Retour d'expérience

python™

Loïc Chevallier

Université Lille 1 (L2EP)

Université Lille1 Sciences et Technologies

L2EP LILLE

MEDEE

Mon parcours professionnel (en lien avec Python)

- **Développeur calcul scientifique (3 ans au L2EP, financ. MEDEE),**
  - **Code_Carmel3D (LAMEL), Salomé / Eficas (EDF & CEA), Spyder *MATLAB*,**
- **Développeur site web (1 an, Rectorat de Lille / CRI),**
  - **Plone (version 3, Zope 2),**
- **Thèse (1996-2000) + postdoc @stro (11 ans)**
  Observatoires de Lyon, Paris ; laboratoires de Varsovie et Lexington
  - **Analytical Radiative Transfer (B. Rutily) code Fortran 77 (ARTY)**
  - **Interface Python (F2PY), etc.**

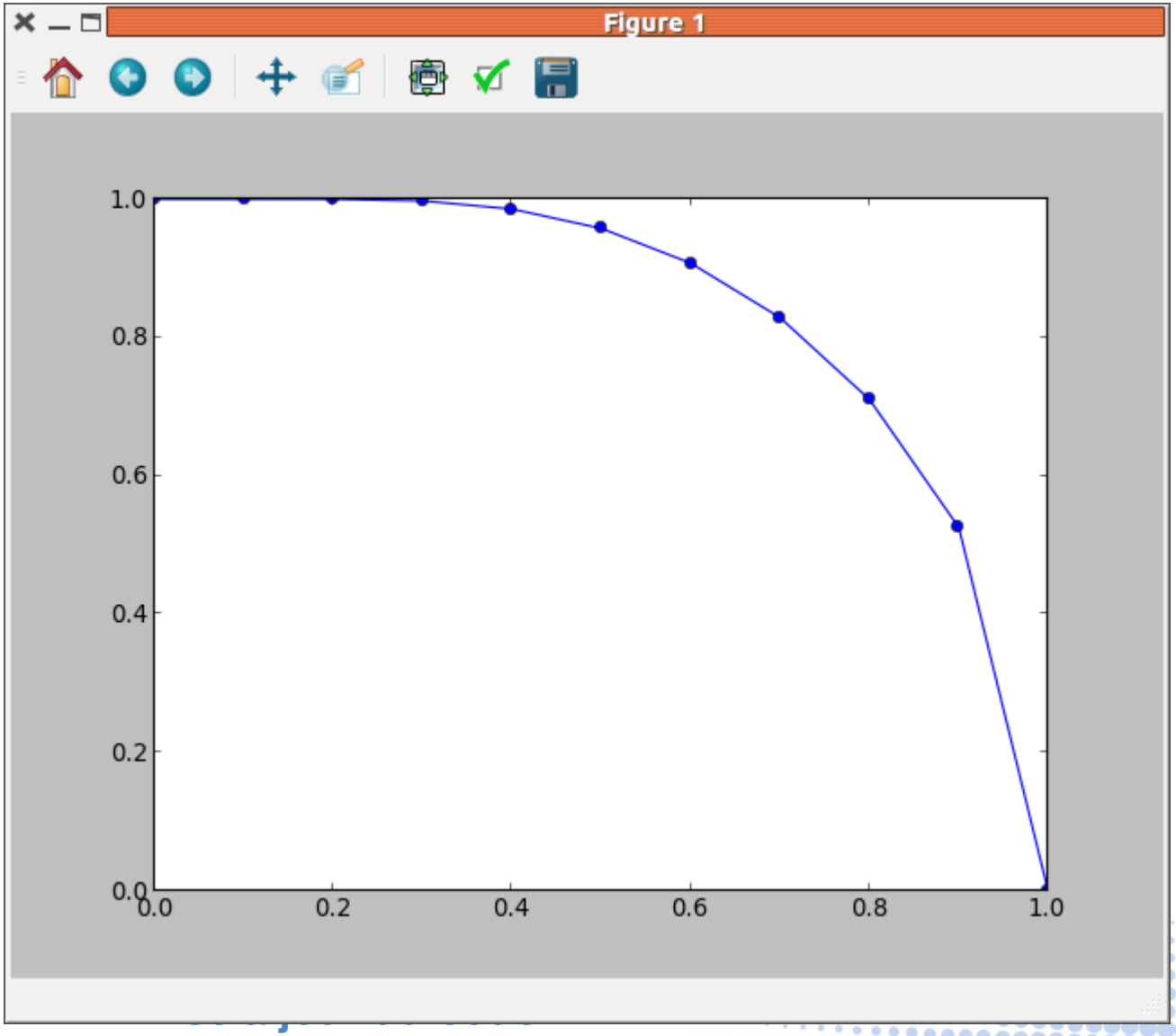
- **Utilisation Python ~ 12 ans,**
  - **Calcul scientifique, visualisation de données, programme terminal, interface graphique, analyse/transformation de texte, base de donnée, application web, IDE, documentation de code (API).**
- **Formation « Python par la pratique » (2005-2007, 3x, cours + 12 TPs).**

Mes débuts... avec ARTY

- **code de thèse (Fortran77, librairie NAG, modulaire, ~ 100 000 lignes) :**
  - **de nombreuses fonctions (~ 100, 50 000 lignes),**
  - **interface utilisateur terminal : choix par liste, visualisation courbes (50 kL),**
  - **2/3 temps déboguage interface utilisateur,**
- **Programmation agile : interface utilisateur en Python + fonctions en Fortran utilisables en Python (F2PY), très simplement :**
- **F2PY :**
  - **Création automatique d'interface pour tous les fichiers du code (arty.pyf),**
  - **Modification manuelle de l'interface (F77, intent) ou ajout directives dans code (Cf2py intent(in)),**
  - **Création du module Python (arty.so),**
  - **Automatisation dans Makefile (make) lors de la mise à jour du code.**

```
$ python
>>> from arty import *
>>> k(0.5)
0.9575040240772688
>>> a_liste = arange(0, 1, 0.1)
>>> a_liste
array([ 0. , 0.1, 0.2, 0.3, 0.4, 0.5,
0.6, 0.7, 0.8, 0.9])
>>> a_liste = arange(0, 1.1, 0.1)
>>> k_liste = [k(a) for a in a_liste]
>>> k_liste
[1, 0.99999999587769239,
0.9999091217152325,
0.9974138168919747,
0.9856238716346568,
0.9575040240772688,
0.9073323166453099,
0.8286347986408346,
0.7104117834878703,
0.5254295126580087, 0]
```

Mes débuts... avec ARTY (suite)

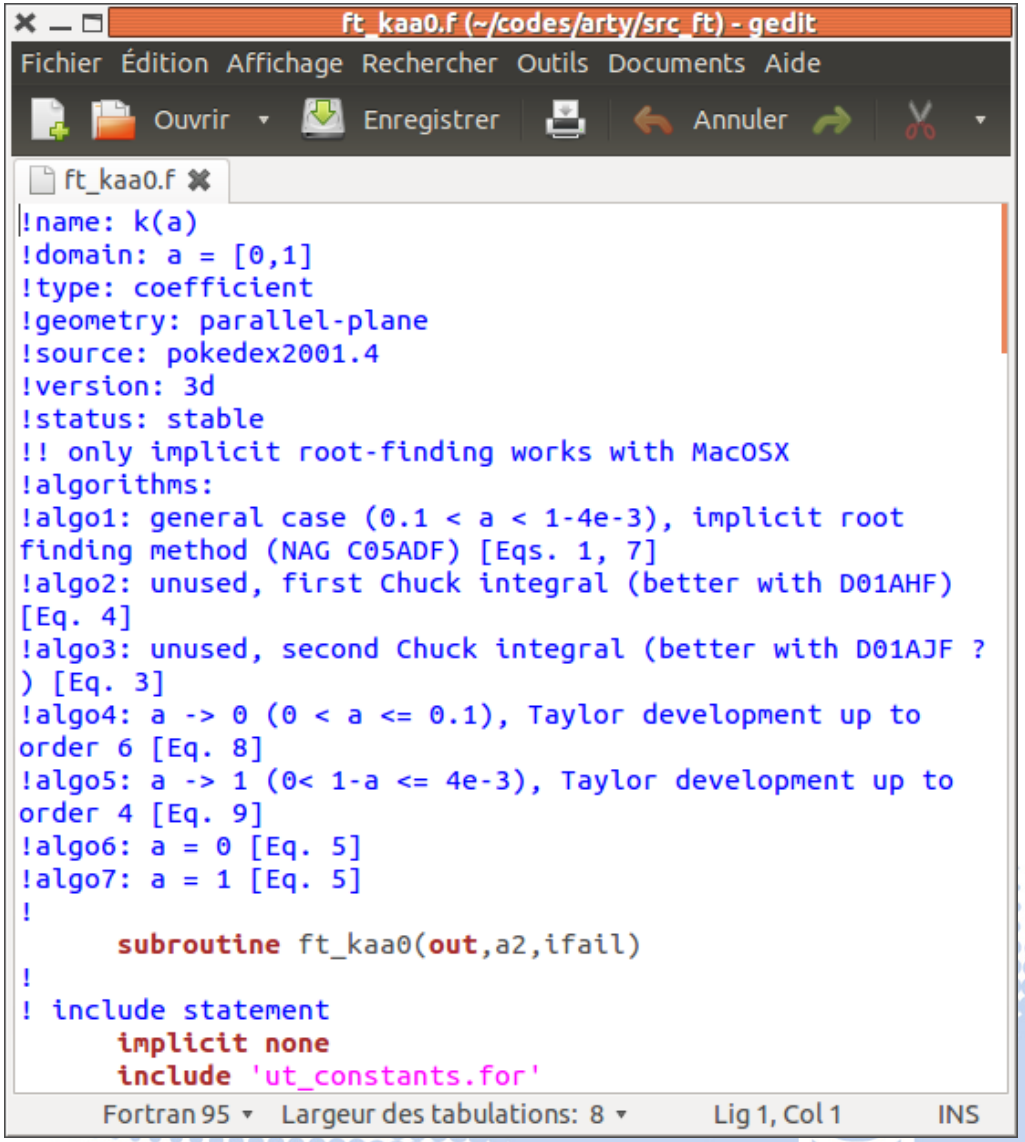



```
$ python
>>> from arty import *
>>> k(0.5)
0.9575040240772688
>>> a_liste = arange(0, 1, 0.1)
>>> a_liste
array([ 0. , 0.1, 0.2, 0.3, 0.4, 0.5,
0.6, 0.7, 0.8, 0.9])
>>> a_liste = arange(0, 1.1, 0.1)
>>> k_liste = [k(a) for a in a_liste]
>>> k_liste
[1, 0.99999999587769239,
0.9999091217152325,
0.9974138168919747,
0.9856238716346568,
0.9575040240772688,
0.9073323166453099,
0.8286347986408346,
0.7104117834878703,
0.5254295126580087, 0]
>>> plot(a_liste, k_liste)
```

Mes débuts... avec ARTY (suite)

- **Documentation automatique ?**
  - Traitement de texte,
  - Essai shell (sed) insuffisant,
  - Python : 957 lignes de code,
  - Doc HTML et LaTeX (PDF).

NAME: k(a)
TYPE: coefficient
DOMAIN: a = [0,1]
GEOMETRY: parallel-plane
SOURCE: pokedex2001.4
ROUTINE NAME: ft_kaa0
ARGUMENTS: out, a2, ifail
ROUTINE TYPE: subroutine
VERSION: 3d
STATUS: stable
FILE: ft_kaa0.f
DIRECTORY: src_ft
LAST MODIFICATION DATE: 2008-11-18 16:35:55
INCLUDED FILES: ut_constants.for
USING: ut_quad1, ut_end1, st_sqrt, f1, st_exp, ft_kea
ut_findroot, f36a, ut_settings
USED BY: ft_f53, ft_pdc3, ft_rhodpmk3a, ft_kda0, tt_
tt_seddington_f3, feta1c, feta1b, feta1a, ft_kr, feta1d,
ft_aba3, ft_pbc3, fetaykp3a, compute1d_a.f, ft_bk3, ft



Refonte ARTY en Python pur : pyARTY

- **Fonctionnalités autres calcul des fonctions:**
  - **validation des entrées, sauvegarde, interpolation, contrôle des erreurs d'arrondi et de fonction, modification paramètres de calcul.**
  - **5ème réécriture de l'interface**

```
19 def H(a, u, methode=auto):
20     r"""Fonction auxiliaire H(a,u).
21     """
22     if methode == 'Neumann ordre 1 (a -> 0)':
23         return 1+a*u/2*ln(1+1/u)
24
25     if methode == 'a = 0 ou u = 0':
26         return 1
```

```
subroutine ft_hab2(out,a2,u,ifail)
...
in(1) = a2(1)
in(2) = a2(2)
in(3) = u
! get settings and check input
call ut_start(ridx,out,in,3,par,ifail)
if (ifail.ne.0) return
...
! Neumann development
np=np+1
par(np) = 1.d0
np=np+1
ns=ns+1
par(np) = a * u / 2.d0 * st_log(1.d0+1.d0/u,status(ns))
...
! put settings
call ut_end1(ridx,out,in,3,par,np,choice,status,ns,ifail)
return
end
```

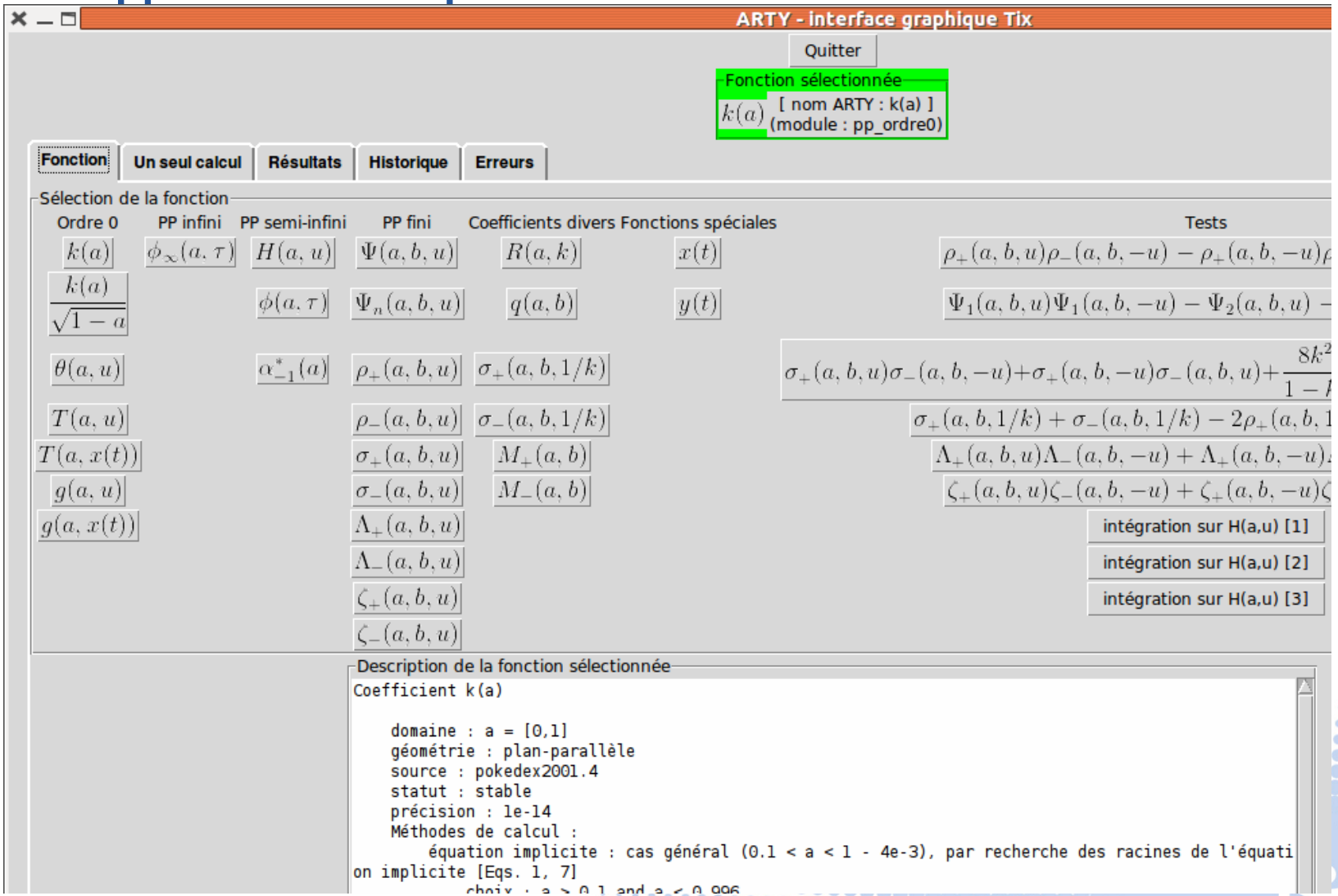
Refonte ARTY en Python pur : pyARTY (suite)

- **Décorateurs @, NAG → GSL :**
  - **Temps de calcul x10**
  - **Contrôle erreur d'arrondi ?**

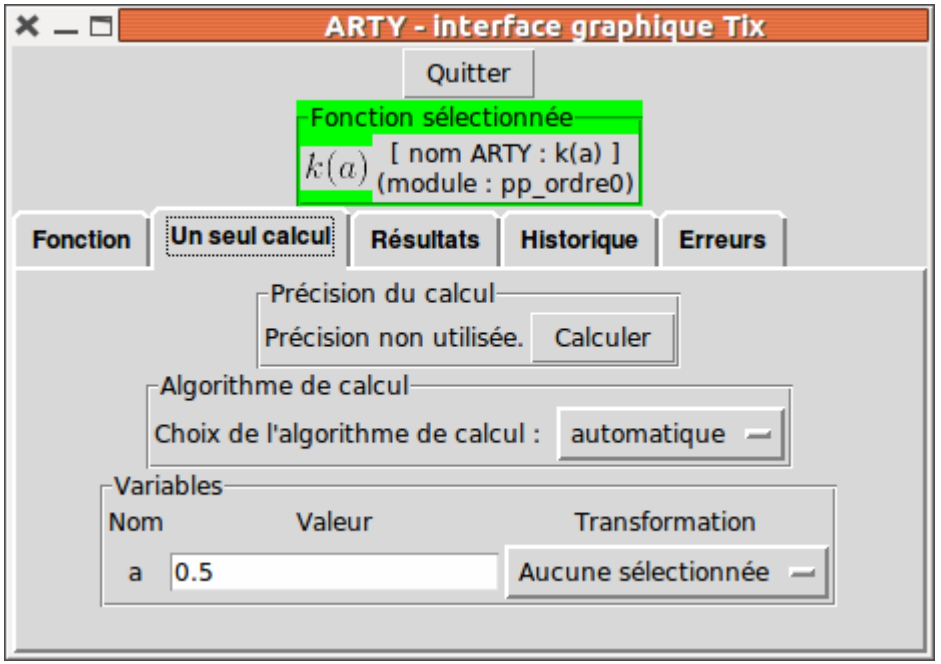
```
@extension
def H(a_, u_, methode=auto,  **options):
    r"""Fonction auxiliaire H(a,u).

    Quelques exemples de résultats

    >>> H(0.5,1)
    1.2512595633832226

    domaine : a = [0,1]; u = R+
    géométrie : plan-parallèle
    source : pokedex2001.16
    statut : développement
    précision : 1e-10
    Méthodes de calcul :
        Neumann ordre 1 (a -> 0) :  développement de Neumann d'
            choix : a > 0 and a <= 1e-7
        a = 0 ou u = 0 :   valeur analytique H(a,0)=1 et H(0,u)=
            choix : a == 0 or u == 0
      Interpolation inactive par défaut :
        paramètres pour a <> 0.5 and u >= 0 and u <= 1 and preci
    """
    a, unMa, u, unMu = valeursReelles(a_, 'x,1-x', u_, 'x,1-x')

    if methode == 'Neumann ordre 1 (a -> 0)':
        return 1+a*u/2*ln(1+1/u)

    if methode == 'a = 0 ou u = 0':
        return 1
```

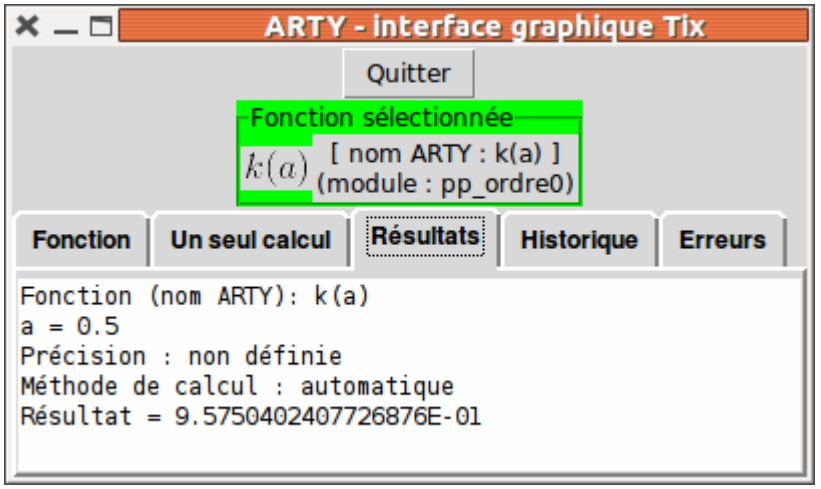
PyARTY : interface graphique Tk/Tix

- **Développement de l'aspect en texte**

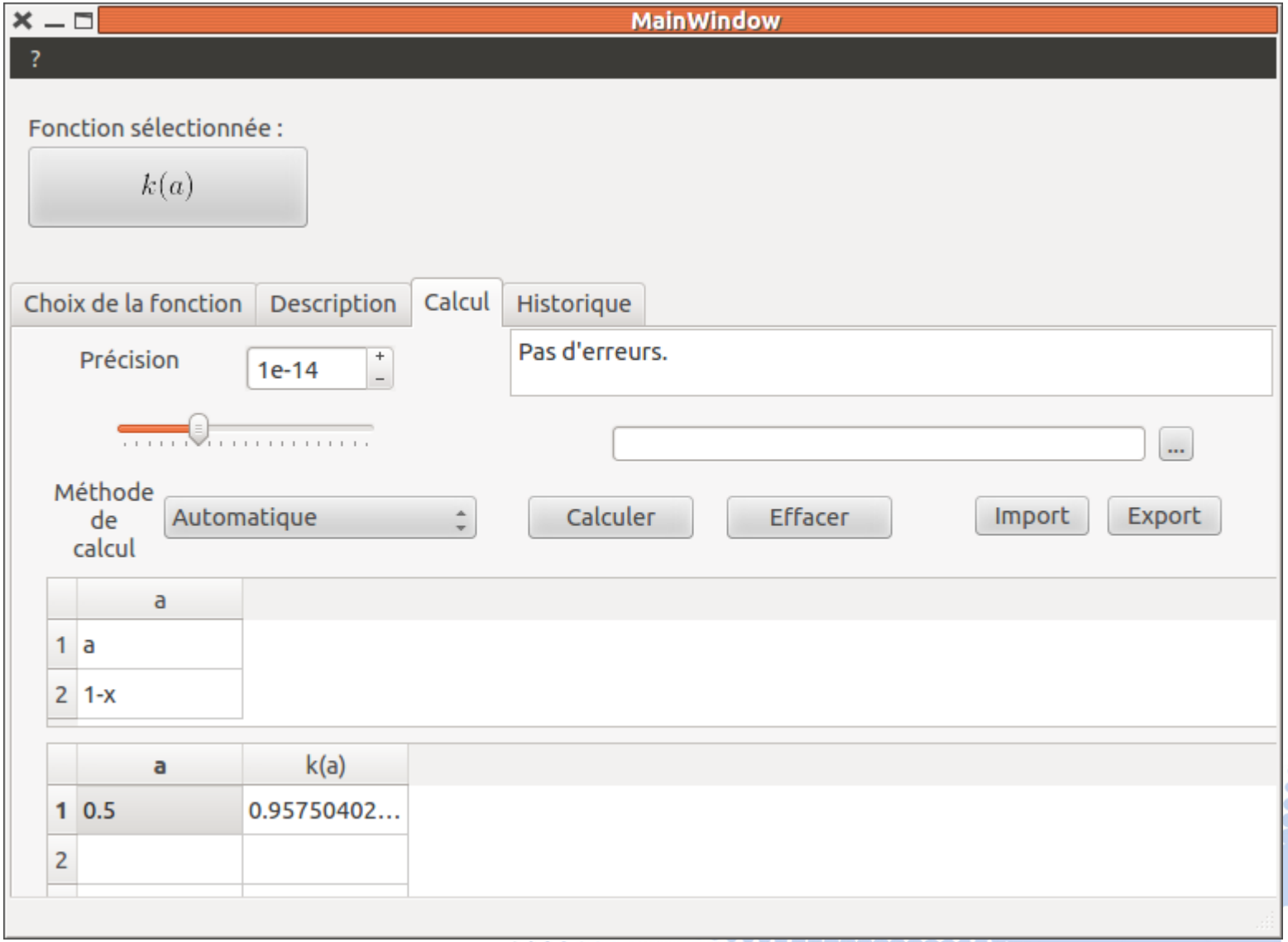


PyARTY : interface graphique Tk/Tix

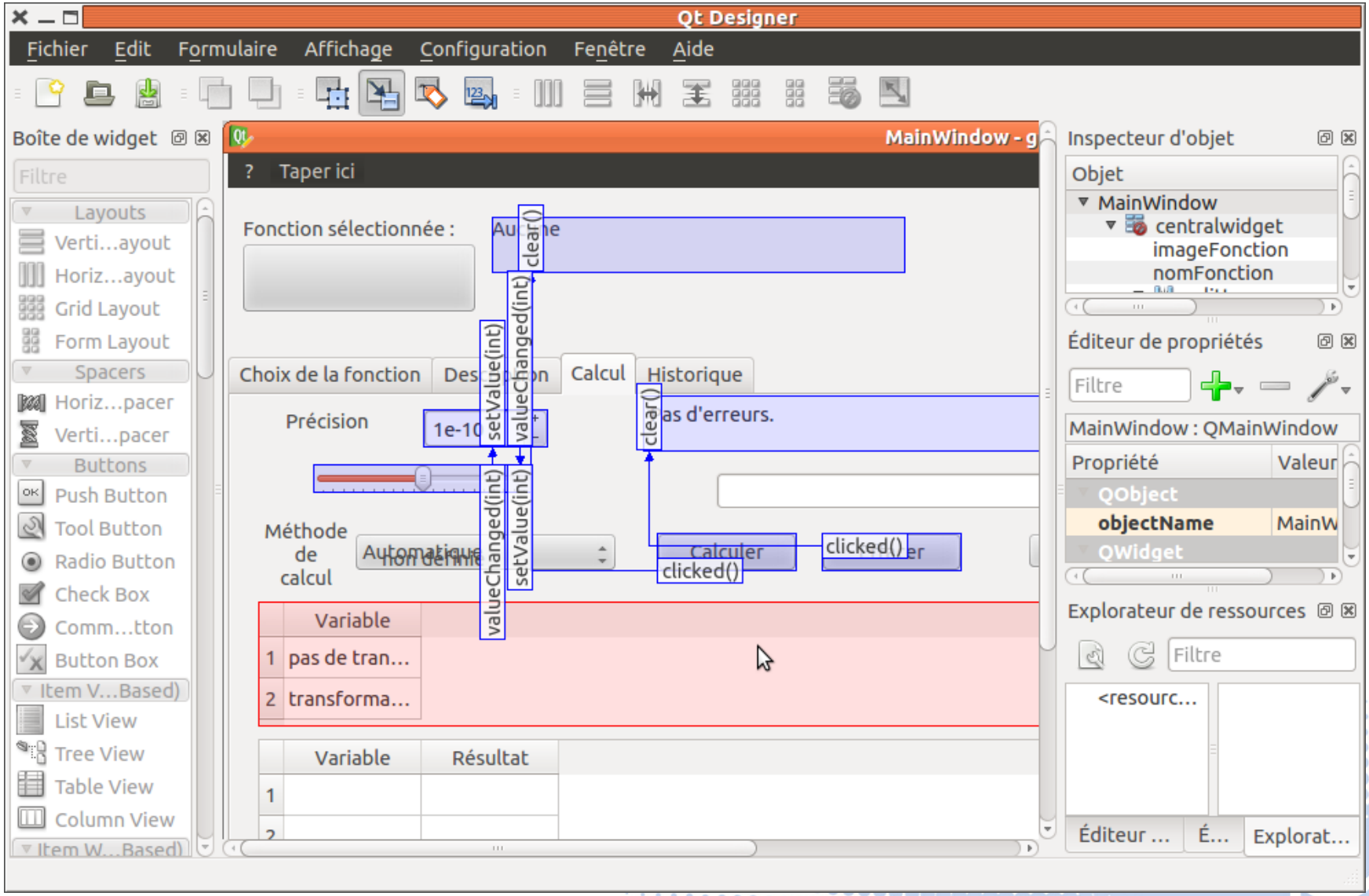




PyARTY : interface graphique Qt4...



PyARTY : interface graphique Qt4 + Qt Designer



Programme Python – module optparse

loic@l2eppc-13a: ~/codes/python/loicmodule

Fichier Édition Affichage Terminal Aide

```
loic@l2eppc-13a:~/codes/python/loicmodule$ ./noar_lancement.py -h
Usage:
        This module will help the user of NOAR code to prepare a run and make some tr
ansformations on output files.
        It is based on shell script 'lancement' version 1 (included in this module).
        This requires Python2.3+ and Numeric external module (not tested).


usage: noar_lancement.py [options]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -n STRING, --name=STRING
                        session name used for output files <session.*>
                        [default: Ptot_x4265_100it]
  -a STRING, --action=STRING
                        action: all,prepare,run,transform to execute all or
                        one the three steps [default: all]
  -s, --shell           use original shell script instead of Python. Less
                        secure.
  -d, --doit            inverse flag to do nothing, just show messages if any.
  -c, --checkscript     shortcut to show the shell script of the chosen
                        action, equivalent to -sdv.
  -q, --quiet           don't print status messages to stdout
  -v, --verbose         print all status messages to stdout
loic@l2eppc-13a:~/codes/python/loicmodule$ ./noar_lancement.py --version
0.13 2005/01/13
loic@l2eppc-13a:~/codes/python/loicmodule$
```

Programme Python

loic@l2e

Fichier Édition Affichage Terminal Aide

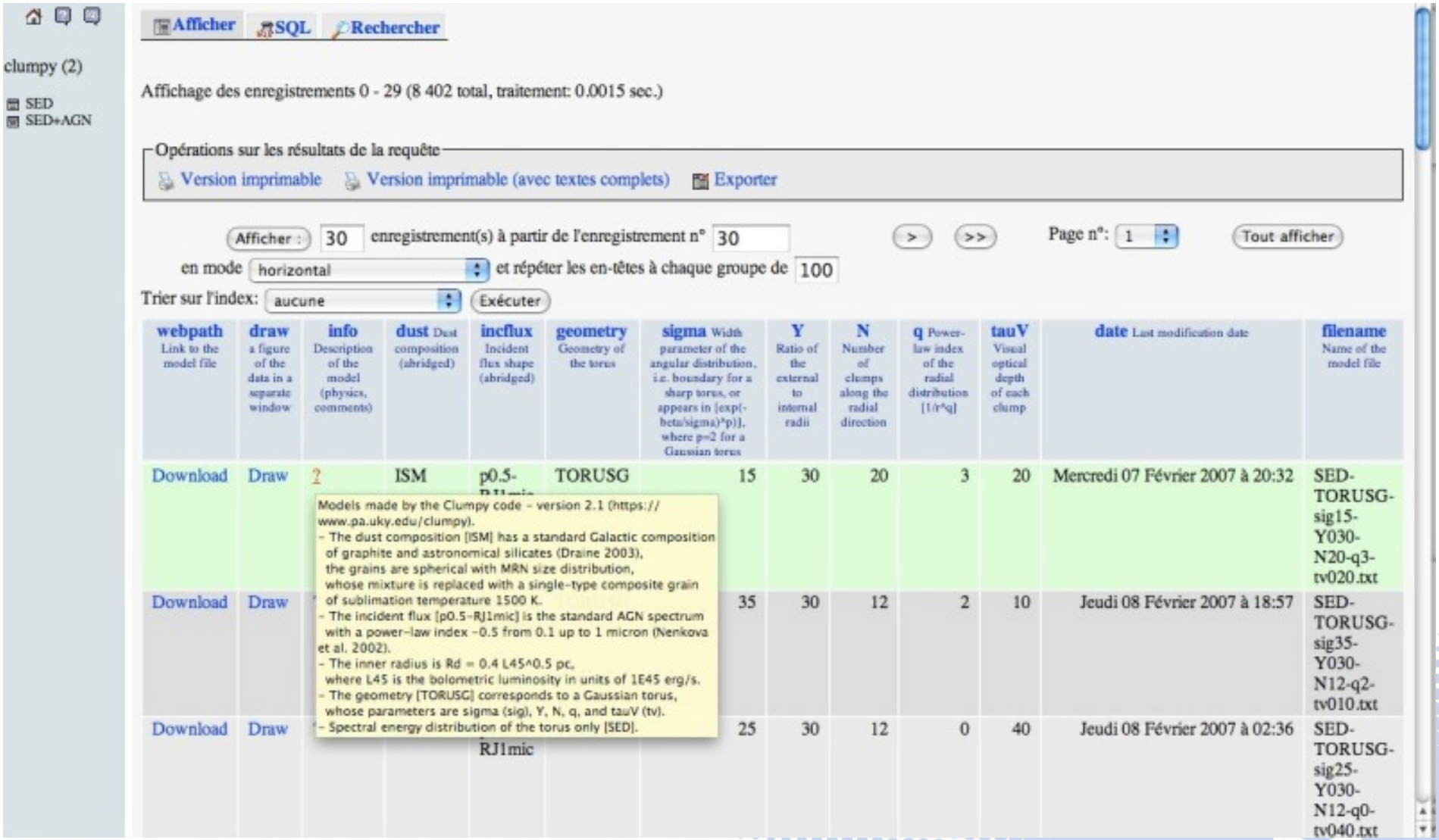
```
loic@l2eppc-13a:~/codes/python/loicm
Usage:
        This module will help the us
ansformations on output files.
        It is based on shell script
        This requires Python2.3+ and


usage: noar_lancement.py [options]

Options:
  --version             show program
  -h, --help            show this he
  -n STRING, --name=STRING
                        session name
                        [default: Pt
  -a STRING, --action=STRING
                        action: all,
                        one the thre
  -s, --shell           use original
                        secure.
  -d, --doit            inverse flag
  -c, --checkscript     shortcut to
                        action, equi
  -q, --quiet           don't print
  -v, --verbose         print all st
loic@l2eppc-13a:~/codes/python/loicm
0.13 2005/01/13
loic@l2eppc-13a:~/codes/python/loicm
```

loic@l2eppc-13a: ~/codes/python/loicmodule

Fichier Édition Affichage Terminal Aide

```
def main():
        import optparse as opt
        usage = __doc__+'\n\n'+"usage: %prog [options]"
        parser = opt.OptionParser(usage=usage,version=__version__)
        parser.add_option("-n", "--name", dest="session", type="string", default=default_session, metavar="STRING"
                  ,help="session name used for output files <session.*> [default: "+default_session+"]")
        parser.add_option("-a", "--action", dest="action", type="string", default="all", metavar="STRING"
                  ,help="action: all,prepare,run,transform to execute all or one the three steps [default: all]")
        parser.add_option("-s", "--shell",
                  action="store_true", dest="shell", default=False
                  ,help="use original shell script instead of Python. Less secure.")
        parser.add_option("-d", "--doit",
                  action="store_false", dest="doit", default=True
                  ,help="inverse flag to do nothing, just show messages if any.")
        parser.add_option("-c", "--checkscript",
                  action="store_true", dest="checkscript", default=False
                  ,help="shortcut to show the shell script of the chosen action, equivalent to -sdv.")
        parser.add_option("-q", "--quiet",
                  action="store_false", dest="verbose"
                  ,help="don't print status messages to stdout")
        parser.add_option("-v", "--verbose",
                  action="store_true", dest="verbose"
                  ,help="print all status messages to stdout")
        (options, args) = parser.parse_args()
        verbose = options.verbose
        session = options.session
        action = options.action
        shell = options.shell
        doit = options.doit
        checkscript = options.checkscript
        if action not in ['all','prepare','run','transform']:
                parser.error("option -a : incorrect value. See help.")
        if checkscript: # just show the shell script, equivalent to -sdv
                verbose = True
                shell = True
                doit = False
                print "SHELL SCRIPT :: BEGIN ::"
        if action in ["prepare","all"]:
                if verbose and not checkscript:
                        print "action : prepare..."
                prepare(session,shell=shell,verbose=verbose,doit=doit)
        if action in ["run","all"]:
                if verbose and not checkscript:
                        print "action : run..."
                run(session,shell=shell,verbose=verbose,doit=doit)
        if action in ["transform","all"]:
                if verbose and not checkscript:
                        print "action : transform..."
                transform(session,shell=shell,verbose=verbose,doit=doit)
        if checkscript: # just show the shell script, equivalent to -sdv
                print "SHELL SCRIPT :: END ::"



if __name__ == "__main__":
        main()
                                                                286,0-1       Bas
```

CLUMPY – phpMyAdmin+MySQL+ explorateur Python



clumpy (2)

- SED
- SED+AGN

Afficher | SQL | Rechercher

Affichage des enregistrements 0 - 29 (8 402 total, traitement: 0.0015 sec.)

Opérations sur les résultats de la requête

Version imprimable | Version imprimable (avec textes complets) | Exporter

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 30 > >> Page n°: 1 Tout afficher

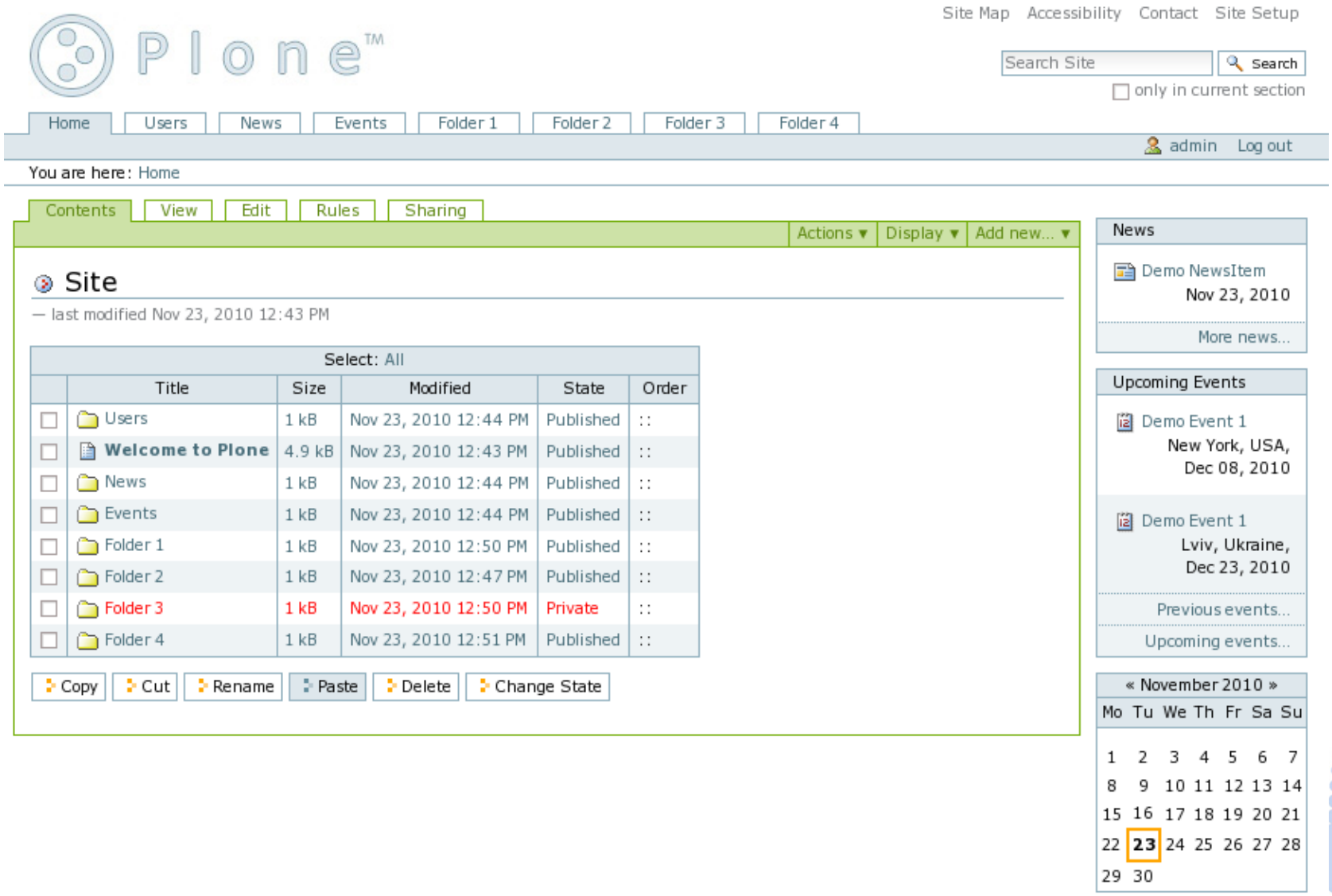
en mode horizontal et répéter les en-têtes à chaque groupe de 100

Trier sur l'index: aucune Exécuter

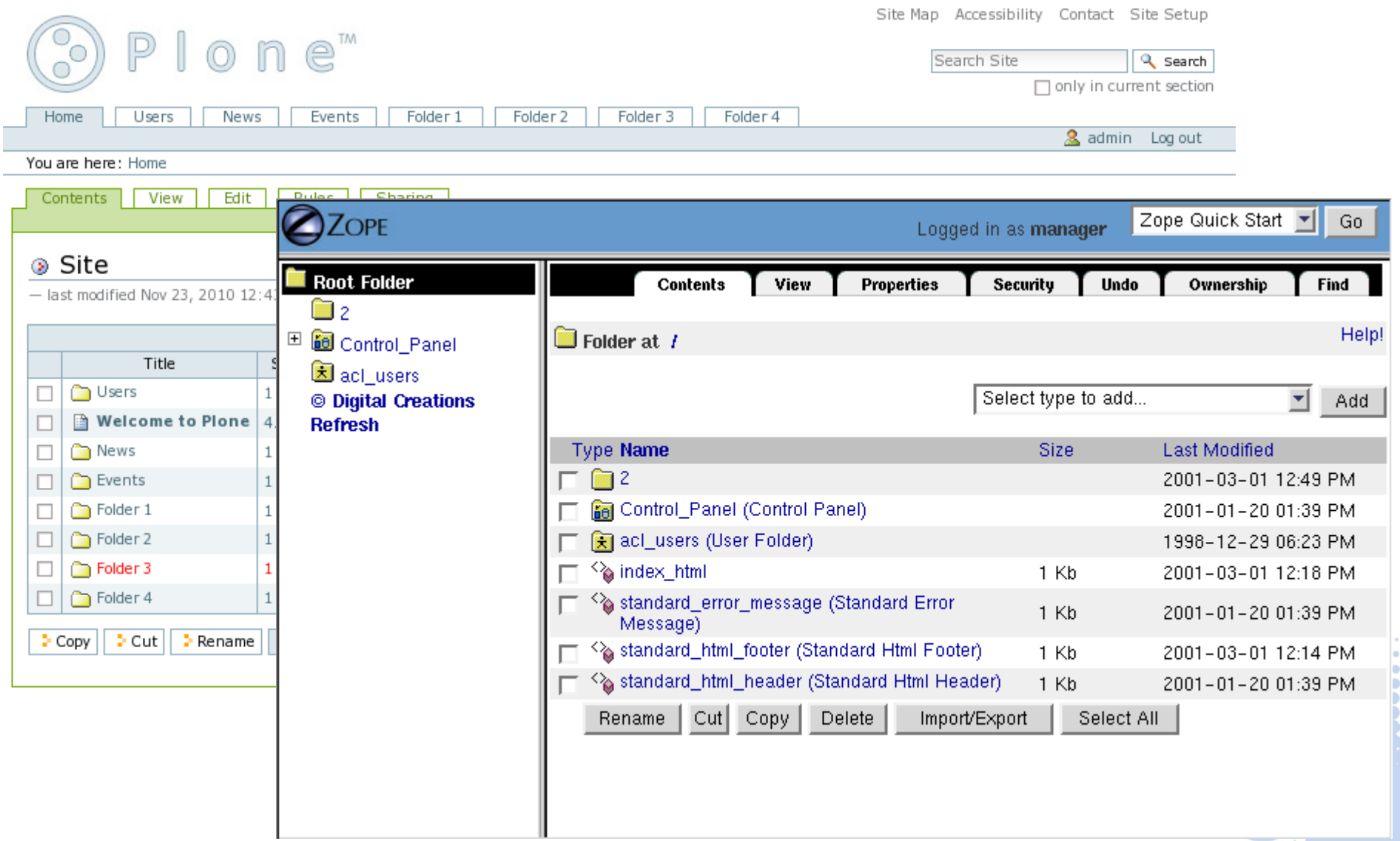
webpath Link to the model file	draw a figure of the data in a separate window	info Description of the model (physics, comments)	dust Dust composition (abridged)	incflux Incident flux shape (abridged)	geometry Geometry of the torus	sigma Width parameter of the angular distribution, i.e. boundary for a sharp torus, or appears in [exp(-beta/sigma)^p)], where p=2 for a Gaussian torus	Y Ratio of the external to internal radii	N Number of clumps along the radial direction	q Power-law index of the radial distribution [1/r^q]	tauV Visual optical depth of each clump	date Last modification date	filename Name of the model file
Download	Draw	?	ISM	p0.5-RJ1mic	TORUSG	15	30	20	3	20	Mercredi 07 Février 2007 à 20:32	SED-TORUSG-sig15-Y030-N20-q3-tv020.txt
Download	Draw					35	30	12	2	10	Jeudi 08 Février 2007 à 18:57	SED-TORUSG-sig35-Y030-N12-q2-tv010.txt
Download	Draw			RJ1mic		25	30	12	0	40	Jeudi 08 Février 2007 à 02:36	SED-TORUSG-sig25-Y030-N12-q0-tv040.txt

Models made by the Clumpy code - version 2.1 (https://www.pa.uky.edu/clumpy).
- The dust composition [ISM] has a standard Galactic composition of graphite and astronomical silicates (Draine 2003), the grains are spherical with MRN size distribution, whose mixture is replaced with a single-type composite grain of sublimation temperature 1500 K.
- The incident flux [p0.5-RJ1mic] is the standard AGN spectrum with a power-law index -0.5 from 0.1 up to 1 micron (Nenkova et al. 2002).
- The inner radius is Rd = 0.4 L45^0.5 pc, where L45 is the bolometric luminosity in units of 1E45 erg/s.
- The geometry [TORUSG] corresponds to a Gaussian torus, whose parameters are sigma (sig), Y, N, q, and tauV (tv).
- Spectral energy distribution of the torus only [SED].

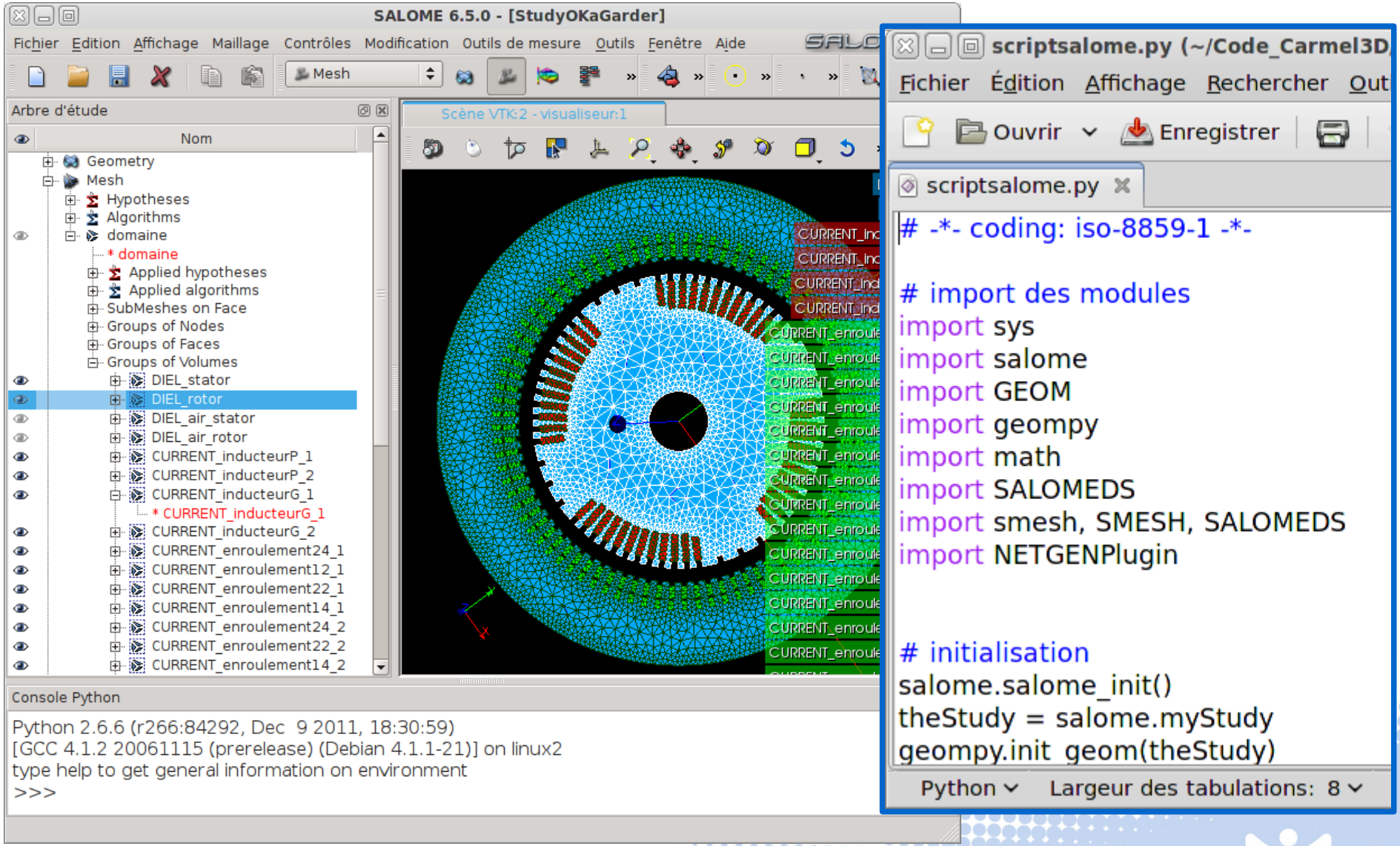
Site intranet DAFOP (Rectorat Lille) – Plone 3



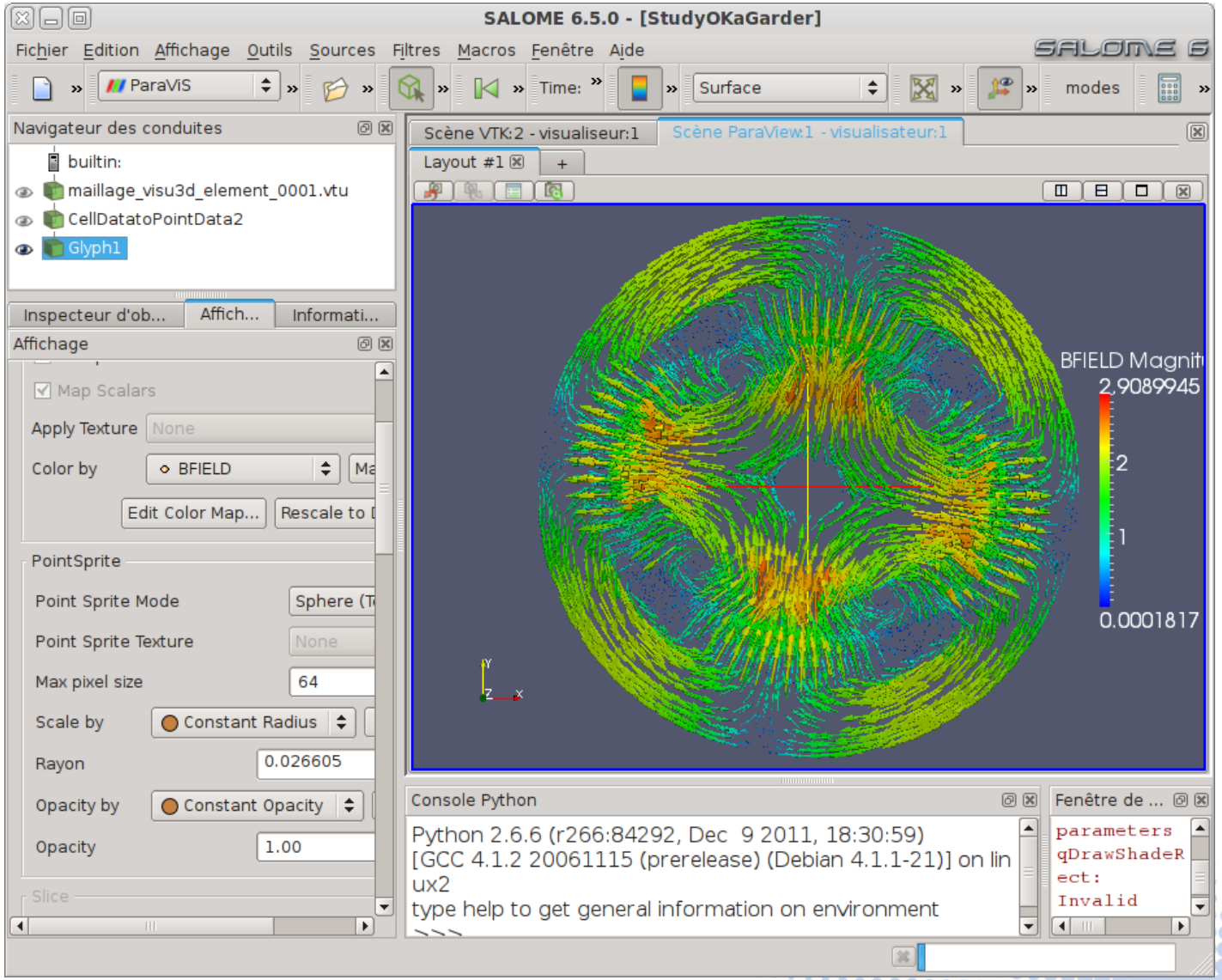
Site intranet DAFOP (Rectorat Lille) – Zope 2



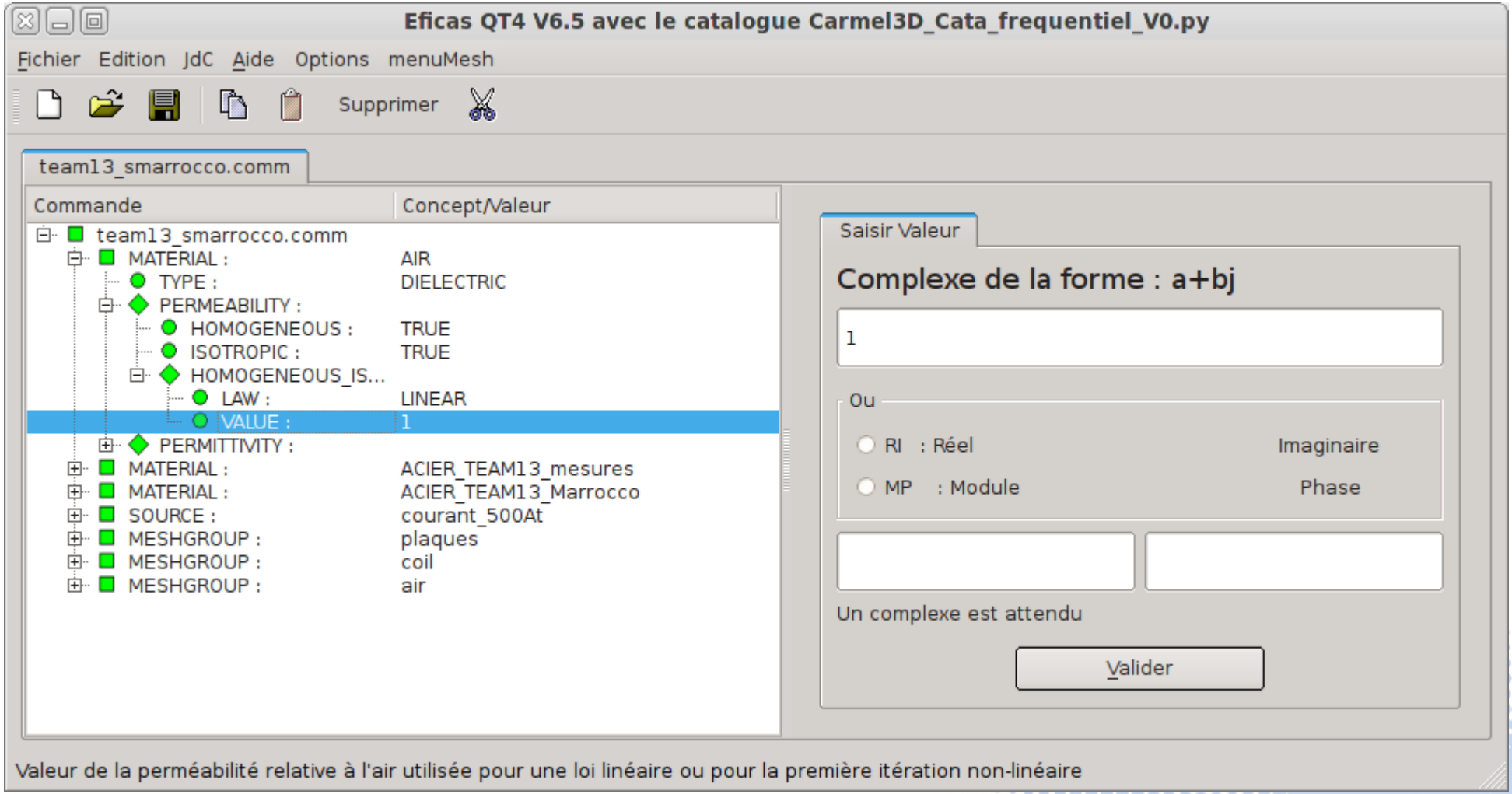
Salomé – maillage



Salomé – carte des champs



Salomé / Eficas – Mise en donnée Code_Carmel3D



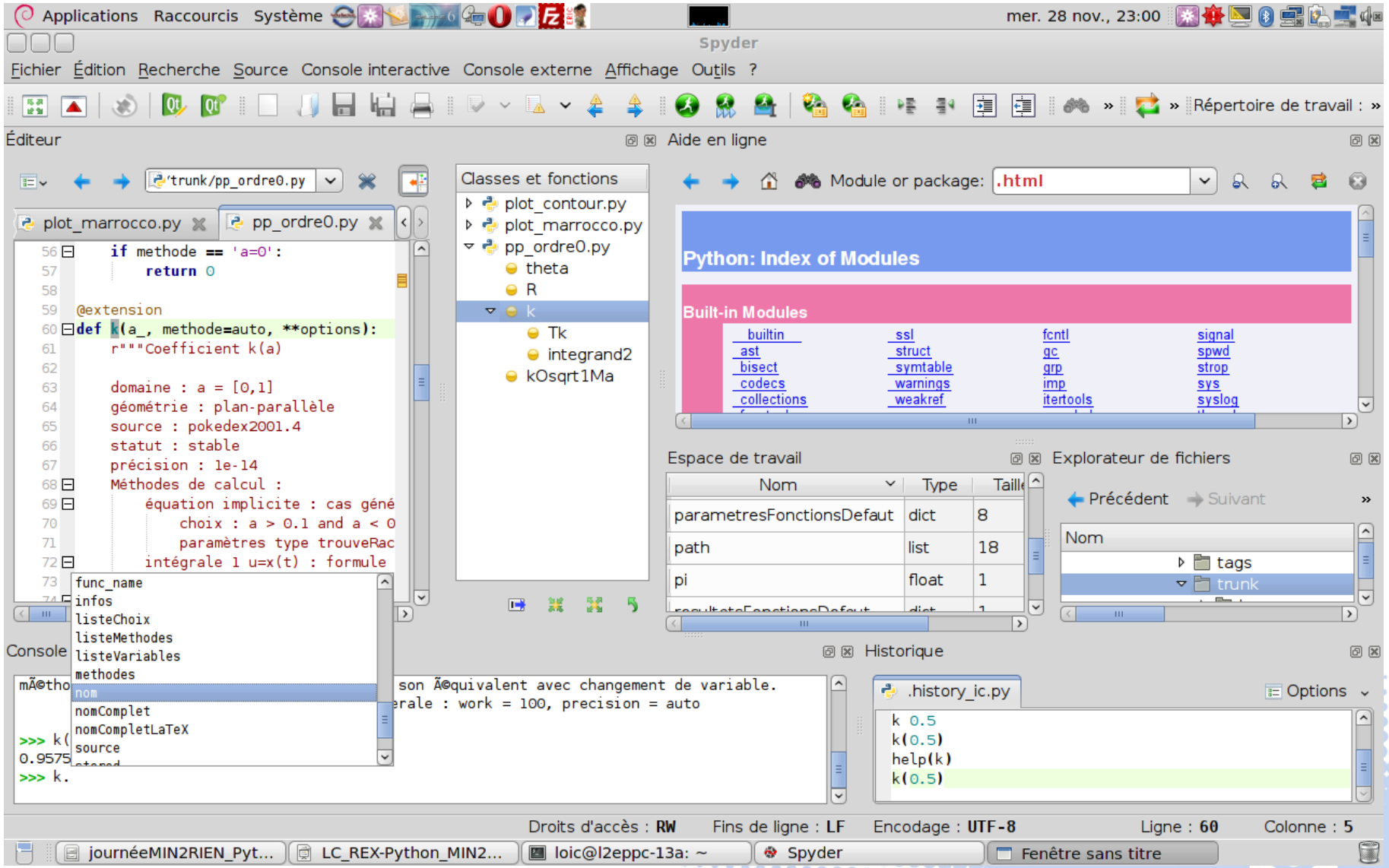
Code_Carmel3D – documentation source Doxygen

- **F90doc → Doxygen : 80 modules, 800 routines**
  - **sed/awk insuffisant, Python OK (313 lignes avec commentaires)**

```
!!
!! Code_Carmel3D
!!
!! Computes the barycenter ….
!!
!! (c) 2008-2012 EDF, USTL and ENSAM
!
module barycenter_module
  !
  ...
contains
  !
  !!
  !! Computation of the barycentric coordinates of
  !! a set of nx points x(:,1),x(:,2) ... x(:,nx)
  !! The coordinates of the points are given...
    subroutine get_barycentric_coord(x,v,lambda)
...
```

```
!
!> Code_Carmel3D
!!
!! Computes the barycenter ….
!!
!! (c) 2008-2012 EDF, USTL and ENSAM
!
module barycenter_module
  !
  ...
contains
  !
  !
  !> Computation of the barycentric coordinates of
  !! a set of nx points x(:,1),x(:,2) ... x(:,nx)
  !!
  !! The coordinates of the points are given...
    subroutine get_barycentric_coord(x,v,lambda)
...
```

Spyder : Scientific Python Development EnviRonment



Autres utilisations

- Création 100+ figures présentation (LaTeX Beamer),
- Conversion de figures Postscript pour présentation (appel gs),
- Scripts de lancement de calcul,
- Figures de qualité publication : Gnuplot, PyX,

- Editeurs : Spyder, Eric, vim, (emacs),
- Environnements de travail : Eric, Spyder, Ipython.

Formation « Python par la pratique » 2007

- Cours et tutoriel,
- TP1 : Installation Python et extensions (modules) en local (non root) sur Linux, MacOSX, Windows,
- TP2 : Shell script (programme en ligne de commande),
- TP3&4 : Manipuler des fichiers de données (ASCII, FITS),
- TP5 : Manipuler des tableaux et matrices (type image astrophysique),
- TP6 : Visualisation de données (2D) et figures qualité publication,
- TP7 : Utiliser des routines/fonctions écrite en Fortran77 (90),
- TP8 : Utiliser des routines/fonctions écrite en C (C++),
- TP9 : Accélérer des fonctions Python (language C intégré à Python),
- TP10 : Documentation automatique (HTML, PDF, etc.) de Python,
- TP11 : Interface web de programmes Python (Apache/CGI),
- TP12 : Interface graphique (Tk).

Historique

- **Guido Van Rossum, universitaire (12/1989),**
- Monty **Python**'s Flying Circus,
- **Python descendant ABC, Modula-3, C,**
- **Pré-requis :**
  - **Langage simple d'enseignement,**
  - **Plaisant pour développeurs Unix/C,**
  - **Extensible.**
- **Pas « trop proche » de la machine :**
  - **Typage dynamique,**
  - **Gestion automatique de la mémoire,**
  - **Syntaxe simple (lisible par utilisateur),**
  - **Programmes clairs (blocs indentés),**
  - **Langage compact (2.7.3 : 90 fonctions internes, 25 utilisés en pratique).**

abs, all, any, apply, basestring, bin, bool, buffer, bytearray, bytes, callable, chr, classmethod, cmp, coerce, compile, complex, **continue**, copyright, credits, delattr, **dict {}**, **dir**, divmod, **enumerate**, **eval**, execfile, **exit**, **file**, filter, **float**, **for**, **format %**, frozenset, getattr, globals, hasattr, hash, **help**, hex, id, **if**, **input**, int, intern, isinstance, issubclass, iter, **len**, license, **list []**, locals, long, map, **min**, **max**, memoryview, min, next, object, oct, **open**, ord, **pass**, pow, **print**, property, quit, **range**, raw_input, reduce, reload, repr, reversed, round, set, setattr, slice, sorted, staticmethod, **str**, sum, super, **tuple ()**, **type**, unichr, unicode, vars, **while**, xrange, zip.

Caractéristiques de Python (qualités), selon moi

« Python est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. » (Stéphane Fermigier, AFUL<=2003, 1999).

- **Gratuit (PSF, open-source, compatible GPL, commerce),**
- **Général (puissant),**
- **Syntaxe simple et claire (facile),**
- **Noyau léger (1 façon de faire),**
- **Compatible (12 ans ? OK),**
- **Compact (/3-5 vs. C, C++, Java),**
- **Bien documenté (tutoriels, fr, *interne*),**
- **Typage dynamique (a=1; a='toto'),**
- **Extensible (C, C++ ou Fortran),**
- **Interactif (>>> a=1 ; print a),**
- **Fourni avec les piles (std, PyPI:26046),**
- **Installation facile (python setup.py install)**

- Portable : multi-OS, Java, .NET,
- Exceptions = gestion d'erreurs,
- Projets légers à complexes,
- Gestion des ressources (mémoire),
- Pas de pointeurs explicites,
- Orienté-objet (C++, héritage multiple, surcharge),
- Types évolués,
- Evolution régulière,
- Dynamique (eval), orthogonal (base petite pour construction riche), réflectif (méta-programmation), introspectif (debugger, profiler en Python).

Peu de limites constatées

- **Lent (interprêté, gestion simplifiée), e.g., pyARTY x20,**
  - → **accélérations possibles**
- **F2PY ne supporte pas les types dérivés**
  - → **Forthon, F2PY+quippy, G3 F2PY dans NUMPY (en cours), fwrap (?)**
- **Objets : tout n'est pas redéfinissable (+, -, etc.)**
  - **contrôle des erreurs d'arrondi pyARTY ?**
- **Un seul thread**
  - → **modules thread, threading (standart),**

Python 2 vs. 3

- 2.7.3 (sécurité) et 3.3.0,
- >>> print 'a=',2 → print('a=',2)
- 2to3.py,
- Tout est unicode (variables) en Python 3,
- Librairie standart et extensions en cours de mise à jour.

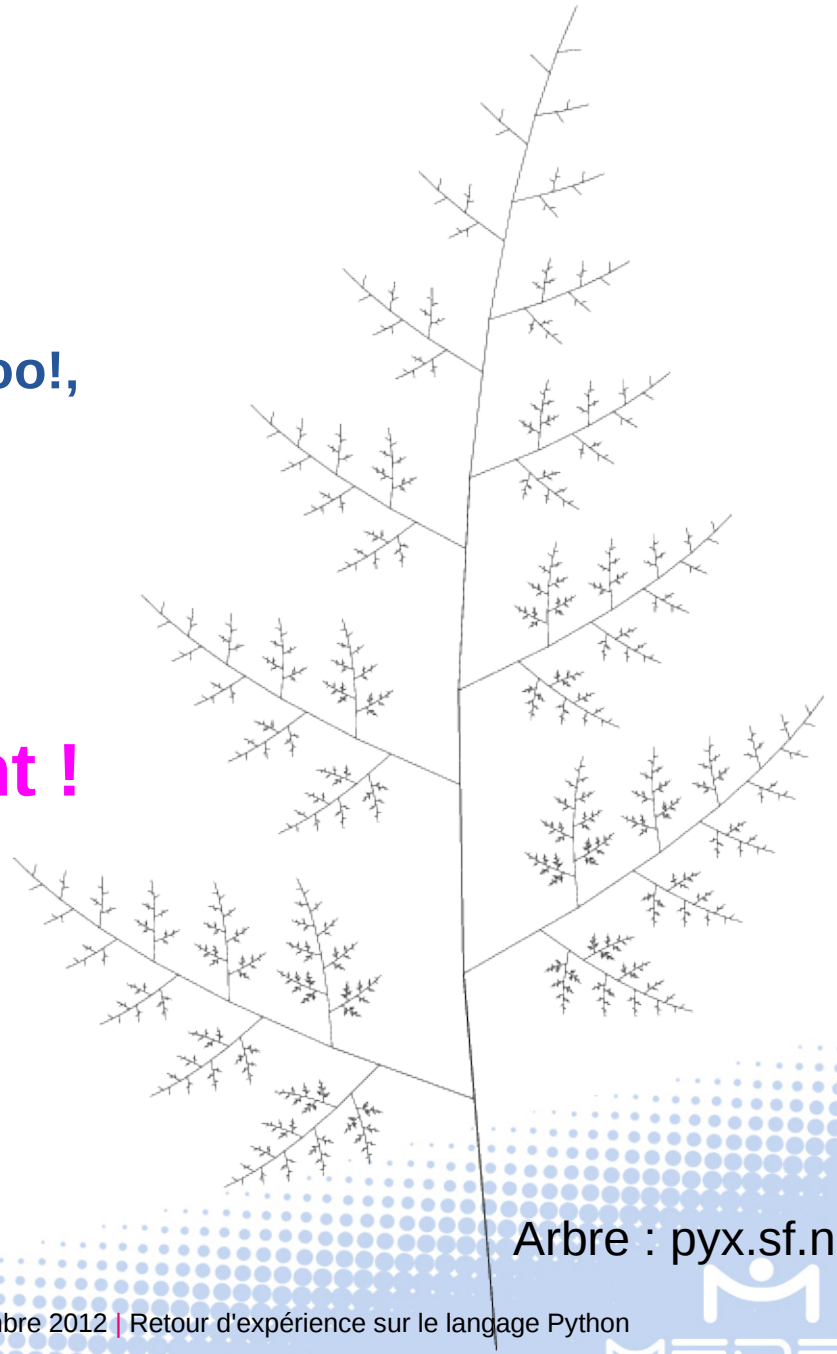
Ressources

- Un site web central : www.python.org,
  - Documentation, tutoriels, versions, extensions PyPI, etc.
- AFPY : association communauté française (www.afpy.org),
- PyconFR : rassemblement d'utilisateurs (www.pycon.fr),
- Livres payants : O'Reilly (en), EYROLLES (fr),
- Livres gratuits anglais (www.greenteapress.com),
- Didacticiel en ligne (http://cscircles.cemc.uwaterloo.ca/0-fr/).

Applications et conclusion

- Outils système RedHat Linux,
- Moteurs de recherche Google, Yahoo!,
- Gestion de projet logiciel Trac,
- Editeur de projet Eclipse + PyDev.

Python est enthousiasmant !



Arbre : pyx.sf.net