

# Post P2, pre P3

Bruno Juliá-Díaz (brunojulia@ub.edu)

Dpto. Física Quàntica i Astrofísica

Facultat de Física

**Universitat de Barcelona**

**Curso 2016/2017**

# Post Práctica 2, Consideraciones globales

- + Estructura de código
  - + Uso de nombres razonables para variables
  - + Encabezado + Comentarios en el código
  - + Resultados correctos / incorrectos
  - + Uso de scripts de gnuplot para generar gráficas
  - + Utiliza precisión doble
- + Haz el programa lo más general posible, e.g. numero de pistones, número de puntos, número de tiempos, tiempo final, etc, etc, etc.

# Fallos interesantes (1)

```
open (20, file="P2-2016-res1-b1.dat")
  tt= 0.0
10  if (tt.le.5.0) then
    call posiciones(L,w,tt,Xs)
    write(20,2) tt, Xs
    tt = tt + 0.1
    goto 10
  end if
close(20)
```

C Ahora abrimos el fichero creado anteriormente y guardamos todos los valores  
C en vectores, es decir, un vector por cada columna

```
open(21, file="P2-2016-res1-b1.dat")
i=0
do i=0,50
  read(21,2) ti(i),y_1(i),y_2(i),xi(i),y_4(i),y_5(i),y_6(i)
  print*,i
2  format(f9.4,2x,f10.4,2x,f10.4,2x,f10.4,2x,f10.4,2x,f10.4,2x,f10.4)
end do
close(21)
```

```
At line 34 of file P2-2016.f (unit = 21, file = 'P2-2016-res1-b1.dat')
Fortran runtime error: End of file
```

```
Error termination. Backtrace:
```

```
#0  0x10bd6eb99
#1  0x10bd6f855
#2  0x10bd6ffb9
#3  0x10be3815b
#4  0x10be386f7
#5  0x10be361fa
#6  0x10be39834
#7  0x10bd658e8
#8  0x10bd65c4e
```

# Fallos interesantes (2)

```
SUBROUTINE posiciones(L,w,tk,xx)
IMPLICIT NONE
INTEGER i,ncoses,t
PARAMETER (ncoses=5)
REAL xx,tk
DIMENSION xx(81)
DOUBLE PRECISION L,w,radi,phi,fase
fase=phi(i)
```

```
DO i=1,ncoses
    CALL radius(i,L,radi)
    xx(i)=radi*cos(w*t+fase)+sqrt(L**2-((radi*sin(w*t+fase))**2))
ENDDO

WRITE (*,*) (xx(i), i=1,ncoses)
RETURN
END
```

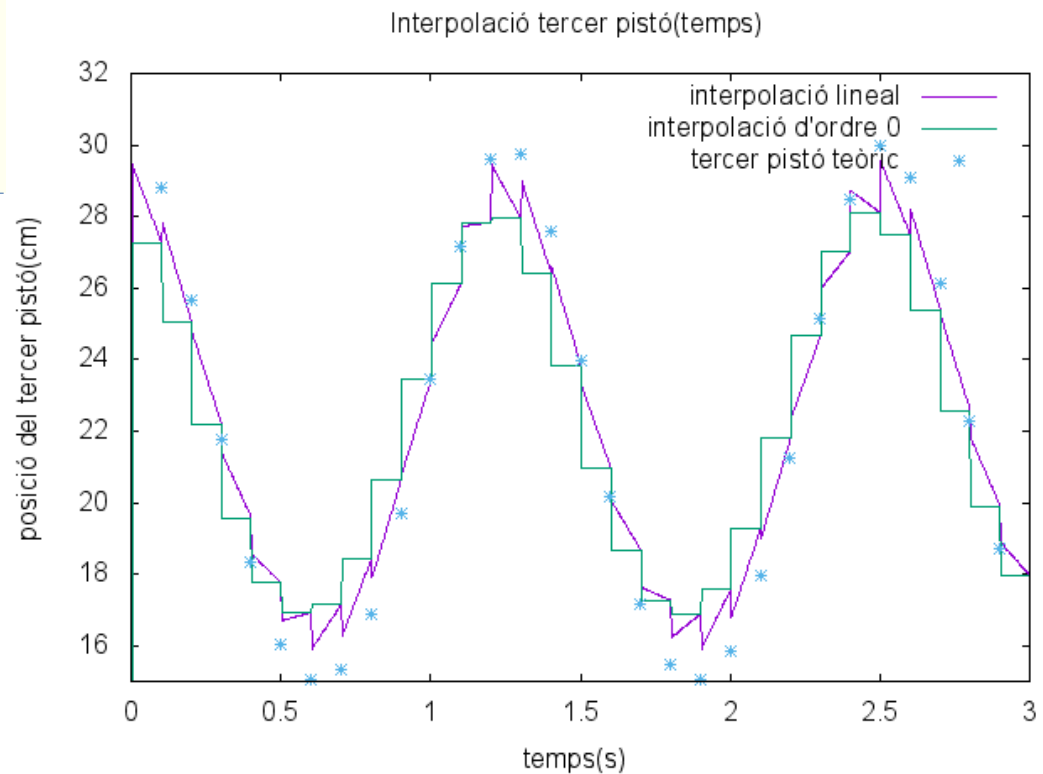
```
SUBROUTINE radius(i,L,radi)
IMPLICIT NONE
INTEGER i,ncoses
PARAMETER (ncoses=5)
DOUBLE PRECISION L,w,radi,a
L=18.5
w=5.d0
a=0.5
DO i=1,ncoses
    radi=L*((i)**(-1))-a
ENDDO
RETURN
END
```

Produce un loop infinito....

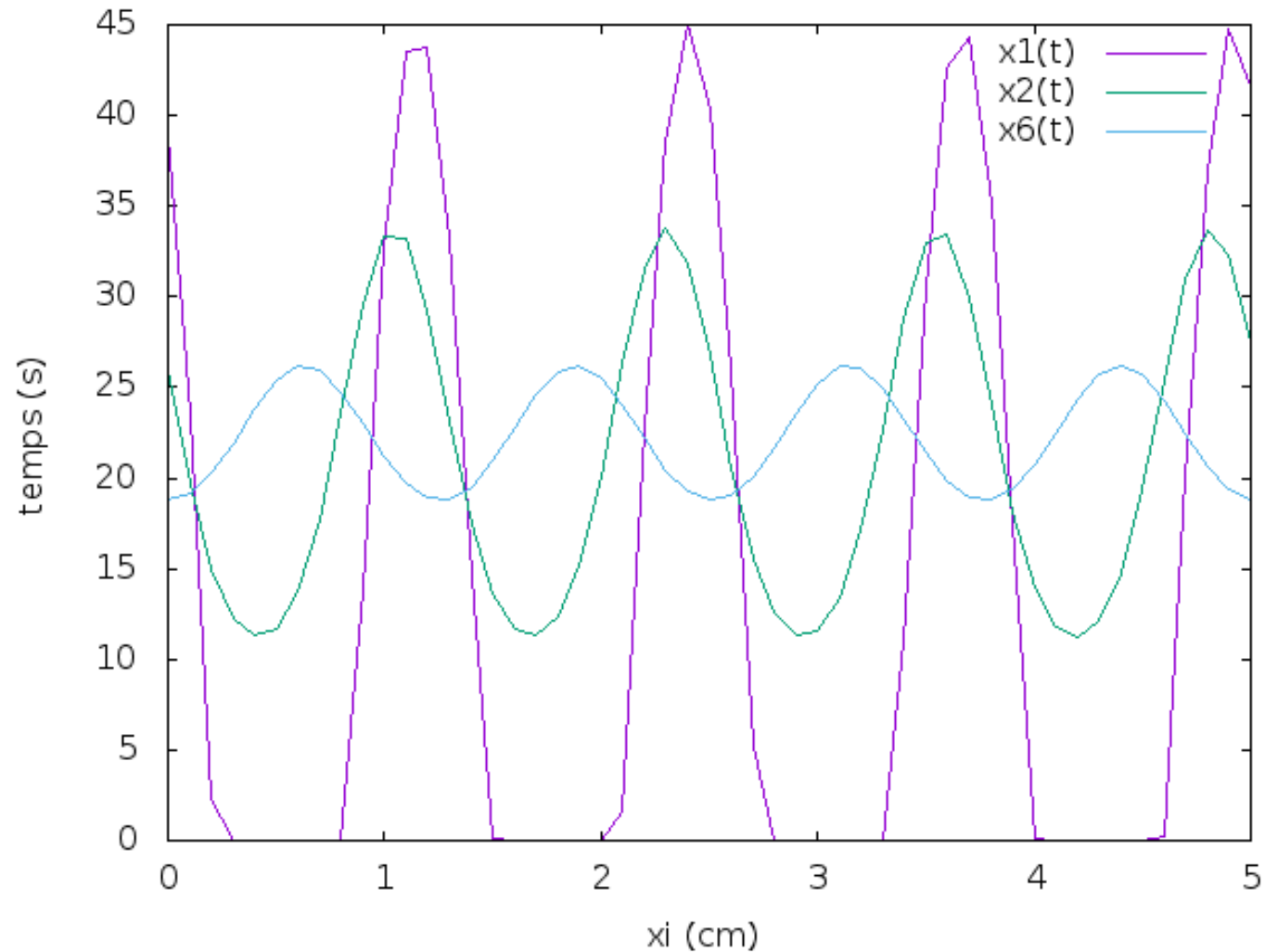
# Fallos interesantes (3)

```
function xinterpo(t2)
parameter(kmax=51)
dimension XI(51)
dimension TI(51)
COMMON/POSI/TI,XI
do k=1,kmax
dif= abs(ti(k)-t2)
if ((dif).LT.(0.1)) then
xinterpo=xi(k)+((t2-ti(k))*(xi(k+1)-xi(k))/(ti(k+1)-ti(k)))
endif
enddo

return
end function
```

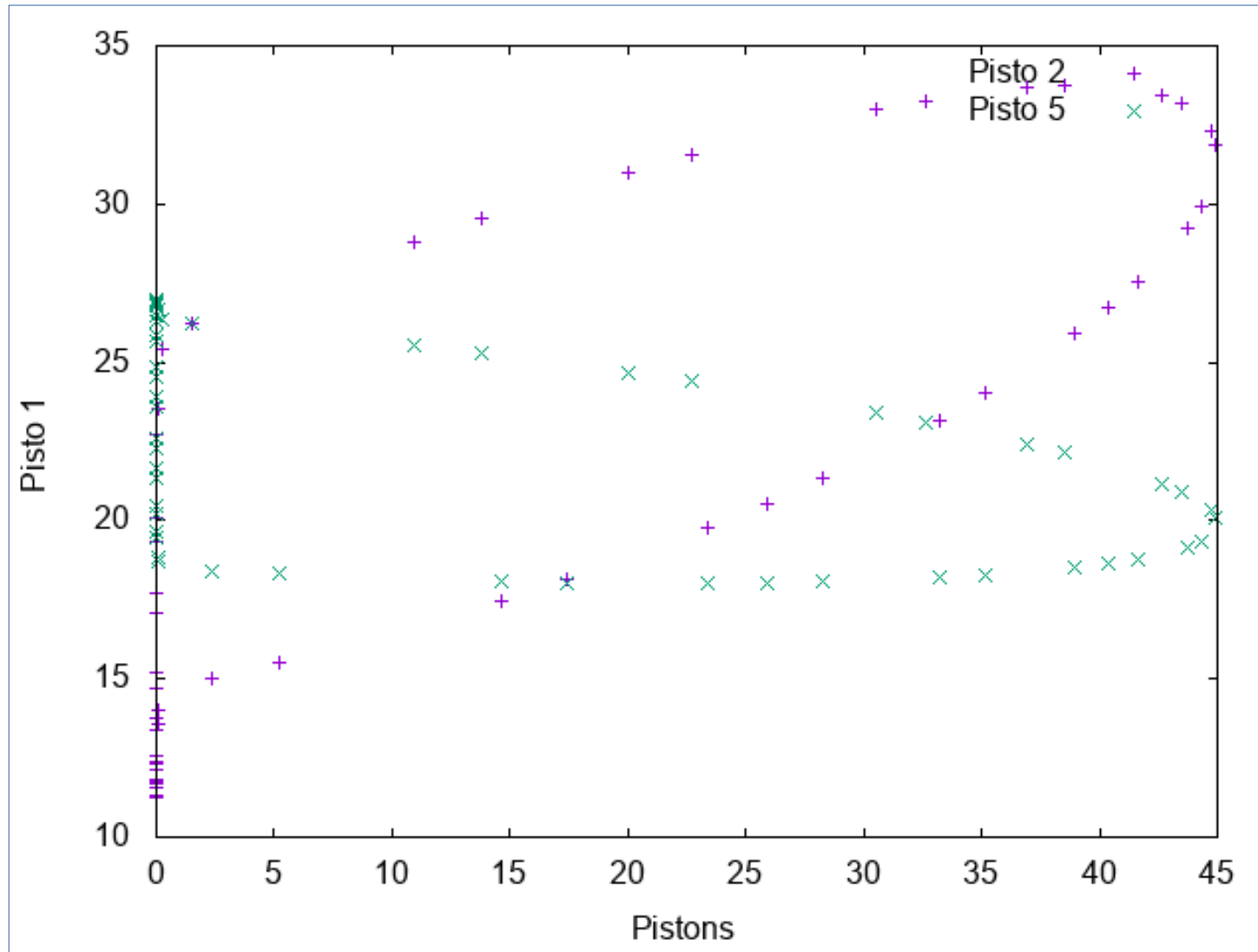


# Figuras (1)



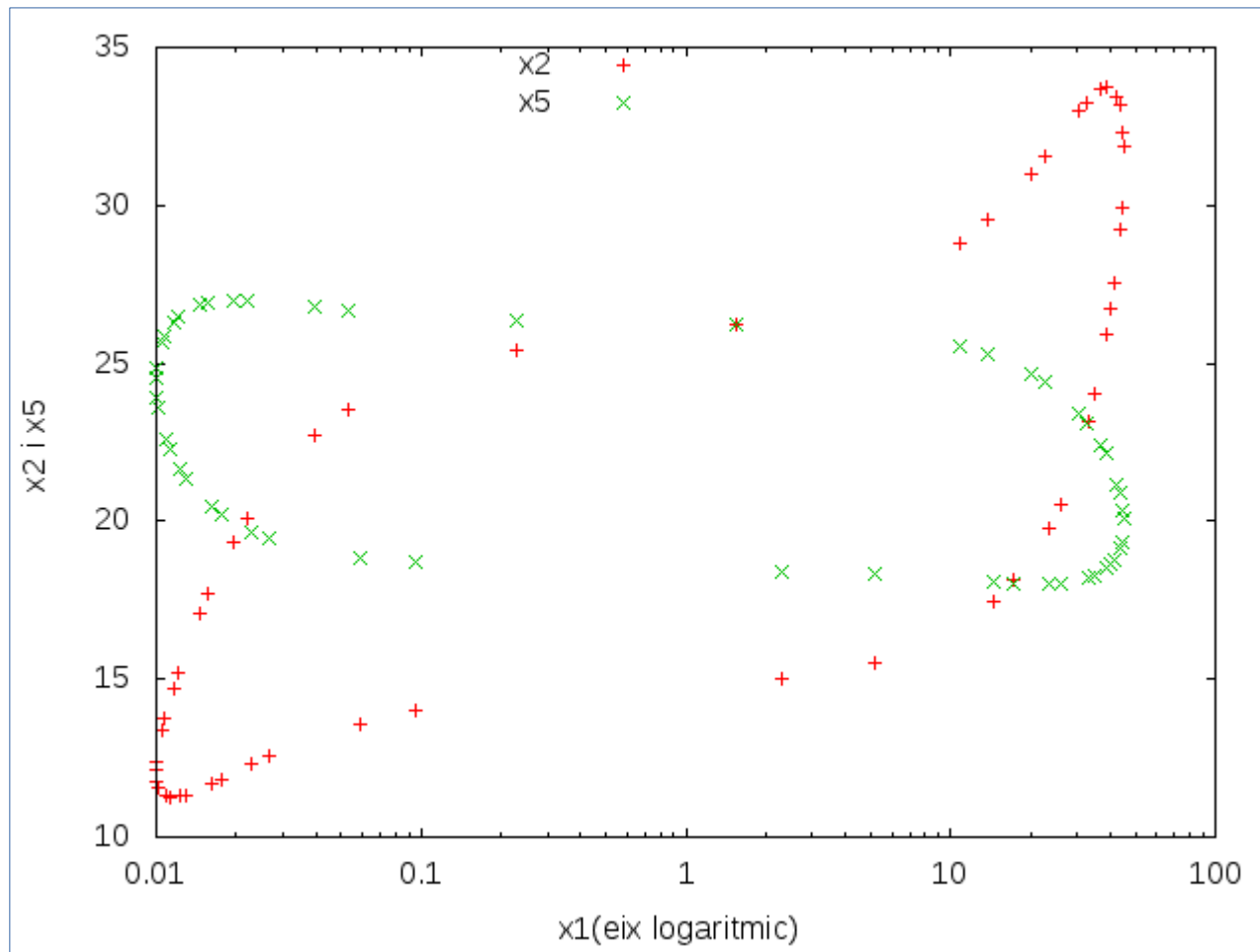
> Unidades  
> Los puntos sobre  
el eje "x" no se ven...

# Figuras (1)



> Unidades  
> Las etiquetas  
están sobre los  
puntos

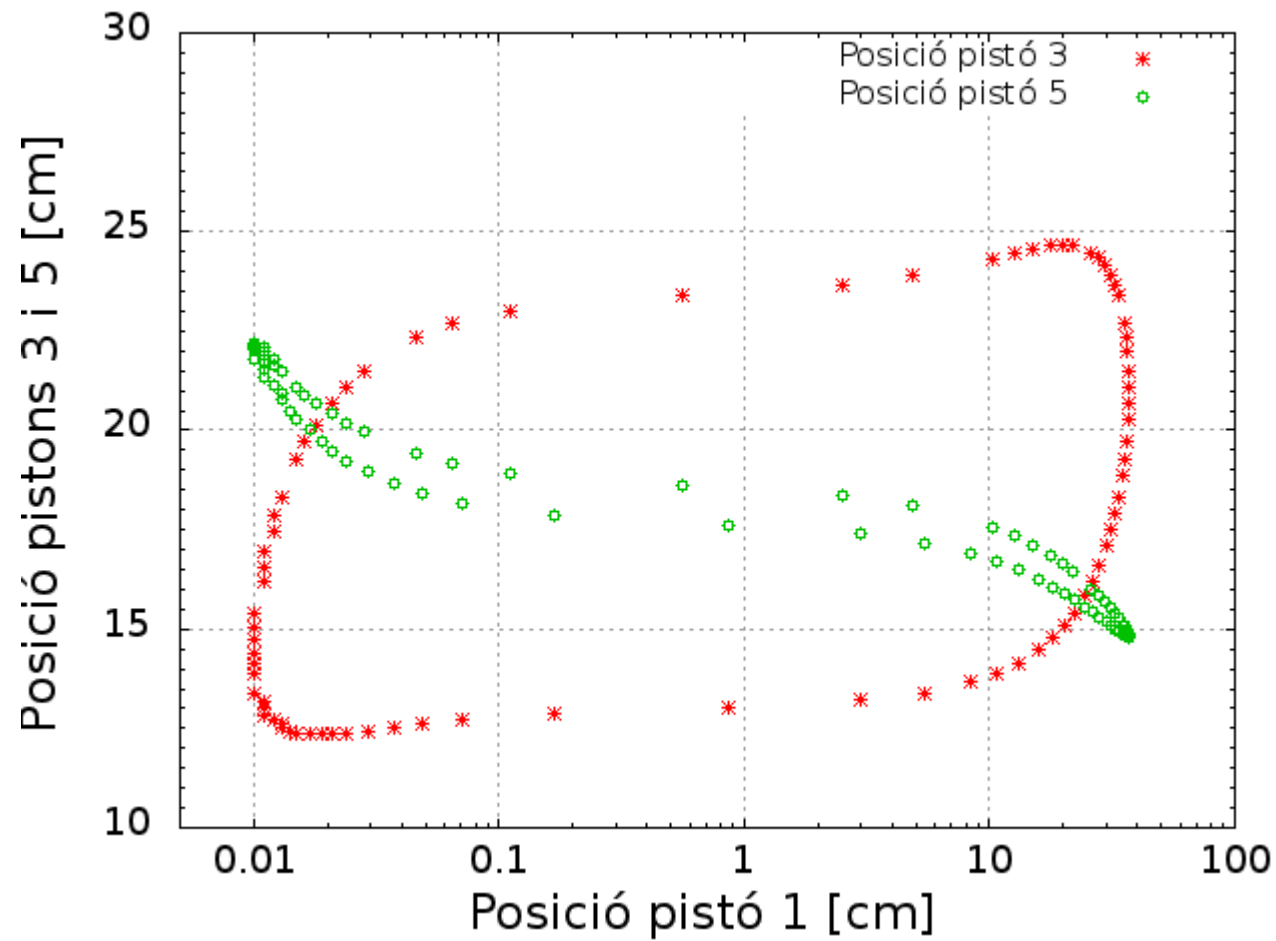
# Figuras (2)



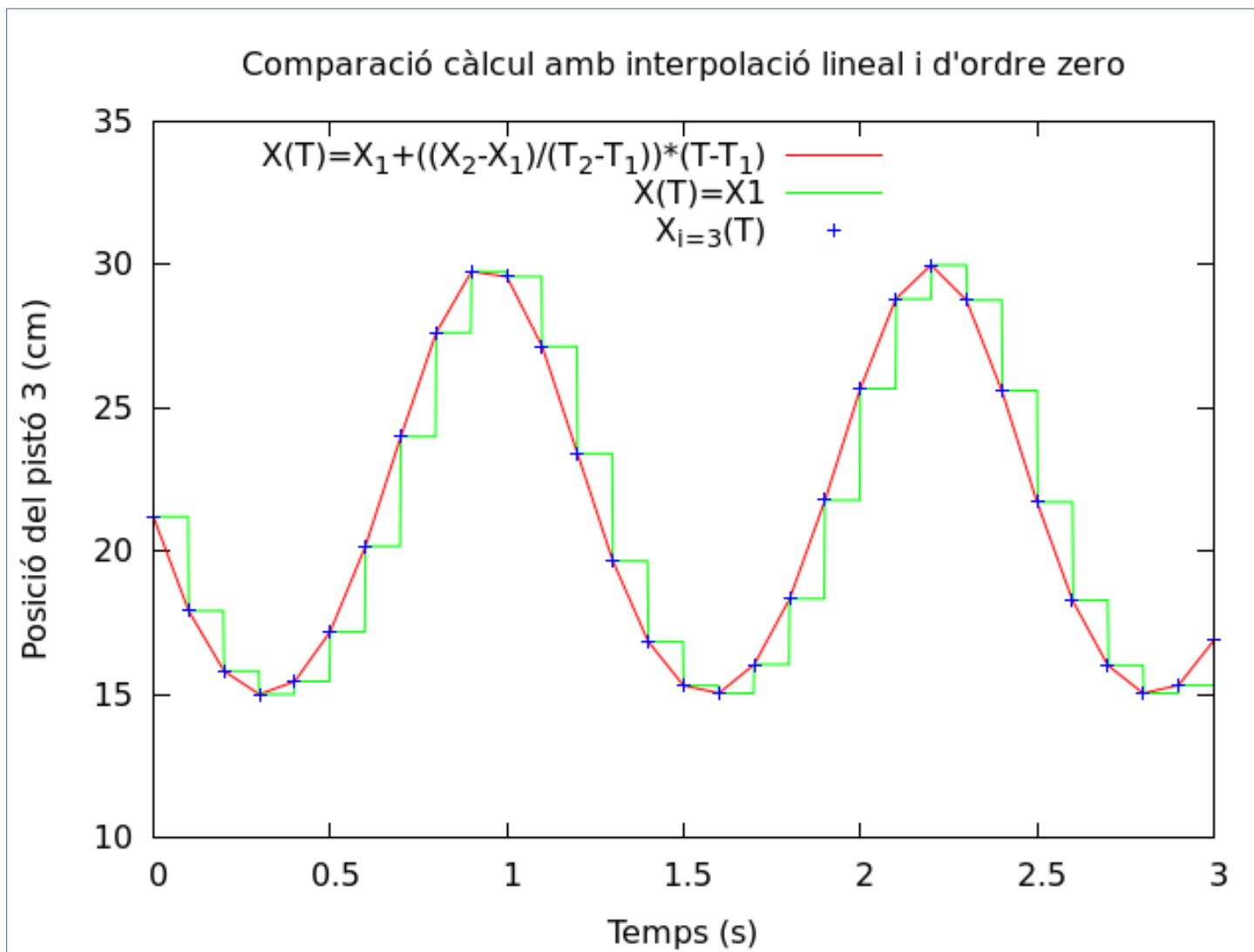
> Unidades  
> Que el eje sea  
en escala  
logarítmica no es  
necesario  
recordarlo en la  
etiqueta



# Figuras (3)

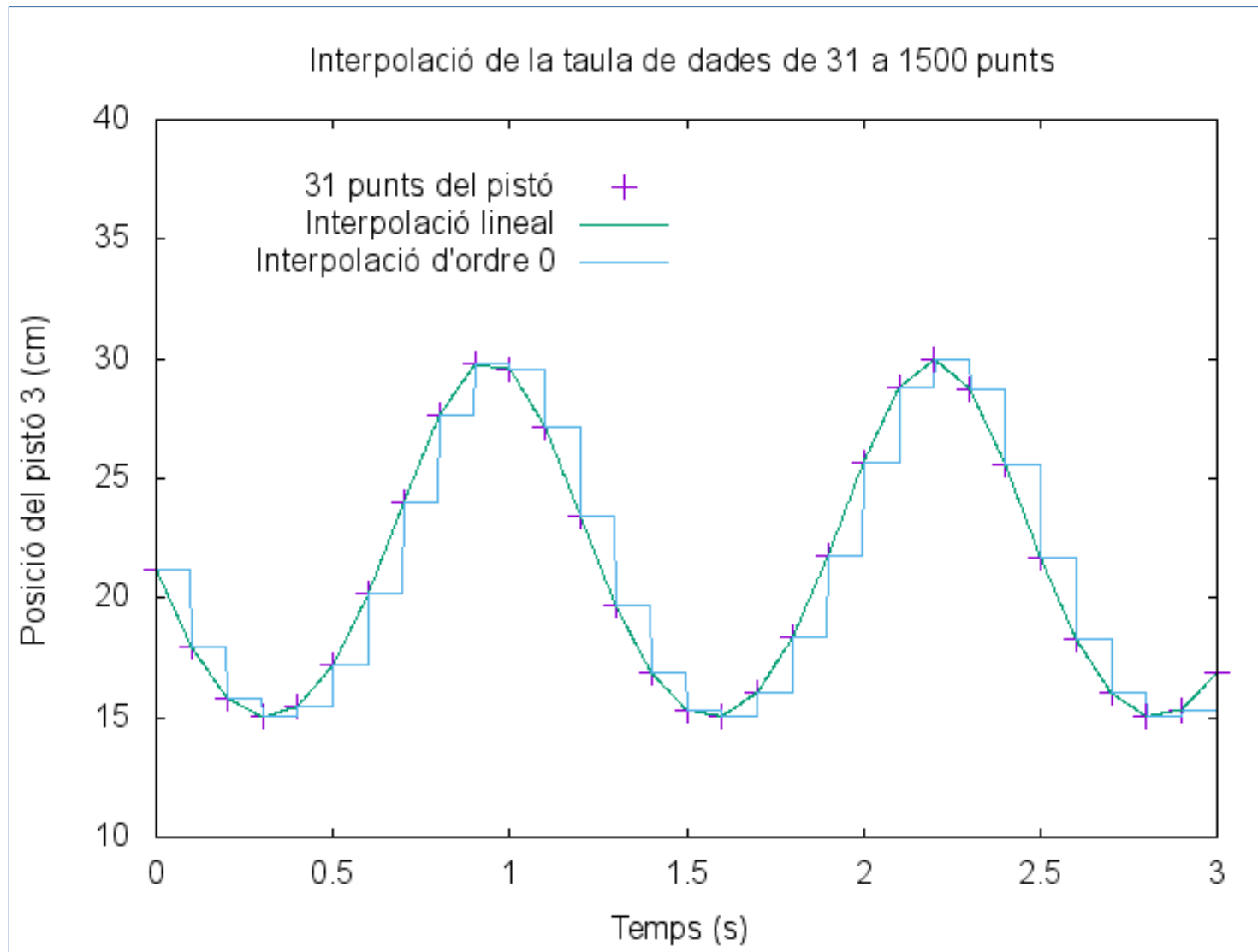


# Figuras (3)



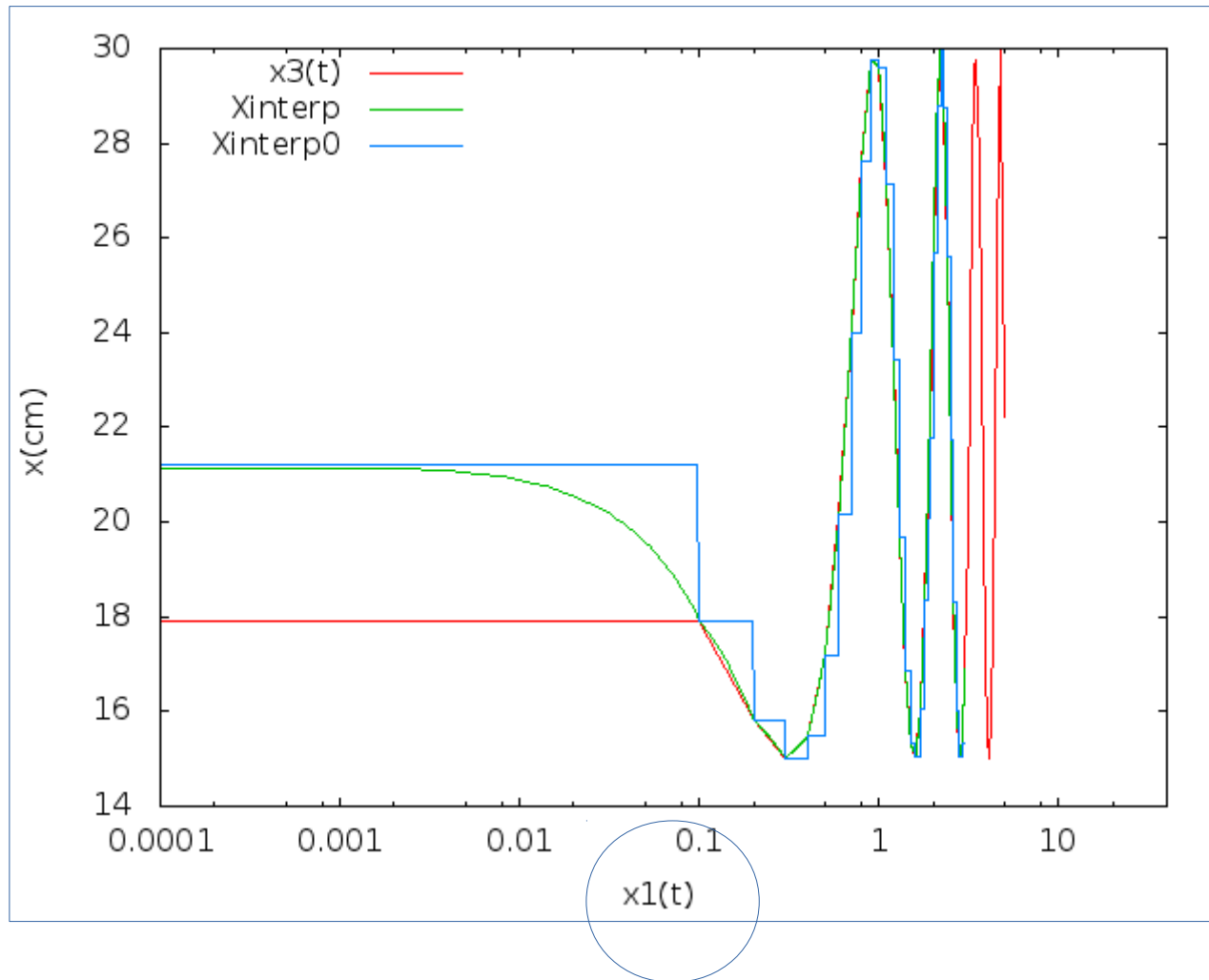
> Etiquetas de difícil comprensión

# Figuras (3)



> Figura bien hecha  
> pero se pedían 51 puntos...

# Figuras (1)



El uso de la escala  
logarítmica  
dificulta la  
comprensión en  
este caso

# Opciones de compilacion

*A vegades tenim problemes per trobar els errors en un codi fortran, aquí teniu unes opcions que faran que el codi executable sigui força més lent però comprobaran internament un munt de coses que us poden ajudar molt a trobar l'error. Enrecordeu-vos de no fer-les servir si ja heu trobat els errors (el programa és de lent a molt més lent)*

**gfortran -g -fbounds-check -Wall -fbacktrace**

*Altres vegades volem posar l'accelerador a tope per tal que el programa corri el màxim possible (... depen del compilador). En aquest cas unes bones opcions per a 'gfortran' són:*

**gfortran -O3 -msse2 -ffash-math -floop-parallelize-all -ftree-vectorize**

*No tots els ordinadors ni en totes les versions de gfortran funcionen!!!!!! SI alguna no funciona treieu-la a veure si amb les altres sí. La que és més candidata no funcionar és '-msse2'.*

*Aquestes van bé en un ordinador amb Intel i7 , Linux i gfortran versio 4.8. Altres versions de gfortran s'hauria d'anar buscant.*

<https://gcc.gnu.org/onlinedocs/gfortran/Option-Summary.html>

# Interpretando al compilador

(7 líneas de encabezado)

```
implicit none
integer i,j,k
i=2
  call suma2(i)
print*,i,j
end
```

```
subroutine suma2(x, xp2)
implicit none
integer x, xp2
  xp2=x+2
end
```

Número de línea

>gfortran err0.f

err0.f:11:72: Warning: Missing actual argument for argument 'xp2' at (1)

# Interpretando al compilador

**>gfortran -Wall err0.f**

```
f951: Warning: Nonconforming tab character in column 1 of line 1 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 8 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 9 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 10 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 11 [-Wtabs]
f951: Warning: Nonconforming tab character in column 2 of line 12 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 13 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 15 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 16 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 17 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 18 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 19 [-Wtabs]
f951: Warning: Nonconforming tab character in column 1 of line 20 [-Wtabs]
err0.f:11:72: Warning: Missing actual argument for argument 'xp2' at (1)
err0.f:9:19: Warning: Unused variable 'k' declared at (1) [-Wunused-variable]
err0.f:11:72: Warning: Missing actual argument for argument 'xp2' at (1)
```

# Interpretando al compilador

```
implicit none
integer i,j,k
real p
i=2
  call suma2(i,p)
print*,i,j
end
subroutine suma2(x, xp2)
implicit none
integer x, xp2
      xp2=x+2
end
```

**>gfortran err1.f**

err1.f:12:20: Warning: Type mismatch in argument 'xp2' at (1); passed REAL(4) to INTEGER(4)



# Error típico de segmentación

```
real x(3)
do i=1,4
  x(i)=i
  print*,i
enddo
end
```

~

**>gfortran err5.f**

**> a.out**

1

2

3

1082130432

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

Backtrace for this error:

#0 0x7FB4CCEA9E08

#1 0x7FB4CCEA8F90

#2 0x7FB4CCAFA49F

#3 0x4007B8 in MAIN\_\_ at err5.f:?

Bus error (core dumped)

# Error típico de segmentación (2)

```
real x(3)
do i=1,4
  x(i)=i
  print*,i
enddo
end
```

**>gfortran -fbounds-check err5.f**

**> a.out**

1  
2  
3

At line 4 of file err5.f

Fortran runtime error: Index '4' of dimension 1 of array 'x' above upper bound of 3

# Volvamos al fallos interesante (2)

```
SUBROUTINE posiciones(L,w,tk,xx)
IMPLICIT NONE
INTEGER i,ncoses,t
PARAMETER (ncoses=5)
REAL xx,tk
DIMENSION xx(81)
DOUBLE PRECISION L,w,radi,phi,fase
fase=phi(i)

DO i=1,ncoses
    CALL radius(i,L,radi)
    xx(i)=radi*cos(w*t+fase)
+sqrt(L**2-((radi*sin(w*t+fase))**2))
ENDDO

WRITE (*,*) (xx(i), i=1,ncoses)
RETURN
END
```

# Volvamos al fallos interesante (2)

```
>gfortran P2-2016.f  
>  
> gfortran -Wall P2-2016.f  
>P2-2016-c2.f:41:15:
```

Compilador “desnudo” no dice nada...

```
xx(i)=radi*cos(w*t+fase)+sqrt(L**2-((radi*sin(w*t+fase))**2))  
1
```

Warning: Possible change of value in conversion from REAL(8) to REAL(4) at (1) [-Wconversion]  
P2-2016-c2.f:59:12:

```
tk=a/10.0d0  
1
```

Warning: Possible change of value in conversion from REAL(8) to REAL(4) at (1) [-Wconversion]  
P2-2016-c2.f:31:34:

```
SUBROUTINE posiciones(L,w,tk,xx)  
1
```

Warning: **Unused dummy argument 'tk'** at (1) [-Wunused-dummy-argument]  
P2-2016-c2.f:49:31: Warning: Unused variable 't' declared at (1) [-Wunused-variable]  
P2-2016-c2.f:41:0:

```
xx(i)=radi*cos(w*t+fase)+sqrt(L**2-((radi*sin(w*t+fase))**2))  
^
```

**Warning: 't' may be used uninitialized** in this function [-Wmaybe-uninitialized]

# Pre-práctica 3

En la práctica 3:

- > Calcularemos integrales definidas
- > Utilizaremos los métodos:
  - + trapecios
  - + Simpson
- > Énfasis en el control del error cometido.
- > Utilizaremos “external” para la función a integrar

# Pre-práctica 3, integración numérica

Objectius: `subroutines/functions`, `common blocks`, `if/then`, `mod`, integració

— Nom del programa principal **P3-2016.f**.

Nom de la subrutina de integració **myinte-2016.f**.

Precisió de reals: **double precision**.

Tots els outputs amb 8 cifres significatives, e.g `format(e14.8)`

- 1) Escriu una subrutina **myintegrator**(*a*, *L*, *m*, **im**, **val**, **fcn**) que calculi per a un valor de *a*, *L*, i *m* la integral

$$\int_{a-L/2}^{a+L/2} \text{fcn } dx \quad (0.4)$$

fent servir la regla trapezoïdal composta si **im**=1, o Simpson composta si **im**=2 amb  $2^m$  intervals, i retorni el valor a **val**. Farem servir la funció a integrar com a **external**.

- 2) Escriu una function **mifun**(**x**) que torni el valor de la funció  $f_1$  (**ifu**= 0),  $f_2$  (**ifu**= 1) de l'apartat 3) o  $f_3$  (**ifu**=2) de l'apartat 5). El valor de **ifu** ha d'arribar a la funció mitjançant un **COMMON BLOCK**.

# Precompilar/compilar/linkar

> El fichero myinte-2016.f debe contener solamente la **subrutina myintegrator**

> El fichero P3-2016.f debe contener todo lo demás, incluido el programa principal

> Para compilar y linkar:

Compilamos la subrutina:

> **gfortran -c myinte-2016.f** → genera un fichero myinte-2016.o  
+ Permite buscar errores por separado

Compilamos el programa y linkamos todo:

> **gfortran P3-2016.f myinte-2016.o -o P3-2016.out**

# myintegrator

> INPUT: **a**, **L**, **m**, **im**, **fcn**  
> OUTPUT: **val**

**m**: número de intervalos,  $2^{**}m$

**im**: método, 1 trapecios, 2 Simpson

**a**, **L**: definen el intervalo de integración

**Fcn**: función a integrar, necesitará external desde el programa principal

$$\int_{a-L/2}^{a+L/2} fcn \, dx$$

Program p3-2016

Declaración de variables

**External fcn**

.....

Call myintegrator(a,L,m,im,fcn)

....

end

Function fcn(x)

.....

fcn= ...

end

Utiliza la funcion “mod” cuando programes los algoritmos de trapecios y Simpson



Function mifun(x)

...  
Common / / ifu

3 casos de ifu:

Si ifu=1 ...

Si ifu=2 ...

Si ifu=3 ...

$$f_1(x) = R \sqrt{1 - (x/R)^2},$$

$$\text{Longitud} = \int_{-R}^R \sqrt{1 + f_1'(x)^2} dx \equiv \int_{-R}^R f_2(x) dx.$$

Considera el canvi de variable  $2x = L \sin(t)$  a l'apartat 3b), defineix  $f_3(t)$  com a la funció que cal integrar en  $t$  un cop fet el canvi de variable

# Resultados

3) Amb la subrutina d'1) i la funció de 2) calcula amb  $2^{10}$  intervals les quantitats següents fent servir els dos mètodes i escriu-les dins del fitxer **P3-2016-res1.dat**.

a) La longitud de mitja circumferència de radi  $R = 3.325$  cm,  $f_1(x) = R \sqrt{1 - (x/R)^2}$ , amb la fórmula,

$$\text{Longitud} = \int_{-R}^R \sqrt{1 + f_1'(x)^2} dx \equiv \int_{-R}^R f_2(x) dx. \quad (0.5)$$

b) La masa total d'una barra de longitud  $L = 4$  m i densitat lineal

$$f_2(x) = \rho_0 \sqrt{1 - (2x/L)^2} (1 - (2x/L))^3 \quad \text{amb } x \in [-L/2, L/2], \quad (0.6)$$

i  $\rho_0 = 1.42$  (Kg/m).

# Resultados

- 4) Estudia la convergència dels resultats obtinguts a l'apartat 3). Estudia com varia l'error dels càlculs 3a) i 3b) amb la longitud dels subinterval  $h$ . Escriu els resultats en dos fitxers **P3-2016-res2.dat**, **P3-2016-res3.dat** amb tres columnes cadascun:  $h$ , resultat trapezis, resultat Simpson, per a 3a) i 3b), respectivament. Fes dues gràfiques **P3-2016-fig1.png** (3a) i **P3-2016-fig2.png** (3b) amb l'error comès en funció d' $h$  ( $m = 2, \dots, 20$ ), comparat amb un ajust "a ull" amb el comportament esperat per a cada mètode. Fes servir escala logarítmica per a les ordenades.
- 5) Considera el canvi de variable  $2x = L \sin(t)$  a l'apartat 3b), defineix  $f_3(t)$  com a la funció que cal integrar en  $t$  un cop fet el canvi de variable i estudia la convergència dels càlculs en funció d' $h$  ( $m = 2, \dots, 20$ ). Escriu els resultats en un fitxer amb 3 columnes:  $h$ , trapezis, Simpson, **P3-2016-res4.dat**. És millor o pitjor que sense el canvi de variable? Fes una gràfica **P3-2016-fig3.png** mostrant la convergència dels resultats comparant els càlculs amb i sense fer-ne el canvi de variable per trapezis i Simpson.

# Comparación trapecios vs Simpson

