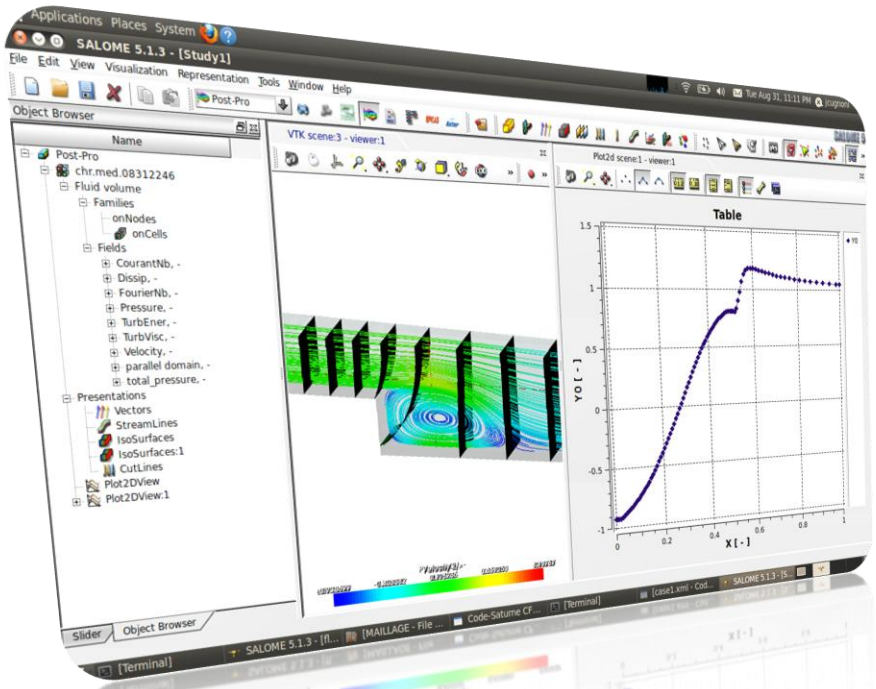
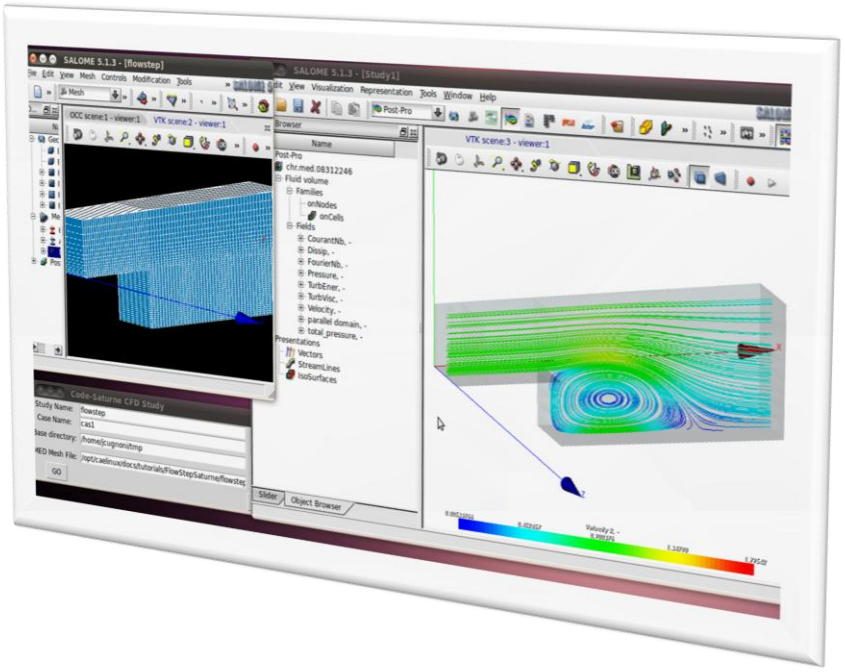
CAELinux : an open source engineering platform

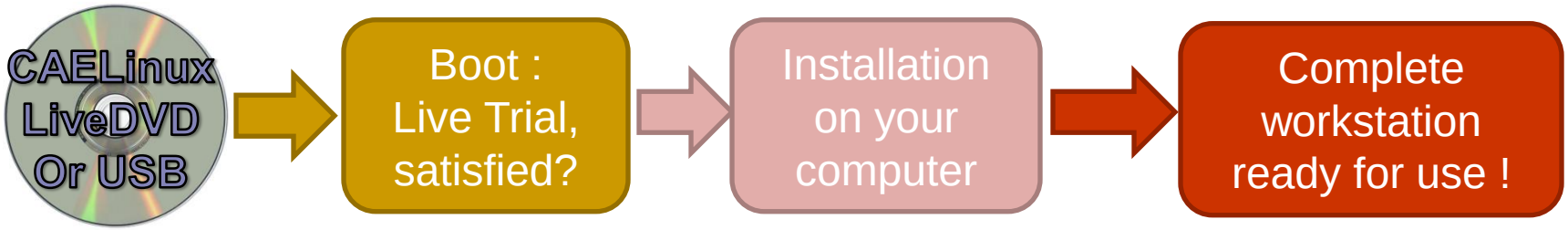




Joël Cugnoni, www.caelinux.com

What is CAELinux ? A CAE workstation on a disk

- CAELinux in brief
  - CAELinux is a « Live» Linux distribution pre-packaged with the main open source Computer Aided Engineering software available today.
  - CAELinux is free and open source, for all usage, even commercial (*)
  - It is based on Ubuntu LTS (12.04 64bit for CAELinux 2013)
  - It covers all phases of product development: from mathematics, CAD, stress / thermal / fluid analysis, electronics to CAM and 3D printing
- How to use CAELinux:



Or

CAELinux virtual Machine installation in OSX, Windows or other Linux

Running a server in Amazon EC2 cloud computing (on demand, charge per hour)

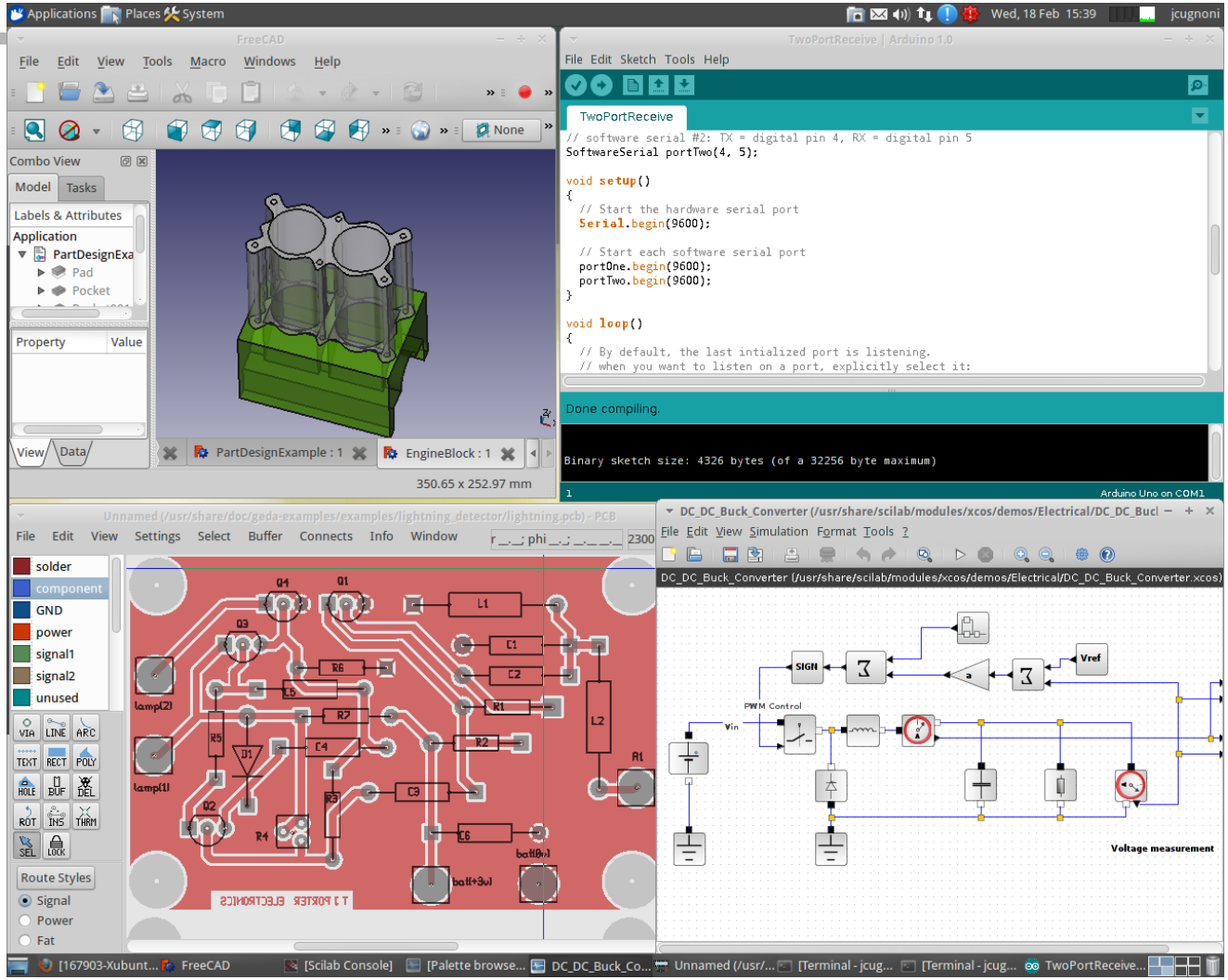
amazon web services™ | EC2

(* except for Tetgen mesher)

CAELinux: History and present

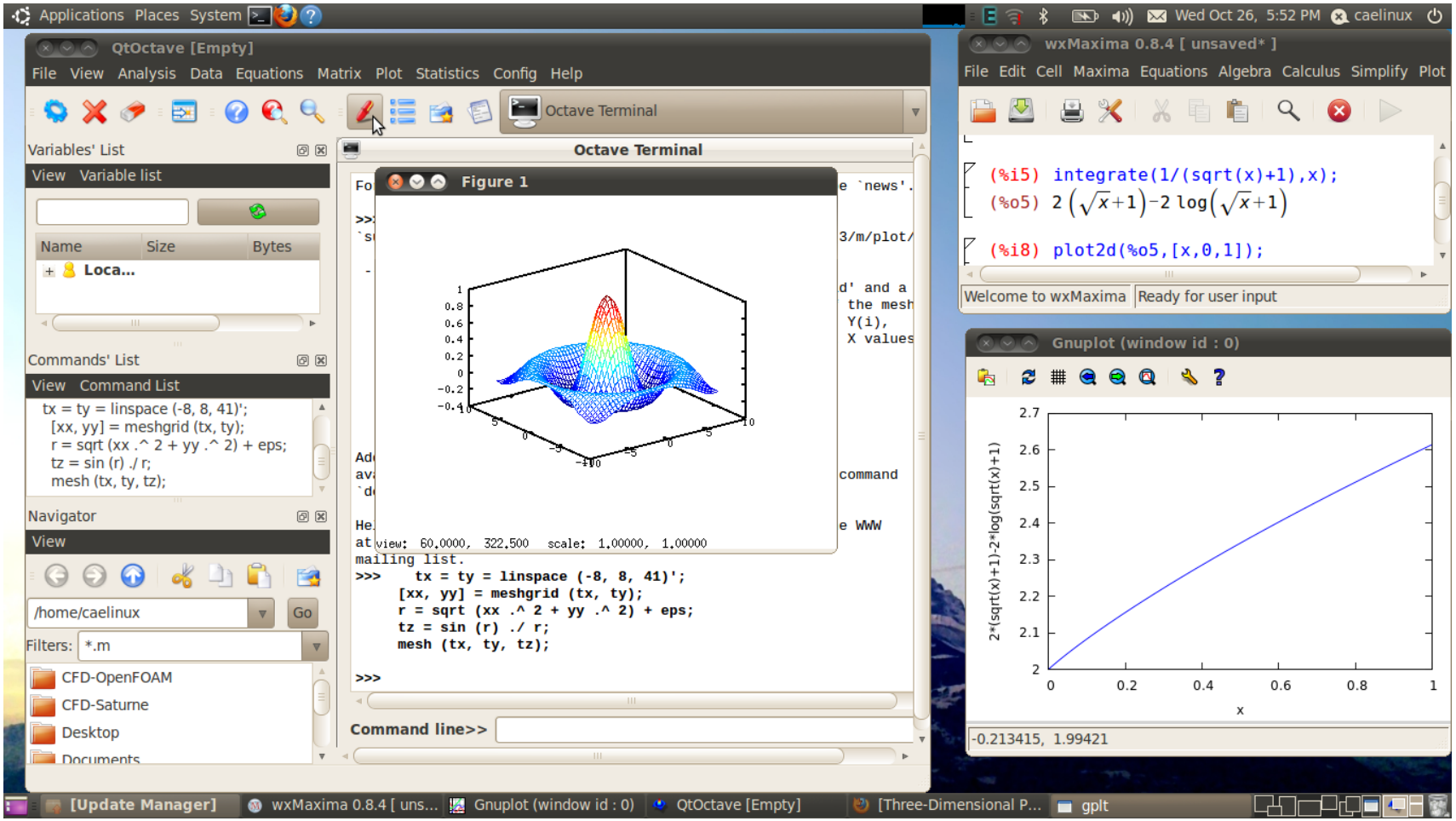
- Past and present:
  - CAELinux started in 2005 as a personal project for my own use
  - Motivation was to promote the use of scientific open source software in engineering by avoiding the complexities of code compilation and configuration. And also, I wanted to have a reference installation of Code-Aster and Salome that I could install for my own use.
  - Until now, 11 versions have been released in ~9 years. One release per year (except 2014).
  - Today, the latest version, CAELinux 2013, has reached 63’000 downloads in 1 year on sourceforge.net.
  - CAELinux is used for teaching in universities, in SME’s for analysis and by many occasional users, hobbyists, hackers and Linux enthusiasts.
  - The main distribution is still developed by myself on my free time which explains the slow updates.
  - A community has formed around CAELinux with many contributors participating in the documentation, tutorials and support on forums and wiki

What can you do with it?



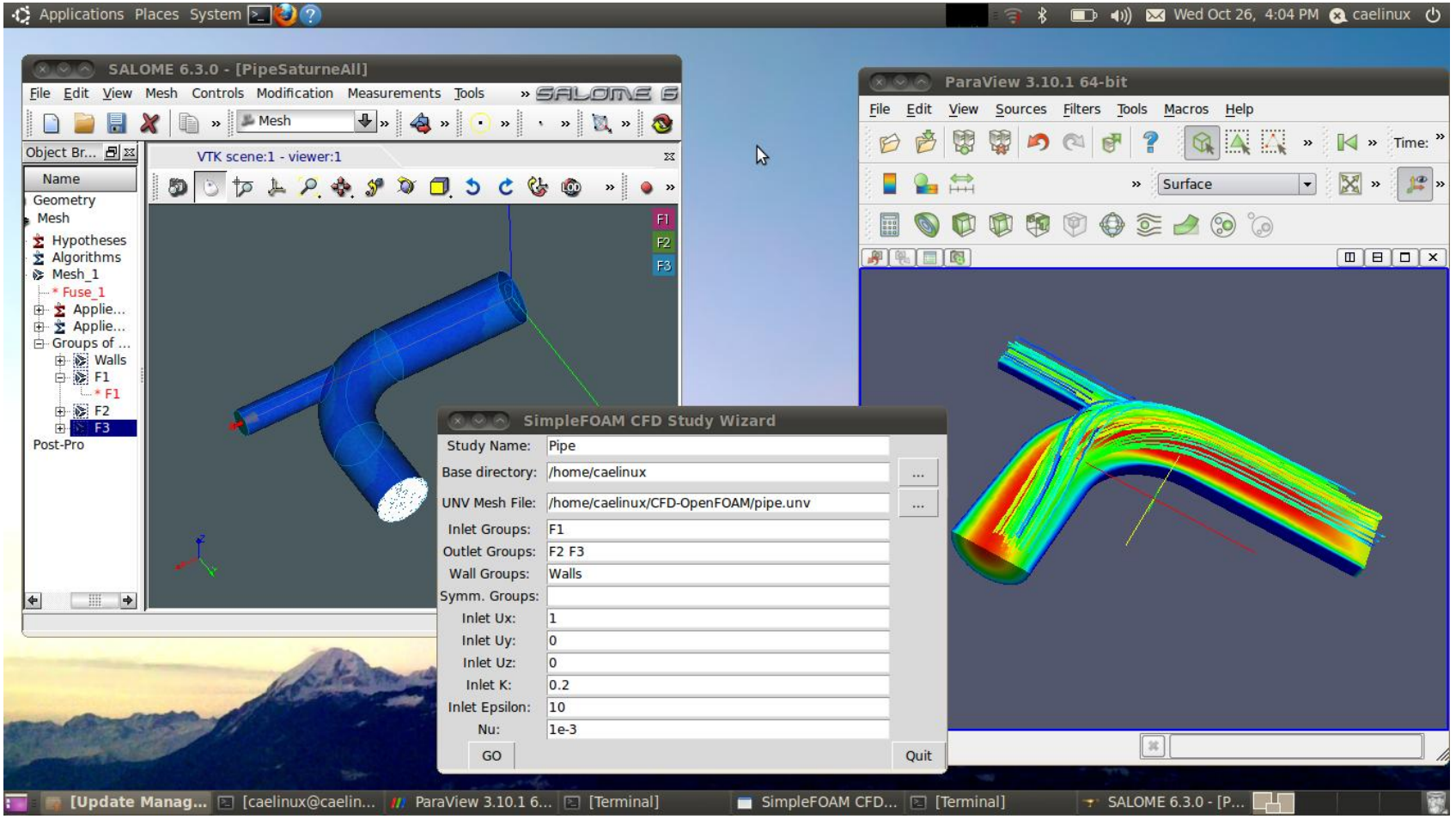
CAD/electronics and system simulation: FreeCAD, gEda, Arduino, Scilab/Xcos

What can you do with it?



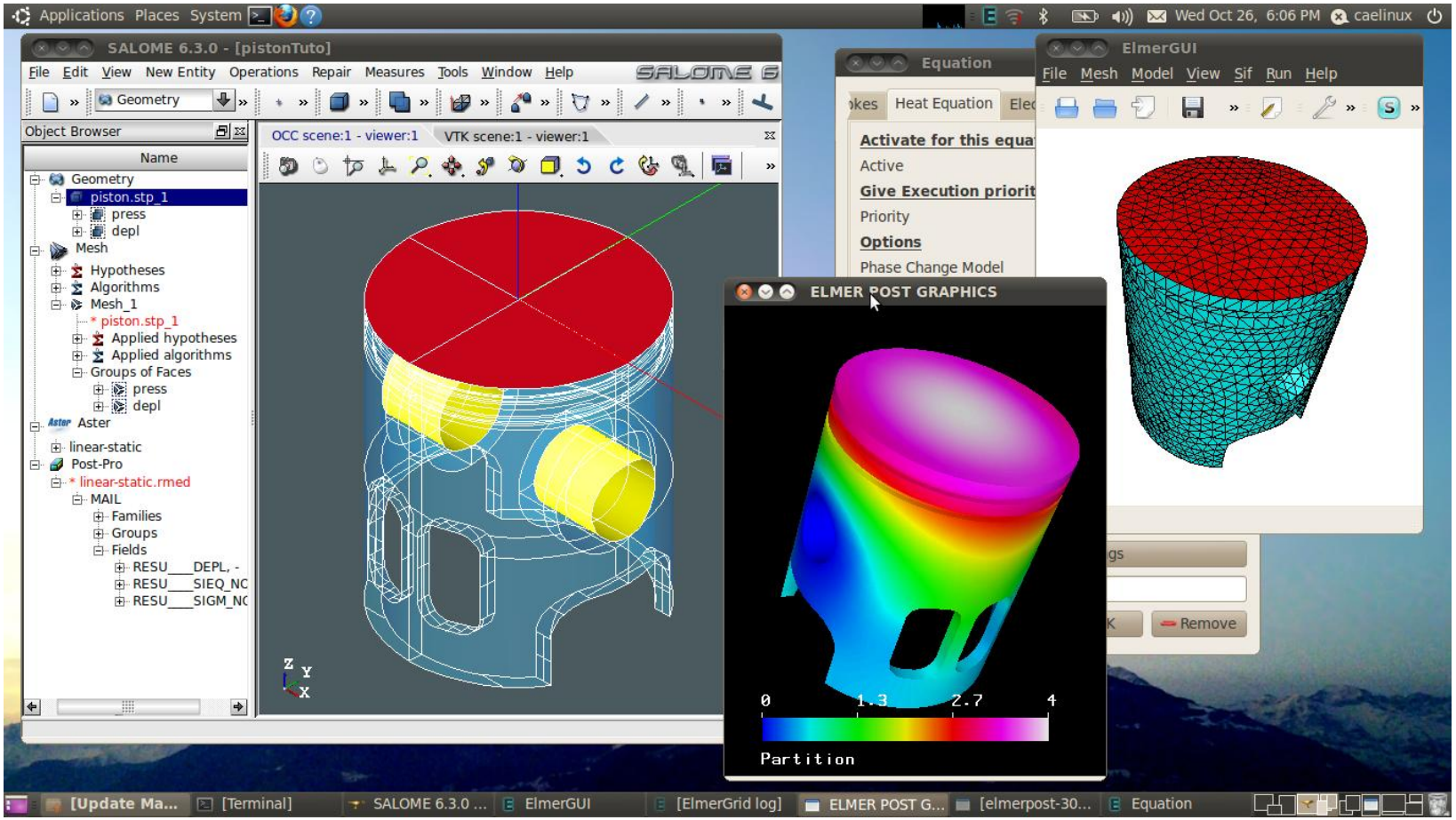
Mathematical modeling and programming: Octave, Maxima, R, Python/Scipy

What can you do with it?



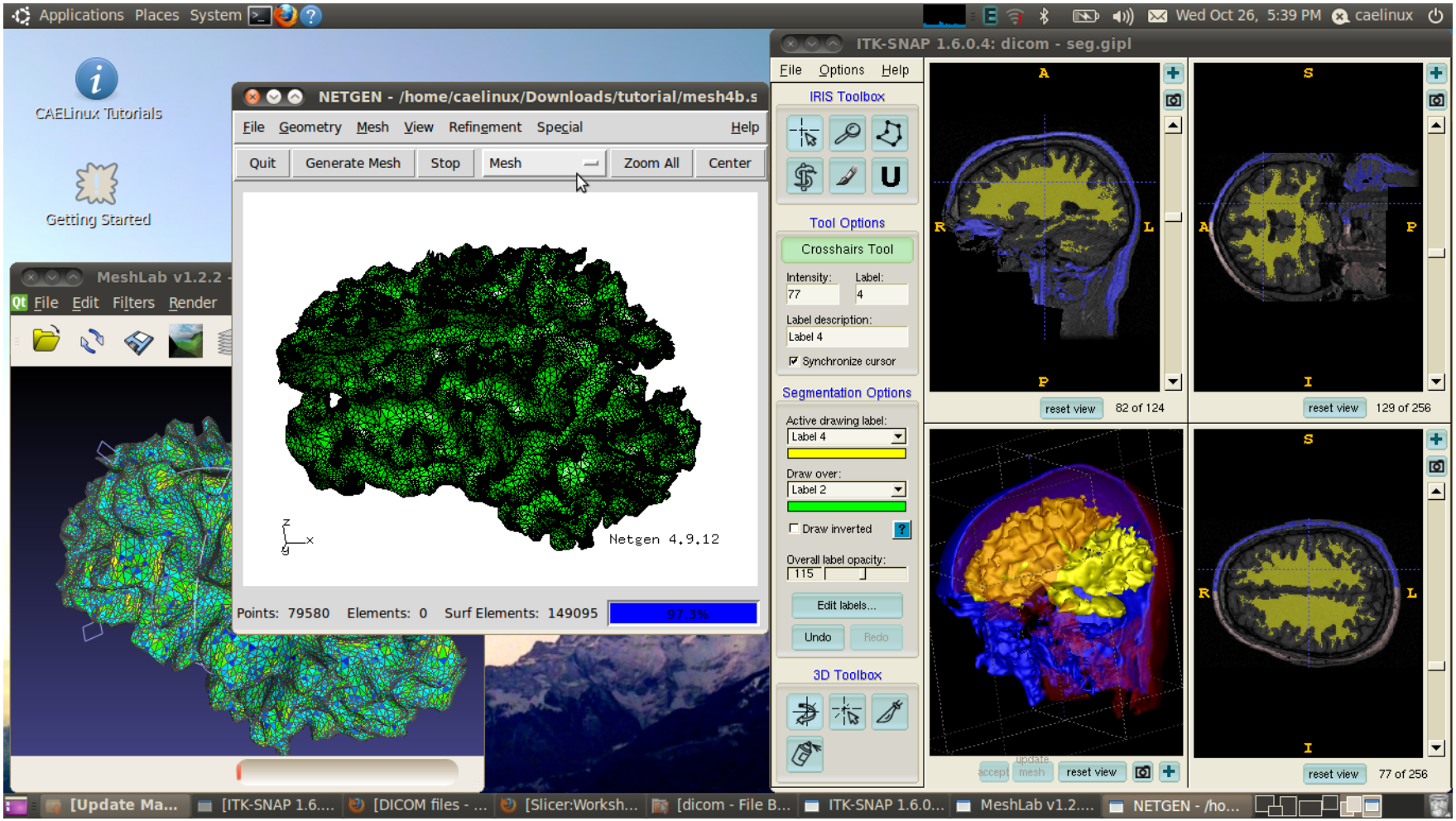
CFD: Salome / OpenFOAM or Code-Saturne

What can you do with it?



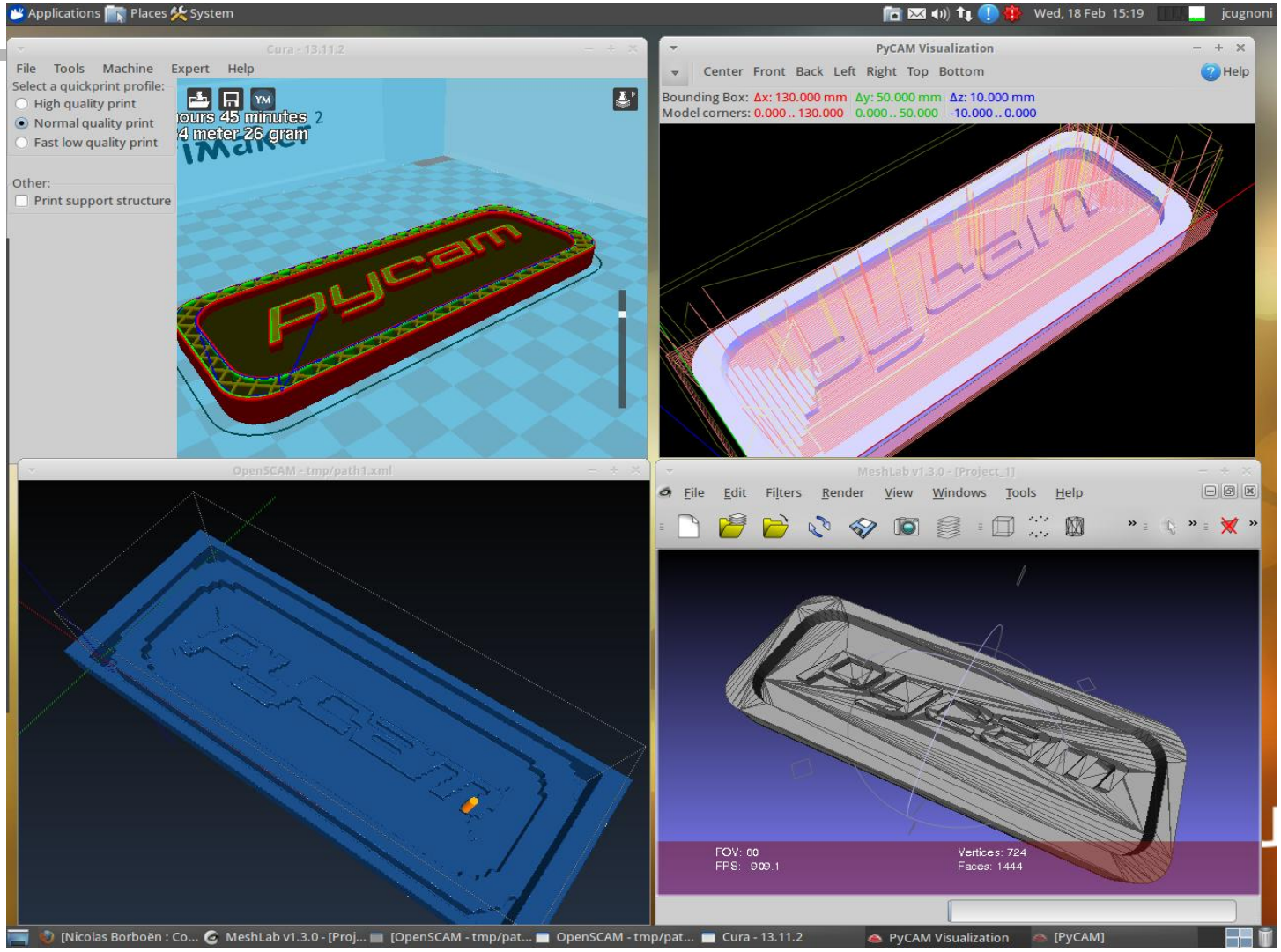
Thermo mechanics: Salome / Elmer or Code-Aster or Calculix

What can you do with it?



Biomedical image analysis and modeling: ITKSnap / VoxelMesher/ Calculix

What can you do with it?



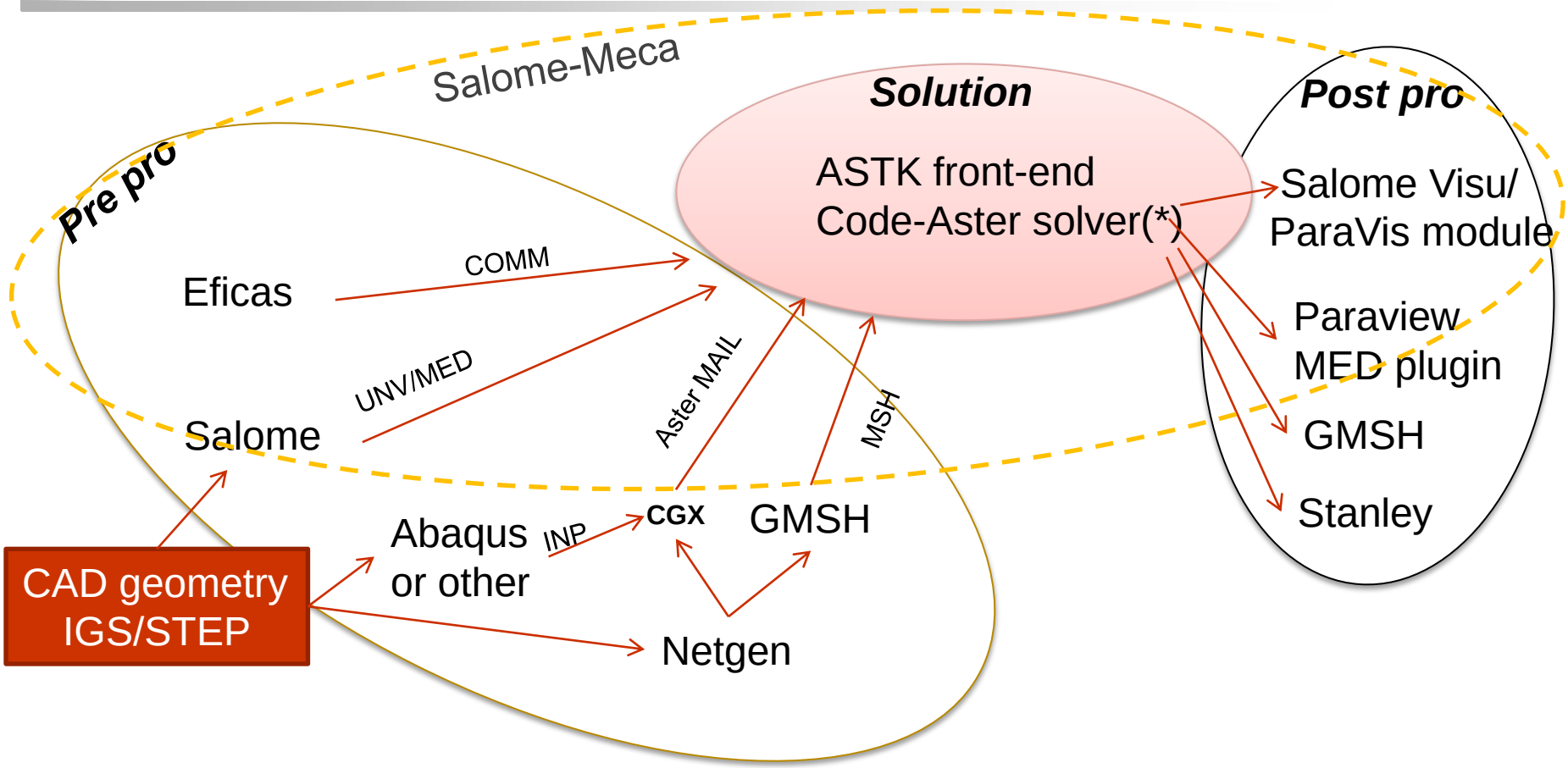
CAM, milling simulation, 3D printing: Pycam, PCB2Gcode, OpenSCAM, Cura

What is in CAELinux ?

CAE analysis process

- Design, Meshing , Pre-pro
  - FreeCAD
  - Salome
  - GMSH
  - Netgen
  - HelyxOS
  - EnGrid
  - Discretizer
  - ElmerGUI
  - CGX
  - …
- Solvers:
  - Code-Aster (FE)
  - Code-Saturne (CFD)
  - Syrthes (FE, thermo)
  - OpenFOAM (FV, CFD/multiphysics)
  - Elmer (FE / Multiphysics)
  - Impact (FE)
  - Gerris (FV, CFD)
  - MBDyn (Multibody)
  - …
- Visu, post-pro
  - Salome
  - GMSH
  - Paraview
- Math / plotting
  - Octave
  - Scilab
  - Maxima
  - R
  - Scipy
  - Gnuplot
  - …

Code-Aster Workflow in CAELinux



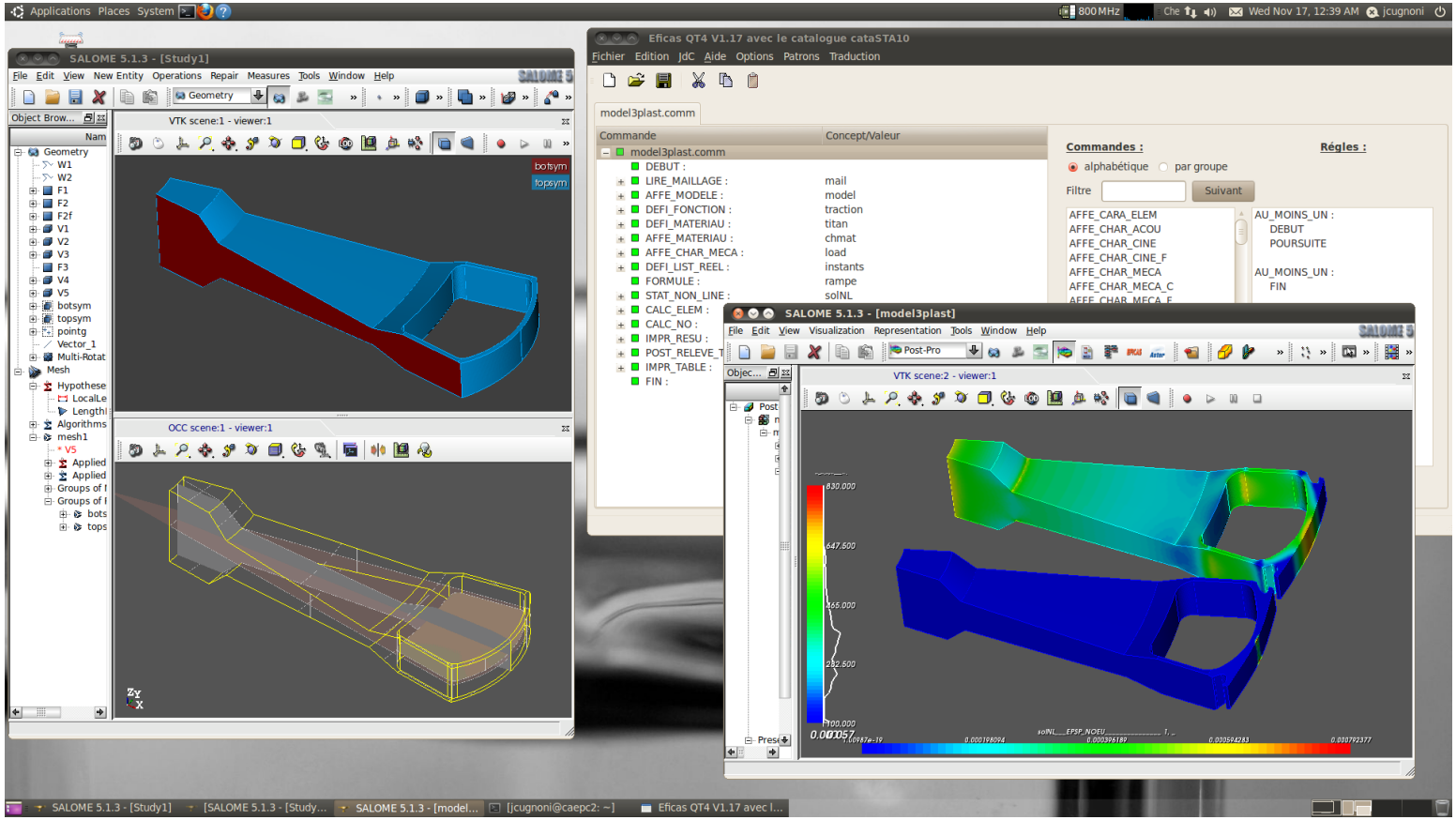
•Two versions of Code-Aster are available in CAELinux:
- STA and NEW from Salome-Meca with OpenMP parallelism (MULT_FRONT solver)
- Custom NEW MPI version with PETSC & MUMPS parallel solvers in /opt/aster

CAELinux / Aster documentation: how to get started

- Getting started:
  - CAELinux Wiki (http://www.caelinux.org):
    many valuable tutorials, some interactive, some on PDF, different level of complexity. Many user contribution in “Contrib” section!
  - Code-Aster.org: free & high quality Training material (http://www.code-aster.org/V2/spip.php?article282)
  - Salome-Platform.org: Salome tutorials for all levels (http://salome-platform.org/user-section/salome-tutorials)
- Documentation on Code-Aster.org :
  - Start with U2 methodological documents, very valuable guidelines & tips
  - Identify the main commands to use and read the U4 docs
  - In case you need it, read the corresponding Reference doc to understand the theory behind
  - Find an validation test (V doc) or search (grep?) in “Astest” folder for a COMM file that is close to what you need, try to replicate it and check.

Salome / Code-Aster: examples

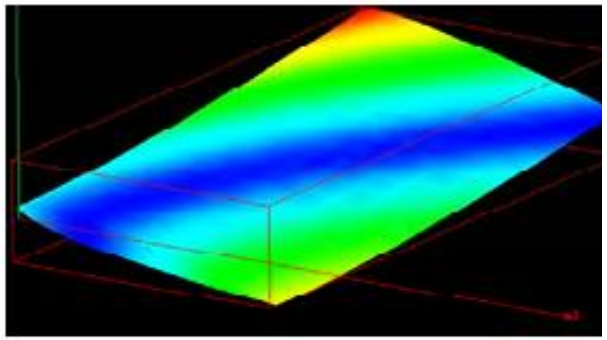
Optimization: Salome + Python + Scipy => Parametric FEA => Code-Aster solver

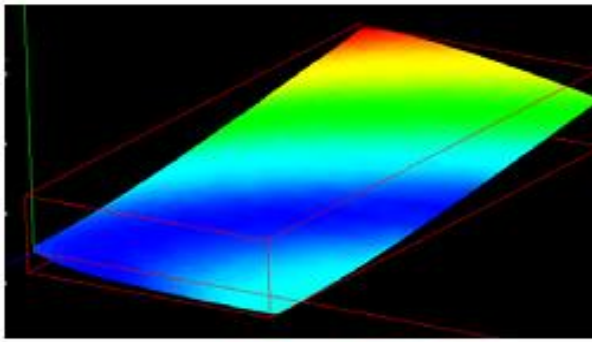


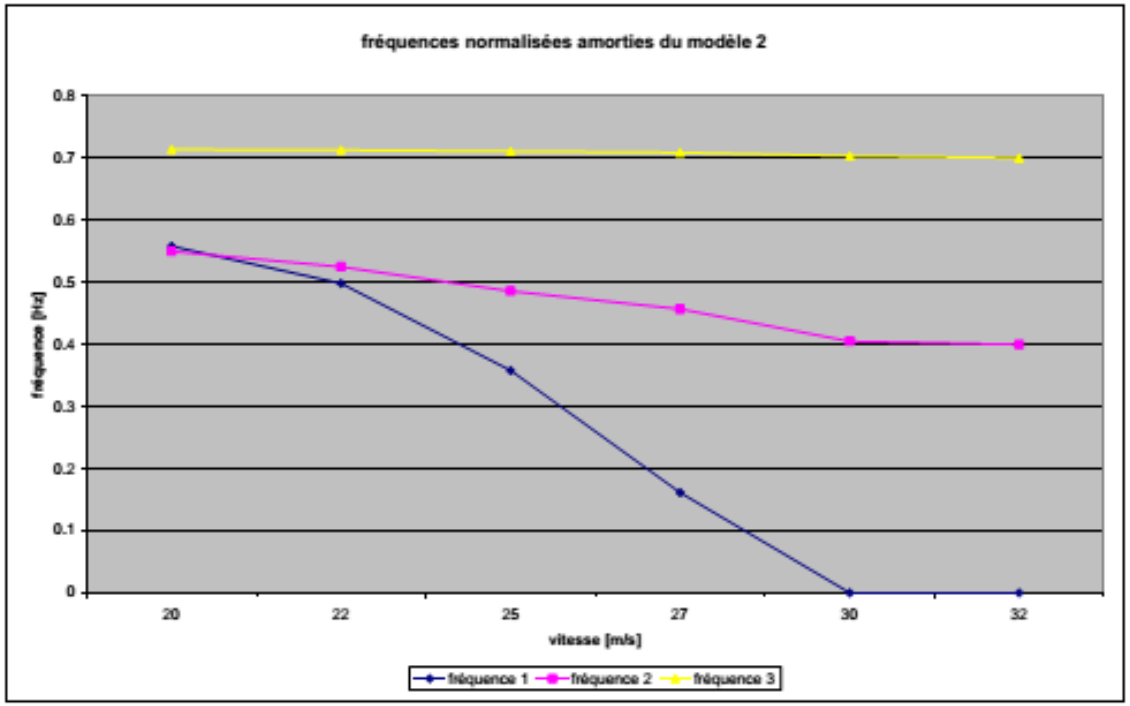
Flexibility of the platform: Python used for external optimization loop (Scipy), in Salome for parametric CAD/FE mesh & within Aster solver for custom post-pro

Fluid-structure interactions

Added mass, wet eigen frequencies and divergence of an hydrofoil





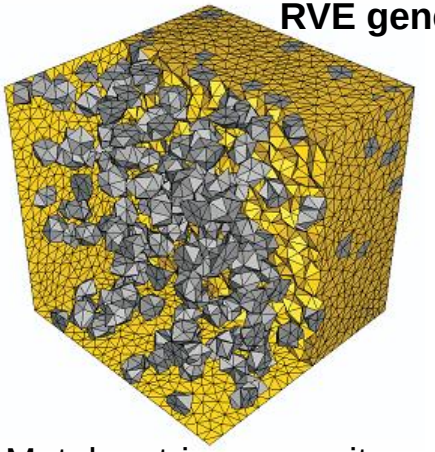


Advantage: unique possibilities of Code-Aster to compute added mass, stiffness and damping from a potential theory.
Issue: hard to validate because of lack of experimental data…

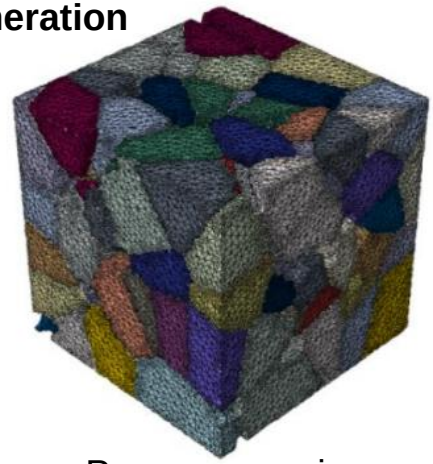
Future work & needs (Research at EPFL)

- Migrate homogenization methods developed at LMAF to Code-Aster

RVE generation

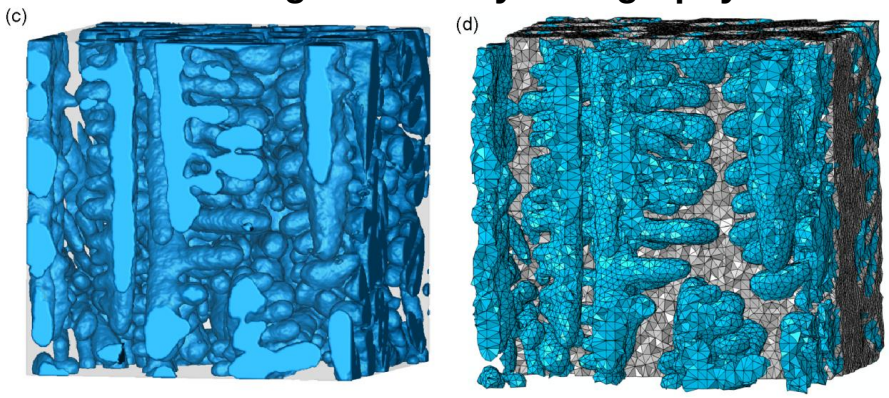


Metal matrix composites

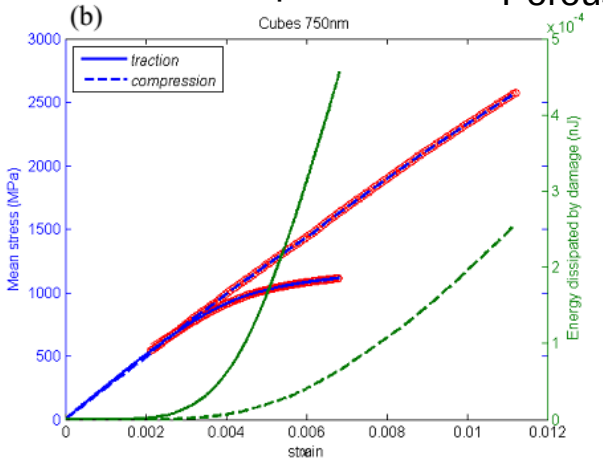


Porous ceramics

Modelling from X-Ray tomography



Alloys (here SnAgCu solder)

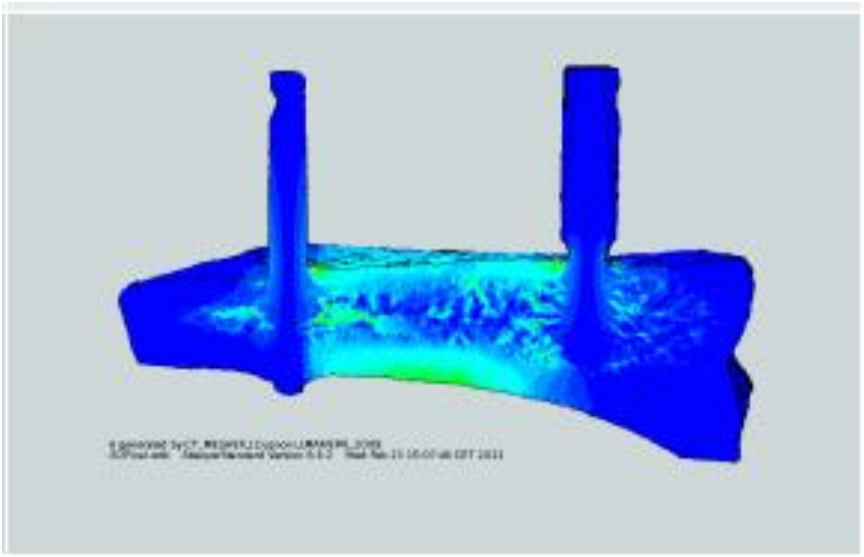


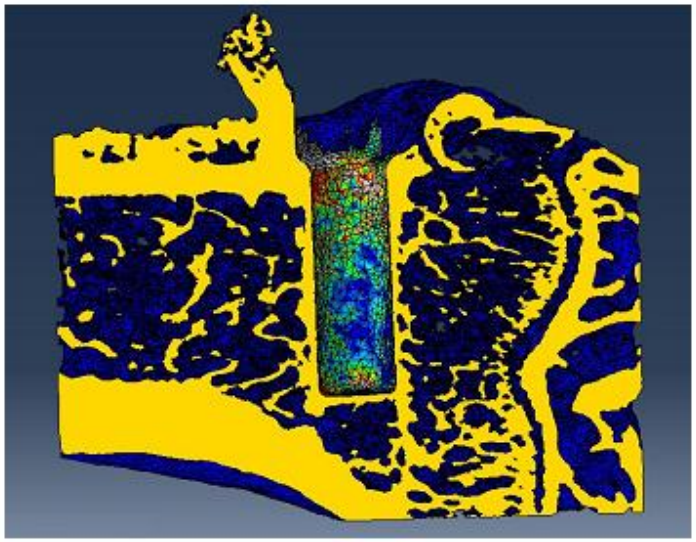
Opportunity:
develop an open source homogenization platform (equivalent to Digimat)

Need:
-high performance parallel processing in non-linear problems & robust solvers, maybe dynamic (explicit)
-cohesive contact models (not elements)

Future work & needs (Research at EPFL)

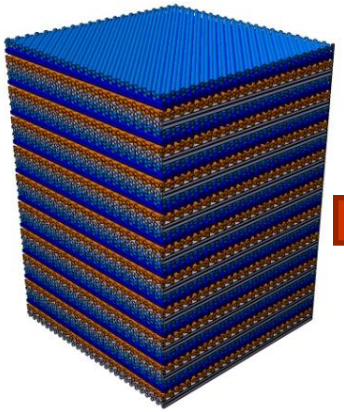
- Migrate LMAF's bone modeling tools (VoxelMesher) to Code-Aster



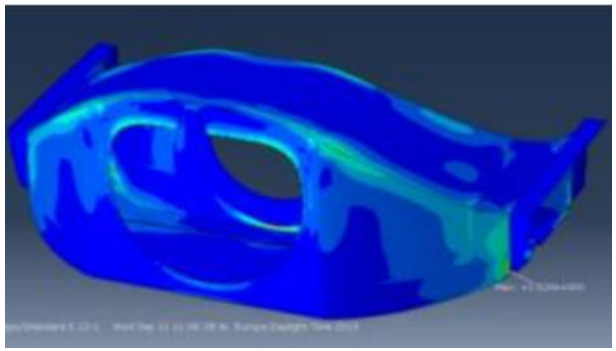


Opportunity:
develop an open source platform for biomedical engineering
Need:
-time to write a mesh writer to Aster (MAIL format) & implement element-wise elastic properties in a COMM file using an external file as input
-cohesive contact models for interface modeling, Drucker-Prager plasticity with damage and efficient parallel solvers (>3MDof)

Future work & needs (Research at EPFL)



Max. Criterion, Critical ply, Failure mode



Max envelop of layerwise criteria

- Further development of Composite modeling tools:
  - Regression, resorption of features in this field recently
  - Improved & simplified inputs for multilayer shells (one MACRO?)
  - Improved performance & simplicity for multilayer shell post-processing and composite failure criteria : implementation of Hashin/Puck, Tsai-Hill, Tsai-Wu, LARC, Hoffman failure criteria
  - Faster post-processing (not layer by layer): computation of envelope of maximum inverse safety factor and critical ply number and failure type through all layers => one field with all relevant results
  - Further development of mixed mode cohesive models for delamination simulation

Code-Aster and CAELinux: experience

- What works well:
  - Versatility and flexibility of Code-Aster / Salome / GMSH environment is great
  - Many complex simulations are possible, many tuning options
  - Very open to code coupling and file transfers, integration with external tools and custom developments in Python or Fortran
  - Requires a trained user which knows what is behind (this is a + in the end)
  - Diffusion of Aster through Salome-Meca is great
  - Frequent updates of Code-Aster, valuable forum and feedback from Devs
  - Aster is at the fore front of research in some domains but remains a generalist FEA solver with excellent multiphysics capabilities
  - Many improvement in parallel solution performance recently, becomes also more robust with contacts

  - For CAELinux: it has found a great audience, is used worldwide!

Code-Aster and CAELinux: experience

- What could be improved:
  - English doc is hard to read… start an open human « translation project »?
  - More methodological docs & intermediate level tutorials to ease learning
  - Salome Wizards could be expanded to help the transition between beginner and expert levels
  - Small community, needs more interactions to keep it alive
  - Default settings in non-linear solution and automatic time stepping could be improved for better performance. Trying to replicate (and set as default) Abaqus time stepping / convergence analysis would be highly beneficial.
  - Display performance issues in Salome Mesh & Visu but improved recently
  - Some inconsistencies in post-processing, issues with Von Mises in tetrahedra, slow post-processing (CALC_CHAMP) compared to solver...
  - Deploying Linux in companies remains an issue, even with Virtual Machines
  - More synergies between actors should be found to mutualize development / training and support
  - For CAELinux: should migrate to an open development model

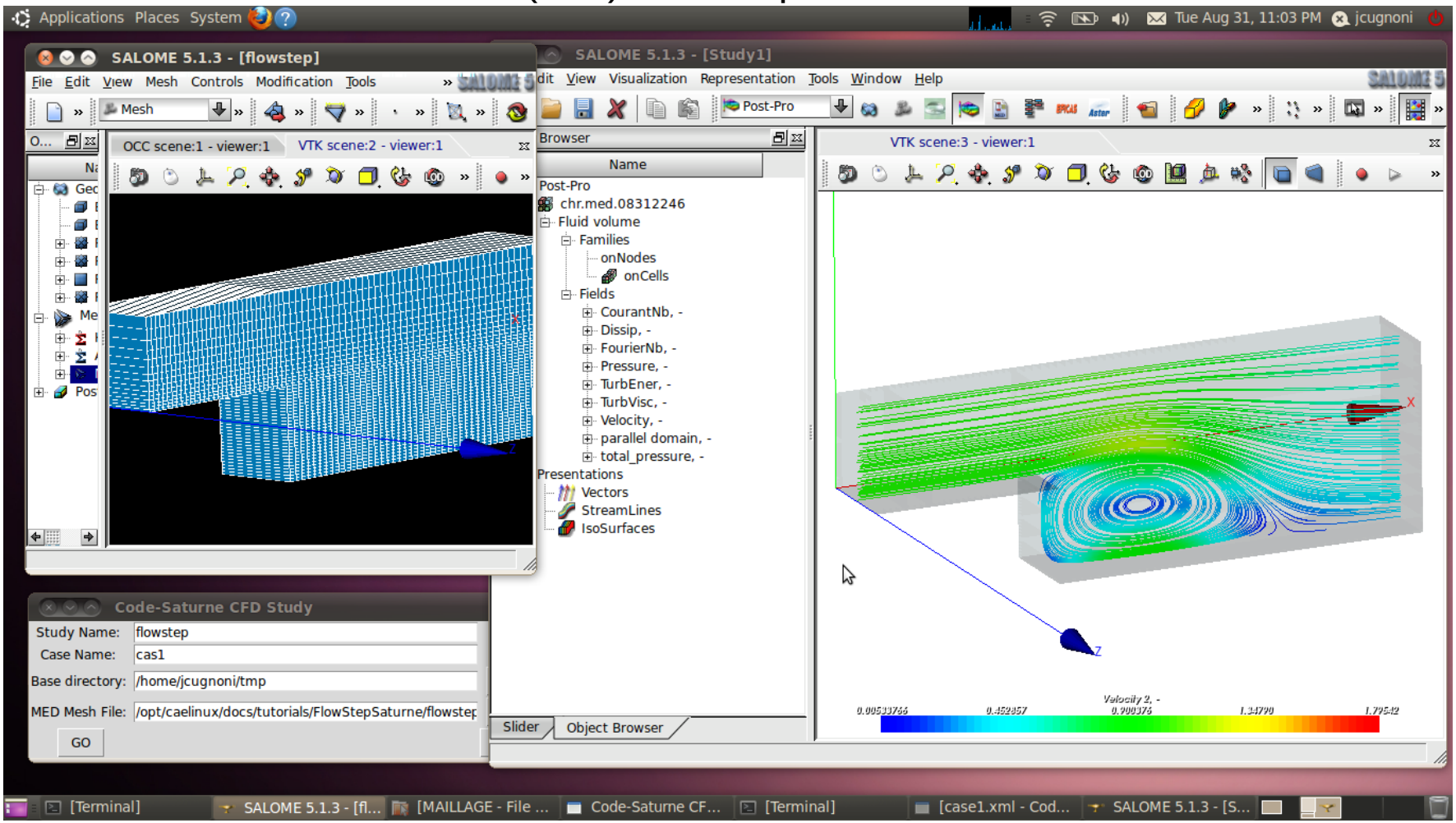
CAELinux: Development and future

- Development process:
  - CAELinux uses Ubuntu LTS 64 bit as a base and Remastersys to build the final ISO image of the distribution
  - Use as much as possible existing Debian/Ubuntu packages, but some are outdated or lacking features like parallel solvers.
  - Build « hand made » packages for key CAE software such as Code-Aster, Code-Saturne, Elmer to use recent versions and enable parallel MPI solvers and optimized math libraries
  - Building a reference image with all chosen packages and final customization (desktop layout, shortcuts, docs & tutorials) using Remastersys.
- Future
  - Development of each package and production / testing of the final distribution is still mostly manual and iterative but ensures good stability.
  - Moving to collaborative development is a goal but transition is difficult
  - a new release is planned for Q4 2015 based on Ubuntu 14.04.

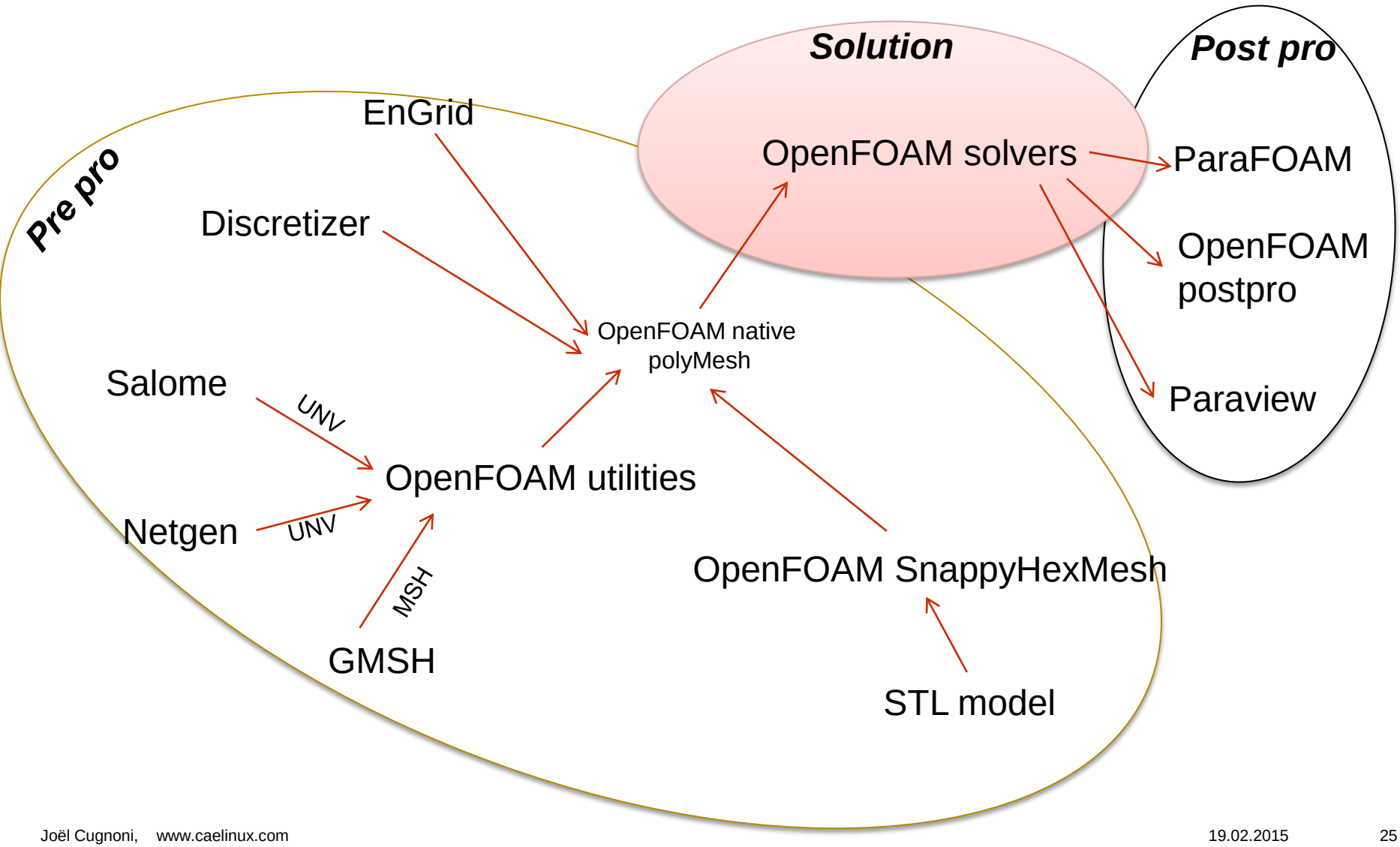
Merci de votre attention !

Other workflows: Code-Saturne

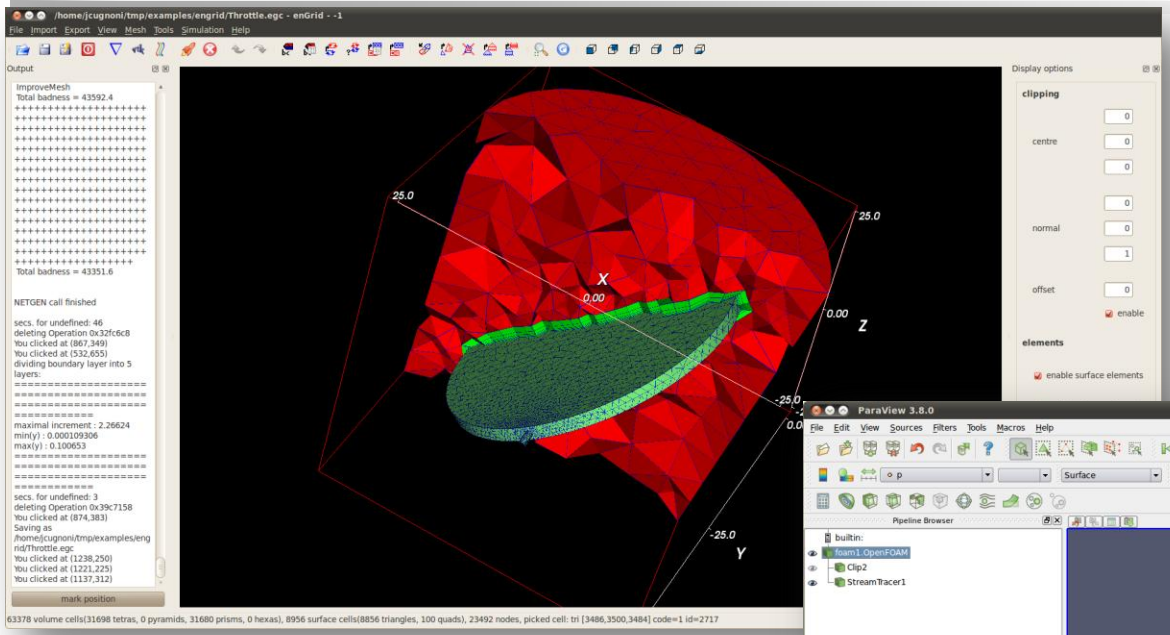
CAD + Meshing in Salome => MED mesh => Code-Saturne Wizard + GUI => Code-Saturne Solver (MPI) => Post pro in Salome or Paraview



OpenFOAM Workflow for CFD

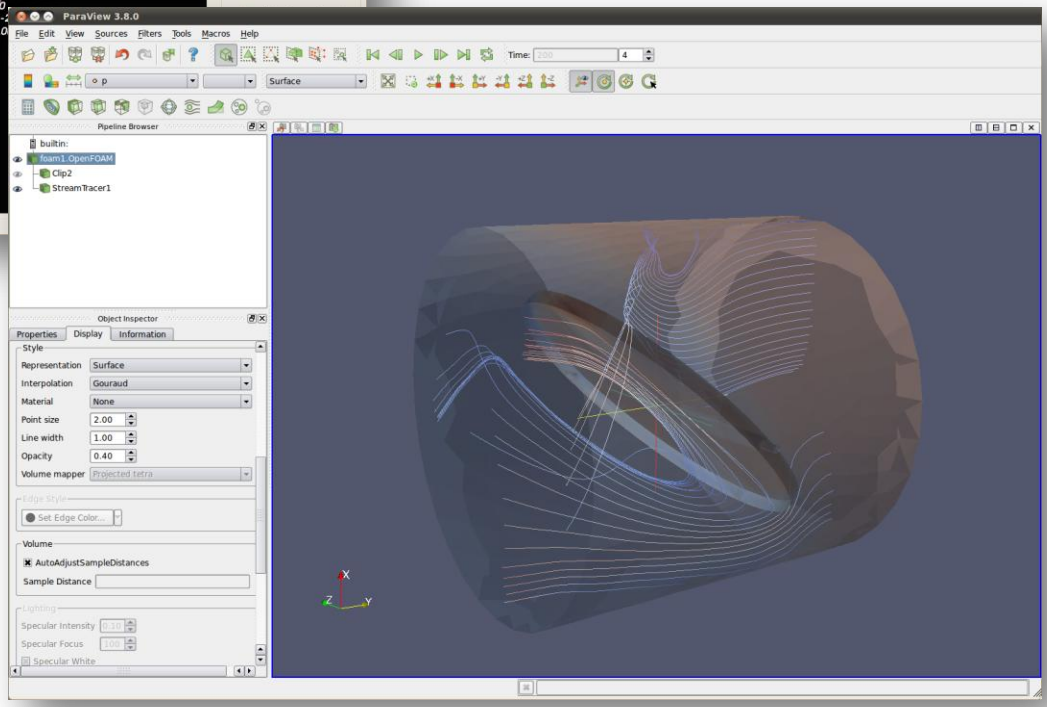


Example: EnGrid – OpenFOAM (CFD)- Paraview



Boitier papillon, CFD

EnGrid:
Maillage mixte prisme/tetra
Prepro OpenFOAM

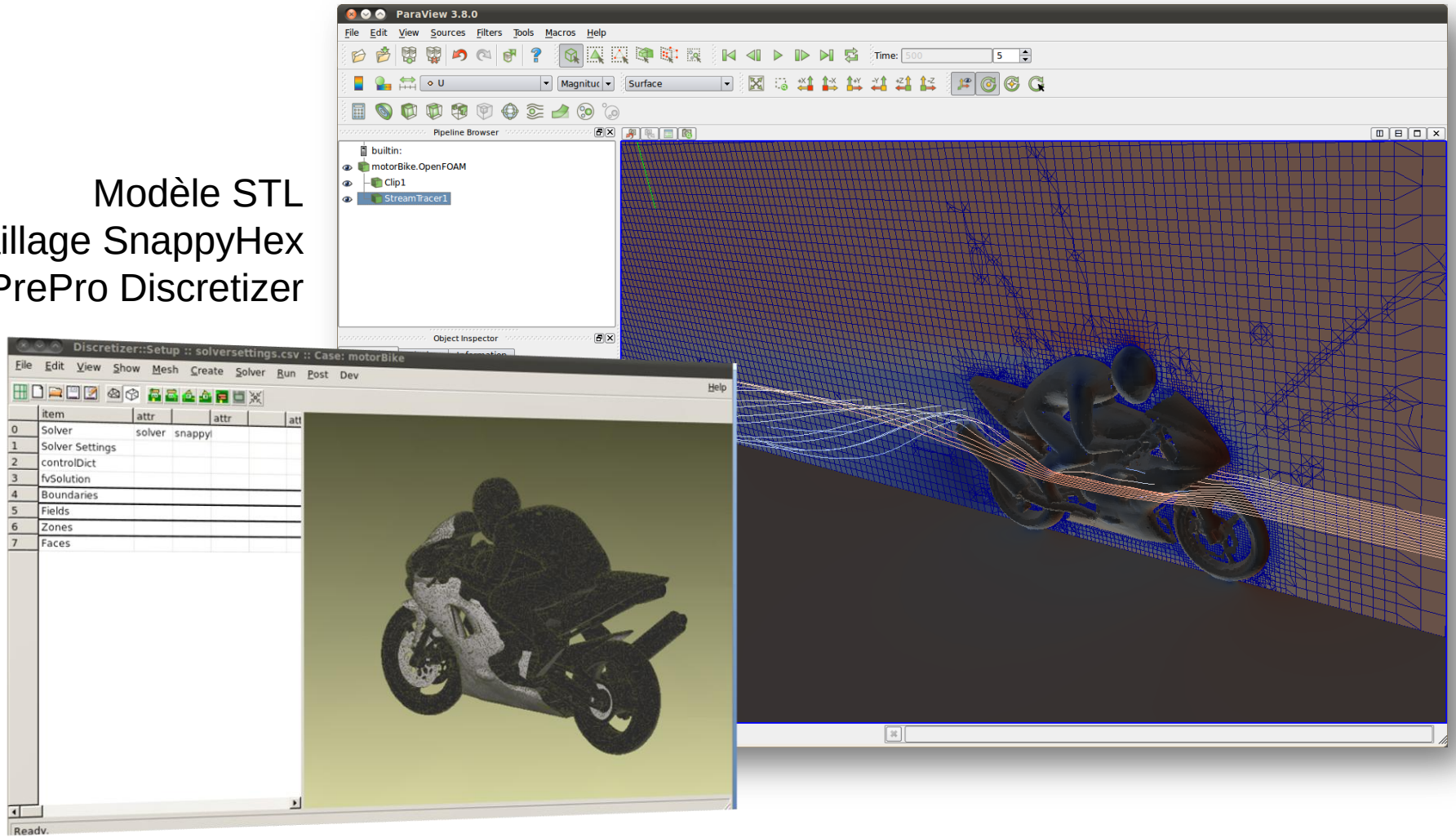


OpenFOAM:
Solveur SimpleFOAM
Stationnaire, turbulent, incompressible

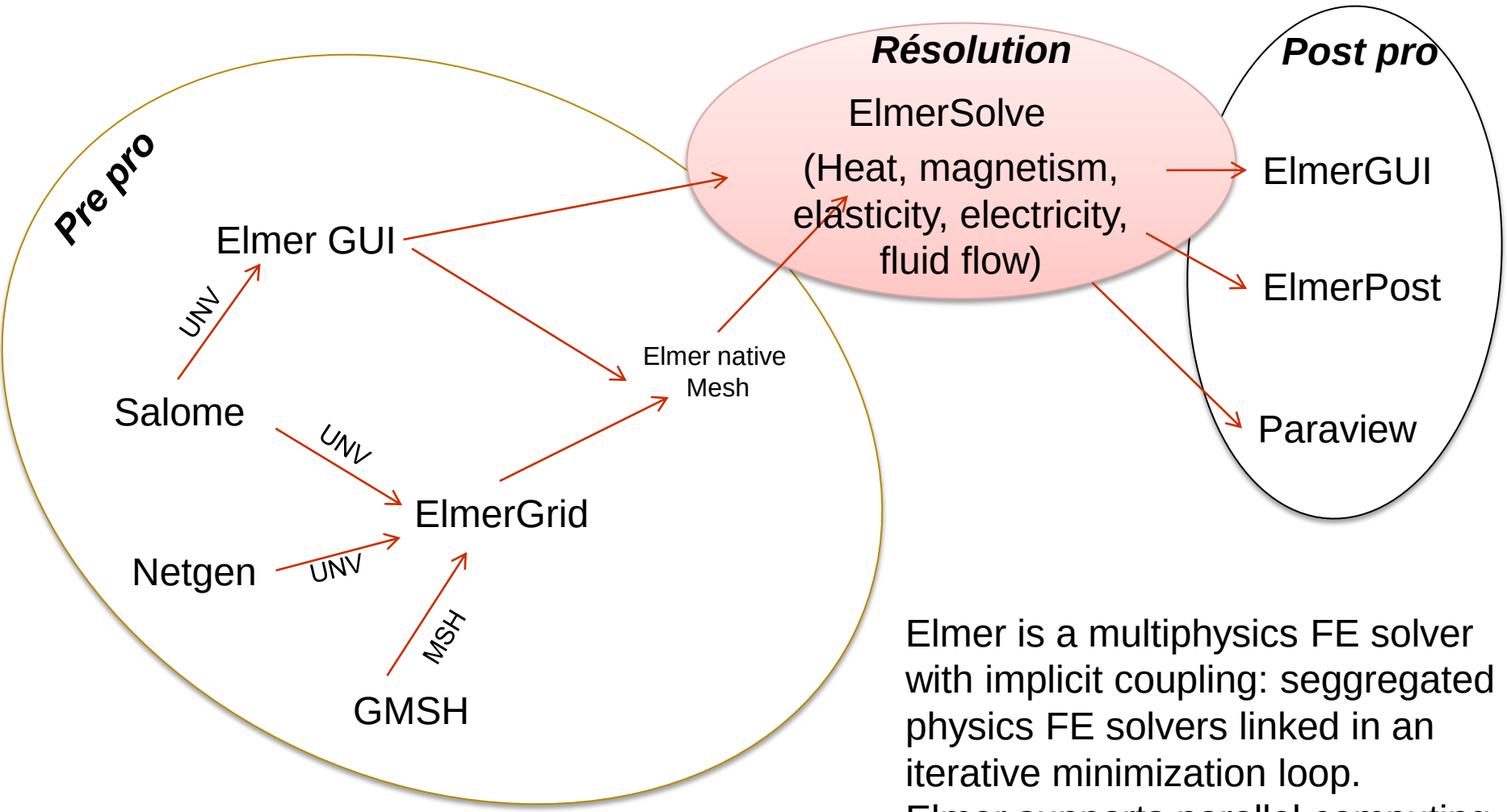
ParaFOAM:
Post-traitement / visualisation

Exemple:
Discretizer::Setup – SnappyHexMesh - OpenFOAM - Paraview

Modèle STL
Maillage SnappyHex
PrePro Discretizer



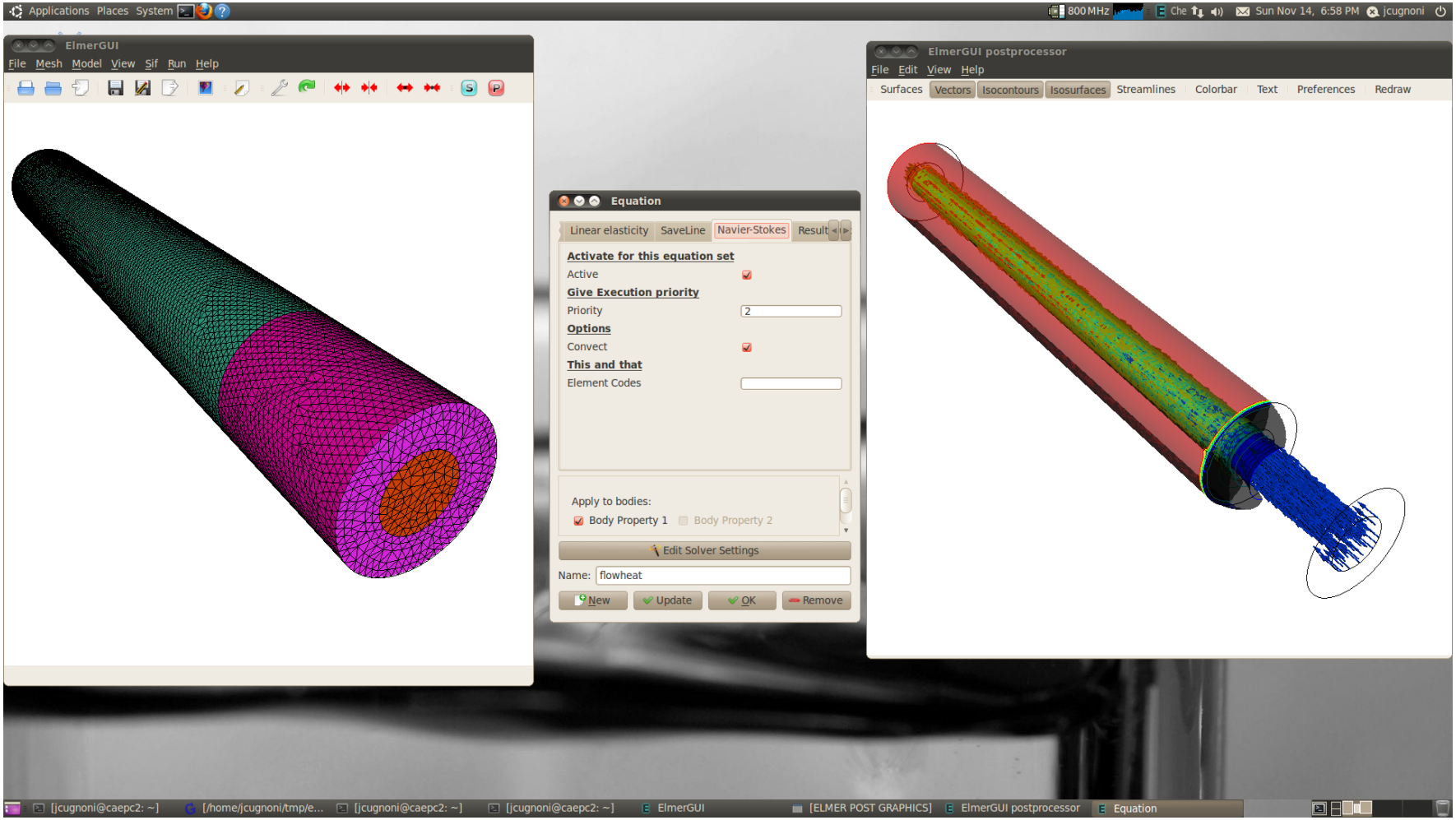
Multiphysics simulation workflow with Elmer (FE)



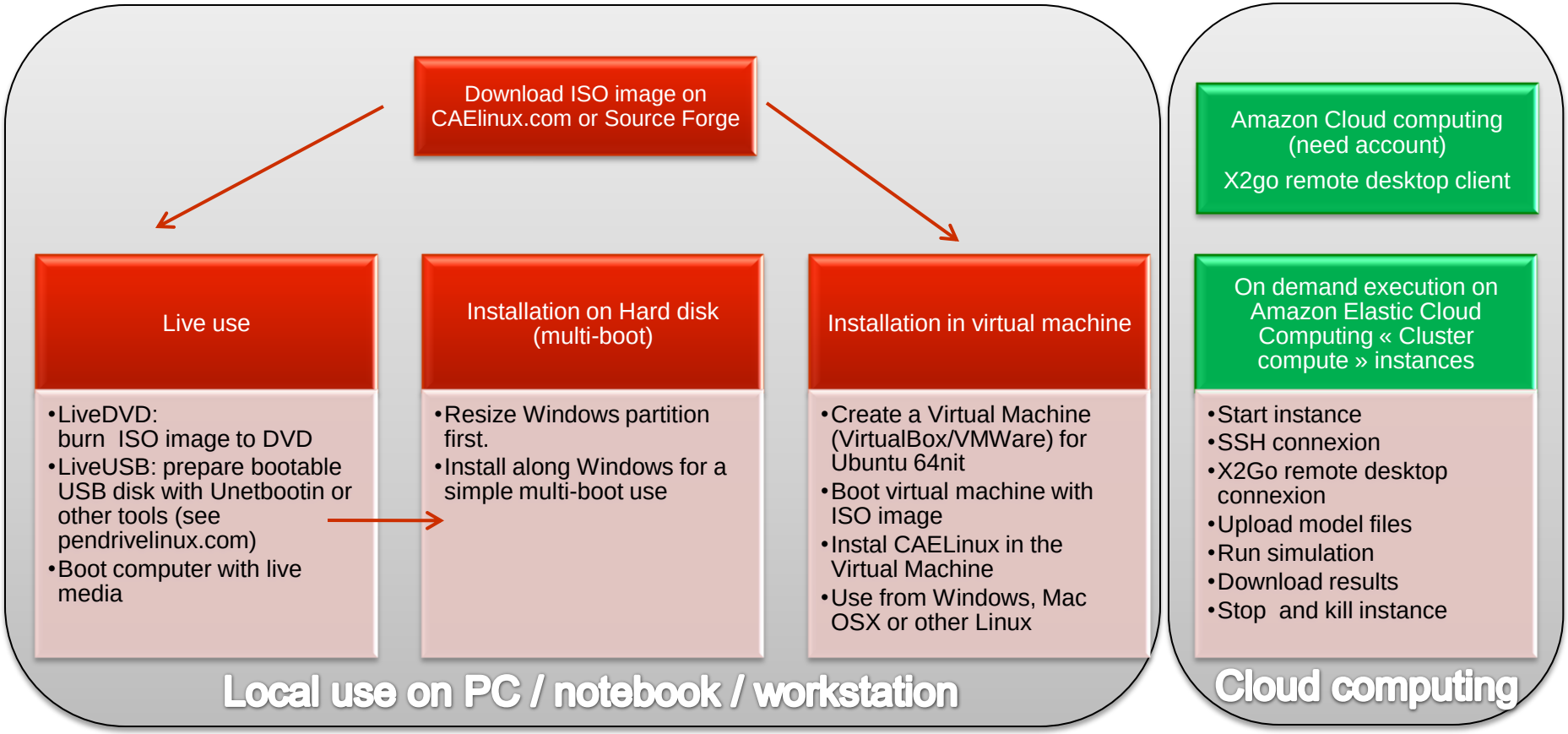
Elmer is a multiphysics FE solver with implicit coupling: seggregated physics FE solvers linked in an iterative minimization loop.
Elmer supports parallel computing through OpenMPI.

Exemple: Salome (unv) - ElmerGUI –Elmer

Echangeur de chaleur tubulaire: Navier-Stokes + Transfert Chaleur (cond. + conv.)

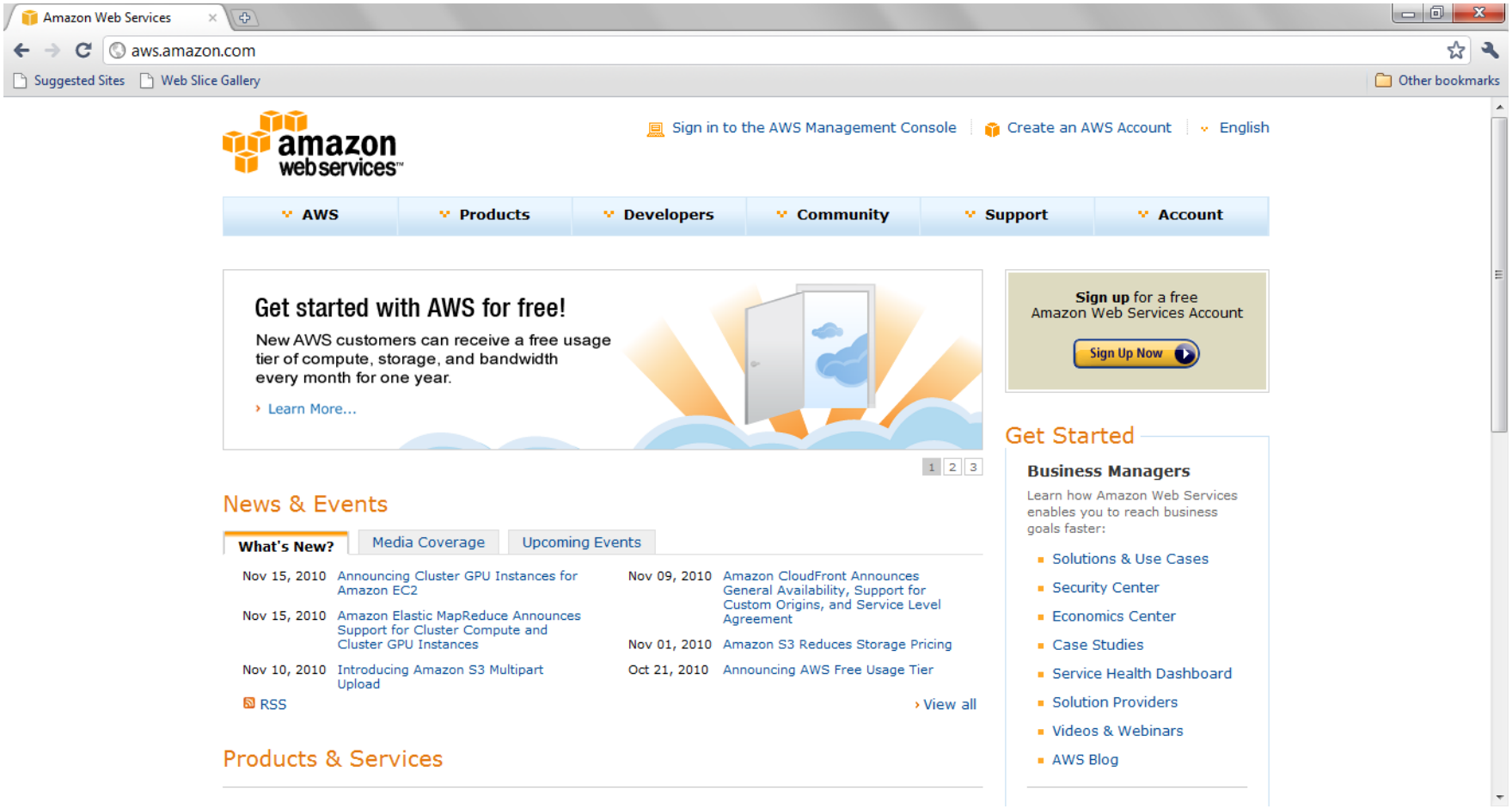


How to deploy / use CAELinux ?



CAELinux in the Cloud on Amazon EC2

- Start instances of CAELinux from aws.amazon.com



CAELinux in the Cloud: remote desktop