



Contents lists available at <http://qu.edu.iq>

Al-Qadisiyah Journal for Engineering Sciences

Journal homepage: <http://qu.edu.iq/journaleng/index.php/JQES>



Analysis of coupling methods: A review

Jean Bosco Samon*, Jemimah Maurelle Ndomou, and Damasse Harold Fotsa

Laboratory of Mechanics, Materials and Photonics, Department of Mechanical Engineering, ENSAI-University of Ngaoundere, Bp 455, Cameroon

ARTICLE INFO

Article history:

Received 15 June 2022

Received in revised form 20 August 2022

Accepted 27 September 2022

Keywords:

Multiphysics

Multiphysics system

Mechatronics

Mechatronics system

Coupling methods

ABSTRACT

The *multiphysics* field is a branch of physics whose objective is to couple at least two *physical systems*. Each is governed by its own principles of evolution or equilibrium such as balance laws or constitutive laws. Many engineering problems can only be described correctly by coupling fields of physics that have historically been developed and taught separately. These problems require on the one hand a good understanding of each physical domain, but above all an analysis of the coupling mechanisms of these physical domains, in order to propose a relevant model capable of solving the problem. A challenge in the *multiphysics (mechatronics)* field is the construction of coupled multiphysics models from experimental observations, as well as the analysis of their mathematical properties. The mathematical analysis of the coupled model must be able to show the well-posedness of the problem at the defined boundary and initial values. For this reason, we have identified several coupling methods: Newton, Gauss-Seidel, JNFK, and direct and explicit coupling. From these methods, it appears that the Newton method is suitable for the coupling of the different disciplines of Mechatronics. A summary table shows the comparative advantages of each method.

© 2022 University of Al-Qadisiyah. All rights reserved.

1. Introduction

The compartmentalization of trades and physical disciplines, combined with the weakness of computer resources with the emergence of numerical calculation has led to the creation of specialized simulation tools. However, the phenomena studied often mix several disciplines. The need for Multiphysics simulations thus appeared very early on [20]. The field of Multiphysics is a branch of physics whose objective is to couple two or more physical systems, each governed by its own principles of evolution or equilibrium, such as balance laws or constitutive laws [1]. Many engineering problems can only be described correctly by coupling fields of physics that have historically been developed and taught separately [21]. This is called coupling. There are two traditional responses to this need. The first is to incorporate simplified models (which can be fed by reference calculations) from other physics disciplines into a specialized code. The

second is the sequential execution of several codes exchanging information. Both approaches have in common the implementation of penalty levers to guarantee the conservatism of the physical response obtained by the modelling. The increase in computing capacity and the desire to improve the modelling associated with the complexity of the technologies studied are increasing the interest in Multiphysics simulations. These objectives are recurrent, but it is not at all certain that they will be achieved if attention is not paid to the way the coupling is carried out. Having separate simulations communicate may introduce limitations in stability, accuracy and robustness that are stronger than those of the components. In addition, the exchange of information may itself be more costly than the execution of the coupled codes. These challenges have led the scientific community to develop efficient and robust coupling techniques. This article is part of this

* Corresponding author.

E-mail address: jboscocosamon@gmail.com (Jean Samon)



process. After defining the coupling system, we will present two families of associated coupling methods. We have deduced a method suitable for mechatronic systems.

2. Some definitions of the concepts

According to [2] coupling is an approach that can be used as a unifying basis for modeling the different domains of mechatronics. We can therefore say that coupling is the unambiguous connection between two or more inputs of the same system. It is then for us to define a coupled system. According to [23] a system is a set of entities that interact and evolve over time. The entities can be continuous or discrete fields. Their evolution is governed by ODEs1 or PDEs2. He thus defines a coupled system as a system made up of several interacting subsystems. They can be differentiated by the model (conservation law, behavioral, etc.) or the numerical methods used. These subsystems interact through interfaces. Geometrically, an interface can correspond to a boundary or to a part or the whole spatial domain. As a sketch, we have fig. 1 below.

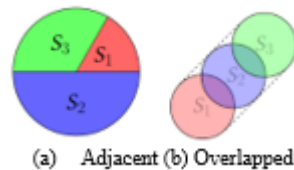


Figure 1 - Arrangement of the subsystems [2]



Figure 2 - Family of couplings [3]

This figure shows us that the subsystems can be adjacent (fluid-structure, solid body dynamics), or overlap each other (overlapped). Partial overlaps are also possible (e.g. electromagnetic-mechanical).

In their paper entitled "Stabilization of staggered solution procedures for fluid-structure interaction analysis", [3] discussed the difficulty of stability in the coupling method. It is this difficulty that has challenged the researchers to the point of leading them to a solution to the problem, namely the different coupling algorithms. The latter are grouped into two families qualified respectively as weak coupling and strong coupling which we shall begin by presenting. Let us note that the type of coupling between two subsystems (each physical for example) can be qualified as "Strong" or "Weak" as shown in figure 2. Concerning the "strong coupling": there is a significant influence of the two subsystems on each other. For "weak coupling": only one subsystem has a significant influence on the other subsystem. The fundamental difference between these two families comes from the way of imposing equilibrium between the different physical systems, i.e. the action-reaction principle. It can also be compared in terms of their speed of convergence. This is measured by the coupling force. The coupling "strength" is determined by the equations of the mathematical model. The value of the coupling "strength" between the subsystems is given by a scaled writing of the model equations which allows some scaled numbers to appear.

2.1 Weak coupling family

This is the first type of coupling to be used in design offices. It is very present in industrial software. This type has the advantage of concentrating the calculation effort on the zones that require it. However, it is mainly reserved for cases where the local detail has little or no influence on the rest of the structure on a global scale. For this reason, a mathematical modeling of weak coupling will be presented in the following. Consider a dynamic system whose state is described by the vector $u(t)=\{u_1(t),u_2(t)\}$ and governed by the following differential system:

$$\begin{cases} \dot{u} = \mathcal{F}(u, t); \\ u(t = 0) = u_0. \end{cases} \tag{1}$$

The coupling between the two subsystems described by their respective entities $u_1(t)$ et $u_2(t)$ or components \mathcal{F}_1 et \mathcal{F}_2 are both functions of $u_1(t)$ et $u_2(t)$ The coupling between the two subsystems, described by their respective entities or components, is said to be weak if only subsystem 1 has a significant influence on the second one.

$$\mathcal{F}_1(u_1, u_2, t) \approx \mathcal{F}_1(u_1, t). \tag{2}$$

That is:

$$\begin{cases} \dot{u}_1 \\ \dot{u}_2 \end{cases} = \begin{cases} \mathcal{F}_1(u_1, t) \\ \mathcal{F}_2(u_1, u_2, t) \end{cases} \tag{3}$$

In the weak coupling model, each of the partial differential equations is solved separately. In the weak coupling model, each of the partial differential equations is solved separately, which requires the transfer of the results of one partial differential equation to the other. In such a case, each system may involve a different mathematical type of variable (real or complex), different time constants or pulses, and different study areas.

2.2 Strong coupling family

In the case where the influence of the local model extends beyond its definition domain, it is easy to understand that weak coupling methods will produce a more or less erroneous solution depending on the situation [4-5]. Therefore, the only way to build an exact coupled solution is to bring the information from the solution of the local problem to the global level. This is the principle of strong coupling methods. In this type of strong coupling, we have direct and iterative coupling methods.

2.2.1 Direct coupling methods

These methods take their name from the fact that the coupling between two models is considered by means of a direct calculation, i.e., without using iterative processes. They can be classified into two main categories: structural reanalysis methods and static condensation structural reanalysis methods. The principle of structural reanalysis is to calculate the response of a structure after a (often localized) change from the principle of structural reanalysis is to calculate the response of a structure after a (often localized)

modification from the knowledge of the response before modification. Most of the time, the procedure consists in evaluating the response of the initial system under different elementary stresses (e.g., m) representative of the modification made to the structure. This is then done by correcting the initial stiffness operator by a matrix of rank m . This correction can then be carried out in a non-intrusive manner with respect to the operator through the Sherman-Morrison and Woodbury formulae Woodbury formulae [8]. Let $\{U_i\}_{i=1\dots m}$ be a family of vectors constructed from the responses under elementary stresses. If $U = [U_1, \dots, U_m]$ then the response of the modified structure can be calculated in a non-intrusive way (i.e., without requiring the assembly and factorization of the modified system):

$$(K + UU^T)^{-1} = K^{-1} - K^{-1} U (U^T K^{-1} U + I)^{-1} U^T K^{-1} \tag{4}$$

Structural reanalysis methods are therefore based on an a posteriori modification of the solution, in contrast to static condensation methods. We take the example (Fig.3) of a structure occupying the domain Ω that is partitioned into a global part Ω_G and a local part Ω_L

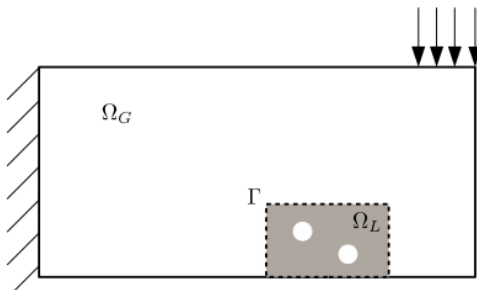


Figure 1 - Partition of the study area (Park and Felippa, 2000).

It is assumed that a localized defect in Ω_L is taken into account. Even if the solution in Ω_G can be affected by the alteration of the model in the local area, the finite element representation of the model defined in Ω_G always remains the same. If the model defined at Ω_L were to evolve or be modified many times (for example during a parametric study), a complete recalculation of the whole structure would require a significant and unnecessary additional cost. Static condensation sub structuring provides a solution to this problem by allowing the construction of a model representing the behavior of the structure in Ω_G involving only the degrees of freedom of the interface Γ [6-7]. We consider here the case of a displacement connection under the assumption that the mechanical problem is formulated in primal variable.

$$\begin{bmatrix} K_G & C \\ C^T & K_L \end{bmatrix} \begin{bmatrix} U_G \\ U_L \end{bmatrix} = \begin{bmatrix} F_G \\ F_L \end{bmatrix} \tag{5}$$

The global model $K_G U_G + C U_L = F_G$ can be condensed so that only the interface displacement has to be considered. The superscripts i indicate the internal degrees of freedom of the models, and the superscripts b indicate the degrees of freedom of the interface.

$$\begin{bmatrix} K_G^{ii} & K_G^{ib} \\ K_G^{bi} & K_G^{bb} \end{bmatrix} \begin{bmatrix} U_G^i \\ U_L^b \end{bmatrix} = \begin{bmatrix} F_G^i \\ -C^b U_L^b \end{bmatrix} \Rightarrow$$

$$(K_G^{bb} - K_G^{bi}(K_G^{ii})^{-1}K_G^{ib})U_L^b + C^b U_L^b = -K_G^{bi}(K_G^{ii})^{-1}F_G^i$$

The Schur complement of K_G , noted can thus be used to substitute the global model in equation (5). This gives the following equation 6:

$$\begin{bmatrix} S_G & C^b \\ (C^b)^T & K_L \end{bmatrix} \begin{bmatrix} U_G^b \\ U_L \end{bmatrix} = \begin{bmatrix} -K_G^{bi} & (K_G^{ii})^{-1} & F_G^i \\ F_L & & \end{bmatrix} \tag{6}$$

In this case, static condensation is used as a decoupling method to connect a fixed global model to a detailed and possibly modifiable local model. Sometimes, this method is also used to provide an indelible finite element model in the form of a "black box" to protect an industrial secret for example. Indeed, the only knowledge of the Schur complement allows an exact consideration of the model, without having access to the details of the latter (geometry, loading cases). In the context of a generic and succinct presentation of the most used coupling methods to date, we refer here to static condensation. However, it should be noted that the process of sub structuring finite element models is more general and can also be used in dynamics and modal calculation. Also, nothing prohibits the joint use of static condensation with other coupling methods, as is the case in Hirai who combines static condensation with structural reanalysis [9].

2.2.2 Iterative coupling methods

As we have just seen, direct coupling methods imply a certain level of interoperability of the solvers used by each of the coupled models. Interoperability of the solvers used by each of the coupled models. The other major family of coupling methods commonly used are the iterative methods. Successive resolutions are performed on the different parts of the model that are to be coupled. Among the common iterative coupling methods are those based on sub structuring [22]. As presented earlier, the structure can be partitioned into substructures. The idea is then to solve the interface problem, but this time in an iterative way (for example via a conjugate gradient method).

The other major class of iterative coupling methods concerns the so-called Schwarz methods [8]. The alternating Schwarz method, historically the oldest, proposes to split the study domain into several domains. The problem is then solved alternately in each of the sub-domains. If we assume a division of the domain into two parts Ω_1 and Ω_2 , then the solution of the problem according to the alternating Schwarz method will be done as follows (Equation 7):

$$\begin{cases} K_1 U_1^{n+1} = F_1 \\ U_1^{n+1}|_{\Gamma} = U_2^n|_{\Gamma} \end{cases}$$

Alternating Schwarz:

This method thus assumes that the problem is solved on each sub-domain

$$\begin{cases} K_2 U_2^{n+1} = F_2 \\ U_2^{n+1}|_{\Gamma} = U_1^n|_{\Gamma} \end{cases} \tag{7}$$

successively. With the advent of computers with distributed architecture, the parallel Schwarz method (Equation 8) appeared. In this case, at each iteration, the problems per subdomain are solved in parallel, in order to take advantage of the potential of such computational means.

$$\begin{cases} K_1 U_1^{n+1} = F_1 \\ U_1^{n+1}|_r = U_2^n|_r \end{cases}$$

Schwarz parallel

$$\begin{cases} K_2 U_2^{n+1} = F_2 \\ U_2^{n+1}|_r = U_1^n|_r \end{cases} \quad (8)$$

In the case where we consider a division of the structure into two sub-domains, these two methods are equivalent since the sequences of solutions thus formed are the same in Ω_1 or Ω_2 . We also speak of a multiplicative Schwarz method for the alternate method and an additive Schwarz method for the parallel method. The rewriting of these methods in the form of a fixed point reveals a sum of operators defined on Ω_1 and Ω_2 respectively for the parallel method and a product for the alternate method.

Finally, we will mention here the "finite element patch" type methods [17-18], generally used for solving multiscale problems. The idea is to consider the solution U as the sum of a global contribution U_G and a local contribution U_L provided by the patch so that $U = U_G + U_L$. As with the Schwarz methods, different strategies are then possible for the imposition of boundary conditions at each iteration. The main interest of such a method is to be able to calculate a local correction in a flexible way, i.e., to make the definition of the local patch independent of the characteristics of the global model. Contrary to direct methods, iterative methods require only weak interoperability between the different models (and thus solvers) used. This makes them very good candidates for the development of non-intrusive methods.

2.2.3 Connections between different models

Whatever the coupling method used, it is necessary to establish a connection between the different models involved. Two types of connection can be distinguished: Interface connections, which aim to establish relationships between the quantities of interest at the interface. In this category, we will find weak connection methods such as the "Mortar" method [10-12]. The principle is to impose the equality of quantities (displacement, effort) in the weak sense, i.e., through a duality relation. Such a connection can be used within strong coupling methods to lead to hybrid finite element methods.

Another example is the Nitsch method [13-15] which allows a Dirichlet edge condition to be imposed in a weak form, by adding an extra work term in the vibrational formulation of the problem. Volume connections through an energy average. In this case, the connection area between the different models is no longer an interface but a covering area: this is the Harlequin method [16]. While a volume connection may offer more flexibility in defining the connection area, it is also much more intrusive than an interface connection method.

This is because a volume energy connection involves the simultaneous consideration of the quantities defined in each of the models. An interface connection, on the other hand, only requires independent unidirectional exchanges between the different models. Despite all these efforts, it sometimes appears that multi-scale/multi-model calculation methods cannot be used in practice. Indeed, in an industrial context, most numerical simulations are performed using commercial software that has been developed and certified for predefined applications. Thus, it is not always convenient to use such software to implement a given multi-scale/multi-model calculation. Moreover, the recent development of supercomputers now makes it possible to carry out very ambitious simulations. Current practices therefore increasingly consist in coupling different models rather

than integrating all the required specificities into a single model. To overcome these difficulties, many authors have developed methods. This is what will be presented in the rest of this article.

3. Stationary coupling methods

Suppose we have two codes f_1 and f_2 working on the variables $x_2 \in \mathbb{R}^{d_2}$ and $x_1 \in \mathbb{R}^{d_1}$. f_1 gives us x_1 knowing x_2 , which we will note $x_1 = f_1(x_2)$ and vice versa, f_2 gives us x_2 as a function of x_1 , i.e. $x_2 = f_2(x_1)$. Carrying out the coupling, therefore, consists of looking for the global solution which verifies equation 9.

$$\begin{cases} x_1 = f_1(x_2) \\ x_2 = f_2(x_1) \end{cases} \quad (9)$$

Often the coupling will be formulated with a dependence of the functions f_i in x_i :

$$\begin{cases} x_1 = f_1(x_1, x_2) \\ x_2 = f_2(x_1, x_2) \end{cases} \quad (10)$$

There may be two reasons for this. This may be a choice not to try to converge each of the codes precisely until the common solution is reached. This is close to an intricate solution method. It also happens that it makes no sense to converge a code on x_i without simultaneously updating x_j . By default, we will assume that the coupling is of the form (10). We will also sometimes rewrite the system in the following form:

$$\begin{cases} F_1(x_1, x_2) = f_1(x_1, x_2) - x_1 = 0 \\ F_2(x_1, x_2) = f_2(x_1, x_2) - x_2 = 0. \end{cases} \quad (11)$$

It is possible to couple more than two equations with the techniques we present here. As the generalization is simple, to simplify the writings, we will limit ourselves to couplings between two disciplines.

3.1 Gauss-seidel method

a. Presentation of the method

The Gauss-Seidel method (often referred to as the Picard method or iterations in the literature), which implements a flexible coupling, is very common and easy to implement. It is applied to $AX=B$ (linear) systems with the matrix A having to be diagonal dominant. In this method, it is simply a matter of iterating on the resolution of the codes until convergence. And one does not stop the iterations from precision. Nevertheless, given that the main concern is to see if the sequence of iterations converges and if so to what order. We will start by presenting the sequence of iterations, then we will find an equivalence with the fixed point equation and then we will check the convergence. Let us note with an exponent n the iteration index. From initial data $(x_1^{(0)}, x_2^{(0)})$, we iterate by posing:

$$\begin{cases} x_1^{(n+1)} = f_1(x_1^{(n)}, x_2^{(n)}) \\ x_2^{(n+1)} = f_2(x_1^{(n+1)}, x_2^{(n)}) \end{cases} \quad (12)$$

$$\varepsilon_1^{(n+1)} = \|x_1^{(n+1)} - x_1^{(n)}\| \text{ et } \varepsilon_2^{(n+2)} = \|x_2^{(n+1)} - x_2^{(n)}\|$$

We generally stop when the increments become small enough. We use this criterion for want of a better one, and we see that it does not give any information on the distance to the solution. It is easy to see that the method if it converges, can only converge to a solution to the problem. However,

convergence is not systematic.

b. Equivalence with the fixed point

The so-called "fixed point" algorithm is used to solve an equation of the type $x = f(x)$. It consists in iterating as follows:

$$x^{(n+1)} = f(x^{(n)}). \tag{13}$$

The Gauss-Seidel method can be seen as a fixed point by posing:

$$x^{(n)} = \begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix} \text{ et } f(x^{(n)}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix}. \tag{14}$$

If one of our functions, say f_1 does not depend on the variable it is used to calculate, i.e. x_1 (as is often the case), we notice that each iteration is equivalent to $x_2^{(n+1)} = f_2(f_1(x_2^{(n)}), x_2^{(n)})$. This is another way of getting back to the fixed point, which has the advantage of reducing the dimension of the space in which we work. The following discussion of the Gauss-Seidel method uses this equivalence. We will therefore be interested in the properties of the fixed point.

c. General condition for convergence

Convergence condition: Let E be a complete metric space and f an application from E into E . If f is k contract i (i.e. $\forall(a, b) \in E^2, d(f(a), f(b)) \leq kd(a, b)$ with $k < 1$, d is the distance between two points of E), then there exists a unique fixed point x_{sol} of f and any sequence satisfying $x^{(n+1)} = f(x^{(n)})$ converges to x_{sol} .

d. Practical conditions for convergence

The last remark leads us to look at the neighborhood of the solution to determine whether the algorithm can converge. We use the Taylor-Lagrange inequality for a function f in R^m in R^m of sufficient regularity. We note J_a the Jacobian of f at a . If there is a real M such that $\|J_{a+h}\| \leq M \forall t \in]0,1[$ then:

$$\|f(a+h) - f(a)\| \leq M \|h\|. \tag{15}$$

We note x_{sol} the fixed-point solution (we assume the existence and uniqueness of the solution here). It is assumed that there exists a real M such that $\|J_x\| \leq M \forall x \in B(x_{sol}, r)$. $B(x_{sol}, r)$ is the ball of center x_{sol} and radius r , any real. Then we have more generally:

$$\forall a, b \in B(x_{sol}, r), \|f(b) - f(a)\| \leq M \|b - a\| \tag{16}$$

Thus, if $M < 1$, this shows that f is contracting on $B(x_{sol}, r)$ and that the algorithm converges if $x^{(0)} \in B(x_{sol}, r)$ from the above. It is therefore sufficient that $\|J_x\| \leq M < 1$ around the solution. If we use the norm L^2 we can write the following sufficient condition ($\rho(A)$ denotes the spectral radius of the matrix A , i.e. $\rho(A) = \max_i |\lambda_i|$ where the λ_i are the eigenvalues, possibly complex, of A).

$$\|J_x\|_2 = \sqrt{\rho(J_x^t J_x)} \leq M < 1 \tag{17}$$

This equation is valid in the vicinity of the solution, and the initialization is chosen in this vicinity. As before, the norm of the Jacobian allows us to reduce the speed of convergence of the algorithm. We can look for a less demanding condition by using its properties: $\forall A$, for any matrix norm we have: $\rho(A) \leq \|A\|$.

$\forall A, \forall \epsilon$ there exists a matrix norm such that $\rho(A) \leq \|A\| \leq \rho(A) + \epsilon$. Thus if we have, in the solution x_{sol} , $\rho(J_{x_{sol}}) < 1$, for a certain norm we have $\|J_{x_{sol}}\| \leq M < 1$. By continuity of the norm, we then have with the same norm $\|J_x\| \leq M' < 1$ on a neighborhood of x_{sol} . We thus have a sufficient condition of local convergence (thanks to the equivalence of norms in finite dimension. It suffices to show the convergence of the sequence for any norm), less demanding than the previous one since the first proposition of the lemma tells us that $\rho(A) \leq \sqrt{\rho(A^t A)}$ $\forall A$. We will try to show that $\rho(J_{x_{sol}}) < 1$ is in fact a necessary condition for "good behavior of the algorithm" in a sense that we will specify. We will then say, by abuse of language, that it is a necessary condition of we will then say, by abuse of language, that it is a necessary condition of convergence.

e. Variant with relaxation

A variant of the fixed-point method is to introduce a relaxation $\alpha > 0$. This is called over-relaxation when $\alpha > 1$. The iterations are then written.

$$x^{(n+1)} = (1 - \alpha) x^{(n)} + \alpha f(x^{(n)}) \tag{18}$$

Let us see what this gives us about the criterion and the speed of convergence. We see that by posing $g = (1 - \alpha) Id + \alpha f$ with Id identity matrix, we return to the classical fixed point with $x^{(n+1)} = g(x^{(n)})$. The previous results therefore apply to g . Let us first consider the one-dimensional case.

One-dimensionnel case :

We have $g' = (1-\alpha)\alpha f'$. The fixed point converges at its maximum speed when the derivative of the function is zero in the solution. This leads us to an optimal relaxation:

$$\alpha = \frac{1}{1 - f'(x_{sol})}$$

This value will generally be unknown since it is based on a property of f in the solution x_{sol} which is unknown a priori. We will therefore look more generally at the behaviour of the algorithm as a function of α . The convergence criterion $|g'(x_{sol})| < 1$ gives us $f'(x_{sol}) \in]1 - \frac{2}{\alpha}; 1[$. The speed of convergence depends on the maximum value of $|g'|$ around the solution x . Compare g' to f' . $g' - f' = (1 - \alpha)(1 - f')$. Since $f' < 1$, we have $g' > f' \alpha < 1$. Now compare g' to $-f'$. $g' + f' = (1 - \alpha) + (\alpha + 1) f'$. Thus, $g'(x) > -f'(x)$, $\alpha < \frac{1+f'(x)}{1-f'(x)}$. Thus, we have:

- If $f'(x_{sol}) < -1$, i.e. if the algorithm, without relaxation, does not converge with an oscillating behavior, a relaxation $\alpha \in]0, \frac{2}{1-f'(x_{sol})}[$ allows it to converge. The optimum is $\alpha = \frac{1}{1-f'(x_{sol})}$.
- If $f'(x_{sol}) \in]-1, 0[$, one can improve the speed of convergence with $\alpha \in]\frac{1+f'(x_{sol})}{1-f'(x_{sol})}, 1[$. The optimum is at $\alpha = \frac{1}{1-f'(x_{sol})}$.
- If $f'(x_{sol}) \in]0, 1[$, one can improve the speed of convergence with $\alpha \in]1, \frac{1+f'(x_{sol})}{1-f'(x_{sol})}[$. This is called over-relaxation. The optimum is at

$$\alpha = \frac{1}{1 - f'(x_{sol})}$$

- If $f'(x_{sol}) > 1$, i.e. if the algorithm, without relaxation, has a non-convergent behavior without oscillation, it will keep this behavior $\forall \alpha > 0$. $f'(x_{sol})$ Since the value of α is not known a priori, the difficulty is to determine α .

More than one-dimensional case:

To improve the convergence of g we will try to minimize the spectral radius of its Jacobian at x , the solution to the problem. We have the following general property, which is easy to check, where $\sigma(A)$ denotes the spectrum of A , i.e. the set of its eigenvalues: $\forall A, \forall P$ polynomial $P(\sigma(A)) = \sigma(P(A))$. Thus, the previous analysis, done in one dimension, applies to each of the eigenvalues of the Jacobian of g . Let us denote λ_i its eigenvalues, and $\lambda_i(\alpha)$ those obtained with relaxation. We have: $\lambda_i(\alpha) = 1 + \alpha(\lambda_i - 1)$. We seek to minimize $\max_i |\lambda_i(\alpha)|$. As before, $\lambda_i > 1 \Leftrightarrow \lambda_i(\alpha) > 1 \forall \alpha > 0$. Therefore, relaxation does not allow the algorithm to converge if $\exists \lambda_i > 1$, and we, therefore, consider that $\lambda_i < 1 \forall i$. Thus, we have $\lambda_i'(\alpha) = (\lambda_i - 1) < 0$. All functions $\lambda_i(\alpha)$ have the same monotonicity and are all 1 in 0. Moreover, it is easy to verify that if $\lambda_i \neq \lambda_j$ then $\lambda_i(\alpha) \neq \lambda_j(\alpha) \forall \alpha \neq 0$. Let λ_j be the eigenvalue which realizes $\max_i |\lambda_i|$. It is easy to see that as long as $\lambda_j(\alpha)$ realizes $\max_i |\lambda_i(\alpha)|$ we can still decrease the spectral radius. And since, $|\lambda_i(\alpha)| = |\lambda_j(\alpha)| \Rightarrow \lambda_i(\alpha) = -\lambda_j(\alpha)$, we see that when $\exists i \neq j$ tel que $\lambda_i \neq \lambda_j$ et $|\lambda_i(\alpha)| = |\lambda_j(\alpha)|$. Since the monotonicities are identical, we can no longer minimize both $|\lambda_j(\alpha)|$ and $|\lambda_i(\alpha)|$. The optimum is thus reached. We notice that, since we have $\lambda_i(\alpha) < 1 \forall \alpha > 0$, the convergence of the algorithm is guaranteed at the optimum.

$\lambda_i(\alpha) = -\lambda_j(\alpha)$ gives us: $\alpha = \frac{1}{1 - \frac{\lambda_i + \lambda_j}{2}}$. Moreover, since the $\lambda_i(\alpha)$ cannot intersect, and λ_j corresponds to either the minimum or the maximum of the eigenvalues, the first $\lambda_i \neq \lambda_j$ which realizes $|\lambda_j(\alpha)| = |\lambda_i(\alpha)|$ is either the maximum (if λ_j is the minimum) or the minimum of the eigenvalues. The optimal relaxation, which we have seen ensures convergence, is therefore: $\alpha = \frac{1}{1 - \frac{\max_i \lambda_i + \min_i \lambda_i}{2}}$. As in the one-dimensional

case, the optimal relaxation we have identified requires knowledge of properties of (f) in the solution of the problem. It will therefore generally be necessary to use approximate methods. A very simple empirical procedure, which works whatever the dimension of the problem, often gives good results. It consists in dividing the relaxation by two each time the convergence criterion increases (be careful with its definition). It is possible, but not essential, to recalculate the iterations where the convergence criterion has increased.

f. Jacobi iterations

The Jacobi method is similar to the Gauss-Seidel method. The difference is that the result of the previous calculation is not used to improve the result of the next calculation. This is equivalent to iterating as follows (n is the iteration number):

$$\begin{cases} x_1^{(n+1)} = f_1(x_1^{(n)}, x_2^{(n)}) \\ x_2^{(n+1)} = f_2(x_1^{(n)}, x_2^{(n)}) \end{cases} \quad (19)$$

We can see that it is also a fixed-point method. The convergence is slower than for Gauss-Seidel, but the algorithm is parallelizable.

The Gauss-Seidel method is generally preferred to the Jacobi method. Thus we can say that the Gauss-Seidel method is easy to implement and often converges. In particular, the convergence takes place whatever the initialization on the coupling is sufficiently weak. However, the method

cannot converge if the couplings involved are too strong. Adding a relaxation can be a solution in this case, but the difficulty is then to parameterize it. Moreover, the linear convergence is all the faster that the functions studied are weakly coupled.

3.2 Newton’s method

a. Presentation of the method

This is another common method, which has many variants. In its original formalism, it is an intricate (and therefore firm) coupling method. Let us put the system to be solved as follows (F_1 and F_2 can be taken from the formulation (20) for example):

$$F(x) = \begin{pmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{pmatrix} = 0, x = (x_1, x_2) \quad (20)$$

It is recalled that, in general, x_i and $F_i(x_i, x_j)$ are vectors of \mathbb{R}^{d_i} . Note that $d = d_1 + d_2$ the number of components of x and define the Jacobian J_x of the system at x as follows (the F_i are here and for the following the components of F , and thus

$$J_x = \begin{pmatrix} \frac{\partial F_1}{\partial x_1}(x) & \dots & \frac{\partial F_1}{\partial x_d}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_d}{\partial x_1}(x) & \dots & \frac{\partial F_d}{\partial x_d}(x) \end{pmatrix} \quad (21)$$

Like the Gauss-Seidel method, the Newton method is iterative. Let us note here again with an exponent n the iteration index. From an initial data $x^{(0)}$ we iterate as follows: We solve in δ_x the linear system: $J_x^{(n)} \delta_x = -F(x^{(n)})$. And then we pose $x^{(n+1)} = x^{(n)} + \delta_x$. We generally stop when $\|\delta_x\|$ or $\|F(x^{(n)})\|$ becomes small enough. As before, we see that the method can only converge to a solution of the problem.

b. Study of the method

The method is based on a Taylor expansion to order 1 around the solution

$$x_{sol} : F(x_{sol}) = 0 \approx F(x^{(n)}) + J_x^{(n)}(x_{sol} - x^{(n)}) \quad (22)$$

Assuming F is sufficiently regular, let us use the Taylor-Lagrange inequality to quantify the error. As before, let us note $D^2 F_x$ the linear application of \mathbb{R}_d in $L(\mathbb{R}_d)$, such that $(D^2 F_x)ab = \sum_{i,j} \frac{\partial^2 F_k(x)}{\partial x_i \partial x_j} a_i b_j$. If there is a real M such that:

$$\|D^2 F_{x^{(n)} + t(x_{sol} - x^{(n)})}\| \leq M \forall t \in]0; 1[$$

then: $\| -F(x^{(n)}) + J_x^{(n)}(x_{sol} - x^{(n)}) \| \leq \frac{M \|x_{sol} - x^{(n)}\|^2}{2} \quad (23)$

Since $F(x_{sol}) = 0$. We introduce the vector η_n of norm less than 1, such that:

$$F(x^{(n)}) = J_x^{(n)}(x_{sol} - x^{(n)}) + \eta_n \frac{M \|x_{sol} - x^{(n)}\|^2}{2} \quad (24)$$

We can therefore write the norm of the error as follows, assuming the Jacobian is invertible:

$$\|\mathcal{E}^{(n+1)}\| = \|x^{(n+1)} - x_{sol}\| = \|x^{(n)} - J_x^{(n)-1} F(x^{(n)}) - x_{sol}\| \quad (25)$$

$$\|\mathcal{E}^{(n+1)}\| = \left\| x^{(n)} - J_{x^{(n)}}^{-1} (J_{x^{(n)}}(x_{sol} - x^{(n)}) + \eta_n \frac{M\|x_{sol} - x^{(n)}\|^2}{2}) - x_{sol} \right\| \quad (26)$$

$$\|\mathcal{E}^{(n+1)}\| = \|J_{x^{(n)}}^{-1} \eta_n\| \frac{M\|\mathcal{E}^{(n)}\|^2}{2} \leq \|J_{x^{(n)}}^{-1}\| \frac{M\|\mathcal{E}^{(n)}\|^2}{2}. \quad (27)$$

Let

$$K = \max_{x \in B(x_{sol}, \|x_{sol} - x^{(0)}\|)} \left(\frac{1}{2} \|J_x^{-1}\| \|D^2 F_x\| \right).$$

$B(x_{sol}, \|x_{sol} - x^{(0)}\|)$ is the ball of center x_{sol} and radius $\|x_{sol} - x^{(0)}\|$ convergence occurs if the following criterion is met:

$$K \|x^{(0)} - x_{sol}\| < 1. \quad (28)$$

Furthermore, it can be shown by recurrence that we have:

$$\|\mathcal{E}^{(n)}\| \leq \frac{(K \|x^{(0)} - x_{sol}\|)^{2^n}}{K}. \quad (29)$$

This shows us that we have a quadratic convergence, thus faster than the Gauss-Seidel one, if the initialization is good enough.

Newton's algorithm can only be defined correctly if we can define the Jacobian of F at all points. However, it happens that the phenomena we are studying introduce breaks in the slope. We are thinking of phase changes. In this case, the previous equations assure us that the convergence of the algorithm always takes place, for a sufficiently good initialization, whether we choose the partial derivatives on the right or on the left at the breaks in the slope.

c. Variant with relaxation

In the same way that we introduced a relaxation for Gauss-Seidel, it is possible to define one for Newton, which we will note $\alpha \in]0; 1]$ and which occurs as follows (δ_x is always calculated in the same way):

$$x^{(n+1)} = x^{(n)} + \alpha \delta_x. \quad (30)$$

It is easy to see that when $\alpha \neq 1$ one loses the quadratic convergence speed of Newton's algorithm. Relaxation cannot therefore be used to accelerate convergence, but it can allow convergence in certain situations where the initial algorithm would not have done so. The Jacobian is then mainly used to determine the direction in which the $x^{(n)}$ and α determines the size of the "step" taken. In principle, we can see that we are approaching a gradient descent type of algorithm, used. Let us insist on the fact that $\alpha = 1$ will always be optimal when approaching the solution for the Newton algorithm, which was not the case for Gauss-Seidel. When using a relaxation with Newton's algorithm, one must therefore have the possibility, at each iteration, to increase α . As with Gauss-Seidel, there is a simple empirical procedure for using relaxation with the Newton method. This involves adding an internal iteration level. Each Newton iteration starts with $\alpha = 1$. This iteration is then recalculated by dividing the relaxation by two each time, as long as the resulting residual is not smaller than that of the previous iteration.

d. Newton's algorithm does not correct

It is simply a Newtonian algorithm in which it is assumed that the solution of the system $J_{x^{(n)}} \delta_x = -F(x^{(n)})$ is not exact. Starting with an initial data $x^{(0)}$ we iterate as follows: We determine δ_x which verifies: $\|J_{x^{(n)}} \delta_x + F(x^{(n)})\| \leq \eta^{(n)} \|F(x^{(n)})\|$. And we pose $x^{(n+1)} = x^{(n)} + \alpha^{(n)} \delta_x$. $\eta^{(n)} \in [0; 1]$ fixes the required precision on the resolution of the linear system and $\alpha^{(n)} > 0$ corresponds to a relaxation as already introduced. As the resolution of the linear system is assumed to be inaccurate, the algorithm does not benefit from the quadratic convergence speed of the Newton and the introduction of the relaxation is less troublesome.

3.3 JFNK, Jacobian-free Newton-Krylov, method

a. Presentation of the method

In particular, it is the basis of the MOOSE coupling platform [19]. The JFNK is an implementation of Newton's method based on a Krylov method for solving linear systems, which is applied at each iteration of Newton to the system $J_{x^{(n)}} \delta x = -F(x^{(n)})$. Krylov's methods have the advantage of involving the system matrix only through matrix-vector products, which we are able to approximate by simple code calls. The JFNK method thus allows to define a sequential (codes are kept) but firm coupling technique.

b. Principle of Krylov methods for solving $Ax = b$

Let m be the dimension of the vector b . Krylov's methods consist in finding an approximation of the solution on a vector space K_n of reduced dimension $n \leq m$, which is constructed as follows: $K_n = Vect(r0; Ar0, \dots, A^{n-1} r0)$, where $r0 = b - Ax0$, with $x0$ an initial estimate of the solution of the linear system. In our case $x0$ is usually taken to be zero and we simply have $r0 = b$. To simplify the writings, we will work in the following directly with the variable b . Note that $(b, Ab, \dots, A^{n-1} b)$ does not necessarily form a family of free vectors, but that if this is not the case, the solution x can be generated by the family $(b, Ab, \dots, A^{n-2} b)$.

c. Use and interest of the JFNK method in the context of coupling

In a problem involving a Jacobian, one can use the following Taylor expansion to approximate the matrix-vector product:

$$J_u v = \frac{F(u+\epsilon v) - F(u)}{2\epsilon} + O(\epsilon). \quad (31)$$

An evaluation of F corresponds to a call to the codes we couple. The method can therefore be used without the need to compute or store the Jacobian of our coupled problem but requires frequent calls to the codes. We thus need one call of the codes per GMRES iteration, in addition to the necessary call per Newton iteration. We see that our approximation of the product $J_u v$ product introduces an error in $O(\epsilon)$. This error can be reduced by using a centered difference, but then we need two calls of the codes per iteration of GMRES.

$$J_u v = \frac{F(u+\epsilon v) - F(u-\epsilon v)}{2\epsilon} + O(\epsilon^2). \quad (32)$$

Let us insist on the fact that the JFNK is only a way to implement Newton's algorithm by circumventing the difficulties linked to the construction of the Jacobian. Newton's algorithm by circumventing the difficulties associated

with the construction of the Jacobian. The discussion about Newton's method applies here. The JFNK introduces new parameters which are the ε used to calculate the product J_v and the convergence criterion of the GMRES method. The former must be chosen small enough to make the Taylor expansion as fair as possible, but large enough to avoid numerical noise. The GMRES convergence criterion is a compromise between accuracy and calculation time.

4. Transient coupling methods

4.1. Method directly derived from stationary techniques

In the previous methods, the time variable does not intervene. We can then ask ourselves how to achieve a coupling in kinetics. To explain it, let us start by giving a general form of the coupling, as we did previously:

$$\begin{cases} \frac{\partial x_1}{\partial t} = G_1(x_1, x_2) \\ \frac{\partial x_2}{\partial t} = G_2(x_1, x_2) \end{cases} \quad (33)$$

Except for very special systems, after choosing a numerical scheme, we obtain resolutions of the type:

$$\begin{cases} x_1(t + \Delta t) = x_1(t) + h_1(x_2(t + \Delta t), \Delta t) \\ x_2(t + \Delta t) = x_2(t) + h_2(x_1(t + \Delta t), \Delta t) \end{cases} \quad (34)$$

Where the h_i are continuous functions such that $\lim_{\Delta t \rightarrow 0} h_i(x, \Delta t) = 0 \forall x$. By posing:

$$\begin{cases} F_1(u_1, u_2) = u_1 - x_1(t) - h_1(u_2, \Delta t) \\ F_2(u_1, u_2) = u_2 - x_2(t) - h_2(u_1, \Delta t) \end{cases} \quad (35)$$

We return to the form already studied:

$$\begin{cases} F_1(u_1, u_2) = 0 \\ F_2(u_1, u_2) = 0 \end{cases}$$

The methods already seen are therefore directly applicable, but we have an additional parameter that allows us to influence the strength of the coupling thus defined at each timestep: the time step itself. In particular, when using a Gauss-Seidel method, the convergence criterion ($\exists M < 1, \|J\| \leq M$) is automatically verified for Δt sufficiently small. Unfortunately, the time step necessary for Gauss-Seidel is sometimes too small to be exploitable (calculation of the transient too long), and we then turn to the other methods already presented. Let us note that there is a purely technical difficulty, but sometimes difficult to circumvent, for the implementation of evolving coupled schemes: the methods presented are iterative, it is thus necessary to be able to recalculate the same time step several times, without modifying the variables at the beginning of the time step (the $x_i(t)$ in our notation). However, the codes sometimes forget $x_i(t)$ during the calculation of $x_i(t + \Delta t)$.

4.2. Explicit coupling

To save calculation time when the couplings are weak, but also because of the technical difficulty we have just mentioned, it is often desired to calculate a transient coupled without iteration at each time step. To do this, we will look at how an error evolves during the calculation of a transient, whether or not the phenomenon being modelled arises from a coupling. Let

us write a general form for an evolution problem:

$$\frac{\partial x}{\partial t} = F(x).$$

It is assumed that a classical explicit scheme is used, which gives us:

$$x(t + \Delta t) = x(t) + \Delta t F(x(t)). \quad (36)$$

Let us note x_e the exact solution of the problem and suppose that at the beginning of the time step $x(t) = x_e(t) + \varepsilon(t)$. We will also note ε' the error introduced by the scheme itself (which is of order 1 in time) on the time step we are calculating:

$$x_e(t) + \Delta t F(x_e(t)) = x_e(t + \Delta t) + \varepsilon'(t + \Delta t). \quad (37)$$

With this we have :

$$\begin{aligned} x(t + \Delta t) &= x_e(t) + \varepsilon(t) + \Delta t F(x_e(t) + \varepsilon(t)) = x_e(t) + \varepsilon(t) + \Delta t (F(x_e(t)) + \\ &F(x_e(t) + \varepsilon(t)) - F(x_e(t))) = x_e(t + \Delta t) + \varepsilon'(t + \Delta t) + \varepsilon(t) + \Delta t (F(x_e(t) + \\ &+ \varepsilon(t)) - F(x_e(t))) \quad \text{Eq (38)} \end{aligned}$$

Thus we can write $\varepsilon(t + \Delta t) = f(\varepsilon(t))$ with :

$$f(u) = u + \Delta t (F(x_e(t) + u) - F(x_e(t))) + \varepsilon'(t + \Delta t) \quad (39)$$

As we move forward in time, the error committed evolves almost as if we were doing fixed point iterations with the function f thus defined. However, the equivalence is not perfect since x_e and ε' evolve with time. To understand what is going on, let us suppose that these quantities are constant (which amounts to saying that we verify the stability of a stationary state), and that we are in one dimension. In this case, convergence occurs if $|f'(u)| < 1$, so it si $F'(u) \varepsilon \Big|_{\Delta t}^{-2}, 0$. The upper bound, $F'(u) < 0$, means that the system under study must itself be (statically) stable. As for the lower bound, we notice that by decreasing the time step it can be arbitrarily small. Thus, with a sufficiently small-time step, the error committed by an explicit coupling scheme can remain bounded. Nevertheless, the error committed is in $O(\Delta t)$. The explicit scheme is therefore accurate to first order in time. The formalism used is valid when we evolve a coupled problem in time, and we do not converge the coupling at each time step. Thus, the way in which the coupling is performed can reduce the accuracy of the solution to first order in time, even though the solvers themselves have better accuracy. This point is particularly important. By saving the work of thinking about the coupling scheme, one risks losing the benefit of all the work invested in the codes themselves.

5. Synthesis

The Gauss-Seidel method is very easy to implement and is the first method to be proposed in the genesis of coupling. This does not mean that the Newton algorithm is superior in all respects to the Gauss-Seidel algorithm. The main advantage of Newton's method is its quadratic rather than linear convergence speed. It is, therefore, to be preferred if a good coupling is desired. If one is simply interested in the solution of the coupled problem, the Gauss-Seidel algorithm is generally sufficient. The use of a relaxation, which can be determined empirically, is sufficient to make it very robust. Only those cases, which are actually quite rare, where the algorithm diverges without oscillating cannot be stabilized with a relaxation.

Newton's algorithm, on the other hand, can be used theoretically on any case, but on condition that the initialization is good enough. Even if the use of a relaxation is likely to greatly improve its behavior, the Newton algorithm is sometimes less robust than the Gauss-Seidel algorithm, because it is more sensitive to noise. The comparative table summarizes our analysis by presenting the advantages and disadvantages of each method presented above. It shows that Newton's method is the most appropriate for the coupling of disciplines in a mechatronic system.

Table1 - Table of methods with their advantages and disadvantages

Methods	Advantages	Disadvantages	Mechatronic systems
Direct methods	Suitable for medium and large sized systems.	For large systems, it requires a lot of memory size.	Not adequate because there is no stability.
Gauss Seidel methods	Suitable for linear systems and linear convergence.	Choice of the stopping condition.	Not adequate because mechatronic systems are linear.
Newton's methods	Appropriate for nonlinear systems and quadratically convergent.	Choice of initial data.	Adequate.
JFNK method	Suitable for nonlinear systems, More memorable, includes more parameters which makes calculations easier.	Choice of initial data.	Adequate but with too many parameters.
Explicit pairing method	Time saving.	Reduced accuracy due to partitioned convergence.	Not Adequate because poor precision emerges.

6. Comparative table

At the end of this coupling analysis, we have noticed that all systems, whatever the size, can be coupled by coupling methods. Depending on the problem to be solved we have to choose the adequate method. Direct methods if the system is of average size but also of large size while envisaging much memory size. Hence the interest in using iterative methods, especially Gauss Seidel and Jacobi. Nevertheless, they are used for linear systems. However, mechatronic systems are non-linear. This justifies the choice of Newton's method. Therefore, for non-linear systems, we have to use the Newtonian method while taking care to choose the initial data carefully.

7. Conclusion

At the end of this coupling analysis, we have noticed that all systems, whatever the size, can be coupled by coupling methods. Depending on the problem to be solved we have to choose the adequate method. That is to say direct methods if the system is of average size but also of large size while envisaging much memory size. Hence the interest in using iterative methods, especially Gauss Seidel and Jacobi. Nevertheless, they are used for linear systems. However, mechatronic systems are non-linear. This justifies the choice of Newton's method. Therefore, for non-linear systems, we have to use the Newtonian method while taking care to choose the initial data carefully.

Authors' contribution

All authors contributed equally to the preparation of this article.

Declaration of competing interest

The authors declare no conflicts of interest.

Funding source

This study didn't receive any specific funds.

Acknowledgements

As corresponding author of this paper, I am the scientific coordinator of the Laboratory of Mechanics, Materials, and Photonics. It is in this laboratory that this research was conducted. I would like to thank all my colleagues for their frank collaborations.

REFERENCES

- [1] David et al 2013 Special Issue on Multiphysics simulations: Challenges and opportunities, Volume 27 Number 1 February 2013 The International Journal of High Performance Computing Applications 27(1) 4–83. The. DOI: 10.1177/1094342012468181 hpc.sagepub.com
- [2] Miladi C., R. Plateauxb, J.-Y. Choleyb, C. Karraa, A. Riviereb M. Haddar : New topological approach for the modelling of mecatronic systems: application for piezoelectric structures, European Journal of Computational Mechanics, 2013 Vol. 22, Nos. 2–4, 209–227, <http://dx.doi.org/10.1080/17797179.2013.820896>
- [3] K. C. Park and Carlos A. Felippa : A Variational Principle for the Formulation of Partitioned Structural Systems, Department of Aerospace Engineering Sciences and Center for Aerospace Structures University of Colorado, Campus Box 429 Boulder, CO 80309, USA, 1999.
- [4] Cresta P: Décomposition de domaine et stratégies de relocalisation non-linéaire pour la simulation de grandes structures raidies avec flambage local, thèse 2008 <https://tel.archives-ouvertes.fr/tel-00363656/document>
- [5] Lionel G. : Approche globale / locale non-intrusive : application aux structures avec plasticité localisée. Mécanique [physics.med-ph]. École normale supérieure de Cachan - ENS Cachan, 2009. Français. <https://tel.archives-ouvertes.fr/tel-00449687/document>
- [6] E. Wyarta, M. Dufloata, D. Coulomb, P. Martinya, T. Pardoenc, J.-F. Remacle, F. Lania : Substructuring FE-XFE approaches applied to three-dimensional crack propagation, Journal of Computational and Applied Mathematics. doi:10.1016/j.cam.2006.03.066
- [7] E.Wyart D.Coulon T.Pardoenc F.Remacle F.Lani : Application of the substructured finite element/extended finite element method (S-FE/XFE) to the analysis of cracks in aircraft thin walled structures, Engineering Fracture Mechanics, Volume 76, Issue 1, January 2009, Pages 44–58 <https://doi.org/10.1016/j.engfracmech.2008.04.025>
- [8] Mehmet A. Akgün, John H. Garcelon, Raphael T. Haftka : Fast exact linear and non-linear structural reanalysis and the Sherman–Morrison–Woodbury formulas, 26 January 2001 <https://doi.org/10.1002/nme.87>
- [9] Mickaël D. Passieux J. Michel S. Stéphane G : Non-intrusive Coupling: Recent Advances and Scalable Nonlinear Domain Decomposition, Archives of Computational Methods in Engineering, 2014. DOI 10.1007/s11831-014-9132-x
- [10] Bernardi C. Nicolas F. Robert G. Owens : An error indicator for mortar element solutions to the Stokes problem, IMA Journal of Numerical Analysis (Volume: 21, Issue: 4, Oct. 2001), Page(s): 857 – 886. DOI: 10.1093/imanum/21.4.857
- [11] Belgacem F : The Mortar finite element method with Lagrange multipliers Laboratoire de Mathématiques pour l'Industrie et la Physique, Unité Mixte de Recherche CNRS–UPS–INSAT–UT1 (UMR 5640), Université Paul Sabatier, 118 route de Narbonne, F-31062 Toulouse Cedex 04, France, FR, volume 84, pages173–197 (1999)
- [12] Bernardi C, Yvon M. Francesca R. : Basics and some applications of the mortar element method, Volume28, Issue2, 08 May 2014, Pages 97–123. <https://doi.org/10.1002/gamm.201490020>
- [13] Anita H. Peter H. : An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Computer Methods in Applied

- Mechanics and Engineering Volume 191, Issues 47–48, 22 November 2002, Pages 5537–5552, [https://doi.org/10.1016/S0045-7825\(02\)00524-8](https://doi.org/10.1016/S0045-7825(02)00524-8)
- [14] Becker R. Peter H. Rolf S. : A finite element method for domain decomposition with non-matching grids, *ESAIM: Mathematical Modelling and Numerical Analysis*, Volume 37, Issue 2, March 2003, pp. 209 – 225, DOI: <https://doi.org/10.1051/m2an:2003023>
- [15] Nguyen V. Pierre K. Marco B. Stéphane P. Elvio B. : Nitsche’s method for two and three dimensional NURBS patch coupling, *Computation Mechanics*, Volume 53, pages 1163–1182 (2014) <https://link.springer.com/article/10.1007/s00466-013-0955-3>
- [16] Paul T. Hachmi B. Nadia E. Tinsley O. Serge P : On the application of the Arlequin method to the coupling of particle and continuum models, *Computational Mechanics* volume 42, pages 511–530 (2008) <https://link.springer.com/article/10.1007/s00466-008-0291-1>
- [17] Picasso M. Jacques R. Vittoria R. : Multiscale algorithm with patches of finite elements, *Research Article, Communications in Numerical Methods in Engineering*, 31 May 2007 <https://doi.org/10.1002/cnm.1019>
- [18] Alexei L. and Pironneau O : Numerical zoom for advection diffusion problems with localized multiscales, *Mathematics, Applied*, October 2010 <https://doi.org/10.1002/num.20642> <https://onlinelibrary.wiley.com/journal/10982426>
- [19] Derek G. Chris N. Glen H. Damien L : MOOSE: A parallel computational framework for coupled systems of nonlinear equations, *Nuclear Engineering and Design*, Volume 239, Issue 10, October 2009, Pages 1768–1778 <https://doi.org/10.1016/j.nucengdes.2009.05.021>
- [20] Cyril P. : Couplages multi-physiques : évaluation des impacts méthodologiques lors de simulations de couplages neutronique/thermique/mécanique. <https://www.theses.fr/2016SACLX007>
- [21] Shreyansh J, Victoire M. , H. N. S. Narayana, Simon B, Joseph D, Victor C, Tianchi C., L. Heuzé, Philippe M, René-Marc M, J. Kabla, Teck L & Benoit L. : The role of single-cell mechanical behaviour and polarity in driving collective cell migration. <https://www.nature.com/articles/s41567-020-0875-z.pdf>
- [22] Mao J : Implementation of the analytic nodal method in the ndf code and applications to candu reactors, *Ecole polytechnique de Montréal*, 2000 <https://publications.polymtl.ca/8798/>
- [23] Thomas Heuzé : Couplages Multiphysiques, 2020 <https://hal.archives-ouvertes.fr/hal-02448292>