



**HAL**  
open science

# Les outils Open Source de modélisation et de simulation du génie électrique

Mohammed Tamali

► **To cite this version:**

Mohammed Tamali. Les outils Open Source de modélisation et de simulation du génie électrique. Master. Les outils Open Source de modélisation et de simulation du génie électrique, Bechar, Algérie, Algérie. 2017, pp.34. cel-01485491

**HAL Id: cel-01485491**

**<https://hal.science/cel-01485491>**

Submitted on 8 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0  
International License

Université de Béchar

Laboratoire des Études Énergétiques en Zones Arides

Équipe Modélisation & Simulation des Systèmes



Using UBCluster

Open Source

Microsoft (R)



VS



MATLAB

(ENERGARID Lab./SimulIA Team)

Projet National de Recherche SITI

Centrale éolienne de  
Kabertan, Adrar

(10 MW) ...



ICESD'2017

2016. 3. 8 16:57





# Présentation



Bechar University was born in 1986 as the National Institutes of Higher Education (INES), in 1992 it becomes University Center and on January 1<sup>st</sup> 2007, it was officially declared as a University. Since then, many research teams have seen the day. In 2011, The Laboratory for Energy Systems Applied Studies to Arid Zones was run by a group of young and well motivated researchers. ENERGARID Lab. Is actually composed by seven research teams, the main task was and still, solving local and real problems altering arid zones, Simulia Team is one of the Lab. components. The workload of Simulia concerns **managing** models and simulating real cases of systems in arid areas under smart environment and using adopted computing means.



## Major research areas:

- Energy & Environment (Modeling & Simulation)
- Application & Usage of heat transfer process in arid zones
- Optimization & Economy of energy usage.
- Development of 2D/3D Mapping Tools of resources in arid zones.



Simulia Team task in the short term is to develop the computer code for modeling and simulation of energy process which can be accessed via Web Interface.

Website of the laboratory team: [www.univ-bechar.dz/energarid/simulia](http://www.univ-bechar.dz/energarid/simulia)



# Plan



**Généralités & Positionnement**  
**Définitions fondamentales**  
**Classification des outils logiciels**  
**Études de cas**  
**Modèles de solution**  
**Conclusions**



**OPEN SOURCE**

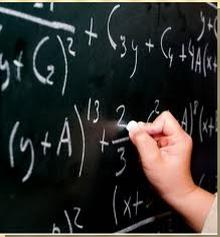
# Généralités & Positionnement

Les méthodologies utilisées par les humains, en rapport avec les tentatives de compréhension des questions fondamentales qui nous entourent et que nous manipulons et prenons comme sujets dans nos études, mettent la lumière sur la complexité de ces mêmes systèmes .

Le niveau de complexité des ces systèmes est élevé, à un niveau où toutes les tentatives de lancement de procédures expérimentales laissent et obligent à considérer des erreurs et encore plus, les effets tangents. Selon la théorie de l'évaluation des performances, l'exigence 'comprendre' le système n'a de réponse que si :

- Nous connaissant largement notre système que les exigences des études ultérieures sont satisfaites.
- Nous avons des références, avec quoi comparer,
- Nous avons un historique susceptible d'être retracé,
- Il y a une possibilité pour faire de l'expérimentation.

Les trois premières conditions satisfont à elles-mêmes. Si telle est le cas, elles nous clarifient l'image. Le quatrième critère exige que l'expérience se fasse effectivement pour que toutes les questions, relatives au problème posé, soient élucidées. Ces variables sont les facteurs du systèmes et peuvent évoluer selon des modalités proposées.



# Généralités & Positionnement

Les outils utilisés par les humains à l'issue d'une production technologique laisse un arrière goût fugitif mais répété alors que la recherche scientifique fondamentale le fait d'une manière pressante, à l'action, mais trop espacé pour prétendre une quelconque réaction.

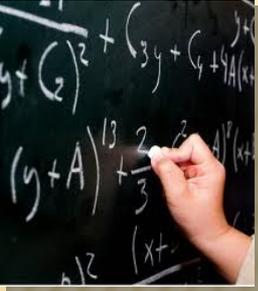
Le niveau de complexité de ces technologies est très fort à un niveau où toute intégration sociale des produits résultants ne peut se faire sans dégâts ou effets secondaires. En contre partie, la majorité des produits technologiques sont juste, la solution immédiate de bon nombre de questions urgentes de la vie courante, en zone urbaine ou rurale. Une sensibilisation pour le compte de l'environnement et sa préservation reste le souci majeur des observateurs et des chercheurs.

L'avènement de produits, de grande consommation, comme l'informatique, l'internet et les télécommunications, à chacun ses outils, a révolutionné les méthodologies de travail des équipes et des laboratoires de recherches. Les outils et suites logiciels sont d'un grand secours dans les travaux de modélisation et de simulation de processus.

L'informatique et les moyens de conception sont les vecteurs qui ont provoqués une amélioration nette à la vitesse de mise en œuvre de moyens exploitables.

Les logiciels, à travers leurs codes, ont joué un grand rôle dans le cours du développement de beaucoup de procédés de Modélisation et ont facilité la compréhension de beaucoup phénomènes de Simulation.

# Généralités & Positionnement



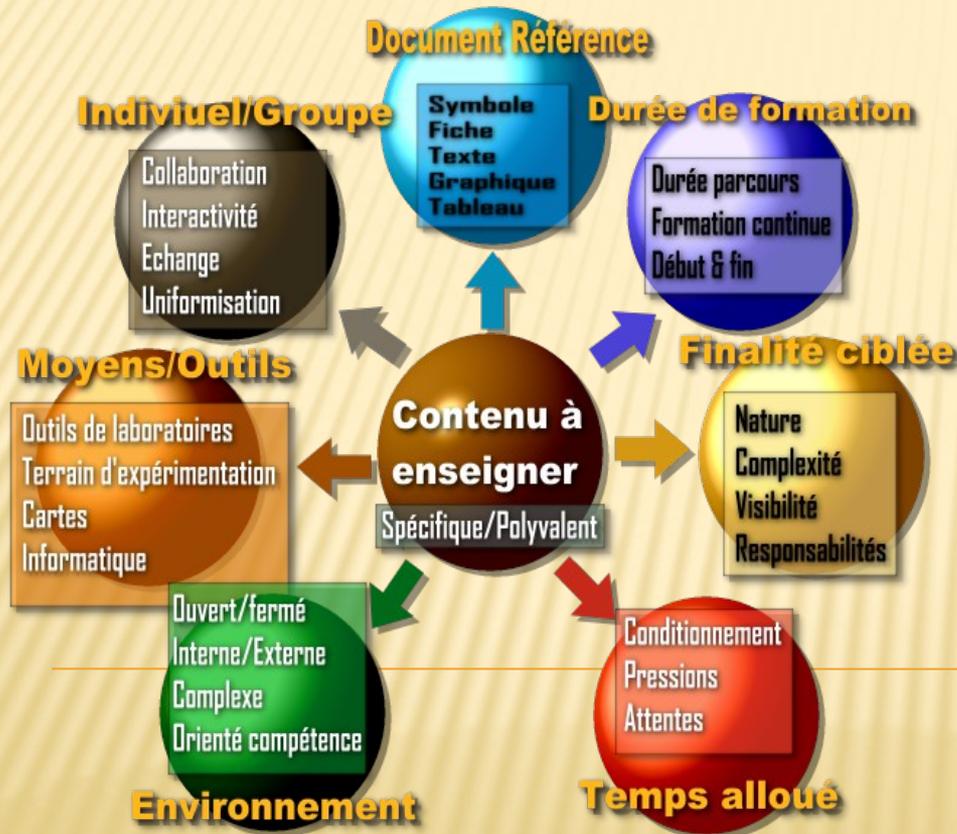
**Les méthodes éducatives valables au moment de la prise en charge du système universitaire sont nettement soupçonnées. Elle sont, actuellement à revoir, pour une éventuelle qualification des fondements de manœuvres, des travaux à entreprendre et des buts futurs à atteindre.**

**La pédagogie est les manières et les méthodes d'enseignement propres à une discipline, à une matière, à un ordre d'enseignement, à un établissement d'enseignement ou à une philosophie de l'éducation.**

**D'après LAROUSSE, La didactique se dit de ce qui vise à l'explicitation méthodique d'un art, d'une science, de quelqu'un qui poursuit ce but dans ses propos, son attitude : Il est très didactique dans sa manière d'exposer une question.**

**La didactique est une réflexion sur la transmission des savoirs, alors que la pédagogie est orientée vers les pratiques d'élèves en classe.**

# Définitions fondamentales



*Il est donc aussi question de méthodes:*

- *expositives : forme d'un exposé,*
- *démonstratives : exercices d'application,*
- *interrogatives : questions - réponses*
- *actives : situations où l'action et l'initiative de l'apprenant prédomineront.*

*Les choix techniques opérés par l'enseignant en fonction des éléments de connaissance qu'il souhaite inculquer relève du champ didactique. Ainsi, le "jeu" mené entre l'enseignant, l'élève et le savoir, semble pouvoir composer un **SYSTÈME**.*

## Le système didactique.

# Définitions fondamentales

## Modélisation

Pour la modélisation, le **système** étudié est appelé **système primaire**. Son modèle (équivalent) est une représentation de la réalité et il est dit **système secondaire**. Par définition un modèle est une représentation simplifiée de la réalité. Le but majeur de cette opération c'est de **DÉCRIRE, PRÉDIRE, EXPLIQUER & RÉAGIR**.

## Modélisation graphique

C'est une stratégie modélisatrice des système utilisant le graphique comme langage et outil de travail.

## Les suites logiciels de simulation

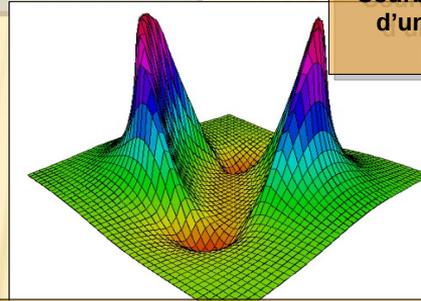
Composition faites des deux entités, Les **SYSTEMES d'INFORMATION (Collections de données et d'informations)** et les **MATHEMATIQUES (Collections d'algorithmes de calcul)**. C'est le moyen utilisé pour allier **efficacité** et **utilisabilité** des études et des coûts équivalents à une **réelle expérimentation**.

**Les objectifs sont généralement pour des stratégies d'aide à la décision.**

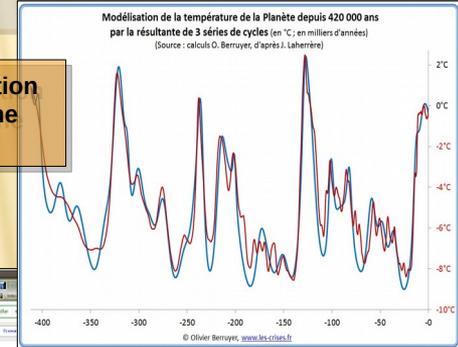
## Finalité de la modélisation

**Technique** : fournir des spécifications claires à reproduire, puis à exploiter

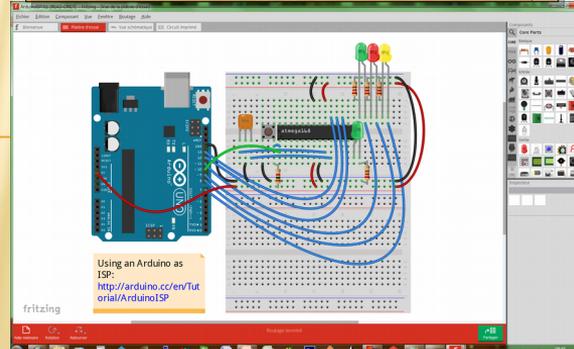
**Intellectuelle** : fournir au métier, une utilité dans les structures sociétales.



Courbes d'évolution d'un phénomène physique



Exemple Arduino ISP sous Fritzing (GPL)



OpenStreetMap  
Visualisation de la carte de Bèchar (Streets)  
Focalisation sur l'Université de Bèchar

Modèle SIG d'une urbanisation  
Sous QGIS (GPL)

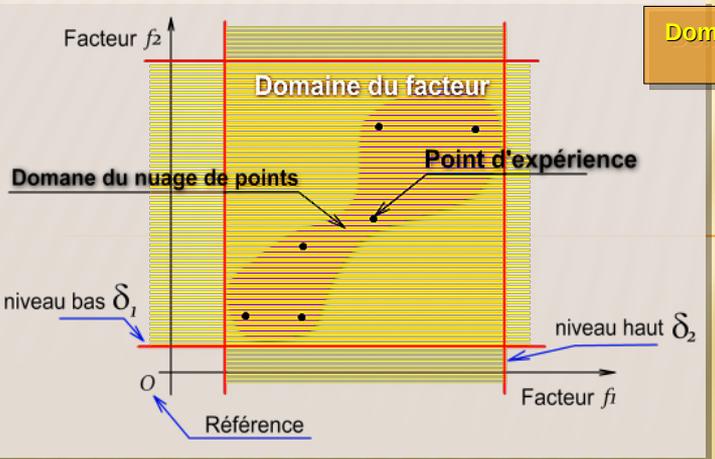
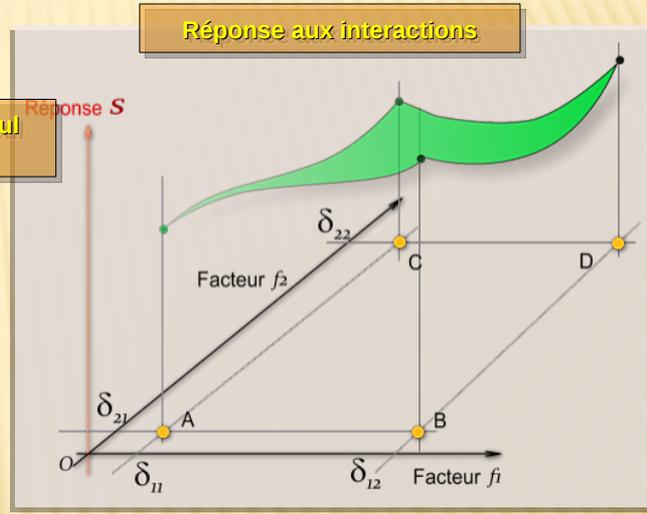
# Définitions fondamentales



Situation d'un seul facteur

**Les Plan d'expériences :** Organisation permettant la conduite des expériences, **L'expérience** peut être menée selon beaucoup de manière, le phénomène étudié est au centre des préoccupations. Le cas le plus simple est celui où la variation d'un seul paramètre est considéré.

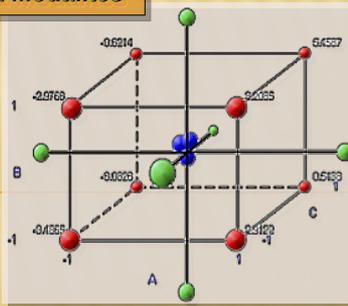
Le système dépend, fondamentalement, de ses composantes intrinsèques qui varient dans le temps et l'espace physiques. Le cas général à considérer est; quand plus d'une composante rentre en interaction.



Domaine de variabilité de deux facteurs

**La réponse du système** étudié dépend essentiellement du type de considérations des rattachements à remarquer dans les interactions afin qu'une réponse du système ne soit ressentie. L'observation des interactions  **doubles**  est la plus simple à exécuter. Les systèmes physiques sont, par défaut, trop complexes.

Facteurs & Modalités



# Définitions fondamentales

Les suites logiciels de **modélisation** et de **simulation** des systèmes et des procédés ne peuvent en aucun cas représenter le **SUBSTITUANT** de l'**expérimentation** et du terrain (réalité des choses).

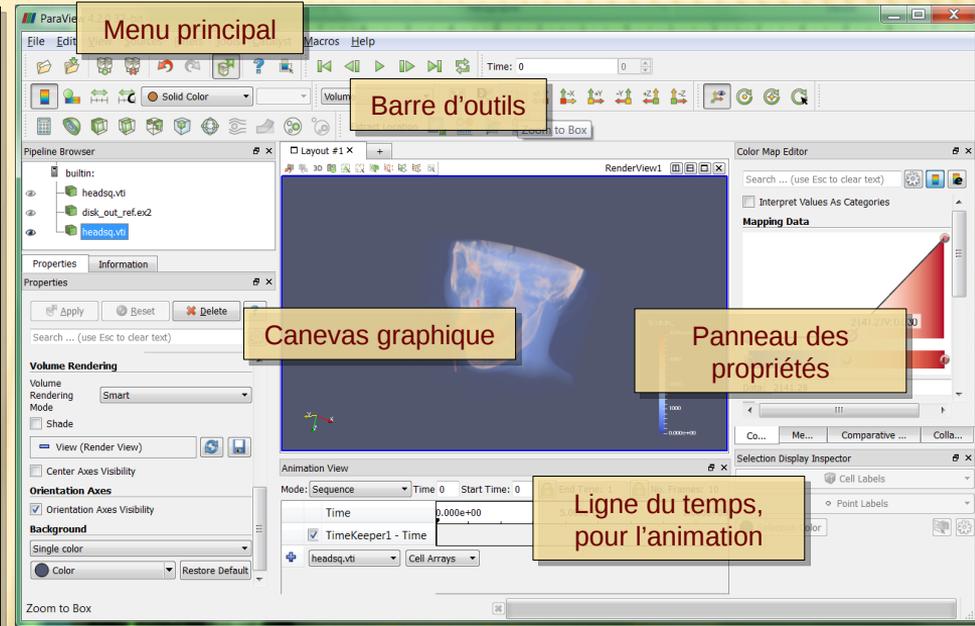
C'est des outils d'aide à la **compréhension**, à la **réduction des coûts** impliqués par l'exercice expérimentale et encore pour répondre à une exigence de la **décision**.

Pour que ces outils atteignent un niveau de maturité de ce fait, ils peuvent être acceptés avec une **CONFIANCE mathématique** appréciable et ils sont dotés d'une base de données (Pour des cas, plus un moteur d'inférence).

Ils sont conçus autour d'un noyau (**PROCESSING CORE**) qui est surmonté par une **Bibliothèque de modules** (unités/Plug-Ins) de bases et d'extension.

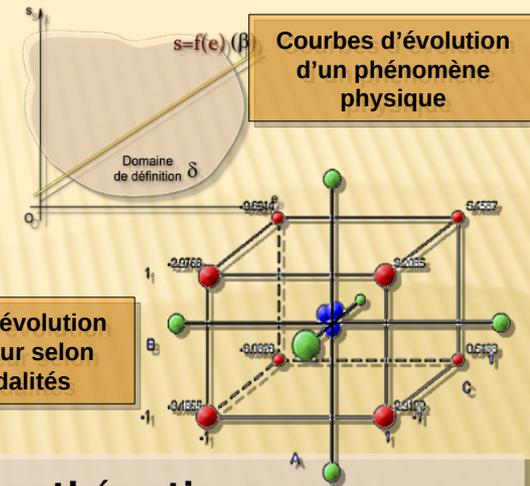
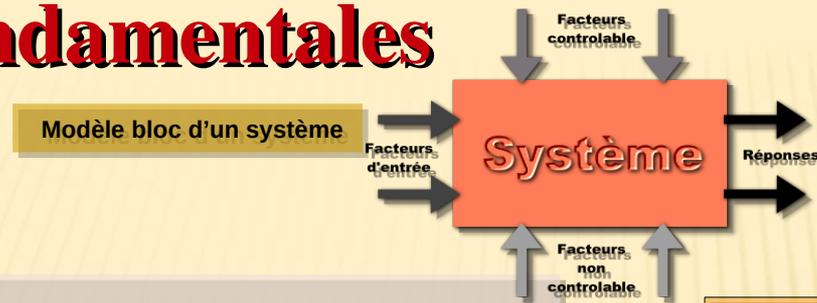
Le plus important de tous ces modules est l'**INTERFACE GRAPHIQUE**. La complexité de cette dernière est de taille, puisque toutes les tâches de traitement se font directement à travers elle. En plus de la zone de modélisation du processus système à modéliser, des menus, des barres d'outils et des panneaux forment le moyen via lequel l'utilisateur interagit en toute sécurité avec l'application. D'autres éléments cachés par défaut et ne sont affichés que si une requête les concernant a été émise.

Logiciel de visualisation graphique  
**Paraview (GPL)**



Le format des données peut être spécifique (\*.dxf pour AutoCAD) ou commun pour beaucoup de logiciels (\*.xml, Standard OpenDocument). Les rendus graphiques ne sont qu'une image de la réalité données.

# Définitions fondamentales



## Expérience

n. f. Expérimenter, acte de procéder à des **essais effectifs** dans un lieu destiné à cette tâche (laboratoire) et pour un but purement **scientifique**. Selon la complexité de la composition du système, on est appelé à faire **beaucoup de tests** (m expériences) pour un même **scénario** et ceci, selon la **dépendance** (ou non) entre les facteurs régissant le système, sujet de l'expérience.

Cette procédure est d'autant plus combinatoire que le nombre de cas à vérifier est beaucoup plus grand ( $m \rightarrow \infty$ ). Les **plans d'expériences** permettent d'organiser au mieux les essais qui accompagnent une recherche scientifique ou des études industrielles. Ils sont applicables à de nombreuses disciplines et à toutes les industries à partir du moment où l'on recherche le lien qui existe entre une grandeur d'intérêt,  $y$  et des variables,  $x_i$ . Il faut penser aux plans d'expériences si l'on s'intéresse à une fonction du type  $y = f(x_i)$ . Avec les plans d'expériences on obtient le maximum de renseignements avec le minimum d'expériences.

Courbes d'évolution d'un facteur selon des modalités

## modélisation mathématique :

Procédures mathématique permettant de choisir un équivalent (Modèle) à un système donné.

### Facteur, Modalité & Plan

**Facteurs** : les composantes d'un système sont les éléments dont il dépend.

**Modalités (Niveaux)** : Niveaux appréciable que peut prendre un facteur relativement à une situation précise du système.

**Plan** : Une composition expérimentale visant à faciliter la tâche de l'expérimentateur en lui présentant une méthodologie finie pour entreprendre ses études et essais.

**Espace expérimental** : Le domaine de variabilité des réponse à toutes les valeurs du facteur étudié selon des modalités.

# Définitions fondamentales

Les suites logiciels de **modélisation**, de **simulation** des systèmes et des procédés sont de deux catégories vis-à-vis des droits et devoir d'utilisation. Pour ces mêmes suites, il y a le développeur du produit, d'un côté, et l'utilisateur final de l'autre. A chacun ses droits mais limitées selon des clauses et les recommandations, sur la base de quoi, le deux partenaires se sont mis d'accord. Ces termes définissent ici ce qui est communément appelé **LICENCE**. Cette dernière décrit clairement les partenaires, le produits et les limites de chacun vis-à-vis de la loi.

Il existe deux types de licence:

- **Propriétaire,**
- **Libre d'utilisation dite encore GPL**

Dans le premier type, le développeur se doit de porter assistance à ses clients et de garantir la continuité du service. Au même moment que les clients ne doivent, et aucun cas, porter préjudice à leur mode ou manière d'utilisation du produit, ne pas l'utiliser qu'avec le consentement du développeur, ne pas l'altérer par un quelconque moyen. La distribution du produit est, de ce fait totalement interdite.

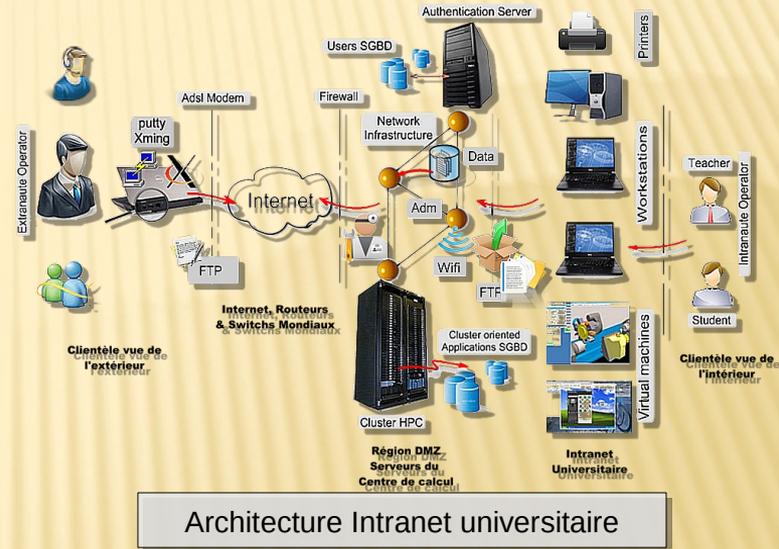
Le client est, par ailleurs, bien en état de connaissance de tout dommage qui pourrait éventuellement lui être causé par son utilisation du produit et pour lequel la partie développeur, décline toute responsabilité.

Pour le deuxième type, le développeur se **partage** les droits/devoirs de **redevelopper** le produit avec son client.

Les suite **GPL** (**General Public Licence**) sont généralement des contributions qui évoluent pour gagner du terrain. Le **paquetage** est distribuer avec les fichiers binaires, le code sources, tous les fichiers bibliothèques et annexes en plus de la documentation.



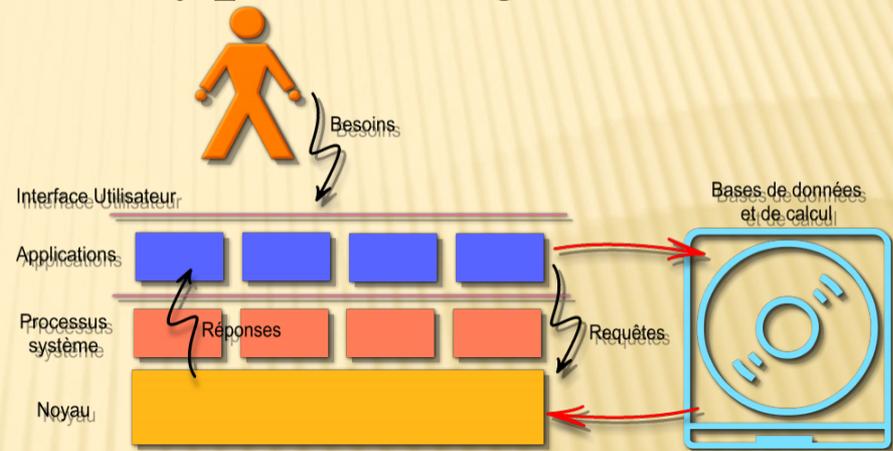
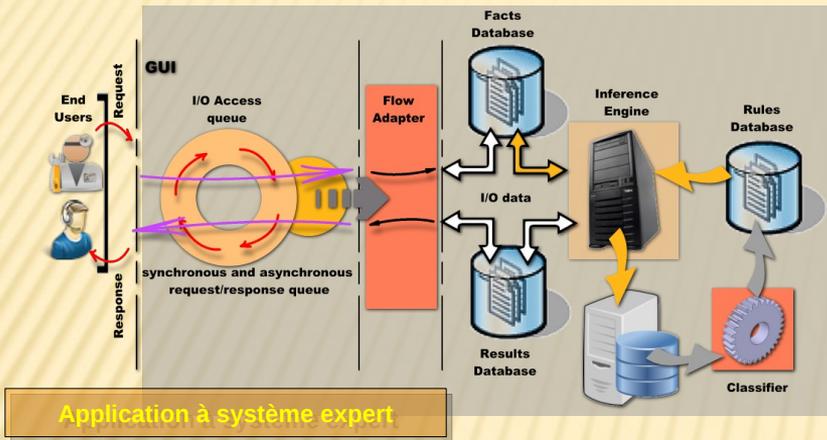
OPEN SOURCE



Architecture Intranet universitaire

Ces suites sont déployées et présentent leur service selon deux mode, Localisé (**Stand Alone/Desktop**) ou bien réparties sur le Web (**Web Based**). Elle peuvent avoir besoin de données **Localisé** ou **Distribué**.

# Définitions fondamentales (Type de logiciels)



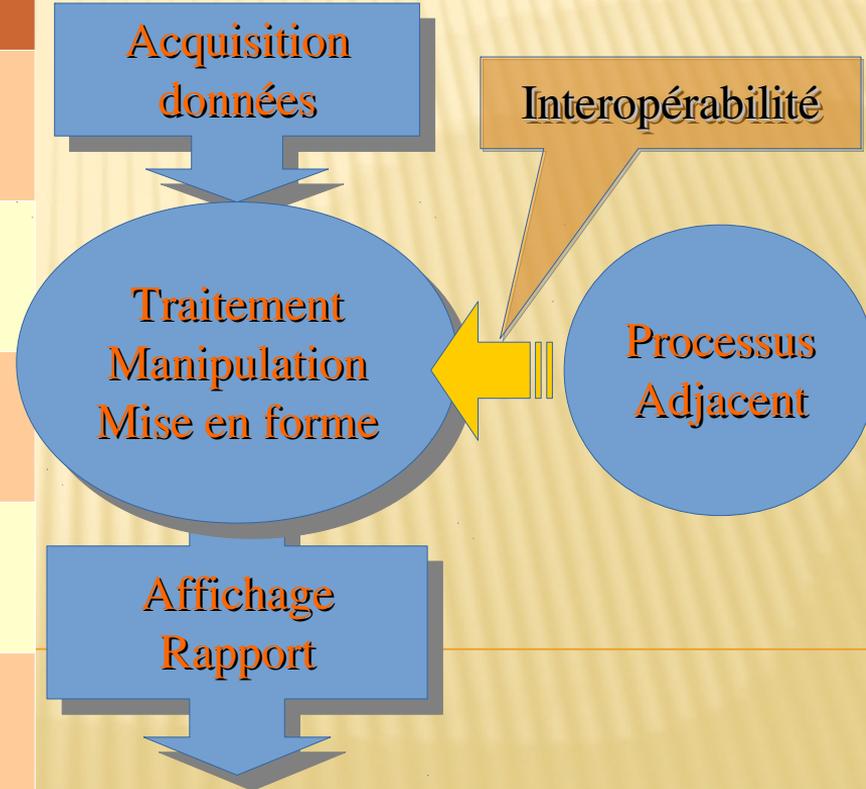
L'architecture de base de tout logiciel utilisé dans le domaine de la technologie pour des fins de modélisation et de simulation des processus, repose sur une structure à trois étages.

Le **noyau**, combine les procédures de modélisation (définitions) et de simulation (traitements) des données se reportant à un système à traiter. Les niveau deux, regroupe, tous les **processus** logiciels du **système** de communication entre le noyau et les niveaux applicatifs.

Le niveau **Application** représente le code interface utilisateur **UI**, moyen utilisé pour inter-changer des informations entre l'utilisateur chercheur et le logiciel en question. Une mémoire, sous forme de base de données est utilisée pour stocker les informations temporaires et celles finales avant qu'elles ne soient transformées par des processus de calcul ou bien affichées sur l'interface utilisateur.

# Définitions fondamentales (Typologie)

Local	Réseau	Web/Cloud
Avec IU (Graphique)	Avec IU (Graphique)	Avec IU (Graphique)
Avec IU (Texte)	Avec IU (Texte)	Avec IU (Texte)
Sans IU (Terminal)	Sans IU (Terminal)	Sans IU (Terminal)
Orienté Fichier	Orienté Fichier	Orienté Fichier
Orientée SGBD	Orientée SGBD	Orientée SGBD



# Classification des outils logiciels

## (Licence des droits légaux d'utilisation)

Il existe deux classes de licences :

- **Propriétaire,**
- **Libre d'utilisation dite encore GPL**

Les logiciels peuvent être :

- *À interface graphique/Sans*
- *Orienté SGBD/Fichier à format spécifique*
- *Déployé en local/Internet (Cloud, distribué)*
- *Interopérabilité intégrée/Non*
- *Reporting détaillé/limité*
- *Documentation large/limitée*
- *Communauté large/limité ou sans*



**OPEN SOURCE**

# Logiciels GPL du Génie Électrique

Dans le domaine du génie électrique, beaucoup d'outils GPL peuvent être utilisés en mode local, de la manipulation des simples circuits électriques jusqu'aux grandes installations et équipements de l'électrotechnique.



OPEN SOURCE

The image shows two overlapping software windows. The background window is KTechLab, displaying a circuit schematic with various components like resistors, capacitors, and diodes. The foreground window is B2Spice, showing a simulation of a Biquad Filter. It includes a circuit diagram with component values (R7=0.7, C1,C2=0.0159uF=1KHz, 0.0080uF=2KHz) and a graph titled "Biquad\_Small Signal AC\_0\_Graph" showing magnitude and phase response. A blue oval with the text "Mode Local" is overlaid on the B2Spice window.

**Logiciel KTechLab**

**Logiciel SPICE**

**Mode Local**

The image shows the Qucs software interface for a transient simulation. It displays a circuit schematic for a "Peltz oscillator" and a graph showing the "Output" signal over time. The graph shows a periodic waveform oscillating between approximately 9.5 and 10.5. The simulation parameters are visible at the bottom of the graph.

**Logiciel Qucs**

# Logiciels GPL du Génie Électrique

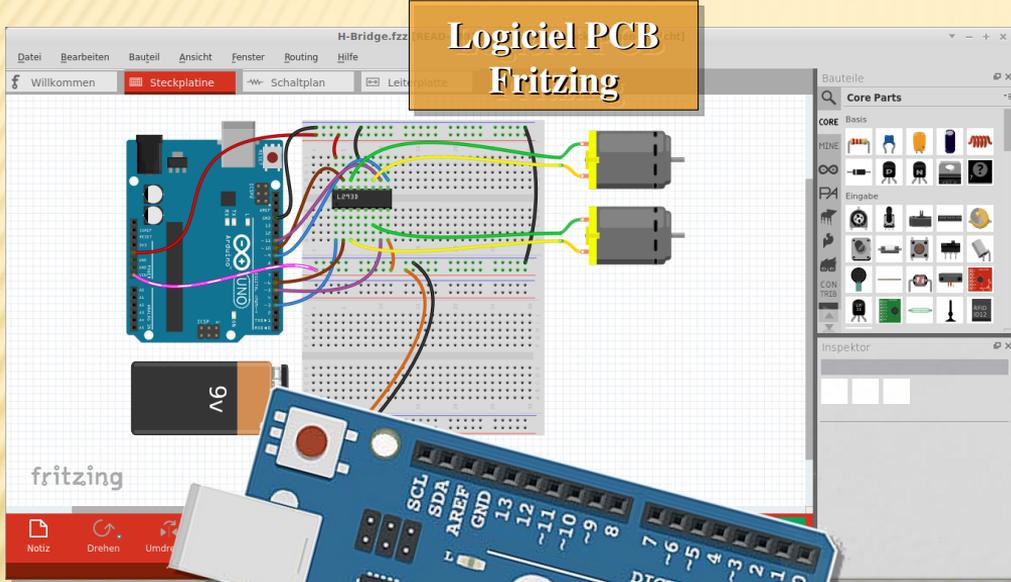
Dans le domaine du génie électrique, beaucoup d'outils GPL peuvent être utilisés sous des environnements Cloud, de la manipulation des simples circuits électriques jusqu'aux grandes installations et équipements de l'électrotechnique.



The screenshot shows the CircuitsCloud website interface in a browser. The browser tabs include 'Boit', 'Sche', 'Tuto', 'Nod', 'Nod', 'Soft', 'G elect', 'Pars', and 'Circi'. The address bar shows 'circuits-cloud.c'. The website header includes the 'CircuitsCloud' logo, navigation links for 'Quick Start', 'Public Circuits', 'Blog', and 'Contact Us', and a user status 'Not logged in. Sign in or create an account.' The main content area features three simulation panels: '7Seg Display' showing a digital circuit with a red '2' on a 7-segment display, 'Interactive Digital Simulation' showing a waveform plot with data points for 'Node 2: 4.17319' and 'Node 3: 4.983838', and another 'Interactive Digital Simulation' showing a circuit diagram with a voltage source and a resistor. A 'Register Now' button is visible at the bottom of the main content area. The browser's taskbar at the bottom shows a file named 'qucs-0.0.19.tar.gz' with a size of 9.8/19.6 Mo and a duration of 4 min rest.

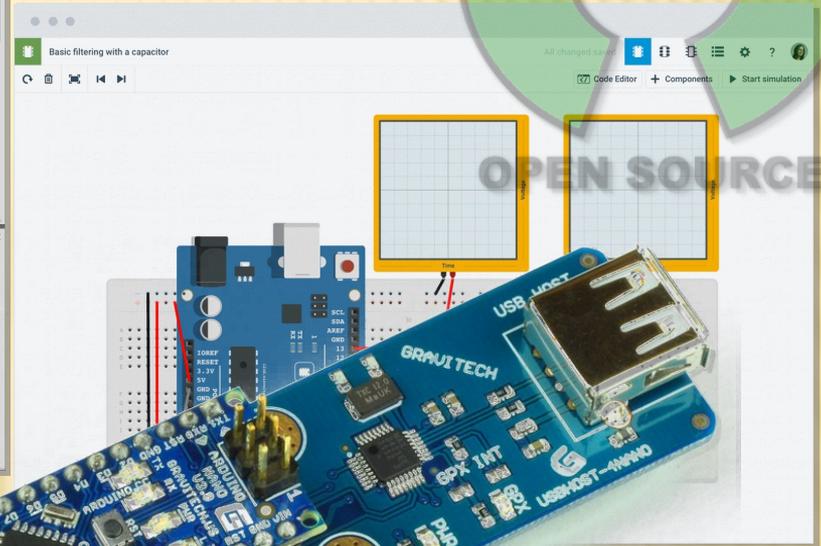
Logiciel  
Circuit Cloud

# Logiciels GPL du Génie Électrique (Simulateur)



Logiciel PCB  
Fritzing

Calcul & Programmation des  
microcontrôleurs 123IO

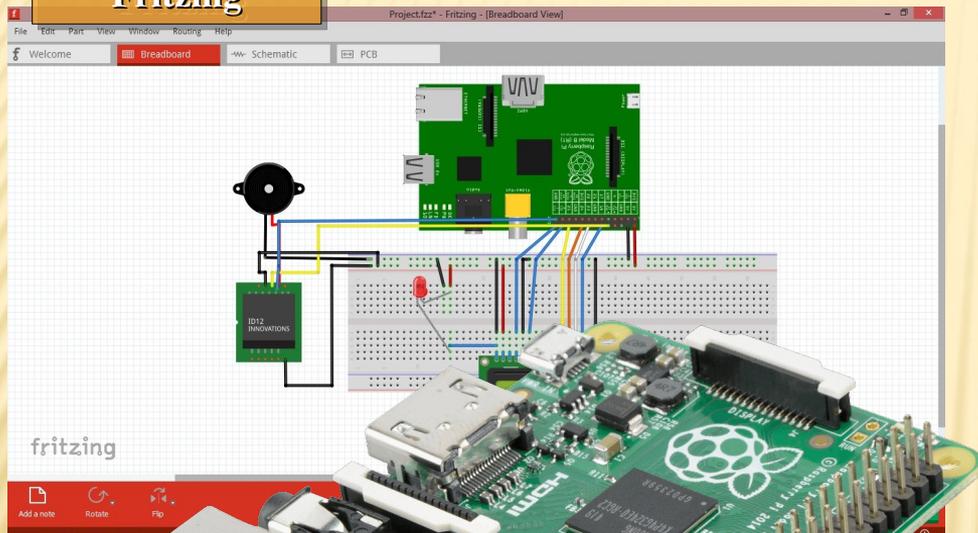


Microcontrôleur Arduino  
UNO

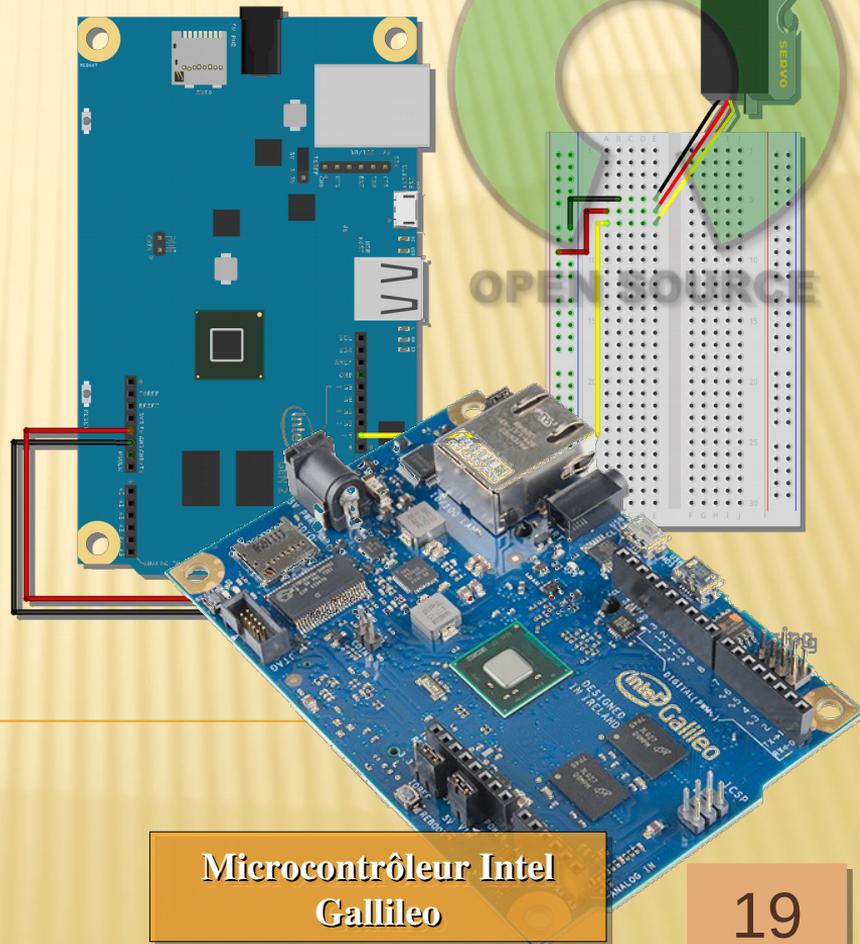
Microcontrôleur Arduino  
Nano

# Logiciels GPL du Génie Électrique (Simulateur)

Logiciel PCB  
Fritzing

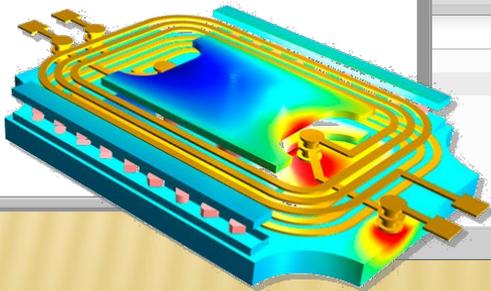
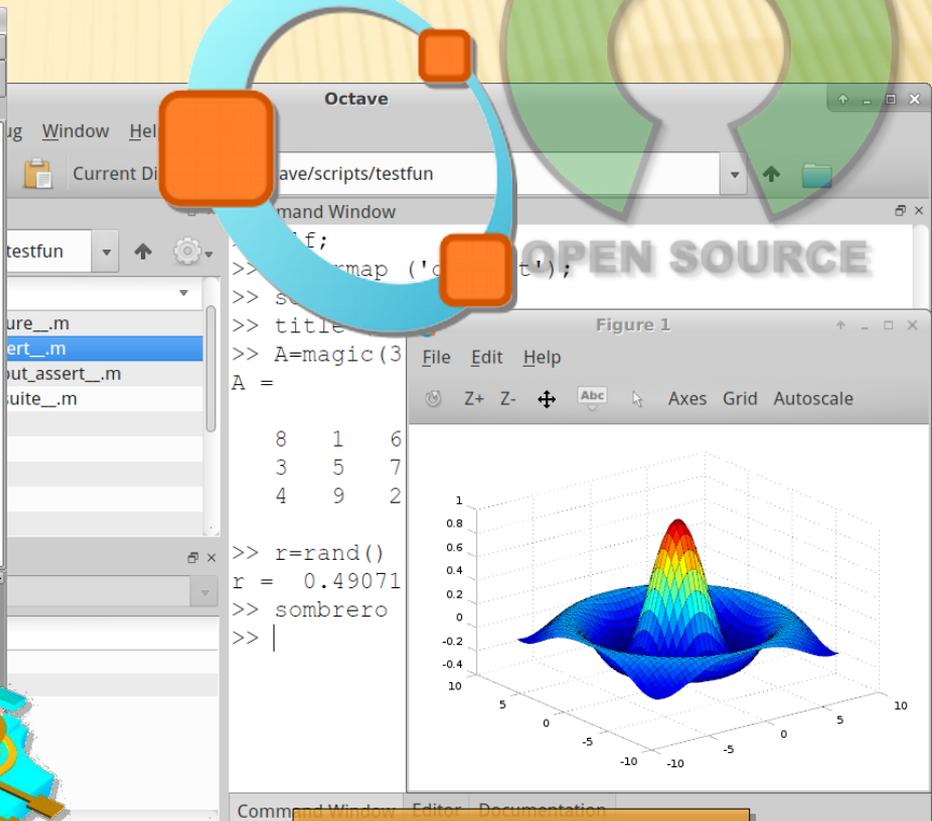
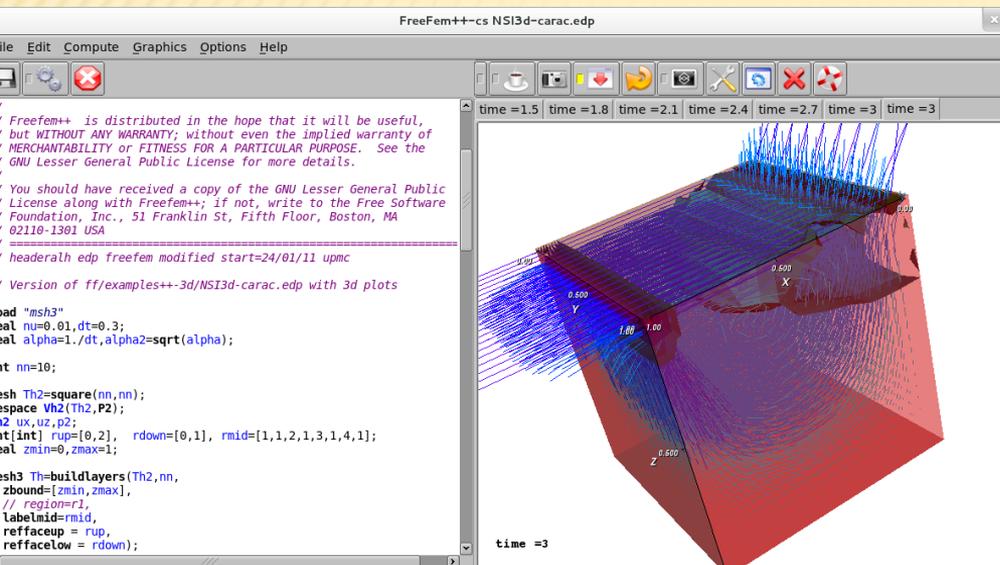


Microcontrôleur  
RaspBerry PI B+



Microcontrôleur Intel  
Galileo

# Logiciels GPL du Génie Électrique (Simulateur)



Logiciel FreeFEM++

Logiciel Octave

# Logiciels GPL du Génie Électrique (Simulateur)

InterPSS Editor

File Edit Run Annotate Report View Format Shape Select Graph Tools Window Help

IEEE14.ieee sample\_case.psse IEEE-14Bus.ipss readme.txt \*DStab-5Bus.ipss

Bus-4 Bus-2 Bus-1

Graphic Editing ▶  
Machine State ▶  
Exciter State ▶  
Governor State ▶  
Stabilizer State ▶

Plot Machine State - Bus Voltage Angle  
Plot Machine State - Machine Pm  
Plot Machine State - Machine Q  
Plot Machine State - Machine Speed  
Plot Machine State - Machine Angle  
Plot Machine State - Machine Pe  
Plot Machine State - Bus Voltage Mag  
Plot Machine State - Machine Eq1  
Machine State Table Output

Logiciel InterPSS Editor

InterPSS 2.2.1 Test

File Edit View Insert Format Data Tools Help

InterPSS Last edit was seconds ago

Input Data File...  
Clear Log Sheet  
User Setting...  
Run Acif  
Run Dcif  
Line Outage Distribution Factor (LODF) ...  
Generation Shifting Factor (GSF) ...  
Contingency Analysis (CA) ...  
Contact Us...  
Version...

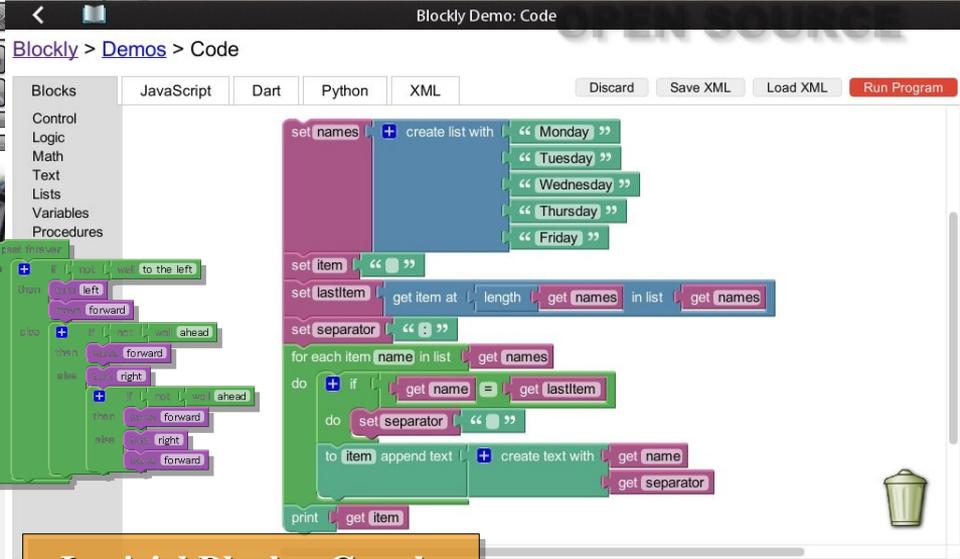
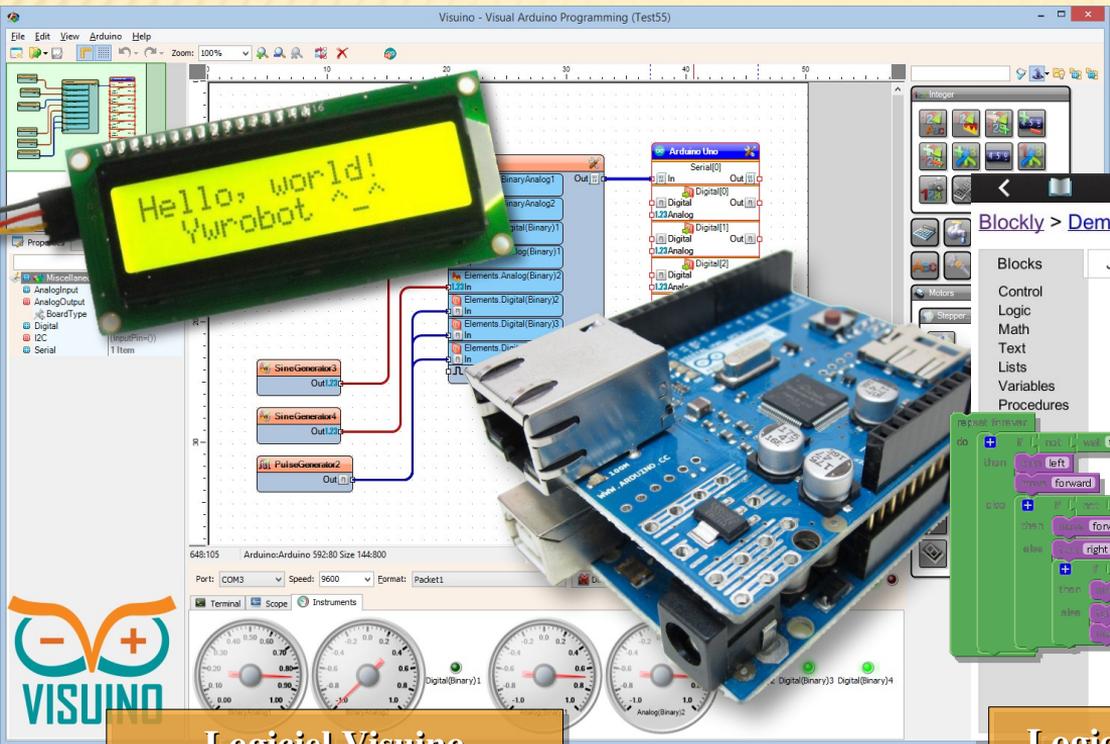
1 Thu Aug 29 2013 17:06:21 GMT-0700 (MST): Copyright ©  
2 Thu Aug 29 2013 17:06:50 GMT-0700 (MST): Loading case: I  
3 Thu Aug 29 2013 17:06:51 GMT-0700 (MST): Format is: IEEE  
4 Thu Aug 29 2013 17:06:51 GMT-0700 (MST): Application is r  
5 Thu Aug 29 2013 17:06:51 GMT-0700 (MST): Uploading case  
6 Thu Aug 29 2013 17:07:09 GMT-0700 (MST): You're at the la  
7 Thu Aug 29 2013 17:07:10 GMT-0700 (MST): Case was load  
8 Thu Aug 29 2013 17:07:23 GMT-0700 (MST): Running AC lo  
9 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): AC load flow sc  
10 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Total network e  
11 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Total network r  
12 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Total network e  
13 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Total network r  
14 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Total network active loss: 0.13393 p.u.  
15 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Total network reactive loss: 0.30122 p.u.  
16 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Max P mismatch : 0 p.u. at bus: Bus9  
17 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Max Q mismatch : 0 p.u. at bus: Bus5  
18 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): Load flow converge.  
19 Thu Aug 29 2013 17:07:26 GMT-0700 (MST): AC load flow calculation completed successfully.  
20 Thu Aug 29 2013 17:09:32 GMT-0700 (MST): Load flow report style is set to: summary  
21 Thu Aug 29 2013 17:09:37 GMT-0700 (MST): Load flow report style is set to: BusNumberStyle  
22 Thu Aug 29 2013 17:09:54 GMT-0700 (MST): Running AC load flow calculation using method: NR  
23 Thu Aug 29 2013 17:09:56 GMT-0700 (MST): AC load flow calculation completed successfully.  
24 Thu Aug 29 2013 17:09:56 GMT-0700 (MST): Total network active loss: 0.13393 p.u.  
25 Thu Aug 29 2013 17:09:56 GMT-0700 (MST): Total network reactive loss: 0.30122 p.u.  
26 Thu Aug 29 2013 17:09:56 GMT-0700 (MST): Total network active loss: 0.13393 p.u.  
27 Thu Aug 29 2013 17:09:56 GMT-0700 (MST): Total network reactive loss: 0.30122 p.u.

Logiciel InterPSS Cloud  
Via Google Drive

# Logiciels GPL (Programmation visuelle)



Pour les simulations en programmant sans avoir à écrire sans faire de CODE.

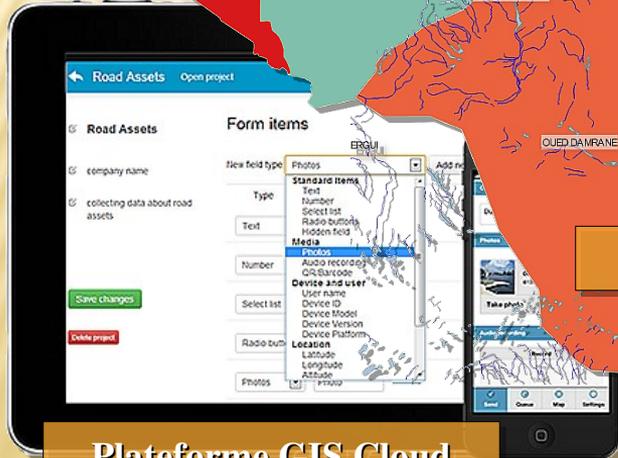
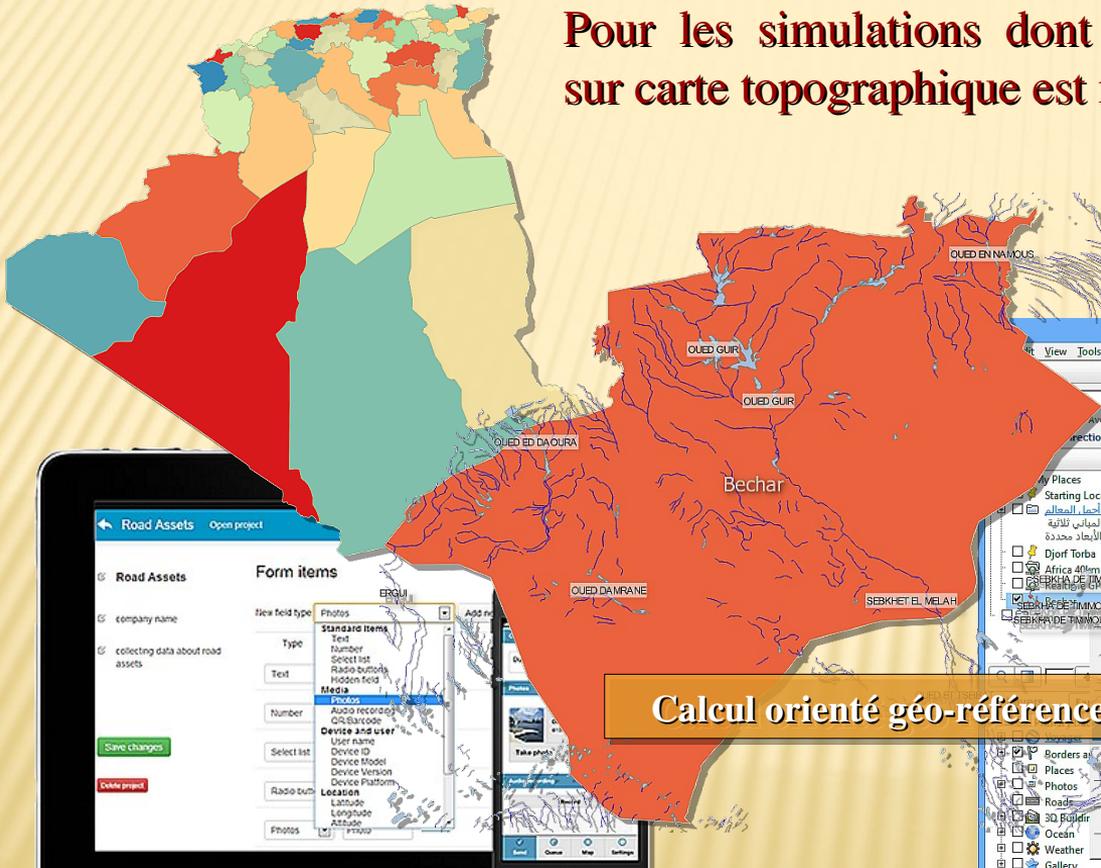


Logiciel Visuino

Logiciel Blockly, Google Visual Programming

# Logiciels GPL (Interface SIG & Simulation)

Pour les simulations dont une projection sur carte topographique est nécessaire.



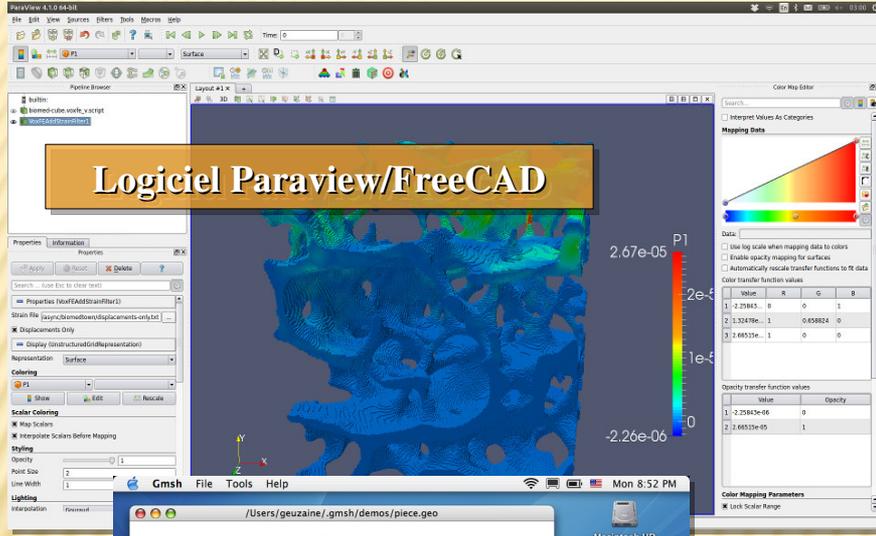
Plateforme GIS Cloud

Calcul orienté géo-référence



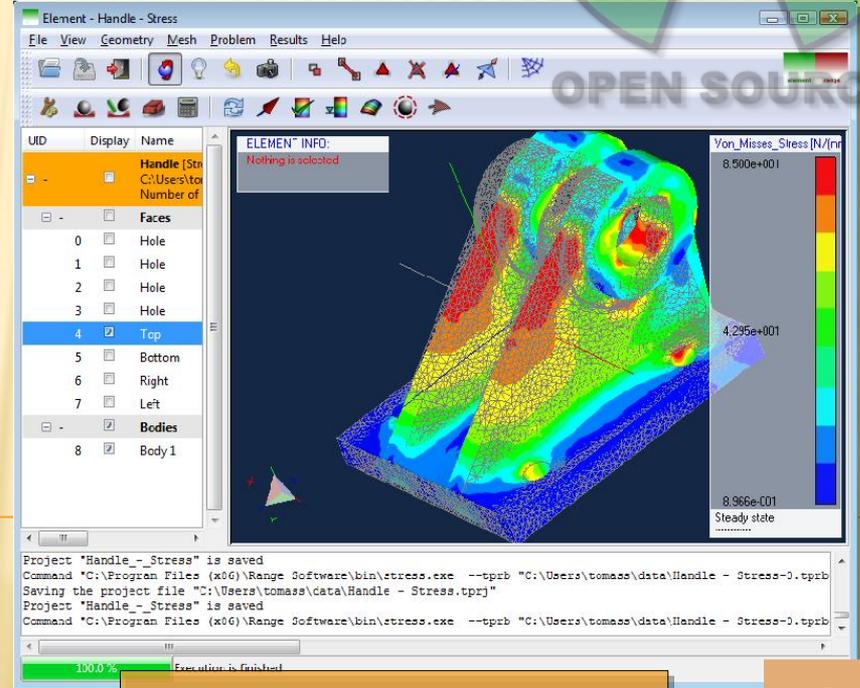
Logiciel QGIS

# Logiciels GPL (Interface FEM & Simulation)



Logiciel Paraview/FreeCAD

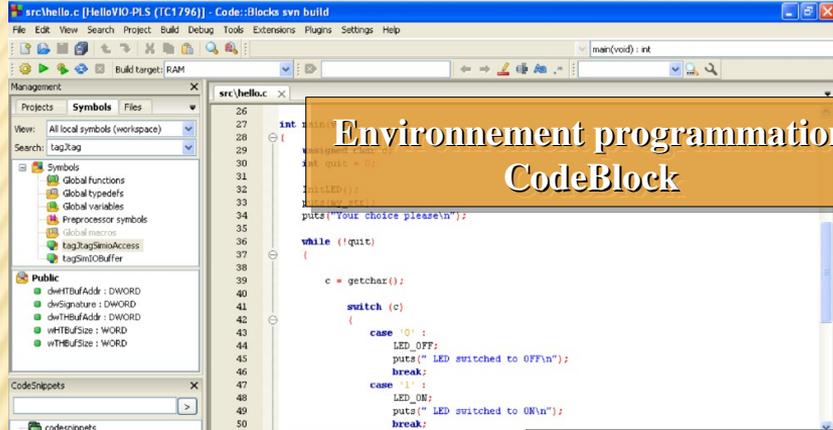
Pour les simulations dont une projection sur espace thermique/mécanique des fluides est nécessaire.



Logiciel GMesh

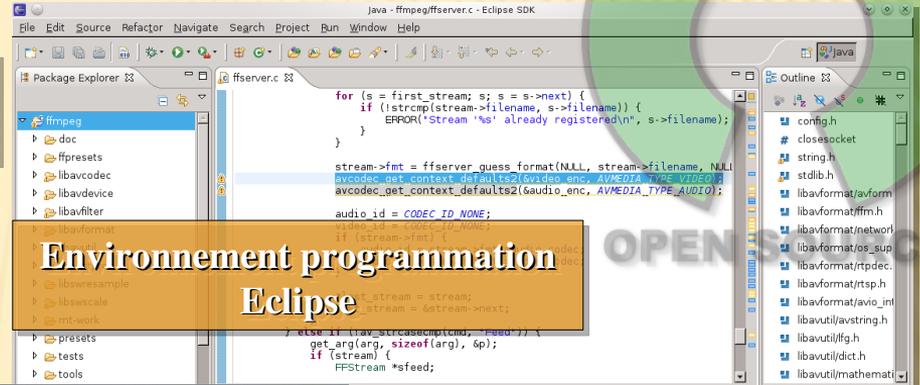
Logiciel Open Cascade

# Logiciels GPL (Interface Processus)



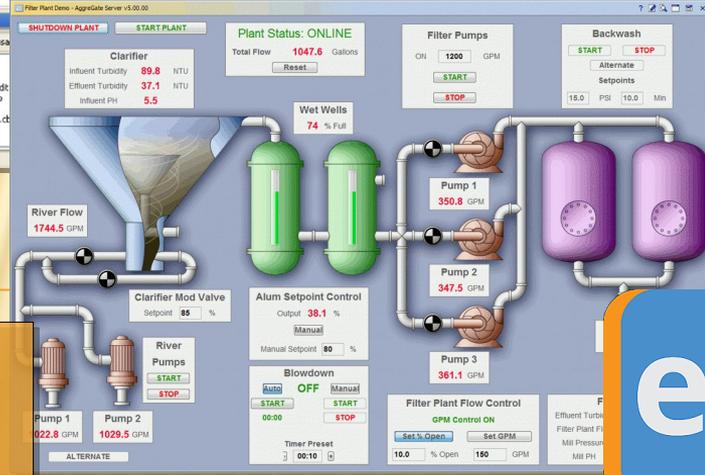
```
src\hello.c [HelloVIO.PLS (TC1796)] - Code::Blocks svn build
File Edit View Search Project Build Debug Tools Extensions Plugins Settings Help
main(void) : int
Build target: RAM
Management
Projects Symbols Files
View: All local symbols (workspace)
Search: tag:tag
Symbols
Global functions
Global typedefs
Global variables
Preprocessor symbols
Global macros
tag:tag:SimioAccess
tag:SimIOBuffer
Public
dwHTBufAddr : DWORD
dwSignature : DWORD
dwHTBufAddr : DWORD
wHTBufSize : WORD
wHTBufSize : WORD
CodeSnippets
codesnippets
```

Environnement programmation  
CodeBlock



```
Java - ffmpegffserver.c - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ffserver.c
for (s = first_stream; s; s = s->next) {
    if (Istrcmp(stream->filename, s->filename)) {
        ERROR("Stream '%s' already registered!\n", s->filename);
    }
}
stream->fmt = ffmpeg_guess_format(NULL, stream->filename, NULL);
avcodec_get_context_defaults2(&stream->codec, AV_CODEC_ID_NONE);
avcodec_get_context_defaults2(&stream->audio, AVMEDIA_TYPE_AUDIO);
audio_id = CODEC_ID_NONE;
if (stream->codec->id == CODEC_ID_NONE)
    stream->codec = avcodec_find_decoder(stream->codec->id);
if (!stream->codec)
    stream = stream->next;
} else if (Istrcmp(stream->filename, "FFStream *sfeed;
get_arg(arg, sizeof(arg), 6p);
if (stream) {
    FFStream *sfeed;
```

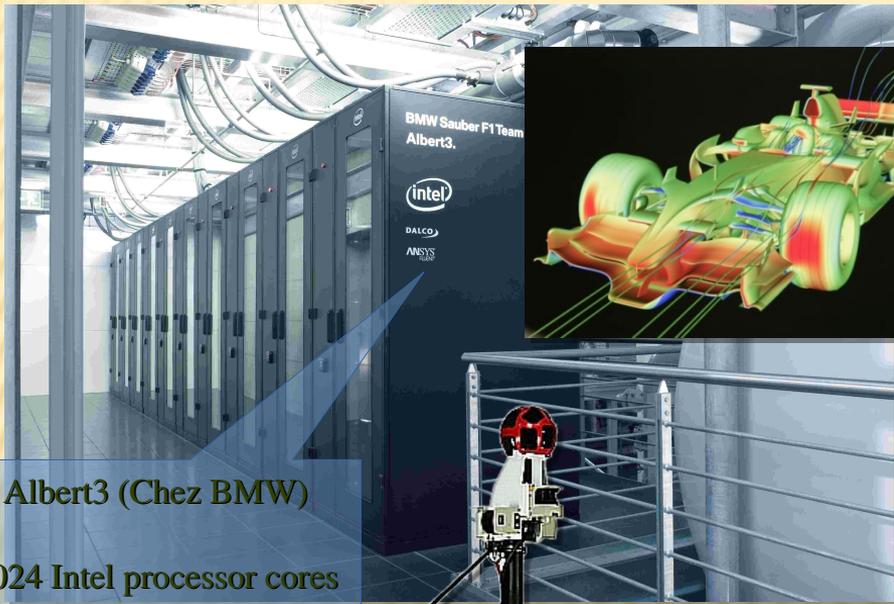
Environnement programmation  
Eclipse



Environnement de supervision  
des processus  
OPEN SCADA (sous Eclipse)



# Logiciels GPL (Environnement HPC)



Albert3 (Chez BMW)

1,024 Intel processor cores  
(256 compute nodes, each  
with two Intel Xeon 5160  
dual core processors)



RaspBerry PI en Cluster

Google Data Center



# Logiciels GPL (Environnement HPC/MPI)

```
/*The Parallel Hello World Program*/  
#include <stdio.h>  
#include <mpi.h>  
main(int argc, char **argv)  
{ int node;  
  MPI_Init(&argc, &argv);  
  MPI_Comm_rank(MPI_COMM_WORLD, &node);  
  printf("Hello World from Node %d\n", node);  
  MPI_Finalize();  
}
```

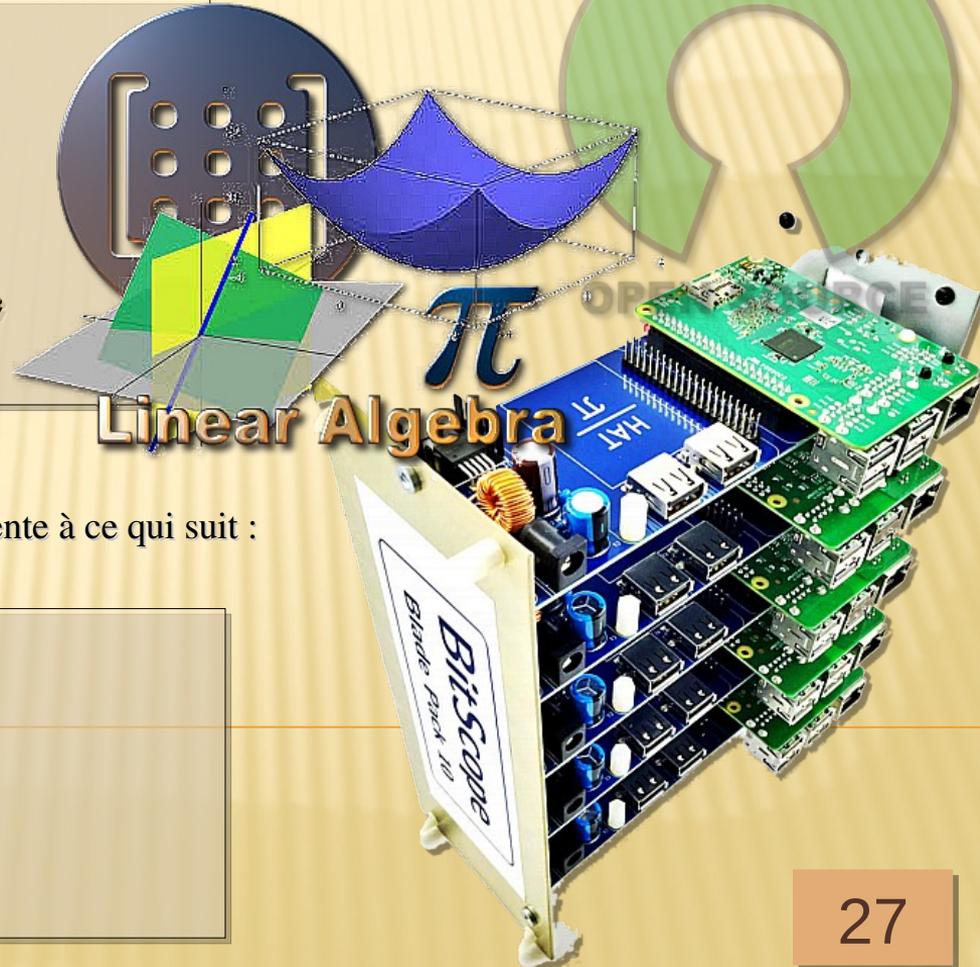
La compilation est lancée par la commande suivante :

```
mpicc hello.c -o hello ;
```

Dans le cas d'UBCluster on s'attend à une réponse équivalente à ce qui suit :

```
Mpirun -hostfile machines hello ;
```

```
Hello World from Node 2  
Hello World from Node 0  
Hello World from Node 4  
Hello World from Node 3  
Hello World from Node 7  
Hello World from Node 1  
Hello World from Node 6  
Hello World from Node 5
```



# Logiciels GPL (Environnement HPC/MPI)

La topologie adoptée pour l'Intranet du centre de calcul de l'université de Béchar porte l'avantage d'être connecté à l'Intranet (LAN, WLAN) et à Internet (WAN). Il est du type **ROCKS Cluster** version 6.3 (2016).

Il intègre beaucoup d'application et d'environnement de calcul :

**FreeFEM++, OpenCascad,**

**Octave**

**Gmesh**

**FreeCAD, ParaView**

**C/C++, BioPerl,**

**BioPython**

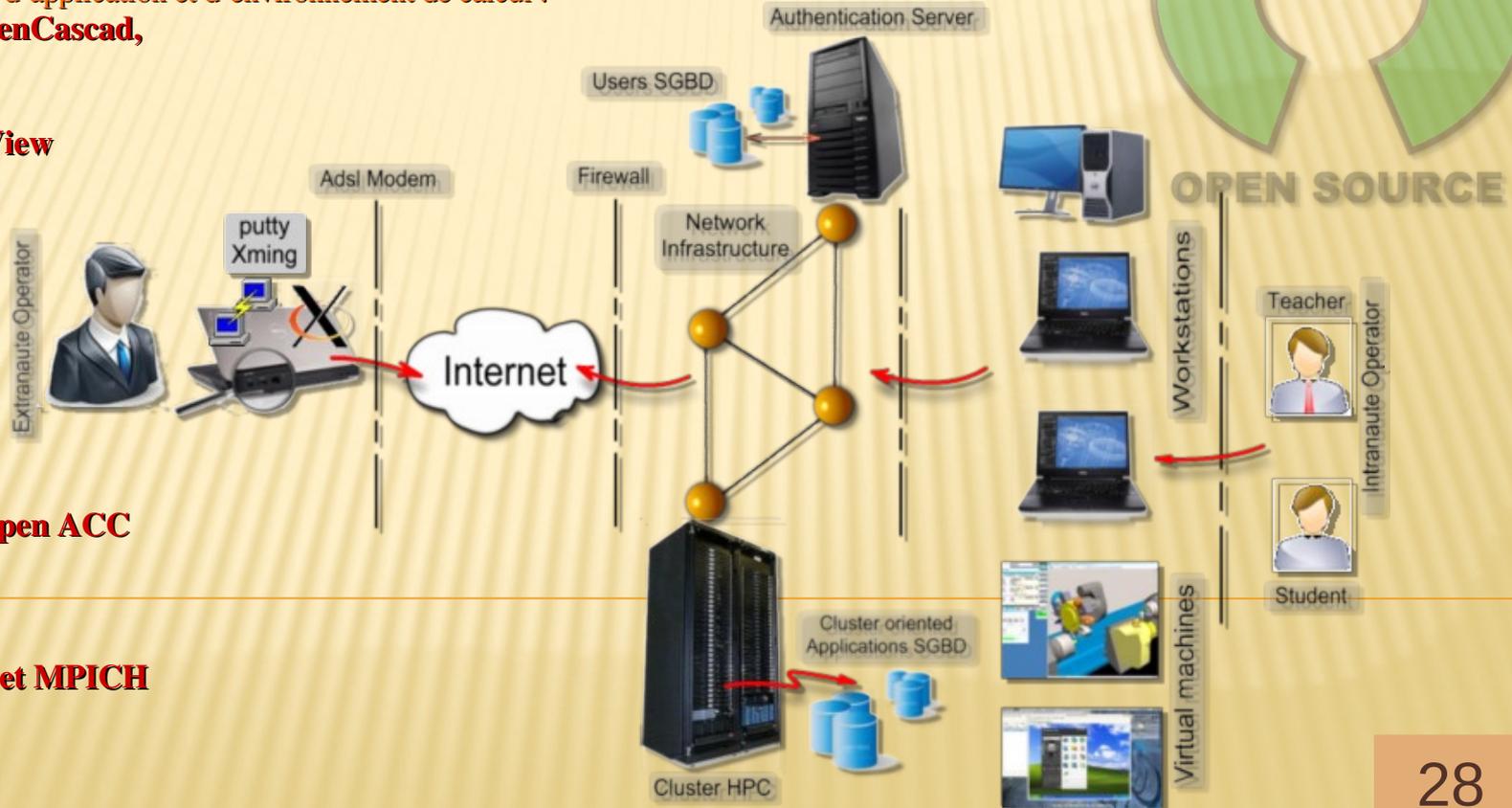
**Fortran, Java, Open ACC**

**SGE Condor**

**Ganglia**

**MPI, Open MPI et MPICH**

**Serveur LAMP**



# Logiciels GPL (Environnement PYTHON)

```
from turing_machine import TuringMachine
initial_state = "init",
accepting_states = ["final"],
transition_function = {("init","0"):( "init", "1", "R"),
                       ("init","1"):( "init", "0", "R"),
                       ("init"," "):( "final", " ", "N"),
                       }
final_states = ["final"]
t = TuringMachine("010011 ",
                  initial_state = "init",
                  final_states = final_states,
                  transition_function=transition_function)

print("Input on Tape:")
t.show_tape()
while not t.final():
    t.step()
print("Result of the Turing machine calculation:")
t.show_tape()
```



OPEN SOURCE

# Logiciels GPL (Environnement PYTHON)

Python est un langage intéressant pour sa facilité et sa richesse en possibilités d'utilisation. En version de base, Python dispose de nombreuses fonctionnalités intégrées en standard. Autour des aptitudes de base standard, Il est extensible par des modules de tout bord, qui sont les contributions de développeurs dévoués à la cause de l'Open Source. Ce sont des bibliothèques qui aident le développeur à travailler sur des projets particuliers. Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python (**cas de tkinter**).

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par **Guido van Rossum**, il a été porté du **Macintosh**, de son développeur, qui travaillait à cette époque au **Centrum voor Wiskunde en Informatica** aux Pays-Bas, jusqu'à se voir associer une organisation à but non lucratif particulièrement dévouée, la PSF (**Python Software Foundation**), créée en 2001. Ce langage a été baptisé ainsi en hommage à la troupe de comiques les «**Monty Python**».



Développeur Informaticien  
Néerlandais (langage Python)  
Guido van Rossum  
31 janvier 1956

CWI



**Monty Python** est le nom d'une troupe célèbre d'humoristes britanniques (1969).

# Logiciels GPL (Environnement PYTHON)

## Installation

La plateforme ANACONDA installe tout le nécessaire (l'environnement PYTHON 3.X/2.X) et donnera accès aux possibilités d'étendre l'installation aux modules PYTHON nécessaires pour le développement de calcul scientifique de l'ingénieur ou du chercheur.

Les différents modules (bibliothèques) nécessaires sont :

**NUMPY** : Analyse numérique & Matrices

**SCIPY** : Extension pour calcul scientifique

**MATPLOTLIB** : Dessin de courbe

**SYMPY** : calcul symbolique

**NETWORKX** : manipulation et gestion des graphes

**PLOTLY** : Librairie en ligne pour dessin de courbes

**EVENTLET** : Librairie gestion événementielle

**TKINTER** : Librairie pour manipulation d'interface graphique

**IPYTHON** : Bibliothèque pour calcul scientifique

**CYTHON** : Interface pour environnement C/C++

**JPYTHON** : Interface pour environnement JAVA

**JUPYTER** : Editeur orienté navigateur

**DJANGO-CMS** : System de gestion de contenus

**WIDGY** : CMS utilisant le Framework Django.

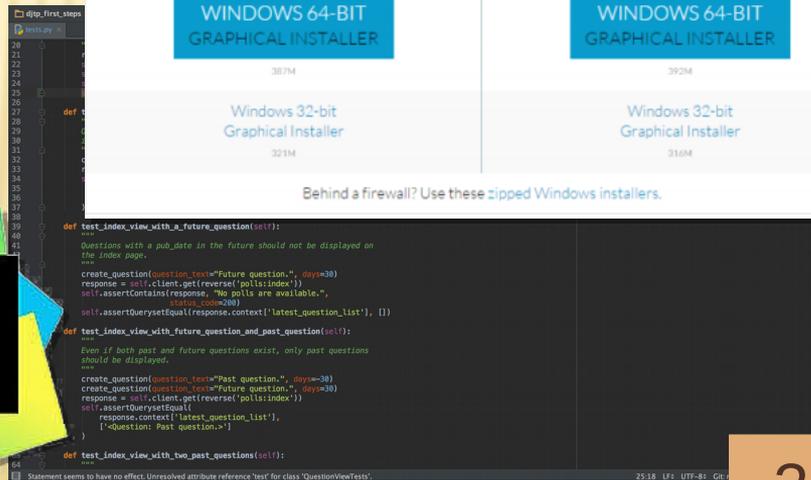
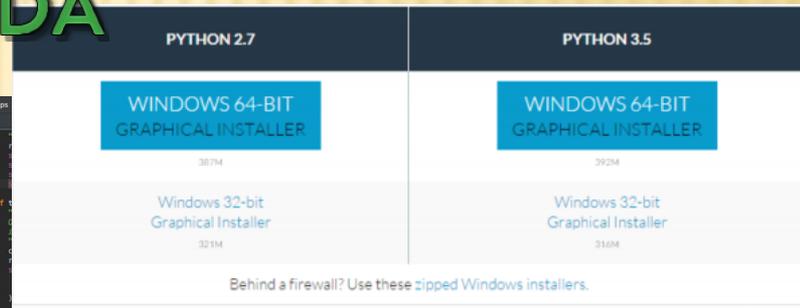
...encore ! ... des milliers de modules.



# ANACONDA

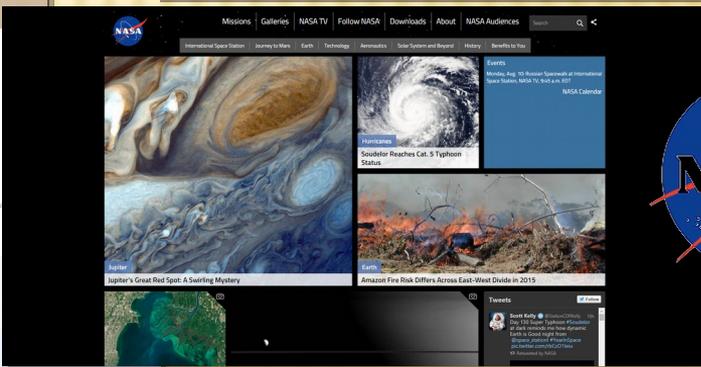
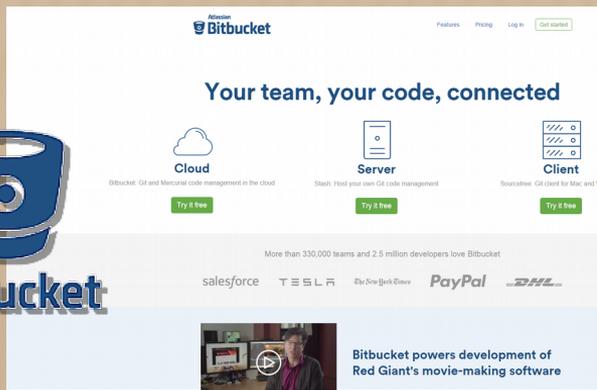
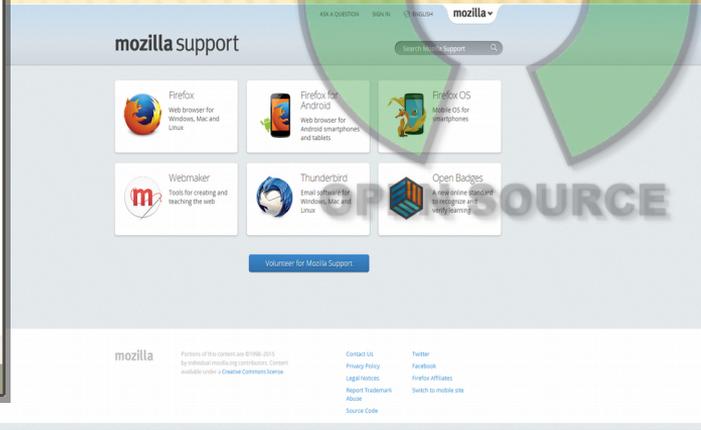
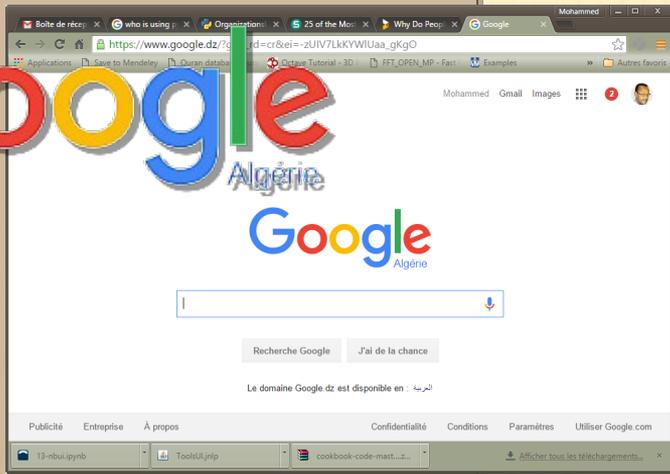


OPEN SOURCE



# Logiciels GPL (PYTHON, qui l'utilise)

YouTube  
DropBox  
Survey Monkey  
Google  
Quora  
Bitly  
Reddit  
Yahoo Maps  
Hipmunk  
Pinterest  
Instagram  
Disqus  
Spotify  
Washington Post  
Firefox  
NASA  
Bitbucket  
Reddit Gifts  
Prezi  
Playfire  
Eventbrite



# Conclusions

- De nos jours, avec l'augmentation de l'utilisation de l'ordinateur individuel/portable, les logiciels (**GPL**) de modélisation et de simulation de processus en génie électrique sont devenus accessibles en version **Mobile/Bureau ou Serveur**.
- Les applications de modélisation et de simulation des processus se réfèrent à tous les aspects de gestion et d'utilisation de données de domaines par projection géographiques numériques sur une interface utilisateur.
- L'outil logiciel **GPL** reste jusqu'à lors, un outil préféré pour la conception des idées et des simulations des situations selon des stratégies locales ou globales.
- L'observation scientifique, la modélisation et la simulation sont des outils de manœuvres, beaucoup importants, mais pas plus. Les ressources logiciels GPL couvrent tous les besoins tant pour la pratique d'une pédagogie que pour l'investigation du domaine des chercheurs.
- Leur optimisation reste pour toujours une question de possibilités offertes à l'opérateur pour améliorer selon son besoin et sans enfreindre à l'équilibre des compositions et relations totales. surtout, en terme de **PROPRIÉTÉS INTELLECTUELLES**.
- Les libertés de l'introduction d'une certaine mise à jour est toujours valable, sauf nécessité de garantir la non interférence avec la sûreté des ensembles voisins.



**MERCI POUR VOTRE  
PATIENCE & ATTENTION**



**OPEN SOURCE**