

---

# Uranie Platform v1.1

## 35th ESReDA Seminar

F. Gaudier

CEA/DEN/DANS/DM2S/SFME/LGLS

Marseille, November, 19 2008

---

cea

---



ROOT

URANIE ...

benchmark ..

Sensitivity ...

# ROOT

---

CERN Large Hadron Collider (LHC)  
ROOT ( <http://root.cern.ch> )



ROOT  
URANIE ...  
benchmark ..  
Sensitivity ...



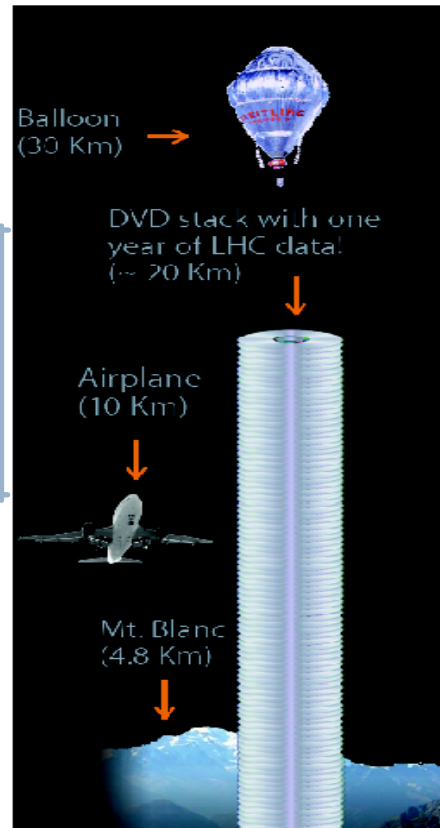
# CERN Large Hadron Collider (LHC)

- particle accelerator
- 27 km circumference tunnel in Genève
- 4 experiments (ATLAS, CMS, ALICE, LHCb)

Study the matter structure

- Research the Higgs boson
- Research new physics

- Data quantity generated : 20 PetaBytes/year
- ROOT is the framework to store, treat and analyze this data



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



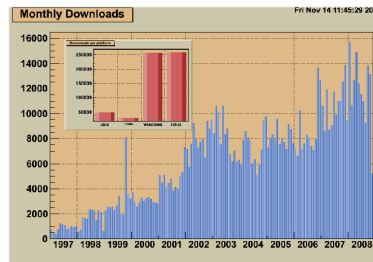
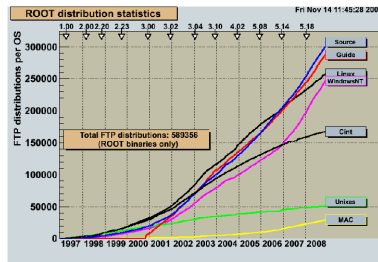
ROOT is an object-oriented framework to large scale data analysis and data mining.



- 10 years of development (C++ with 3-4 releases/year )
- multi-platform (Unix, Windows, Mac OS X)
- Offer :
  - A **C++ interpreter**
  - A hierarchical object-oriented database (machine independent, highly compressed, supporting schema evolution and object versioning)
  - Shared libraries (*automatic loading with "rootmap"*)
  - Advanced statistical analysis tools
  - Advanced visualization tools
- **License LGPL**



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



URANIE : CEA/DEN Uncertainty Platform

URANIE : Fonctional diagram

URANIE : Graphical User Interface

URANIE - XML User Interface

URANIE : Batch mode

Projects using URANIE

"DataServer" module - URANIE ASCII file format



ROOT





URANIE ...

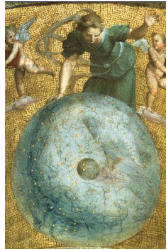
benchmark ...

Sensitivity ...

# URANIE : CEA/DEN Uncertainty Platform



-  ROOT (CERN),  MIXMOD (Gaussian Mixtures - INRIA),  
 OPT++ (Optimization - Sandia), CLUB (Computation - CNES),
-  QT (Graphical User Interface)
- Data access :
  - Flat file with header ( "Salomé Table" )
  - TTree (internal ROOT)
  - SQL Data base (MySQL, PostgreSQL, ...)
- Uncertainty/Sensitivity methods in URANIE
  - Design Of Experiments (SRS, LHS, ROA, qMC, MCMC, Copulas)
  - Surrogate models (Polynomial , Artificial Neural Networks, Splines)
  - Non Intrusive Spectrale Projection : Generalized Polynomial Chaos
  - Sensitivity analysis (Pearson, Spearman, Sobol, Fast, Morris)
  - Optimization, Multi-Criteria (Genetic Algorithms)
  - Computing distribution



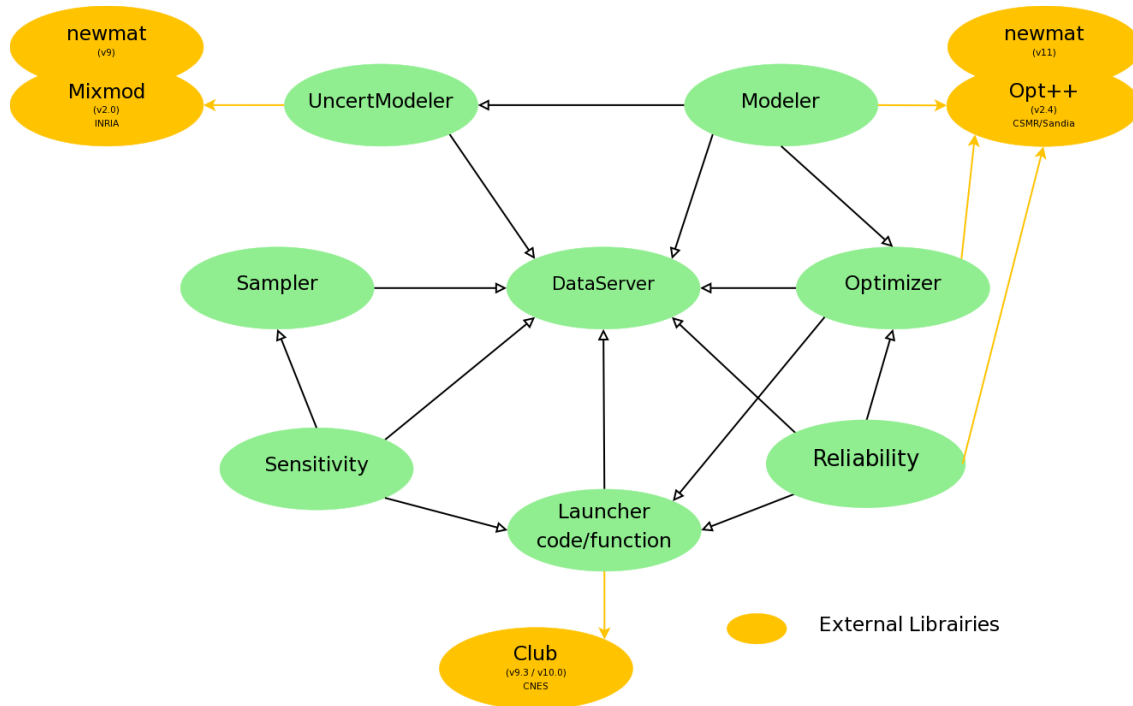
ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



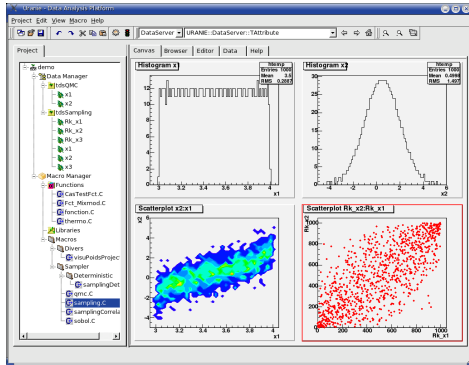
# URANIE : Functional diagram



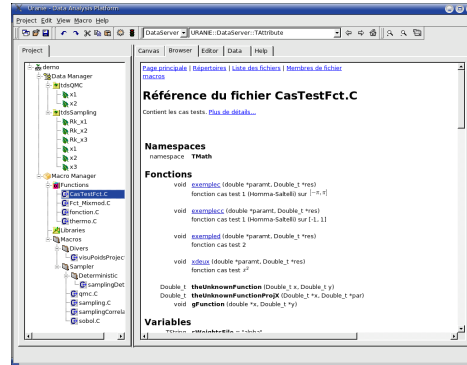
ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



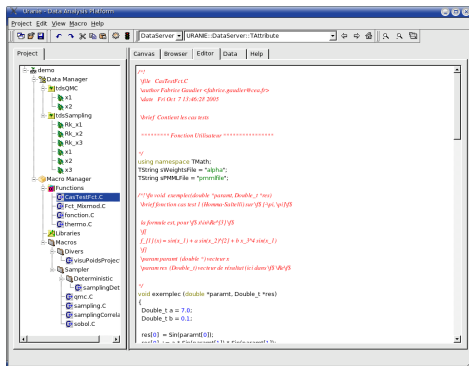
# URANIE : Graphical User Interface



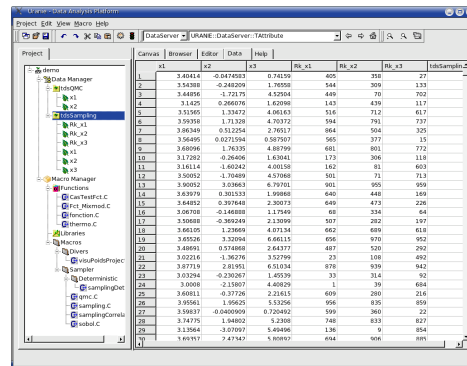
Visualization



User Help



Editor



Spreadsheet

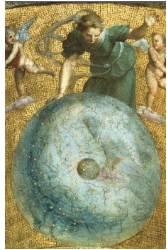


# URANIE - XML User Interface



XML file ( problem\_uranie.xml )

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE Problem SYSTEM "/home/uranie/tools/share/uranie/uranie.dtd" >
<Problem>
  <Header name="Etude" title="projet GENTR">
    <Application name="uranie" version="0.4"/>
  </Header>
  <DataDictionary>
    <DataField name="x1" law="uniform" min="0.5" max="1.5"/>
    <DataField name="x2" law="normal" mean="2.5" std="0.25"/>
  </DataDictionary>
  <Sampler method="SRS" N="1500" export="data/sampler_SRS_1500.dat"/>
  <Sampler method="LHS" N="1000" export="data/sampler_LHS_1000.dat"/>
</Problem>
```



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...

uranie -s problem\_uranie.xml



# URANIE : Batch mode



> root myScript.C

```
ernacs: myScript.C
File Edit View Cgds Tools Options Buffers C++ Help
Open Close Save Print Cut Copy Paste Undo Spell Replace Info Compile Debugs News

myScript.C
void myScript()
{
  gStyle->SetPalette(1);
  // The input variables
  TAttribute *x1 = new TAttribute("x_{1}");
  TAttribute *x2 = new TAttribute("x_{2}", 0.10, 0.30);
  TAttribute *x3 = new TAttribute("x_{3}", 200., 300.);
  x3->setKey("x_{3}");

  // Generate the sample
  TSampling *fsampling = new TSampling("lhs", 100);
  fsampling->addVariable(x1, "normal", 0.20, 0.04);
  fsampling->addVariable(x2, "normal", 0.20, 0.05);
  fsampling->addVariable(x3, "uniform");
  fsampling->genSample();

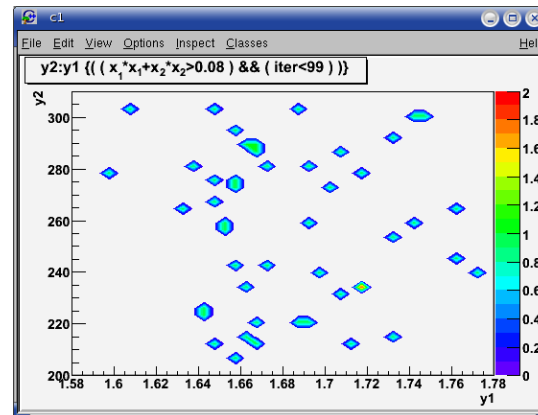
  // Le répertoire où se trouve les fichiers d'entrées de référence
  TString sDirectory = gSystem->Getenv("URANIESYS");
  sDirectory += TString("/bin");

  // The input files
  TInputFile *file1 = new TInputFile( Form("%s/input1.dat", sDirectory.Data());
  TInputFile *file2 = new TInputFile( Form("%s/input2.dat", sDirectory.Data());
  // Add the attributes in the input files
  file1->addAttribute(x1);
  file1->addAttribute(x2);
  file2->addAttribute(x3);

  // The output file and output variables
  TOutputFile *fout = new TOutputFile("output1.dat");
  fout->addAttribute(new TAttribute("y1"));
  fout->addAttribute(new TAttribute("y2"));
  fout->addAttribute(new TAttribute("y3"));

  // Definition of the code
  TCode *mcode = new TCode();
  mcode->setWorkingDirectory(gSystem->Getenv("HOME") +
  mcode->setCommand(Form("%s/bin/launcherUranie/bldon")); // The working directory
  mcode->addInputFile(file1); // The commands
  mcode->addInputFile(file2); // The input files
  mcode->addOutputFile(fout); // The output file

  // code launcher
  TLauncher *launcher = new TLauncher(fsampling->getTupleSample(), mcode);
  launcher->setVarDraw("y2:y1", "%{1}x_{1}x_{2}x_{2}>0.08", "contz");
  launcher->run();
}
Run ->XEmacs: myScript.C ID: /B' (C++ down Font Abbrev)---L1--C0--A11-----
Not over a window.
```



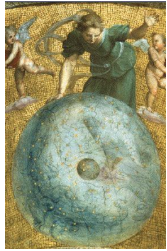
ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# Projects using URANIE

---

- LEONAR tool for severe accidents in french nuclear reactor (**CEA-EDF**)
- PSI-Matador Methodology : Dosimetry computation in french nuclear reactor (**CEA-EDF**)
- EHPOC project : Meteor code (**CEA**)
- Sensitivity Analysis for Cathare code (**CEA/Areva TA**)
- ALLIANCES platform (**CEA/ANDRA/EDF**)  
is to provide a working environment for the simulation and analysis of phenomena to be taken into account for waste storage and disposal studies.
- European project **NURESIM/NURISP**  
The European Platform for NUClear REactor SIMulations, NURESIM, is a Common European Standard Software Platform for modeling, recording, and recovering computer data for nuclear reactors simulations.



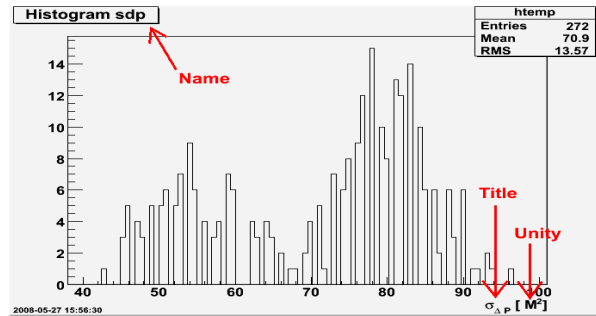
ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# "DataServer" module - URANIE ASCII file format



```
#NAME: geyser
#TITLE: geyser data
#DATE: Mon Mar 12 23:41:09 2007
#COLUMN_NAMES: x1| sdp
#COLUMN_TITLES: x_1| #sigma_{#Delta P}
#COLUMN_UNITS: Sec| M^{2}
----- empty line -----
3.600 79.000
1.800 54.000
...
```

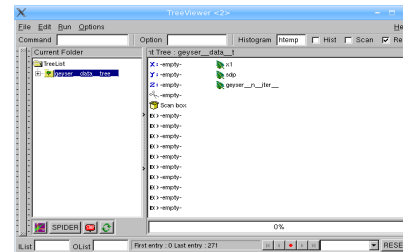


`tds->draw("sdp");`

Only the "#COLUMN\_NAMES:" line is obligatory

**WARNING** : the empty line between the header and the matrix data

```
TDataServer *tds = new TDataServer();
tds->fileDataRead("geyser.dat");
tds->draw("sdp");
tds->startViewer();
```



# benchmark "FlowRate"

---

benchmark "FlowRate"

benchmark "FlowRate" : Analytical Function

benchmark "FlowRate" : Code

Polynomial regression from Ho and Xu(2000)

benchmark "FlowRate" : Macro Sampling

benchmark "FlowRate" : Macro Sampling

benchmark "FlowRate" : XML Interface



ROOT

URANIE ...

benchmark ..

Sensitivity ...



# benchmark "FlowRate"

Analytical function of  $R^8$  in  $R$  defined by :

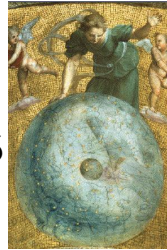
$$f(x) = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_\omega}\right) \left[ 1 + \frac{2LT_u}{\ln\left(\frac{r}{r_\omega}\right)r_\omega^2 K_\omega} + \frac{T_u}{T_l} \right]}$$

with

1.  $r_\omega \sim \mathcal{U}[0.05, 0.15]$  : radius of borehole (m)
2.  $r \sim \mathcal{U}[100, 50\ 000]$  : radius of influence (m)
3.  $T_u \sim \mathcal{U}[63\ 070, 115\ 600]$  : Transmissivity of upper aquifer ( $m^2/\text{year}$ )
4.  $T_l \sim \mathcal{U}[63.1, 116]$  : Transmissivity of lower aquifer ( $m^2/\text{year}$ )
5.  $H_u \sim \mathcal{U}[990, 1\ 110]$  : Potentiometric head of upper aquifer (m)
6.  $H_l \sim \mathcal{U}[700, 820]$  : Potentiometric head of lower aquifer (m)
7.  $L \sim \mathcal{U}[1\ 120, 1\ 680]$  : length of borehole (m)
8.  $K_\omega \sim \mathcal{U}[9\ 855, 12\ 045]$  : hydraulic conductivity of borehole (m)



Fang, p 35



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# benchmark "FlowRate" : Analytical Function

Prototype : void myFunction (Double\_t \*inp, Double\_t \*out)

$$y = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_\omega}\right) \left[ 1 + \frac{2LT_u}{\ln\left(\frac{r}{r_\omega}\right) r_\omega^2 K_\omega} + \frac{T_u}{T_l} \right]}$$

```
#include "TMath.h"

void WaterBoreHole(double *x, Double_t *y)
{
  Double_t drw = x[0], dr = x[1];
  Double_t dtu = x[2], dtl = x[3];
  Double_t dhu = x[4], dhl = x[5];
  Double_t dl = x[6], dkw = x[7];

  Double_t dnum = 2.0 * TMath::Pi() * dtu * ( dhu -dhl);
  Double_t dlnronrw = TMath::Log( dr / drw);
  Double_t dden = dlnronrw * ( 1.0 + ( 2.0 * dl * dtu ) / ( dlnronrw * drw * drw * dkw)
+ dtu / dtl );

  y[0] = dnum / dden;
}
```



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# benchmark "FlowRate" : Code



Usage: flowrate [-d] [-k] [[-f] file] [[-k] file] [-h|-?]  
-d: debug mode  
-v: verbose mode  
-f: input file with flag [flowrate\_input\_with\_flag.in]  
-k: input file with key [flowrate\_input\_with\_keys.in]  
-h,-?: this help message



```
flowrate_input_with_k...
#
# INPUT FILE with KEYS for the "FLOWREATE" code
# \date 2008-04-22 12:53:35
# parameters : 8
#
date = 123456 ;

Rw = 0.0500 ;
R = 33366.67 ;
Tu = 63070.0 ;
Tl = 116.00 ;
Hu = 1110.00 ;
Hl = 768.57 ;
L = 1200.0 ;
Kw = 11732.14 ;

end = 6 ;

Raw-----XEmacs: flowrate_input_with_keys.in 12:07
```

with Keys

```
flowrate_input_with_f... flowrate_input_with_f...
# INPUT FILE with FLAG for the "FLOWREATE" code
# \date 2008-04-22 12:55:17
#
new Implicit_Steady_State sch {
  frottement_pari { 0.0500 33366.67 }
  tinit 0.0
  tmax 1000000.
  nb_pas_dt_max 1500
  dt_min 1110.00
  dt_max 768.57
  Facsec 1000000.
  kWh 11732.14
  information_Tu Champ_Uniforme 1 63070.0
  information_Tl Champ_Uniforme 1 116.00
  information_L {
    precision 1200.0
  }
  convergence {
    criterion relative_max_du_dt
    precision 1.e-6
  }
  Solveur Newton3 {
    max_iter_matrice 1
    max_iter_implicite 1
    date 5654321
    seuil_convg_implicite 1.e-6
    assemblage_implicite 10
    solveur_lineaire BiCGS
    preconditionneur ILU
    seuil_resol_implicite 1.e-5
  }
}
```

with Flags

ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# Polynomial regression from Ho and Xu(2000)

Uniform Design from Ho and Xu(2000) : 32 points

$$\begin{aligned} \log(\hat{y}) = & 4.156 + 1.9903(\log r_\omega + 2.3544) - 0.0007292(L - 1400) - .003554(H_l - 760) \\ & + .0035068(H_u - 1050) + .000090868(K_\omega - 10950) - .00087286(\log(r) - 8.8914) \\ & + .000015325(H_u - 1050)(H_l - 760) + .00000026487(L - 1400)^2 \\ & - .0000071759(H_l - 760)^2 - .0000068021(H_u - 1050)^2 \end{aligned}$$

```
void WaterBoreHoleHoXu(double *x, Double_t *y)
{
  Double_t drw = x[0], dr = x[1];
  Double_t dtu = x[2], dtl = x[3];
  Double_t dhu = x[4] - 1050.0, dhl = x[5] - 760.0;
  Double_t dl = x[6] - 1400.0, dkw = x[7] - 10950.0;

  Double_t yHoXu = 4.1560 + 1.9903 * (TMath::Log(drw) + 2.3544);
  yHoXu -= .0007292 * dl;
  yHoXu -= .003554 * dhl;
  yHoXu += .0035068 * dhu;
  yHoXu += .000090868 * dkw;
  yHoXu += .000015325 * dhu * dhl;
  yHoXu += .00000026487 * dl * dl;
  yHoXu -= .0000071759 * dhl * dhl;
  yHoXu -= .0000068021 * dhu * dhu;
  yHoXu -= .00087286 * (TMath::Log(dr) - 8.8914);

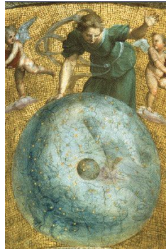
  y[0] = TMath::Exp(yHoXu);
}
```



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# benchmark "FlowRate" : Macro Sampling



```
{
TDataServer *tds = new TDataServer("sampllhs", "Sampling LHS");
tds->addAttribute( new TUniformDistribution("rw", 0.05, 0.15));
tds->addAttribute( new TUniformDistribution("r", 100.0, 50000.0));
tds->addAttribute( new TUniformDistribution("tu", 63070.0, 115600.0));
tds->addAttribute( new TUniformDistribution("tl", 63.1, 116.0));
tds->addAttribute( new TUniformDistribution("hu", 990.0, 1110.0));
tds->addAttribute( new TUniformDistribution("hl", 700.0, 820.0));
tds->addAttribute( new TUniformDistribution("l", 1120.0, 1680.0));
tds->addAttribute( new TUniformDistribution("kw", 9855.0, 12045.0));

TSampling *sampling = new TSampling(tds, "lhs", 500);
sampling->generateSample();

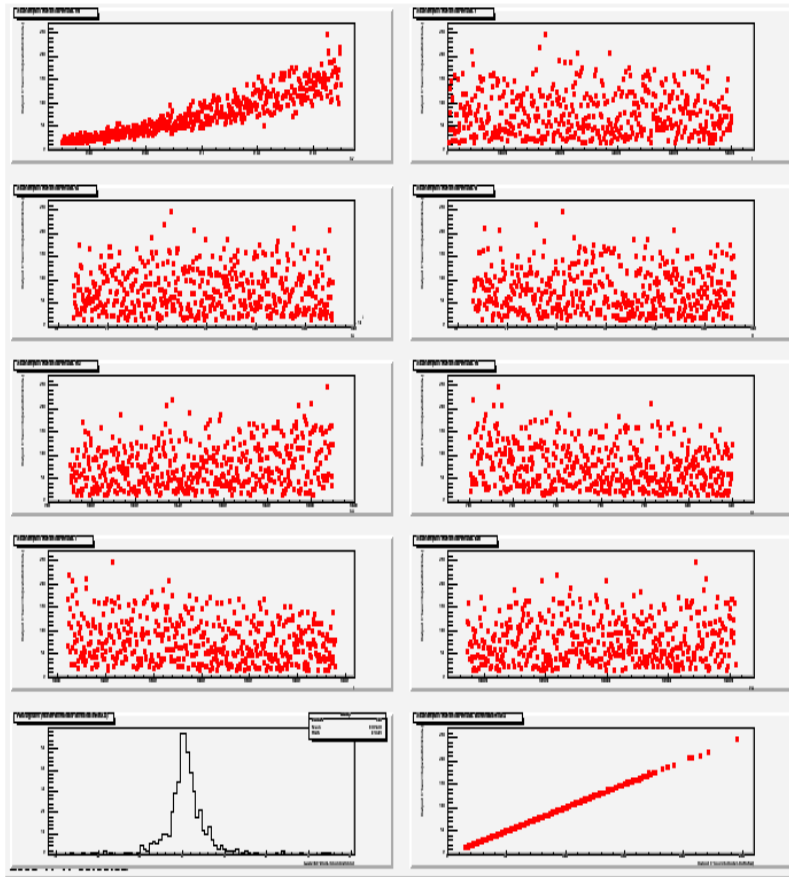
TLauncherFunction * tlf = new TLauncherFunction(tds, WaterBoreHole,"", "waterborehole");
tlf->run();

TLauncherFunction * tlf = new TLauncherFunction(tds, WaterBoreHoleHoXu,"", "boreholeHoXu");
tlf->run();

TCanvas *c = new TCanvas();
c->Clear(); c->Divide(2,5);
c->cd(1); tds->draw("waterborehole:rw");
c->cd(2); tds->draw("waterborehole:r");
c->cd(3); tds->draw("waterborehole:tu");
c->cd(4); tds->draw("waterborehole:tl");
c->cd(5); tds->draw("waterborehole:hu");
c->cd(6); tds->draw("waterborehole:hl");
c->cd(7); tds->draw("waterborehole:l");
c->cd(8); tds->draw("waterborehole:kw");
c->cd(9); tds->draw("waterborehole-boreholeHoXu");
c->cd(10); tds->draw("waterborehole:boreholeHoXu");
c->SaveAs("../figure/flowrateborholeSampling.png");
}
```

ROOT  
URANIE ...  
benchmark ..  
Sensitivity ...

# benchmark "FlowRate" : Macro Sampling



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# benchmark "FlowRate" : XML Interface



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...

```
1<?xml version="1.0" encoding="iso-8859-1"?>
2<?stylesheet type="text/xsl" href="/home/uranie/tools/share/uranie/uranie.xslt"?>
3<!DOCTYPE Problem SYSTEM "/home/uranie/tools/share/uranie/uranie.dtd">
4<Problem>
5  <Header name="boreholeXML" title="Launch the Borehole function in XML" debug="0">
6    <Application name="uranie" version="1.0"/>
7  </Header>
8  <DataDictionary>
9    <DataField name="rw" law="uniform" min="0.05" max="0.15"/>
10   <DataField name="r" law="uniform" min="100.0" max="50000.0"/>
11   <DataField name="tu" law="uniform" min="63070.0" max="115600.0"/>
12   <DataField name="tl" law="uniform" min="63.1" max="116.0"/>
13   <DataField name="hu" law="uniform" min="990.0" max="1110.0"/>
14   <DataField name="hl" law="uniform" min="700.0" max="820.0"/>
15   <DataField name="l" law="uniform" min="1120.0" max="1680.0"/>
16   <DataField name="kw" law="uniform" min="9855.0" max="12045.0"/>
17 </DataDictionary>
18 <Sampler method="LHS" N="100" export="waterhole_sampler_lhs.dat"/>
19 <Launcher macro="FunctionsToCompile.C" function="WaterBoreHole" output="wh" export="waterholelhs.dat"/>
20 <Sampler method="SRS" N="2000" export="waterhole_sampler_srs.dat"/>
21 <Launcher function="WaterBoreHoleHoXu" input="r:rw:tu:tl:hu:hl:l:kw" export="waterholeholusrs.dat"/>
22</Problem>
```



# Sensitivity Analysis

---

Sensitivity Indexes from Regression (SRC)

Morris Method (27 evaluations)

Sobol Indexes from "Saltelli" Method

Sobol Indexes from "Fast" Method

Sobol Indexes from Polynomial Chaos



ROOT

URANIE ...

benchmark ..

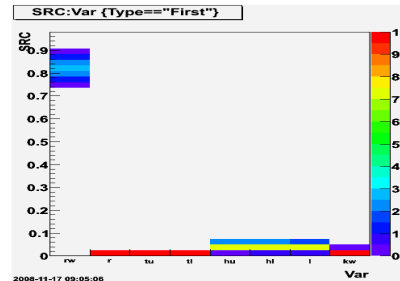
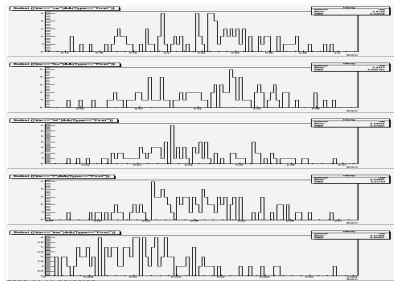
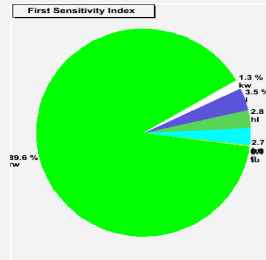
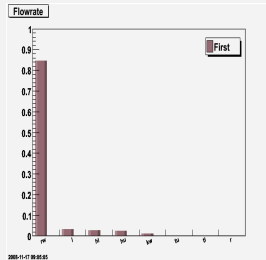
Sensitivity ...



# Sensitivity Indexes from Regression (SRC)



```
TRegression * treg = new TRegression(tds, "rw:r:tu:tl:hu:hl:l:kw", "waterborehole", "SRC");
treg->computeIndexes();
treg->drawIndexes("Flowrate", "", "nonewcanv,hist,first");
treg->drawIndexes("Flowrate", "", "nonewcanv,pie,first");
```

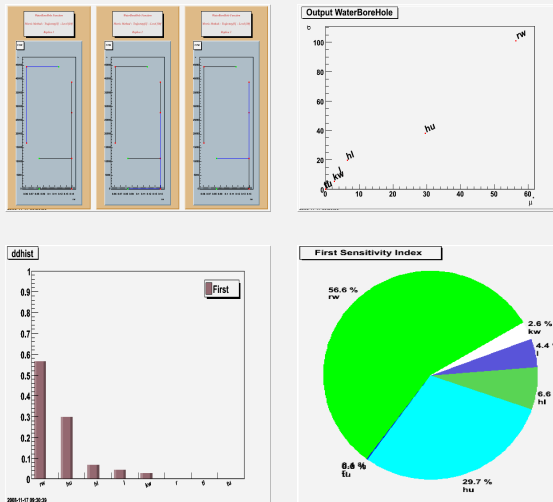


ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# Morris Method (27 evaluations)

```
TMorris * tmor = new TMorris(tds, WaterBoreHole, 3, 10)
tmor->computeIndexes();
tmor->drawSample("rw:r", 1);
tmor->drawIndexes("mustar");
tmor->drawIndexes("Flowrate", "", "nonewcanv,hist,first");
tmor->drawIndexes("Flowrate", "", "nonewcanv,pie,first");
```



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...

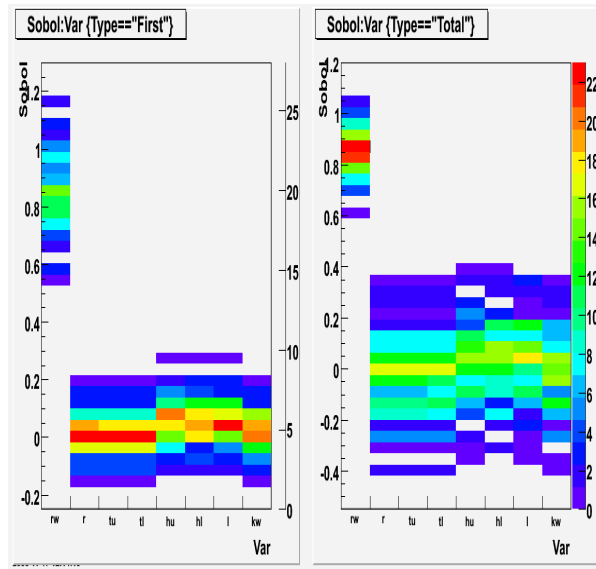
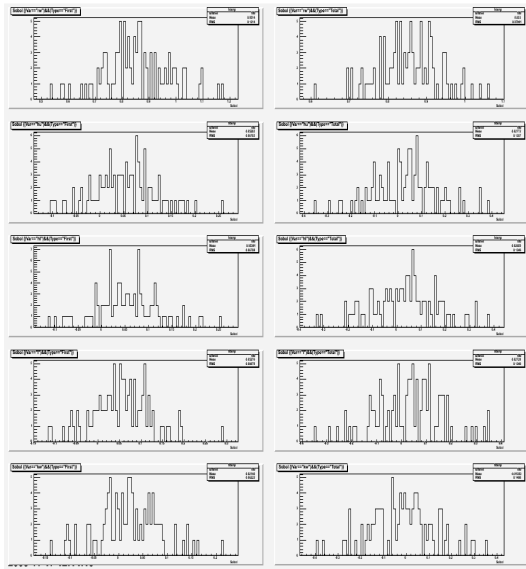


# Sobol Indexes from "Saltelli" Method



```
TSobol * tsob = new TSobol(tds, WaterBoreHole, 40000);  
tsob->computeIndexes();
```

Monte-Carlo : 100 simulations with 40 000 evaluations each one:



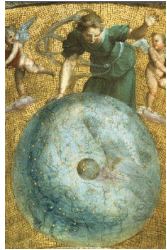
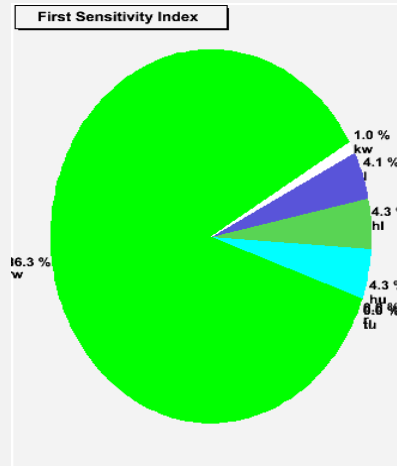
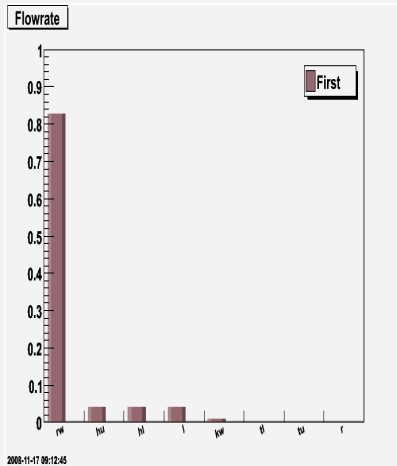
ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



# Sobol Indexes from "Fast" Method



```
TFast * tfast = new TFast(tds, WaterBoreHole, 500);  
tfast->computeIndexes();  
tfast->drawIndexes("Flowrate", "", "nonewcanv,hist,first");  
tfast->drawIndexes("Flowrate", "", "nonewcanv,pie,first");
```



ROOT  
URANIE ...  
benchmark ...  
Sensitivity ...



