

Guión Práctica 8

Ajuste de datos experimentales

José Díaz

10 de mayo de 2006

1. Introducción

En esta práctica estudiaremos los diferentes métodos de ajuste de datos experimentales. Leed atentamente el contenido del tema 8 de Prácticas de Cálculo Numérico en C++, practica8.pdf y repasad lo estudiado en clase de teoría.

2. Creación del directorio de trabajo

Conectaros en un ordenador en Linux. Cread un directorio de trabajo con vuestro 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) cread un directorio con vuestro nombre (ej. eva) escribiendo el siguiente comando:

```
mkdir eva
```

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

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/eva como podéis verificar escribiendo

```
pwd
```

en el terminal alfanumérico.

3. Descarga y descompresión de la práctica

Con el browser web váis a la página de cálculo numérico

`www.uv.es/~diazj/`

y os bajáis

P8.tgz

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

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

```
mv P8.tgz /home/calnum/eva
```

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

```
cd /home/calnum/eva
```

y descomprimís el fichero `P8.tgz` con el comando:

```
tar -zxvf P8.tgz
```

Se creará el directorio

```
/home/calnum/eva/P8
```

Bajáis al directorio `/home/calnum/eva/P8`

```
cd P8
```

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 `templates.h` estan programados algunas funciones patrón utilizadas. Los programas se compilan con la función `chi2.cpp` y las librerías `TNT` y `gnuplot_i++` en la mayoría de los casos

```
g++ -I. -I./templates -o polfit polfit.cpp chi2.cpp gnuplot_i_calnum.cpp  
salvo linfit.cpp que se compila con
```

```
g++ -o linfit linfit.cpp
```

Estudiad el contenido de cada uno de los programas y realizad los ejercicios propuestos en el manual de la práctica 8, `practica8.pdf` incluido en el directorio `P8`.

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 `P8ej1.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 P8ej1 P8ej1.cpp
```

y ejecutáis el programa en un fichero de salida `P8ej1.res`

```
./P8ej1<P8ej1.dat>P8ej1.res
```

Hacéis esto para todos los ficheros.

Para confeccionar la memoria utilizáis `emacs`, u otro editor `ascii`, pero no `msword` o editores de `RichText`. Creáis un fichero llamado `P8_nombre_apellido1_apellido2.txt`,

y con emacs escribís vuestro nombre y grupo de prácticas (José Pérez García. Grupo BL3) en la primera línea e incluís en él todos los ficheros por orden correlativo: P8ej1.cpp, P8ej1.dat, P8ej1.res, P8ej2.cpp, Escribís cualquier comentario a final de los ficheros de resultados del ejercicio correspondiente. Separáis cada ejercicio con una línea como por ejemplo

=====

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

6. Antes de abandonar la sesion de prácticas

Os colocáis en /home/calnum/eva con
`cd ..`

Hacéis un archivo comprimido de P8, con vuestro nombre, para que no se confunda con P8.tgz original.

```
tar -zcvf P8eva.tgz P8
```

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

```
tar -ztvf P8eva.tgz
```

Poneís este fichero en vuestro espacio web de la universidad y lo salváis en disco o memoria USB. Tomad al menos dos medidas de precaución, para evitar pérdida 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. Realizáis un archivo comprimido de P8 excluyendo los ficheros que no hayáis creado o modificado vosotros, como los programas originales y practica8.pdf. Esto se logra con, si la alumno se llama eva lucas marcos, mediante

```
tar -zcvf P8eva_lucas_marcos.tgz --interactive --confirm P8
```

respondiendo y a cada fichero que queréis guardar.

Depositáis el fichero P8eva_lucas_marcos.tgz en el aula virtual de Cálculo Numérico, en practica 8.

8. Ejercicios a presentar como memoria

1. Considerad la función

$$f(x) = a \cdot \sin(x) + b \cdot e^{-x}$$

- a) Representadla con gnuplot para los valores $a=2$ y $b=20$.

b) Generad 40 puntos con `gendat` en el intervalo $[0,10]$, con error de los puntos 1 y dispersión de los errores 0. Utilizad el modelo de función combinación lineal de funciones con 2 funciones, y tomad 2 y 20 como los valores de los parámetros. Representad el conjunto de datos, creado en el fichero `clfun.gpl`, con `gnuplot`

```
plot "clfun.gpl" with errorbars
```

c) Ajustad el conjunto de datos con `funfit.cpp`. Anotad el valor de χ^2 y de la probabilidad de χ^2 . Realizad el proceso 10 veces. Para automatizarlo, utilizad el fichero `genclfun.d` en el que deberéis verificar el valor correcto de los parámetros. Cada ejecución de

```
./gendat<genclfun.d
```

produce un fichero de datos distinto, llamado `clfun.d`, que ajustáis con

```
./funfit<clfun.d
```

Anotáis χ^2 y χ^2/ν para cada ejecución. Con el programa `chi2` generad la distribución χ^2 con 38 grados de libertad (compilad con `g++ -o chi2 chi2main.cpp chi2.cpp`). Representad un histograma con los valores de χ^2 obtenidos junto con la distribución obtenida. Ponéis los puntos de este histograma, normalizados a la densidad de probabilidad (es decir frecuencias divididas por el número total de sucesos -10 en vuestro caso- y divididas por la anchura del intervalo), en un fichero, por ejemplo `chi2.gpl` y lo dibujáis después de `chi2-nu38` con

```
replot "chi2.gpl" with boxes
```

¿Qué conclusión extraéis? Tomad cuatro o cinco intervalos de clase, por ejemplo $[20,25]$, $[25,30]$, $[30,35]$, $[35,40]$, $[40,60]$.

Entregad el histograma, uno de los resultados de ajustes y el fichero `clfun.d` con el que se ha obtenido.

2. Generad con `gendat` una parábola $y = ax^2 + bx + c$. Tomad 20 puntos comprendidos en el intervalo de x $[0, 40]$ con error medio de los puntos 20 y dispersión del error 5. Tomad como parámetros $a = 0,1$, $b = 5$, y $c = 0,5$. A continuación ajustad el fichero de datos que os sale `parabola.d` con el programa `polfit.cpp` a una parábola y a una recta. Para esto último, tendréis que cambiar el número de parámetros de `parabola.d` de 3 a 2 (editando con `emacs` por ejemplo). ¿Podéis decidir cual es el mejor de ambos modelos mediante el valor de χ^2 ? Volved a generar los puntos, pero esta vez con un error promedio de 4 y dispersión del error 1. ¿Se puede ahora decidir entre los dos modelos? Razonad la respuesta. Entregad los dos ficheros `parabola.d` y los resultados de los 4 ajustes.

3. Las partículas elementales y en general todos los elementos radiactivos se desintegran con una vida media τ y su abundancia en función del tiempo sigue la ley exponencial

$$N(t) = N_0 e^{-t/\tau}$$

donde N_0 es el número de partículas en $t = 0$. Esta distribución se genera mediante el modelo `tau` de `gendat`.

- a) Generad un conjunto de 20 puntos mediante `gendat`, con la opción `tau`, eligiendo 10000 como valor de N_0 y tomando $\tau = 2$ y en intervalo entre 0 y 10. El error sale como $\sqrt{N}$ independientemente del valor que le déis a σ . Si la unidad de tiempo son microsegundos, esta ley correspondería a la desintegración de muones ($\tau = 2\mu s$). Intentad aproximarlos con `polfit` con un orden elevado, por ejemplo 5. Comentad el resultado obtenido. La ley, tal como la acabamos de escribir es no lineal. Deberéis de copiar `tau.gpl` en `parabola.gpl` (`cp tau.gpl parabola.gpl`) para que dibuje los resultados del ajuste con los datos en vez de con los datos del ejercicio 2.
 - b) Escribid la ley anterior tomando logaritmos. Resulta una ley lineal. Ajustad el logaritmo de las ordenadas mediante una recta en función de la abscisa, utilizando por ejemplo `linfit.cpp`, convenientemente modificado. Podéis también sustituir las ordenadas por sus logaritmos y los errores por los errores de los logaritmos en el fichero de datos, y utilizar `polfit.cpp`, aunque es más simple tomar los logaritmos de las ordenadas en `linfit2.cpp`, y sustituir `sigma[i]` por `sigma[i]/y[i]` ¿Que valor del tiempo de vida y del número inicial de partículas encontráis? (`linfit2.cpp` es el programa `linfit.cpp` con las funciones de ajuste $f_1(x) = 1$. y $f_2(x) = x$.)
4. Generad mediante `gendat`, un conjunto de 40 puntos en el intervalo $[0, 3]$ utilizando la función tipo 5, $y = (a + 0,1 * x)^n$, tomando $a = 1$ y $n = 20$. Tomad error de los puntos 0.01 y dispersión del error 0. Se genera el fichero `pol.d` en el que figuran 21 parámetros de ajuste en la primera línea. Ajustadlo con `polfit.cpp` y `ortopolfit.cpp`. Recompilad `polfit.cpp` para que dibuje `pol.d` en vez de `parabola.d`. En general saldrán resultados distintos, debido a problemas numéricos en la resolución de las ecuaciones normales en `polfit`, a pesar de que los valores del intervalo de generación se han elegido de forma que no haya pérdida de precisión ni por overflows ni underflows. Dismuid el orden de la aproximación, cambiando el primer número de `pol.d`, hasta que `polfit` y `ortopolfit` den el mismo valor de χ^2 con dos cifras decimales, En este orden de ajuste, los problemas de mal condicionamiento empiezan a desaparecer. ¿Que valor M os sale para el orden de ajuste en que `ortopolfit` y `polfit` dan el mismo resultado? Entregad los resultados de `polfit` y `ortopolfit` para $n=21$, y $n=M$.