Abstract

In modern engineering, the use of virtual prototyping is continuously and significantly growing. However one of the recent challenges arising in computational analysis consists in taking accurately and efficiently into account strong coupling between different physical fields. Depending on the applications, multiple solvers or environments are involved in the design flow and, consequently, integrated solutions require effective interfacing and data exchange between different software components.

The OOFELIE® toolkit, developed by Open Engineering, is a multi-physics, multi-methods and multi-analyses commercial application dedicated to numerical simulation. Thanks to its open architecture, it is an ideal platform to develop integrated and innovative solutions.

In the present paper, relevant applications based on the OOFELIE framework are presented in the expanding domain of Fluid-Structure Interaction (FSI), Optics and Concurrent Design, with focus on the platform interfacing capabilities and on the communication schemes and resolution strategies developed within these projects.

1 Introduction

As technologies are in constant evolution, engineer’s needs in terms of numerical simulation are permanently growing, especially in multidisciplinary designs. However, historical codes have shown their limits and reveal to be unable to combine efficiently several physical fields simultaneously. This fact is particularly highlighted when optimisation or control is involved in the design flow of critical components of spatial equipments.

1.1 The OOFELIE framework

Open Engineering, the innovation unit of the SAMTECH Group, is developing OOFELIE® that constitutes a new generation of multi-physics CAE software designed to take into account, in the same analysis, multiple physical fields in a strongly coupled formulation and to combine multiple methods in a single computation scheme.

The OOFELIE name stands for Object Oriented Finite Elements Led by Interactive Executor. This numerical simulation toolkit was indeed initially based on the finite element method (FEM) but, thanks to the continuous development strategy of Open Engineering, it now also involves the boundary element method (BEM), the finite difference method (FDM) and finite volume method (FVM).

Written using the C++ language following Object Oriented Programming techniques (OOP), OOFELIE consists in a collection of classes dedicated to numerical simulation.

OOFELIE, the core of the products developed by Open Engineering, is active in most coupling of the physical fields, such as vibro-acoustics, electromagnetism, piezoelectricity, thermo-mechanics and fluid-structure interaction.

As illustrated on Figure 1, OOFELIE® is the key angle of the Open Engineering multiphysics integrated solution and it demonstrates extensive interfacing capabilities. The link between the solver and the CAD world is provided by SAMCEF Field which OOFELIE is interfaced with. There also exists a strong interaction between OOFELIE and external simulation software or environment such as FINE™/Hexa (Numeca International), ZEMAX®, MEMS Pro (SoftMEMS) and Excel® (Microsoft). In terms of post-processing, OOFELIE data and simulation results can be exported to several visualisations software (SAMCEF Field, ParaView, Matlab®, …) or processed directly in specific applications (ZEMAX®, CFView™, Excel®, …)

In addition to the integrated environment in which OOFELIE is plugged in, a command line interpreter gives access to the high and low level capabilities of the software in terms of numerical simulation that makes OOFELIE an effective toolkit for R&D on numerical methods and to evaluate innovative solution scheme. Furthermore, OOFELIE modules are available as shared libraries and, as OOFELIE has been recently provided with Python bindings, OOFELIE classes are also available in the Python interpreter.
Open Engineering Focus on Multi-Physic Offer

Figure 1: The OOFELIE framework interfacing capabilities

The relevant interfacing capabilities of OOFELIE make this framework particularly well suited to develop innovative solutions in domain where data exchanges between several applications is a key feature.

1.2 The SAMCEF Field environment

SAMCEF Field is the powerful graphical interface (GUI) that drives OOFELIE.

SAMCEF Field is an integrated CAD-based design and modelling environment that covers all the consistent steps of Computer Aided Engineering (CAE) and gives access to a broad spectrum of analysis.

In addition to the importation of well-known CAD formats and its powerful shape healing capabilities, a key feature of the SAMCEF Field modeller is its ability to handle parameterised geometries. In this last case, the user can generate the 3D geometry of its model from a 2D sketch, as illustrated on Figure 3, or directly create parameterised 3D entities.

Thanks to its contextual menu, its clear data tree and its pop-up windows that are physical concept oriented, this modelling and analysis environment reveals to be particularly user friendly and very efficient to guide the user in the different steps of the analysis, from the creation of the model (pre-processing) to the analysis of simulation results (post-processing).

Figure 2 summarises the different steps of a model analysis and illustrates the interactions between SAMCEF Field and OOFELIE.

In terms of post-processing, SAMCEF Field proposes the classical results visualisation features such as result display on mesh or geometry, plot drawing, ...

Figure 2: Sequence diagram for OOFELIE driven by SAMCEF Field
SAMCEF Field can be seen as the default “multi-purpose” GUI that drives OOFELIE, in parallel with dedicated environment that are application dependent.

2 Application to Fluid-Structure Interaction

Fluid-Structure Interaction (FSI) constitutes a major challenge in multi-physics applications. In such a problem, two non-overlapping domains are considered, the first one describing a solid that can be deformed, and the second one describing a fluid media that is in movement.

These two domains only interact at their common faces, which are called fluid-structure interfaces. The fluid applies forces (due to the pressure or the viscous stresses) on the structure, which is deformed consequently. If the structure is deformed, the interfaces are also deformed, and thus the fluid domain, which modifies the fluid flow.

Moreover, if the user is interested to thermal problems, at the interfaces, the temperature must be equal and the thermal fluxes must be conserved.

There is thus a two-way coupling between the fluid and the structure, with exchange of energy (cinematic or thermal) at the fluid-structure interfaces. The structure imposes the deformation of the interfaces, as well as the velocity of these deformations (and possibly the temperature), and the fluid imposes forces to the structure at these interfaces.

2.1 Solution strategy

Such a problem can be solved in two different ways

- Monolithic approach: a single domain is considered and the equations for the structure and the fluid are solved in a single shot, including the constraints at the interfaces
- Staggered approach: the equations on the two domains are solved separately and the necessary data are exchanged at the interfaces by the solvers

In this application, the second approach has been considered and a coupling between OOFELIE® and the well-known CFD software FINE™/Hexa (Numeca International) has been developed.

The solver of FINE™/Hexa, HEXSTREAM solves the equations on the fluid domain and computes the forces at the interfaces. Then, OOFELIE introduces the interface forces in the structural problem, solves this last one and computes the new deformation of the structure, which will induce new positions for the nodes at the interfaces. These displacements, as well as the velocities, are thus computed and transferred to HEXSTREAM, which will deform the fluid mesh according to the displacements at the interfaces, and compute the new values of the fluid flow.

As two different mesh generators are used (SAMCEF Field for the structure mesh, and HEXPRESS™, the mesh generator of the FINE™/Hexa package, for the fluid mesh), the skin meshes, i.e. the meshes at the fluid-structure interfaces, will be different for the structure and the fluid domain. This allows

- to use the best mesh generator available for each domain
- to use non-conform meshes at the interfaces, which is interesting as the fluid mesh generally reclaims more cells than the structure one.

However, such an approach demands operators to transfer variables from one skin mesh to the other.
The transfer operations that are necessary for a F.S.I. problem with our solution are thus

- transfer variables from one skin mesh to another, the two meshes being totally non-conform, even if they represent exactly the same geometry
- transfer the same variables from one solver to another (OOFELIE and HEXSTREAM) in an efficient way

2.2 Interpolation through meshes

2.2.1 Energy conservation at the interfaces

The equality of the displacements and the velocities at the fluid structure interfaces can be written by

\[ u_F = u_S \]

as the fluid values must be equal to the structure ones. In a discrete way, it is written

\[ \bar{u}_F = H\bar{u}_S \]

where \( H \) is the transfer matrix.

On the other side, the forces that act on the structure are the ones induced by the fluid. In a discrete way

\[ \tilde{F}_S = G\tilde{F}_F \]

To ensure that the virtual works of the fluid and the structure acting on the interface are equal, it can be demonstrated that it is necessary to have

\[ G = H^T \]

The only transfer matrix that must be computed is \( H \). This matrix can not have columns with only zero values, because otherwise, its transposed would have lines with only zero and forces acting on some nodes of the structure would be zero. To have non-zero columns in \( H \), it is necessary that all structure nodes appear in at least one stencil of interpolation for the fluid nodes. This is automatically done if the fluid skin mesh is at least as fine as the structure one. As the fluid mesh is generally much finer than the structure one, this should not induce problems. The following figure illustrates a 2D mesh problem with a solid cylinder and fluid around it. The displacements and the velocity must be transferred from the coarse solid mesh to the fine fluid mesh.

![Figure 5: 2D mesh interfaces representative of FSI models](image)

2.2.2 Radial Basis Functions

It is thus necessary to compute a transfer matrix \( H \) giving weights to interpolate a value on a fluid skin node, from values on the structure skin nodes. The method that is used is the Radial Basis Functions (RBF) method.

Let us suppose that a scalar quantity \( u \) is given at the structure nodes \( X = \{ x_1, \ldots, x_N \} \), and that we want to transfer this information to the fluid nodes \( Y = \{ y_1, \ldots, y_M \} \).

A radial basis function interpolant is a function of the form

\[ s(x) = \sum_{j=1}^{N} \alpha_j \phi(\|x - x_j\|_2) + p(x) \]

where \( \phi \) is a conditionally positive definite function of a certain order \( m \) and \( p \) is a polynomial of degree \( m-1 \).
The conditions that we have to impose are

\[ s(x_j) = u_j \quad j = 1, \ldots, N \]

and conditions on the coefficients of \( p \).

Possible functions are

- **Thin-Plate Splines (TPS)**

\[ \phi(\|x\|) = \|x\|^2 \log(\|x\|) \]

- **Euclid Hat function (EU)**

\[ \phi(\|x\|) = \left(1 - \frac{\|x\|}{r}\right)^2 \left(\frac{1}{2} \frac{\|x\|}{r} + 1\right) \]

where the subscript “+” means that only positive values are taken into account. The radius \( r \) defines the support of the radial basis function. From a mathematical point of view, a large support radius (compared to the data density) yields a good approximation order while a small support radius leads to a stable system that can easily be solved. In practice, a reasonable fixed support radius for the fluid-structure-interaction problem has to guarantee a full coverage of the interpolation space.

- **Beckert and Wendland (BW) Compactly supported functions**

\[ \phi(\|x\|) = \left(1 - \frac{\|x\|}{r}\right)^4 \left(4 \frac{\|x\|}{r} + 1\right) \]

As presented here, the RBF method demands the inversion of a square matrix that has a size equal to the double of the number of structure skin nodes. For large industrial problems, this inversion can be very costly, and the method can not be used. An improvement of this method has thus been considered. At the beginning of the computation, every interface is divided into a certain number of patches, whose size \( N \) is prescribed by the user. Every fluid skin node is first localized in the patch, and the global transfer is solved by part. A RBF function is computed on every patch, and the variables at the fluid skin nodes are interpolated through the corresponding function. The more the patches are big, the more precise the interpolation is, but also the more the computation costs.

### 2.3 A communication model: OOFELIE and MPI

The name MPI stands for Message Passing Interface. It is a language-independent communications protocol used to program parallel computers. Large engineering problems are solved on parallel computers using this specification. These problems can be solved from a single program running on several processors (SPMD: single program multiple data) or from several programs inchanging information (MPMD: multiple programs multiple data). Several implementations of this protocol are available to address the variety of existing computer platforms (cluster of Linux PC, Supercomputers ...).

OOFELIE is provided with MPI capabilities and has been successfully compiled with two open source MPI implementations: LAM/MPI and MPICH2. This enables OOFELIE to exchange data with another OOFELIE executable or with another application compiled with the same MPI implementation.

In the framework of the development of FSI solution, OOFELIE and HEXSTREAM are exchanging information using the LAM/MPI protocol.

#### 2.3.1 LAM/MPI

The name LAM stands for Local Area Multicomputer. It is a portable programming environment and development system for heterogeneous computers that features a complete implementation of MPI-1.2 standard and much of MPI-2 standard. Information about LAM/MPI can be found on the following web site: http://www.lam-mpi.org.

#### 2.3.2 MPICH2

MPICH2 is an all-new implementation of MPI from the group at Argonne National Laboratory. This portable implementation of the MPI protocol features all MPI-2 standard functionalities with the exception of support for the "external32" portable I/O format. Information about MPICH2 can be found on the following web site: http://www.mcs.anl.gov/research/projects/mpich2/index.php.
2.4 A coupled resolution scheme

The transfer of the variables from one mesh to another is made in OOFELIE. Displacements and velocities are computed on the fluid skin mesh, and have then to be sent to HEXSTREAM. HEXSTREAM computes then the forces at the interfaces, and sends them to OOFELIE, on the fluid skin mesh, and OOFELIE transfers the variables on the structure skin mesh. In order to reduce the number of connections between the two codes, only one operation is made for each transfer. All the variables to transfer are thus gathered in a single vector.

The technique that is used in this work to transfer the variables from one code to the other is MPI. The two codes are compiled with LAM/MPI. Indeed, LAM/MPI is the only implementation of the standard MPI that can exchange information from one code to another (MPMD).

A complete FSI analysis is performed in three steps

- **The pre-processing step:**
  The user launches the SAMCEF Field GUI that pilots OOFELIE that starts the FINE™/Hexa GUI. These two graphical interfaces are used to prepare the computation. Actually, SAMCEF Field creates the geometrical fluid domain and OOFELIE exports it in a STL format file that is used by the HEXTREAM solver. In this step, several files are created for the communication between the codes. No MPI is used.

- **The processing step:**
  OOFELIE and HEXTREAM are launched in a LAM/MPI environment and exchange information at the fluid-structure interfaces. All the communications are done with MPI commands and the only files that are created are the ones containing the solution (result files).

- **The post-processing step:**
  The post processing software (presently ParaView) is launched and reads the files containing the solution.
The complete sequence diagram is illustrated on Figure 5. It can be seen that the user only interacts on graphical interfaces for the pre-processing and post-processing steps.

3 Application to Optics

For a growing number of applications having to perform in demanding conditions such as space instruments optics (telescopes, optical benches, etc), fibre optics or optical mMicrosystems (MOEMS), the need for high dimensional stability, increased accuracy and predictable performances raises the standards for the design of optical devices. In most cases, the use of computer-based design approaches for optical devices is growing.

However, from telescopes to optical fibres, a more integrated optics (and mechanical, thermal, …) analysis is required for design in laser applications, MOEMS, optical computers, biomedical camera or implants, injected micro prisms, optical switches, etc. While separated applications existed to perform individual analyses for both optical and structural behaviours, an integrated software solution allowing a coupled analysis of these interactions in the same environment was lacking. Based on the same parametric geometry, such a tool would not only allow consistent accurate formulations and design flow, but also open the door to coupled design optimisation.

Open Engineering benefited from a partial funding from ESA to develop such an integrated numerical simulation tool: OOFELIE® for Advanced Optics. This solution allows analysing the coupled interaction between the optical behaviour and its structural medium due to mechanical or thermal solicitations. The resulting application (see Figure 6) is an enhancement to the existing OOFELIE toolkit to enable the dialogue with the optical simulation software ZEMAX®, offering the ability to exchange the geometry of optical surfaces and results associated between the codes, mainly through direct in-memory dialog. The whole is embedded in SAMCEF Field, a consistent pre-post graphical user interface which includes parametric model management.

3.1 An optical solver: ZEMAX

ZEMAX is the software that optical engineers and designers around the world choose for lens design, illumination, laser beam propagation, stray light, freeform optical design and many other applications.

In the framework of this integrated solution, it is the optical design software used to model the optical part of a studied component.

ZEMAX enables the description of the optical problem in term of surfaces. While several types of surfaces are available, Standard and Even Asphere surfaces can be converted into so-called Grid Sag, Zernike Fringe Sag or Zernike Standard Sag surfaces. Those surfaces provide the user with the possibility to add deformations data to the initial description of the surface.

3.2 A communication model: OOFELIE as a DDE client

The general design procedure of an optical device starts with the generation of the optical design. In the integrated solution, the optical analysis software used is ZEMAX. Then, the thermo-mechanical design is performed. The creation of the model is performed in the SAMCEF Field graphical user interface while the solution is computed by OOFELIE’s resolution methods.

Using information related to this mechanical problem, the initial optical design within ZEMAX is modified in order to compare its optical performances with the nominal ones. A frequent modification consists in describing the deformation of the surface of an optical entity by a grid of sag points or by a linear combination of Zernike polynomials.

So, information must be exchanged between OOFELIE and ZEMAX. To reduce human intervention, it is OOFELIE that adds data in the description of ZEMAX’s surface and is going to retrieve optical indicators from ZEMAX.

Dynamic data exchange (DDE) is a powerful feature of the Microsoft® Windows operating environment. DDE provides a protocol through which applications can exchange data of all sorts. DDE always occurs between a client application and a server application. The DDE client application initiates the exchange by establishing a conversation with the server to send transactions to the server. A transaction is a request for data or services. The DDE server application responds to transactions by providing data or services to the client.

OOFELIE was adapted to play the role of the DDE client as ZEMAX can play the role of the DDE server.
This mode of communication between OOFELIE and ZMAX reduces human interventions during data transfers between the two applications.

### 3.3 Coupled resolution method

Specific capabilities, adapted to the simulation of optical problems, were introduced inside OOFELIE and are detailed hereafter.

#### 3.3.1 Mesh improvement

The optical problem is described in ZEMAX and some of its optical surfaces are exported using a standard CAD exchange format (STEP or IGES). These files are then imported in SAMCEF Field where they constitute the basis for the creation of the model of the structure that will support the optical devices.

But, optical surfaces containing aberrations such as even or odd aspheric surfaces may not be very accurately represented using standard CAD exportation exchange formats. Indeed most of the time such surfaces are represented using NURBS (Non-Uniform Rational B-Splines) curves and surfaces.

Often, operations between geometric entities (such as Boolean or cut operations) are changing the parametric description. These modifications can lead to a potential precision loss in shapes representation.

Once the geometry of the model has been created, it is meshed. During this operation the geometry can also be approximated by the meshing process. For example, curved geometries can be represented using linear elements.

The consequence is that the mesh’s nodes might not exactly be on the initial geometry. This potential lack of precision can be problematic for optical systems where the allowed deformations must be close to 10 nanometres. For such cases, the initial geometry could be represented with an error of the same order of magnitude as the requested tolerance for the deformations.

However, each optical surface, in the mechanical problem, is linked to the same optical surface in the optical problem. Therefore, as OOFELIE can retrieve information from ZEMAX, it can perform a verification process of the sag of each node of the mesh belonging to an optical surface. If the sag is not represented with a sufficient precision, OOFELIE can compute the sag value using the analytical equation of the underlying surface (coefficients of this analytical equation are retrieved from ZEMAX) thereby increasing the sag precision.

Figure 7 presents the sag correction for a parabolic reflector surface with a circular aperture.

#### 3.3.2 Generation of a Grid Sag

The deformation of an optical surface due to thermo-mechanical loading can be converted into a Cartesian grid of points loadable by ZEMAX. Each value of the grid represents the sag affected to the discrete point of the grid.

For each optical surface of the mechanical problem, OOFELIE can generate such a file. OOFELIE also automatically modifies the optical problem to:

- Change the type of the surface to GRID SAG.
- Load in the extra data editor of ZEMAX the generated grid file.

For each optical surface in the mechanical problem declared as Grid Sag, the user can specify the numbers of points in each direction of the grid that is going to be generated.

#### 3.3.3 Generation of Zernike coefficients

The deformation of an optical surface due to thermo-mechanical loading can be expressed as a linear combination of Zernike polynomials.

ZEMAX can deal with two families of Zernike polynomials: the Fringe and the Standard polynomials [5]. These two families are also managed by OOFELIE.
For both families, OOFELIE can retrieve the coefficients of the combination following the methodology proposed by Malacara [6]. These coefficients are then automatically introduced into ZEMAX’s extra data editor.

When the surface’s deformations are going to be expressed as a linear combination of standard Zernike polynomials, the user can specify the number of polynomials to use. If Fringe Zernike coefficients are to be used all the 37 polynomials are considered in the retrieving process.

Table 1 presents the seven first coefficients retrieved when the deformations are represented with Fringe Zernike polynomials.

<table>
<thead>
<tr>
<th>Fringe polynomial number</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2.95324e-005</td>
</tr>
<tr>
<td>2</td>
<td>-9.80621e-008</td>
</tr>
<tr>
<td>3</td>
<td>-4.82057e-006</td>
</tr>
<tr>
<td>4</td>
<td>-3.06545e-007</td>
</tr>
<tr>
<td>5</td>
<td>-1.08925e-007</td>
</tr>
<tr>
<td>6</td>
<td>-3.12359e-010</td>
</tr>
<tr>
<td>7</td>
<td>-5.37026e-008</td>
</tr>
</tbody>
</table>

Table 1: First computed coefficients

The retrieving process of Zernike coefficients is executed either at the end of the linear static analysis or after the determination of the transient response of the system.

3.3.4 Management of rigid body motion

According to the type of solicitation applied on the model, the deformation of the surface of an optical device can be quite large. Figure 8 presents the displacement normal to the surface of a mirror due to a force applied on the back of the mirror.

Figure 8: Normal displacement (rigid body motion included)

Figure 9: Normal displacement (rigid body motion removed)

The displacement is dominated by a tilt motion. Figure 9 presents the normal displacement when the rigid motion that describes the tilt movement is removed from the total displacement.

The complex local deformation is highlighted and the area where the pressure force is applied appears.

OOFELIE was adapted in such a way that the rigid body motion parameters that are part of the total displacement are identified and separated from the elastic component of the displacement (see Figure 10).

To preserve the same effect in ZEMAX, the rigid body motion parameters are introduced in the properties of the initial surface in order to change the position of the surface’s vertex and to change the orientation of the surface. Then the elastic displacement is used to describe the shape of the deformed surface using either a grid of sag point of a linear combination of Zernike polynomials.

3.3.5 Modified ZEMAX problem

One output of the integrated simulation consists in a modified optical model: some surfaces of the initial optical problem are converted to Grid Sag of Zernike Sag type of surface. The description of the deformation of the optical
surfaces reflects the impact of the solicitation applied on the supporting structures of the optical device. As the description of optical surfaces is modified, the optical performances of the system are modified too. The comparison of the new performances with the nominal ones gives the designer information related to the perturbation the optical system is submitted to due to the considered applied solicitation. Figure 11 presents the wave front map of an initially perfect optical system when the gravity effect deforms one mirror in the system. Obviously, for a ground based device, the gravity effect can impact in a negative way the performances of the system.

At the writing time of this document, from the modified optical problem, OOFELIE retrieves the following optical performance indicators:

- Merit function value
- Spot diagram
- Wave front map

Figure 10: Decomposition of the total displacement into rigid body motion and elastic contributions

Figure 11: Wave front map for mirror (under gravity effect)

### 3.4 Conclusion

Extensions in terms of computation and data exchange that are specifically targeted to the design of optical systems were introduced into OOFELIE. They have been presented as well as the close connection with the optical analysis software ZEMAX.

In order to increase the precision of the mesh’s nodes belonging to optical surfaces, the sag of these nodes can be recomputed according to the analytical equation used by the optical design.

The deformation of the surface of optical entities can be translated into valuable information for ZEMAX. Either a grid of points or a set of coefficients used in a linear combination of Zernike polynomials can be generated. The rigid body motion contribution contained in the total displacement can also be separated in order to highlight the elastic contribution.

The optical design is then automatically updated using in-memory communication between OOFELIE and ZEMAX. Optical performance indicators are retrieved from ZEMAX. These automated processes minimize the human interventions during the design phase and thus reduce the risk of loss of data integrity.

The fact that all the analysis steps are performed in the same environment is a great step forward in the modelling of opto-thermomechanical systems.

### 4 Application in a context of Concurrent Design

#### 4.1 Concurrent Design

It is observed in practice that a lot of multidisciplinary designs result from iterations between multiple teams, each bringing its own expertise in one specific physical field. This approach usually starts at system level that means a representation of the global interactions between components. This systems approach is sometimes called 0-D or functional approach. The main and most popular tools supporting this process are Microsoft Excel® and Matlab-Simulink®. Starting like this, from a simplified representation of a complex system, allows, in a first phase of design, to quickly identify the constraints, energy flows and interactions between each physical behaviour of the system and to scan a number of possible solutions, within the limitations of the simplified model assumptions and the tools capabilities.

Once a candidate design is selected, the detailed design work is then split between different teams for the further phases of design. These secondary phases increasingly make use of accurate 3D numerical simulations. These are, however,
not able to share and exploit to a significant extent models used in the preliminary conceptual phase. And, reciprocally, the preliminary conceptual phase is not currently able to exploit high accuracy tools efficiently.

However, this classical design approach is an iterative process, which generally requires multiple integrations of design modifications. This results in very time-consuming, costly and inflexible designs.

The Concurrent Design (CD) paradigm offers a marked departure from this classical design process, by moving towards simultaneous and continuous, collaborative, interdisciplinary design made of fast iterations, taking into account multiple design requirements together.

The process implemented in the Concurrent Design Facility (CDF) of the European Space Research and Technology Centre (ESTEC) relies on an Integrated Data Model (IDM) to communicate the needed mission and design parameters and other data between applications/specialties and uses in a very extensive way Microsoft Excel® spreadsheets, plus general and domain-specific tools.

In this context, OOFELIE® was used to develop a demonstrator that illustrates how high accuracy, multidisciplinary tools can be used in a fast design mode, by using model reduction methods. The proposed design flow is illustrated on Figure 12.

4.2 Model reduction

The principle of model reduction consists in processing numerical models that can reveal to be particularly costly in terms of simulation time, to extract models that involve few variables but remain representative of the model behaviour of interest.

A model is reduced in terms of retained degrees of freedom (retained DOF). These DOF correspond to the variables in terms of which the behaviour of the SEM is expressed. In practice, the number of retained DOF can vary typically from a few DOF (a MEMS actuator) to thousands of DOF (the wing of an aircraft) depending on the application, as illustrated on Figure 13.

Reduced models, called Super Element Models (SEM) in the OOFELIE terminology, are particular elements that can be used as any other finite elements: they can be simulated stand alone, combined together or included in a classical finite element model. The reduction procedure in time and memory consuming, but this operation is performed only once. This allows also dissociating the generation step and the analysis steps. Furthermore the same SEM can be involved in multiple analyses or duplicated many times in the same model without additional cost.

The benefits of using reduced model are the following:
- Dissociate generation step (costly) and analysis steps (straightforward)
- Reduced simulation time
- Decrease required memory for simulation
- Re-use of models without additional cost
- Protection of Intellectual Property
- Ability to export finite element models to different simulation environments
Model reduction, also called sub-structuring, is a practice widely used in linear structural mechanics. Nevertheless, features addressed in this project were challenging in the following ways.

A first requirement was the generalisation and the adaptation of well known reduction methods, based on model projection techniques, to multidisciplinary models involving strong coupling between the physics. This aspect was not straightforward and multiple approaches have been considered. In the case of thermo-mechanical applications addressed in this project, an innovative approach has been successfully evaluated and implemented.

The second critical point consisted in taking into account model parameters in the reduction algorithms. Even in classical 3D model simulations, the parameterisation of the models does not correspond to a feature commonly integrated in most numerical simulation software. However, research activity has been conducted and methods based on a surface-response approach have been successfully implemented.

As illustrated on Figure 14, parameterised multiphysics reduced model of OOFELIE is an self-consistent data structure that can be generated from different sources of models and that is interfaced with multiple simulation environments.

4.3 A communication model: OOFELIE physical classes embedded in Python

A challenging aspect of this project was the ability to establish an efficient dialog between multiple independent applications. Communication requirement involved data exchange but also access to member functions in the context of Object-Oriented Programming (OOP).

An efficient solution was provided by interfacing OOFELIE physical classes with the Python interpreter.

4.3.1 Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but an effective approach to Object-Oriented Programming (OOP). Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems. New built-in modules are easily written in C or C++. Python is also usable as an extension language for applications that need a programmable interface.

4.3.2 The COM protocol

COM stands for Component Object Model. It is a technology from Microsoft that provides a language-neutral binary standard for building components in one language that are reusable in another one. Components (binary code) developed following the COM standard are created with one programming language (C++ for instance). Those objects are then published and are finally available as reusable components in another programming environment (Visual basic for example). COM enables thus the integration of code developed in different languages.

The first important thing to understand with COM is the fact that COM does not constitute an object-oriented language but a standard. COM does not specify how an application should be structured. It is up to the programmer to manage the language, the structure, and the implementation of its application. COM thus specifies an object model and programming requirements enabling the interaction between those COM components.
The COM world can consequently be virtually separated in two parts. On one side, there is the infrastructure (the back end) provided by an operating system (Windows) which consists in a set of services (API) and compilers that can support the COM standard. On the other side, there are the components created with those compilers that can be published and used.

The “Win32 Extensions” provide the Python language with new capabilities that satisfies the requirements of the COM communication protocol. Therefore, the dialog between any Python–based applications and other software implementing a COM interface is straightforward.

### 4.3.3 Python bindings for OOFELIE classes

OOFELIE physical classes were interfaced with Python. This is often referred to as providing OOFELIE with “Python bindings”. Those bindings were generated by using SWIG, a tool used to connect components written in C and C++ with a variety of high-level programming languages. With those developments, the user has now the choice between two command line interpreters. On one hand, the default command line interpreter of OOFELIE. It is the standard interpreter of OOFELIE which is used to interface OOFELIE with SAMCEF Field. On the other hand the Python interpreter which enables the integration of OOFELIE physical classes to be used in simulations already based on the Python language.

As most Microsoft applications implement the COM protocol, the dialog between Microsoft Excel and OOFELIE capabilities was achieved in terms of client/server communication through the COM capabilities of Python.

Communications between OOFELIE and Microsoft Excel have been established by improving the OOFELIE interfacing capabilities. OOFELIE classes are now available as Python modules and the OOFELIE executable is provided with COM objects and interfaces that can be registered, in the operating system, in order to enable Microsoft Excel to cooperate with a reduced finite element model. Dialog between the two applications is obtained in terms of client/server communication.

### 4.4 OOFELIE in MS Excel

In order to provide a user-friendly interface for reduced model simulations that fully integrates in the CDF platform, a dedicated interface was developed in Microsoft Excel. This new product, “OOFELIE in MS Excel”, introduces in one of the mostly used software in engineering the powerful capabilities of finite element methods, through reduced models. This last point is an innovative breakthrough in terms of simulations tools.

The “OOFELIE in MS Excel” interface, illustrated on the Figure 16, consists in an additional menu in Excel and specific worksheets dedicated to steady-state and transient simulations of reduced models.

The reduced models are accessed and simulated by OOFELIE that behaves as a server (using the COM dialog protocol) to answer the Excel requests. Indeed, Excel does not perform any simulation but is used as an interface to dialog with the user and graphically display transient results.

The workbook is initialised by loading a SuperElementModel (SEM, reduced model in the OOFELIE terminology). During this operation, a client/server is established between Excel and OOFELIE that access the SEM and provides Excel with the description of the SEM content: parameters, degrees of freedom (DOF), load cases, … These data are used to create in the worksheets the appropriate tables corresponding to the SEM structure.

Two worksheets are created, respectively dedicated to steady-state and transient simulations. Each worksheet includes the appropriate tables for defining simulation settings (post-processing) and analysing results (post-processing).

Pre-processing tables correspond to the possible user entries in terms of parameters, boundary conditions, loads and simulation settings. Post-processing tables involve, depending on the analysis, DOF value display, as well as first and second derivatives, reactions and result plot selection.

In the transient analysis worksheet, a dedicated procedure has been introduced to handle time dependent functions (sinus, ramp, steps).
A particular attention was placed on the worksheet appearance and behaviour in order to guide the user in the tables and provide robustness in terms of data consistency. A colour scheme was defined, in accordance with the CDF usage: internal data cells are protected, consistency of entries are checked and table content updated according to user actions to avoid inconsistency between user entries and displayed results.

When user entries are filled in a valid way, a simulation can be performed. At this moment, a new client / server connexion is established between Excel and OOFELIE. The user entries are provided to OOFELIE that instantiates a superelement in an internal simulation model, applies the expected boundary conditions and loads, and performs the required simulation. Then, OOFELIE provides Excel with the simulation results in terms of DOF, derivates ... These results are displayed in the worksheet according to the type of results and analysis.

When results are available in the worksheet, a backward substitution (see Figure 12) can be performed. A client / server connexion is established between Excel and OOFELIE, Excel send the DOF values to OOFELIE that access the initial 3D definition, the reduced model and generated SAMCEF FIELD results that correspond to the expansion of the reduced model results to the whole 3D model. These results can be visualised in a SAMCEF FIELD session.

Figure 17 summarises the different steps involved in the simulation of a reduced model using “OOFELIE in MS Excel”, presented here above, and illustrates the interactions between Excel and OOFELIE.
5 Conclusion

OOFELIE is the simulation framework of Open Engineering dedicated to the development of innovative solutions. OOFELIE consists in a collection of C++ classes addressing all the steps of multi-physics numerical simulation: model description, solution scheme and results processing.

OOFELIE capabilities are in constant evolution in term of data management and resolution strategies. For example, in order to fulfil the requirements of three new applications partially funded by ESA contracts, innovative developments were introduced in OOFELIE.

Taking advantage of the existing MPI capabilities of OOFELIE, a fluid structure interaction (FSI) solution was set up that introduced a close connection between OOFELIE and Fine/Hexa not only at the data transfer level but also at the algorithmic level.

In term of Optics, OOFELIE was provided with DDE communication capabilities. This opened the door of interoperability with ZEMAX. The development inside of OOFELIE of the post processing capabilities of structural results in term of a grid of sag point or as a linear combination of Zernike polynomials enabled the introduction of a new application that automates most of the operation performed in the coupled analysis of opto-thermo-mechanical problems.

While high fidelity simulation requests complex numerical models leading to long resolution times, it exists a need for almost real-time simulations, as required in the context of the Concurrent Design facility of ESA, that can be achieved using model reduction techniques. In this project, OOFELIE was provided with extensions to generate so-called superslement models (SEM) that are used to represent, with a small number of unknowns, the behaviour a large numerical model. The ability to perform SEM simulation within Microsoft Excel has also been introduced. To this end, the physical classes of OOFELIE have been interfaced with the Python interpreter. The Win32 extension of Python enables the interoperability with Microsoft application using the COM communication protocol. This communication scheme is used to enable Excel to interact through Python with OOFELIE components.

For commercial applications, Open Engineering has embedded OOFELIE with SAMCEF Field, the new generation of GUI developed by the SAMTECH group.

OOFELIE is not open source software. It is a commercial application that constitutes the basis of all the commercial products developed and proposed by Open Engineering to its partners. However, OOFELIE has several open aspects. Firstly, the OOFELIE executable comes with its own command line interpreter which enables the user to describe its problem but also to introduce its own resolution strategy if the provided ones are not sufficient. For advanced users, OOFELIE capabilities are also proposed under the form of a collection of shared libraries. This version is referred to as the SDK version of OOFELIE. Using Object Oriented Programming techniques, this version enables extending OOFELIE capabilities (for the re-engineering of existing software) or introducing OOFELIE components into another application (to add it FEM capabilities for instance). Finally, complete access to the source code of OOFELIE is provided to some strategic partners of Open Engineering.

References