

Transition de Matlab à Python

Brigitte Bidégaray-Fesquet

Laboratoire Jean Kuntzmann
Univ. Grenoble Alpes, France

Journée Python
8 mars 2018



Histoire informatique personnelle

- 1982 Début en BASIC. Premier code industriel pour mon voisin plombier.
- 1986–88 Prépa, programmation en BASIC.
- 1988–91 Licence, maîtrise, DEA en mathématiques : Pascal.
- 1989–90 Cours de licence d'informatique : assembleur, LISP, Prolog, C, ...
- 1991–2001 Recherche : FORTRAN 77, puis 90/95, + Mathematica/Axiom.
 - 1996– Enseignement : Matlab, Maple.
- 1996-2001 Formation continue : Interfaçage Matlab/C ou Fortran.
- 2001–2015 Recherche : Matlab, Maple.
 - 2015 Apprentissage de Python.
- 2015– Recherche : Python



Principales réalisations

Codes de calcul scientifique

- Codes Maxwell-Bloch 1D, 2D, en Fortran 90/95, Matlab, Python.
- Nautil (analyse de von Neumann de schémas numériques), en Maple.
- SPASS, librairie Matlab pour le traitement du signal non uniforme.
V2.1 déposée à l'APP. pySPASS bientôt sur vos écrans !
- Contrôle du diabète, natif Python.

De nombreux très petits codes d'illustration d'articles, pour l'enseignement. . .



Mes raisons pour passer à Python

- 2011 : présentation de SPASS à la Grenoble Innovation Fair, sans connexion internet et donc accès à la licence Matlab. Nécessité de redévelopper pour l'occasion une partie du code pour être compatible avec Octave.
- impossibilité de travailler dans le train, au camping, . . .
- de plus en plus de gens autour de moi qui travaillent en Python.
- apprentissage de Python.
- élégance et clarté du langage Python.



Mon premier code Python

Ben, ce n'était pas du calcul scientifique. . .

- Apprentissage pas dirigé vers des applications en calcul scientifique.
- Langage particulièrement adapté au traitement de chaînes de caractère et à l'utilisation d'expressions régulières.

Application : traitement d'une extraction au format csv des données de la collection HAL de mon laboratoire.

. . . Revenons au calcul scientifique.



Transition Matlab → Python pour Maxwell-Bloch

point de départ

Un code matlab

- déjà pas trop mal écrit en Matlab ;
- avec déjà des structures ressemblant à des objets ;
- de 17 fichiers, une trentaine de fonctions, pour environ 800 lignes de code, dont
 - 8 fichiers de calcul
 - 9 fichiers pour les entrées/sorties
- des fichiers de données pour décrire différents cas-test prédéfinis.

Ce qui ressemble vraiment, les parties de calcul

Matlab

```
function E = SIT(t,Wave)
    freq = (2.*(t+Wave.t_offset)-Wave.t_on)/Wave.t_on;
    if (abs(freq)<1.)    % t_offset < t < t_offset+t_on
        E = Wave.Emax*sin(Wave.omega0*t)/cosh(10*freq);
    else
        E = 0.;
    end
end
```

Python

```
def SIT(t,wave):
    freq = (2.*(t+wave.t_offset)-wave.t_on)/wave.t_on
    if abs(freq) < 1:    # t_offset < t < t_offset+t_on
        return wave.Emax*sin(wave.omega0*t)/cosh(10*freq)
    else:
        return 0.
```

les entrées

Matlab (dans un fichier Matlab)

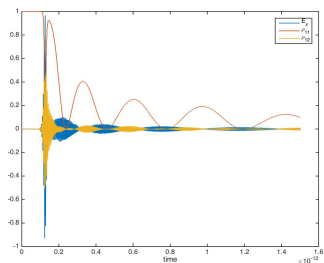
```
% Electromagnetic field – Wave variable  
T.Wave.type = 'sin';  
T.Wave.Emax = .5;           % Maximum amplitude  
T.Wave.omega0 = 1.;        % Wave main frequency
```

Python (dans un fichier JSON)

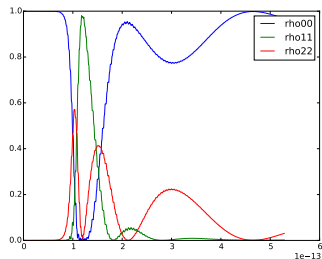
```
"wave": { "type": "sin",  
          "Emax": ".5",  
          "omega0": "1." },
```


les sorties aussi

Matlab



Python



et plus généralement la structure du code : lecture des données dans un fichier, calculs, affichage de courbes.

Ce qui diffère vraiment

Il y a moins de fichiers en Python, car les variables sont regroupées en deux paquets (paramètres et variables) et les fonctions qui les traitent sont devenues des méthodes et mises dans le même fichier.

Comparer plusieurs exécutions est plus naturel

bidouillé en Matlab

```
run SIT;  
T1 = Init(Const,T);  
T1 = MB(T1);  
Plot(T1, Const);
```

Python

```
from TestMB import *  
test = TestMB("SIT")
```



Ce qui diffère vraiment (suite et fin)

On doit mieux connaître les librairies car il va falloir importer à la main les fonctionnalités utilisées

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from math import sin, cos, cosh, pi, exp
from cmath import exp as cexp
from numpy import zeros, eye, dot, diag, ndarray, trace,
                    [meshgrid, sinc, complex
from scipy import linalg
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import pyplot, cm
from time import clock
```



Ne nous mentons pas. . . pySPASS

Ce n'est pas toujours aussi facile.

Pour le code pySPASS

- Librairie initiale en Matlab : environ 7000 lignes sans les commentaires répartis dans environ 80 fichiers.
- Programmation initiale avec des structures presque objet.
- Mais beaucoup plus de types d'objets à définir.

Bref ce n'est pas fini (20% du code migré).

Mais ce travail est agréable à faire.

