Application internship

Implementing calculation methods in a digital engineering platform

Yann SUDRAT

Course supervisor: Edward Mark NORRIS
School contact: Nicolas BARRAL

November 16, 2023
ENSEIRB-MATMECA and OpenSPDM

Table of contents

Note: this English version of the report was translated from the master version in French using the DeepL translation application

I/ Introduction

1) Simulation Process Data Management (SPDM): Management of simulation data and processes

a) General

With the development of numerical simulation, various industrial sectors have been able to reduce their need for testing and prototyping by relying on the growing capabilities of numerical solvers. However, cutting-edge technologies and the diversity of sectors require highly specialized and varied capabilities, leading to a consequent increase in the number of software packages available and in use, reaching several hundred, sometimes within the same company. Calculation codes and scripts developed in-house for specific applications add to the complexity and diversity of simulation software.

Figure 1: Examples of simulation software used at Naval Group

Image credits [1]: Damien Laval, Naval Group 2021, Presentation Examples of Naval Group Use Cases

At the same time, industries are being held to ever higher standards in terms of process reliability, traceability and efficiency. The multiplicity of digital solutions and the growing amount of data to be processed make these requirements costly in terms of time and money. What's more, the compatibility of different software packages (whether competing or not) makes data exchange between different teams, companies or subsidiaries difficult to organize.

The aim of Simulation Data and Process Management (SPDM) is to establish a framework within which those involved in digital simulation, as well as managers and administrators, can carry out their duties, launch and manage projects, exchange with teams or customers, share and save data, etc. These working methods are embedded in a software interface linked to a secure database containing all the files relevant to the company's projects, from simulation files (models, meshes, etc.) to project reports and project management files (specifications, monetization, etc.). Numerical simulation processes consist of a chain of actions, the results of which depend on previous actions, so a graphical method is needed to represent and store all analysis attempts.

This not only keeps a permanent record of all projects carried out by the company, but also ensures data traceability by keeping a record of the various actions and modifications carried out on a project, as well as the people involved in these actions. Data retention becomes exponentially more important as physical testing is replaced by analysis-based certification.

Historically, SPDM is closely related to Product Lifecycle Management (PLM) and Product Data Management (PDM), which provide a framework for companies to manage the evolution, modification and characteristics of their products and manufacturing processes. SPDM applies this type of procedure not to physical products, but to computer data.

b) State of the art

Simulation data management solutions were developed in the late 1990s at Airbus and BMW, with the SimManager reference software. The data model was subsequently documented as a standard by the German Automotive Industry Association (VDA). A second software package, SimDataManager, was developed in line with VDA guidelines 4967. Many companies then tried their hand at SPDM software development, including major players in the field of digital simulation software development, such as Dassault Systèmes with the 3DEXPERIENCE platform, Ansys with Minerva and Siemens, with mixed results. Although large companies are largely interested in the subject (Toyota, Bosch, Airbus, Safran, ...), they are looking to implement SPDM in their respective frameworks, which are extremely varied. An SPDM solution must therefore either be developed with the aim of integrating exclusively with a particular company, or be particularly flexible and parameterizable to adapt to the different needs and particularities of each company.

Indeed, it is impossible for a company with several thousand engineers, dozens of teams and the need to process the equivalent of years of work in digital data to modify its entire structure to adapt to third-party software. Some SPDM software packages have encountered and continue to encounter this obstacle to large-scale implementation in their sector, not least because of their choice of data models, which are different from and less flexible than the existing standard.

2) OpenSPDM

OpenSPDM is an open-source SPDM software package, implementing the standardized SDM data model, which was then developed in collaboration with major manufacturers interested in a flexible, adaptable and customizable solution to their needs. OpenSPDM is based on a core set of functionalities common to all SPDM practices, which can then be extended by companies to meet their more specific needs.

It consists of a computer-installed part, which contains all the methods, scripts and tools required for its operation, and an Internet client, linked to a database. The latter is supported by and developed on Aras Innovator, a software suite published by Aras Corporation for PLM, collaborative development, project management, etc. Aras Innovator enables the development of a fully customizable Internet client, by creating personalized objects, parameters relevant to these objects, relationships between these objects, etc.

The common core is made up of 3 types of objects, each with its own properties and interacting with the other two:

- The "Study" object is the main object of a project. As its name suggests, it represents a simulation study being carried out by the company. It contains several parameters that identify it, such as its name or number, its owner in the database and its purpose. It also contains the project's input and output data (geometry, specifications, final results, etc.).
- The "Task" object is included in a "Study". It enables the study to be broken down into a succession of steps to be carried out by one or more engineers, such as "Meshing", "Post-Processing" or "Resolution". Each "Task" is associated with a name, the action to be performed, the tool to be used, and the input and output data (if any) for the task.
- The "Artifact" object, which acts as a data container. It is in this type of object that the input and output data for "Studies" and "Tasks" are stored. In this way, each "Study" and "Task" is linked to two "Artifacts", one for input and the other for output. Artifacts" have a parameter that identifies the type of data stored in them (geometry, document, image, etc.).

In this way, each project corresponds to a "Study" containing a series of "Tasks", interspersed with "Artifacts" containing important data from the study. This structure offers several advantages.

Firstly, it enables you to keep track of all the actions carried out at each stage of a project, who carried them out, when they were carried out, etc. This greatly facilitates data traceability, preventing some of them from being lost, while enabling them to be accessed by people other than their authors.

Secondly, it enables tasks to be visualized in the form of a graph, making it easier to understand and manage the project, and to visualize the work done and remaining (see Figure 2).

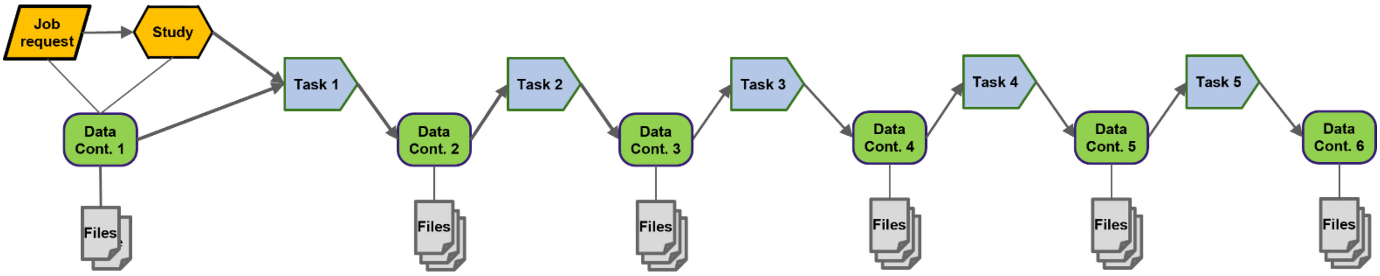


Figure 2: Study graph

Image credit [2]: Edward Mark Norris, A standardized approach to building CAE application connectors to SPDM solutions, NAFEMS World Congress 2021

Thirdly, it makes it easier to make changes or branches in the study (see Figure 3). For example, if at the end of a study the results are inconclusive, it is very easy to go back to a previous task and create a branch to modify certain parameters (of the mesh, for example). This variant of the study can then be taken to resolution, and the operation repeated until satisfactory results are obtained. This methodology makes it possible to keep track of all the work carried out, including that which was unsuccessful. Note that comparative studies are particularly concerned by this methodology (see Part III).

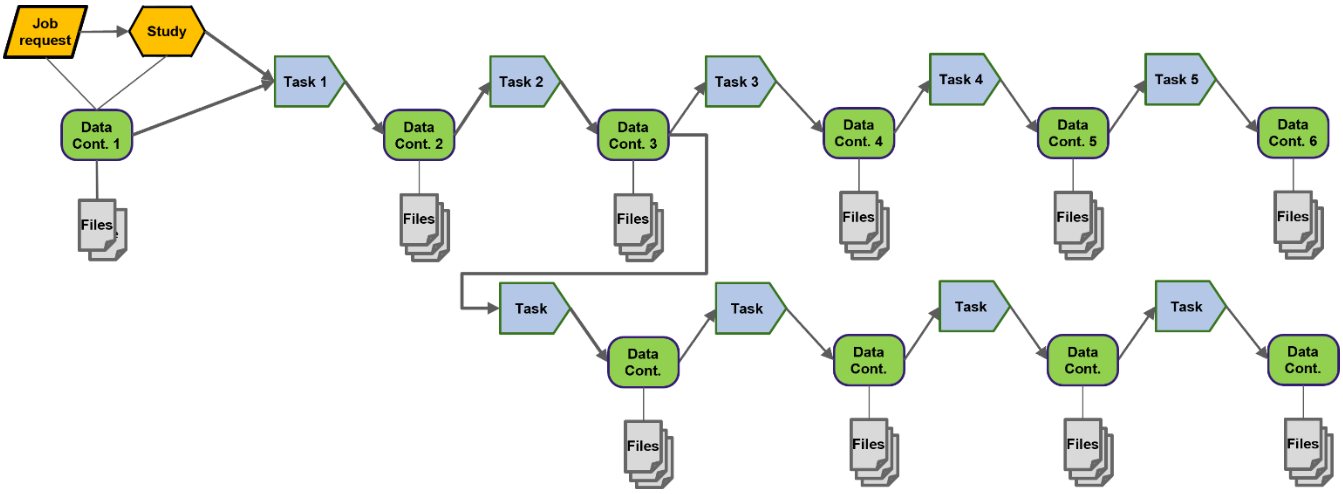


Figure 3: Plot of a study with a connection

II/ Connector development

1) General

If the compatibility of numerical simulation software is already a problem when it comes to integrating it into a given framework, it's even more difficult to make the same software compatible with all the numerical solutions available. SPDM software does just that, by enabling engineers to efficiently couple their data with the various software packages they need. An SPDM solution should also not use the conventional file system as an intermediary, since this would not solve the problems of storage, organization, traceability and accessibility of current systems. It is therefore necessary to automate data transfers between the SPDM interface and a given simulation software. Such a tool is called a connector.

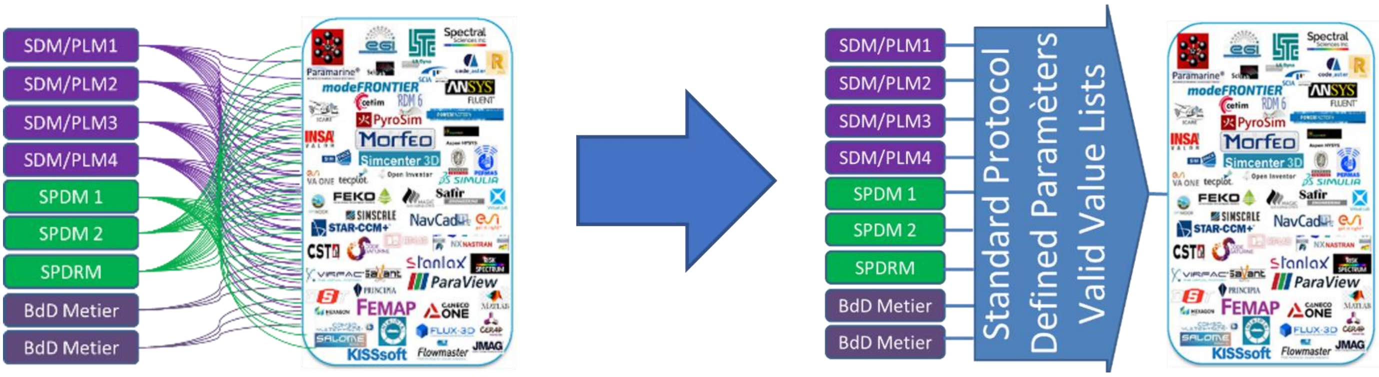


Figure 4: Connectors for interfacing SPDM/PLM and simulation software

Image credits [1]: Damien Laval, Naval Group 2021, Naval Group Use Case Examples presentation

The connectors required differ from company to company and evolve over time, depending on the licenses available and the company's choices. It is therefore essential that all connectors operate according to identical protocols, with common parameters defined and valid (see Figure 4).

In OpenSPDM, connectors work as follows: an application, launched locally, establishes a link between the Internet client and the machine via Web Sockets. Thanks to this communication, instructions given from the client can be used to launch command files dedicated to each tool. The parameters of the action performed are written to a configuration file and read by the connectors.

To save the work done, a Python script uses the asyncio library, which allows you to create an asynchronous sub-process, which exists in parallel with the main script and can be launched from another Python script. The task of this sub-process is to launch the upload tool dedicated to the Aras Innovator database.

2) Abaqus CAE connector

Abaqus CAE is the graphical user interface for the Abaqus solver (Standard and Explicit), developed by Dassault Systèmes. The automation language used by Abaqus CAE is Python 2.7, accompanied by libraries dedicated to Abaqus commands. To integrate the commands needed to use Abaqus CAE with OpenSPDM into the software interface, we decided to create an Abaqus plugin, i.e. a set of Python files read by the software when it starts up. These files generate a number of buttons, each launching a different function (Figure 5).

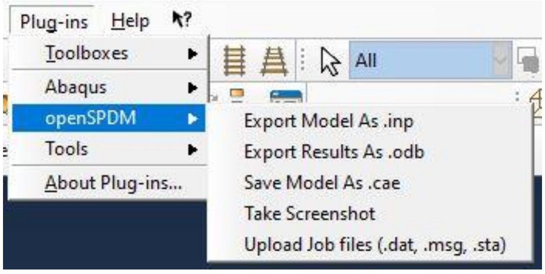


Figure 5: Abaqus CAE plugin menu in the software interface

An Abaqus CAE plug-in consists of a "core" file containing all the functions launched by the buttons, and a file containing all the information relating to the buttons: name, accessibility, location in the menu, function to be launched, core file, etc. (see Figure 6).

The following functionalities were developed during this project:

1. Export in .inp format :

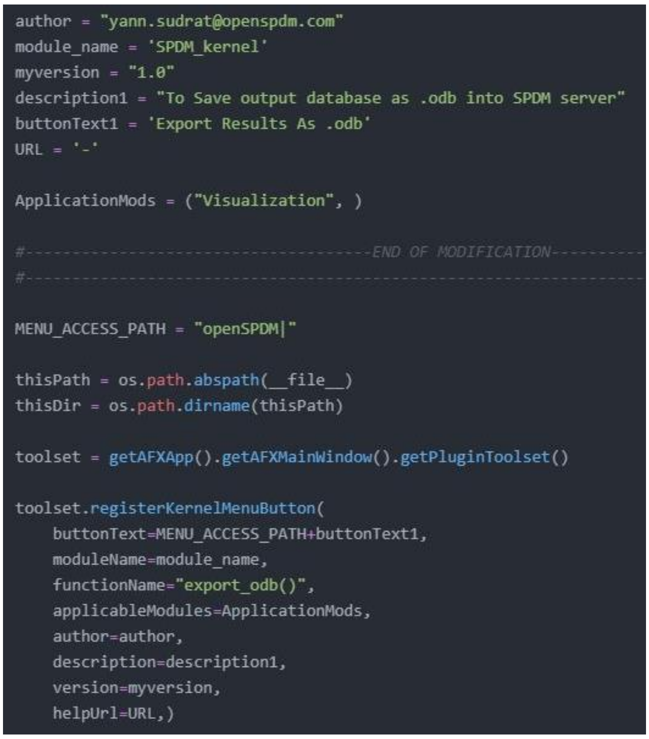
   The inp (input file) format is characteristic of Abaqus solver input files. It differs from .cae model files in that it contains only the data required by the solver (mesh, initial conditions, etc., but not the geometry, for example). It is much smaller, which simplifies storage and distribution (if calculations are run or checked by another person, for example).
   This function allows you to export the work performed in this format and upload it to the database, together with the task's output artifact.

2. Export in .cae format :

   The .cae format contains the complete database of a model processed by Abaqus CAE, including geometry, partitioning, assembly, etc. For the heaviest models, this file can become very large and costly in terms of computer resources. However, it is in this file that all modifications made in the graphical interface are saved, so it is essential to be able to upload this file to the database.

3. Export results in .odb format :

   After a calculation has been solved, all the requested results are stored in an .odb file (output database). Engineers will use this file to determine whether the results are satisfactory, or whether another iteration is necessary, so it must be uploaded.
   This function can only be launched in the "Visualization" module, as the file must already exist.

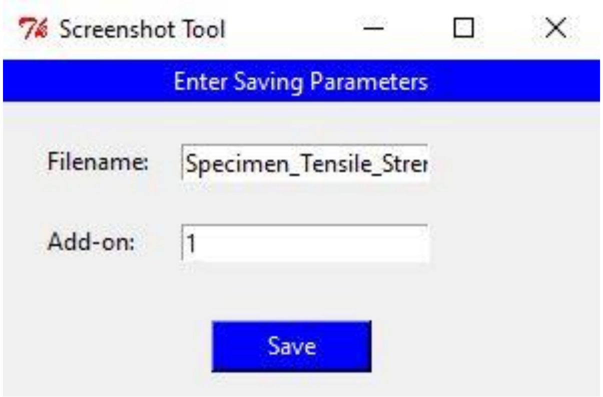


```
author = "yann.sudrat@openspdm.com"
module_name = 'SPDM_kernel'
myversion = "1.0"
description1 = "To Save output database as .odb into SPDM server"
buttonText1 = 'Export Results As .odb'
URL = '-'

ApplicationMods = ("Visualization", )

#------------------------------------END OF MODIFICATION----------
#-----------------------------------------------------------------

MENU_ACCESS_PATH = "openSPDM|"

thisPath = os.path.abspath(__file__)
thisDir = os.path.dirname(thisPath)

toolset = getAFXApp().getAFXMainWindow().getPluginToolset()

toolset.registerKernelMenuButton(
    buttonText=MENU_ACCESS_PATH+buttonText1,
    moduleName=module_name,
    functionName="export_odb()",
    applicableModules=ApplicationMods,
    author=author,
    description=description1,
    version=myversion,
    helpUrl=URL,)
```

Figure 6 File containing information on one of the functions, export and save output file

4. Screenshots and animations :

   To justify and illustrate the results, screenshots and animations can be saved and uploaded directly from the interface. A graphical interface has been developed using the TKinter Python library to ensure the uniqueness of saved screenshot and animation names (Figure 7).



Filename: Specimen_Tensile_Strer

Add-on: 1

Figure 7: Graphical interface developed for Abaqus CAE screen capture

5. Exporting control files :

   When solving a problem, Abaqus writes important steps in the solution, errors encountered, warnings, etc. to control files. These functions are essential to ensure that the simulation runs correctly and that the results obtained are consistent and satisfactory.

6. Export geometry in .stp format :

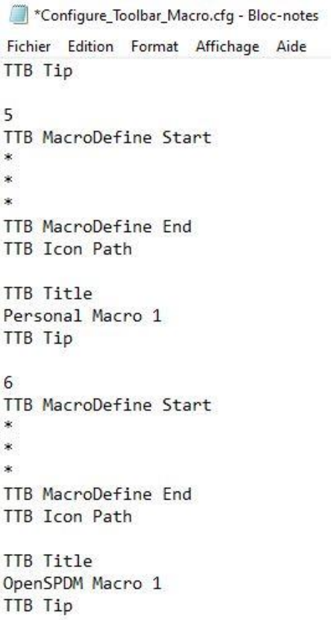
   If Abaqus CAE is to be used to modify a geometry, but another solver or mesher is to be used afterwards, it must be possible to save and export the modifications made in a standardized format that can be read by most solvers. The STEP format (.stp or .step) meets these criteria.

All these procedures work on the same basis: Python scripts save the requested file locally, before reading a previously established configuration file to upload the data to the right place in the database. A diagram of how the Abaqus connector works is provided in Appendix A.

3) LS-PrePost connector

LS-PrePost is the pre- and post-processing software linked to the LS-Dyna explicit solver, developed by Ansys. Its automation language is specific to the software, derived from C. Where Abaqus CAE works on a plug-in principle, LS-PrePost allows you to develop and load "Macros", i.e. sequences of commands in the LS-PrePost language, which are executed when the macro is launched. These macros are read from a file (extension .cfg) provided for this purpose, and are displayed on the software interface in a customizable toolbar, also characterized by a file (extension .ttb).

All macros are written one after the other, and have several parameters: an associated number, the commands to be performed, an icon (optional) to be displayed, a title and a short description (see Figure 8).


*Configure_Toolbar_Macro.cfg - Bloc-notes
Fichier Edition Format Affichage Aide

```
TTB Tip

5
TTB MacroDefine Start
*
*
*
TTB MacroDefine End
TTB Icon Path

TTB Title
Personal Macro 1
TTB Tip

6
TTB MacroDefine Start
*
*
*
TTB MacroDefine End
TTB Icon Path

TTB Title
OpenSPDM Macro 1
TTB Tip
```

*Figure 8: Macro file construction, with * representing command positions*

The macros developed during this project are similar to those developed for the Abaqus connector, and are listed below. In fact, to have a minimum viable product, only two functions need to be developed: saving modeling files and screen capture. OpenSPDM has been designed so that all other functions can be added by engineers specializing in the software concerned, since they are already familiar with software automation.

1. Export in .k format :

   The .k (key file) format is unique to LS-Dyna and is the analogue of the .inp file for Abaqus, in that it contains only the mesh (position of nodes and elements) and the keywords that enable the problem to be modeled and put into data. Its advantages are therefore the same as those mentioned above, and the connector must be able to handle this type of file.

2. Export in .proj format

   The .proj format, again unique to LS-Dyna, is the equivalent of the .cae file and contains all the data for the current project, and it is in this type of file that modifications made by the graphical interface can be saved. If a task takes several days to complete, it is this file that is saved by the engineers.

3. Screenshot:

   To justify the progress and results of a task or simulation, the LS-PrePost connector also features screen capture.

4) Coreform Cubit connector

Coreform Cubit is a meshing software program specializing in hexahedral element meshing, with numerous tools for processing CAD (Computer-Aided Design) models, automatic meshing, ...

Cubit is automated by means of Python scripts (version 3.9 for this project), made accessible in the interface by menu configuration files that allow custom options to be added to the native menus (see Figure 9).

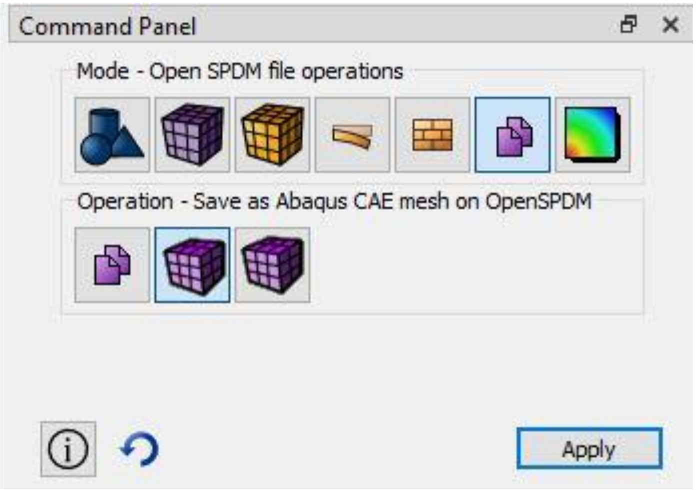

Command Panel
Mode - Open SPDM file operations
Operation - Save as Abaqus CAE mesh on OpenSPDM
Apply

Figure 9: OpenSPDM custom menu for Cubit

As Cubit is primarily a meshing program, and less concerned with modeling edge and initial conditions and solving (although some options are available), a necessary feature not present in other connectors is the ability to export meshes in formats readable by solvers.

1. Export in .cub5 format:

   The .cub5 format is the standardized format used by Coreform Cubit to store current project data (geometry, modifications, etc.), like the .cae and .proj extensions. The arguments justifying the need for such a function are therefore the same as before.

2. Export meshes :

   Cubit is designed for use with a large number of numerical solvers, including Abaqus and LS-Dyna, as well as NASTRAN. Cubit's native functionality therefore includes the ability to export meshes in the extensions required by these different solvers. In the current version of the project, the connector can only export to the .inp and .k formats used by Abaqus CAE and LS-Dyna respectively.

III/ Numerical simulation with openSPDM: an example

1) Mesh convergence study

a) Case studies

The case study considered here is a simple tensile test on a specimen standardized by ISO-3167 [3], made of 7075 aluminum. The stress-displacement response will be plotted for 3 different, increasingly refined mesh sizes, to measure the impact of mesh quality on certain characteristic values of metals, such as yield strength. The elastic and plastic behavior of the specimen will be modeled.

This entire study will be carried out within the OpenSPDM framework.

b) Job chart

The study is divided into tasks as shown in Figure 10. The geometry and its idealization is common to the whole study, but branches into three different variants at the meshing stage, with each branch representing a different mesh. The steps for each branch after meshing are strictly identical.

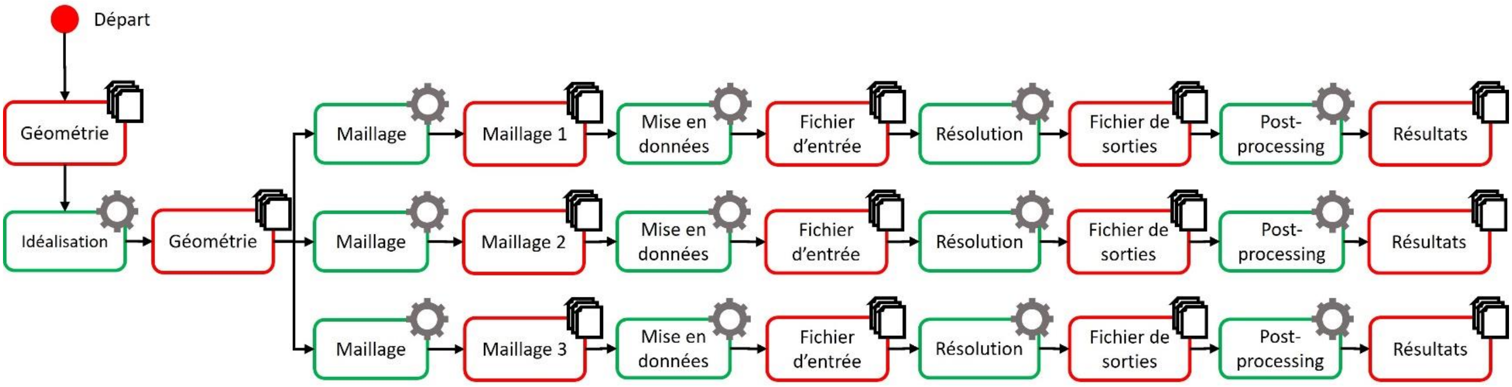


Figure 10: Graph of mesh convergence study tasks

c) Geometry and material

The geometry considered is shown in Figure 11, these dimensions being standardized by ISO-3167. In order to simplify the mesh and improve its quality, this geometry is partitioned into simpler blocks shown in Figure 12.

In order to obtain good results, we plan to use hexahedral elements. For such elements, the parts of this geometry that are difficult to mesh are the 4 fillets that frame the thin part of the specimen. The rest of the partition is made up of rectangles. This makes it easier for the mesher to process these parts.

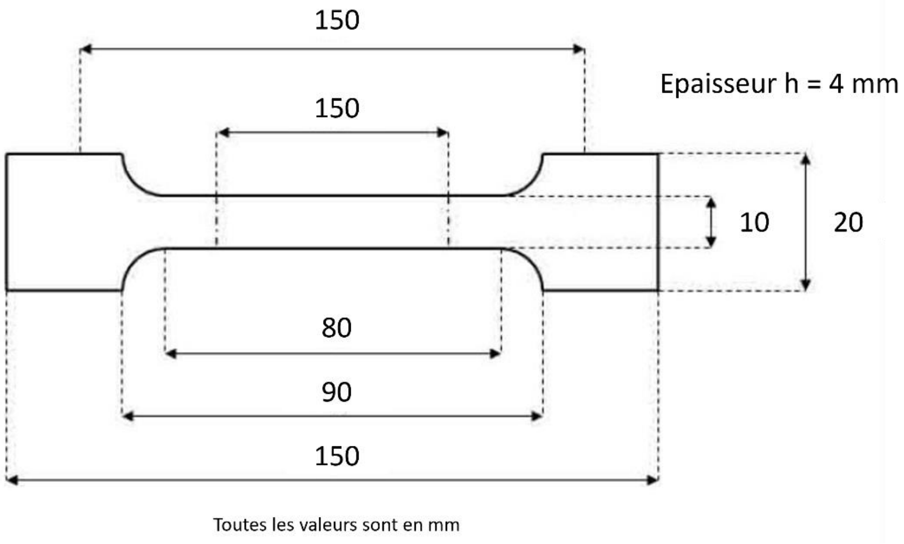


Figure 11: Diagram of the test specimen in question

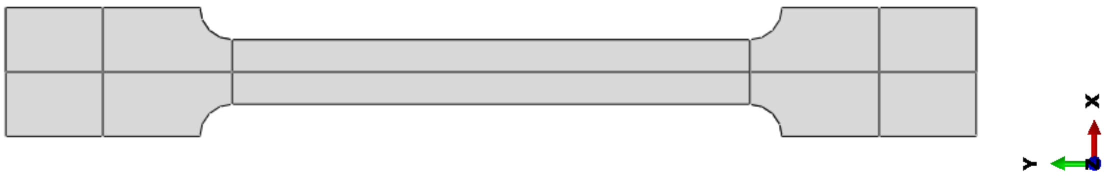


Figure 12: Partitioning the specimen

The material chosen is a 7075 aluminum alloy, widely used in the aeronautics industry for its good compromise between weight, flexibility and rigidity, whose characteristics must therefore be perfectly known. The alloy's composition and some of its mechanical characteristics are listed in Table 1 [4][5] below. The alloy is isotropic, with an elasto-plastic behavior law following a Johnson-Cook law, the formulation of which is given in Appendix B.

Aluminium 7075			
Composition		**Mechanical properties**	
Aluminum	87.1 - 91.4 %	Density ρ (g.cm)$^{-3}$	3.0
Zinc	5 - 6 %	Young's modulus E (MPa)	71000
Magnesium	2.1 - 2.9 %	Poisson's ratio ν (∅)	0.32
Copper	1.2 - 2 %	**Johnson Cook coefficients**	
Other components (Si, Ti, ...)	0 - 0.5 %	A (MPa)	546
		B (MPa)	678
		C	0.024
		n (∅)	0.71
		m (∅)	1.56

Table 1: Mechanical and chemical properties of aluminium alloy 7075

d) Meshes

As mentioned previously, the elements used will be hexahedral elements. However, as the thickness is negligible compared to the other dimensions (see Figure 11), we have chosen to model our problem with "Continuum Shell" elements, which enable us to reduce the time and convergence of the calculation. In particular, given the absence of bending and shearing in the model, and therefore of hourglassing and shear-locking, the elements used will be **SC8R** elements (see Figure 13). This is a Shell Continuum element with **8** nodes and Reduced, i.e. it has only one integration point at its center.

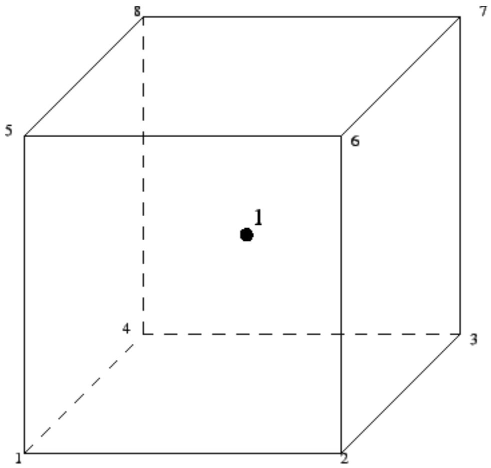


Figure 13: SC8R element

Image credits: MIT [6]

The 3 mesh sizes considered are: 5mm, 2mm and 1mm. We thus obtain the following meshes (Figure 14).

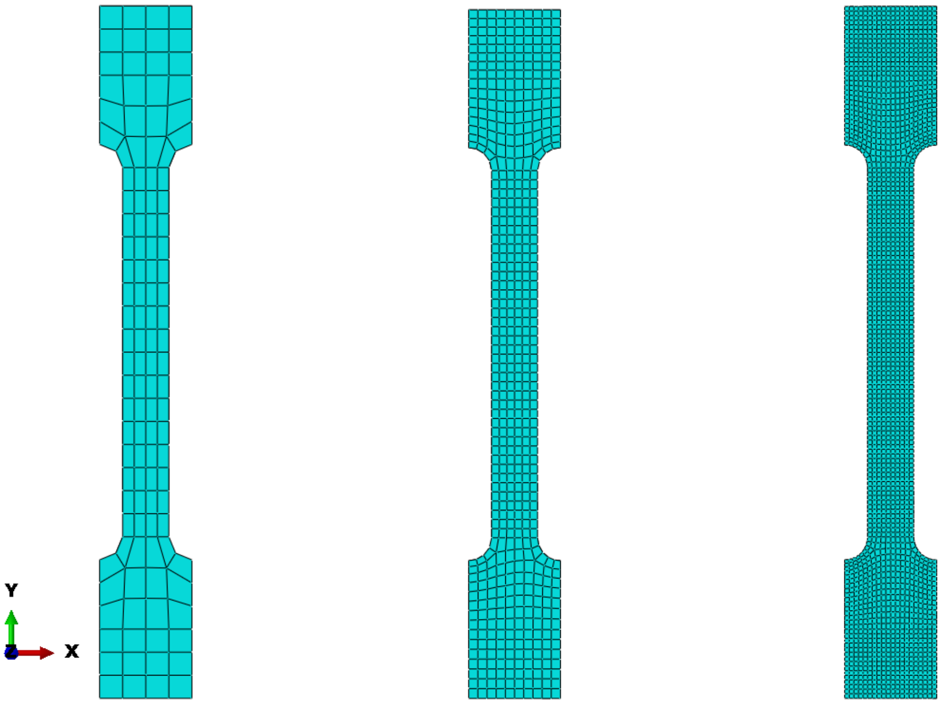


Figure 14: From left to right, mesh sizes 5mm, 2mm, 1mm

e) Loading cases

We simulate the presence of movable jaws by blocking displacements in the X and Z directions while imposing displacements of + and - 10 mm on the ends (Figure 15). This value of 10 mm was chosen in the knowledge that the maximum elongation at break of aluminum is around 33%, which in our case corresponds to 33% of 80mm, or 26.4 mm. As material damage is not modeled, stricture and fracture phenomena cannot occur, and we must not reach this displacement limit.

Assuming that the tensile force applied to the specimen is sufficiently slow, a quasi-static progression is assumed. Our model will then contain only one static simulation step, the parameters of which are given in Table 2. These parameters have been chosen to obtain a sufficient number of images for smooth animation.

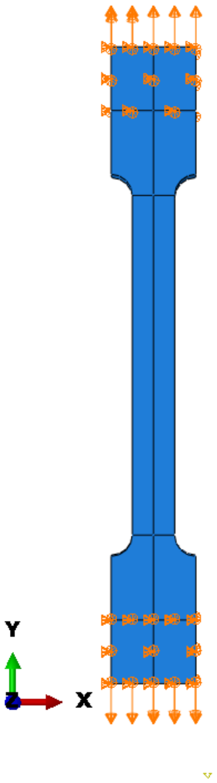


Figure 15: Specimen loading case

Static stage	
Number of iterations limit	100
No initial time	0.01
No minimum time	1e-9
Maximum time step	0.05

Table 2: Static stage parameters

f) Post-processing, results and conclusion

For the Von Mises constraint, the following results are obtained (Table 3):

Mesh size (mm)	Von Mises stresses (MPa)
1	790,2
2	791,2
5	805,7

Table 3: Von Mises constraints as a function of mesh size

This gives us the following graph (Figure 16):

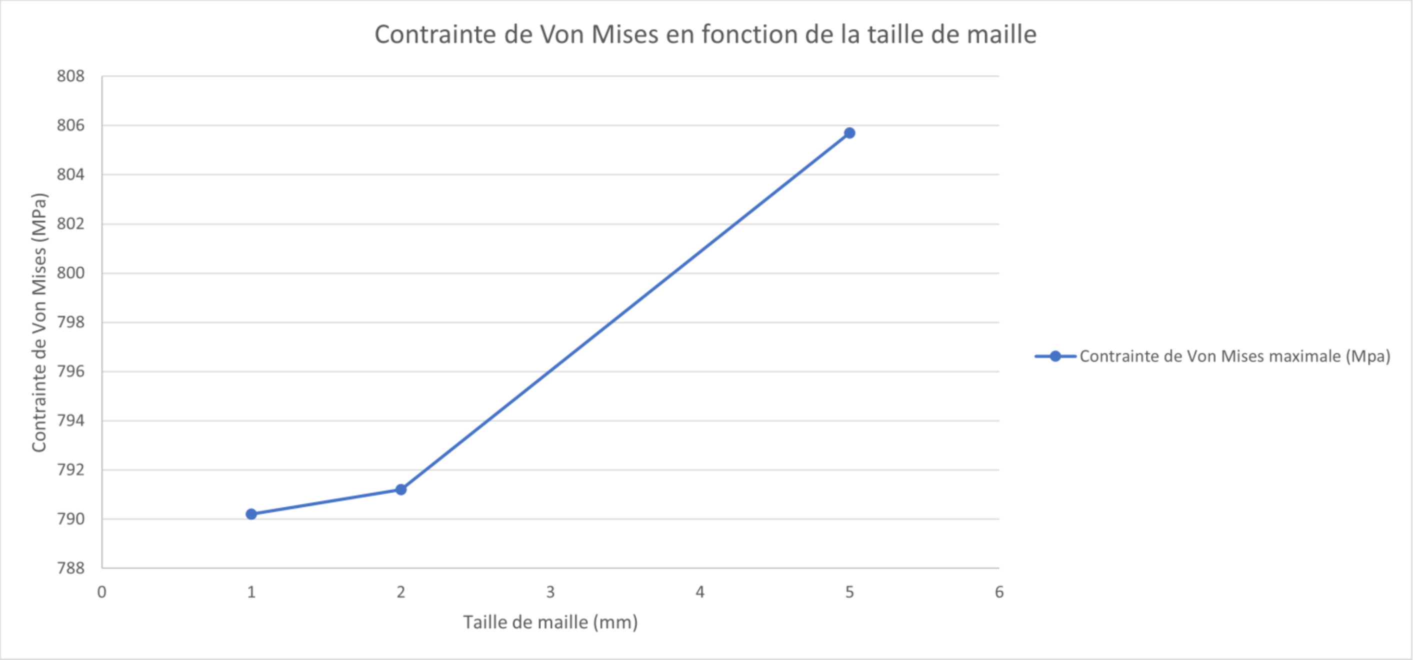


Figure 16: Von Mises stress curve as a function of mesh size

The graph above shows that the Von Mises stress value converges to around 790 MPa at the end of displacement. This high value is due to a maximum displacement that approaches the phenomenon of striction without actually modeling it. However, this convergence confirms the validity of the model, while offering an optimal mesh size between number of elements (and therefore calculation time) and accuracy around 2mm. Indeed, increasing the number of elements beyond this value brings only negligible changes.

The stress-displacement responses for the three cases are then plotted below (Figures 17, 18 and 19).

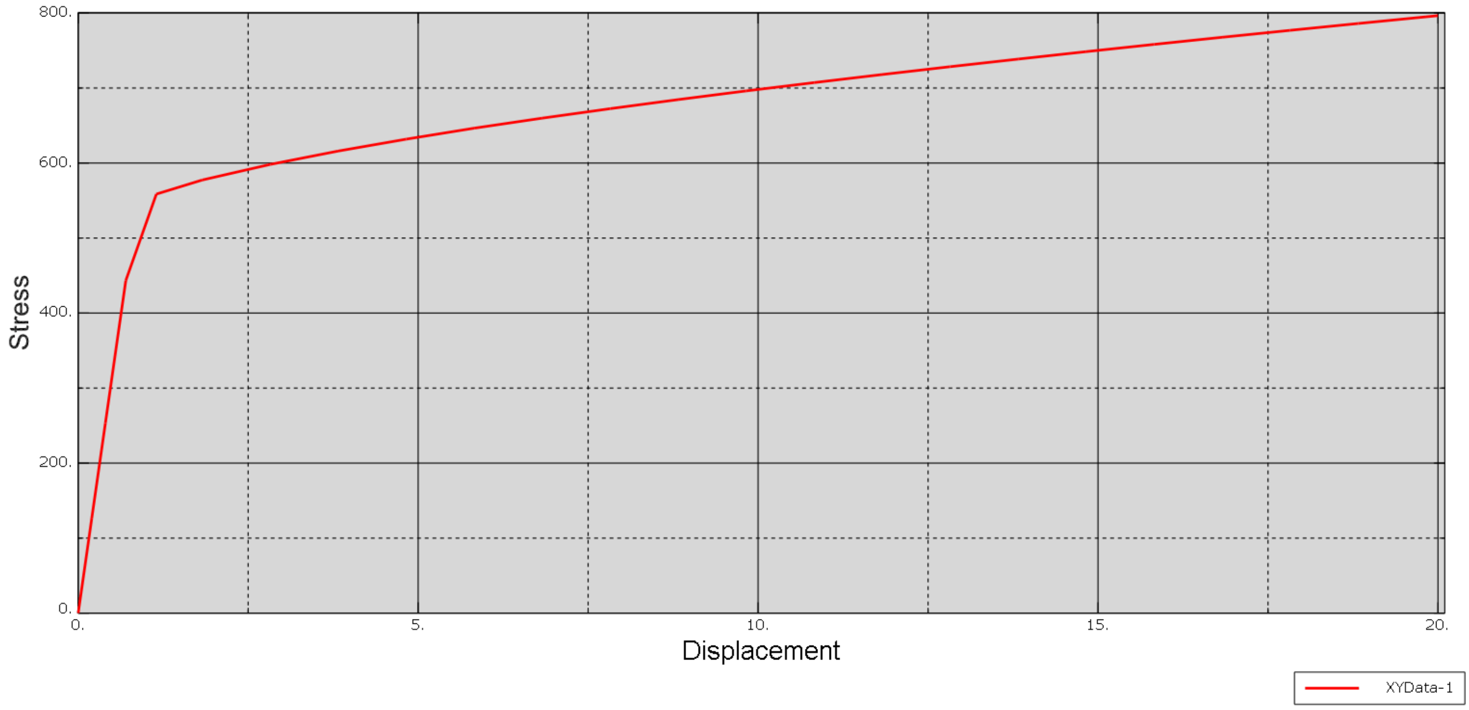


Figure 17: Displacement stress response for a 5mm mesh size

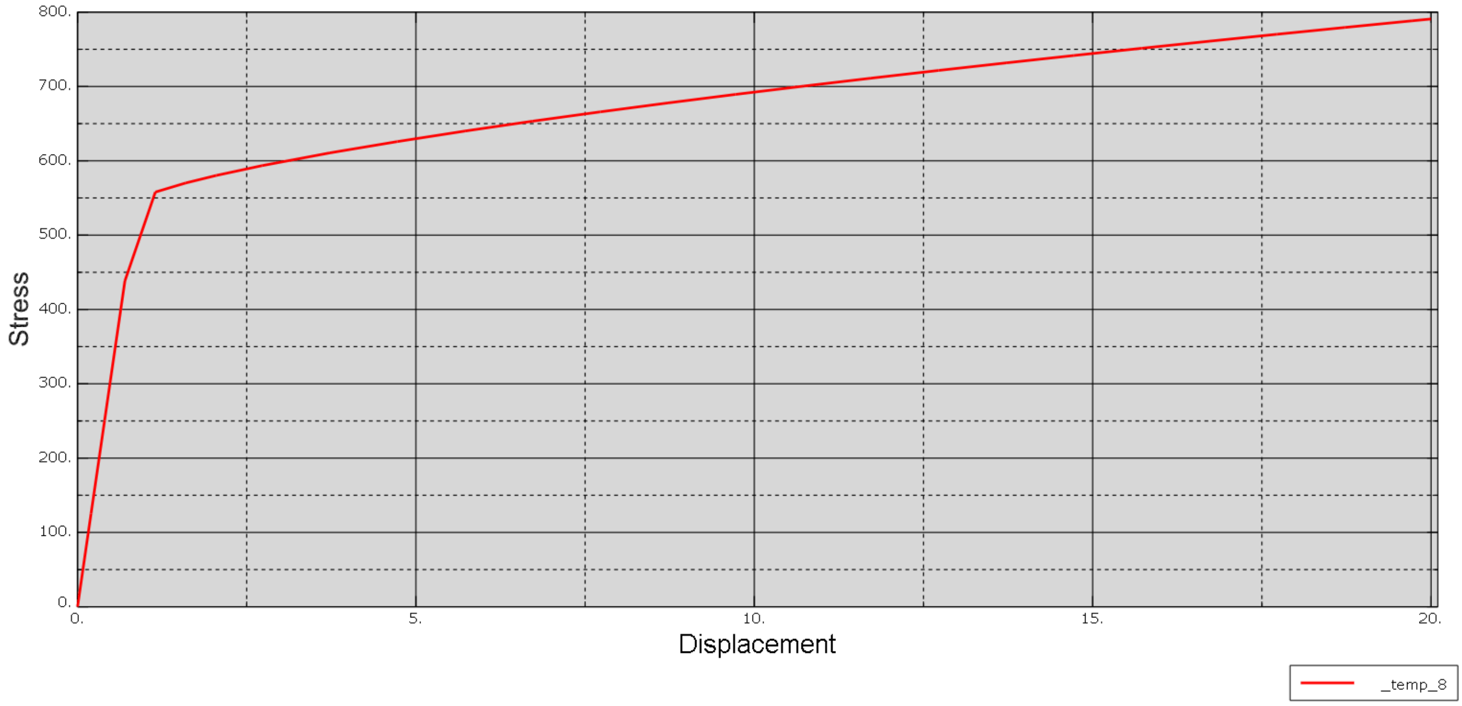


Figure 18: Displacement stress response for a 5mm mesh size

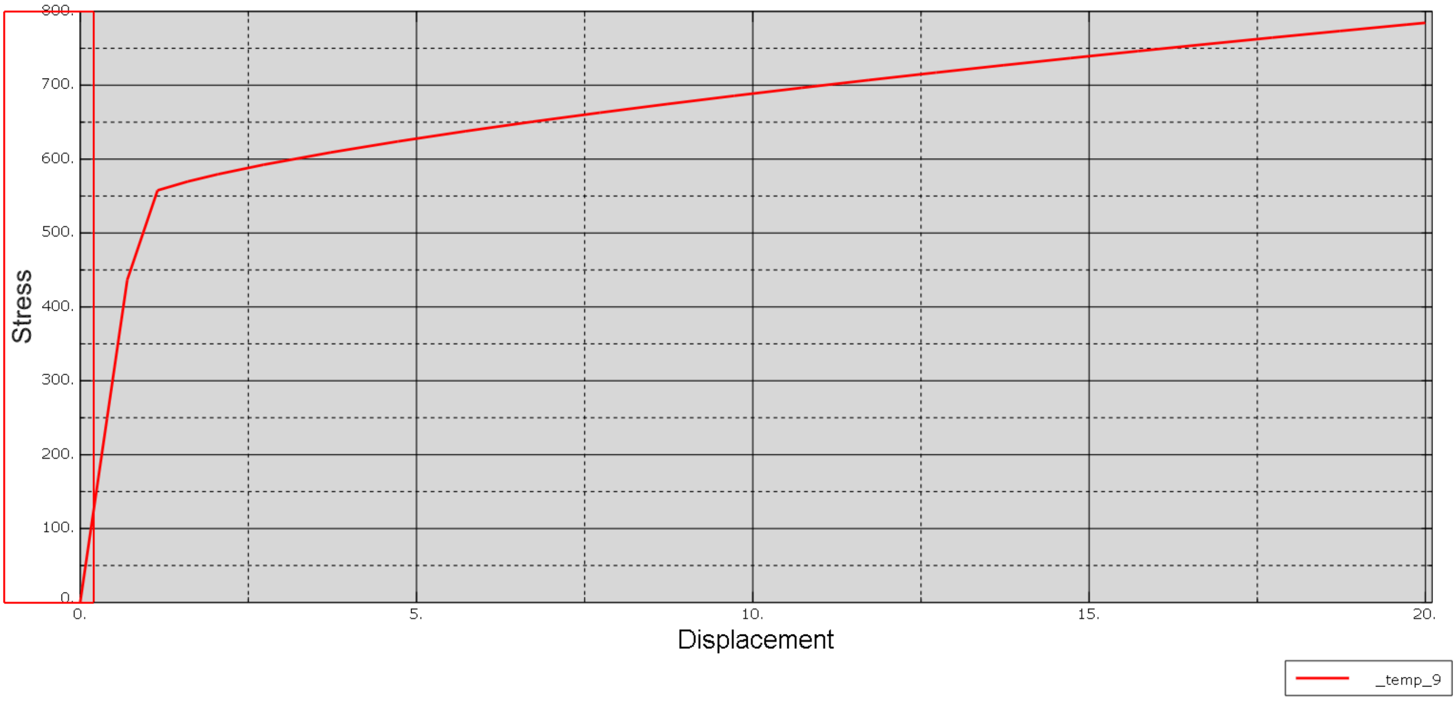


Figure 19: Displacement stress response for a mesh size of 1mm

These 3 figures have many similarities:

- Their overall evolution, where they all correspond to tensile graphs with a linear part representing the elastic part, and the plateau characteristic of plasticity.
- Their values are very close, which is to be expected given the small differences in Von Mises stresses noted above.

These figures once again validate the coherent modeling of our problem. Certain material parameters, such as yield strength, allow us to take the verification a step further. By noting these limits for each case and comparing it with a reference value [5], we can draw up the following table (Table 4).

Mesh size (mm)	Yield strength (MPa)
1	≃ 440
2	≃ 440
5	≃ 440
Reference value	480

Table 4: Yield stress values as a function of mesh size and reference value [5].

The simulation therefore matches the expected behavior of 7075 aluminum, once again validating the modeling and results.

2) Another example: solver comparison

As part of this project, an additional study not presented here was carried out to compare the performance of two different solvers (Abaqus and LS-Dyna) on the same geometry. This type of method, also known as "double blind simulation", is extremely important in fields where physical tests are unfeasible, such as nuclear weapons, oil and gas, etc.

IV/ Conclusion

SPDM tools such as OpenSPDM therefore enable companies to implement efficient, sustainable working methods that are highly customizable to their needs, while ensuring the preservation, reliability and traceability of their simulation data. The data model developed in the late 90s and reused by OpenSPDM is designed to adapt to all working methodologies, while leaving room for engineers to develop their own tools.

The choice of open source software for OpenSPDM encourages the wider adoption of SPDM methods, not just by large companies with the means to obtain expensive licenses, but also by universities and small and medium-sized businesses.

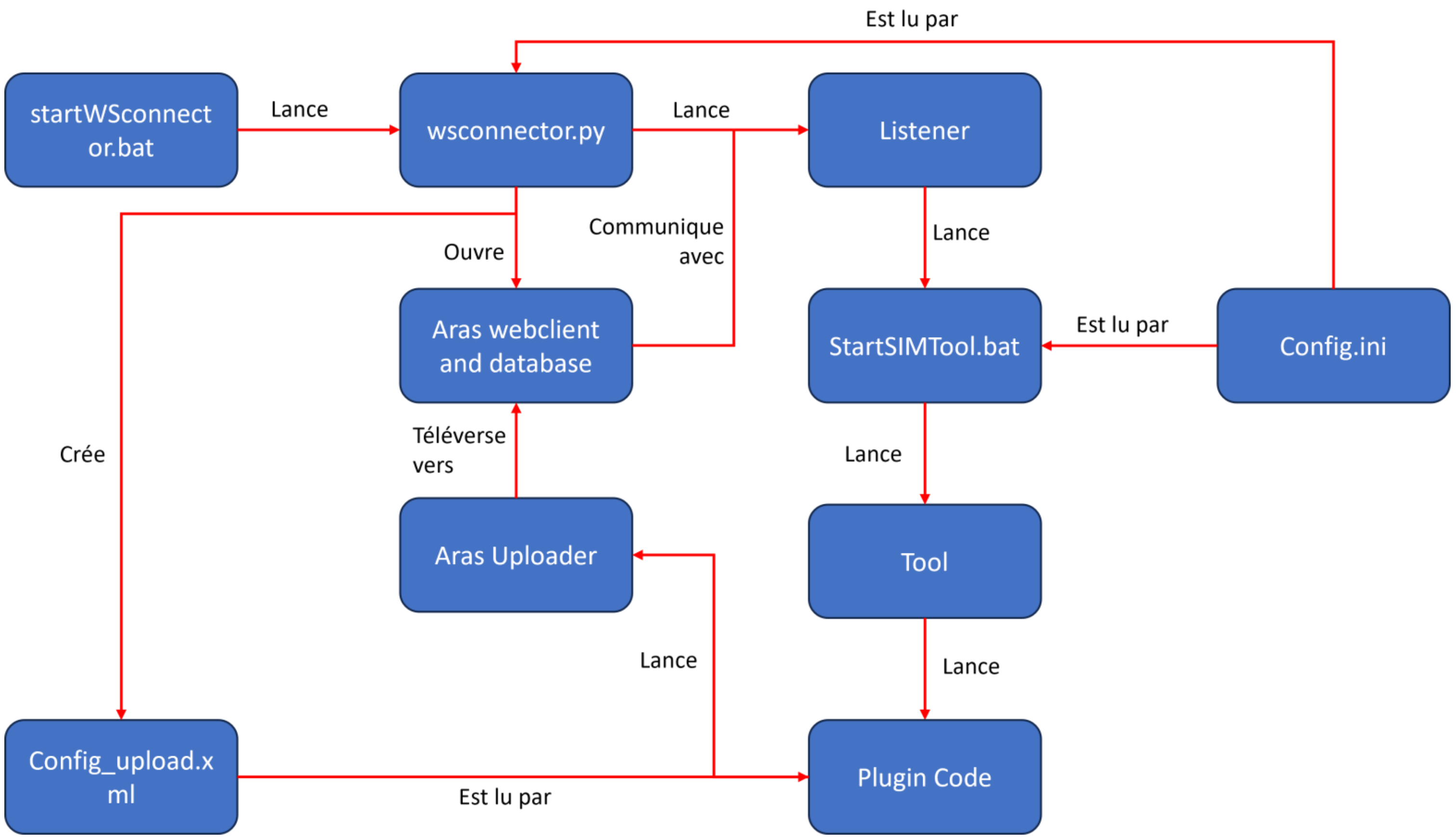
The connectors developed during this project provide a minimum viable product for the three simulation software packages Abaqus CAE, LS-PrePost and Coreform Cubit. As seen in Part III, it is therefore possible to carry out classic numerical simulation studies, such as a mesh convergence study, and to draw all the conclusions from these studies.

It should be noted, however, that there are numerous possibilities for further development of the connectors, such as support for new simulation software (Hyperworks, NASTRAN, etc.), new extensions for the Cubit connector, the addition of graphical interfaces when launching functions to give the user greater freedom over parameters, etc.

In addition, there are issues to be considered when connecting via the Internet, first and foremost for the security of confidential data, but the possibility of an entirely local mode in the event of inaccessibility to a network could also be envisaged. Uploading would then be automated the next time you connected to a secure network.

V/ Appendices

A) Abaqus CAE connector operating diagram



Appendix A: Schematic diagram of the Abaqus CAE connector for OpenSPDM

B) Johnson Cook model for metal plasticity

$$\sigma = (A + B\varepsilon^{n}) + (1 + C \cdot ln\, \dot{\varepsilon})(1 - (T^{*})^{m})$$

Equation 1: Johnson Cook model, [4]

With :

- σ stress in MPa
- A and B material constants in MPa
- C, n and m unitless material constants
- ε deformation
- T^* specific temperature in K

VI/ Bibliography

[1]: Damien Laval, Naval Group 2021, Naval Group Use Cases presentation

[2] : Edward Mark Norris, A standardized approach to building CAE application connectors to SPDM solutions, NAFEMS World Congress 2021

[3]: http://www.atomer.fr/1/1_dimensions-eprouvettes-essais-de-traction.html, last visited on 15/11/2023

[4]: Hassanifard, S. & Mousavisogolitappeh, Hojjat & Varvani-Farahani, A.. (2018). The influence of low-plasticity burnishing process on the fatigue life of friction-stir-processed Al 7075-T6 samples. Fatigue & Fracture of Engineering Materials & Structures. 42. 10.1111/ffe.12950.

[5]: https://matmatch.com/learn/material/7075-t6-aluminium, last visited on 15/11/2023

[6]: https://web.mit.edu/calculix_v2.7/CalculiX/ccx_2.7/doc/ccx/node27.html, last visited on 15/11/2023