientes de la carpeta Equipo del escritorio.

7. Entregar la memoria

El listado en papel de las práctica descrito en el apartado 5 lo entregáis en la siguiente sesión de prácticas, o en su defecto lo depositáis en la casilla de correos del profesor en el Dpto. de FAMN.

Depositáis el fichero `P4_nombre_apellido1_apellido2.txt` como respuesta a la Práctica 4 en el Aula Virtual de Cálculo Numérico.

8. Ejercicios a presentar como memoria

1. Compilad y ejecutad el programa `interpol.cpp` para las funciones de Runge, Berstein y $\sin(x)$, con órdenes del polinomio interpolador de 5, 10 y 20 puntos, en el intervalo $[-1,1]$, tomando los nodos tanto espaciados uniformemente, como en los ceros de los polinomios de Chebychev (En total $3 \times 3 \times 2 = 18$ ejecuciones).

Describir brevemente una síntesis del conjunto de los resultados. (10 líneas). Adjuntad en la memoria únicamente, aparte de las 10 líneas, las tres gráficas que consideréis más ilustrativas. Los comentarios deben de ser realmente **personales**, incluso si se ha hecho la práctica compartiendo ordenador,

2. Interpolad la función $x \exp(x)$ en 10 puntos igualmente espaciados, en el intervalo $[0,2]$ (polinomio de grado 9). Representad el resultado gráficamente en un intervalo mayor que $[0,2]$ ($\Delta > 0$) (se deja a vuestra opción el tamaño, si probáis para varios tamaños distintos, seguro que tendréis algo que opinar) y comentad el resultado. Entregad sólo la gráfica que ilustre lo mejor posible vuestros comentarios.
3. Repetid el ejercicio anterior para la función $f(x) = \arctan(x)$ en el intervalo $[-1,1]$. Compilad y ejecutad `interp01.cpp` poniendo como puntos adicionales de cálculo del polinomio interpolador $x = 0,3$ y $x = 2$. ¿Cuál es el polinomio de orden más bajo que consigue una precisión de 10^{-5} en $x = 0,3$? Si se no se conociese la forma analítica de la función y solo se dispusiese de puntos de interpolación, una forma de estimar el error en un punto, es observar la diferencia entre polinomios de orden consecutivo. Con este criterio ¿Para que orden del polinomio podemos estimar que se ha alcanzado una precisión de 10^{-5} en $x = 0,3$? Comentad el comportamiento para $x = 2$. Presentad aquí solo los resultados del programa.
4. Escribid un programa que lea un fichero de datos producido por `gendat.cpp` (unos 10 o 15 puntos) y calcule el polinomio interpolador en 100 puntos entre el primer y último dato, utilizando la función

```
T Neville (const Vector <T >& x, const Vector <T >y, T & xx)
```

y los dibuje como línea continua mediante la librería `gnuplot_i++`, junto con los datos iniciales que se dibujarán. como puntos discretos. Presentad el programa, un conjunto de datos y sus resultados correspondientes.
5. Interpolad la función de Runge en el intervalo $[-1, 1]$ con 5 y 10 nodos igualmente espaciados, mediante splines. Comparad con el ejercicio 1. ¿Cuál es la conclusión? ¿Que tipo de splines funciona mejor, libres o sujetos? Comentad el resultado. En la memoria en papel presentad sólo las gráficas, aparte de vuestro comentario **personal**.

Nota: En esta práctica es muy fácil generar una cantidad impresionante de gráficas y resultados. Limitaros a presentar únicamente lo especificado en cada apartado para que la memoria sea legible fácilmente y ocupe un tamaño en papel razonable.