

Journée Plume du 4 février 2010

Exemple de migration réussie de MATLAB[®] vers Python

*Logiciels d'acquisition, de visualisation et de traitement de données
Calculs interactifs avec Spyder*

Pierre Raybaut



Club d'Utilisateurs Python

CEA/Centre DAM Île-de-France

Plan



- ☐ Généralités sur **le langage Python**

- ☐ Python au **CEA/DAM Île-de-France**

- ☐ Pourquoi migrer de **MATLAB® vers Python ?**

- ☐ Démonstration des bibliothèques Python internes

 - ☐ Introduction

 - ☐ **guidata** : gestion de jeux de paramètres

 - ☐ **guiqwt** : visualisation de signaux et d'images

- ☐ Exemple d'une migration réussie depuis **MATLAB®** :

 - ☐ Application : logiciel **Spectrométrie neutron**

 - ☐ Développements / calculs interactifs : environnement **Spyder**

Généralités sur le langage Python



- Langage de programmation ayant six caractéristiques essentielles :

dynamique

généraliste

libre

très haut niveau

orienté objet

gratuit

- Autres langages dynamiques (non compilés) : Java, Ruby, **MATLAB®**, **IDL®**
- Autres langages généralistes : **C/C++**, **Fortran**, Java, Ruby
- Autres langages orientés objet par conception : **C++**, Java, Ruby
- Autres langages libres et gratuits : **C/C++**, **Fortran**, Java, Ruby



Source : <http://www.python.org>

Généralités sur le langage Python

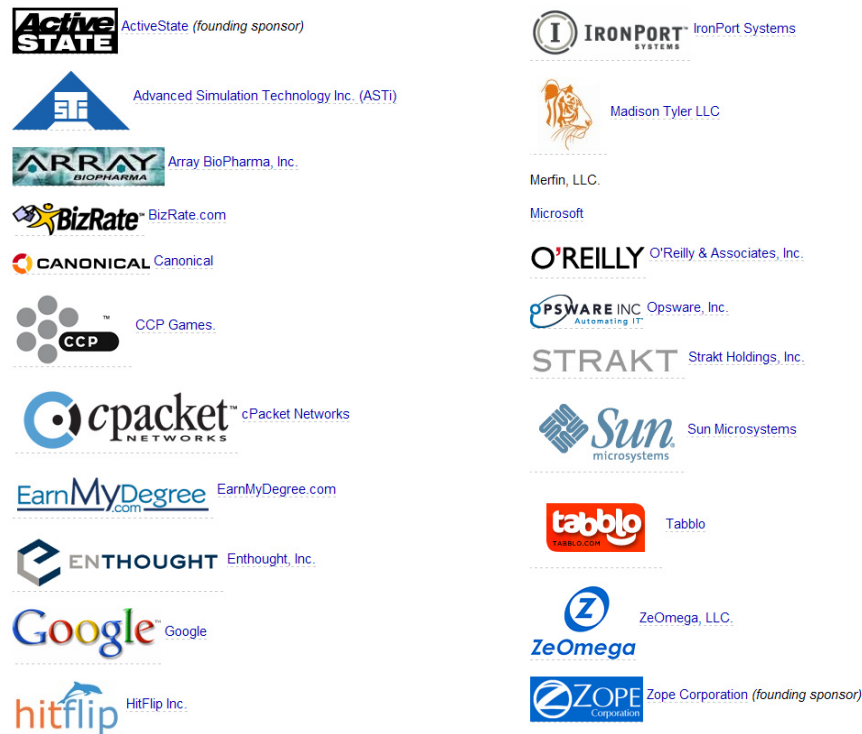
- Python est développé et maintenu par Google, Microsoft, Sun, ...



Python Software Foundation

The mission of the Python Software Foundation is to promote, protect, and advance the Python programming language, and to support and facilitate the growth of the international community of Python programmers.

—from the [Mission Statement page](#)

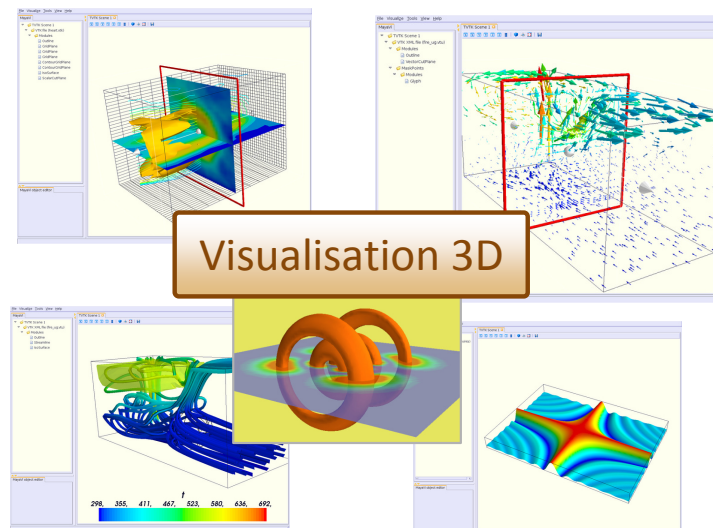
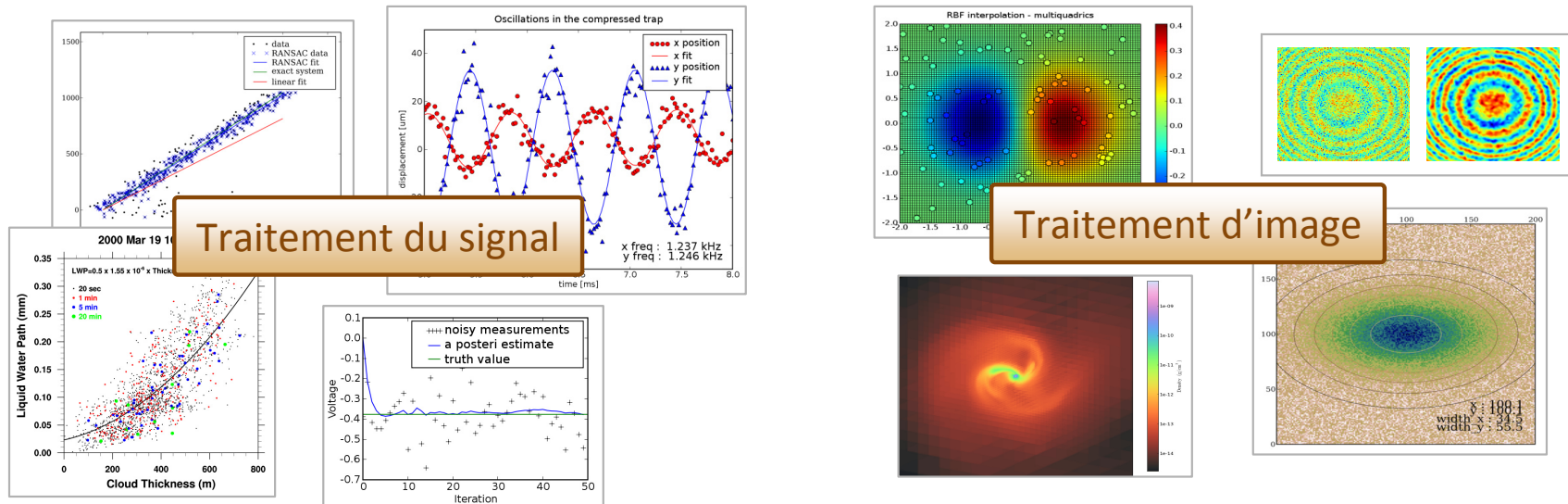


Source : <http://www.python.org>

Et si vous ajoutez à cela...

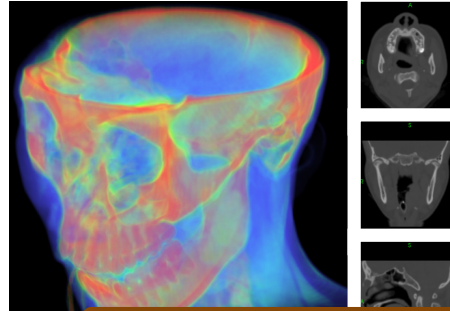
Généralités sur le langage Python

...l'existence de bibliothèques scientifiques de qualité :



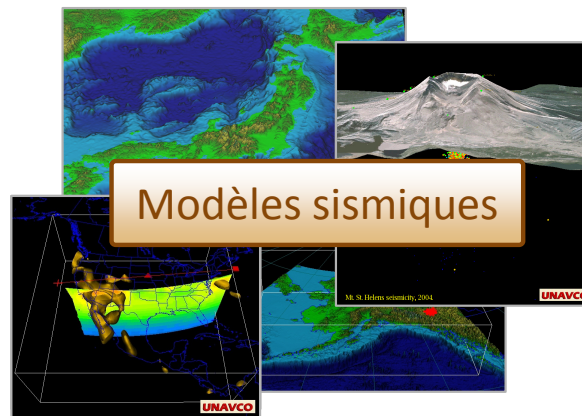
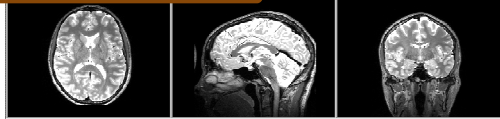
Généralités sur le langage Python

...et une forte communauté d'utilisateurs dans le monde entier :

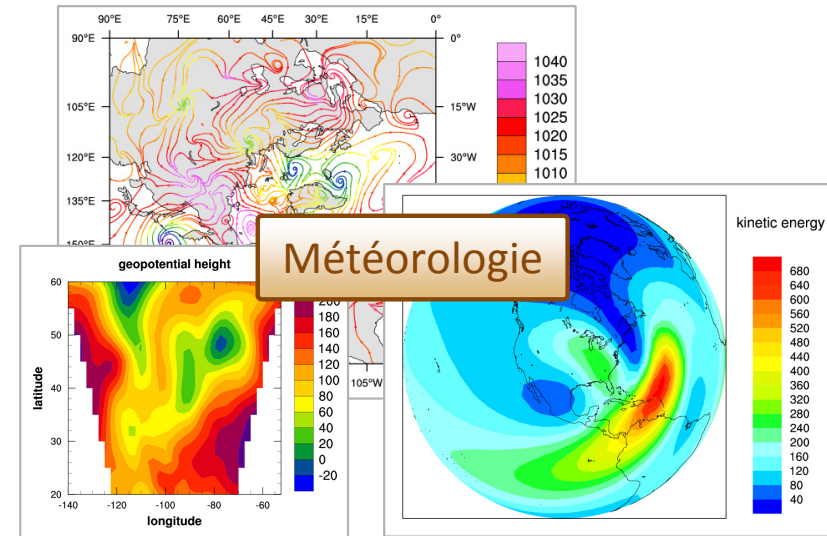


Imagerie médicale
(visualisation 3D et 4D)

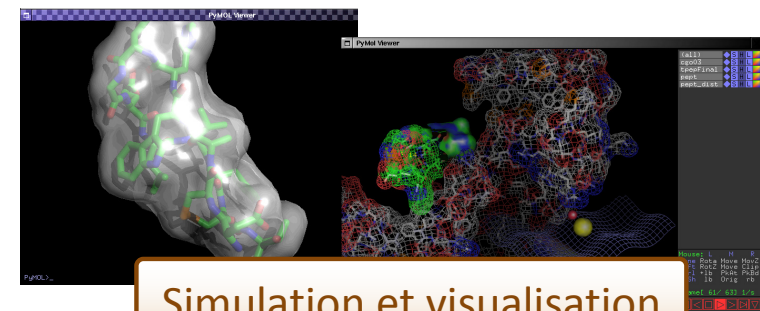
Imagerie médicale



Modèles sismiques



Météorologie



Simulation et visualisation
en biologie moléculaire

Généralités sur le langage Python

...vous obtenez un langage largement utilisé :
(cadre scientifique/technique)



– Recherche :

- CEA (LETI, CESTA, Saclay,...), ONERA, Soleil, INRA
- CERN, DLR, Google, NASA, JPL, ...
- Et nos homologues de *Los Alamos* et *Livermore*



– Industrie :

EDF, THALES Avionics, THALES Air Traffic Management, Dassault, Airbus, Eurocopter, Nokia, STMicroelectronics, Philips, IBM, ...



Plan

☒ Généralités sur le langage **Python**



☐ Python au **CEA/DAM Île-de-France**

☐ Pourquoi migrer de **MATLAB® vers Python ?**

☐ Démonstration des bibliothèques Python internes

☐ Introduction

☐ **guidata** : gestion de jeux de paramètres

☐ **guiqwt** : visualisation de signaux et d'images

☐ Exemple d'une migration réussie depuis **MATLAB®** :

☐ Application : logiciel **Spectrométrie neutron**

☐ Développements / calculs interactifs : environnement **Spyder**

Python au CEA/DAM Île-de-France

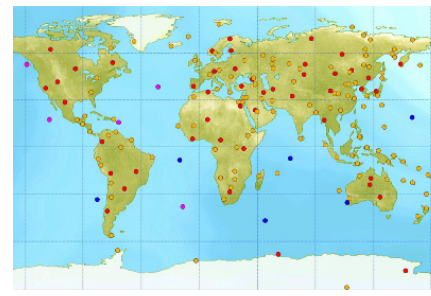
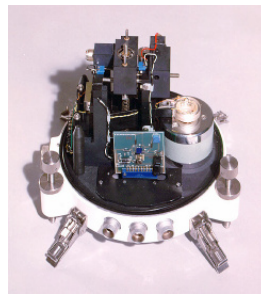
- Les missions du CEA DAM Île-de-France (Bruyères-le-Châtel, 91) :



- **Conception des armes nucléaires**



- **Lutte contre la prolifération et le terrorisme nucléaire**



- **Expertises :**

- Recherche en sciences de la terre
 - Ingénierie et assistance à maîtrise d'ouvrage
 - Génie parasismique
 - Expert de la France dans le cadre de la surveillance des traités (TNP et TICE)

Python au CEA/DAM Île-de-France



- Python est largement utilisé au CEA/DAM Île-de-France de manière significative sur des outils de production :

- Depuis 2004 pour l'interfaçage des codes de calcul intensif
- Depuis 2007 pour l'acquisition, l'analyse, le traitement et la visualisation de données (signaux, images, volumes)

- 3 domaines d'utilisation :

- **Physique expérimentale** (conception de « diagnostics »)
- **Physique théorique** (simulations numériques)
- **Informatique** (calcul intensif)

- Création en 2008 du Club d'Utilisateurs Python :

- ~ 60 membres
- Organisation de séminaires (~ 2/an)
- Création de plusieurs groupes de travail pour coordonner les actions



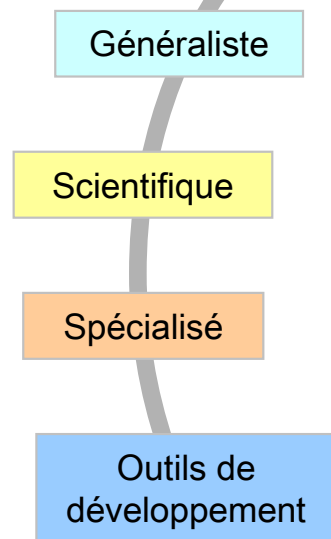
Club d'Utilisateurs Python

CEA/Centre DAM Île-de-France

Python au CEA/DAM Île-de-France



Logiciels relatifs à Python
installés sur les réseaux internes :

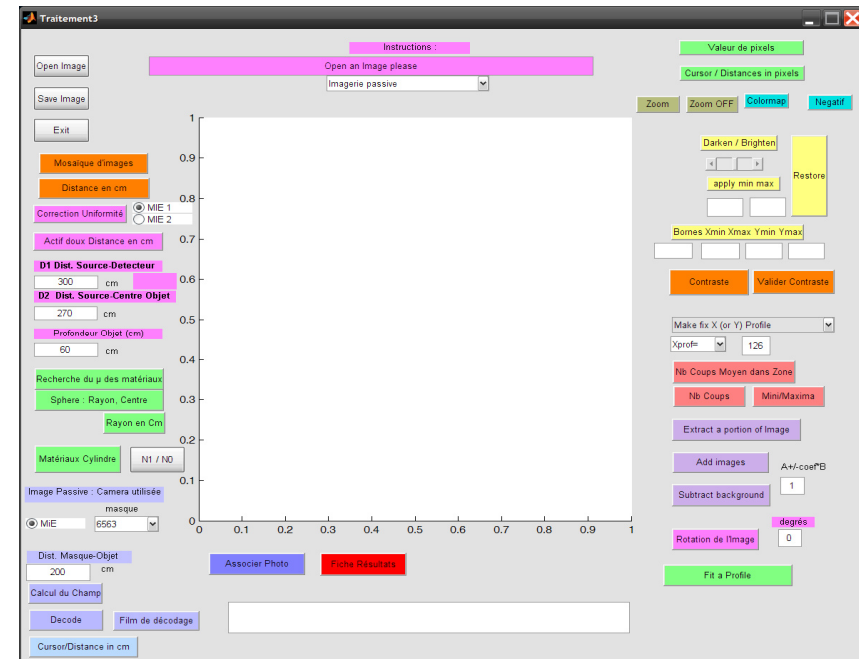
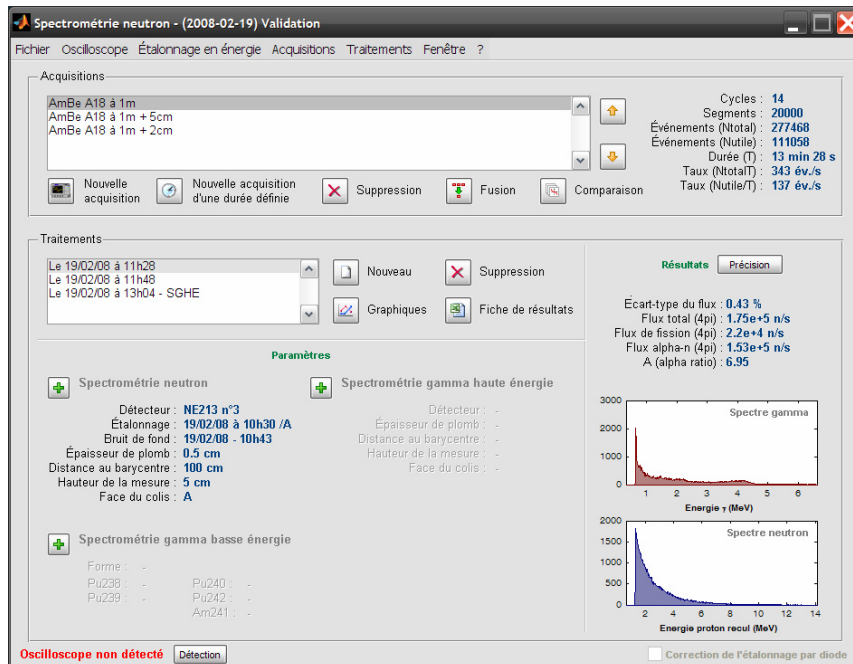


Nom	Version	Description
Python	2.6.2	Bibliothèque standard
PIL	1.1.6	Python Imaging Library (manipulations de base d'images)
PyQt4	4.5.4	Interfaces graphiques Qt
PyTables	2.1.1	Gestion de gros volumes de données (HDF5 v1.8.0)
ViTables	2.0	Explorateur de fichiers HDF5 ou PyTables
h5py	1.2.0	Interfaçage de l'API de la bibliothèque HDF5 v1.8.0
reportlab	2.3	Création de documents PDF
Cython	0.11.2	Développement d'extensions C pour Python
Sphinx	0.6.2	Création de documentations à partir de texte restructuré
rst2pdf	0,11	Conversion de texte restructuré en PDF à l'aide de <i>reportlab</i>
NumPy	1.3.0	Tableaux à N dimensions et algèbre linéaire
nose	0.11.1	Outil de tests utilisé par NumPy
SciPy	0.7.0	Traitement du signal et de l'image
matplotlib	0.99.0	Représentations graphiques 2D pour le calcul interactif
PyQwt	5.2.0	Représentations graphiques 2D pour les applications Qt
VTK	5.4.2	Visualisation 3D
basemap	0.99.4	Projections cartographiques (extension de matplotlib)
pydicom	0.9.3	Manipulation d'images au format DICOM
IPython	0.10.0	Console interactive Python améliorée
Spyder	1.0.0	Environnement de développement interactif
SciTE	1.77	Editeur basé sur Scintilla
Eclipse	3.3.?	Environnement de développement multi-langage
Pydev	1.4.4	Plugin Eclipse pour le développement en Python
CDT	?	Plugin Eclipse pour le développement en C/C++
Photran	4.0.?	Plugin Eclipse pour le développement en Fortran
pylint	0.18.1	Analyse de code (factorisation, bonnes pratiques, erreurs...)
mercurial	1.3.1	Système de contrôle de version



- Exemple de l'**Intervention nucléaire** :
 - Contexte :
 - Lutte contre le terrorisme **NRBC**
 - Dispositif DCI piloté par le RAID : participation du **CEA/DAM**
 - Mission d'une des équipes du CEA intervenant sur le terrain :
diagnostic d'un objet N ou R
 - Moyens opérationnels déployés sur le terrain pour diagnostiquer l'objet :
 - **Matériels** (détecteurs et instruments d'acquisition)
 - **Logiciels** :
 - commerciaux/libres (besoins généralistes)
 - internes (besoins spécifiques)

- Exemple de l'Intervention nucléaire :
 - Logiciels développés en interne :
 - traitement du signal et de l'image
 - pilotage d'instruments et acquisition
 - analyse de données nucléaires



Plan



☒ Généralités sur le langage **Python**

☒ Python au **CEA/DAM** Île-de-France

☐ Pourquoi migrer de **MATLAB®** vers **Python** ?

☐ Démonstration des bibliothèques Python internes

☐ Introduction

☐ **guidata** : gestion de jeux de paramètres

☐ **guiqwt** : visualisation de signaux et d'images

☐ Exemple d'une migration réussie depuis **MATLAB®** :

☐ Application : logiciel **Spectrométrie neutron**

☐ Développements / calculs interactifs : environnement **Spyder**

Pourquoi migrer de **MATLAB®** vers **Python** ?



- Pourquoi migrer de MATLAB® vers Python ?
 - Utilisation de bibliothèques généralistes* de qualité
 - Gain de performance : Python s'interface facilement avec **C/C++/Fortran**
 - Maîtrise du code source : passer à Python, c'est entrer dans le monde du **logiciel libre**
 - Amélioration de la qualité du code source :
 - *Pérennité* : un script Python **se maintient plus facilement**
 - *Généricité* : le code source Python **se réutilise et s'adapte facilement** (fondamentalement **orienté objet**)
 - Coût : Python est **gratuit**

...avec les **mêmes fonctionnalités scientifiques**** que MATLAB®

(*) Exemples : interfaces graphiques (Qt, GTK, ...), réseau, structures de données (XML, HDF5, ...), bases de données, calcul parallèle, etc.

(**) Python + NumPy + SciPy $\simeq$ MATLAB® + Signal proc. toolbox + Image proc. toolbox

Plan



☒ Généralités sur le langage Python

☒ Python au CEA/DAM Île-de-France

☒ Pourquoi migrer de MATLAB® vers Python ?

☐ Démonstration des bibliothèques Python internes

- ☐ Introduction

- ☐ **guidata** : gestion de jeux de paramètres

- ☐ **guiqwt** : visualisation de signaux et d'images

☐ Exemple d'une migration réussie depuis MATLAB® :

- ☐ Application : logiciel **Spectrométrie neutron**

- ☐ Développements / calculs interactifs : environnement **Spyder**

Démonstration des bibliothèques Python internes

- Quels sont les besoins fonctionnels d'un logiciel de traitement en termes d'interfaces graphiques ?



- **Jeux de paramètres :**

- Paramètres de type variable : nombres réels (grandeurs physiques), entiers (indices de tableaux), chaîne de caractères (noms de fichier), booléen (activation d'une option), etc.
- Manipulations courantes :
 - Saisie de chaque paramètre via une interface graphique en adaptant chaque *widget* au type du paramètre concerné
 - Stockage des valeurs saisie : convention de stockage
 - Utilisation des valeurs pour des calculs (par exemple) sans perdre l'information d'appartenance au jeu de paramètres
 - Affichage de ces valeurs dans une interface graphique, en même temps que les résultats de calculs (par exemple), en adaptant l'affichage au type du paramètre concerné



Démonstration des bibliothèques Python internes

- Quels sont les besoins fonctionnels d'un logiciel de traitement en termes d'interfaces graphiques ?



- **Visualisation de données 2D :**

- **Type de données :**
 - Représentation graphique de courbes
 - Affichage d'images
- **Fonctionnalités / qualités requises :**
 - Interactivité
 - Performances
 - Modification des propriétés des objets représentés via une interface graphique



Démonstration des bibliothèques Python internes

- Quels sont les besoins fonctionnels d'un logiciel de traitement en termes d'interfaces graphiques ?



- **Jeux de paramètres**



- **Visualisation de données 2D**



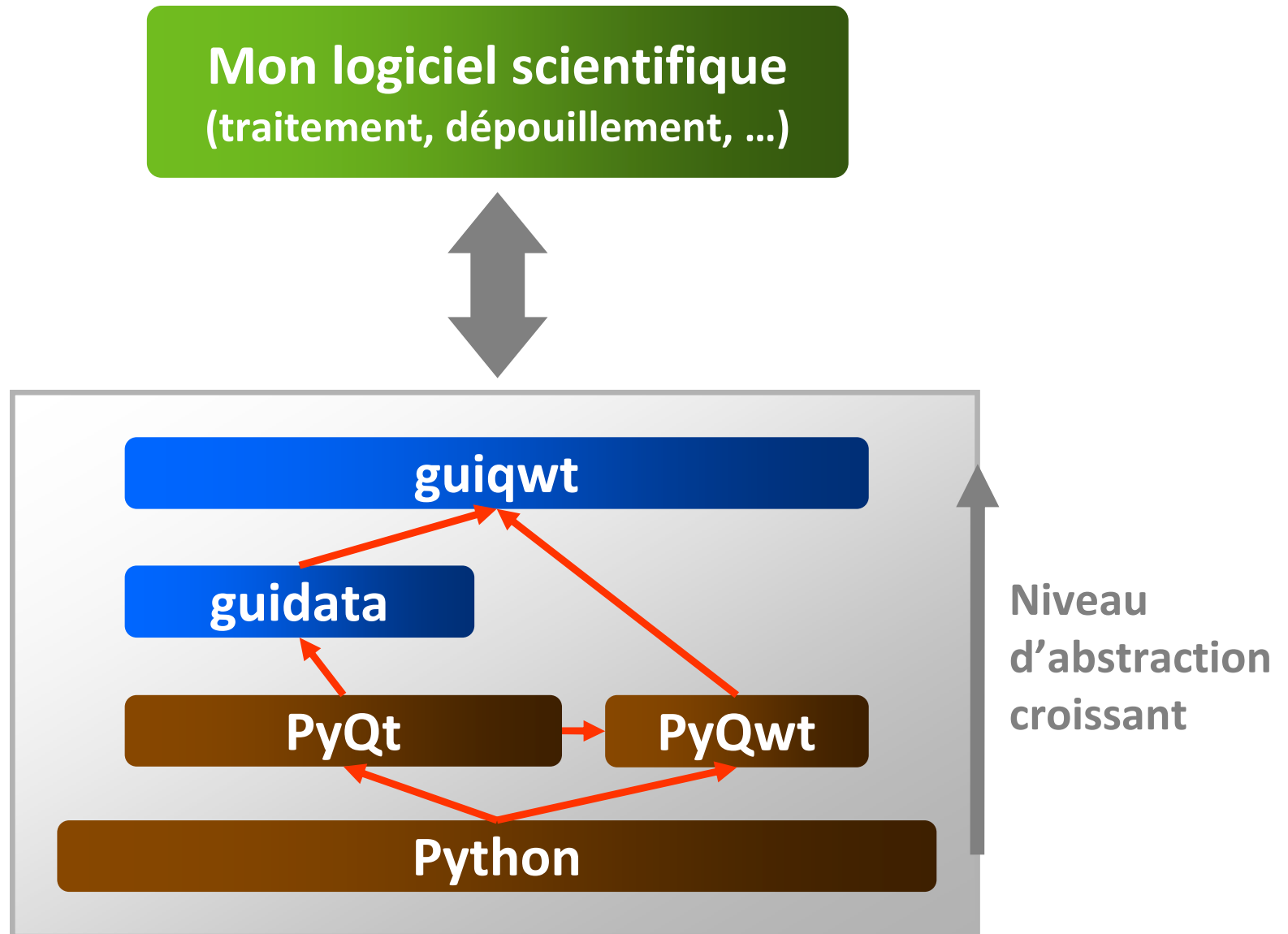
Démonstration des bibliothèques Python internes

- Dépendances de **guidata** et **guiqwt** :



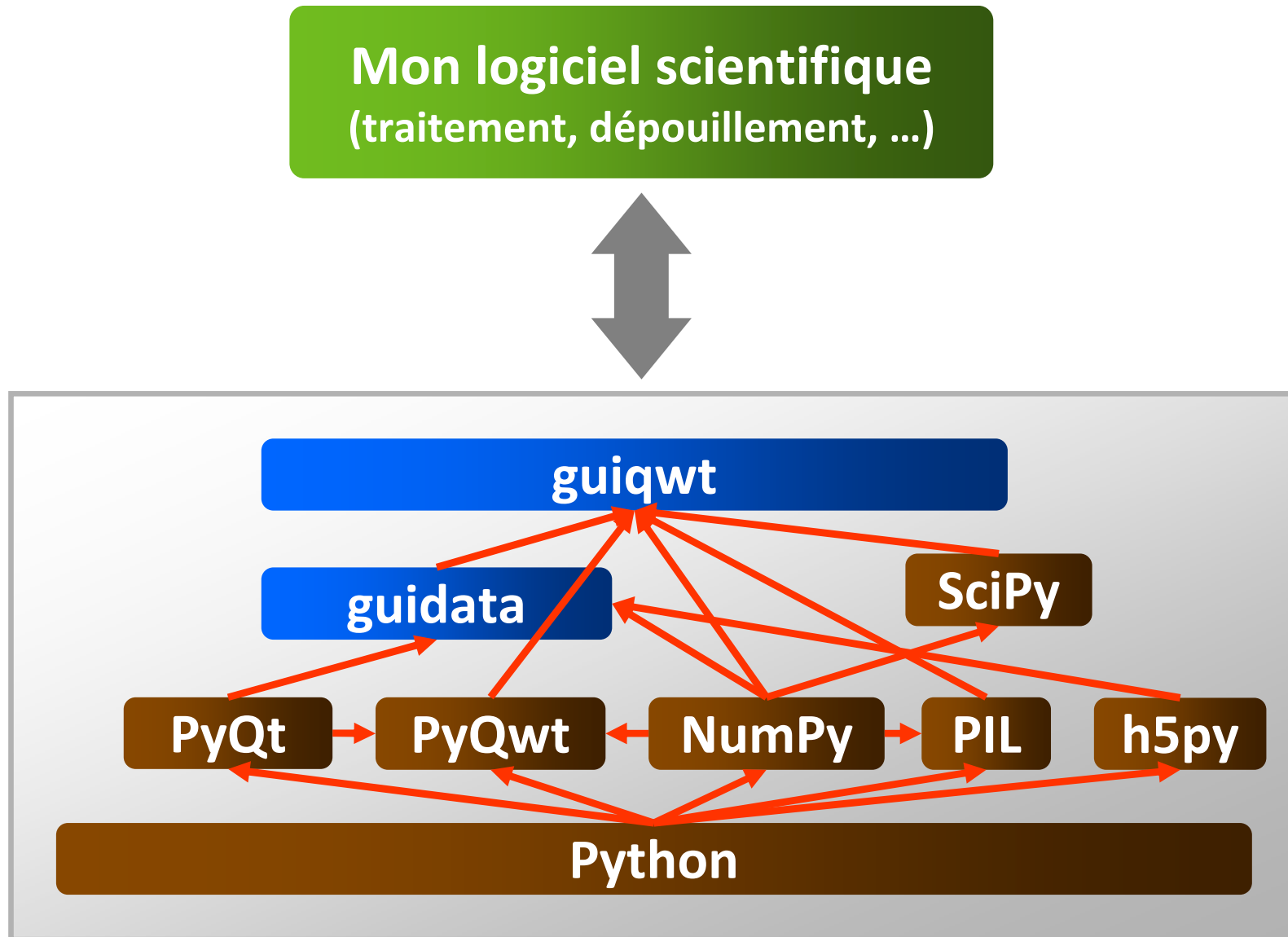
 Bibliothèques internes

 Bibliothèques libres



Démonstration des bibliothèques Python internes

- Dépendances de **guidata** et **guiqwt** :



Démonstration de Spyder

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script with a class `ArrayModel` and its `__init__` method. The `__init__` method initializes various attributes like `self.dialog`, `self.changes`, `self.readonly`, `self.test_array`, and `self._data`. The `ArrayModel` class inherits from `QAbstractTableModel`.

The `Console externe` window shows the output of the script, displaying a 5x5 array of floating-point numbers and the results of a comparison operation:

```
Running... 00:01:55 [x] Interagir [ ] Débuguer [x] Variables [Run] [Quitter] [Kill]
[ 0.94876509 0.8324598 0.24973361 0.7056157 0.87080709]
[ 0.38232321 0.13274058 0.54044065 0.54490118 0.75551441]
[ 0.81094951 0.37599384 0.46039142 0.02001402 0.71441868]
[ 0.50077434 0.6914647 0.57049383 0.58545685 0.33891245]
in: [ True False True]
out: [ True False True]
True
out: [ 1 2 3]
```

The `Explorateur de variables globales` window shows the following variables:

Nom	Type	Taille	Valeur
arr	int8	(3,)	Min: 1 Max: 3
arr_in	bool	(3,)	Min: False Max: True
arr_out	bool	(3,)	Min: False Max: True

The `Console interactive` window shows the current state of the variables:

```
Console interactive
Droits d'accès: RW Encodage: UTF-8 Ligne: 69 Colonne: 35
```


Plan



- ☒ Généralités sur le langage **Python**

- ☒ Python au **CEA/DAM Île-de-France**

- ☒ Pourquoi migrer de **MATLAB® vers Python ?**

- ☒ Démonstration des bibliothèques Python internes

 - ☒ Introduction

 - ☒ **guidata** : gestion de jeux de paramètres

 - ☒ **guiqwt** : visualisation de signaux et d'images

- ☐ Exemple d'une migration réussie depuis **MATLAB®** :

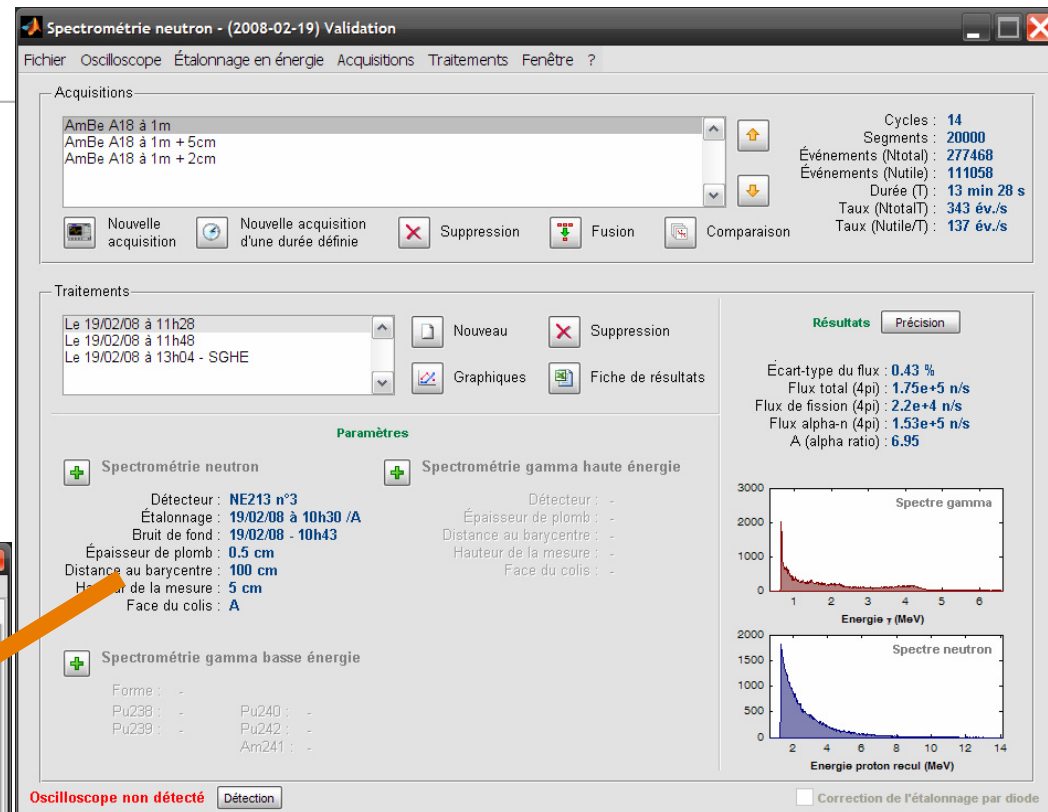
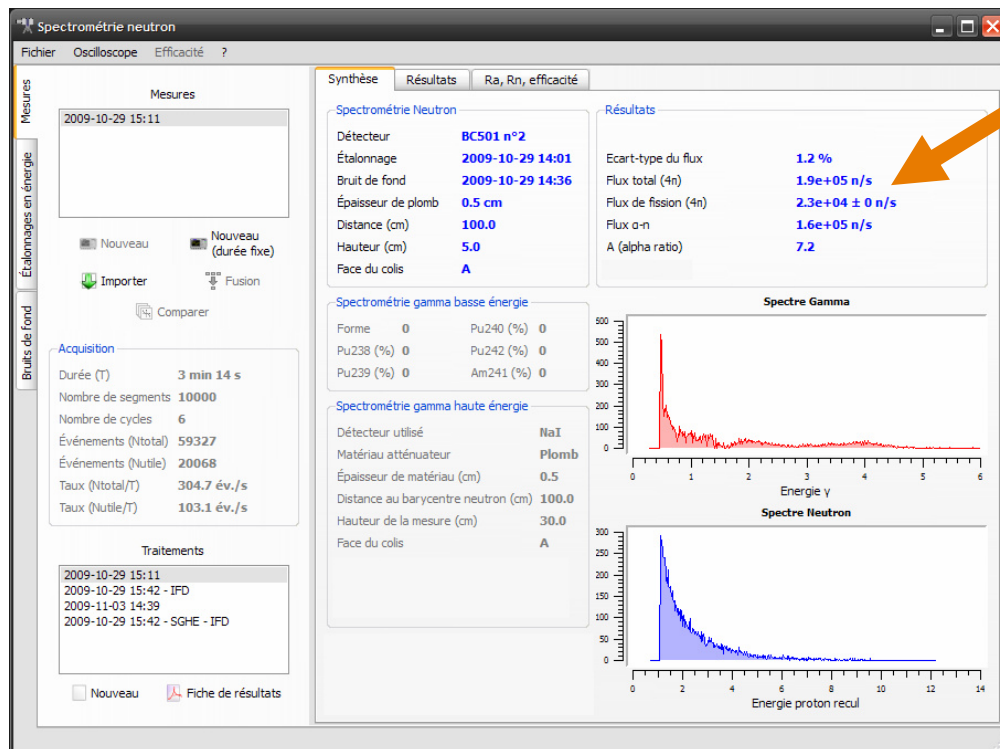
 - ☐ Application : logiciel **Spectrométrie neutron**

 - ☒ Développements / calculs interactifs : environnement **Spyder**

Démonstration du logiciel Spectrométrie neutron



Version 2.0 : **MATLAB**[®]



Version 3.0 : **Python**

Plan



✓ Généralités sur le langage Python

✓ Python au CEA/DAM Île-de-France

✓ Pourquoi migrer de MATLAB® vers Python ?

✓ Démonstration

✓ Introduction

✓ **guidata** : gestion de jeux de paramètres

✓ **guiqwt** : visualisation de signaux et d'images

✓ Exemple d'une migration réussie depuis MATLAB® :

✓ Application : logiciel **Spectrométrie neutron**

✓ Développements / calculs interactifs : environnement **Spyder**

Questions