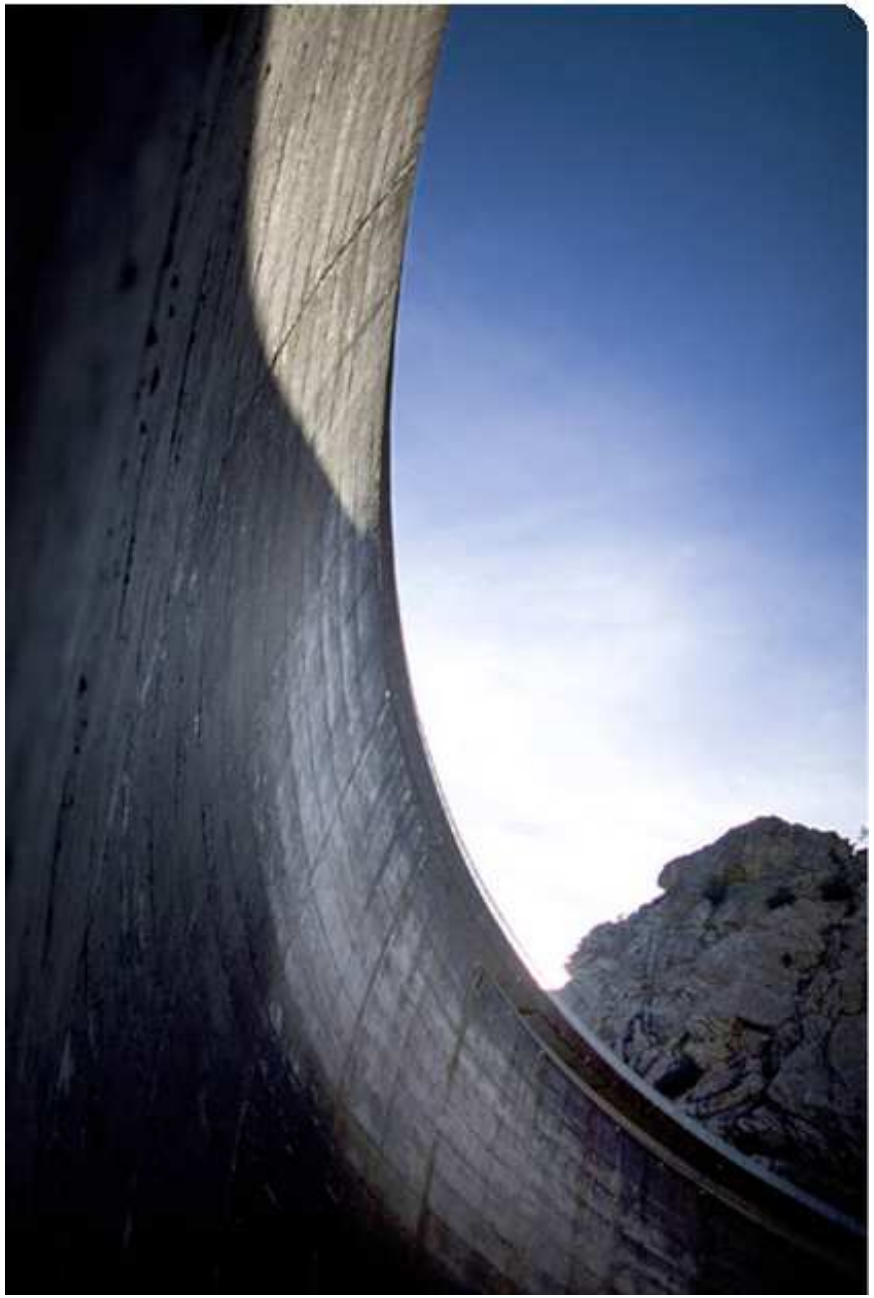
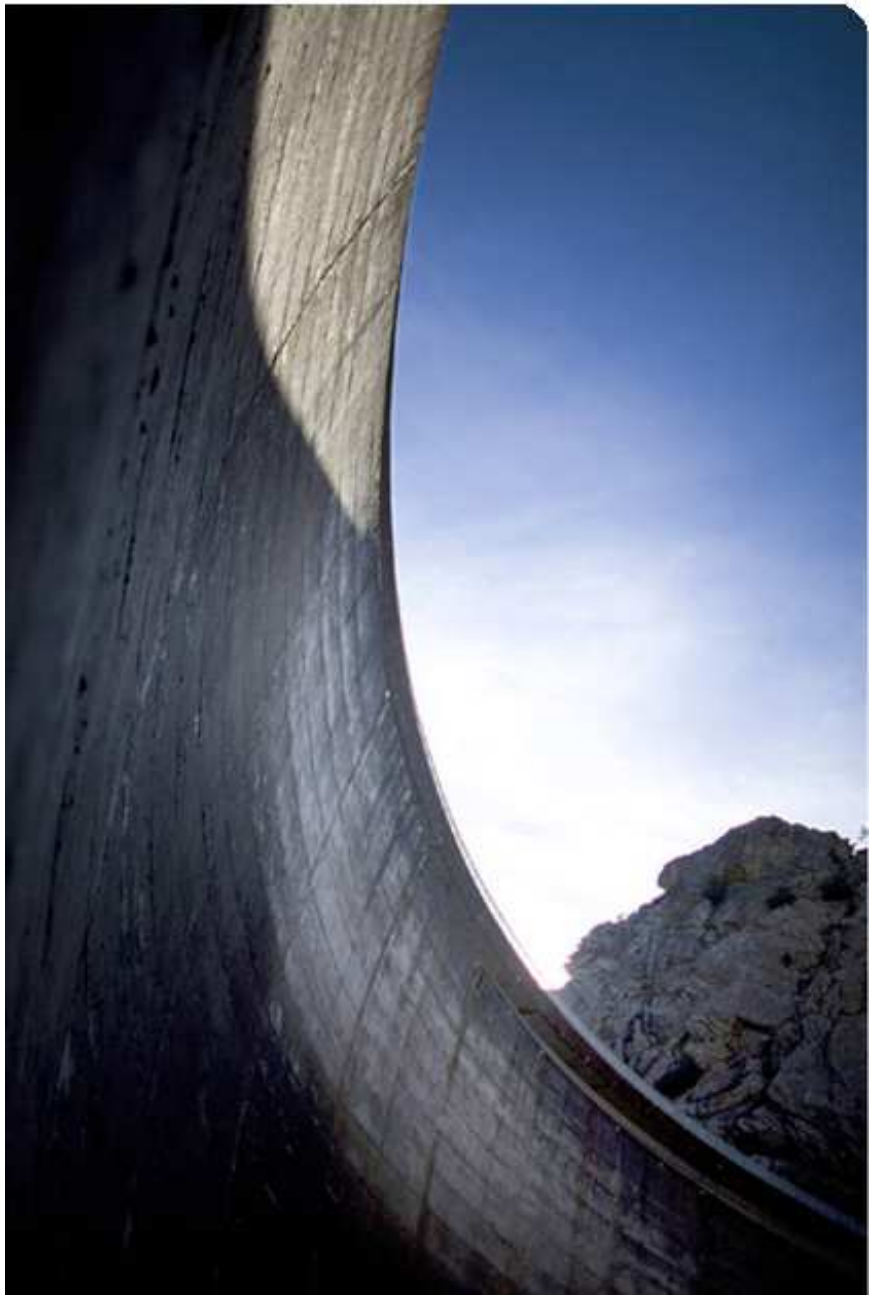
# *Code_Aster*, an opensource FEM solver

May 24th 2013



# TOC

1. Short overview (Code_Aster and Salome-Meca)
2. Focuses on capabilities
3. Focuses on Python
4. Open source and users communities
    - EDF motivations
    - Code_Aster community building



# TOC

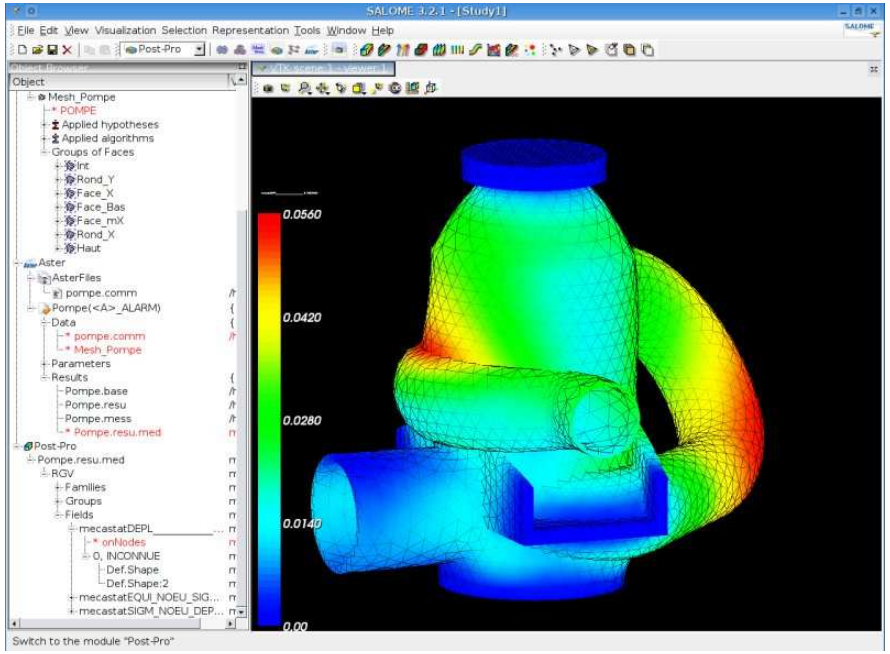
# Short overview

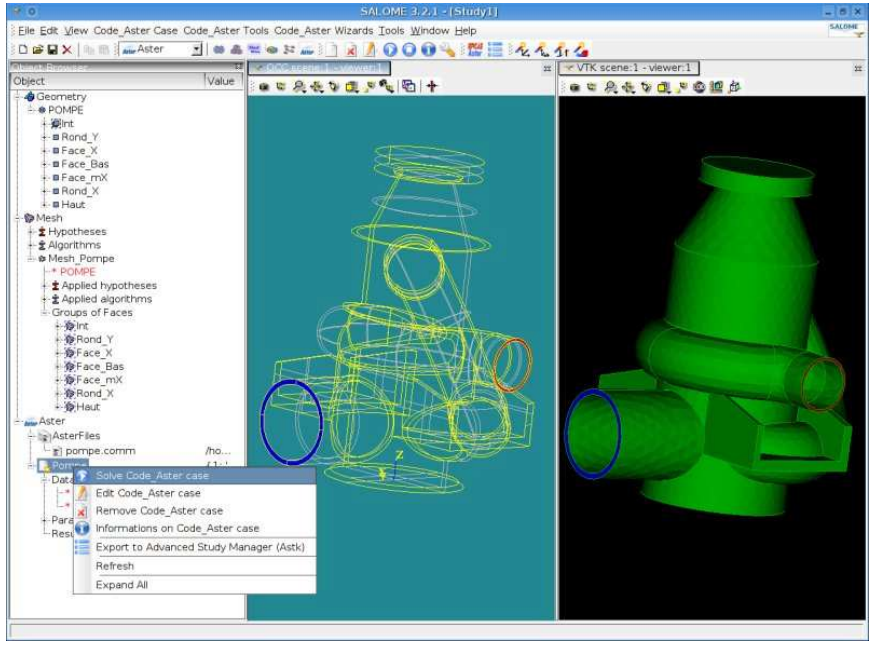
- **An all-purpose FEA simulation software for structural analysis :**

Code_Aster®

- **Plugged in an user-friendly environment :**

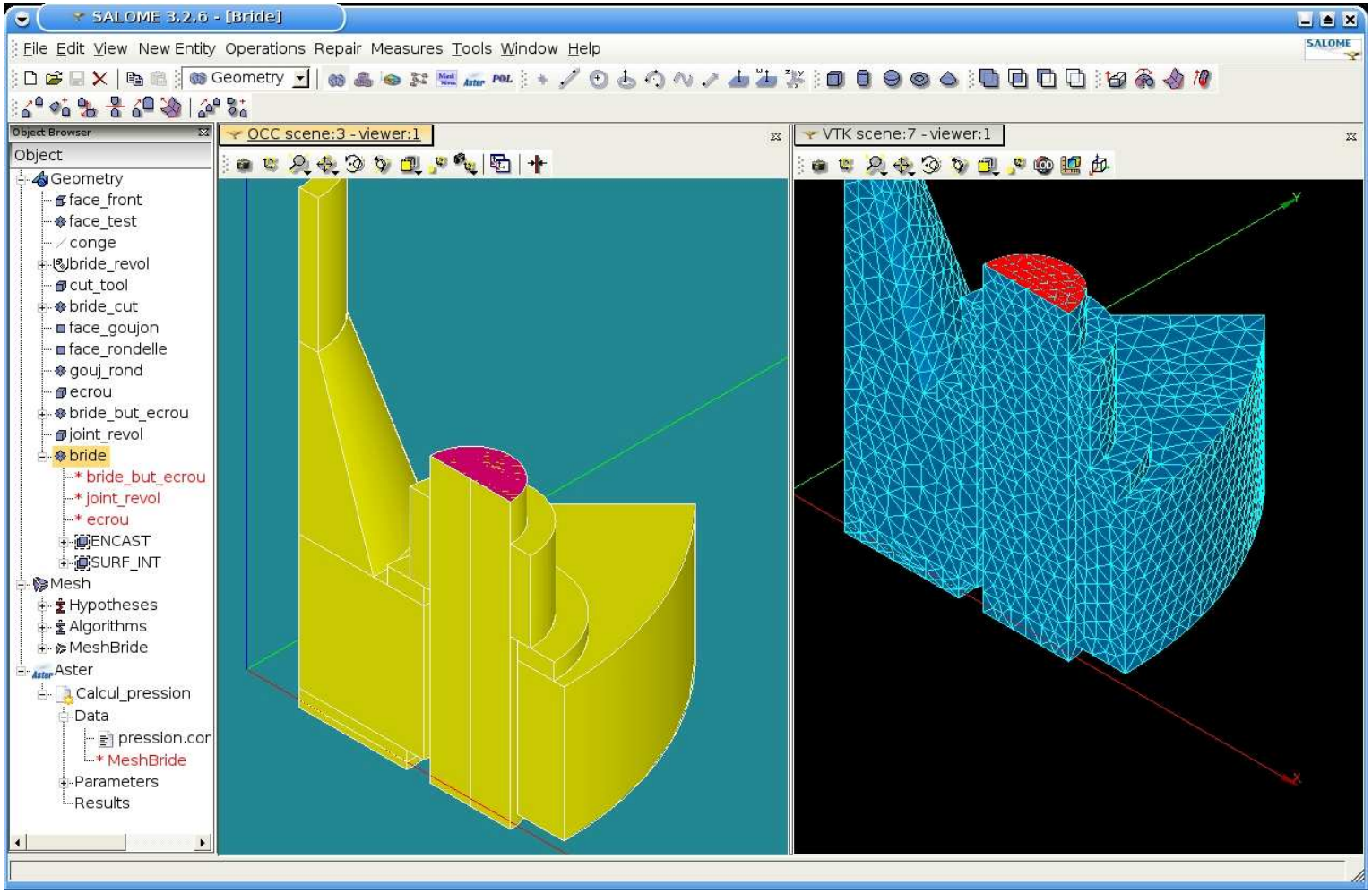
Salome-Meca 2013.1



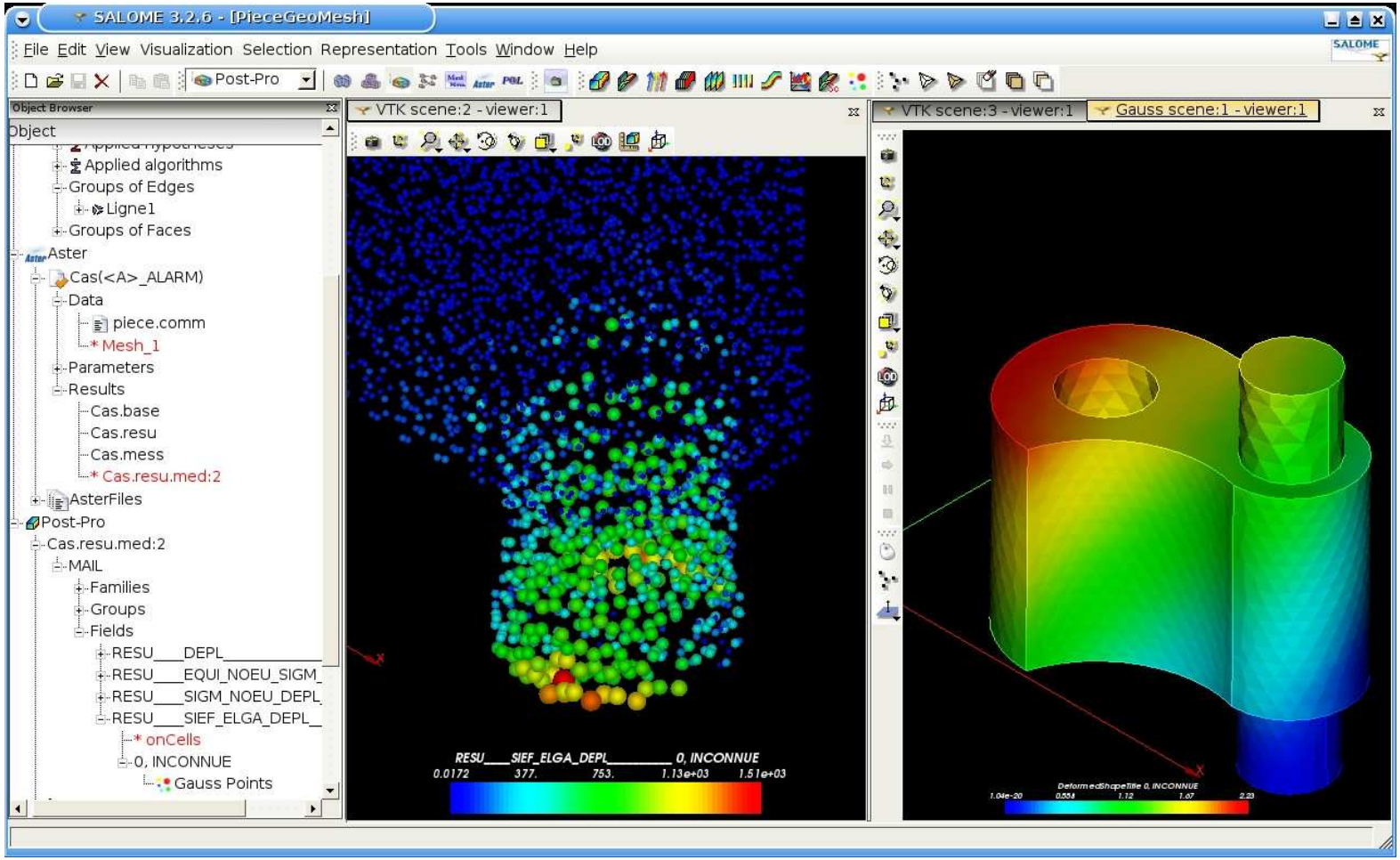
# Short overview : Salome

## Pre-processing environment



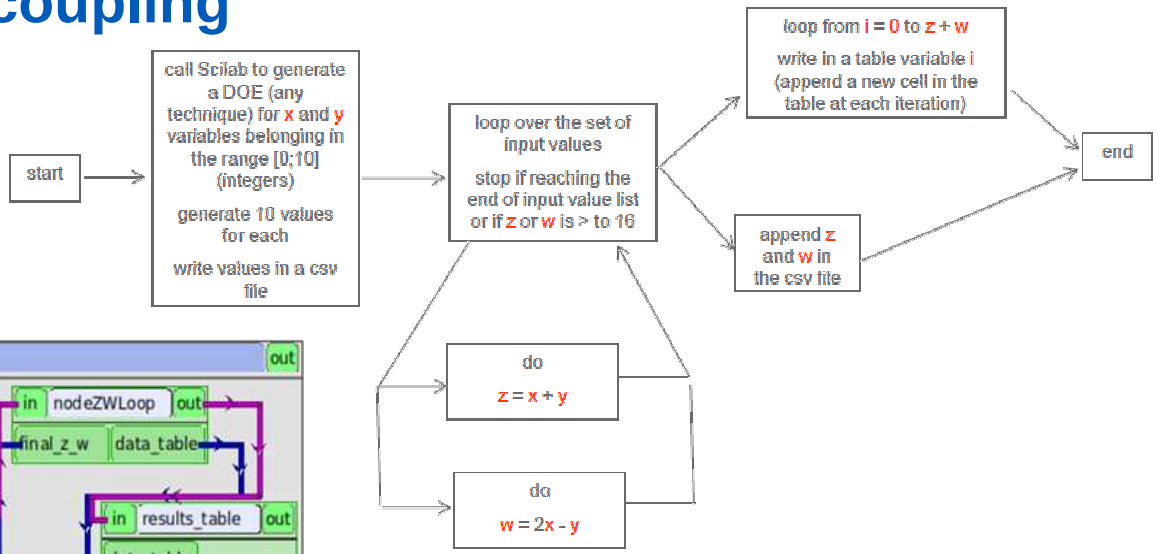
# Short overview : Salome

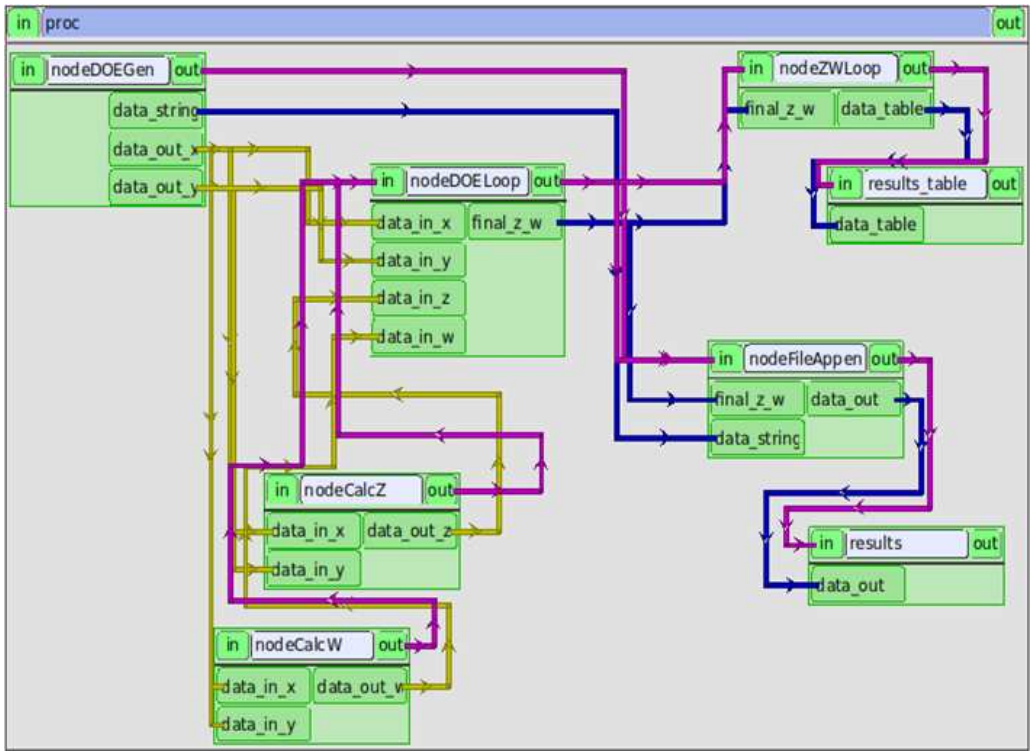
## Post-processing environment



# Short overview : Salome

## Supervision and code coupling



# Short overview : *Code_Aster*

## A large scope of capabilities

- Finite element / finite volume
- Thermics, acoustics, fluid (Darcy's law), mechanics
- Statics and dynamics, linear or not
- Modal analysis, harmonic and random response
- 400 finite elements : 3D, 2D, shells, beams, pipes …
- Interaction with other physics (coupled or not) : fluid, soil structure computation, electro-magnetism …

## Non linearities and specific features

- Contact, friction
- Large displacements, large deformations
- Behaviour models (> 100 constitutive laws)
- Porous media, fracture mechanics, damage, fatigue, welding, seismic analysis …

# Short overview : *Code_Aster*

## Quality insurance and nuclear safety

- ISO 9001 quality certified
- « Important to Safety » labelled by the French Nuclear Safety Authority
- Positive evaluation by the HSE Nuclear Directorate (General Design Agreement of the english EPR)

## Open source (GPL) software

- Freely available since 2001
- 50 000 downloads/year
- « Best french open-source project  for industry » award in 2006
- Development Partnership
  - IFP (French Oil Institute)
  - LaMCoS (Contact and Structural Mechanics Laboratory)

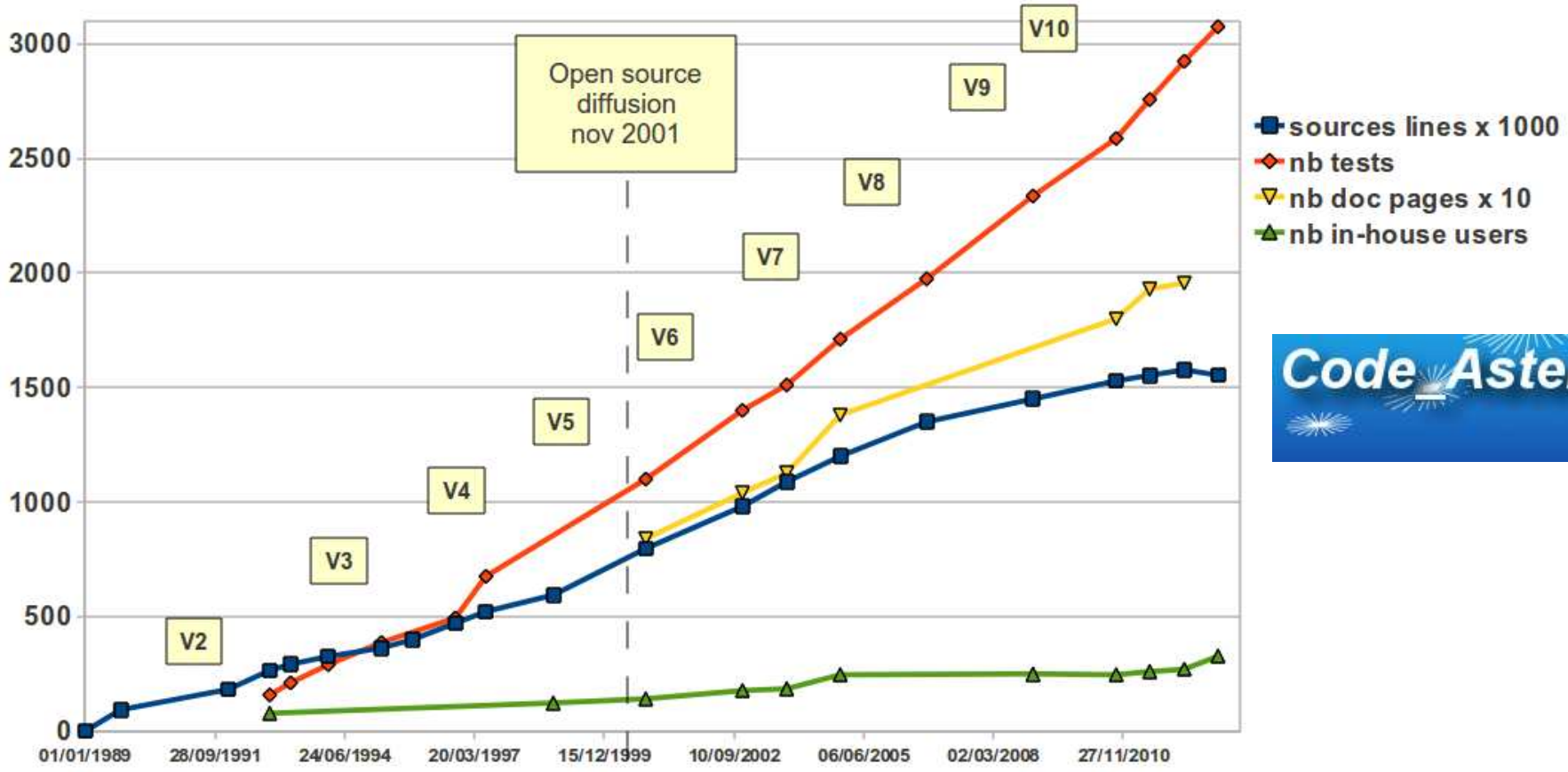
## English speaking community

- The whole *Code_Aster* documentation (19 000 pages) is automatically translated in english (Systran)
- Active community on the *Code_Aster* forum and the CAELinux wiki
- The course materials are freely available in english

# Short overview : *Code_Aster*

## A mature software : 22 years of development

- For version 11 : 62 developers, 20 project contributors, 743 improvements



# Short overview : why does EDF develop its own codes ?

## Specific simulation issues

- EDF **operates mechanical material**, it is not a manufacturing company : the challenges of understanding the physics and existing systems are predominant on the design optimization;

- **Specific physics** related to the operated materials (neutronic, free surface flows ...) and specific simulation issues (aging of materials and equipment, operational maintenance issues);

- Require the capability to conduct frontier studies;

- Involve **quick development capabilities** to answer engineering units' needs and reduce time to end-user

  ➔ tools accommodate both studies and research interest;

# The need for a comprehensive and coherent set of skills


**1- Model :**
*From physics to equations*

**2- Analyse and code :**
*Equations - algorithms - codes*

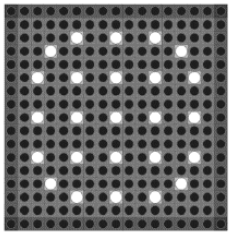
**3- Adapt to HPC architectures**

**4- Verification :**
*Code coverage, unit tests, simple models*

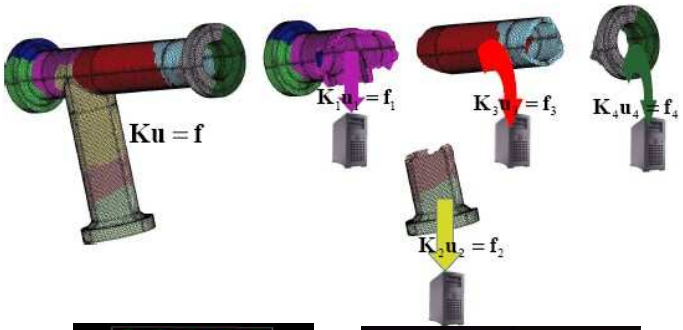
**5- Pre and post-processing :**
*Mesh, visualization*

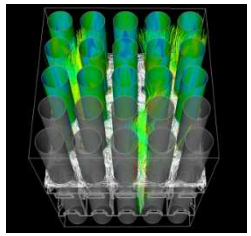
**6- Build studies how-to :**
*Probabilistic approaches, coupled physics*

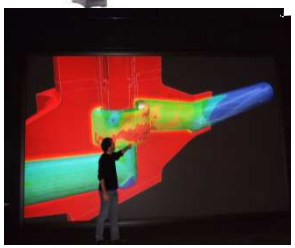
**7- Validation :**
*Benchmarks, scope of use, adequation to specific needs*

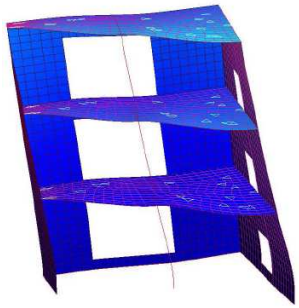


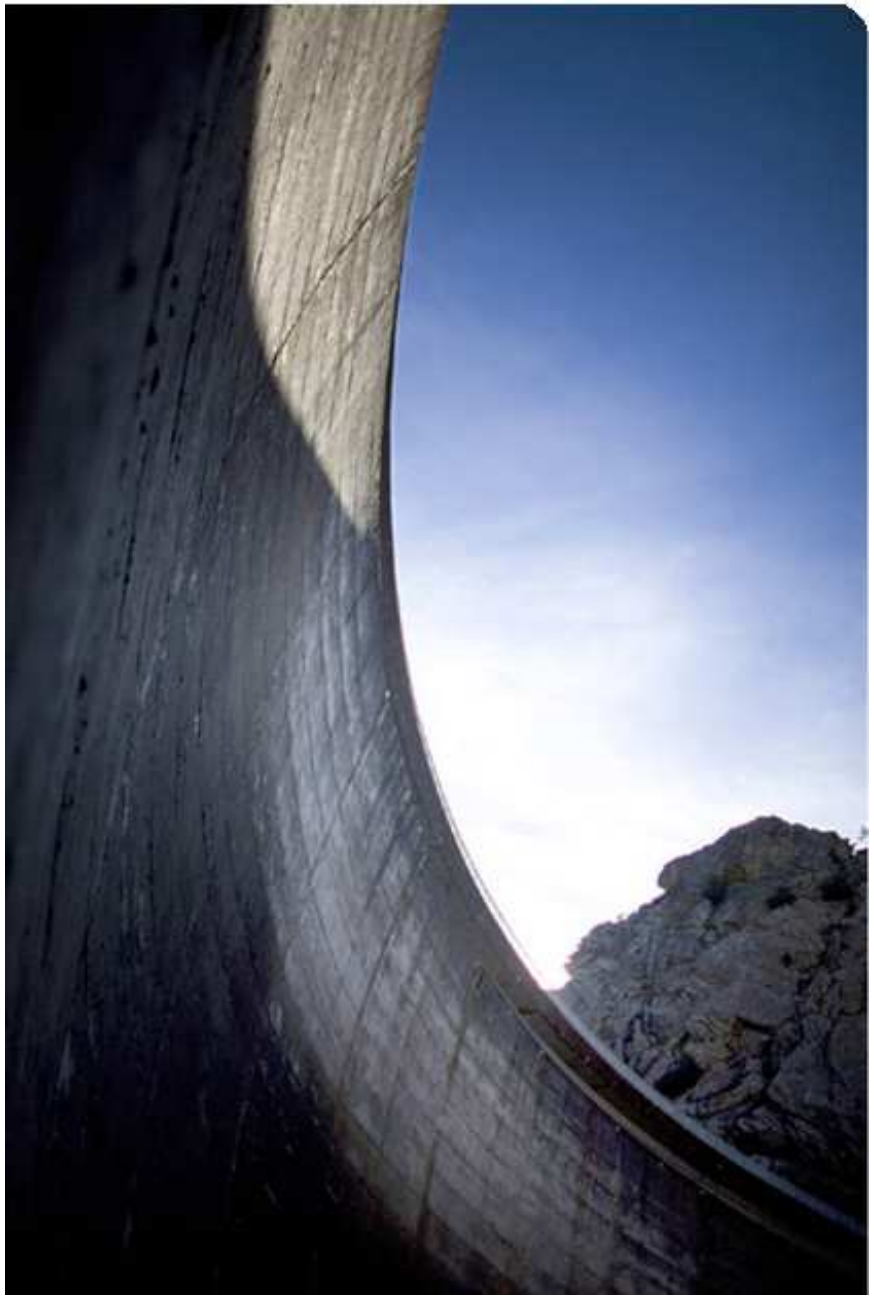
$$\frac{1}{v}\frac{\partial\phi(\vec{r},\vec{\Omega},E,t)}{\partial t} = -\left[\vec{\Omega}\cdot\vec{\nabla}+\Sigma(\vec{r},E)\right]\phi(\vec{r},\vec{\Omega},E,t)+ \frac{\chi(\vec{r},E)}{4\pi}\int dE'\nu\Sigma_f(\vec{r},E')\int d^2\Omega'\phi(\vec{r},\vec{\Omega}',E',t)+ \int d^2\Omega'\Sigma_s(\vec{r},\vec{\Omega}'\leftarrow\vec{\Omega},E\leftarrow E')\phi(\vec{r},\vec{\Omega}',E')+Q_e(\vec{r},\vec{\Omega},E,t)$$



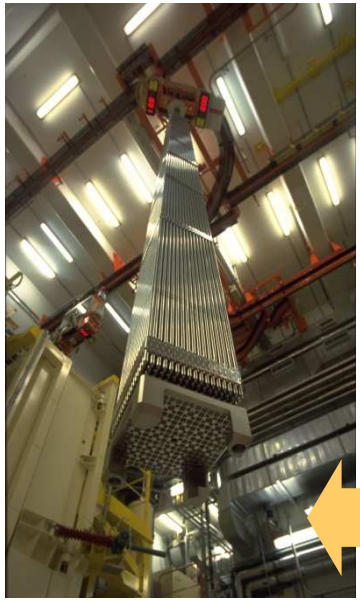


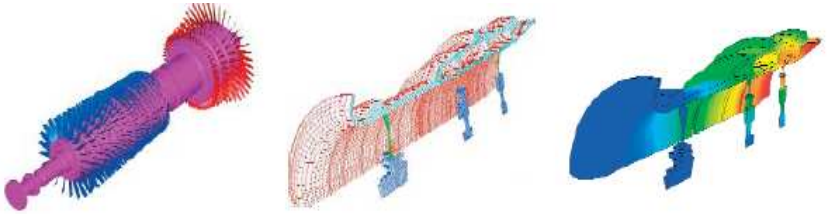




# TOC

# Focuses on capabilities (1/4)

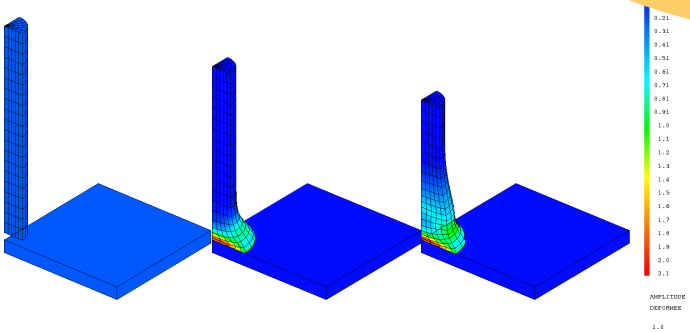


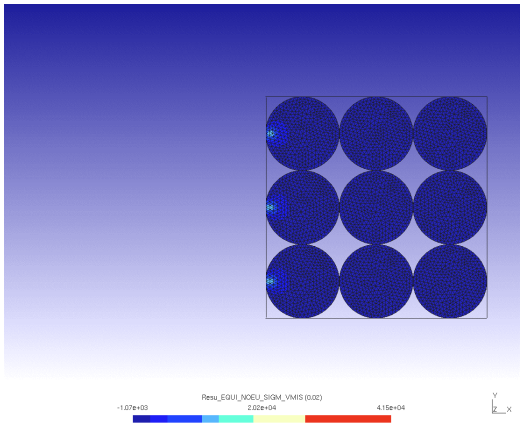


**Structural finite elements**

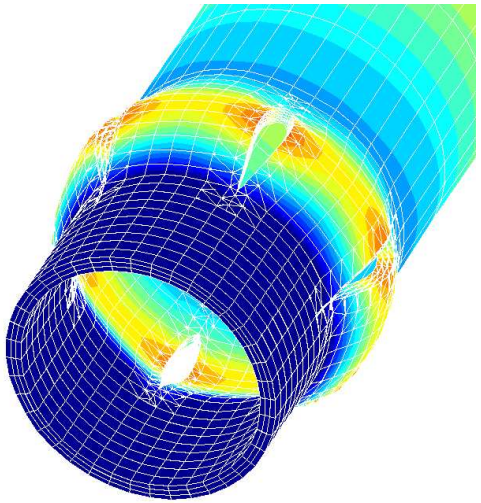
**Contact-friction**

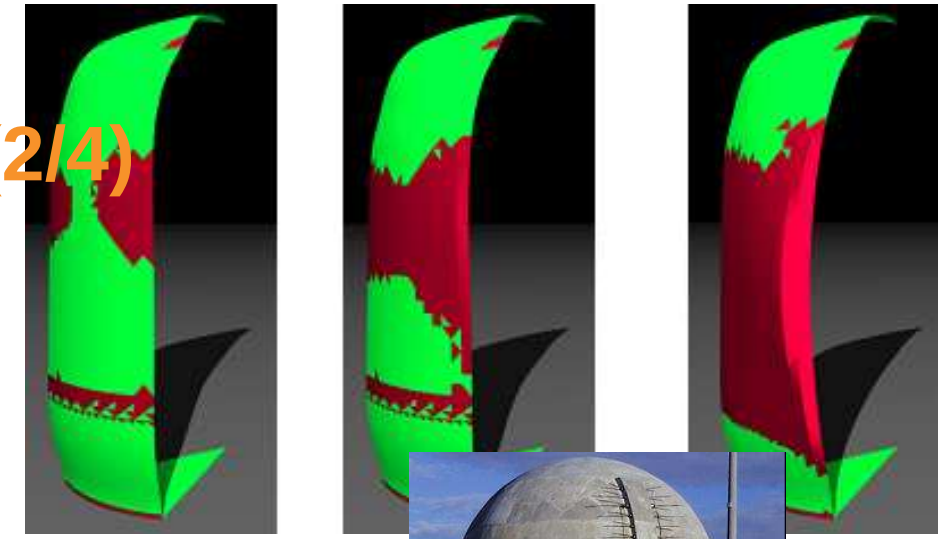
**Large deformations/rotations**

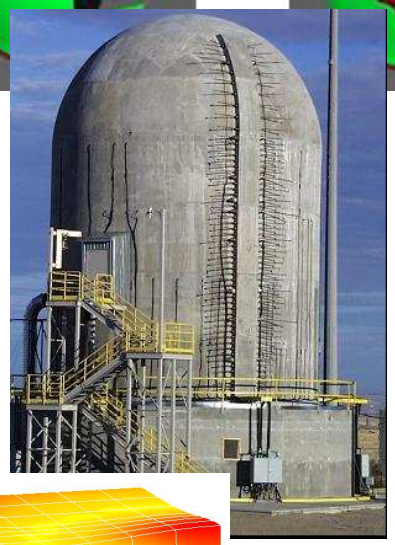




# Focuses on capabilities (2/4)



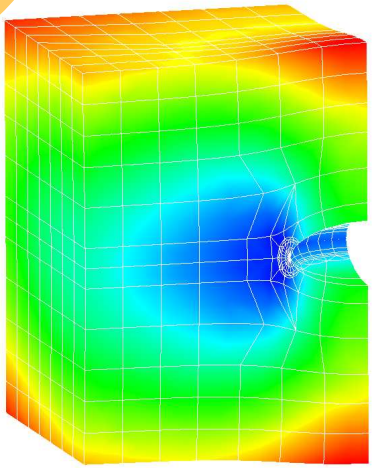


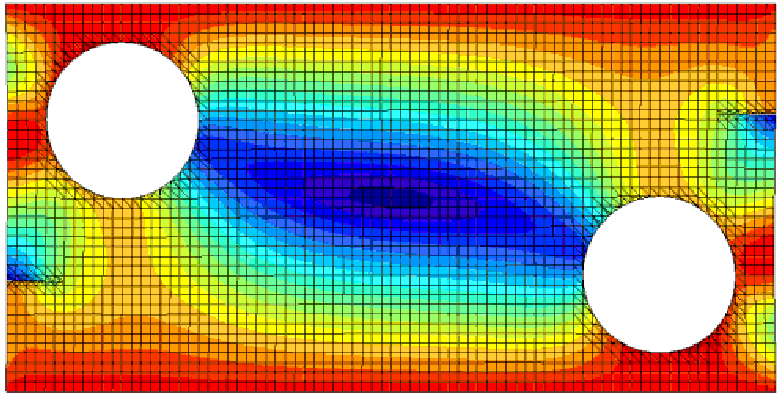


**Damage**

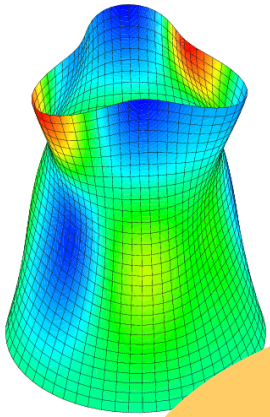
**Fracture**

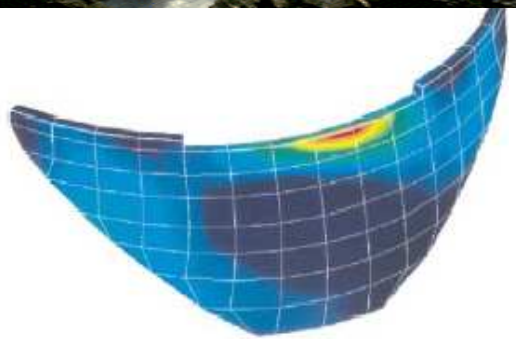
**Fatigue**





# Focuses on capabilities (3/4)



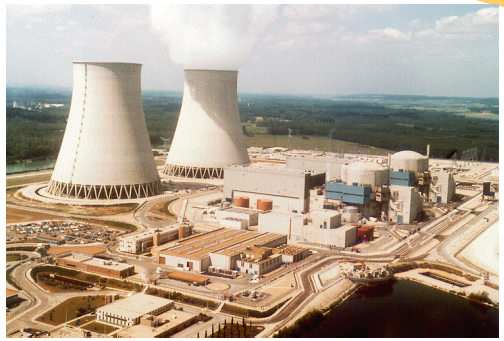


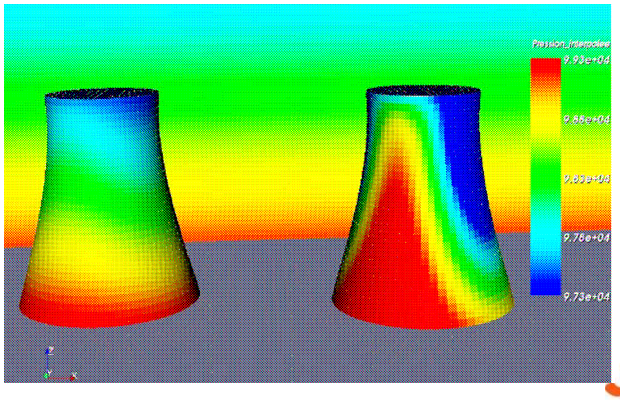
**Dynamics**

**Acoustics**

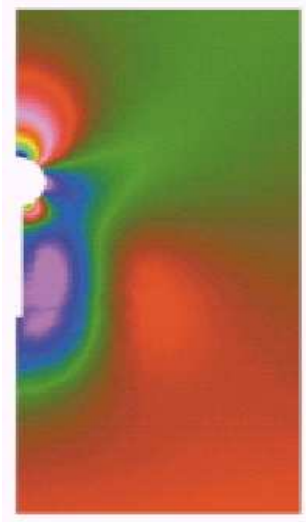
**Fluid forces**

**SFSI, SSI**





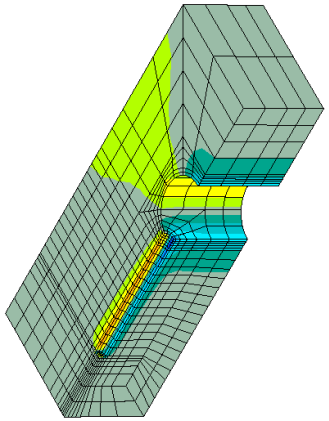
# Focuses on capabilities (4/4)

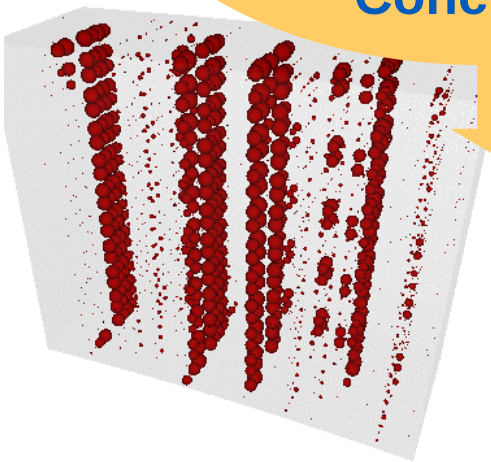


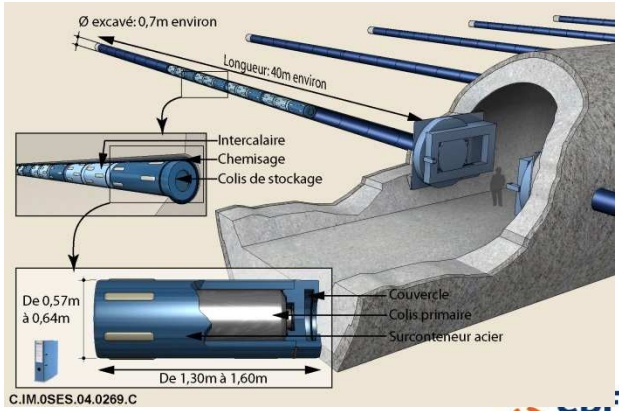
**Porous media and geomaterials**

**Civil engineering**

**Concrete**

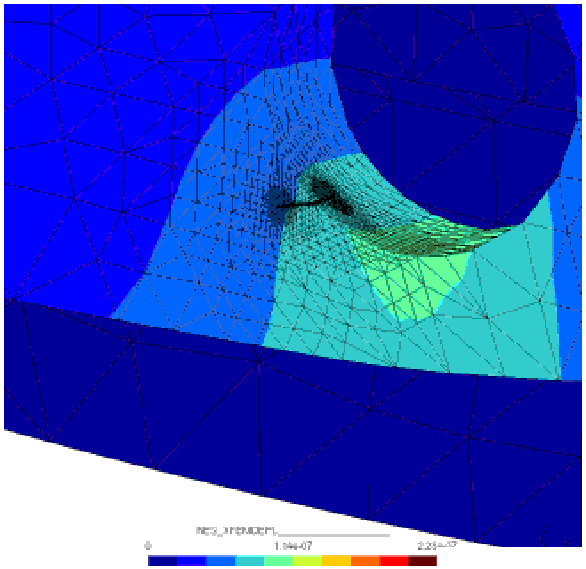


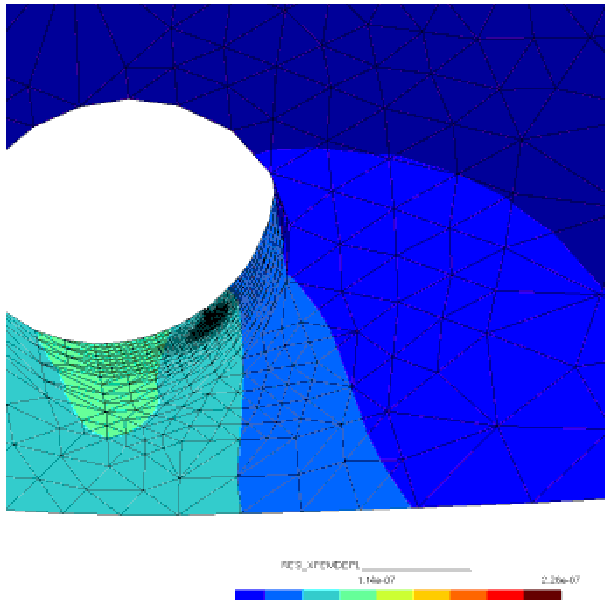


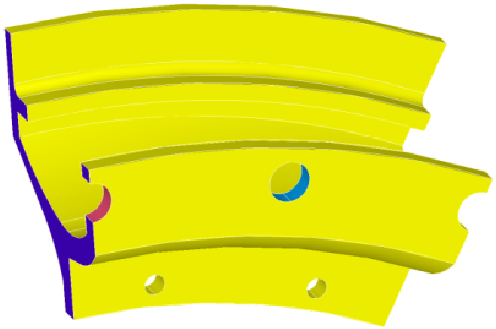


# Focuses on capabilities : samples

## 3D crack propagation : X-FEM and mesh adaptation

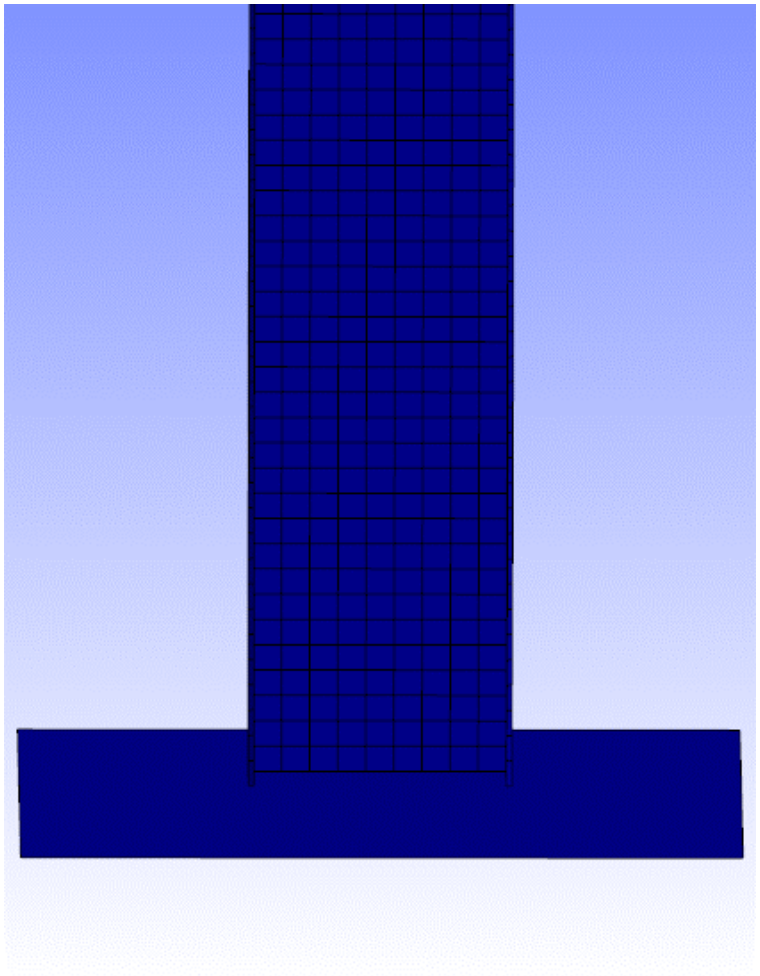


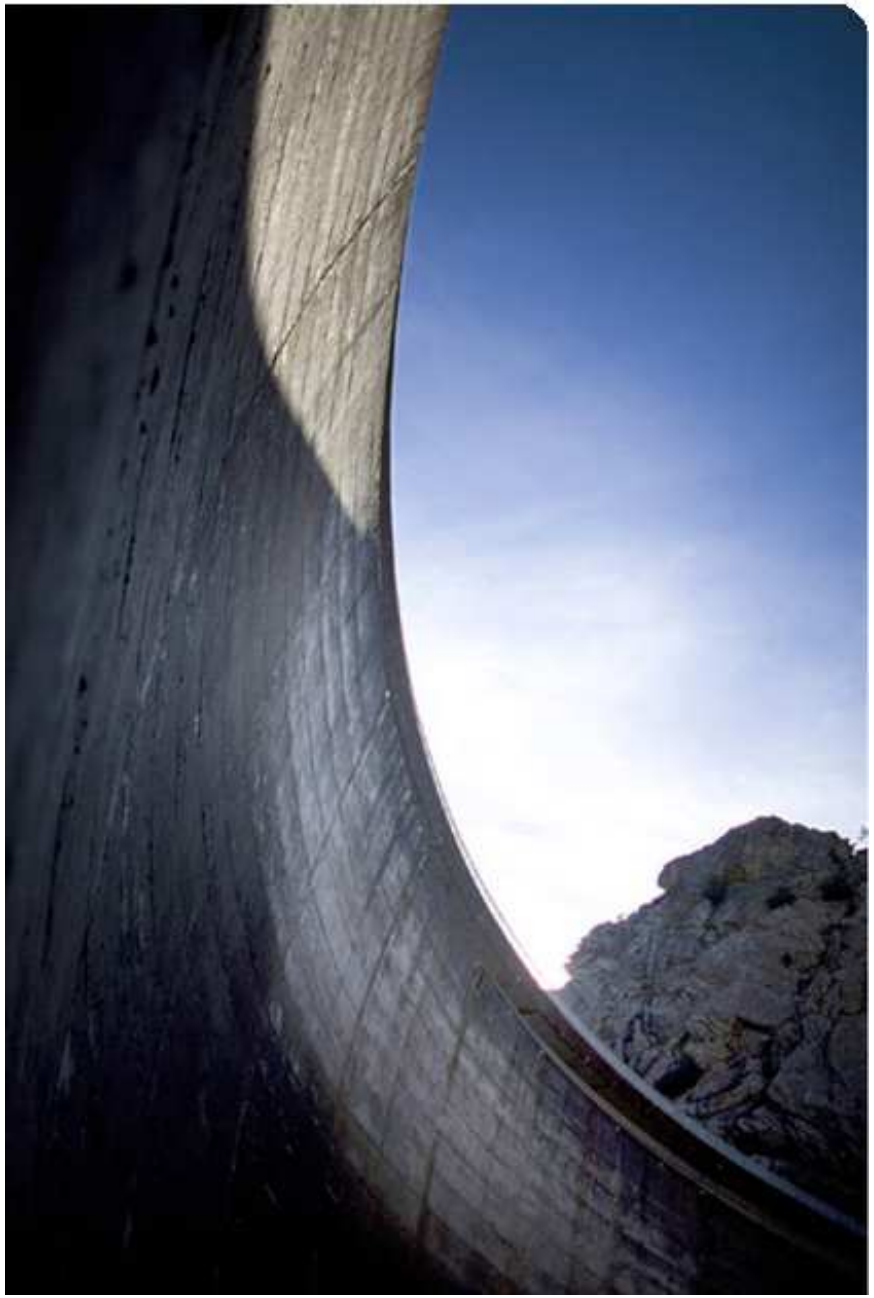




# Focuses on capabilities : samples

## A trebly nonlinear computation : contact, plasticity and large strains





# TOC

# Focuses on Python

## Python and *Code_Aster*

- The first scripting langage was developped as a standalone parser
- The switch to Python was done in 1999 (*Code_Aster* v6)
  - We tried to keep a similar syntax
  - Result is a Python compatible syntax that matches closely the former syntax
- Technical details
  - Use a legacy memory allocator written in Fortran that manages data structures as in C
  - This allocator needs to name the structure after the name of the variable on the left of the equal sign
  - Because of the way memory management is done, this script cannot be executed by a regular Python interpreter
- The switch to Python brought a couple of advantages
  - Parametric studies where made possible without redesigning a new language with its own loop constructs
  - It is now possible to write Aster macro-command in regular Python instead of writing everything in Fortran
  - Some pre or post processings are now executed in Python with the help of the Numpy module

# Focuses on Python : simple example

```
POURSUITE()

# plusieurs valeurs d'un mot-cle d'une commande
# sans python

G1=CALC_G(RESULTAT=RESU,
          COMP_ELAS=_F(RELATION='ELAS',),
          OPTION='CALC_G',
          THETA=_F(FOND_FISS=FF,
                   R_INF=2.,
                   R_SUP=4.,))

G2=CALC_G(RESULTAT=RESU,
          COMP_ELAS=_F(RELATION='ELAS',),
          OPTION='CALC_G',
          THETA=_F(FOND_FISS=FF,
                   R_INF=0.666.,
                   R_SUP=1.666.,))

G3=CALC_G(RESULTAT=RESU,
          COMP_ELAS=_F(RELATION='ELAS',),
          OPTION='CALC_G',
          THETA=_F(FOND_FISS=FF,
                   R_INF=1.,
                   R_SUP=2.,))

G4=CALC_G(RESULTAT=RESU,
          COMP_ELAS=_F(RELATION='ELAS',),
          OPTION='CALC_G',
          THETA=_F(FOND_FISS=FF,
                   R_INF=1.,
                   R_SUP=4.,))

IMPR_TABLE(TABLE=G1,);
IMPR_TABLE(TABLE=G2,);
IMPR_TABLE(TABLE=G3,);
IMPR_TABLE(TABLE=G4,);

FIN()
```

```
POURSUITE()

# plusieurs valeurs d'un mot-cle d'une commande
# avec python

RI=[2. , 0.666 , 1. , 1. ]
RS=[4. , 1.666 , 2. , 4. ]

nbc=len(RI)     # permet de recuperer la longueur de la liste RI
G = [None]*nbc # pour initialiser la liste de concepts

for i in range(0,nbc) :

   G[i]=CALC_G(RESULTAT=RESU,
               COMP_ELAS=_F(RELATION='ELAS',),
               OPTION='CALC_G',
               THETA=_F(FOND_FISS=FF,
                        R_INF=RI[i],
                        R_SUP=RS[i],))

   IMPR_TABLE(TABLE=G[i],);

FIN()
```

Tables created are : `G_0`, `G_1`, `G_2` and `G_3`

# Focuses on Python : definition, evaluation

## Define a formula using a Python function

```
def heaviside(x):
   if x < 0.:
      return 0.
   else:
      return 1.
H = FORMULE(VALE='heaviside(X)', NOM_PARA='X')
```

## Evaluate a function or formula

- Example with a function

```
f   = DEFI_FONCTION(VALE=( 0.,2.e11,
                           20.,2.5e11),
                    NOM_PARA='INST')
mat = DEFI_MATERIAU(E=f(15.), ...)
print f(15.) # returns 2.375e11
```

- Another example

```
g   = FORMULE(VALE='Y**2+ X',NOM_PARA=('X','Y',))
print g(1., 2.) # returns 5
print H(-2),heaviside(6.) # returns 0., 1.
```

# Focuses on Python : acces to values

## Content of a list concept or function

- Example with a list :

```
LIST = DEFI_LIST_REEL(DEBUT=0.,
                      INTERVALLE=_F(JUSQU_A = 10.,NOMBRE = 2,))
lst = LIST.Valeurs()
print lst # returns [0.0, 5.0, 10.0]
```

- Example with a function :

```
f   = DEFI_FONCTION(VALE=( 0.,2.e11,
                           20.,2.5e11),
                    NOM_PARA='INST')
temp = f.Absc() # temp contains [0., 20.]
val = f.Ordo()  # val contains [2.e11, 2.5e11]
list_x,list_y = f.Valeurs()
```

## Content of a table

- Conversion in Table object : **`tab = TABRES.EXTR_TABLE()`**
- Sorting methods, extraction filters, printing : **`help(tab)`**
- Several spreadsheet-like manipulations available through **`CALC_TABLE`**

# Focuses on Python : advanced usage

## Advanced mathematical calculations

Using numpy Python module :

- Advanced math functions
- manipulation of arrays
- Used in `CALC_FONCTION`, `INFO_FONCTION` ...
- Always available (pre-requisite for Aster)

## Statistical calculations, signal processing

- Possibilities through the use of third-party modules

## Graphics

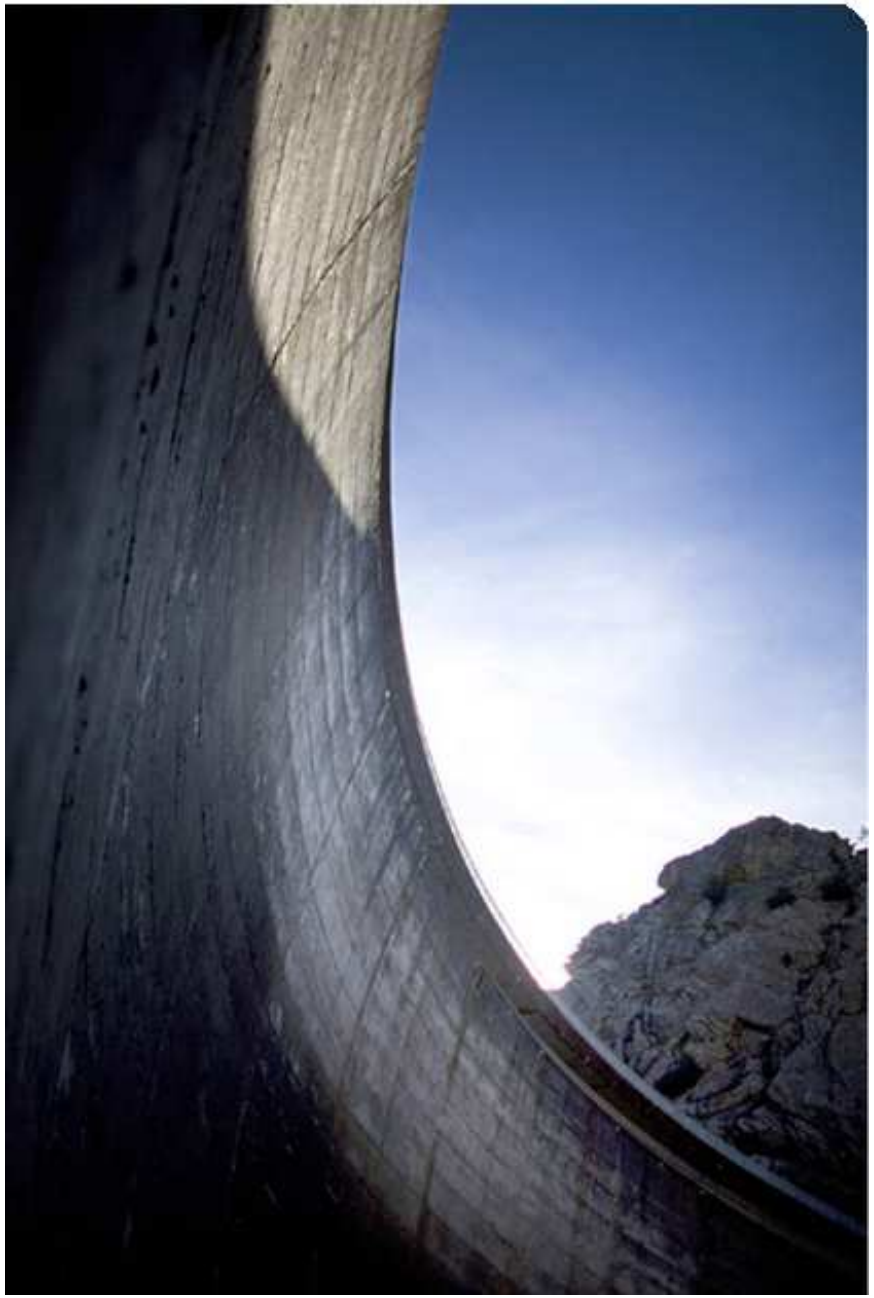
- Interactive plots, visualization
- Enable GUI production : software tools
- Can use `EXEC_LOGICIEL` for calling external tools (including mesher)

# Focuses on Python : access to content of Aster concepts

- **Access methods for datastructures**
  - Meshes
    - List of mesh groups,
      Python objects  corresponding to a mesh
  - Fields
    - EXTR_COMP method
  - Every Fortran / Jeveux object
    - getvectjev, getcolljev (+ put method for modifying concepts)
    - Res = aster.getvectjev("MA  .COORDO  .VALE  ")

- **Advanced use mainly dedicated to high level Aster macro commands rather than user command file**

- **Beware of memory consumption !**



# TOC

# Open Source distribution : EDF's motivations

## Improve codes :

- By the multiplication of users

Code_Aster

250 internal users at EDF

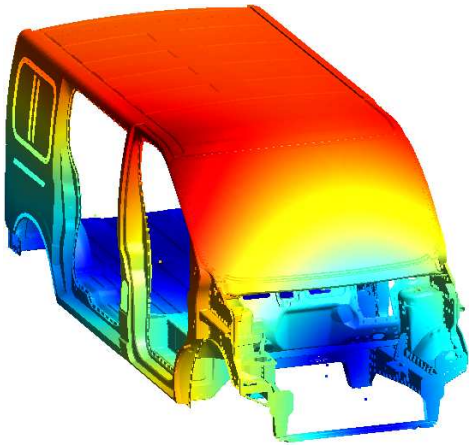
2000 downloads of each release
+ uncontrolled diffusion

- By user feedback (when they provide it !) through :
  - benchmarks
  - validation
  - bug reports (code as well as documentation)
  - validity limits of models

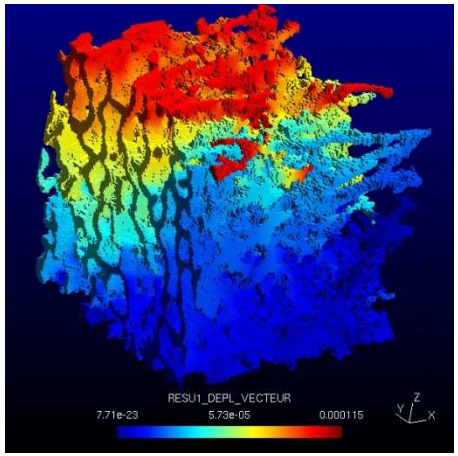
# Open Source distribution : EDF's motivations

## Improve codes :

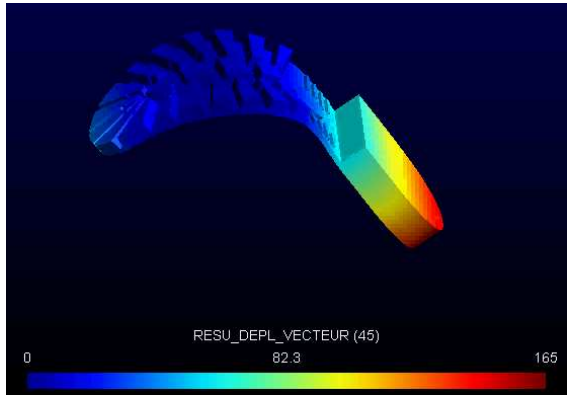
- By extending the range of validity or opening new simulation fields



ArcelorMittal
Body in white of light commercial vehicule




University of Sherbrooke (Canada)
Bone vertebra with "medical glue"




Centre technique du cuir / Université Lyon 1
Hyperelastic simulation of an elastomeric sole

***Code_Aster* simulations by the Open Source user community**

# Open Source distribution : EDF's motivations

## Recognition by adoption :

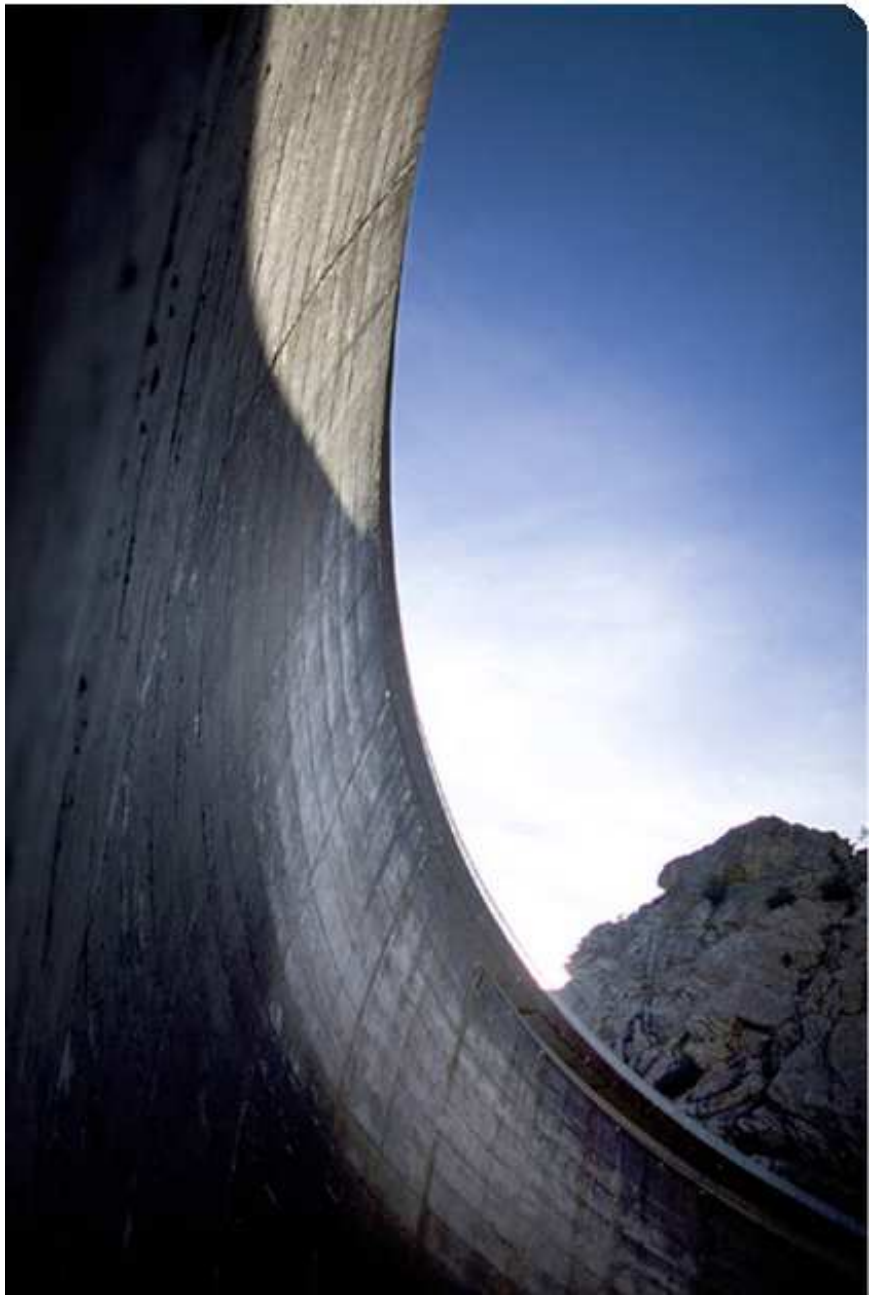
- **Asset valuation and peer recognition**
  - For a software, distributing a code as open-source is the equivalent of a publication in a peer-reviewed journal
  - Facilitation of “third party review"
  - Significant communication and reputation vector for EDF R&D
  - Key motivator of the core-team (generation X and Y)
- **Process transparency** and social responsibility
  - OS diffusion considered favorably by the English Safety Authority
  - Dissemination of the code and research work
- **Facilitate the dissemination** and acceptance of methods and models
  - Goal : to become a standard, at least a reference
- **An open-source base for cooperations**
  - Developments at shared costs
  - Easier capitalization for Phd works and research
  - Software base for industrial and scientific partnerships

# Open Source distribution: EDF's motivations

## Dissemination of knowledge and skills :

- Through **education** and **research**
  - Transfer of research to industry facilited (no more POC in Matlab that can't be used in industrial environment)
  - Having a pool of students and graduates already trained at our tools

- Create an **eco-system of skills among** our services providers and partners



# TOC

# Community building: 3 stages

## 2001 - 2005: the emergence

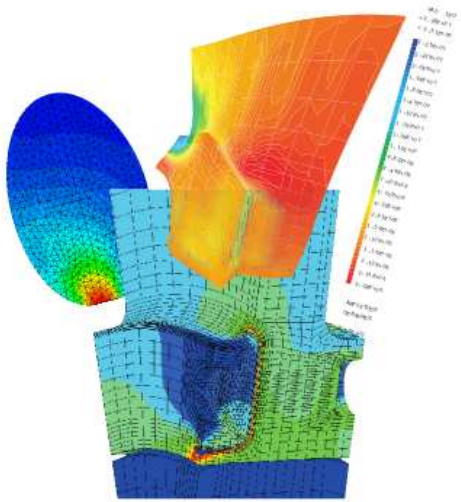
19 octobre 2001

- Following a failed attempt of commercial distribution

- 2001: Open Source distribution of the version 6 under the **GNU-GPL**
- 2003: first public event or “User Day"

Code_Aster Libre

- Voluntary investment (and passionate) of 6/7 developers from the core-team who “moderate" the forum
- One-way exchanges : much assistance, little feedback
- With few exceptions (IFP LAEGO ...), few expressions of interest from institutional (companies, laboratories), but some interest by individuals (often anonymous)



EDF

# Community building: 3 stages

## 2006 – 2011 : consolidation

- 2005 : 10,000th post on the forum, >500 active members
- 2006 : Code_Aster receives French "Golden Lutèce " award for the best free project in a large company

- 2007 : First partnership agreement with the French Oil Institute

EDF + IFP Energies nouvelles

- 2009 : Code_Aster User Day for Geosciences
- 2010 : 28,000th message on the forum, >800 active members, 26 messages per day, 40 bug reported per year, 2000 downloads per release
- 2011 : more and more forum posts are in english. Documentation is available in English

# Community building: 3 stages

## 2011 and later : towards a convergence of interests around the code

- Creation of ***Code_Aster*-ProNet** (professional network) :
  - Create multilateral exchanges between institutions, overcomes the limits of a public and anonymous forum;
  - Increase the visibility of members on their use and their work;
  - Group and structure demands to service providers;
  - Leverage opportunities for cooperative development.

**. . . + and academic services (already mentioned)**

# Comments

## “free” implies accepting the advice of others

- Expose themselves to the possibility of judgment, sometimes unfair, sometimes cruel
- Ensure e-reputation

## “free” implies an obligation of reactivity

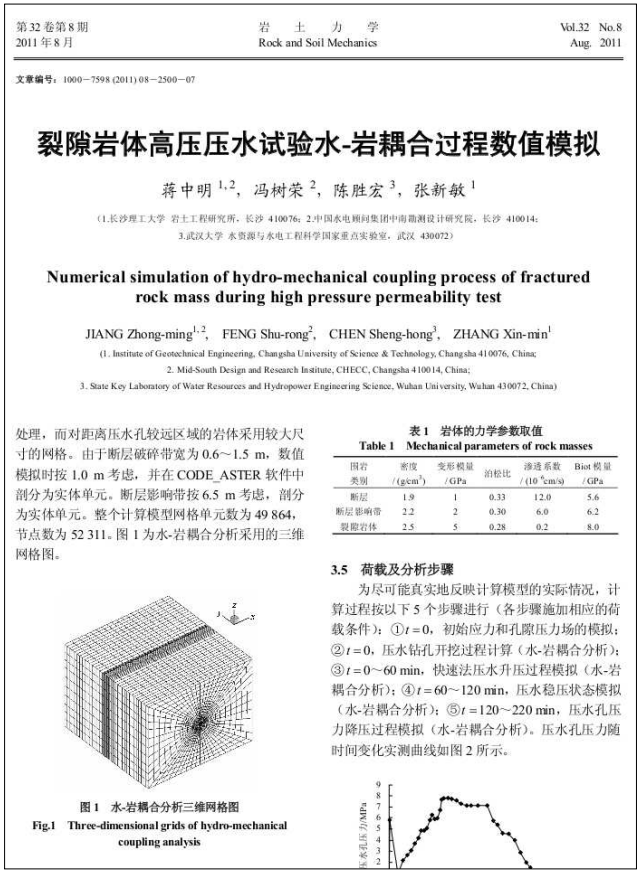
- Find the right balance (*all posts do not deserve a response*)
- Be open, listening to good ideas (*serendipity*)
- And when they emerge, be able to implement (*agile development methods*)

## “free” implies allocating resources

- To communicate (*Newsletter, interventions on the forum*)
- To navigate through the forum posts and initiatives on the net
- To maintain the relationship with those who are not involved in the forum (*ProNet*)
- To consider development that deserve recognition
    - But with a lot of profit in return in case of success (*validation by others, core-team motivation, various feedback, collaboration ...*)

# Comments

## ➧ We must accept that things escape you ... a little



第32卷第8期 岩 土 力 学 Vol.32 No.8
2011年8月 Rock and Soil Mechanics Aug. 2011

文章编号：1000－7598 (2011) 08－2500－07

裂隙岩体高压压水试验水-岩耦合过程数值模拟

蒋中明[1,2]，冯树荣[2]，陈胜宏[3]，张新敏[1]

（1.长沙理工大学 岩土工程研究所，长沙 410076；2.中国水电顾问集团中南勘测设计研究院，长沙 410014；
3.武汉大学 水资源与水电工程科学国家重点实验室，武汉 430072）

Numerical simulation of hydro-mechanical coupling process of fractured rock mass during high pressure permeability test

JIANG Zhong-ming[1,2], FENG Shu-rong[2], CHEN Sheng-hong[3], ZHANG Xin-min[1]

(1. Institute of Geotechnical Engineering, Changsha University of Science & Technology, Changsha 410076, China;
2. Mid-South Design and Research Institute, CHECC, Changsha 410014, China;
3. State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan 430072, China)

处理，而对距离压水孔较远区域的岩体采用较大尺寸的网格。由于断层破碎带宽为0.6～1.5 m，数值模拟时按1.0 m考虑，并在CODE_ASTER软件中剖分为实体单元。断层影响带按6.5 m考虑，剖分为实体单元。整个计算模型网格单元数为49 864，节点数为52 311。图1为水-岩耦合分析采用的三维网格图。

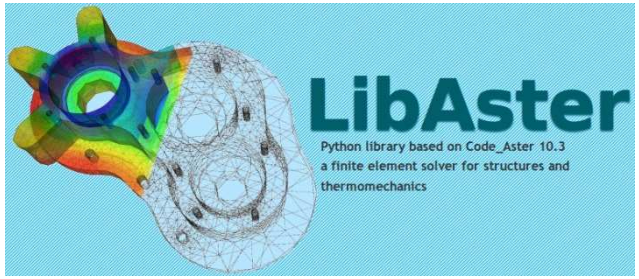
表1 岩体的力学参数取值
Table 1 Mechanical parameters of rock masses

| 围岩类别 | 密度 /(g/cm³) | 变形模量 /GPa | 泊松比 | 渗透系数 /(10⁻⁸cm/s) | Biot模量 /GPa |
|---|---|---|---|---|---|
| 断层 | 1.9 | 1 | 0.33 | 12.0 | 5.6 |
| 断层影响带 | 2.2 | 2 | 0.30 | 6.0 | 6.2 |
| 裂隙岩体 | 2.5 | 5 | 0.28 | 0.2 | 8.0 |

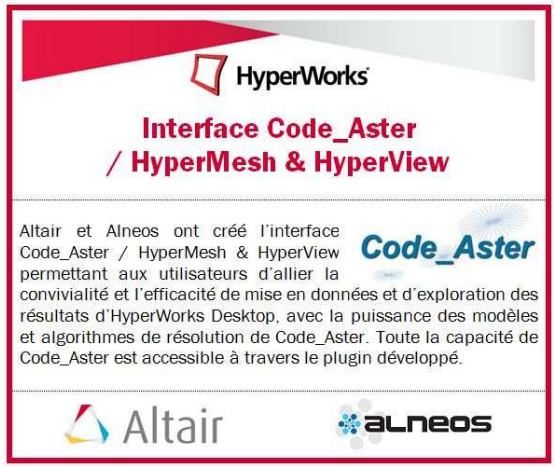
3.5 荷载及分析步骤

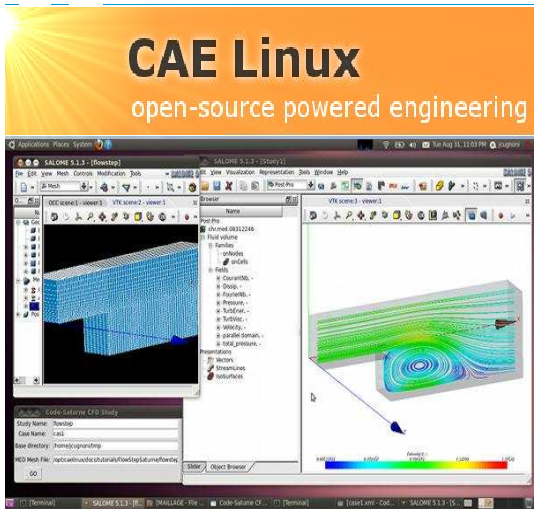
为尽可能真实地反映计算模型的实际情况，计算过程按以下5个步骤进行（各步骤施加相应的荷载条件）：①$t=0$，初始应力和孔隙压力场的模拟；②$t=0$，压水钻孔开挖过程计算（水-岩耦合分析）；③$t=0$～60 min，快速法压水升压过程模拟（水-岩耦合分析）；④$t=60$～120 min，压水稳压状态模拟（水-岩耦合分析）；⑤$t=120$～220 min，压水孔压力降压过程模拟（水-岩耦合分析）。压水孔压力随时间变化实测曲线如图2所示。

图1 水-岩耦合分析三维网格图
Fig.1 Three-dimensional grids of hydro-mechanical coupling analysis







Merci


MERCI