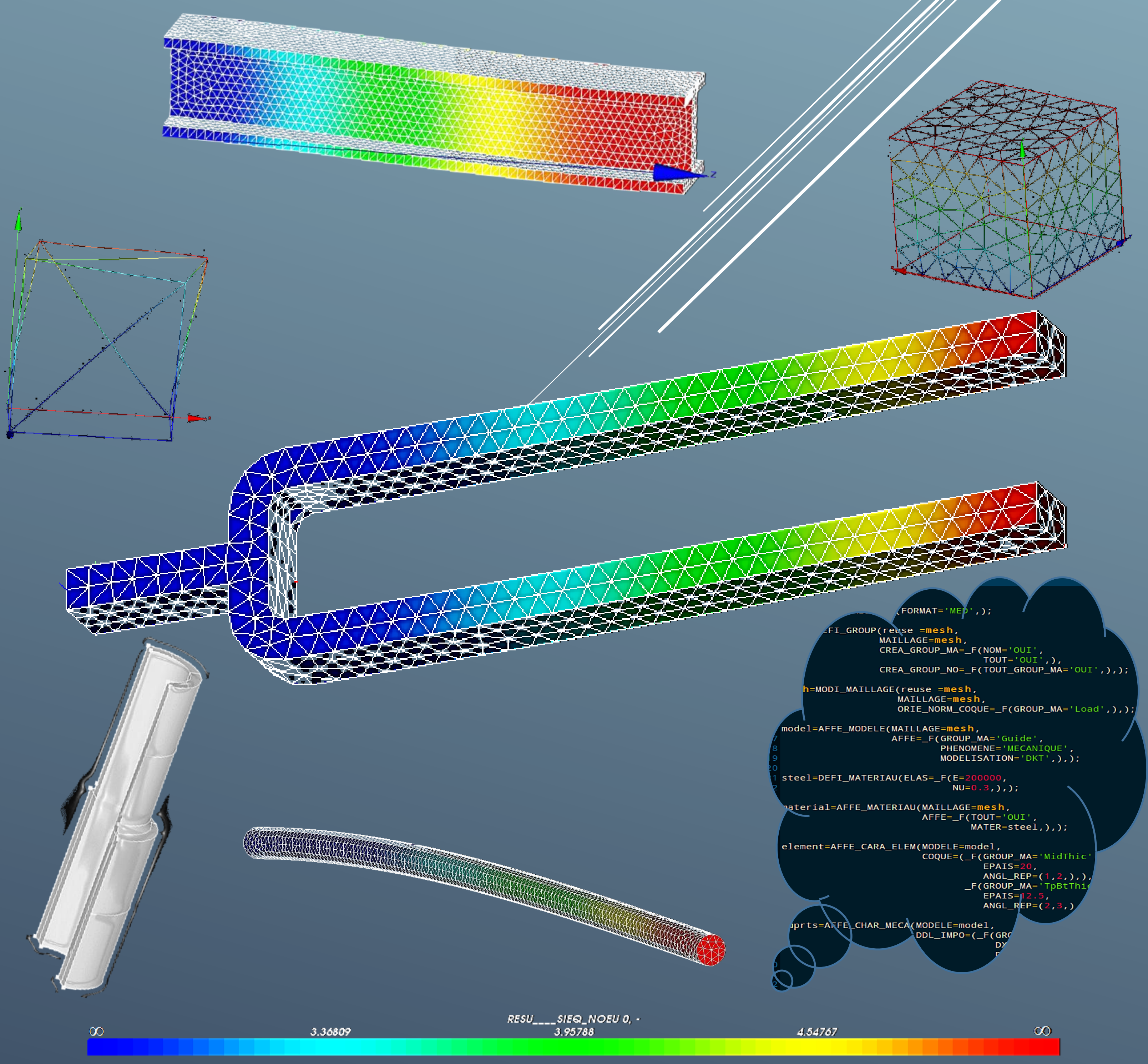
code_aster

FINITE ELEMENT ANALYSIS

A FIRST COURSE TO

Code_Aster

Guide through to Salome_Meca and Code_Aster



Salome-Meca 2016

Stefanos Syllignakis

Petr Vosynek

Finite Element Analysis Method using

Open_Source_Software

Step-by-Step Guide...

Finite Element Analysis Method using

Open_Source_Software

Step~by~Step Guide…

Credits and Copyright

Written by: Bc. Syllignakis Stefanos
sylst3f@gmail.com

Main Editor: Ing. Petr Vosynek, Ph.D
petr.vosynek@gmail.com

Review and Editor: Ing. Marek Benešovský
******@vutbr.cz

Preface

The presented material was created within the Erasmus+ project of the student Stefanos Syllignakis under the leadership of Petr Vosynek. It is basically support material for the subject 6KP and its English version 6KP-A (basics of computational modeling using finite element method) taught in the Institute of Solid Mechanics, Mechatronics and Biomechanics, Faculty of Mechanical Engineering, Brno University of Technology.

Computer labs of 6KP and 6KP-A are composed of active exercises under the current interpretation of the fundamentals associated with the type of elements and also from a separate project for a group of students. The texts were made in two versions, for the computing open_source system Salome_Meca (C_A) and for computing system ANSYS Workbench v16.2.

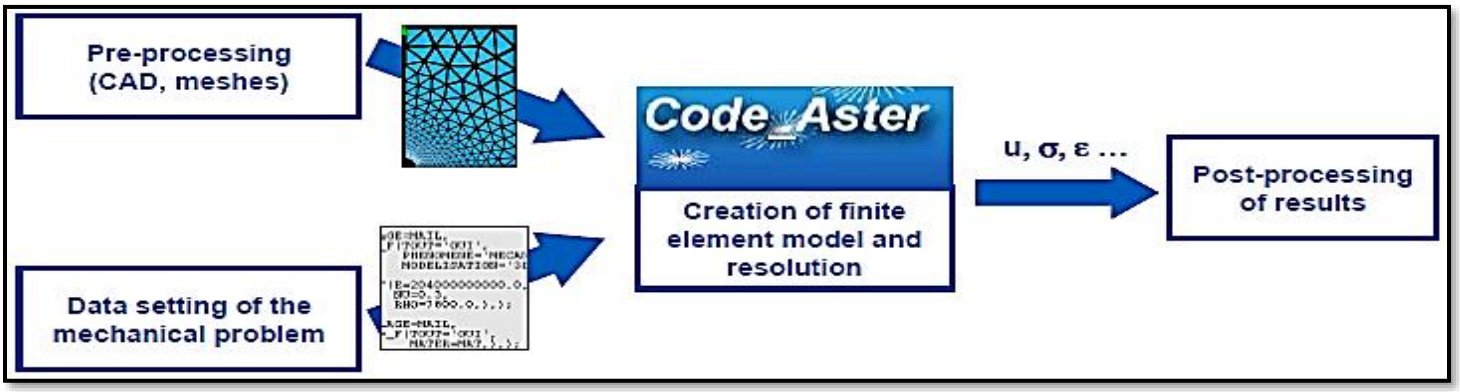
Contents

Introductory

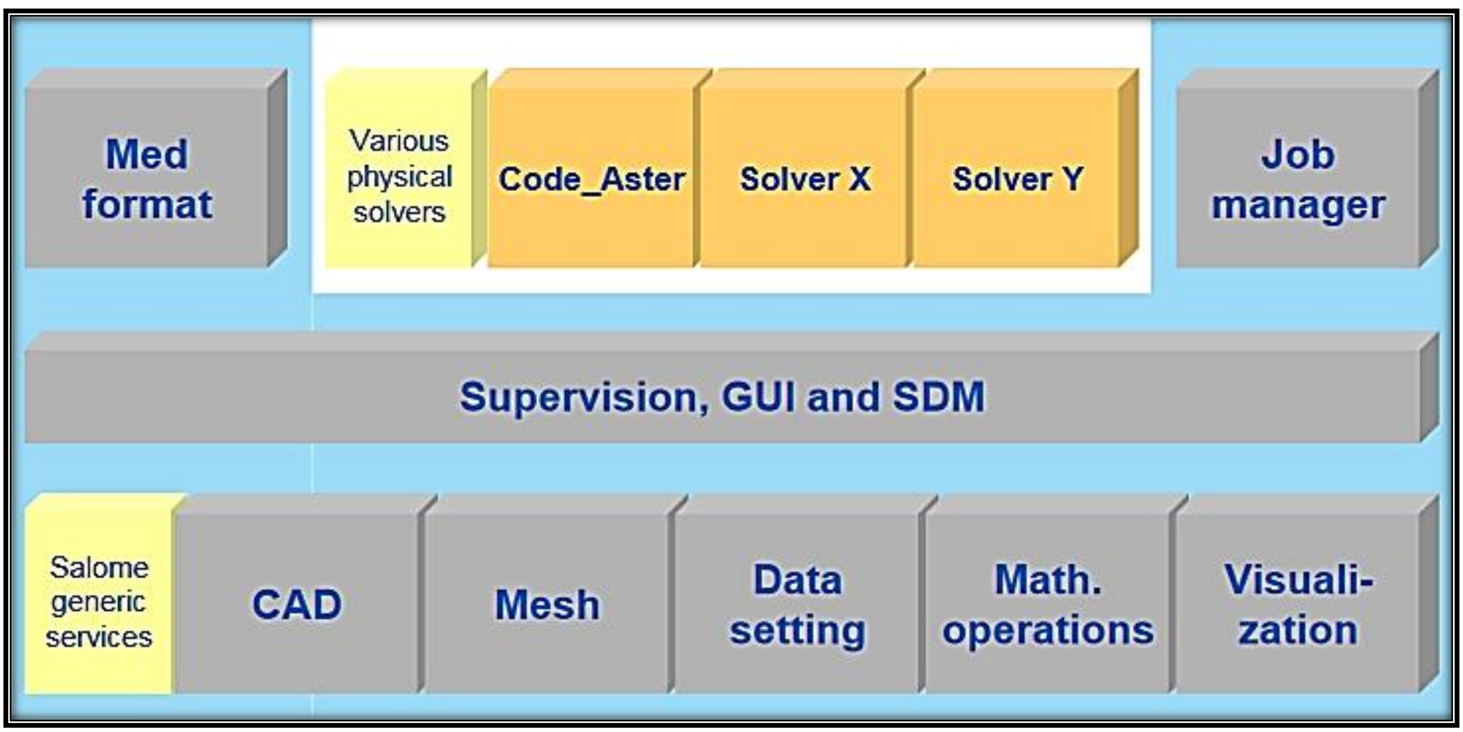
Reference --> EDF, *Code_Aster and Salome_Meca course material.*

General Principles of Code and Platform

- # General Principles of Code and Platform:
  - Code_Aster is a "stand-alone" thermos-mechanical solver.
    - No integrated GUI to create geometries and meshes.
    - No colorful post-processing.
    - With study data prepared in a text file.
    - Input: Mesh and data setting.
    - Output: Physical fields (displacement, stress-strain, temperature).

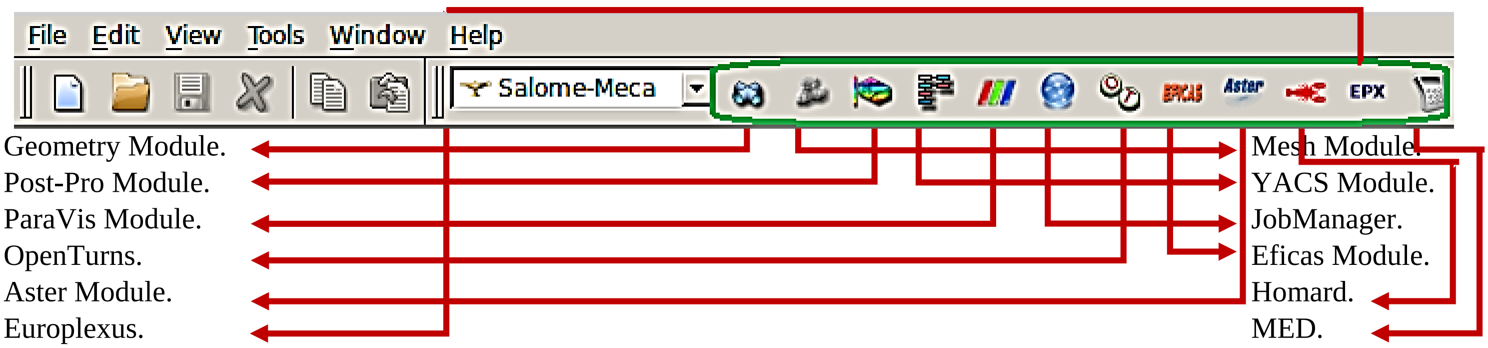


- # Salome is a generic framework for pre- and post-processing.



- # Code_Aster is a thermo-mechanical solver.
- # Salome is a generic platform for physical simulations.
  - Salome_Meca = Salome + Code_Aster (Software integration).

- # Advantages:
  - Easy installation of a complete framework (Linux).
  - A consistent and continuous graphical environment.
    - Access from different modules to main Salome study elements: meshes, results.
    - Graphical selection of topological entities for data setting of Code_Aster.
  - Possibility of using different pre- and post-processing tools.
    - Importation of meshes and geometries prepared by GEOM and SMESH Salome modules.
    - Importation of different input mesh formats and output result formats.
  - Possibility for a “stand-alone” use of Code_Aster solver.
- # Salome_Meca is the integration of Code_Aster solver in the Salome Platform.



Presentation of Code_Aster

- # Code_Aster is an acronym for analysis of structured and thermos mechanics for studies and research, is a general Finite Element Analysis software coming from EDF (Electricite De France) R&D Department.
- # It can handle mechanical thermal and associated phenome in all sort of analysis: Linear Statics, Non-Linear Statics, Dynamics, Thermics and more.
- # Due to its numerous capabilities, Code_Aster is a very complex affair and its somewhat unfriendly user interface makes the learning curve quite steep at the beginning.
- # An all-purpose code for thermos-mechanical study of structures.
  - With a wide variety of models:
    - More than 400 finite elements: 3D, 2D, Shells, Beams, Pipes.
    - More than 100 constitutive laws.
    - A wide range of solvers: Mechanical Statics and Dynamics, Vibrations, Modal and Harmonic Analysis, Thermo-Hydro-Mechanical Coupled Problems, Thermic, Acoustics, etc.
  - A computational software used by engineers, experts and researchers.
    - Studies: A need of robust, reliable, tested and qualified industrial simulation code at EDF.
    - Researches: Continuous integration of new models in the development versions.
- # Solving three types of non-linear problems.
  - Material Behavior: About a hundred nonlinear constitutive laws.
  - Kinematics: Large displacements, large strains, large rotations.
  - Contact and/ or friction.
- # Advanced features in mechanics.
  - Porous media, fracture mechanics, fatigue, damage, metallurgy, seismic analysis, rotating systems.

Code_Aster Definitions.
- FONCTIONS [FR]: Tabulated (discrete) function depending on one parameter.
- NAPPE [FR]: Tabulated (discrete) function depending on two parameters.
- FORMULE [FR]: Continuum formula depending on several parameters.

Code_Aster Parameters:
- ABSC_CURV → Curvilinear abscissa.
- DX, DY, DZ → Displacements along X, Y, Z.
- DRX, DRY, DRZ → Rotation along X, Y, Z.
- X, Y, Z → Coordinates X, Y, Z.
- EPSI → Strain.
- SIGM → Stress.
- INST → Time.
- TEMP → Temperature.

File Extensions:
- .comm → Command File.
- .mmed → MED file containing mesh.
- .mess → Text file containing output of the solver (message, errors, warnings, etc.).
- .resu → Text file containing results of the simulation in a table format.
- .rmed → MED file containing results of the simulation.
- .base → Folder containing the output database of the solver.
- .bhdf → File containing the base of calculations in HDF format.
- .btc → Launch script generated by the service.

Eficas standard concepts:
- DEBUT [Start].
- Material definition.
- What type of Mesh to read.
- Type of modeling [1D, 2D, 3D, mechanical, thermal].
- Assign previously defined material to the model.
- Add geometric boundary conditions.
- Add load boundary conditions.
- Define the type of analysis [static, linear, non-linear, dynamic, etc.].
- Results to be calculated at the elements and nodes.
- Save the results to a MED file.
- FIN [Finish].

Wide Range of Finite Elements

Continuous mechanics.
- 3D --> Linear, quadratic, reduced or full integration.
- 2D --> Plane strain, plane stress, axi-symmetry.
  --> Integration of non-linear behavior in plane stress.

Structural elements.
- 2D elements: Shells, plates…
- 1D elements: Beams, bars, cables, pipes…

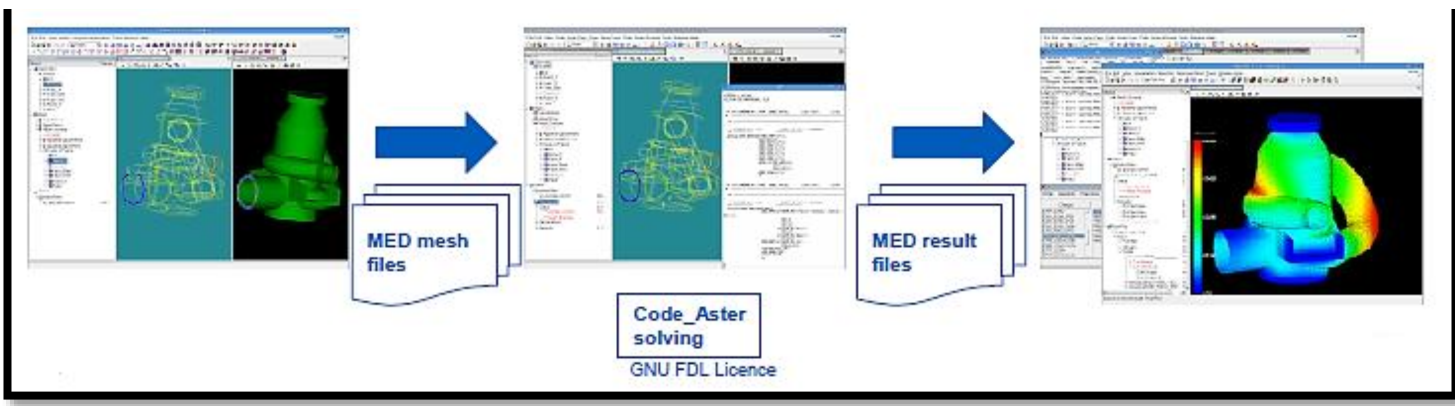
Connections and assemblies.
- Linear relationships between degrees of freedom, transmission of torques effort.

Discontinuous media (cracks and joints).
- XFEM level-sets.
- Joint elements and CZM (Cohesive Zone Model).

Algorithms and Analysis Methods

- Mechanical solvers.
    - Linear or non-linear static: **MECA_STATIQUE, STAT_NON_LINE…**
    - Dynamic on physical basis: **DYNA_LINE_TRAN, DYNA_NON_LINE…**
    - Modal analysis: **CALC_MODAL, MODE_ITER_*…**
    - Dynamic on modal basis: **DYNA_TRAN_MODAL, DYNA_VIBRA…**
- Other physics.
    - Thermic: **THER_LINEAIRE, THER_NON_LINE…**
    - Acoustics: **PHENOMENE ACOUSTIQUE…**
    - Metallurgy (for welding applications).
    - FSI: Fluid structure interaction.
    - Thermos-hydro-mechanical coupling.
- Tools for resolution.
    - Sub structuring, control of the nonlinear algorithms.
    - Several algebraic solvers, sequential or parallel, direct or iterative.
    - Post-processing tools: **CALC_CHAMP, POST_CHAMP, POST_DYNA_*…**

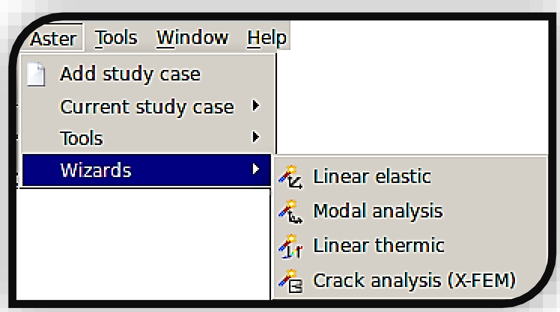
Sketching Geometry → Arranging Mesh Inputs → Visualizing Mesh Outcome → Adjusting Code_Aster → C_Aster Solver → Post-Processing Visual.

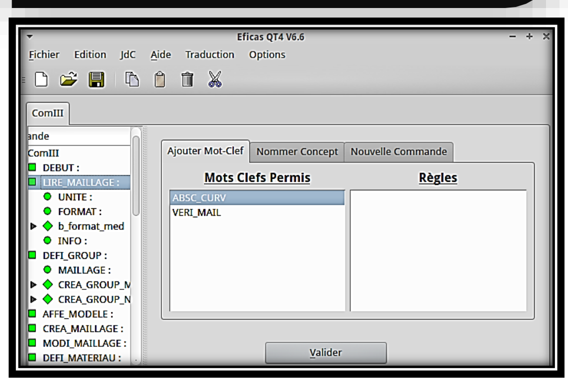


CAD Modeler/ Salome (GEOM) → Mesh tool/ Salome (SMESH) → Data setting/ Eficas, Wizard → Computation / ASTK → Visualization App/ PARAVIS

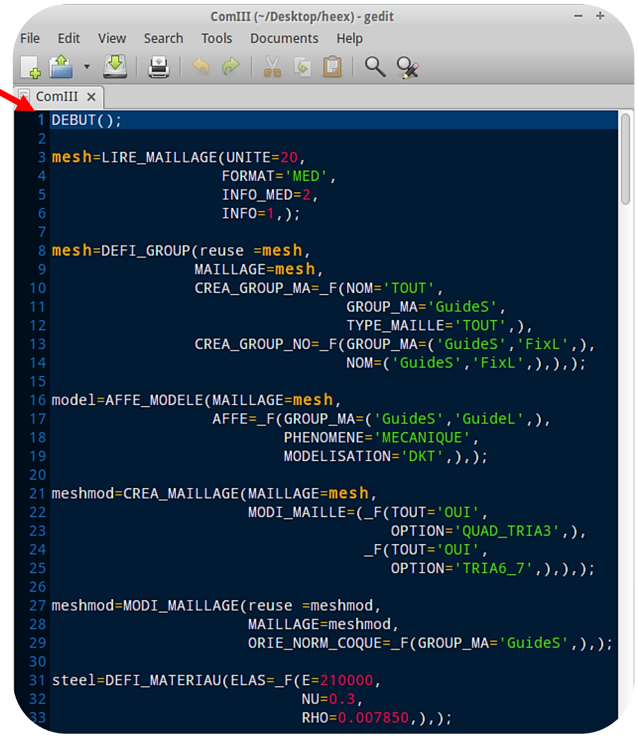
Creation of a Command File

- With a Wizard.
    - Available in Salome_Meca only.
    - 4 different wizards in Salome_Meca 2013.1.
        - Linear Elastic Analysis.
        - Modal Analysis.
        - Linear Thermal Analysis.
        - Crack Analysis.
- With the graphical command file editor (Eficas App).
    - Provided with Salome_Meca.
    - Provided with Code_Aster standalone.
    - Cannot handle python control flow instructions (if, while, for).





With your favorite text editor (ex. Gedit,..).



Data Setting and Command Language

- # Command file is also a Python script.
  - However, we should only focus on the Code_Aster commands.
- # Command file composes of a sequence of Code_Aster specific commands.
  - Each specific command composes of keywords and defines, assigns or uses data as input.
- # Most of the commands produce “concepts”.
  - On the left side of the equal sign (=).
  - The concepts generated by one command can be used as an input to the following command.
- # Command file contains no geometry description.

- # Examples:
  - steel = DEFI_MATERIAU (ELAS=_F(E=210000, NU=0.3, RHO=0.007850,) ,);

Command

Argument of the Simple Keyword

Factor Keyword

Name of the concept, User choice

Simple Keyword

  - IMPR_RESU (RESU=_F (RESULTAT = RESU),);

Factor Keyword

Command

Argument of the Simple Keyword

No Concept is Produced

Simple Keyword

  - “reuse” keyword is used to extend an existing concept.

    mesh = LIRE_MAILLAGE (FORMAT=’MED’)
    mesh = DEFI_GROUP (reuse = mesh,
    MAILLAGE = mesh,
    CREA_GROUP_NO = _F (GROUP_MA = ‘upper’,) ,);

Reuse of the MA Concept Defined by LIRE_MAILLAGE

Indicator of a Factor Keyword

  - Since commands are sequential, a concept must be created before being used.

    MESH = LIRE_MAILLAGE ();
    STEEL = DEFI_MATERIAU (ELAS = _F (E=20500e06, NU=0.3,) ,);
    CHAMT = AFFE_MATERIAU (MAILLAGE=**MESH**,
    AFFE = _F (TOUT=’OUI’,
    MATER=**STEEL**,) ,);

Salome's Possibilities

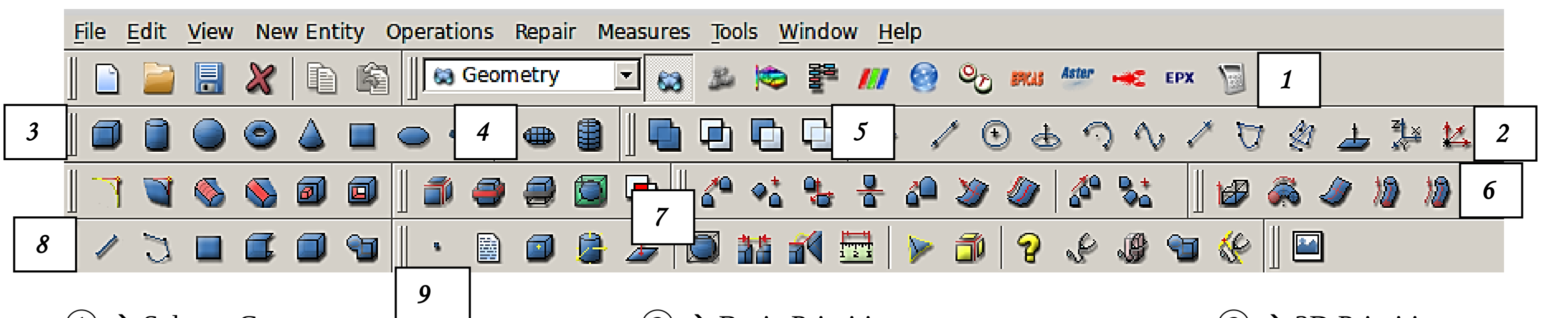
Reference --> EDF, *Code_Aster and Salome_Meca course material.*

What can we do with Salome:
- Import and export, repair and clean, create and modify geometries (CAD).
- Mesh, quality control, import/ export meshes.
- Manipulate the physical and numerical properties of a geometry.
- Manage the various stages of a computation: receive data, setting up a calculation, return results.
- Run the computation sequences and coupling between solvers.
- View and post-process the results.

Salome, GEOM Module:
- Design of geometric objects.
- Import/ export of objects of different CAD formats.
- Repairs and correction of CAD models.
- Adaptation of CAD models for computer simulation.
- Based on the Open Cascade technology.

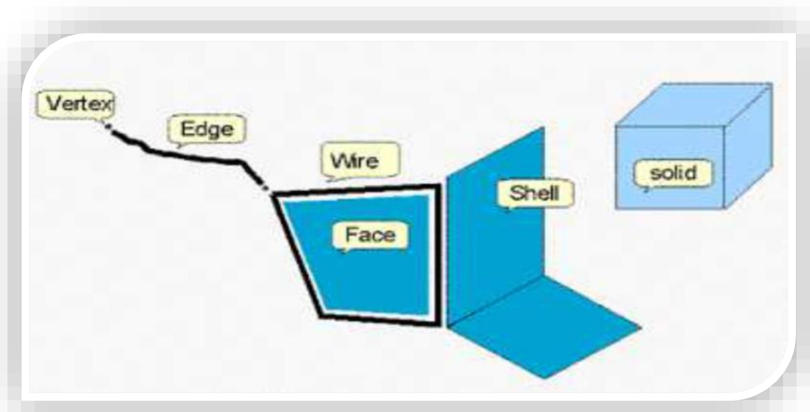
Salome, GEOM Module: Import & Viewers
- Various import/ export formats: STEP, IGES, BREP, STL, ACIS.
- Graphic functionalities: Transparency, coloring, shading/ wireframe, rotation, zoom, pan.

Salome, GEOM Module: GUI



① → Salome Components. ② → Basic Primitives. ③ → 3D Primitives.
④ → Advanced Primitives. ⑤ → Boolean Operations. ⑥ → Generations.
⑦ → Transformations. ⑧ → Constructions. ⑨ → Measures.

Salome, GEOM Module: Terminology
- Vertex.
- Edge.
- Wire (set of edges).
- Face (constructed from a closed wire).
- Shell (set of faces).
- Solid (constructed with a second shell).
- Compound (set of any types).



Salome, GEOM Module: Conception
- Geometric Primitives
  - 1D: Line, circle, ellipse, arc, curve, vector.
  - 2D: Plan, working plan.
  - 3D: Parallelepiped, cylinder, sphere, torus, cone.
  - Sketch 2D/3D: Construction of complex lines or surfaces.
  - Construction of elementary objects: Vertex, edge, wire, shell, solid, compound.
  - Advanced Primitives: Tee pipe.

  - Operations
    - Boolean: Merge, join, cut, intersect.
    - Fillet, chamfer, partition.
    - Explode: decomposition into basic objects.
  - Generations
    - Extrusion, revolution, filling, pipes.
  - Transformations
    - Translate, rotate, mirror, scale, offset.
    - Multi-translation, multi-rotation.

Salome, GEOM Module: Partitions and Objects
  - Define the relevant topological entities to ease mesh creation, calculation setup and post-processing result.
  - Geometric Groups: Manage geometric objects from which can be created element/ node groups required when assigning boundary conditions or material properties.
  - Geometric Operations to Define Partitions in a Mesh: Non-manifold geometry, for hexahedral meshing splitting non-hexahedral shape into hexahedral ones, define different refinements, merge.

Salome, MESH Module
  - The geometric model is imported from GEOM Module.
  - Information and quality control of meshes.
  - Groups of nodes or elements and operations on these groups.
  - Various import/ export formats: MED, UNV, STL, CHNS.
  - Modifications of meshes
  - Principles:
    - One algorithm for each dimension [1D: Wire Discretization].
    - One hypothesis for each algorithm [Nb. Segments].
  - The mesh calculation starts from the smallest to the largest dimension:
    - 0D → 1D → 2D → 3D.

Salome, SMESH Module: Algorithms
  - For each dimension, several algorithms are available:
    - 1D: Wire Discretization, Projection 1D…
    - 2D: Quadrangle (mapping), NETGEN 1D-2D, BLSurf…
    - 3D: Tetrahedrons (NETGEN), Tetrahedrons (GHS3D), Hexahedrons (i, j, k)…
  - Some algorithms are multidimensional:
    - 1D-2D: BLSurf, NETGEN 1D-2D…
    - 1D-2D-3D: NETGEN 1D-2D-3D…

Salome, SMESH Module: Hypothesis
  - The parameters of a meshing algorithm are set through a Hypothesis.
  - Ex. for 1D mesh:
    - Algorithm 1D: Wire Discretization.
    - Associated Hypothesis: Nb. Segments = 4, Equidistant Distribution.
  - Ex. for 2D mesh:
    - Algorithm 1D: Wire Discretization.
    - Associated Hypothesis: Nb. Segments = 4, Equidistant Distribution
    - Algorithm 2D: Quadrangle (mapping).

Salome, PARAVIS Module: Terminology
  - The Salome post-processing module based on ParaView.
    - With functionalities added by EDF.
      - MED interface.
      - Integration points.
      - Modal animation.

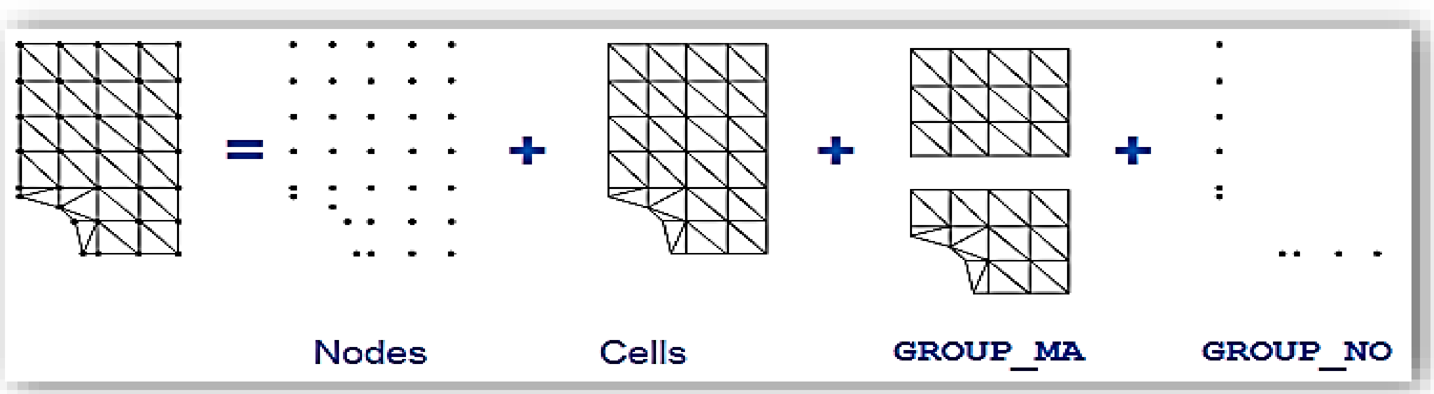
- Filters.
  - In ParaViS, the data is managed by means of filters.
    - Ex. Deformed Shape, cutting plane …
  - The filters depend on the type of data. They can be chained.
    - Initial data → Filters 1 → Filter 2 → …
- Views.
  - A surface on which you can see the data.
    - Ex. 3D, 2D, histogram, plot…
- Displays.
  - The data can be seen in different ways in different views.
    - Ex. Surface, Wireframe, PointSprite…

Code_Aster Possibilities

Special commands: DEBUT, FIN, POURSUITE
- The DEBUT command:
  - Begins execution, ignoring the previous lines.
  - Sets the logical units to be used for messages and files results.
  - Basic usage: **DEBUT ( )**
- The POURSUITE command:
  - Restarts execution from a base provided as an input.
  - Useful to continue a calculation initiated with the same version of Code_Aster.
  - Recommended to decouple the calculation of post-treatment.
  - Basic usage: Calculation: **DEBUT ( ) … FIN ( )**

    Post: **POURSUITE ( ) … FIN ( )**
- The FIN command:
  - Ends the command file and ends the run, ignoring the following lines.
  - Closes the base at the end of execution: folder containing all the concepts generated during the calculation (mesh data, intermediate structures, results).
  - Specifies the format used for the produced base: .hdf or Aster.

What is a Mesh?
- Coordinates of nodes.
- Cells defined by their connectivity.
- Groups of cells (GROUP_MA) and groups of nodes (GROUP_NO).



Read the Mesh

- Meshes at Aster format or MED formats:
  - Code_Aster format:
    - **mymesh = LIRE_MAILLAGE ( )**
  - MED format:
    - **mymesh = LIRE_MAILLAGE (FORMAT = 'MED')**
- Meshes of other formats:
  - Commands PRE_***: PRE_GIBI, PRE_IDEAS, PRE_GMSH.
  - Ex. **PRE_*** ( )**

    mymesh = LIRE_MAILLAGE ( )
- Results:
  - Command **IMPR_RESU**
  - For instance in MED format: **IMPR_RESU (FORMAT = 'MED',**

    RESU = _F (RESULTAT = myresult,) ,);

The Choice of Finite Elements

- A finite element is:
  - A geometric description, provided by the mesh.
  - Shape functions.
  - Degrees of freedom.
- Its choice determines:
  - The equations that are solved.
  - Discretization and integration hypothesis.
  - The output fields.
- 3D elements examples:
  - **MODELISATION = '3D'**
    '3D_SI'
    '3D_INCO'
- 2D element examples (plane strain, plane stress, axi-symmetry):
  - **MODELISATION = 'D_PLAN'**
    'C_PLAN'
    'AXIS'
- 1D element examples (beams, pipes, cables):
  - **MODELISATION = 'POU_D_T'**
    'TUYAU'
- Shell elements examples (plane strain, plane stress, axi-symmetry):
  - **MODELISATION = 'DKT'**
    'DST'
    'COQUE_3D'

But also many other mechanical 3D, 2D, 1D elements like: Sub-integrated, incompressible, non-local, fluid-structure coupled THM….