

# Aplicacions Híbrides, Matlab i Programari Lliure

Guillem Borrell Nogueras<sup>1</sup>

guillemborrell@gmail.com

<http://torroja.dmt.upm.es/guillem/blog/>

<sup>1</sup>Englobe Technologies  
Laboratorio de CFD  
ETS Ingenieros Aeronáuticos  
Universidad Politécnica de Madrid

Juliol 2007

# Continguts

Aplicacions  
híbrides, Matlab i  
Programari Lliure

Guillem Borrell i  
Nogueras

## Aplicacions híbrides

Aplicacions  
híbrides

### Octave

Exemples

Octave  
Exemples

### Python

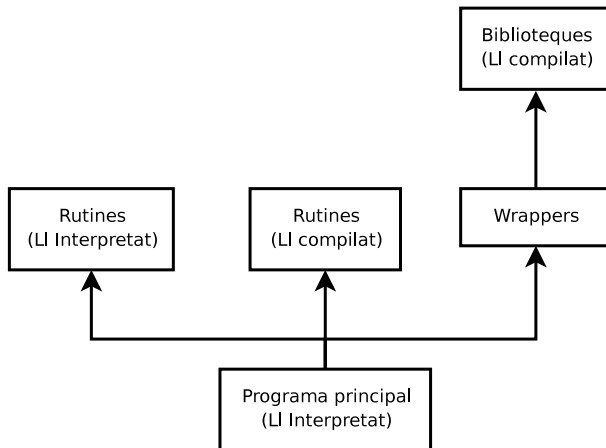
El llenguatge Python  
Exemples

Python  
El llenguatge Python  
Exemples

# Aplicacions híbrides

Aplicacions  
híbrides, Matlab i  
Programari Lliure

Guillem Borrell i  
Nogueras



Aplicacions  
híbrides

Octave  
Exemples

Python  
El llenguatge Python  
Exemples

# Per què serveixen?

- ▶ Crear programes complexos
  - ▶ Amb molta velocitat d'execució
  - ▶ Sense perdre versatilitat
  - ▶ Mínim cost de manteniment
- ▶ Crear petits programes
  - ▶ Que s'han d'utilitzar un altre cop en un futur
  - ▶ On es fa un procés no òptim en un llenguatge interpretat

## Model de desenvolupament

Quan un llenguatge interpretat es queda petit no es reescriu tot el codi, només les rutines necessàries.

# Per són tan poc populars?

En el fons només hi ha un motiu:  
pocs les coneixen

- ▶ Cal confiar en un llenguatge interpretat
- ▶ Necessiten més coneixements de programació
- ▶ Cal saber més d'un llenguatge de programació
- ▶ Cal saber utilitzar eines específiques
- ▶ El setup de l'entorn de desenvolupament no és trivial

# Què es fa avui?

- ▶ En el ram científic es fa servir Matlab (MEX)
- ▶ Matlab no és Software Lliure
- ▶ El SL disposa de recursos per substituir-lo:
  - ▶ Com a llenguatge interpretat.
  - ▶ Com a llenguatge per aplicacions híbrides.

# Dues preguntes essencials

Aplicacions  
híbrides, Matlab i  
Programari Lliure

Guillem Borrell i  
Nogueras

Aplicacions  
híbrides

Octave  
Exemples

Python  
El llenguatge Python  
Exemples

Els dos usuaris principals de software per ciència i enginyeria són les Empreses i les Universitats.

- ▶ Els cal una alternativa a Matlab?
- ▶ Els interessa la llibertat?

# I les seves respostes

Aplicacions  
híbrides, Matlab i  
Programari Lliure

Guillem Borrell i  
Nogueras

Aplicacions  
híbrides

Octave  
Exemples

Python  
El llenguatge Python  
Exemples

Existeixen motius tècnics i no només filosòfics per  
utilitzar programari lliure.

- ▶ Octave
  - ▶ Busca la compatibilitat amb Matlab.
  - ▶ Projecte madur (+ 15 anys).
  - ▶ GNU/Linux, Windows i MacOS.
- ▶ Python
  - ▶ Llenguatge de programació consistent i madur.
  - ▶ Necessita mòduls per tasques científiques.
  - ▶ NumPy, SciPy, Matplotlib, MayaVI...
  - ▶ Punt de vista complementari a Matlab.
  - ▶ + 20 SO

- ▶ Part del projecte GNU
- ▶ Naixement independent de Matlab
- ▶ Actualment compatible en un 90-95%
- ▶ Escrit en C++ i STL
- ▶ Molt bona comunicació amb C++
- ▶ Més problemes amb C i Fortran.

# Un exemple senzill

## Atractor de Lorentz

## La funció escrita en .m

```
function xdot=lorentz(x,t)
a=10;b=28;c=8/3;
xdot(1,1)=a*(x(2)-x(1));
xdot(2,1)=x(1)*(b-x(3))-x(2);
xdot(3,1)=x(1)*x(2)-c*x(3);
```

El temps necessari per integrar-la: 5.1 segons

# Un exemple senzill

## La mateixa funció escrita en C++ (octfile)

```
#include <octave/oct.h>
DEFUN_DLD (eqlorentz, args, ,
    "Ecuacion de Lorentz en C++")
{
    ColumnVector xdot (3);
    ColumnVector x (args(0).vector_value());
    int a=10;
    int b=28;
    double c=8./3;
    xdot(0) = a*(x(1)-x(0));
    xdot(1) = x(0)*(b-x(2))-x(1);
    xdot(2) = x(0)*x(1)-c*x(2);

    return octave_value (xdot);
}
```

Per integrar-la: 0.36 segons! Per què tanta diferència?  
Ha sigut tan difícil?

- ▶ Octave disposa d'una gran biblioteca per càlcul matricial
  - ▶ Localització d'errors
  - ▶ Tipus específics per vectors, matrius, tensors.
- ▶ C++: OO, perfecte per aplicacions grans
- ▶ Suport per MEX
- ▶ C i Fortran més complicat
- ▶ Increment de velocitat significatiu en:
  - ▶ Bucles.
  - ▶ Temps de reserva per funcions.



Possiblement el millor llenguatge de programació del món

# Eines per aplicacions híbrides

- ▶ Extensions (C)
- ▶ Pyrex
- ▶ SWIG
- ▶ WEAVE (blitz, inline)
- ▶ f2py
- ▶ ctypes

# Un exemple senzill

```
def lorentz(x):  
    xdot=empty(3)  
    a=10  
    b=28  
    c=8./3.  
    xdot[0] = a*(x[1]-x[0])  
    xdot[1] = x[0]*(b-x[2])-x[1]  
    xdot[2] = x[0]*x[1]-c*x[2]  
  
    return xdot
```

# Un exemple senzill amb WEAVE

```
def lorentz_weave(x):
    xdot=empty(3)
    code="""
double a=10;
double b=28;
double c=8./3.;
xdot[0] = a*(x[1]-x[0]);
xdot[1] = x[0]*(b-x[2])-x[1];
xdot[2] = x[0]*x[1]-c*x[2];
return_val=1;

err = weave.inline(code,
    ['x','xdot'],
    type_converters=converters.blitz,
    compiler='gcc')

return xdot
"""
```

- ▶ Un wrapper serveix per utilitzar una llibreria escrita amb un llenguatge compilat
- ▶ Són de gran utilitat per reutilitzar codi
- ▶ La majoria de llenguatges interpretats són en el fons un intèrpret + una sèrie de wrappers
- ▶ La majoria de wrappers necessiten coneixements profunds de C i del propi llenguatge interpretat

# Un wrapper sense fer mal

```
from ctypes import c_int, POINTER
import numpy as np
from numpy.ctypeslib import load_library, ndpointer

def dgesv(N,A,B):
    A = np.asfortranarray(A.astype(np.float64))
    B = np.asfortranarray(B.astype(np.float64))

    cN=c_int(N)
    NRHS=c_int(1)
    LDA=c_int(N)
    IPIV=(c_int * N)()
    LDB=c_int(N)
    INFO=c_int(1)

    lapack=load_library('liblapack.so','/usr/lib/')
```

# Un wrapper sense fer mal II

```
lapack.dgesv_.argtypes=[POINTER(c_int), POINTER(c_int),
                        ndpointer(dtype=np.float64,
                                  ndim=2,
                                  flags='FORTRAN'),
                        POINTER(c_int), POINTER(c_int),

                        ndpointer(dtype=np.float64,
                                  ndim=2,
                                  flags='FORTRAN'),
                        POINTER(c_int), POINTER(c_int)]

lapack.dgesv_(cN, NRHS, A, LDA, IPIV, B, LDB, INFO)
return B

print dgesv(2, np.array([[2, 5], [-6, 3]]), np.array([[1, 0]]))
```

# Conclusions

- ▶ Les aplicacions híbrides són útils en el camp científic i tècnic.
- ▶ Van guanyant popularitat.
- ▶ Matlab es el llenguatge de partida més comú.
- ▶ El SL ofereix alternatives millors.
- ▶ Octave.
- ▶ Python.

Moltes Gràcies.  
Preguntes?