

Existen diversas definiciones de los términos en la literatura:

- programación **concurrente**
- programación **paralela**
- programación **distribuida**

Una posible distinción (según mi opinión) es:

- la *programación concurrente* se dedica más a **desarrollar** y **aplicar** conceptos para el uso de recursos en paralelo (desde el punto de vista de varios actores)
- la *programación en paralelo* se dedica más a **solucionar** y **analizar** problemas bajo el concepto del uso de recursos en paralelo (desde el punto de vista de un sólo actor)

Otra posibilidad de separar los términos es:

- un programa concurrente define las **acciones** que se pueden **ejecutar simultáneamente**, es decir, están en progreso
- un programa paralelo es un programa concurrente diseñado de ser **ejecutado en hardware paralelo**
- un programa distribuido es un programa paralelo diseñado de ser **ejecutado en hardware distribuido**, es decir, donde varios procesadores no tengan memoria compartida, tienen que intercambiar la información mediante de transmisión de mensajes/datos.

Intuitivamente, todos tenemos una idea básica de lo que significa el concepto de concurrencia.

# lo más simple

```
#include <cstdlib>
#include <cmath>
#include <iostream>

int main(
    int argc,
    char* argv[]
) {
    if(argc<2) {
        std::cout<<"# need argument\n";
        std::cout<<"# no data generated\n";
        return 1;
    }
    const double dmax(strtod(argv[1],0));
    for(double d(0.0); d<dmax; d+=0.1)
        std::cout<<d<<" " <<1.0-exp(-d)<<"\n";
    return 0;
}
```

- compilación para depuración:

```
g++ data.cpp -o data
```

- ejecución del programa con salida a fichero:

```
./data 5 > D.gdt
```

- visualización de resultados con gnuplot:

```
plot "D.gdt" w lp lw 3
```

(<http://www.gnuplot.info>, <http://www.gnuplotting.org/>  
o cualquier otro programa que podéis manejar)

- todos estos comandos agrupamos en scripts correspondientes
  - `compile.sh` para compilar
  - `run.sh` para ejecutar
  - `D.gpl` para visualizar

- miramos otros datos

0 2

0.1 3

0.2 4

0.3 5

0.4 6

0.5 7

0.6 8

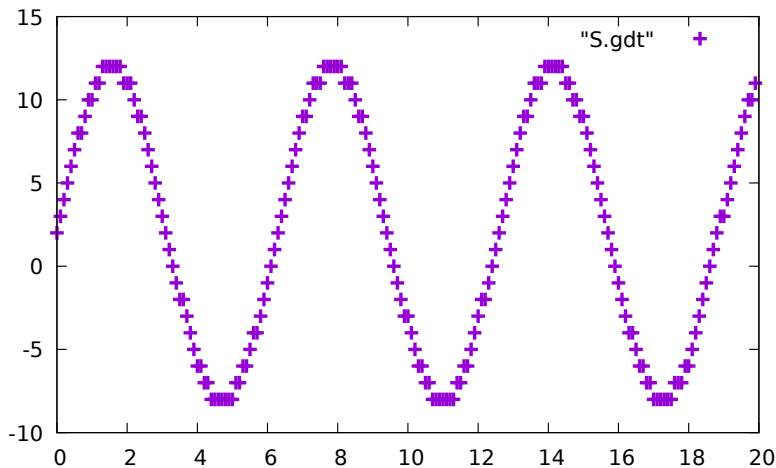
0.7 8

0.8 9

0.9 10

- y aproximamos los datos con una función adecuada

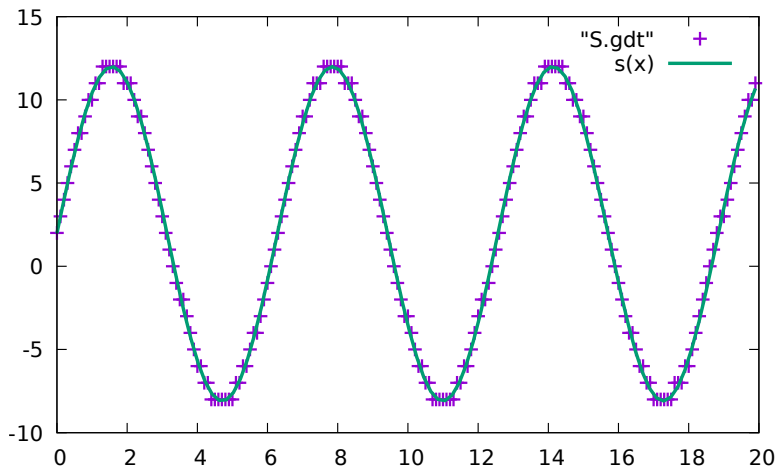
# visualización de los datos





# aproximación con función *adecuada*

$$s(x) = a_0 \cdot \sin(x - a_1) + a_2$$



## aproximación con tal función en gnuplot

```
# start values for the fitting of the underlying function
a_0=10.0
a_1=1.0
a_2=1.0

#our parameterized exponential function
s(x) = a_0*sin(x-a_1)+a_2

# fit data to the curve
fit [0:20] s(x) "S.gdt" u 1:2 via a_0,a_1,a_2

# plotting the data with fit
plot "S.gdt" w lp lw 2, s(x) lw 3
```

# info del algoritmo Marquardt-Levenberg de gnuplot

```
iter      chisq      delta/lim  lambda   a_0          a_1          a_2
  0 1.0112498375e+04   0.00e+00  5.80e+00  1.000000e+01  1.000000e+00  1.000000e+00
  1 2.3022645454e+03  -3.39e+05  5.80e-01  5.400103e+00  1.544881e-01  1.813287e+00
  2 1.9863785699e+02  -1.06e+06  5.80e-02  9.917101e+00 -1.381719e-01  1.949054e+00
  3 1.6103874361e+01  -1.13e+06  5.80e-03  9.951574e+00 -2.263053e-03  1.949272e+00
  4 1.5268905754e+01  -5.47e+03  5.80e-04  1.004252e+01 -3.565976e-03  1.949272e+00
  5 1.5268904319e+01  -9.40e-03  5.80e-05  1.004252e+01 -3.554176e-03  1.949272e+00
iter      chisq      delta/lim  lambda   a_0          a_1          a_2
```

After 5 iterations the fit converged.

final sum of squares of residuals : 15.2689

rel. change during last iteration : -9.39583e-08

```
degrees of freedom      (FIT_NDF)                : 197
rms of residuals        (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.278401
variance of residuals   (reduced chisquare) = WSSR/ndf : 0.0775071
```

```
Final set of parameters          Asymptotic Standard Error
=====                          =====
a_0          = 10.0425           +/- 0.0282      (0.2808%)
a_1          = -0.00355418      +/- 0.002749   (77.36%)
a_2          = 1.94927          +/- 0.01974    (1.013%)
```

correlation matrix of the fit parameters:

```
          a_0    a_1    a_2
a_0      1.000
a_1      0.037  1.000
a_2     -0.037  0.064  1.000
```

- podemos **generar** ficheros con datos
- podemos **visualizar** los datos con un programa (p.ej. gnuplot)
- podemos **aproximar** los datos con funciones adecuadas
- podemos **interpretar** los datos con (cierto) sentido
- podemos trabajar con argumentos por línea de comando
- usamos *scripts* si es posible