

MÈTODES NUMÈRICS

Grau en Matemàtiques i itinerari doble Matemàtiques–Física, 2013–2014

Sessió introductòria – Laboratori docent matemàtiques

Aquest guió és una proposta de sessió de treball típica a l'entorn Linux del laboratori docent de matemàtiques. De tot el software que farem servir, trobareu més detalls a la *Guia de supervivència informàtica* al campus virtual.

Heu de tenir en compte que l'entorn de treball del laboratori docent de matemàtiques és diferent del de les aules del SID. La principal diferència és que, així com a les aules del SID teniu accés a un espai compartit on podeu deixar els vostres fitxers (que és accessible també des de Windows), al laboratori docent aquest espai no existeix. Heu de guardar la vostra feina a una clau o a un servidor extern abans de deixar l'aula.

- 1.– Boteu amb Linux (primera opció del menú d'arrencada) i obriu una sessió. Per obrir l'escriptori, heu d'entrar el username i password que hi ha a una etiqueta enganxada a l'ordinador.
- 2.– Obriu una terminal, clicant al logo del KDE (icona de l'esquerra de la barra de sota), pestanya *Aplicacions, Sistema→Terminal (Konsole)*. També podeu entrar “Konsole” al quadre de text de cerca.
- 3.– Obriu una finestra per editar, clicant al logo de KDE, pestanya *Aplicacions, Utilitats→Editor de text avançat (Kate)*. També podeu entrar “Kate” al quadre de text de cerca.

Farem servir l'editor Kate per escriure, compilar i executar programes. Farem servir la terminal per representar resultats.

Observeu que podeu canviar entre Kate i la terminal mitjançant [ALT]+[TAB].

- 4.– Activeu la terminal. Poseu-vos al directori home de l'usuari del vostre ordinador,

```
cd
```

Creeu un directori on treballarem, i poseu-vos-hi:

```
mkdir mn
cd mn
mkdir pr00
cd pr00
```

Després del `cd` no cal que teclegeu “pr00” completament: si teclegeu “pr” i premeu el tabulador, apareixeràn automàticament els dos zeros. Això passa sempre que hi ha una única possible completació. Si n'hi ha més una, una segona pitjada a sobre del tabulador us mostrarà totes les possibilitats.

- 5.– A l'editor Kate, teclegeu el programa de la figura 1.

Abans de començar a teclejar, guardeu el fitxer (*Fitxer→Desa com a*) com a

```

#include <stdio.h>
#include <math.h>

int main (void) {
    double dx;
    int i;
    dx=2*M_PI/100;
    for (i=0; i<=100; i++)
        printf("%.16G %.16G\n", i*dx, sin(i*dx));
    return 0;
}

```

Figura 1: Programa d'exemple per entrar amb l'editor Kate.

`mn/pr00/sinus.c`

(apareixereu a `/home/usuari??/Documents`, heu de pujar a `/home/usuari??` primer). L'editor ara sap que esteu escrivint un programa en C, i farà servir colors adequadament.

- 6.— Cap al fons de la finestra de Kate, cliqueu el botó *Terminal*. La finestra d'edició es dividirà verticalment en dos, de manera que la part de sota serà una terminal. Apareixereu directament al directori `/home/usuari??/mn/pr00` (observeu que s'executa una comanda `cd`) perquè és on heu guardat el fitxer que esteu editant.

A la terminal de Kate, compileu el programa:

```
gcc -o sinus -g -Wall -pedantic sinus.c -lm
```

En aquesta línia,

- `gcc` és la comanda que executeu, que és el compilador de C (es diu “gcc” per “Gnu C Compiler”).
- L'opció `-o` especifica el nom del fitxer on es guardarà el programa executable (`sinus` en aquest cas, és tradició a Unix no posar extensió `.exe` com a Windows).
- L'opció `-Wall` demaneu que el compilador escrigui tots els warnings (per defecte no avisa de moltes coses perilloses).
- L'opció `-pedantic` força la sintaxi de l'standard ISO C 90 (potser a alguna pràctica no posarem el `-pedantic`).
- `sinus.c` és el fitxer amb el programa que voleu compilar (que acabeu d'escriure a l'editor).

- `-lm` és una opció de l'enllaçador, que el fa enllaçar contra la llibreria matemàtica. Això us permet usar funcions com `sin()` o `fabs()`, que no són a la biblioteca standard del sistema `libc` (no confongueu això amb la biblioteca standard ANSI C descrita als manuals de C, que sí que té aquestes funcions).

Aquesta comanda només l'haureu d'escriure un cop per sessió. Les properes vegades que vulgueu compilar, només heu de recuperar aquesta comanda amb les fletxes del cursor.

Important: graveu el fitxer amb el codi (`sinus.c`) abans de compilar!! (no es fa automàticament).

7.— Executeu el programa a la terminal de Kate:

```
./sinus
```

Veureu que escriu els valors

$$\begin{array}{cc} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_{100} & y_{100} \end{array}$$

on $x_i = i2\pi/100$, $y_i = \sin(x_i)$ per $i = 0 \div 100$.

Volem representar gràficament aquests valors. Per a això, primer els hem de guardar dins un fitxer. Escriviu a la terminal de Kate

```
./sinus > sinus.txt
```

El símbol “>” és una redirecció de sortida, que fa que el què abans véieu a la terminal vagi a parar a un fitxer anomenat `sinus.txt`. Si feu

```
ls -l
```

en veureu la entrada corresponent al directori de treball. Si voleu veure el seu contingut, el podeu obrir amb l'editor, o, millor, el podeu veure amb la comanda `less`:

```
less sinus.txt
```

Podeu pujar i baixar amb les fletxes. Per sortir, premeu la tecla `q`.

8.— Ara representarem gràficament els resultats produïts per `sinus.c`. Per a això, activeu la terminal, poseu-vos al directori en què estem treballant,

```
cd
cd mn/pr00
```

(recordeu que podeu usar el tabulador) i engegueu el programa gnuplot:

```
gnuplot
```

Gnuplot és un programa per graficar basat en línia de comandes, que és especialment pràctic per representar fitxers de dades. També és altament scriptable (ho explotarem a les següents pràctiques). Per representar les que hem generat, feu

```
plot 'sinus.txt' using 1:2 with lines
```

En aquesta comanda:

- “plot” vol dir dibuixar en dues dimensions,
- `sinus.txt` és el fitxer que conté les dades que volem representar,
- “using 1:2” vol dir que, dins cada línia del fitxer, la columna 1 representa la coordenada x i la columna 2 la coordenada y
- “with lines” vol dir que cada parella de punts s’unirà amb línies.

Experimenteu canviant 1:2 per 2:1 i `lines` per `points`, `dots`, `impulses`, `linespoints`.

Les comandes gnuplot es poden abreujar: podem fer

```
plot 'sinus.txt' u 1:2 w l
```

- 9.— A més de graficar a la pantalla, també voldrem poder insertar una gràfica dins un document. Per a això, podeu fer, dins gnuplot (el que hi ha darrera els `#` són comentaris i no els heu d’escriure):

```
set terminal png          # graficarem en format PNG
set output 'sinus.png'    # no volem el dibuix per pantalla
                           # sinó a un fitxer "sinus.png"
replot                   # es genera el dibuix
set out                  # tanca el fitxer de sortida
set terminal x11          # per tornar a graficar a una finestra
```

(aquest darrer “set terminal” pot ser `set terminal wxt` o altres valors, depenent de l’entorn en què useu gnuplot. Quan obriu gnuplot, la darrera cosa que escriu abans de passar-vos el control és “Terminal type set to 'XXX'”. Heu de fer servir aquest). Si, a la terminal de Kate, escriviu “`ls -l`”, veureu el fitxer “`sinus.png`” que s’acaba de generar. Podeu visualitzar-lo amb la comanda `display`:

```
display sinus.png
```

que és una comanda que permet visualitzar fitxers en diversos formats gràfics com PNG, GIF, JPEG o TIFF. També permet fer edició bàsica.

Per incloure la gràfica en L^AT_EX, és preferible el format `eps`:

```
set term post eps color solid
set out 'sinus.eps'
replot
set out
```

De fet, és preferible en la major part dels casos, perquè és un format vectorial (`png` és un format raster).

- 10.— (Podeu saltar-vos aquest punt si feu el següent) En la instal·lació de Linux del SID disposeu d'un entorn complet d'oficina anomenat `openoffice.org`, que en particular inclou un processador de textos (pestanya *Aplicacions*, menú *Oficina*) molt semblant al Word, que sap llegir raonablement bé fitxers `.doc`.

Obriu-lo, escriviu algun text, i inseriu-hi el dibuix que acabeu de generar (dins el fitxer `sinus.png`).

- 11.— Si sabeu escriure en \LaTeX , genereu un fitxer `.eps` (tal com està especificat al punt 9), incloueu-lo dins el document amb `\includegraphics{}` i genereu un fitxer pdf amb el vostre document.

Si la instal·lació de \LaTeX que feu servir no admet fitxers gràfics `.eps` però sí `.pdf`, podeu convertir el `.eps` generat amb `gnuplot` a `.pdf` des d'una terminal¹ mitjançant la comanda `epstopdf`:

```
epstopdf sinus.eps
```

Podeu obrir tant fitxers `.eps` com `.pdf` amb `okular`:

```
okular sinus.eps
okular sinus.pdf
```

- 12.— Sortiu de `gnuplot`: a la terminal de `gnuplot`, escriviu

```
quit
```

- 13.— Anem a fer (compilar, de fet) un programa una mica més complicat, amb l'estructura que tindran els programes que haureu de fer a les pràctiques. Feu un directori anomenat `prg` sota el vostre directori al servidor, i poseu-vos-hi:

```
cd
cd mn/pr00
```

- 14.— Baixeu el fitxer `prg.tar.gz` escrivint

```
wget http://mat.uab.cat/~jmm/mn/prg.tar.gz
```

¹No la terminal on teniu `gnuplot` obert! Obriu-ne una altra, o teclejeu aquestes comandes a la terminal de Kate.

a la terminal.

Desempaqueteu-lo amb

```
tar xvzf prg.tar.gz
```

i poseu-vos al directori `prg` que s'acaba de crear amb

```
cd prg
```

15.— L'objectiu és generar tres programes:

- Un anomenat `c2jd`, que converteix una data de calendari a dia Julià (JD).
- Un altre anomenat `jd2c`, que converteix un JD a una data de calendari.
- Un altre anomenat `gentblhr` que, a partir del fitxer mestre de sessions del grau, genera una línia per cada sessió de cada assignatura.

16.— El fitxer `Makefile` és un fitxer preparat per a ser interpretat per la comanda `make`, que és una utilitat present a tots els sistemes Unix (i a tots els compiladors seriosos d'altres sistemes operatius). Aquesta utilitat permet automatitzar la compilació de programes que es generen a partir d'una gran quantitat de fitxers.

L'estructura dels fitxers `Makefile` és: diverses entrades de la forma

```
objectiu : dependencies
          comanda_per_generar_lobjectiu_a_partir_de_les_dependencies
```

Davant la comanda, forçosament ha d'haver un caràcter tabulador (no es permeten 8 espais).

Examineu el contingut del fitxer `Makefile` amb la comanda `less`:

```
less Makefile
```

Per sortir de `less`, premeu `q`. Obtindreu ajuda en prémer `h`.

També podeu obrir el fitxer `Makefile` des de Kate (en aquest cas, tindreu il·luminació sintàctica).

17.— Genereu els executables `c2jd`, `jd2c` amb

```
make c2jd
make jd2c
```

Observeu com es generen els codis objectes (fitxers `'.o'`) a partir dels codis fonts (fitxers `'.c'`) primer (fase de compilació), per després poder generar els executables a partir dels codis objectes (fase d'enllaçat). El fitxer de capçalera `jd.h` permet que els fitxers `'jd2c.c'` i `'c2jd.c'`, que criden les funcions `c2jd()` i `jd2c()`, es puguin compilar sense tenir accés al codi d'aquestes rutines (observeu que efectivament és així). Aquest principi és que permet que, als vostres programes, pogueu cridar funcions com `sin()` o `printf()` sense tenir accés al seu codi font.

18.— Feu execucions de prova. Per exemple:

```
./c2jd 1990 1 1
./jd2c 2447892.5
./c2jd 1992 1 1
./jd2c 2448622.5
./c2jd 1994 1 1
./jd2c 2449353.5
./c2jd 1996 1 1
./jd2c 2450083.5
./c2jd 1998 1 1
./jd2c 2450814.5
```

Quin és el dia Julià corresponent al 26 d'octubre de 1985, a les 1:20 hora de Califòrnia (UT-7 en horari d'estiu)?



19.— Processeu el fitxer dels horaris:

```
make gentblhr
./gentblhr
./gentblhr 2011 festes.txt < master_sessions_assigs_mates.txt
```

Si afegiu |less, podeu veure el resultat paginat.

```
./gentblhr 2012 festes parcials_s2.txt < mestre_assigs_mates-s2.txt | less
```