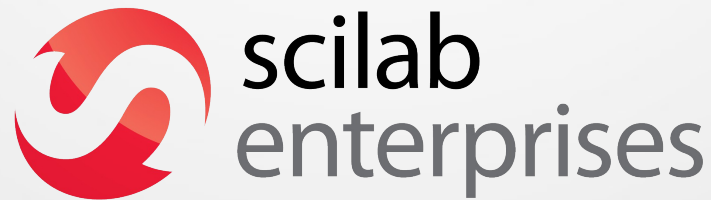


**FOSDEM**

Sylvestre Ledru / February 2nd, 2013



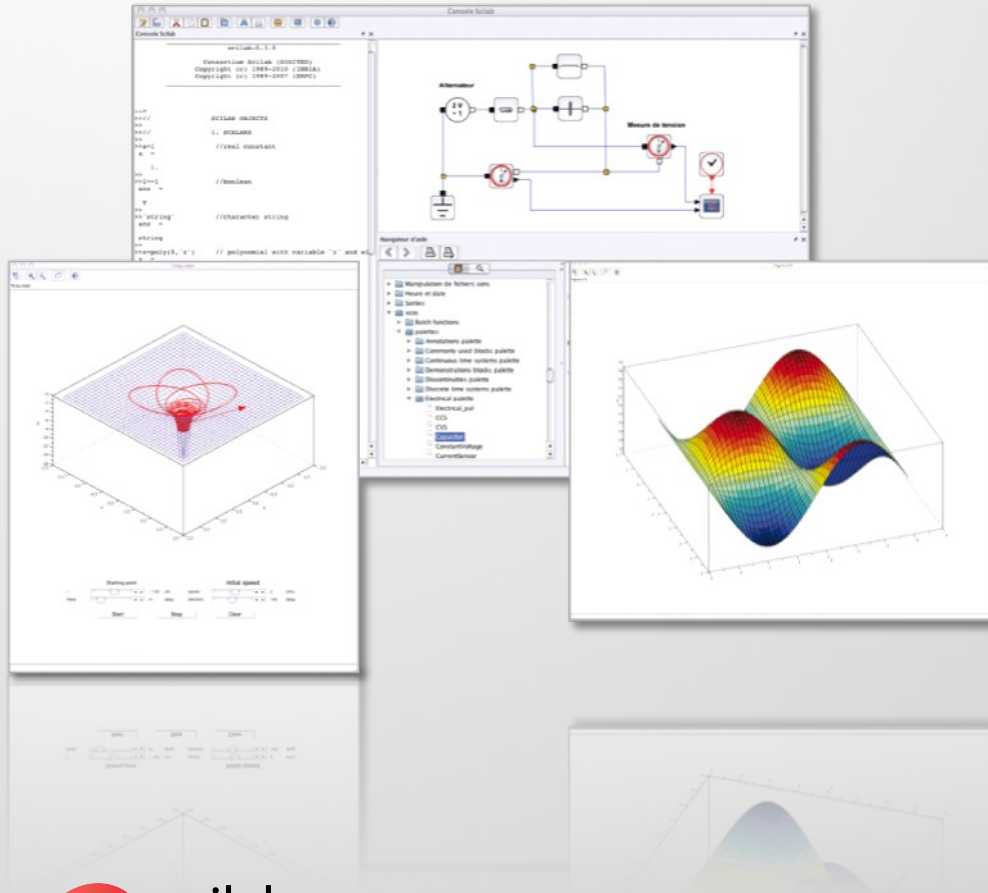
**Professional Services & Support for  
Scilab,**  
Free Open Source Software for Numerical  
Computation

# Sylvestre Ledru

- Operation manager at Scilab Enterprises
  - Responsible of GNU/Linux & Mac OS X
  - Community manager for Scilab
  - ... and also for IRILL
  - Debian Developer
- Hint : come to see me today tomorrow !

# Scilab Software

# Free and Open Source Solution



## Powerful computation software

- Numerical computation engine **easy to embed into applications**
- **Extended capabilities** with professional & specialized modules
- CeCILL license (GPL compatible)

# Scilab - CLI

```
Paramétrage de scilab-full-bin (5.3.3-2) ...
Paramétrage de scilab (5.3.3-2) ...
Paramétrage de scilab-doc (5.3.3-2) ...
Paramétrage de scilab-sivp (0.5.3-2) ...
Traitement des actions différées (« triggers ») pour « menu »...
[23:59:08][sylvestreglosin] ~$ scilab-cli

-----
                    scilab-5.3.3
-----

      Consortium Scilab (DIGITEO)
      Copyright (c) 1989-2011 (INRIA)
      Copyright (c) 1989-2007 (ENPC)
-----

Initialisation :
  Chargement de l'environnement de travail

-->a=2*[2,3]
a =

    4.    6.

-->
```



Navigateur de fichiers

r.play/scilab/modules/core/

Nom

- core
  - ..
  - build
  - demoss
  - etc
  - examples
  - help
  - includes
  - jar
  - locales
  - macros
    - OS\_Version.bin
    - OS\_Version.sci
    - buildmacros.bat
    - buildmacros.sce
    - check\_modules\_xml.bin
    - check\_modules\_xml.sci
    - check\_versions.bin
    - check\_versions.sci
    - cleanmacros.bat
    - edit.bin
    - edit.sci
    - getshell.bin
    - getshell.sci
    - lib
    - names
    - perl.bin
    - perl.sci
    - toolboxes.bin
    - toolboxes.sci
    - typeof.bin
    - typeof.sci
    - ver.bin
    - ver.sci

Filtre fichier/répertoire

 Respecter la casse  Express...

Console Scilab

```

immediate_drawing = "on"
background = -2
visible = "on"
rotation_style = "unary"
event_handler = ""
event_handler_enable = "off"
user_data = []
tag = ""

```

--&gt;banner

```

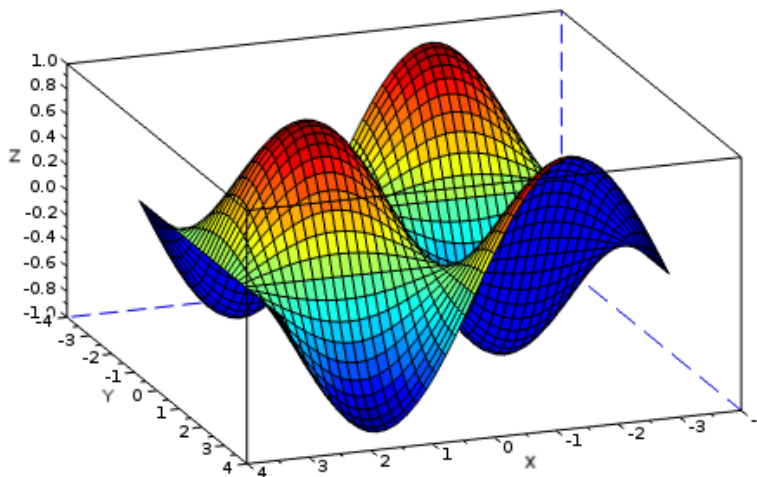
-----
                    scilab-branch-master

                    Scilab Enterprises
                    Consortium Scilab (DIGITEO)
                    Copyright (c) 1989-2012 (INRIA)
                    Copyright (c) 1989-2007 (ENPC)
                    -----

```

--&gt;

Figure n°0



Navigateur de variables

Nom	Dime...	Type	Visibil...
h	1x1	Handl...	local
a	122x2	Double	local
string...	279x1	Chaîn...	local

Historique des commandes

```

exit
// -- 14/02/2012 18:34:21 -- //
exit
// -- 14/02/2012 18:54:09 -- //
a=rand(122,2)
a=rand(122,2);
a=rand(122,2)
string=ls
string=ls()
plot3d()
f=gcf()
ls
clear
banner
clear
clf
plot3d()
stringaze=ls()
a=rand(122,2)

```

# With Embedded Applications

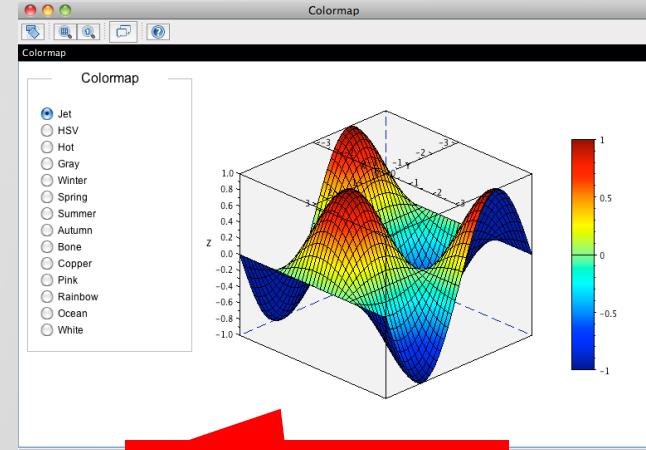
```

1 //Définition directe de la transformée de Fourier discrète
2 //-----
3 function xf=DFT(x, flag);
4 n=size(x, '*');
5 //Calcul de la matrice de Fourier (n by n !)
6 if flag==1 then, //transformation inverse
7 am=exp(2*pi*i*(0:n-1)*(0:n-1)/n);
8 else //transformation directe
9 am=exp(-2*pi*i*(0:n-1)*(0:n-1)/n);
10 end
11 xf=am*matrix(x,n,1); //dft
12 xf=matrix(xf,size(x)); //mise en formz
13 if flag==1 then,xf=xf;/end
14 endfunction
15
16 //Comparaison avec l'algorithme de la transformée rapide:
17 a=rand(1,1000);
18 norm(DFT(a,1) - fft(a,1))
19 norm(DFT(a,-1) - fft(a,-1))
20
21 timer();DFT(a,-1);timer()
22 timer();fft(a,-1);timer()
    
```

Editor

	1	2	3	4	5
a	0,2113	0,6654	0,8782	0,7264	
b	0,756	0,6284	0,0684	0,1985	
c	2,2113e...	0,8497	0,5608	0,5443	
d	0,3303	0,6857	0,6624	0,2321	
e					
f					
g					
h					
i					

Variable Editor



2-D/3-D Visualization

**ode\_discrete**  
ordinary differential equation solver, discrete time simulation

**Calling Sequence**  
y=ode('discrete',y0,t0,tmax,tstep,f)

**Arguments**  
y0: real vector or matrix (initial conditions).  
t0: real scalar (initial time).  
f: external i.e. function or character string or list.  
tmax: integer (final time).  
tstep: integer vector.

**Description**  
With this syntax (first argument equal to 'discrete') ode computes recursively  $y(k+1)=f(k,y(k))$  from an initial state  $y(k_0)$  and returns  $y(k)$  for  $k$  in tmax. tstep(1) must be greater than or equal to t0.

**Examples**  
y1=[1;2;3]; t0=0; tmax=10; tstep=1; y0=y1; f=@(k,y) [y(1)+y(2); y(2)+y(3); y(3)+y(1)];  
y=ode('discrete',y0,t0,tmax,tstep,f);  
// New y evaluation at (t0,y0,y1)  
y0=[discrete',y1,t0,tmax,f\_function]

Embedded Help

External Modules Manager

**Aerospace - ATOMS**

**CelestLab**

Version: 2.1.1-1  
Auteur(s):  
Description:  
CelestLab is a library of space flight dynamics functions written in Scilab. This library has been developed by CNES (Centre National d'Etudes Spatiales) for mission analysis purposes. It is used for trajectory analysis and orbit design for various types of missions (around Earth, interplanetary...).

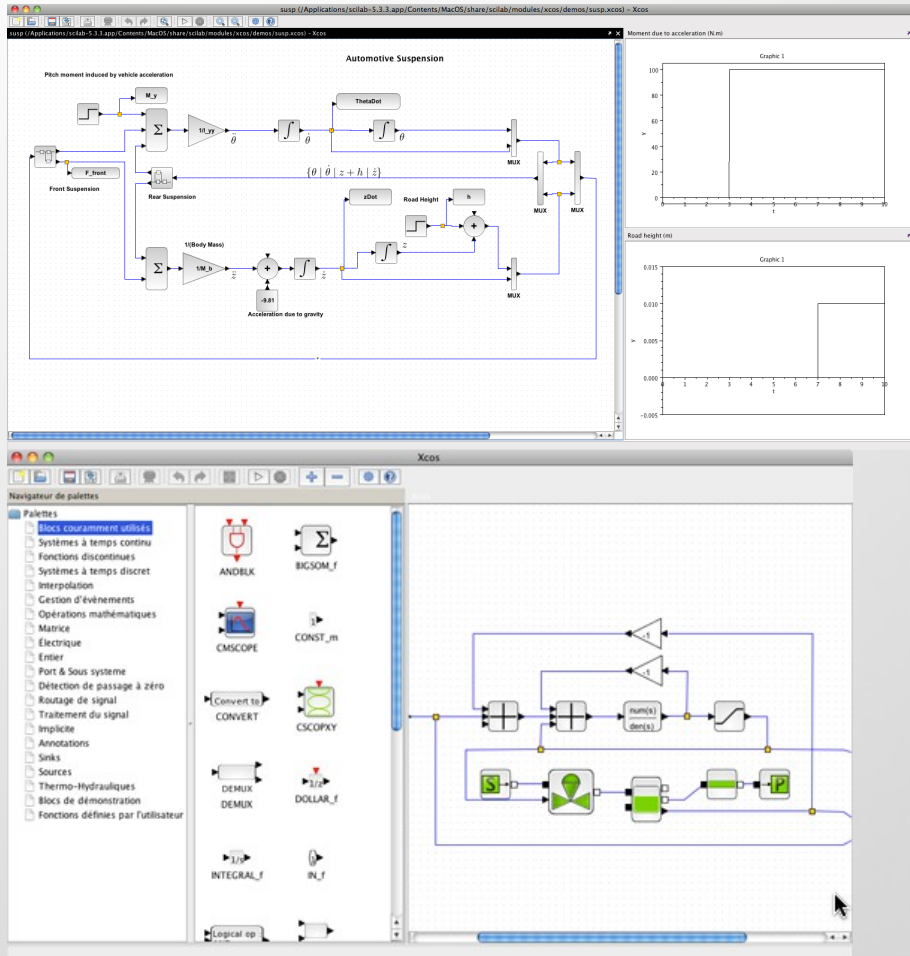
You may leave comments below (any remark, suggestions...). But if you would like to report bugs, please go to: <http://forge.scilab.org>

Voir aussi:  
<http://atoms.scilab.org/forum/viewtopic.php?p=11>

Date de sortie: 2011-01-06  
Taille du téléchargement: 2 Mo

Installer

# And Xcos, Modeling & Simulation of Dynamic Systems



- **Professional tool for industrial needs**
- Intuitive and ergonomic interface
- **Model building, edition and customization**
- Embedded Modelica Compiler
- **Freely available and distributed with Scilab**



# What for ?

---

- Scilab can be used:
  - A powerfull calculator
  - To develop complex applications
  - As a prototyping application
  - ...

# What for ? (2)

---

- Scilab can be used:
  - Link and use a load level library into a high level language
  - Computing engine
  - Control external devices
  - Anything ?

# History of Scilab

# History of Scilab



- Started in the mid 80
- Inspired by the Matlab fortran
- Fortran was too complex to handle matrices
- Needed to do researchs at Inria for CACSD (Computer Aided Control System Design)
- Called Blaise

# History of Scilab

---

- Commercialisation through Simulog under the name Basile in 1984
- First release (1.1) as *opensource* software in 1994
- From 2003 to 2008, through the Scilab consortium hosted by Inria
- Change of licence to CeCILL (GPL compatible) in May 2008

# History of Scilab

---

- Phase 2 : From 2008 to 2012, the Scilab consortium is hosted by the Digiteo foundation
- Industrialisation of the product:
  - Strong focus on usability, look and feel and user experience.
  - Stability
  - Improvement of the documentation
  - Legacy management

# History of Scilab

---

- 2011 : *Scilab Entreprises* created for the *classical* open source business model  
Most of the current employees being founders  
Spin off of Inria
- Currently 17 employees at Versailles

# Scilab Enterprises

- Focus on Scilab and its ecosystems
- Manage the software, its extensions to provide a full numerical platform within the production context of the customers
- Move from a research environment to a software editor



# Services & Support

- Free software  
=> Important and strong diffusion
- The main alternative to Matlab / Simulink
- We are the best to help on Scilab and its extensions

# Services & Support

- Development and optimization of customer applications
- Realization of in-house optimized, customized or extended versions of Scilab

# Services & Support

- Scilab Long Term Support
- Migrations to Scilab...  
From Excel or Matlab/Simulink

# Services & Support

- Training
- Commercial external modules

# Free software and industry

## Used for

- Design of rockets (Ariane)
- Computation of spatial trajectories (ATV)
- Design of future planes (Falcon)
- Modelisation of geochimist reactions
- Modelisation of stamping of cars
- ...

# Advantages for customers

- Cost
- Credible alternative to proprietary solutions
- *Friendly* license : Easy deployment

# Advantages

- Access to the source code
- Independance from a single editor
- Close relationship with the editor



# Drawbacks

- More complex business model
- Development on the software are usually on the corporation expensives
- The software is free, why should I pay anything ?

**Quality : It is about tools**

# Requirements

- Definition of clear process about the inclusions of new features, bug fixing, etc
- Unitary tests for new features
- Non regression tests with bug fixing (about ~3 000 tests)
- Each new feature should be documented (!) with examples and images if relevant

# Rules

- Definition of coding style for the various languages (C, C++, Java, Scilab, etc)
- Integration of hooks in git to apply them automatically (astyle is your friend)

# Deployment of tools

- Nightly build
- Tests exectuions
- Continuous integrations (Jenkins)  
Build with various options (minimal, full, other compilers, etc)  
Produce :
  - Scan-build results
  - Code coverage

# scan-build

```
265     /* Now calling umfpack routines */
266     if (A.it == 1)
267     {
268         stat = umfpack_zi_symbolic(mA, nA, A.p, A.irow, A.R, A.I, &Symbolic, Control, Info);
269     }
270     else
271     {
272         stat = umfpack_di_symbolic(mA, nA, A.p, A.irow, A.R, &Symbolic, Control, Info);
273     }
274
275     if ( stat != UMFPACK_OK )
276     {
277         freeCcsSparse(A);
278
279         Scierror(999, _("%s: An error occurred: %s: %s\n"), fname, _("symbolic factorization failed"), stat);
280         return 1;
281     }
```

17 ← Taking true branch →

18 ← Assuming 'stat' is not equal to 0 →

19 ← Taking true branch →

20 ← Memory is never released; potential leak of memory pointed to by 'pdBIB'

# Code coverage (Icov)

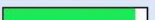





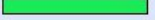
Current view: [top level](#) - core/src/cpp

Test: [scilab-code-coverage.info](#)

Date: 2013-02-01

Legend: Rating: low: < 75 % medium: >= 75 % high: >= 90 %

	Hit	Total	Coverage
Lines:	203	216	94.0 %
Functions:	29	31	93.5 %
Branches:	154	197	78.2 %

Filename	Line Coverage ( <a href="#">show details</a> )	Functions	Branches
<a href="#">backtrace_print.cpp</a>	 91.9 % 34 / 37	100.0 % 1 / 1	52.6 % 20 / 38
<a href="#">demangle.cpp</a>	 100.0 % 2 / 2	100.0 % 1 / 1	- 0 / 0
<a href="#">hashtable_core.cpp</a>	 90.1 % 64 / 71	88.2 % 15 / 17	61.0 % 25 / 41
<a href="#">namstr.cpp</a>	 100.0 % 28 / 28	100.0 % 1 / 1	100.0 % 20 / 20
<a href="#">partition.hxx</a>	 100.0 % 8 / 8	100.0 % 2 / 2	100.0 % 8 / 8
<a href="#">search_functions.cpp</a>	 94.8 % 55 / 58	100.0 % 3 / 3	93.8 % 30 / 32
<a href="#">unrolled_algorithms.hxx</a>	 100.0 % 12 / 12	100.0 % 6 / 6	87.9 % 51 / 58

# Code coverage (Icov)

```
    {
46 : XMLNodeSet::XMLNodeSet(const XMLDocument & _doc, xmlXPathObject * _xpath):XMLList(),
    {
46 :     nodeSet = xpath->nodesetval;
46 :     if (nodeSet)
    {
46 :         scope->registerPointers(nodeSet, this);
46 :         size = nodeSet->nodeNr;
    }
    else
    {
0 :         size = 0;
    }
46 :     scilabType = XMLSET;
46 :     id = scope->getVariableId(*this);
46 : }

123 : XMLNodeSet::~XMLNodeSet()
    {
41 :     scope->unregisterPointer(nodeSet);
41 :     scope->removeId(id);
41 :     xmlXPathFreeObject(xpath);
82 : }

0 : void *XMLNodeSet::getRealXMLPointer() const
    {
0 :     return static_cast < void *>(nodeSet);
    }
```



# **Transition from a research project to a software editor**



# Transition from a research project to a software editor

---

- From *politic* perspective
  - Objectives ?
  - New features ?
  - Roadmap
  - Time constraints

# Transition from a research project to a software editor

- From the human perspective
  - Hard to change the mentalities
    - Most of the developers hate constraints!
  - Being a developer is an actual job as researcher is
  - Engineers stay longer (INRIA: 2 to 5 years)
  - Some contributors do not accept that
  - Some users do not accept that

# Transition from a research project to a software editor

---

- From a technical perspective
  - Things are not done the same way
  - Uniformisation
  - Importance of the technological choices
  - Importance of the dependencies (libraries)
  - Clean process

# Transition from a research project to a software editor

- Example : Code review
- Each commits to Scilab code has to go through a code review process
  - Pro :
    - Management is easier
    - Better quality
    - Easier to force requirements
    - Every follows the same rules
  - Con :
    - Slower
    - Can frustrate some devs

# Transition from a research project to a software editor

- Classic example: Inclusion of thirdparty sources into the source tree

Pro:

- Can be patched
- Do not need thirdparty libraries installed on the system (do not need of a complex ./configure)
- Do not need to interact with upstream

Con:

- Unmaintainable on a long run
- Hard to follow new upstream releases
- Some bugs are not forwarded upstream

# Transition from a research project to a software editor

- Clean process ?
  - How to close a bug ?
  - How to remove a deprecated feature from the language ?
  - How to handle major and minor releases ?
  - How to integrate a new feature into the language ?
  - ...

# Transition from a research project to a software editor

- Example: How to integrate a new feature ?
  - Write a SEP – Scilab Enhancement Proposal
    - What is it supposed to do ?
    - What would be the profile of the function ? (when applies)
    - How is it going to work ?
    - What is the expected behaviour with other existing functions ?
    - Which version is targeted ?
  - Validation



