DOI: 10.13009/EUCASS2023-302
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

Convergence of Machine Learning and Finite Element methods, the smart superelements 10th EUCASS – 9th CEAS

Jean-Daniel Lecuyer, Vicky Tsianika*, Tony Favaloro* and Bruce De Witt†*
** Hexagon, Route de l'Orme Bloc B, 91190 Saint-Aubin*
† Portalliance, 116 route d'Espagne, 31100 Toulouse
Jean-daniel.lecuyer@hexagon.com
Vicky.tsianika@hexagon.com
Anthony.favaloro@hexagon.com

Abstract

In finite element analysis for structural simulation, demand for computer resources has always exceeded existing capabilities. Once engineers discovered this method, the size of engineering problems quickly grew to exceed the capacity of the existing systems. This process has repeated itself time and time again. A solution to this problem can be achieved, for many years, by using the long-time known superelements [1]. The principle used here is that the model is divided into a series of components, each of which is processed independently resulting in a set of matrices that are reduced to a boundary and describe the behavior of the component as seen by the rest of the structure. This Reduced Order Modeling approach, with a fully description by a set of stiffness, damping, mass, loads matrices, has proven for a long time to be efficient and robust [2]. Reduced simulation models decrease the computational complexity of high-fidelity engineering models, allowing for data compression and encapsulation of large discretized partial differential equation (PDE) models.

In the other hand, those last years, all the Artificial Intelligence and Machine Learning technologies and capabilities have shown dramatic growing interest and great improvements to become mature [3].

The field of computational mechanics is undergoing massive transformation due to the arrival of those various breakthrough Machine Learning (ML) and Reduced Order Modeling (ROM) technologies [4]. With the new Smart Superelement technology (SSE), MSC Nastran extends the aforementioned superelement methodology by combining it with Machine Learning to provide parameterized superelements for linear static, linear dynamic, and nonlinear analyses. The parametrized approach enables the analyst to efficiently conduct what-if studies by capturing modeling variations (i.e., material parameters, geometrical parameters, etc.) even during the initial design stages when requirements or material availability may not be finalized. Smart Superelements can be used in assembly structural models with parameters available for optimization and robustness analysis. In this paper, we shall explore the aforementioned new SSE concept after some introductions on the superelements and the machine learning technologies. We will then expose a case study in the aerospace industry using a realistic industrial aircraft finite element model.

These new techniques combined with the unprecedented availability of computing resources (e.g., cloud), provide a unique structural simulation solution, to tackle previously unattainable engineering problems.

DOI: 10.13009/EUCASS2023-302

1. Introduction

The artificial intelligence (AI) is a broad concept that embrace data analysis, data science. Thanks to continuous improvements in the computational resources (hardware), the maturity of AI, and Machine Learning (ML) methods especially, achieved a certain level of maturity that allows to increase dramatically its use cases range.

In the other hand, the Finite Element method (FEM) has been largely used for decades, allowing accurate and efficient structural simulations in all industries, including aerospace. It is worth to mention the finite element analysis program NASTRAN was originally developed for NASA in the late of 1960s under U.S. government for the aerospace industry. Part of this simulation solution, the superelement feature has allowed engineers to leverage a Reduce-Order Modeling technique to avoid reaching some hardware limitations and gain in efficiency for their engineering processes. However, superelements have some limitations that will be exposed later, and we recognized that combining FEM with ML methods could reduce significantly them. The aim of this paper is to present an innovative framework, called smart superelements, that combines so FEM and ML and should provide new simulation values for many engineering processes, for example in the aerospace industry.

1.1 The finite element method and the superelements

In analysis based on the finite element method, it is quite usual to have the demand for computer resources that exceeds the existing capabilities. In the early days of computers, when engineers were solving 3 x 3 problems by hand, computers were able to handle problems as large as 11 x 11. Once engineers discovered this FEM ability, the size of engineering problems quickly grew to exceed the capacity of the existing systems. This process has repeated itself time and time again. Today, modern computers are capable of solving problems involving more than 100,000,000 equations with 100,000,000 unknowns, which is still not enough to satisfy the needs of many engineers as more detail is added to finite element models and higher fidelity solutions are required.

The limits on hardware resources, combined with budget restrictions (large runs and stochastic variations can be time-consuming), limits the ability of engineers to solve large, complicated problems with high fidelity meshes. A solution to these problems (both hardware and time budget), can be achieved for many models by using the superelements method.

By using superelements, the analyst can not only analyze larger models (including those which exceed the capacity of your hardware), but he can also become more efficient in performing the analysis, thus allowing more analytical design cycles or iterations in the analysis. Another benefit of superelements efficiency can be realized when models are subjected to probabilistic or stochastic analysis by varying portions of the structure.

Even in the design optimization area, the use of superelements has become automated to help reduce the overall optimization costs.

1.1.1 The superelement principle

The principle used in superelement analysis is often referred to as substructuring. That is, the model is divided into a series of components, each of which is processed independently resulting in a set of matrices that are reduced to a boundary and describe the behavior of the component as seen by the rest of the structure.

Often these components are comprised of logical groupings of elements (an engine, a wing, a fender, the exhaust system, etc.), hence the term superelement.

The reduced boundary matrices for the individual superelements are combined to form assembly matrices which are referred to as the residual matrices. The residual matrices are solved using standard techniques for calculating displacements (and velocities, accelerations for dynamic solutions). The residual solution is then imposed on the boundary of each superelement so that the data recovery (calculation of displacements, stresses, etc.) for the boundary can be combined with the data recovery for the body loads on the superelement.

In static analysis, the theory used in superelement processing is exact. In dynamics, the reduction of the stiffness is exact, but approximations occur during the reduction of the mass and damping matrices. The dynamic solution can be improved dramatically by augmenting the static reduction with additional dynamic degrees of freedom in a method called component modal synthesis, which has been used now for quite a long time.

Please note the superelement principle can be used in buckling analysis, the reduction of the differential stiffness uses the same theory as the static reduction but doesn’t provide an accurate solution compared to the non-reduced system.

Finally, the superelements method can be used in nonlinear analysis, but the superelement is limited to a linear reduction in its initial orientation as we will see with more details later in this paper.

DOI: 10.13009/EUCASS2023-302

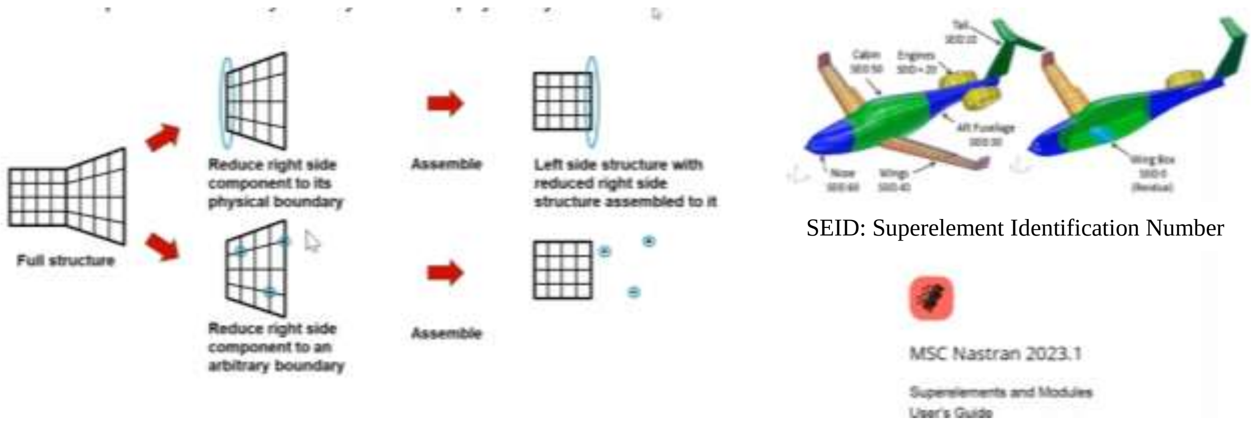


Figure 1: The superelement principle and a use case example for an aircraft model

1.1.2 Fundamentals of Superelement Analysis

As seen before, superelement analysis can be described as a form of substructuring, allowing each superelement to be processed independently of all other superelements. The processing of each superelement results in a reduced set of matrices (mass, damping, stiffness, and loading) that represent the properties of the superelement as seen at its connections to adjacent structures. Once all superelements have been processed, these reduced matrices are assembled in what is known as the residual structure, and the assembly solution is performed. Data recovery for each superelement is performed by expanding the solution at the attachment points, using the same transformation that was used to perform the original reduction on the superelement.

Superelements can consist of physical data (elements and grid points) or can be defined as an image of another superelement or as an external superelement (a set of matrices from an external source to be attached to the model). Figure 2 demonstrates a simplified illustration of condensing a structure to its boundaries, solving a reduced system, and back-expanding the solution to obtain the data recovery for the superelement.

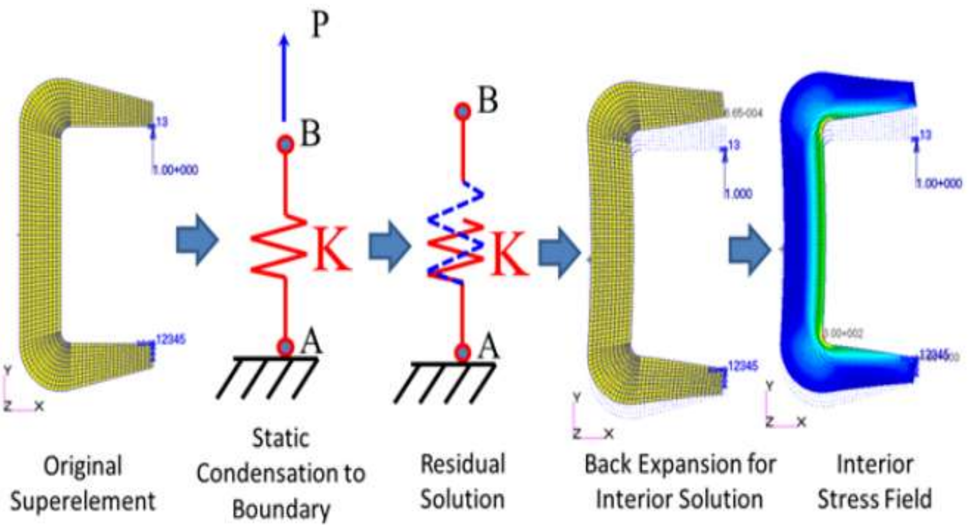


Figure 2: Simplified Depiction of Superelement Reduction, Solution, and Data Recovery

The following figure illustrates the possible superelement in an aerospace engineering context. In Figure 3, a model of an aircraft is shown. It will be used all along through this paper. Different sub-assemblies can be modeled as superelements: the tail, the wings, etc. Concerning the wings, we can expect a planar symmetry, so that one wing sub-assembly can be used for the superelement creation. This is called a primary superelement (the actual geometry for the superelement is defined in FEM input data). The other wing can be an image (with a mirror effect) of the first (primary) wing. An image superelement is a superelement that uses the geometry of another superelement to describe it. These image superelements can save processing time in that they are able to re-use the reduced stiffness, mass, and damping matrices from their primary superelement, which reduces the number of calculations needed. Full data recovery is

DOI: 10.13009/EUCASS2023-302

available for image superelements. An image superelement can be an identical image or a mirror image, like in this case. In Figure 3, the superelement 2 can be a mirror image copy of the primary superelement 1.

Please note that superelement images can have their own unique loadings. Only the stiffness, mass and damping are identical to the primary. Another type of superelement is an external superelement, where a part of the model is represented by using matrices from an outside source (from another FEM analysis run). For these matrices no internal geometry information is available; only the grid points to which the matrices are attached are known. This superelement type will not be used in the smart superelement context of this paper.

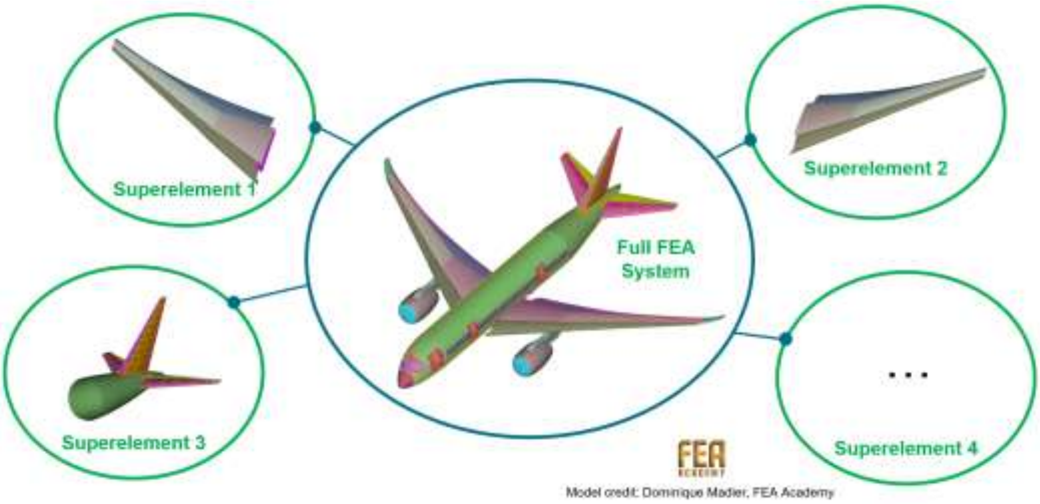


Figure 3: An aircraft FEM model and superelements examples

The following figure describes the different steps when leveraging superelements, with three key steps: a first phase dedicated to the reduction, a second phase for residual processing, solution and a final phase for data recovery.

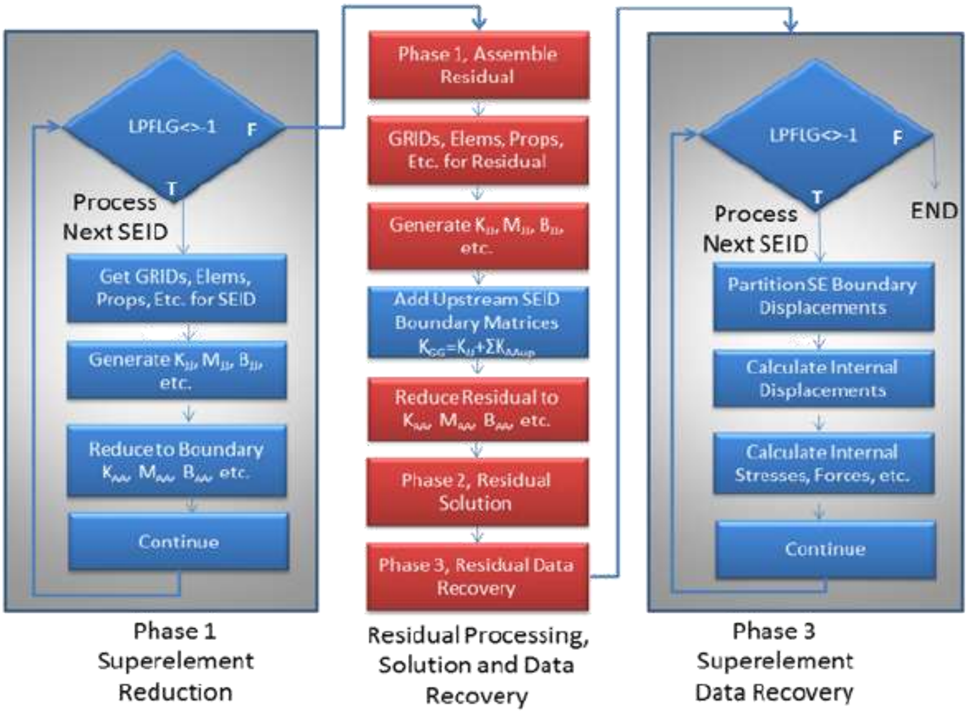


Figure 4: Flowchart for Superelement Processing

DOI: 10.13009/EUCASS2023-302

1.1.3 Outputs of superelement processing

As shown previously, superelements are, in a nutshell, matrix representations of structural characteristics with different generated outputs:

- KAA – the stiffness matrix reduced to the boundary. It includes the static reduction and eventually the modal degrees of freedom, DOF, if they were requested
- MAA – the mass matrix reduced to the boundary. It includes the statically reduced mass and modal DOF, if they were requested
- MUG1 – the displacement output transformation matrix (OTM). It contains the transformation from the boundary solution points of the residual structure to the boundary points of the superelement
- MUG10 – the OTM transformation for any interior points that were added during the condensation
- PA – the reduced load vectors if any load were requested to be reduced when the superelement was created

All those matrices are basically the output data to be used by the machine learning algorithms in the context of the smart superelements as we will see in further paragraphs of this paper. Please note other matrices exist for specific superelement cases, such as acoustic coupling, structural damping, fluid structure partitioning, but will not be considered in this paper and this smart superelements context.

1.1.4 Advantages and limitations of the superelements

Efficiency is the primary reason to use superelements. A finite element model is rarely analyzed only once. Often the model is modified and re-analyzed time and time again. By only analyzing the part of the structure which changes, the user can save significant time. Without using superelements, each analysis can cost the price of a complete solution.

Because superelements can be processed individually with less computer resources required than a complete, non-superelement solution, it is often possible to submit individual superelement processing runs using fast queues (or on local workstations instead of servers), rather than waiting and running the complete problem at once using an overnight queue.

However, the usage of superelements leads to some limitations. The reduction is a linear operation, meaning the mechanical behavior of the superelement itself can only be linear (small displacement, linear material behavior, no contact). Therefore, KAA, MAA and other matrixes, exposed in the previous subchapter, remain constant. Besides, any change in the model properties, loads or material properties inside a superelement trigger the necessity of recompute it, so that we can consider it as “fixed”. Finally, superelements can be used for optimization processes (SOL200 in MSC Nastran) but recompute of the superelements is required for any not negligible change inside them.

1.2 Machine Learning dedicated to the engineering

1.2.1 Basics of Machine learning

Machine Learning, part of the Artificial Intelligence domain, embraces a large bunch of various approaches: supervised, unsupervised, reinforcement learning. For the purpose of the Smart Superelement method, which can be considered as belonging to the Physics-Informed Machine Learning framework, the supervised machine learning paradigm will be leveraged, meaning the data used to build the mathematical, predictable, models are predefined, such as the common cat versus dog example shown in Figure 5.

Part of the data (inputs, outputs) will be used to train the ML models by performing predictions. Other data will be used to validate the accuracy of the generated models. As long as the predictions remain inaccurate, the training process will occur. In case of the smart superelements, this process will be used to predict the structural behavior of a complex assembly trough a reduced order modelling (ROM) approach. Please note, by essence, the traditional superelement method is already a ROM technique somehow.

DOI: 10.13009/EUCASS2023-302

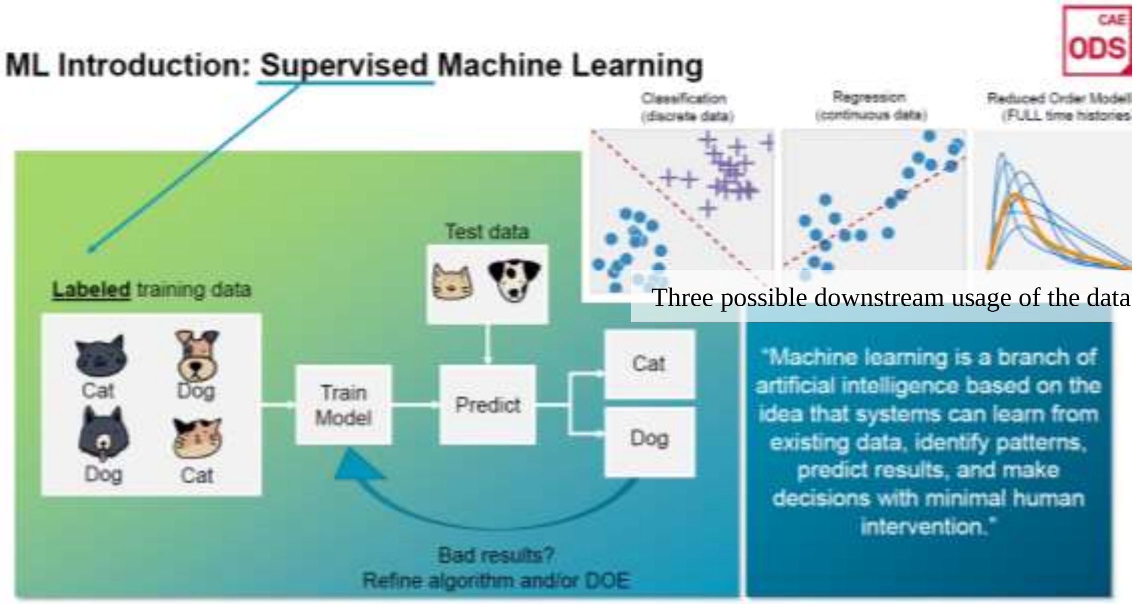


Figure 5: A basic Supervised Machine Learning example to predict if an image shows a dog or a cat [7]

1.2.2. The Reduced Order Modeling approach

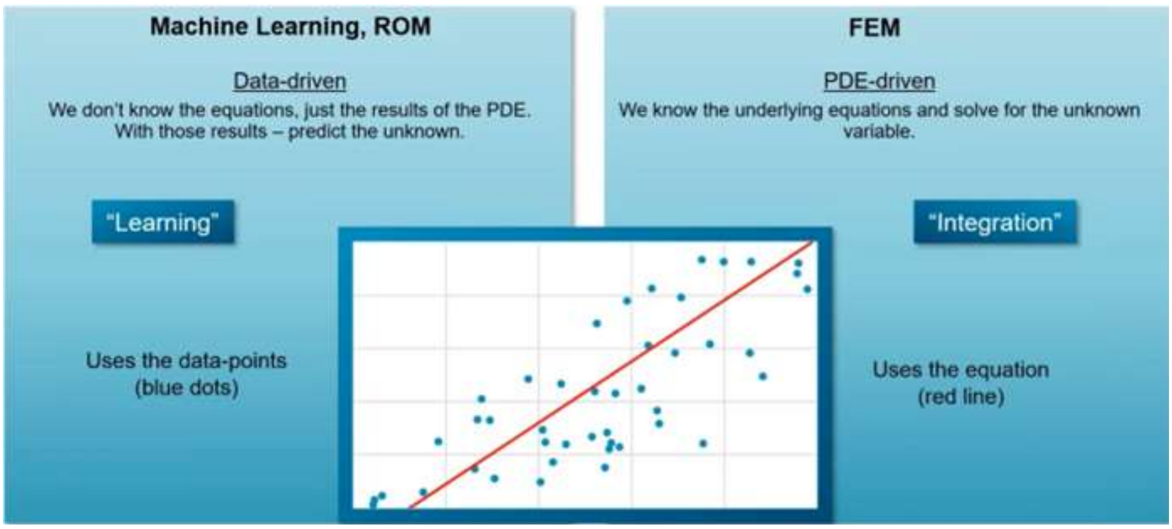
Three steps are usually seen when a reduced-order model is generated:

- Decomposition where the system is split up in multiple simpler systems
- Reduction where the volume of data is compressed, reduced, to keep only the relevant data
- Reconstruction where the system is built back again after solving the simple systems and predictions are possible

The decomposition phase can perform using a large set of techniques such as Proper Orthogonal Decomposition (POD, algebraic approximation), neural network (deep, fully connected or multilayer perceptron), Central Voronoi Tessellation (CVT) or interpolation (kriging, radial basis functions, etc.) methods, as available in Odyssee CAE.

Please note that, unlike response surface methods where only smoothed solutions on certain criteria are obtained, POD technique, but also CVT or neural network, provides adequate solution when strong non-linearities occur.

1.2.3 Machine Learning versus Finite Element Method summary



*PDE: Partial Differential Equation

Figure 6: Summary of ML versus FEM comparison

DOI: 10.13009/EUCASS2023-302

Even if those two approaches are different, synergies are possible, by connecting the machine learning techniques with data coming from Finite Element Method based simulations. The data used for the model reduction through machine learning are related to finite element models (FEM) data, where input parameters exist. The consequence will be a reduced order model that is able to predict the mechanical behavior of a system that could be provided by a FEM solver, but with a dramatically reduced time.

Synergies make not only sense with the FEM method, but all the simulation types, such as Computational Fluid Dynamics (CFD).

2. The Smart Superelements

After recognizing the benefits and limitations of the existing superelements method in one side and of the machine learning techniques in the other side, it became clear that more values can be provided by connecting those two. This led to the development of a new solution under the Physics-Informed Machine Learning (PIML) framework. We are planning soon to leverage Mechanics-Informed Neural Network (MINN) techniques, when Neural Networks will be used for the machine learning and mechanical solvers would be used for the mechanical simulations (please see the last chapter, Conclusion and Perspectives, of this paper.

Figure 7: Example of possible usage of smart superelements in an automotive industry context

2.1 The Smart Superelement principle

The aim of the innovative smart superelement method is to predict the matrices of superelements (KAA, MAA, etc. as exposed in a previous paragraph) through machine learning, so without performing the usual complete, time-consuming superelement processing for each superelement (condensation process).

It is quite usual to have many evolutions, versions of superelements. The current engineering process requires a complete new superelement processing all the time any change occurs inside the superelement. With this new approach, several superelement processing tasks have been already executed through a Design of Experiment (DOE), and the data generated by this DOE are used by machine learning algorithms to predict the matrices of a new superelement where one or several modifications occurred, without so the need of an extra superelement processing, (condensation process).

Usual reasons of superelements evolutions or versions are changes in mesh, material or model properties, loads, etc.

Figure 8: The different steps of the smart superelements process

DOI: 10.13009/EUCASS2023-302

As shown in Figure 8, the smart superelement process includes 4 steps:

- First, one MSC Nastran input deck creation to generate a first version of a superelement (with specific values for the parameters that could change) is required
- Then, a Design of Experiment is run by using the previous input deck in Odyssee CAE, but where simulation parameters will change. The execution of this DOE provides data usable for the next steps: various MSC Nastran input decks, according to the parameter changes, and the associated superelement matrices (KAA, MAA, etc.). This DOE step is completed here with the Odyssee CAE software
- Then, the Machine Learning process can occur leveraging the data mentioned earlier. After a training run, a Reduced-Order Model (ROM) is obtained inside Odyssee CAE and can be exported leveraging the Functional Mockup Unit (FMU) leveraging the Functional Mockup Interface (FMI) open standard
- Finally, this ROM of the superelement (and eventually several ROMs) can be integrated in a full Finite Element model to assess the structural performance of the entire system

2.2 Smart superelements are parameterized Superelements

The figure below shows a concrete example on how various smart superelement (SSE) versions, depending on the input parameters that have been selected, can lead to various versions of the entire aircraft system to be used for structural simulations.

Version 1 of the full system uses version 1 of SSE 1 (right wing sub-assembly), version 3 of SSE 2 (left wing sub-assembly – could be also a copy of SSE 1.1 with a mirror effect) and version 1 of SSE 3 (tail sub-assembly).

Version 2 of the full system uses version 3 of SSE 1 (right wing sub-assembly), version 3 of SSE 2 (left wing sub-assembly – could be also a copy of SSE 1.3 with a mirror effect) and version 1 of SSE 3 (tail sub-assembly).

Version 3 of the full system uses version 2 of SSE 1 (right wing sub-assembly), version 2 of SSE 2 (left wing sub-assembly – could be also a copy of SSE 1.2) with a mirror effect) and version 3 of SSE 3 (tail sub-assembly).

Any other combination could also be considered obviously.

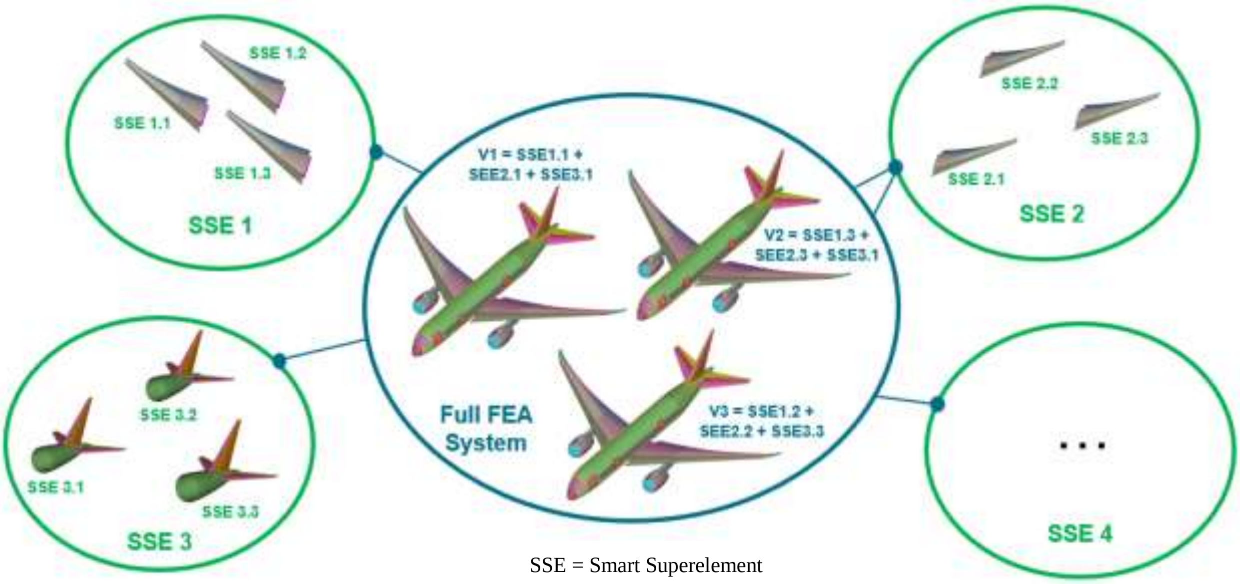


Figure 9: Possible full aircraft system versions depending on the selected Smart Superelements versions

All the versions are related to FEM parameters modifications. Such parameters could be related to:

- Material properties: Young modulus, Poisson’s ration, Mass density, etc.
- Element properties: thickness, non-structural mass, area of beam cross section, beam cross-section dimensions, etc.
- Connection properties: stiffness, damping, mass etc.
- Loads definition: magnitudes for instance
- Etc.

DOI: 10.13009/EUCASS2023-302

3. Smart superelements use cases

As for regular superelements, smart superelements can be used for a static analysis context (static condensation process) or a linear dynamic analysis context (additional dynamic reduction).

3.1. Linear Static analysis context

The figure 10 shows the different steps of the process in case of the usage of a smart superelement in a linear static analysis context.

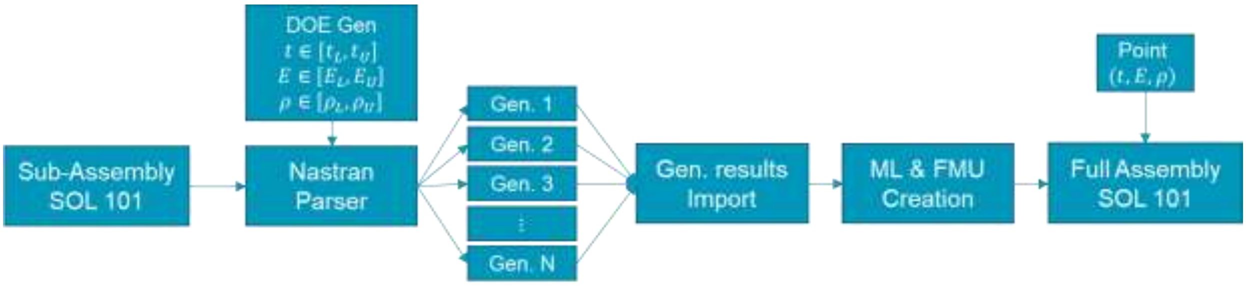


Figure 10: Smart superelement flowchart in case of a linear static analysis

It all starts with a FE model of a sub-assembly (or several), a template model. SOL 101 generates a superelement with a static condensation of this sub-assembly.

Then, a MSC Nastran parser in Odyssee CAE is used to identify the locations in the input deck where the parameters we want to modify are defined. In this flowchart example, three parameters are going to be changed: t for thickness of a shell property, with a value between the values t_L and t_U, E for the Young Modulus of one material, with a value between the values E_L and E_U and ρ for the Poisson's ratio of the same material, with a value between the values ρ_L and ρ_U. Through the parser, we are so able to automatically changes the Nastran input deck according to the values identified by the Design of Experiment algorithm, such as Optimal Latin Hypercub, for those three parameters.

Several Nastran input decks (Gen. 1, Gen.2, etc.) are generated for all the combinations of those three parameters, and batch simulation jobs are executed to get the superelement matrices for each combination.

Finally, all the data are imported inside Odyssee CAE software to perform the machine learning, to generate the Reduce Order Model exported as a Functional Mockup Unit. This reduce model is the so called smart superelement.

Finally, this superelement can be integrate in the full assembly, the global FE model. We are able at that time to specify any value for the three parameters (inside the range) and run the analysis to get the structural performance of the overall system with the specific configuration.

3.2. Linear Dynamic analysis context

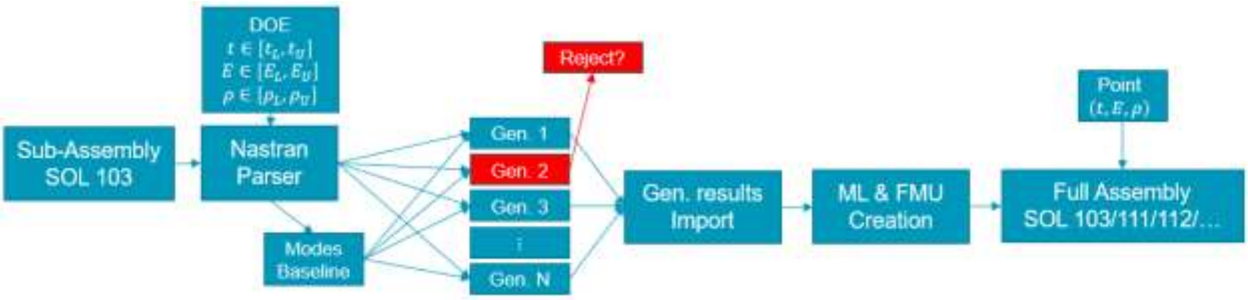


Figure 11: Smart superelement flowchart in case of a linear dynamic analysis

As already exposed in the first Introduction chapter, dynamic analyses with superelements, and so smart superelements, require specific attention, with a dynamic reduction on top of the static one. Eigenvalues and associated mode shapes are required and lead to some approximations. Accuracy will depend on the number of the eigenvalues.

DOI: 10.13009/EUCASS2023-302

With the parameter changes, the eigenvalues and mode shapes sequence may change from one batch run (generated superelement during the DOE in Odyssee CAE) to another. The order of the mode shapes according to the eigenvalues may change. To track such issue, a *mapmodes* feature is available in the FE solver so that the baseline model and all the superelements generated can be safely compared and used. In case of a mode shape of one generated superelement shows significant difference with any mode of the baseline, different options exist, and, for instance, this mode is automatically rejected to ensure a full consistency for the ROM (please see paragraph 4.2. for more information).

For this dynamic use case, the rest of the process is similar to the previous one (static). The final simulation with the full assembly including smart superelements can leverage any type of dynamic analysis (SOL 102/111/112/… in MSC Nastran).

4. Aerospace example

We will now look at an aircraft wing model as a demonstrator for the usage of smart superelements. A linear dynamic analysis of the full aircraft will be performed, with the wing modeled as a smart superelement.

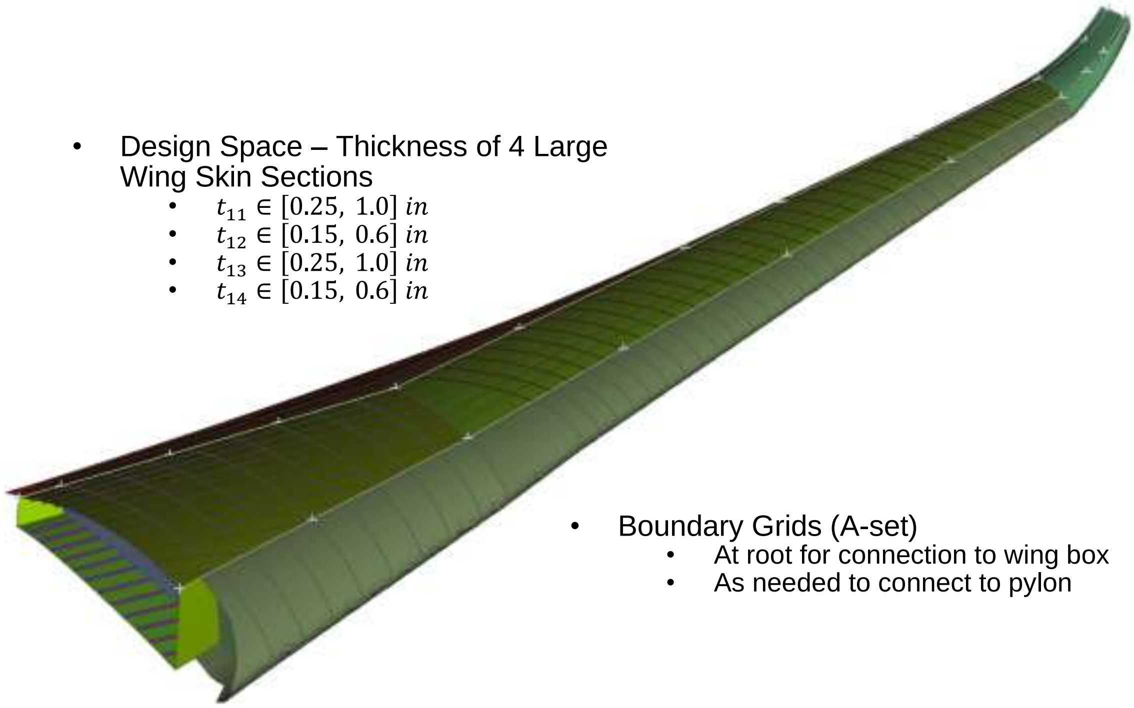


Figure 12: Example problem – an aircraft wing

The identified modifiable parameters for this wing sub-assembly are the thicknesses associated with shell properties. The identified boundary grids for the condensation are at the wing root, for the connection to the wing box and at the location closed to the pylon that connects to the engine. The design space, the ranges for those 4 thickness values, are defined according to Figure 12. Beam elements along the length of the wing (in white color in Figure 12) have been selected to allow a reduced visualization (data recovery at all the elements of the smart superstructure is not required), and to improve the efficiency of the overall process.

```
base > DM7.21_LHWing_base.bdf
assign output2='baseline_modes.op2',unit=50, delete
SOL 103
CEND
mapmodes(baseline)=50
METHOD = 100
RESVEC = YES
EXTSEOUT(ASMBULK=MANQ,EXTBULK,EXTID=1021,HDF5)
LOAD=30
```

DOI: 10.13009/EUCASS2023-302

Figure 13: Example problem – an aircraft wing

Figure 13 shows the beginning of the MSC Nastran input deck. As a dynamic reduction is expected, a SOL 103 for a normal modes analysis, has been defined. The *mapmodes* option, as discussed earlier can be seen. This input deck will be used as a template so for the ones generated during the Design of Experiment process. The parser tool will allow adequate modifications of the PSHELL thickness values for the batch runs, according to the points defined by the DOE algorithm (please see Figure 15 and thk11, thk12, thk13 and thk14 items).

4.1 Design of Experiment

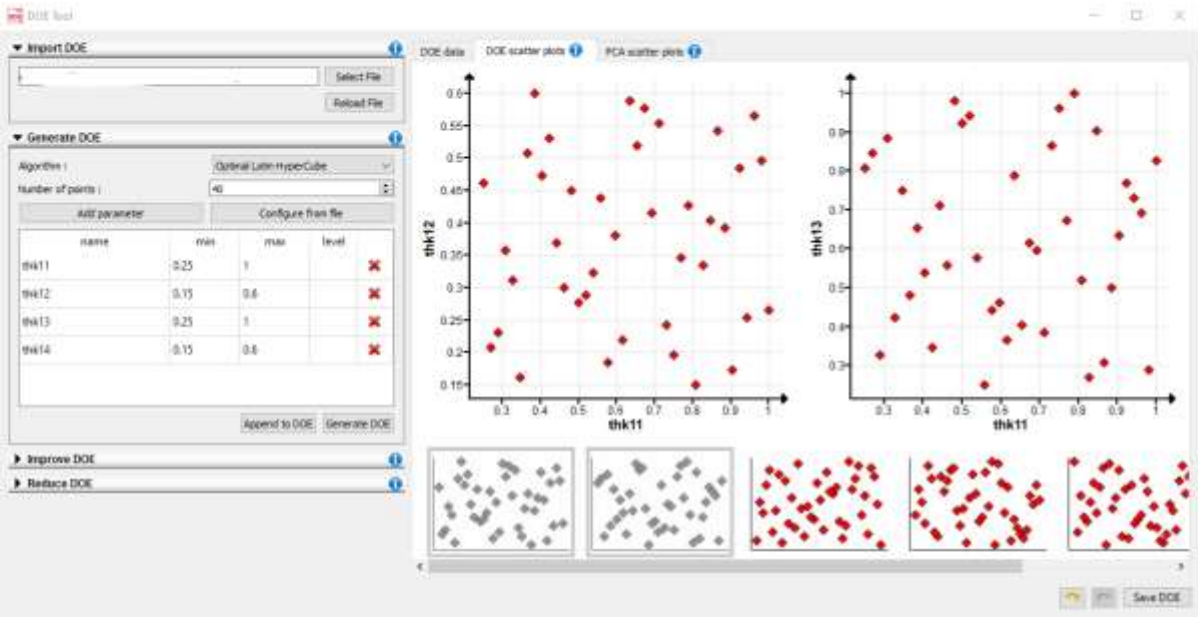


Figure 14: Design of Experiment – various points generated by the Optimal Latin Hypercub algorithm

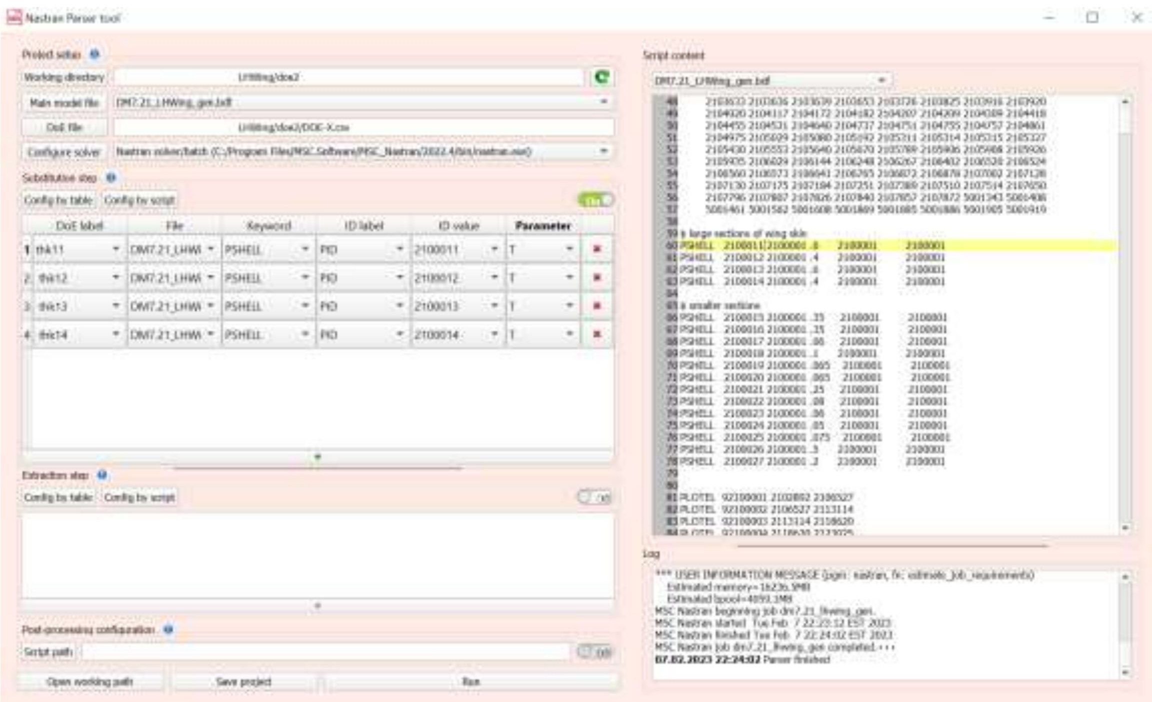


Figure 15: Design of Experiment – MSC Nastran parser to generate relevant input decks for the DOE

For this particular model, in Odyssee CAE, the Optimal Latin Hypercube algorithm has been used for the DOE points, 4 PSHELL Nastran keywords have been parsed to reused thickness values associated with those DOE points.

4.2 Machine Learning to generate the ROM

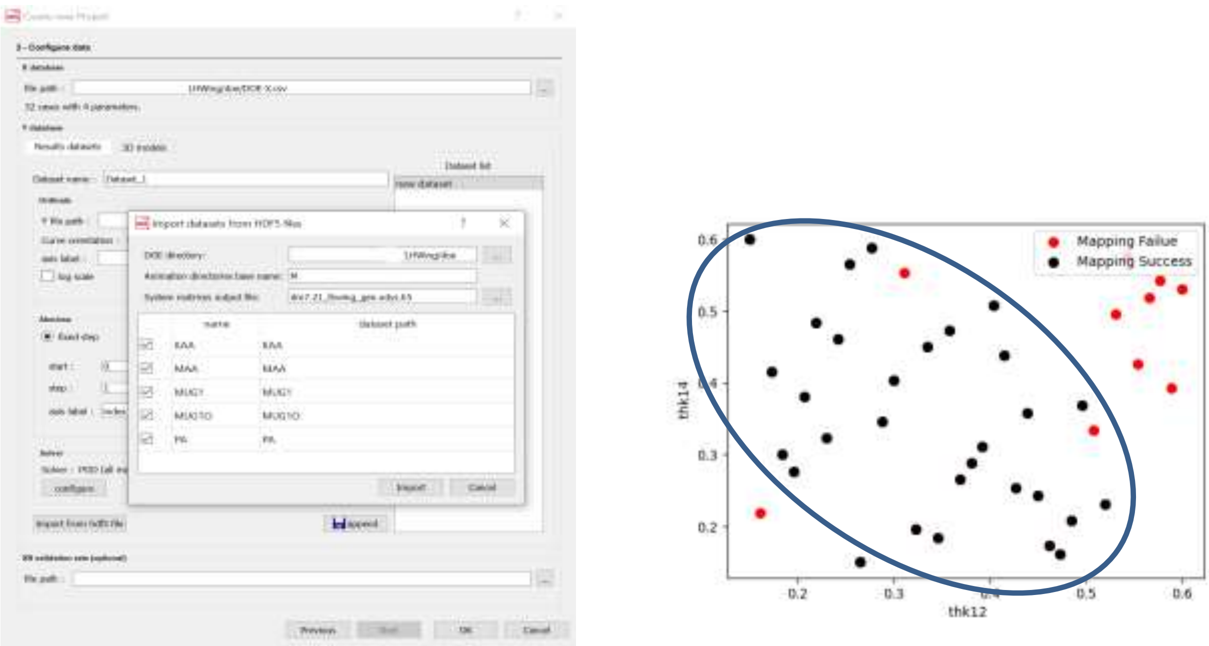


Figure 16: Structural matrices available as DOE output data, one DOE points map for two parameterized thicknesses

As explained earlier, large mode shifts between the generated superelements during the DOE process and the baseline superelement can lead to mapping issues, because of the inconsistency for an interpolation perspective. This is the case in this example (please see red points in the graph in Figure 16), because of the large design space that has been defined. The structure, the thicknesses have changed so much that large mode shifts occur. In such case, three options are available. First, we can accept the reduced design space (inside the blue ellipse in Figure 16), no new run is performed. Second option is to rerun the failed points, but with a lower threshold, leading to some extra work and a few repeated runs (specific *mapmodes* related information is available in results files to provide enough information for engineering judgment). The last option is the move the baseline point. In such case, all runs must be repeated as the baseline is modified, but this allows a better position of the baseline point according to the design space.

For all the generated superelements during the DOE, all expected matrices (KAA, MAA, etc.) are available inside HDF5 files as we can see in Figure 16. Those data will be used as output data for the machine learning process. The parameter values for the DOE points will be used as the input data for the same Machine Learning process.

After the machine learning has been completed (here with a POD algorithm), a ROM, the smart superelement, is obtained, and can be exported in the FMI format as indicated earlier. This means we have now the possibility to estimate almost instantly all the relevant structural matrices of the superelement according to the thickness values, the parameters, the user can select. This FMU can be included, consumed, in the full assembly MSC Nastran input deck. And the DESVAR and DVPREL1 MSC Nastran keywords are used to define the exact thickness values to be used in this full assembly simulation.

5. Results and discussion

5.1. Validation of the smart superelement in a corner of the Design Space [1.0, 0.6, 1.0, 0.6]

For a first validation of the smart superelement method in a dynamic simulation context, it has been decided to test the results with the ROM -smart superelement- approach with the results obtained with a classical FEM. Please note the selected point in the design space doesn't match with any point of the DOE to avoid any trivial result. Values for thicknesses selected here are 1 for t_{11}, 0.6 for t_{12}, 1 for t_{13} and 0.6 for t_{14}. The initial model, used for the template input deck, is also the baseline model used for eigenfrequencies results comparison as shown in the table below. This model used a "center" point regarding the design space as the associated point is defined with 0.5 for t_{11}, 0.3 for t_{12}, 0.5 for t_{13} and 0.3 for t_{14}, values defined for the various thicknesses.

DOI: 10.13009/EUCASS2023-302

In the table below, first column are results with the baseline model, second column are results with thickness values at the validation point with a full FE model, third column are results with thickness values at the validation point but with the full model containing the smart superelement. Next two columns show the value changes in percentages in comparison with the baseline model, and the last Error column if about results difference between the full FEM and the model with SSE. As we can see, results with the model containing the smart superelement are very closed top the reference, the results with a model with only finite elements, indicating good accuracy.

Table 1: Eigenvalues results comparison between a model with smart superelement and a full FE model

Eigenfrequencies (Hz)					
Baseline	**Corner Point**		**Change vs Baseline**		
Full Model	**Full Model**	**SSE Model**	**Full Model**	**SSE Model**	**Error**
0.7672	**0.7325**	**0.7325**	**-4.52%**	**-4.52%**	**0.00%**
1.6329	**1.6715**	**1.6714**	**2.36%**	**2.35%**	**0.01%**
1.6339	**1.6909**	**1.6917**	**3.49%**	**3.54%**	**0.05%**
2.2466	**2.3660**	**2.3661**	**5.31%**	**5.32%**	**0.01%**
2.4065	**2.4911**	**2.4911**	**3.52%**	**3.52%**	**0.00%**
2.4385	**2.5987**	**2.6021**	**6.57%**	**6.71%**	**0.13%**
2.4928	**2.6799**	**2.6842**	**7.51%**	**7.68%**	**0.16%**
2.9016	**2.9408**	**2.9411**	**1.35%**	**1.36%**	**0.01%**
3.1948	**3.2038**	**3.2037**	**0.28%**	**0.28%**	**0.00%**
3.3619	**3.5051**	**3.5053**	**4.26%**	**4.27%**	**0.01%**
3.5180	**3.5137**	**3.5132**	**-0.12%**	**-0.13%**	**0.01%**
3.5467	**3.5378**	**3.5368**	**-0.25%**	**-0.28%**	**0.03%**
3.6536	**3.8166**	**3.8173**	**4.46%**	**4.48%**	**0.02%**
4.3812	**4.5029**	**4.5029**	**2.78%**	**2.78%**	**0.00%**
4.6145	**4.7402**	**4.7418**	**2.72%**	**2.76%**	**0.03%**
4.9729	**4.9729**	**4.9729**	**0.00%**	**0.00%**	**0.00%**

5.2. 3D results visualization

As indicated in previous chapters, data recovery is available for the superelements, and therefore for the smart superelements also. It means we have the ability to visualize 3D results (displacements, stresses, etc.) inside the reduced order model (whatsoever superelement or smart superelement).

In this particular example, the data recovery has been restricted to a subset of the wing model, to a few beam elements, as already indicated at the beginning of the chapter 4. One of the reasons could be intellectual property protection: for example, if the ROM needs to be shared with a third-party company and the details of the design should not be shared. However, some 3D visualizations are required for the post-processing even with this confidentiality issue. Another reason is to speed the analysis obviously, as the output fields are not predicted for the entire sub-assembly.

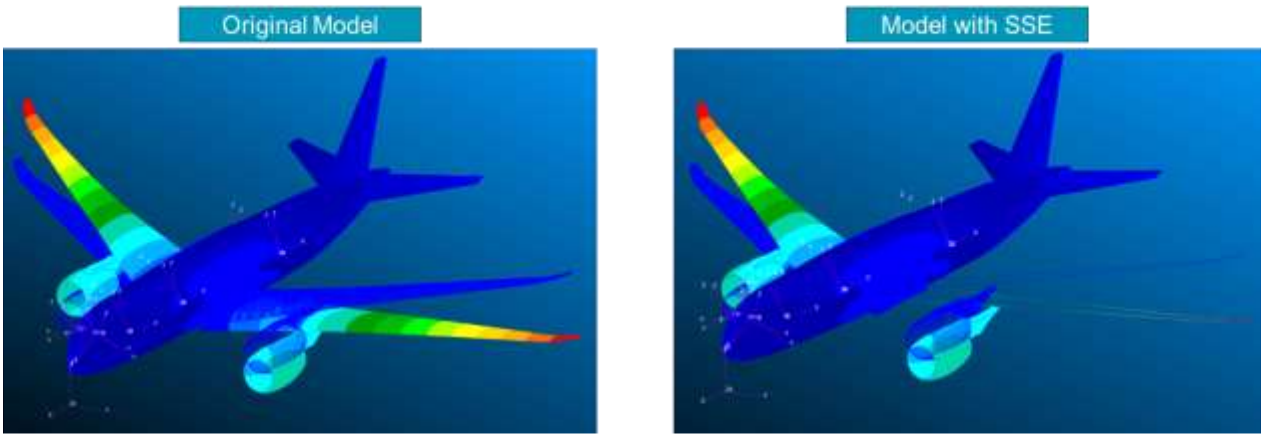


Figure 17: Displacement results (first mode) comparison between a full FE model and a model with a SSE

DOI: 10.13009/EUCASS2023-302

6. Conclusion and perspectives

The aim of the paper has been to emphasize a new method, the smart superelements in MSC Nastran which embeds Machine Learning (ML) technologies from Odyssee into Finite Element Analysis (FEA). This new ROM technique has been validated through various models related to different industries, including the aircraft model with a wing sub-assembly as a smart superelement exposed here. Validations show a great accuracy level of this new method when used properly. This innovation should allow engineers to accelerate the design process and should enhance the engineering collaboration, as it simplifies the way people can work on different sub-assemblies of a global FE model. Controlled data recovery capabilities are crucial in this collaboration perspective.

Regular superelements are fixed, non-parameterized. Those first smart superelements allow them now to be parameterized, providing a new great agility for the structural simulation users.

We are working right now on the ability to include mesh morphing in the parametrization of the smart superelement, by leveraging technologies from our partner Detroit Engineered Products (DEP). Mesh morphing can be defined through an input parameter for the smart superelement, leading the capability to the ROM to undergo significant geometry modifications.

Besides, we are also working on non-linear smart superelements. Regular superelements are limited to a linear reduction. Thanks to Machine Learning algorithms, we are convinced we should be able pretty soon to remove this limitation to add now non-linear structural behavior to those ROMs, thanks to this smart superelement framework. For such non-linear smart superelements, we are planning to leverage Neural Networks, leading to the possibility to enter the Mechanics-Informed Neural Network (MINN) domain.

To conclude, I would like to emphasize again the use of Functional Mockup Units for the consumption of the smart superelements in the full assembly simulation model, offering a complete openness to other applications, to third party software, thanks to the FMI open standard, and adding another agility enabler for the structural simulation users.

References

[1] MSC Nastran 2022.4 - Superelements and Modules User's Guide
[2] Wing Structure Static Analysis using Superelement, Procedia Engineering, September 2012
[3] Coupled Aerostructural Optimization using High-Fidelity Aerodynamics and Surrogate Modeling for the Structure, AIAA Aviation 2022 Forum
[4] Machine Learning and CAE complementarity for automotive project development, MR, Di Valentin, Stellantis, HXG 2021 with AI/ML Conference
[5] Embedding Machine Learning into Finite Element Analysis to Accelerate Design and Enhance Collaboration, AIAA webinar, February 2023
[6] MSC Nastran 2022.4 – Dynamic Analysis User’s Guide, chapter 20 Nastran Smart Superelement
[7] Odyssee CAE 2023.1 - Lunar User Guide