

# Scilab

**Le calcul numérique libre pour l'enseignement, la  
recherche et l'industrie**

Claude Gomez  
Directeur général  
Scilab Enterprises

# Scilab : le logiciel et l'organisation



- Créée en 2010, issue d'Inria, suite du consortium Scilab
- Exclusivité de la marque, du développement et de la diffusion de Scilab

Services et produits

Distribution

Développement

Scilab = logiciel **gratuit**

- Formation, support
- Migrations vers Scilab
- Scilab LTS (« Long Term Support »)
- Réalisation de versions adaptées
- Développement et optimisation d'applications clients
- Modules externes commerciaux

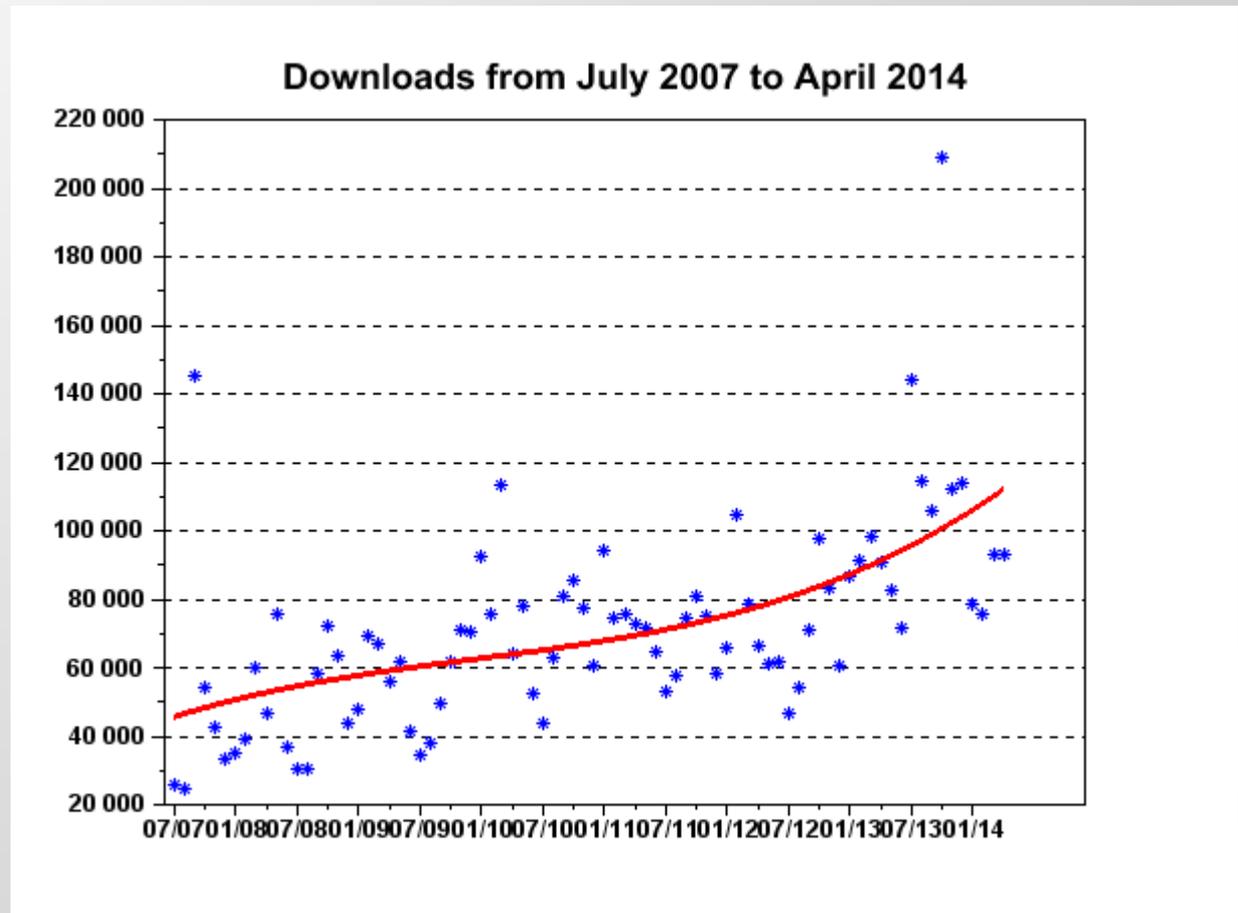
Scilab Enterprises en avril 2014 :

- 23 personnes
- Clients principaux : Airbus Group, ArcelorMittal, Dassault Aviation, CEA, CNES, DGA, EDF, SANOFI, THALES



# Scilab dans le monde

Environ 100 000 téléchargements mensuels à partir de [www.scilab.org](http://www.scilab.org) depuis 150 pays



# La distribution Scilab

## ■ Scilab

La super calculatrice graphique et programmable

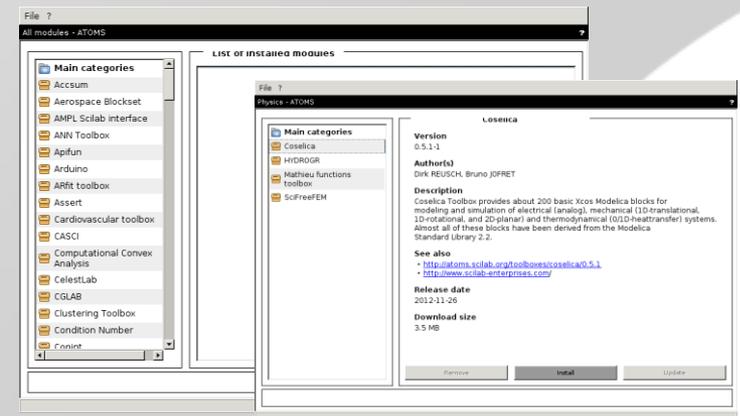
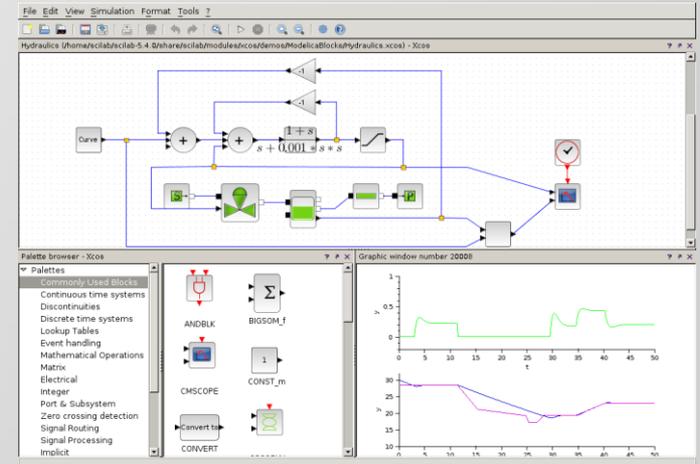
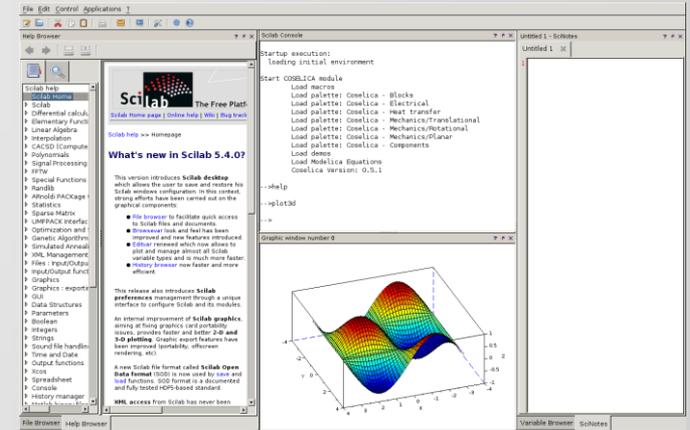
## ■ Xcos

Modélisation et simulation des systèmes dynamiques

## ■ ATOMS

(AuTomatic mOdules Management for Scilab)

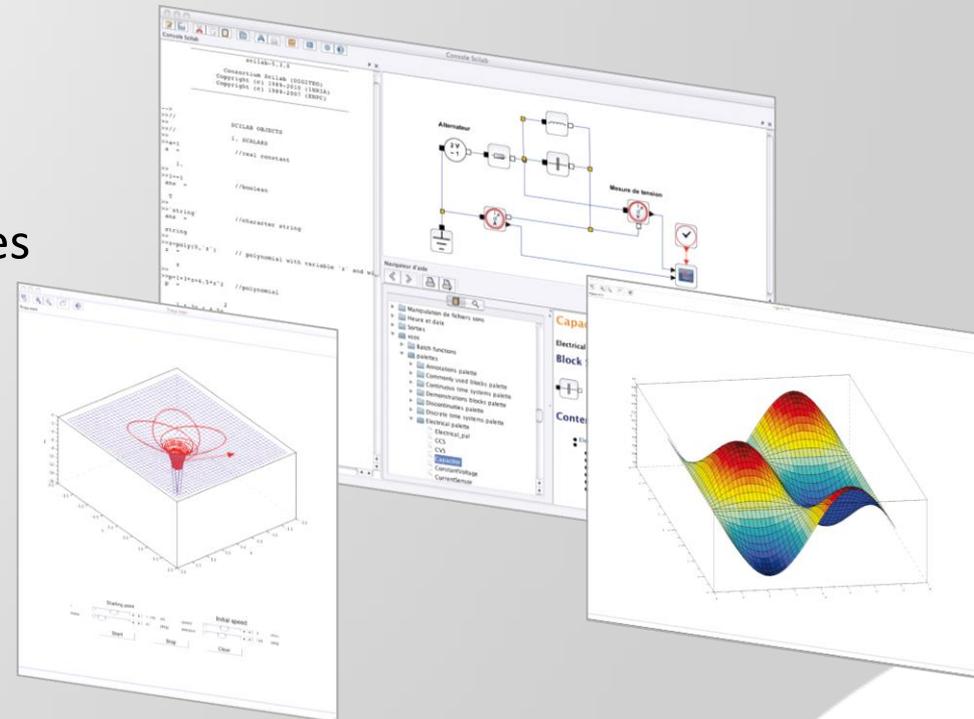
Gestion des modules externes



# Scilab : les fonctionnalités

Plus de 2 000 fonctions :

- Fonctions mathématiques
- Calcul matriciel, matrices creuses
- Polynômes et fractions rationnelles
- Simulation : équations différentielles
- Commande classique et robuste, optimisation LMI
- Optimisation différentiable et non différentiable
- Interpolation, approximation
- Traitement du signal
- Statistiques



# Un environnement convivial : programmation facile

Explorateur de fichiers

Console

Gestion des variables

The screenshot displays the Scilab environment with three main windows:

- Navigateur de fichiers (File Explorer):** Shows the file system structure with folders like Applications, Library, Network, System, Users, Volumes, bin, cores, dev, etc., home, net, private, sbin, tmp, usr, var, and mach\_kernel.
- Console Scilab:** Shows the Scilab version (5.4.0-alpha-1) and copyright information. It displays the initialization process and the execution of commands: `a=rand(4,4)` resulting in a 4x4 matrix of random numbers, `spec(a)` resulting in a 4x4 matrix of eigenvalues, and `b=testmatrix('magic',3)` resulting in a 3x3 magic square matrix.
- Navigateur de variables (Variable Manager):** A table listing variables with their names, dimensions, types, and visibility.

Nom	Dimension	Type	Visibilité
b	3x3	Double local	
a	4x4	Double local	
home	1x1	Chaîne local	
PWD	1x1	Chaîne local	
%tk	1x1	Booléen local	
%F	1x1	Booléen local	
%T	1x1	Booléen local	
%nan	1x1	Double local	
%inf	1x1	Double local	
SCI	1x1	Chaîne local	
SCIHOME	1x1	Chaîne local	
TMPDIR	1x1	Chaîne local	
%gui	1x1	Booléen local	
%fftw	1x1	Booléen local	
%t	1x1	Booléen local	
%f	1x1	Booléen local	
%eps	1x1	Double local	
%io	1x2	Double local	
%i	1x1	Double local	
%e	1x1	Double local	
%pi	1x1	Double local	
%modalWarning	1x1	Booléen global	
%driverName	1x1	Chaîne global	
%exportFileNa...	1x1	Double global	
%toolboxes	1x1	Double global	
%toolboxes_dir	1x1	Chaîne global	
%helps	1x1	Double global	

**Historique des commandes (Command History):**

```
// -- 01/01/2012 10:42:45 -- //  
a=rand(4,4)  
spec(a)  
b=testmatrix('magic',3)
```

Historique de Commandes





# Pourquoi utiliser Scilab ?

## **Scilab est un logiciel libre = gratuit**

- Facile à installer partout
- Grande communauté d'utilisateurs

**Mais la liberté ne suffit pas...**

## **Un logiciel convivial avec de nombreuses fonctionnalités**

- La plupart des fonctions pour les mathématiques appliquées
- Graphique intégré basé sur OpenGL
- Module Xcos comparable à Simulink
- Facilité pour rajouter des modules externes

## **Le type de logiciel utilisé par les ingénieurs**

- Pour tout le calcul numérique
- Grandes et petites entreprises

## **Prise en charge totale de Scilab par une organisation**

- Scilab développé professionnellement par Scilab Enterprises
- Supports et services

# Education

Scilab dans les programmes des CPGE depuis la rentrée 2013 :

- Classes préparatoires économiques et commerciales
- Classes préparatoires scientifiques

Des livrets avec des modules dédiés :

- Scilab pour les mathématiques dans le secondaire
- Scilab / Xcos pour les STI

Un logiciel qui sera ensuite utilisé  
dans le monde industriel

# Comment utiliser Scilab ?

**Scilab = un calculateur graphique  
simple et puissant**

# Un langage matriciel

- Le type par défaut de Scilab est la matrice de doubles
- Un nombre est considéré comme une matrice de doubles de taille 1x1

- Définition d'un vecteur :

$$A = [ 1, 2, 3, 4 ] \quad [ 1 \quad 2 \quad 3 \quad 4 ]$$

- ou bien :

$$A = [ 1; 2; 3; 4 ] \quad \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

- Définition d'une matrice :

$$A = [ 1, 2, 3, 4 ; 5, 6, 7, 8 ] \quad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

▪  $B = A(1,1)$  // extrait l'élément (1,1)

▪  $C = A(2,4)$

▪  $D = A(5,2)$

$E = A(12)$  // extrait le 12<sup>e</sup> élément

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

- $B = A(:,1)$  // extrait la première colonne
- $C = A(\$,:)$  // extrait la dernière ligne
- $D = A(2:\$-1,:)$  // extrait toutes  
// les lignes de 2 à la dernière -1

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

$x = [1,3,5]$

$E = A(x,x)$  // extrait les couples  
// (1, 3, 5) x (1, 3, 5)

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

- Addition :  $C = A + B$
- Soustraction :  $D = A - B$
- Multiplication matricielle :  $A * B$
- Multiplication « élément par élément » :  $A .* B$
- Produit de Kronecker :  $A .* . B$

$$A .* . B = \begin{bmatrix} A_{1,1} * B & \dots & A_{1,n} * B \\ \vdots & & \vdots \\ A_{m,1} * B & \dots & A_{m,n} * B \end{bmatrix}$$

- Puissance  $^{\wedge}$  :
  - Multiplication matricielle :  $B = A * A = A^{\wedge}2$
  - Élément par élément :  $B = A.^{\wedge}2$
  
- Transposée (conjuguée)  $'$  :  $B = A'$

- Résolution d'un système linéaire :  $\backslash$  (multiplication par l'inverse)
  - Solution de  $A * x = b$  :  $x = A \backslash b$   
( $x = A^{-1} * b$  si  $A$  est carrée et inversible)

Attention : l'inverse n'est pas calculée !

- La fonction **find** retourne les indices des éléments d'une matrice qui satisfont une condition :
  - pour trouver les indices des éléments d'une matrice qui ont une valeur comprise entre 50 and 100 :  
**`find(80<M & M<100)`**
  - pour changer en 1 la valeur des éléments d'une matrice qui ont une valeur comprise entre 50 et 100 :  
**`M(find(80<M & M<100))=1`**

Grande efficacité !

# Quelques exemples

- Calculs matriciels rapides :

```
A=rand(1000,1000); b=rand(1000,1);  
x=A\b; norm(A*x-b)  
vp=spec(A);
```

- Tracés 2D :

```
plot(real(vp), imag(vp), "*r");  
x=linspace(-%pi,%pi,1000);  
clf; plot(x, sin(x), "r", x, cos(x), "g");
```

- Courbe 3D :

```
k=tan(%pi/27); t=linspace(-40,40,1000);  
x=cos(t)./cosh(k*t); y=sin(t)./cosh(k*t); z=tanh(k*t);  
clf; param3d(x,y,z);
```

## ■ Image de projection 2D :

- Un fichier texte de données : 105 Mo
- 10 millions de points = 3817 abscisses, 2881 ordonnées
- Valeurs de  $z$  entre 1 et 256
- But = faire une projection en 2D dans le plan ( $xOy$ ) avec des couleurs correspondant aux valeurs

1. On lit le fichier et on met les données dans la matrice **M** :

```
M=fscanfMat("mandel.txt");
```

2. On ouvre une fenêtre graphique, on choisit une belle colormap et on trace les points colorés selon leur valeur :

```
f=scf(1);
```

```
f.color_map = rainbowcolormap(256);
```

```
Matplot(M);
```

On veut modifier les données : on change la valeur de certains points.

1. On change en 1 la valeur des points qui ont une valeur comprise entre 80 and 210 :

```
M(find(80<M & M<210))=1;
```

2. On refait le tracé dans une nouvelle fenêtre graphique :

```
f=scf(2);
```

```
f.color_map = rainbowcolormap(256);
```

```
Matplot(M);
```

#### **Accélération de la lecture du fichier texte :**

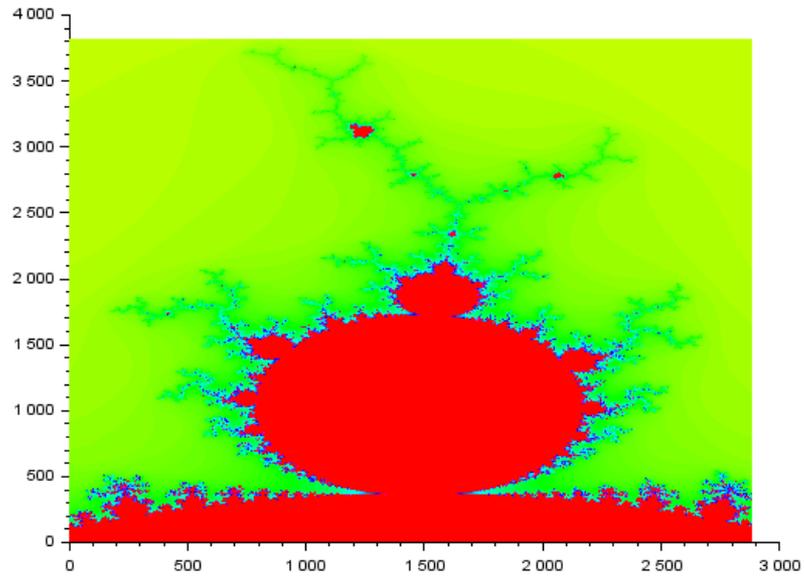
1. On sauve la matrice dans un fichier binaire Scilab SOD (Scilab Open Data) basé sur le standard HDF5 : 0,3 seconde

```
save("mandel.sod", "M");
```

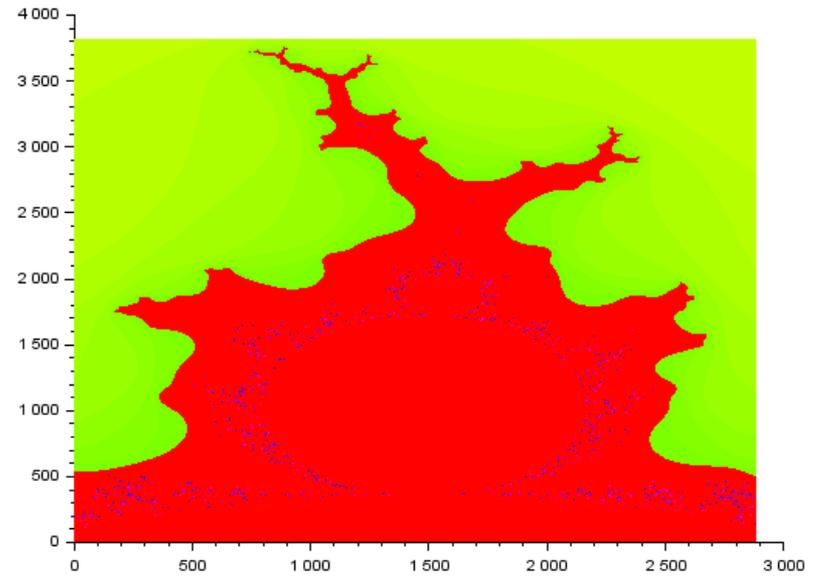
2. Le chargement dans Scilab est maintenant très rapide : 0,9 seconde

```
load("mandel.sod");
```

## Les deux tracés :



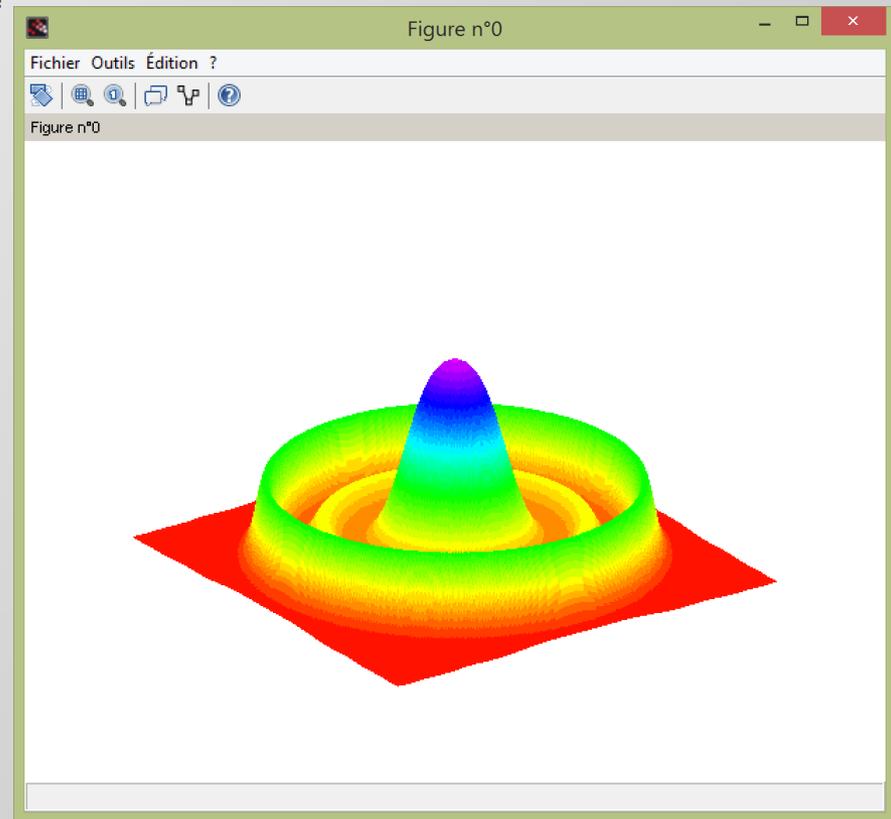
Premier tracé



Deuxième tracé

## ■ Belle surface : 90 000 points

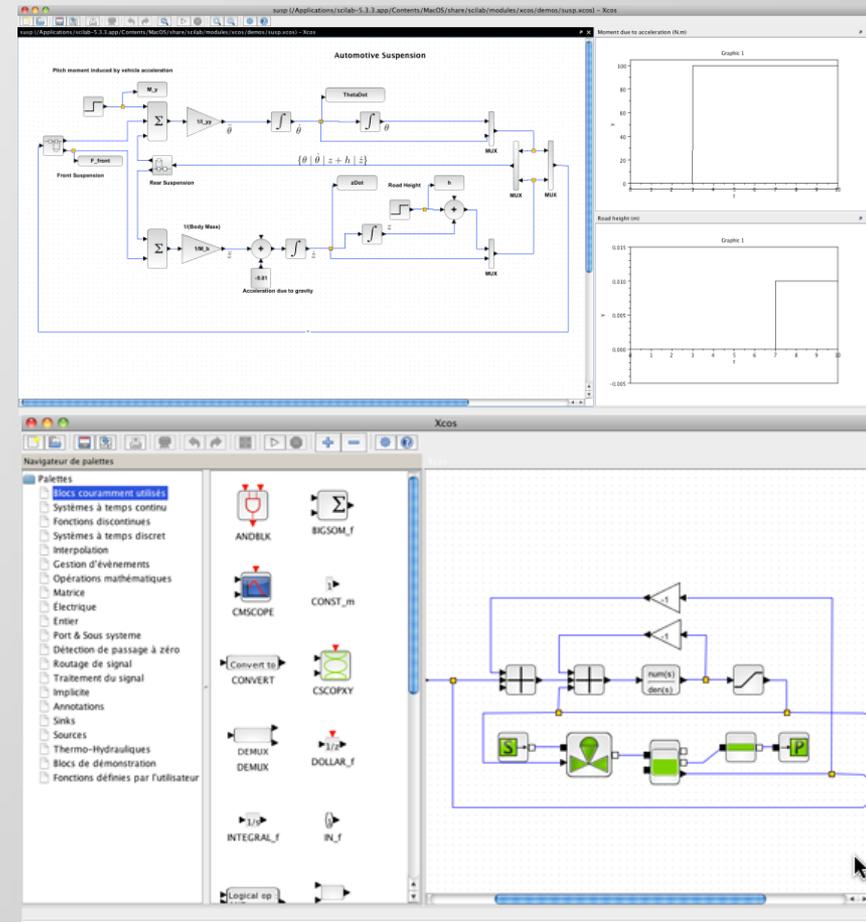
```
function z=f(x,y) // fonction qui définit la surface
    z=exp(exp(-x^2-y^2)*(exp(cos(x^2+y^2)^20)+..
    8*sin(x^2+y^2)^20+2*sin(2*(x^2+y^2))^8));
endfunction
x=linspace(-1.5,1.5,300); y=linspace(-1.5,1.5,300); z=feval(x,y,f);
f=scf(0); f.color_map=rainbowcolormap(32);
surf(x,y,z); // tracé de la surface
e=gce(); e.color_mode=-1;
a=gca(); a.box="off";
a.axes_visible=["off","off","off"];
a.x_label.visible="off";
a.y_label.visible="off";
a.z_label.visible="off";
```



# Xcos : systèmes dynamiques

# Modélisation et simulation de systèmes dynamiques

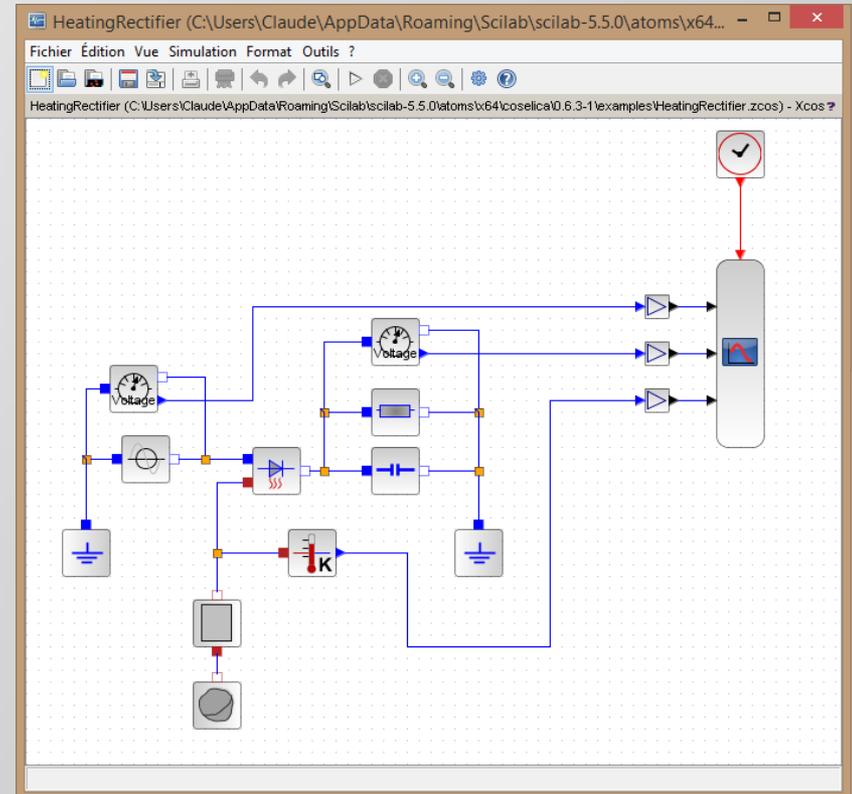
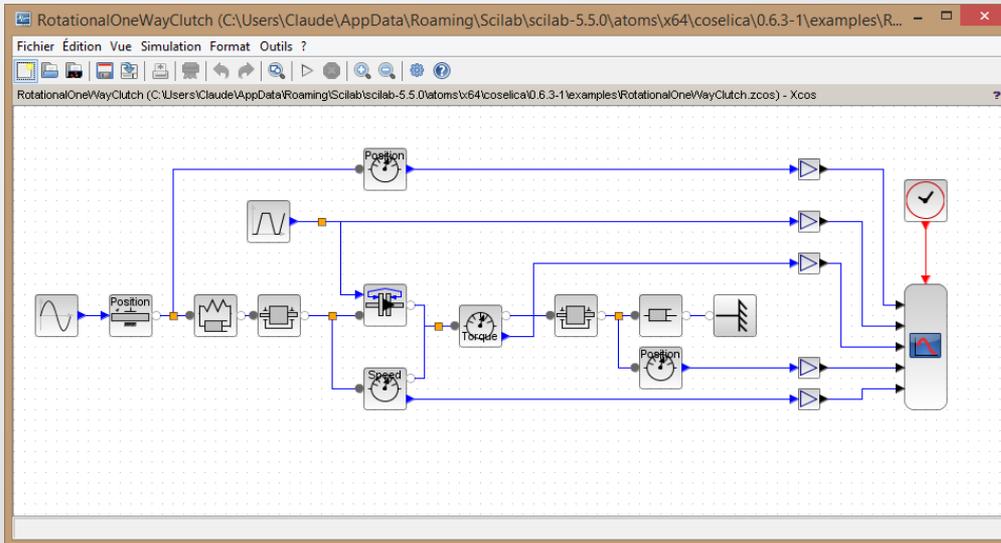
- Une IHM conviviale pour modéliser et simuler les systèmes dynamiques hybrides à l'aide de blocs diagrammes : construction et édition de modèles
- Compilateur Modelica inclus
- Inclus dans la distribution Scilab



# Le module Coselica pour la simulation multi physique

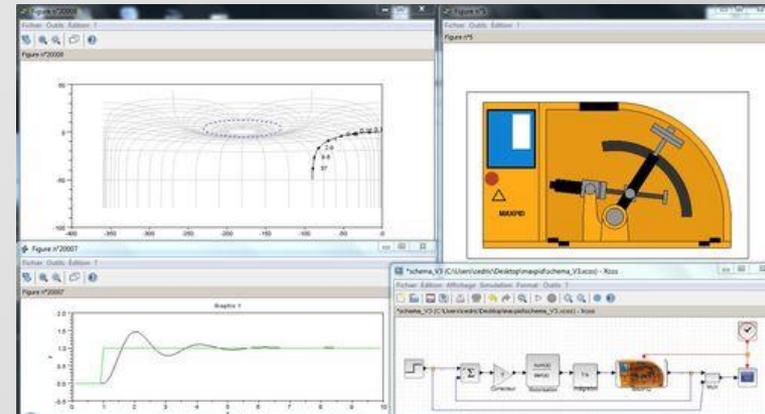
200 blocs acausaux (en langage Modelica) :

- Systèmes électriques analogiques.
- Systèmes mécaniques : 1D (translations, rotations) et 2D planaire.
- Echanges thermiques 0D/1D.



# Pour les sciences de l'ingénieur

- Le module CPGE
- Le module SIMM pour la modélisation multi physique (basé sur Coselica)



Le livret associé

# Scilab = un langage pour programmer

Environ 1 300 fonctions Scilab sont écrites en Scilab

# Scilab pour programmer

- Editeur convivial, langage mathématique proche du langage naturel :

```
function c=Dicho(f,a,b,epsilon)
    c=(a+b)/2;
    while (b-a)/2 >= epsilon
        c=(a+b)/2;
        if f(a)*f(c)<0 then
            b=c;
        else
            a=c;
        end
    end
endfunction
```

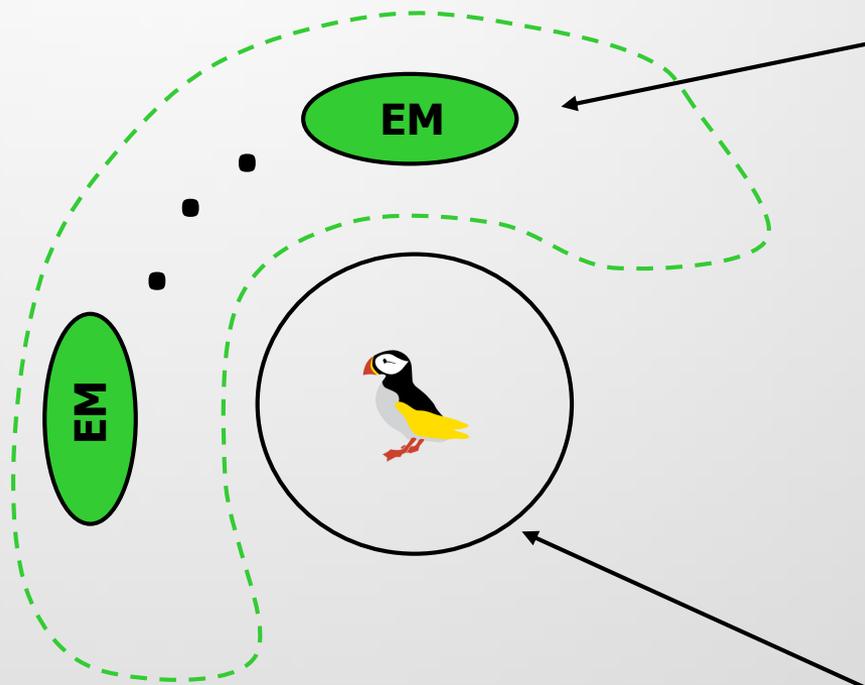


syntaxe vectorielle Scilab

Méthode du pivot de Gauss avec  
recherche de pivots partiels :

```
function x=Gauss(A,b,eps)
    n=size(b,"*"); x=b;
    for k=1:n-1
        // cas où le terme diagonal est près de 0
        // recherche d'un élément non nul dans la colonne
        if abs(A(k,k))<eps then
            kk=find(abs(A(k:n,k))>eps);
            if kk==[] then
                disp("Matrice non inversible");
                return;
            end
            // échange des lignes k et kk dans A et dans b
            kk=kk(1);
            lignek=A(k,:); A(k,:)=A(kk,:); A(kk,:)=lignek;
            lignek=b(k); b(k)=b(kk); b(kk)=lignek;
        end
        // algorithme de Gauss
        for l=k+1:n
            p=A(l,k)/A(k,k);
            for m=k:n
                A(l,m)=A(l,m)-A(k,m)*p;
            end
            x(l)=x(l)-x(k)*p;
        end
    end
    if abs(A(n,n))<eps then
        disp("Matrice non inversible");
        return;
    end
    // méthode de remontée
    x(n)=x(n)/A(n,n);
    for i=n-1:-1:1
        s=0;
        for j=i+1:n
            s=s+A(i,j)*x(j);
        end
        x(i)=(x(i)-s)/A(i,i);
    end
endfunction
```

**Etendre Scilab = ajouter un  
module externe**



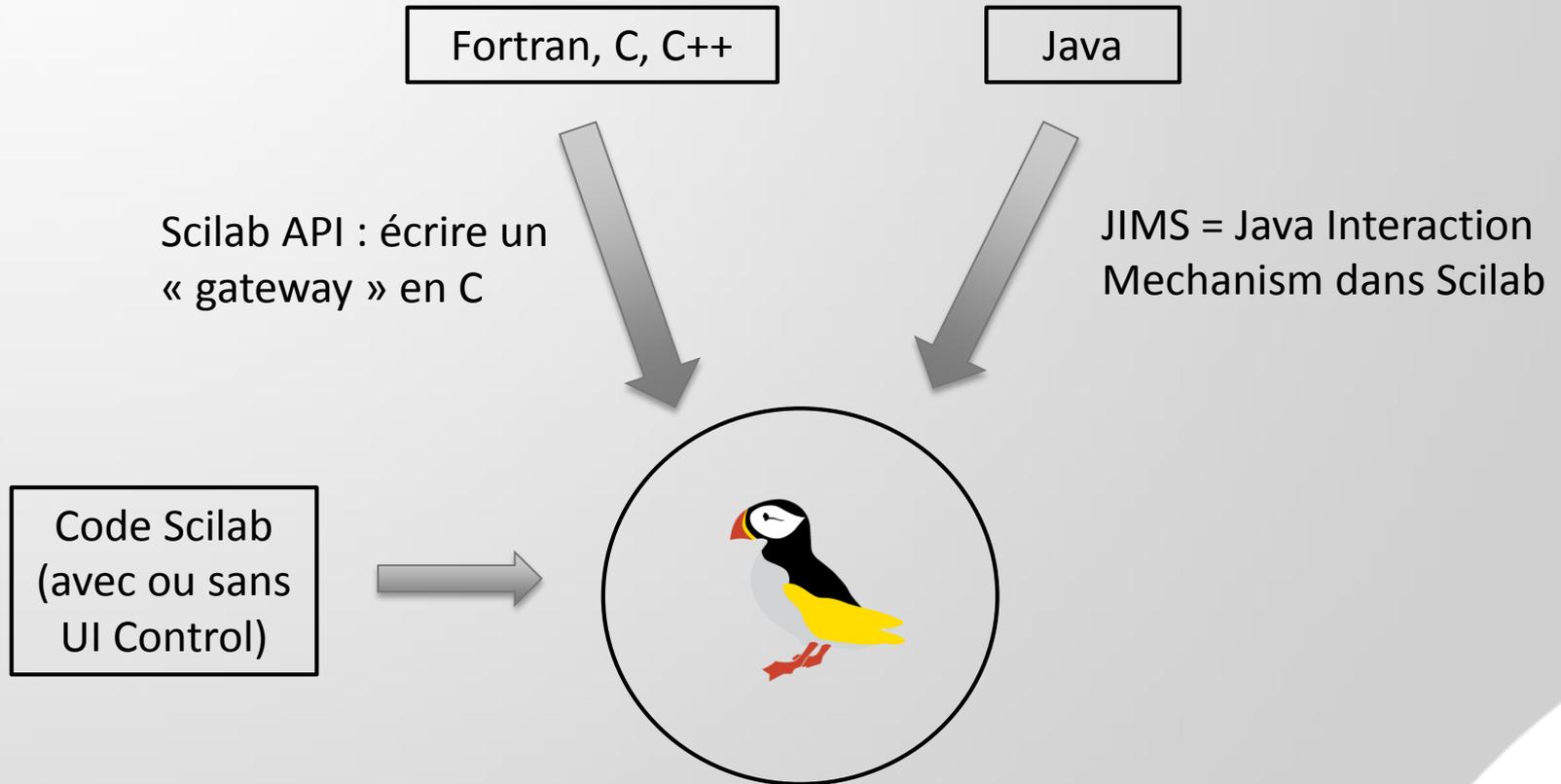
## Modules externes :

- Open Source ou pas
- Libre ou commercial

## Scilab :

- Libre et Open Source

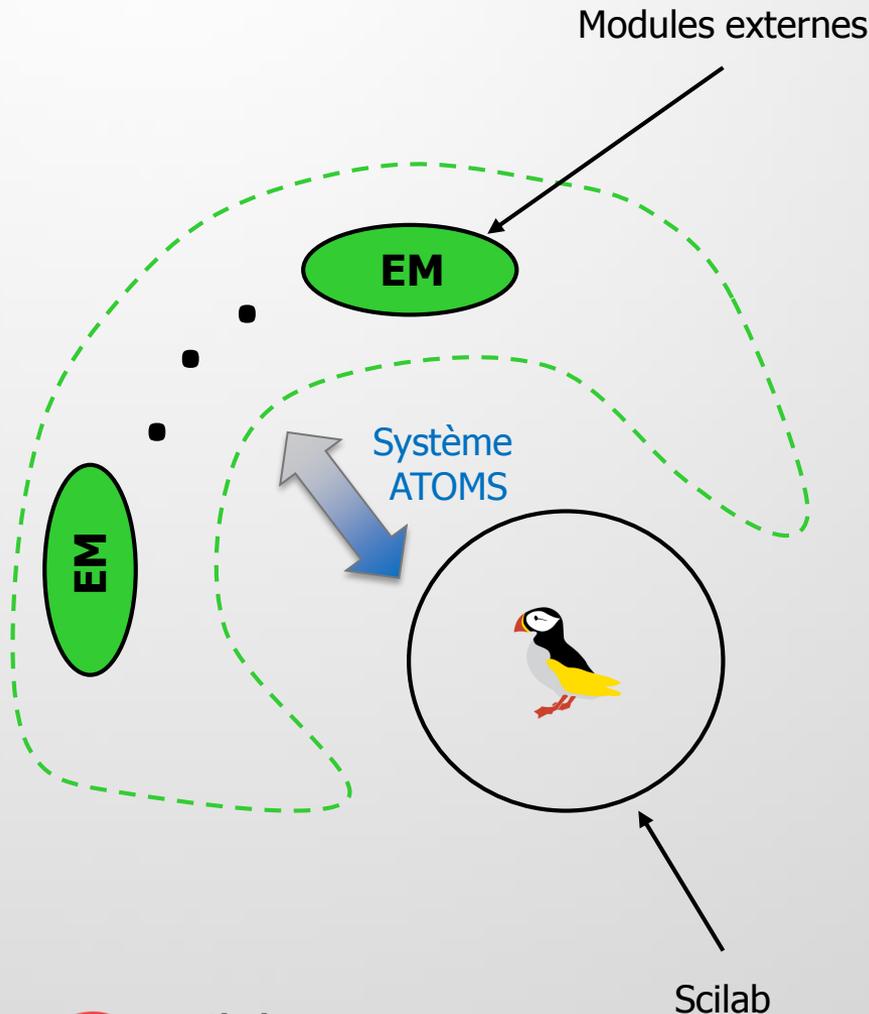
# Créer une application complète en utilisant le bon langage



# Utiliser le système ATOMS



Installation interactive dans Scilab



Avantages des modules ATOMS :

- Indépendant des sorties des versions Scilab : mises à jour faciles
- Fonctionne sur toutes les architectures (code Fortran, C ou C++ compilé par Scilab Enterprises sous Windows, Mac OS X, Linux)
- Gestion de la dépendance des modules ATOMS

# Scilab dans l'industrie

## L'exemple du CNES

## **ATOMS module for space mechanics and flight dynamics**

- Freely available and Open Source
- Used by CNES and ESA for mission analysis
- Library of Scilab code: functions easily re-used for making new programs

## CelestLab topics

Topics	Contents
Coordinates and Frames	<ul style="list-style-type: none"><li>- Change of coordinates</li><li>- Dates manipulation</li><li>- Change of reference frames</li><li>- Orbital element transformations</li><li>- Rotations and quaternions</li></ul>
Geometry and Events	<ul style="list-style-type: none"><li>- Orbital events computation</li><li>- Orbital geometry</li></ul>
Interplanetary	<ul style="list-style-type: none"><li>- Interplanetary transfer</li><li>- Three body analysis</li></ul>
Models	Earth motion, density models
Orbit properties	<ul style="list-style-type: none"><li>- Keplerian formulas</li><li>- Orbit characteristics (sun synchronism, repeat orbits, frozen orbits)</li></ul>
Relative motion	Chlohessy-Wiltshire formalism
Trajectory and manoeuvres	<ul style="list-style-type: none"><li>- Orbit propagation (analytical)</li><li>- Manoeuvre computation</li><li>- Dispersion analysis</li></ul>
Utilities	<ul style="list-style-type: none"><li>- Various support functions including graphics</li></ul>

## CelestLab and mission analysis practices

- **Coding Scilab scripts using CelestLab is easy. This encourages people to develop their own scripts.**
- **CelestLab is developed by people in charge of mission analysis. It is a shared product.**
- **When an analysis is completed, there is an assessment on whether a part can be incorporated in CelestLab.**
- **CelestLab demos are a efficient solution for answering recurrent questions and can easily modified if needed.**
- **CelestLab is well documented and is more and more used as a source of information on a laptop.**

## Examples of computations made by CelestLab: visibility of satellite constellations from a geographical zone

