
Introducció al FORTRAN i gnuplot

part III

Compilació

- ❖ Per poder compilar i executar el nostre programa hem de fer:
 - ❖ `gfortran programa.f90 -o programa.out`
 - ❖ `./programa.out`

Errors i problemes habituals

- ❖ Els errors més típics treballant amb reals són: arrodoniment, overflow / underflow, i pèrdua de xifres significatives.

```
! Problemes d'arrodoniment
! Ricardo Vazquez Sep 2021

PROGRAM arrodoniment

IMPLICIT NONE
REAL :: x1 = 2e30, x2 = 2e8, x3

x3=x1+x2
WRITE(*,*) "x1 i x2 valen:" , x1,x2
WRITE(*,*) "x3 val", x3
WRITE(*,*) "x3-x1 val",x3-x1
WRITE(*,*) "x2 val", x2

END PROGRAM arrodoniment
```

```
x1 i x2 valen:  2.000000003E+30  200000000.
x3 val  2.000000003E+30
x3-x1 val  0.000000000
x2 val  200000000.
```

Problemes d'arrodoniment per fer servir
nombres amb una precisió fixa

Overflow

- ❖ **Overflow**: En reals tindrem problemes amb nombres més grans de 10^{38} . El compilador no dirà res i l'execució donarà “**Infinity**”

```
! Veiem errors d'overflow
! Ricardo Vazquez Sep 2021

PROGRAM overflow

IMPLICIT NONE
REAL :: x1 = 2.e30, x2

x2 = x1 + 2.e20
WRITE(*,*) "Els valors son:", x1,x2
WRITE(*,*) "1/(x1*x2) = ", 1.0/(x1*x2)
WRITE(*,*) "x1*x2 = ", x1*x2

END PROGRAM overflow
```

```
Els valors son: 2.000000003E+30 2.000000003E+30
1/(x1*x2) = 0.00000000
x1*x2 = Infinity
```


Overflow

- ❖ **Solució:** Passar a DOUBLE PRECISION

```
! Veiem errors d'overflow  
! Ricardo Vazquez Sep 2021
```

```
PROGRAM overflow
```

```
IMPLICIT NONE
```

```
DOUBLE PRECISION :: x1 = 2.d30, x2
```

```
x2 = x1 + 2.d20
```

```
WRITE(*,*) "Els valors son:", x1,x2
```

```
WRITE(*,*) "1/(x1*x2) = ", 1.0/(x1*x2)
```

```
WRITE(*,*) "x1*x2 = ", x1*x2
```

```
END PROGRAM overflow
```

Hem de canviar 2.e30 per 2.d30

```
Els valors son:  2.0000000000000000E+030  2.0000000002000001E+030  
1/(x1*x2) =     2.4999999997499998E-061  
x1*x2 =        4.0000000004000002E+060
```

Pèrdua xifres significatives

- ❖ Quan treballem amb reals molt propers i fem diferències, ens podem quedar amb poques xifres significatives.

$$X1=4.002323+/-0.000001$$

$$X2=4.002200+/-0.000001$$

$$X1-X2=0.000123+/-0.000002$$

- ❖ Els nombres X1 i X2 els coneixíem amb 7 xifres significatives, la diferència només amb 3.
- ❖ S'arregla fent servir **DOUBLE PRECISION** en comptes de **REAL**

NaN (Not a Number)

- ❖ El compilador no avalua totes les expressions, només mira que tinguin sentit.

```
! NaN  
! Ricardo Vazquez Sep 2021
```

```
PROGRAM nan
```

```
IMPLICIT NONE
```

```
WRITE(*,*) sqrt(-3.0)
```

```
END PROGRAM nan
```

```
nan.f90:8:16:
```

```
8 | WRITE(*,*) sqrt(-3.0)  
  |                1
```

```
Error: Argument of SQRT at (1) has a negative value
```

Genera un error de compilació

NaN (Not a Number)

- ❖ El compilador no avalua totes les expressions, només mira que tinguin sentit.

```
! NaN
! Ricardo Vazquez Sep 2021

PROGRAM nan

IMPLICIT NONE
REAL :: a = -3.0

WRITE(*,*) sqrt(a)

END PROGRAM nan
```

NaN

L'error apareix a l'execució, perquè el compilador no assigna el valor -3.0 a la variable [a](#).

Errors i warnings

- ❖ Per a compilar farem:
 - ❖ A Linux / Mac:
 - ❖ `gfortran programa.f90 -o executable.out`
 - ❖ A Windows:
 - ❖ `gfortran programa.f90 -o executable.exe`
- ❖ Si tot va bé, el sistema no diu res. S'ha generat un nou fitxer executable (comproveu que sigui nou mirant l'hora de creació).
- ❖ La compilació pot produir:
 - ❖ **Warnings**: problemes no tan greus per parar la compilació. Convé no tenir-ne.
 - ❖ **Errors**: el procés es deté sense generar un executable. Apareix una descripció de l'error i localització.

Errors i warnings

- ❖ Mireu sempre els missatges del compilador en ordre. Moltes vegades basta resoldre el primer i s'arregla tot.

```
! Generem un error
! Ricardo Vazquez Sep 2021
```

```
PROGRAM error
```

```
IMPLICIT NONE
```

```
INTEGE :: x
```

```
DO x=1,40
```

```
    WRITE(*,*) x
```

```
ENDDO
```

```
END PROGRAM error
```

```
error.f90:7:1:
```

```
7 | INTEGE :: x
```

```
  | 1
```

```
Error: Unclassifiable statement at (1)
```

```
error.f90:8:4:
```

```
8 | DO x=1,40
```

```
  | 1
```

```
Error: Symbol 'x' at (1) has no IMPLICIT type
```


Errors tipics I

```
! Generem errors  
! Ricardo Vazquez Sep 2021
```

```
PROGRAM error  
IMPLICIT NONE  
INTEGER :: i,j,k  
i=2  
CALL sum2(i,j)  
WRITE(*,*) i,j  
END PROGRAM error
```

```
SUBROUTINE sum2(x)  
IMPLICIT NONE  
INTEGER :: x, xp  
xp=x+2  
RETURN  
END
```

```
error2.f90:8:14:
```

```
8 | CALL sum2(i,j)  
  |                1
```

```
Error: More actual than formal arguments in procedure call at (1)
```

L'ordre i el nombre d'arguments a les
funcions o subrutines ha de ser
el correcte

Errors tipics II

```
! Generem errors
! Ricardo Vazquez Sep 2021
```

```
PROGRAM error
IMPLICIT NONE
INTEGER :: i,j,k
i=2
CALL sum2(i)
WRITE(*,*) i,j
END PROGRAM error
```

```
SUBROUTINE sum2(x, xp)
IMPLICIT NONE
INTEGER :: x, xp
xp=x+2
RETURN
END
```

```
error3.f90:8:12:
```

```
8 | CALL sum2(i)
  |              1
```

```
Error: Missing actual argument for argument 'xp' at (1)
```

L'ordre i el nombre d'arguments a les
funcions o subrutines ha de ser
el correcte

Errors tipics III

```
! Generem errors  
! Ricardo Vazquez Sep 2021
```

```
PROGRAM error  
IMPLICIT NONE  
INTEGER :: i,j,k  
REAL p  
i=2  
CALL sum2(i,p)  
WRITE(*,*) i,p  
END PROGRAM error
```

```
SUBROUTINE sum2(x, xp)  
IMPLICIT NONE  
INTEGER :: x, xp  
xp=x+2  
RETURN  
END
```

```
error4.f90:9:14:
```

```
9 | CALL sum2(i,p)  
  |              1
```

```
Error: Type mismatch in argument 'xp' at (1); passed REAL(4) to INTEGER(4)
```

El tipus d'argument a les
funcions o subrutines ha de ser
el correcte

Errors tipics IV

```
! Generem errors
! Ricardo Vazquez Sep 2021
```

```
PROGRAM error
IMPLICIT NONE
INTEGER :: i,j,k
COMMON/vars/i
```

```
i=2
CALL sum2(i,j)
WRITE(*,*) i,j
END PROGRAM error
```

```
SUBROUTINE sum2(x, xp)
IMPLICIT NONE
INTEGER :: x, xp
COMMON/vars/x
xp=x+2
RETURN
END
```

```
error5.f90:18:4:
```

```
18 | xp=x+2
    |      1
```

Error: COMMON attribute conflicts with DUMMY attribute in 'x' at (1)

```
error5.f90:17:12:
```

```
17 | COMMON/vars/x
    |              1
```

Error: COMMON attribute conflicts with DUMMY attribute in 'x' at (1)

```
error5.f90:14:17:
```

```
14 | SUBROUTINE sum2(x, xp)
    |                  1
```

Error: COMMON attribute conflicts with DUMMY attribute in 'x' at (1)

No podem passar a una subrutina, una variable que ja li estem donant amb el COMMON.

Errors tipics V

```
! Generem errors
! Ricardo Vazquez Sep 2021

PROGRAM error
IMPLICIT NONE
INTEGER :: i(10),j,k

DO j=1,20
    i(j)=j
ENDDO
END PROGRAM error
```

error6.f90:9:4:

```
8 | DO j=1,20
  |           2
9 |     i(j)=j
  |         1
```

Warning: Array reference at (1) out of bounds (20 > 10) in loop beginning at (2)

Program received signal SIGBUS: Access to an undefined portion of a memory object.

Backtrace for this error:

```
#0  0x102cf5147
#1  0x102cf41db
#2  0x1afdab4a3
zsh: bus error ./a.out
```

La posició del vector no està definida.
Dona Warning, però falla a l'execució.

Errors tipics VI

```
! Generem errors
! Ricardo Vazquez Sep 2021

PROGRAM error
IMPLICIT NONE
INTEGER :: i = 3, j = 0, k
k=i/j
END PROGRAM error
```

Intentem dividir per zero.
No dona problemes de compilació,
ni d'execució.

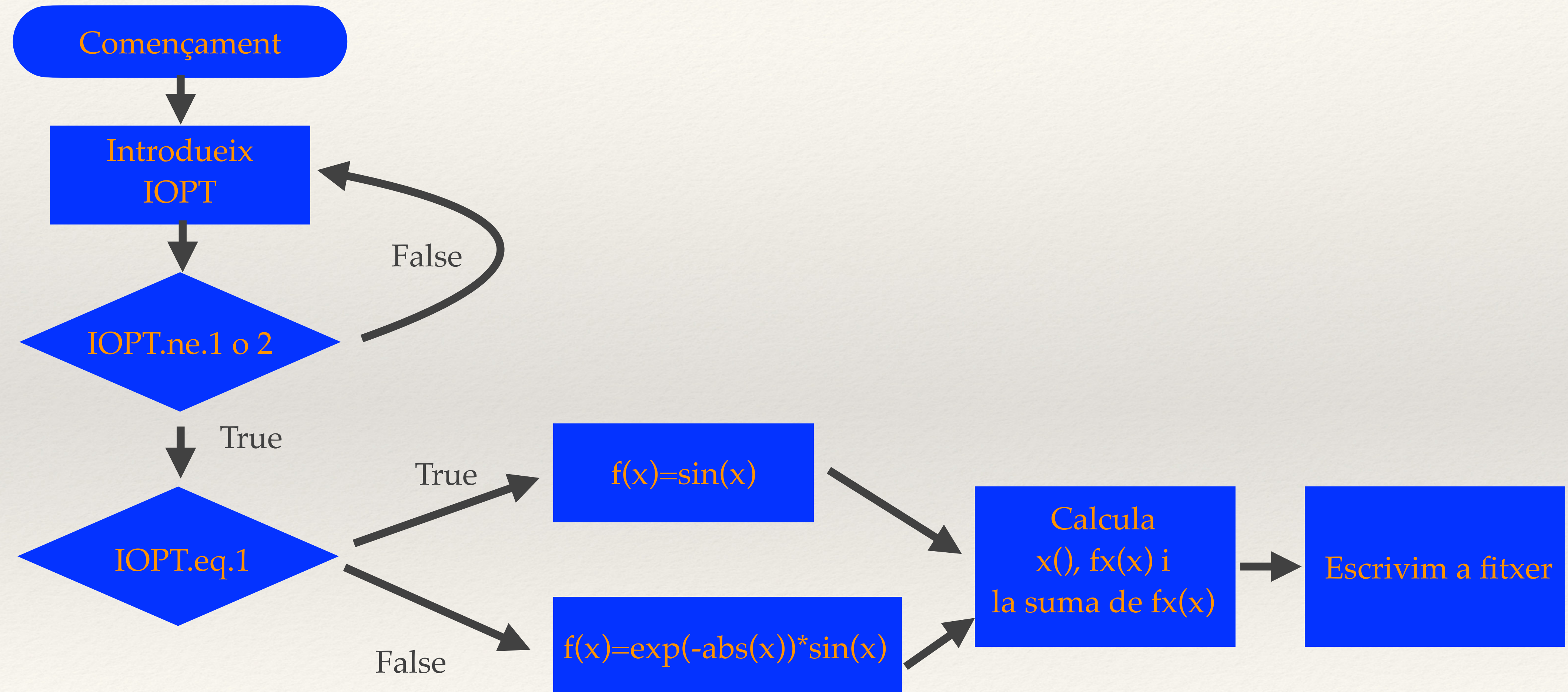
```
! Generem errors
! Ricardo Vazquez Sep 2021

PROGRAM error
IMPLICIT NONE
INTEGER :: i = 3., j = 0., k
k=i/j
WRITE(*,*) k
END PROGRAM error
```

El resultat varia depenent del tipus de
variable

Si és INTEGER -> k = 0
Si és DOUBLE PRECISION -> k = Infinity

Exemple d'un programa (diagrama fluxe)



Exemple I

```
! Exemple senzill complert
! Fem lectura i escriptura
! Fem IF
! Fem DO
! Construim una taula amb posicions (entre -5 i 5) i valors per dos funcions
! Calculem la suma dels valors de la funcio
! Les funcions son f(x)=sin(x) i f(x)=exp(-|x|)*sin(x)

PROGRAM example

IMPLICIT NONE
! nombre de punts a calcular
INTEGER :: ndim = 100
REAL :: xx(100), func(100), suma = 0.
INTEGER :: iopt = 0, i
```

Totes les variables amb declaració explícita

Un enter, que és el nombre de punts que calculem (100)

xx, **func** són vectors dels punts i valors de $f(x)$.

Enter usat per a controlar el flux.

Enter com a comptador en un bucle

Exemple II

```
WRITE(*,*) "Quina funcion vols? (1) sin(x), (2) exp(-|x|)*sin(x)"  
READ(*,*) iopt  
  
DO WHILE (iopt.NE.1.AND.iopt.NE.2)  
WRITE(*,*) "opcion incorrecta, escriu 1 o 2"  
    READ(*,*) iopt  
ENDDO
```

Demana a l'usuari que introdueixi un numero.
Si el valor no és 1 o 2, continua preguntant fins a
que es doni una opció correcte.

Exemple III

```
! calcula els valors de la funcio
DO i=1,ndim
! posicions de -5 a 5 amb pas constant
xx(i)=-5. + real(i-1)*10./ndim
IF (iopt.EQ.1) THEN
    func(i)=sin(xx(i))
ELSE
    func(i)=exp(-abs(xx(i)))*sin(xx(i))
ENDIF
ENDDO

! calcula la suma dels valors de la funcio
DO i=1,ndim
    suma=suma+func(i)
    WRITE(*,*) suma,func(i)
ENDDO
```

Bucle principal.

Calcula els valors de la funció per a 100 valors de x equiespaiats entre -5 i 5.

Calcula el valor de la suma de tots els valors de la funció

Exemple IV

```
! escrivim a un fitxer
OPEN(15,FILE='sortida.dat')
DO i=1,ndim
    WRITE(15,"(f9.3,2x,f12.6)") xx(i), func(i)
ENDDO
CLOSE(15)

! escrivim per pantalla la suma
WRITE(*,*) 'la suma val ', suma

END PROGRAM example
```

Escrivim a un fitxer amb format.

Escrivim un resultat a la pantalla

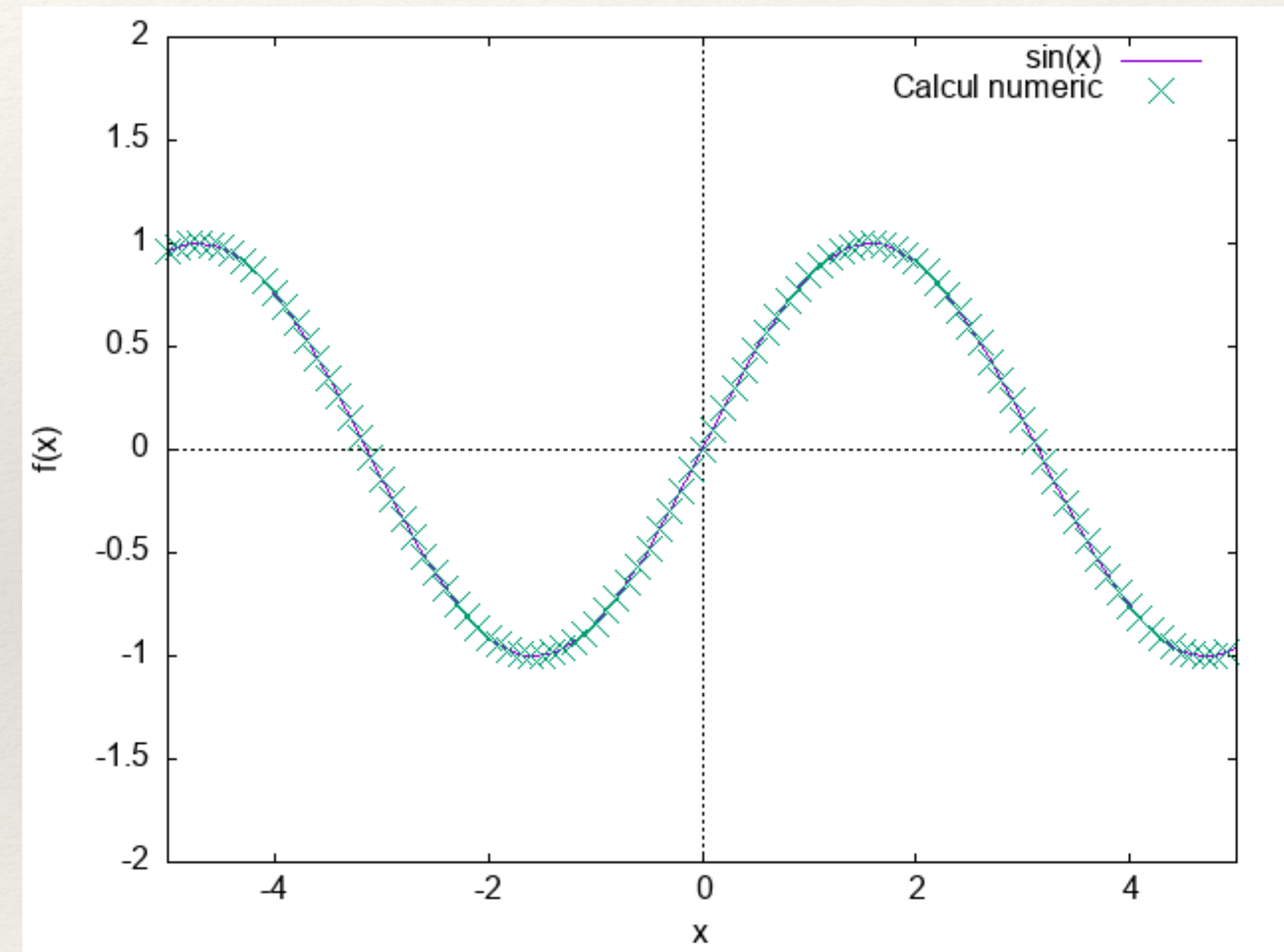
Representem les dades

```
set xrange[-5:5]
set yrange[-2:2]
set xlabel "x"
set ylabel "f(x)"
set xzeroaxis
set yzeroaxis

set term png
set output "fig1.png"
plot sin(x), "sortida.dat" using 1:2 title "Calcul numeric" pointsize 2
```

Per executar cada línia abans hem de fer
gnuplot [enter]

És més convenient escriure les línies en un arxiu i executar com
gnuplot plot.gnu

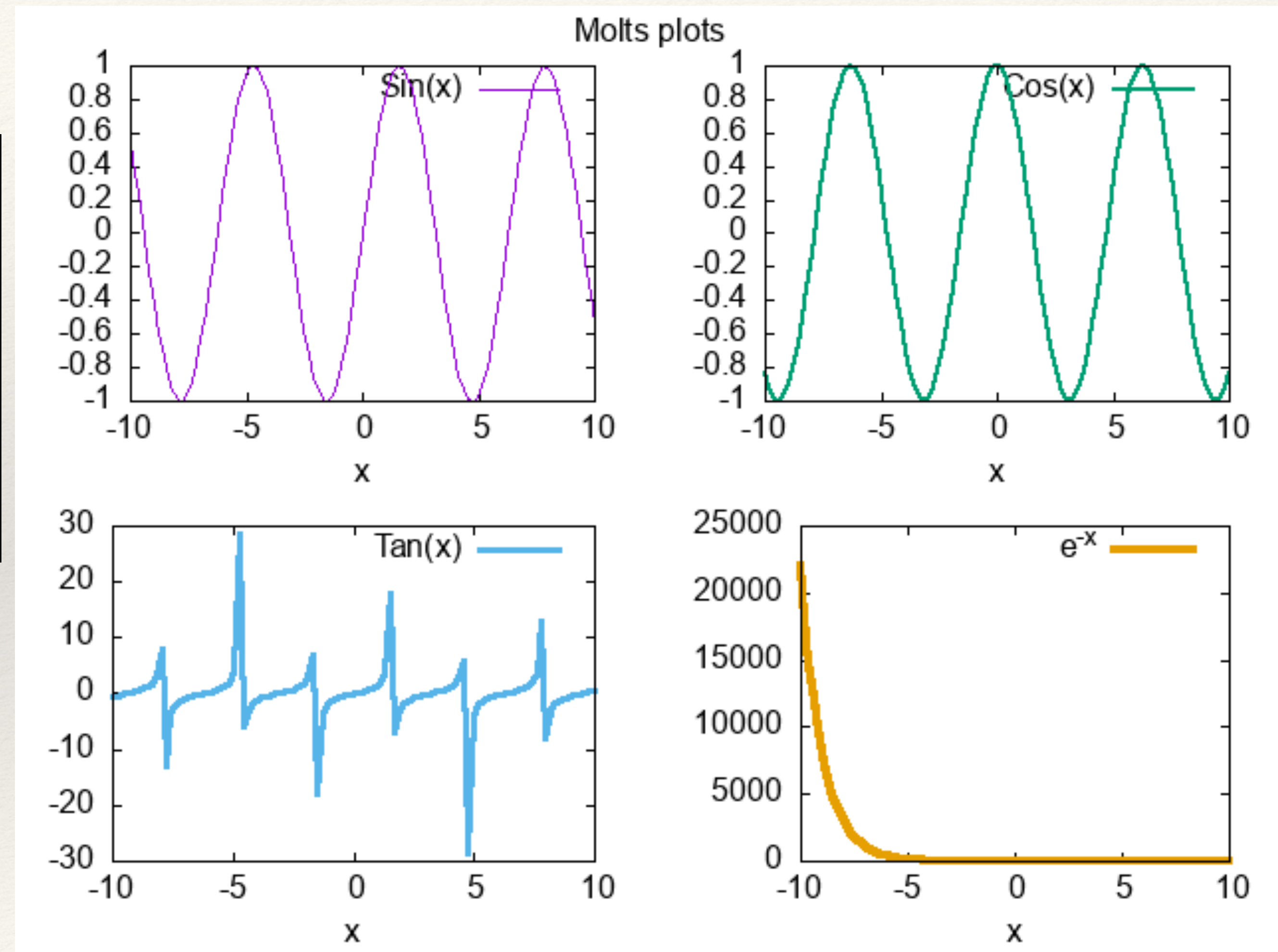


Gnuplot opcions

- ❖ `with lines` uneix els punts amb línies
- ❖ `with points` es poses només punts
- ❖ `with linespoints` es posen punts i línies rectes
- ❖ `linewidth k` escull el gruix de la línia (k és un número)
- ❖ `linetype k` determina el tipus de línia (i també el color)

Exemple multiplot

```
set term png
set output "multiplot.png"
set multiplot title "Molts plots" layout 2,2
set xlabel "x"
plot sin(x) title "Sin(x)" linewidth 1 linetype 1
plot cos(x) title "Cos(x)" linewidth 2 linetype 2
plot tan(x) title "Tan(x)" linewidth 3 linetype 3
plot exp(-x) title "e^{-x}" linewidth 4 linetype 4
unset multiplot
```



Exemple dades

```
set xzeroaxis
set yzeroaxis
f(x) = sin(x)*exp(-abs(x))
set xrange[-4:4]
set yrange[-0.5:0.5]
set xlabel "x" font ",18"
set ylabel "f(x)" font ",18"
set term png
set output "errorbars.png"
plot f(x), "dades.dat" u 1:2:3 with errorbars t"Mesures" ps 2
```

