



HAL
open science

On the implementation in Abaqus of the global–local iterative coupling and acceleration techniques

Omar Bettinotti, Stéphane Guinard, Eric Véron, Pierre Gosselet

► To cite this version:

Omar Bettinotti, Stéphane Guinard, Eric Véron, Pierre Gosselet. On the implementation in Abaqus of the global–local iterative coupling and acceleration techniques. *Finite Elements in Analysis and Design*, 2024, 236, pp.104152. 10.1016/j.finel.2024.104152 . hal-04522597

HAL Id: hal-04522597

<https://hal.science/hal-04522597>

Submitted on 26 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

On the Implementation in Abaqus of the Global-Local Iterative Coupling and Acceleration Techniques

Omar Bettinotti¹, Stéphane Guinard², Eric Véron¹ and Pierre Gosselet³

¹Dassault Systèmes Simulia Corp., Johnston, RI (USA)

²IRT Saint Exupéry, Toulouse, France

³Univ. Lille, CNRS, Centrale Lille, UMR 9013 - LaMcube, France

March 26, 2024

Abstract

This paper presents results and convergence study of the Global-Local Iterative Coupling through the implementation in the commercial software Abaqus making use of the co-simulation engine. A hierarchical modeling and simulation approach is often required to alleviate modeling burdens. Particular focus has been devoted here on convergence acceleration and performance optimization. Two applications in statics with nonlinear material behavior and geometrically nonlinear formulation are considered here: first a holed curved plate under traction with elastic-plastic material, then a pre-stressed bolted joint connecting two plates between each other and subjected to traction load. Three different convergence acceleration techniques are compared in terms of convergence performance and accuracy. An inexact solver strategy is proposed to improve computing time performance. The results show promising results for the coupling technology and constitute a step forward in the availability of non-intrusive multi-scale modeling capabilities for complex structures and assemblies.

1 Introduction

Due to the complexity of aeronautical structure assemblies, engineers are used to building models across multiple levels of abstraction. At the highest level, a comprehensive model of the full aircraft is built from mid-surfacing and de-featuring the exact shapes of the assembly, with imposing a coarse finite elements mesh size, and simplifying assembly connections and interactions. Material non-linearities are usually not considered at this level. The lower levels of abstractions are locally and hierarchically built, considering the exact geometry at the lowest level where a more refined finite elements mesh and material non-linearities are usually considered. As the lowest levels are local, their simulations are based on internal forces or boundary conditions coming from their highest level.

A main challenge today for aeronautic industry is to foster more agile connections between the different abstraction levels: by setting up accelerated paths from top to bottom levels, upwards and backwards (accelerated design); and also by refurbishing data flows between tests and simulations at different levels in the “test pyramid” (simulation aided certification).

As a preliminary step towards such high level objectives, the sub-modeling technique is often employed. It consists in applying the results of the global coarse simulation as boundary conditions in the local more refined simulation. As the sub-modeling technique

is sequential (global-to-local), an error is introduced and quantified as the difference between the integrals of the local reaction forces and global interface forces, where the local model cuts the global one. This error is due to different modeling assumptions in the global and local models (constitutive law, connection modeling, mesh refinement...). A more accurate coupling technique would require the global and local models to exchange data in both directions, paying the price of a higher computational cost, as simulations run concurrently.

Multiple strategies exist to couple global and local simulations together. The most straightforward strategy consists in directly installing the local finite elements model into the global model with bonding interactions, possibly from solid to shell, and then running a full monolithic simulation. An alternative strategy consists in using domain decomposition techniques, pioneered in [39, 29] as Schwarz alternating method, reformulated exchanging Neumann conditions in [40] or imposing the interface constraints by Lagrange multipliers in [20, 19]. A more comprehensive summary of domain decomposition techniques is presented in [30]. A chief advantage of domain decomposition techniques is introduced in [32, 11, 33, 21] for dynamic applications, as different time integration schemes or time increments can be applied to each sub-domain (sub-cycling), without compromising stability. An interesting generalization to a multi-physic application is presented in [12].

However, the chief issue with the aforementioned structural multi-scale techniques, the monolithic and the domain decomposition ones, is that the models abstractions need to be generated and adjusted for each simulation, aggravating the proliferation of models and increasing modeling complexity. A less intrusive approach from the (non-negligible) modeling point of view is to keep the global coarse models unchanged, as it is the case of the sequential sub-modeling technique, and concurrently iterate between the global and local simulations. This is what we will refer to as the Global-Local Iterative Coupling (GLIC) in the following. After the pioneering works of [46], the formulation of the GLIC applied to nonlinear statics was revisited in [27, 26], reviewed in [43], studied in the context of shell-to-solid coupling in [34, 35] and [2] and of uncertainty quantification in [42]. In similar manner to the sub-cycling in domain decomposition, the GLIC was also extended to explicit dynamics in [7, 8] and applied to a high-velocity impact causing composite delamination in [9]. More recent works have proposed the application of the Generalized Finite Element Method to the local model in [38, 37] and the possibility to evolve the size and location of the local patch in order to capture the propagation of fracture using non-local damage phase-field equations in [3, 4, 41, 28].

Although a global-local approach relieves from highly time consuming modeling tasks (that are not scalable with computing resources), a chief drawback for the GLIC is the computational cost introduced with global-local iterations. The computing times of the GLIC are approximately equal to the total computing time of the direct simulation multiplied by the average number of global-local iterations during the simulation. As summarized in [31, 10, 5], acceleration techniques can be employed as mitigation of increased computing costs and to improve robustness, similarly to fluid-structure interaction techniques as, for instance, the Anderson acceleration scheme [6], more recently revisited in [13].

With these works, the Authors explore and test the GLIC as implemented in Abaqus through co-simulation, which is a special technique that allows users to run simulations concurrently, exchanging quantity values through a finite element interface. In the case of the GLIC, co-simulation is run to couple the global simulation to the local one and to exchange displacements and reaction forces. The fields are exchanged through a user-defined surface in common between the two finite element models. A configuration file needs to be written to provide instructions to the Abaqus co-simulation engine on how to coordinate the two simulations.

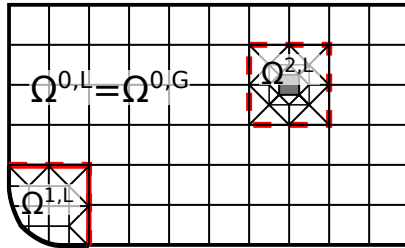


Figure 1: Reference problem with two local patches and a non-empty complement zone.

Particular focus has been devoted on the study of the convergence and on the investigation of inexact convergence, as in classical multi-grid schemes as for instance [22, 23, 24, 25], in order to reduce even further the computation expenses. After this Introduction, Section 2 is devoted to the formulation of the iterative scheme and the presentation of accelerators, Section 3 describes some challenging use cases and Section 4 proposes concluding remarks and future works.

2 Formulation and study of the Global-Local Iterative Coupling

We consider a (quasi)-static mechanical problem set on a domain Ω . We assume that a first *global* (superscript G) finite element modeling is available, capable of correctly grasping the long-range trends in the structure. The global model need not be linear, in which case an incremental study is conducted, and for simplicity reason we only consider one load increment in this paper, multiple increments with time grids adapted to the models have been studied in [10]. The system to be solved can be written as:

$$\text{Find } \mathbf{u} \text{ so that } \mathbf{f}_{int}^G(\mathbf{u}^G) + \mathbf{f}_{ext}^G = 0, \quad (1)$$

where \mathbf{f}_{ext} stands for the (generalized) external forces, \mathbf{f}_{int} is the vector of internal forces and \mathbf{u} is the vector of unknown displacements (Dirichlet conditions are assumed to have been eliminated). Of course, in the case of a linear problem, the internal forces take the form of a linear application:

$$\mathbf{f}_{int}^G(\mathbf{u}^G) = -\mathbf{K}^G \mathbf{u}^G \quad (2)$$

where \mathbf{K}^G is the sparse symmetric definite positive stiffness matrix.

We suppose that this global model is insufficient in some *local* areas: the geometry, the heterogeneity or the material law are too simplified, or the mesh is too coarse. . . . These regions $(\Omega^{s,G})_{s>0}$ are assumed to be non-overlapping sets of connected elements of the global mesh. For simplicity we consider that the meshes are matching at the interface, that is to say that there is a one-to-one correspondence between degrees of freedom. A possibility to implement this hypothesis is to have a copy of the global mesh of the interface on each patch and to use a tie constraint with the refined mesh.

On these regions, there exists a *local* modeling (superscript L) with adapted geometry, mesh, elements, material law. . . , nevertheless we suppose that the interface is preserved in the modelings: $\partial\Omega^{s,L} \cap \Omega = \partial\Omega^{s,G} \cap \Omega$. The zone of the global mesh which is not covered by patches, sometime referred to as complement zone, is written with index 0: $\Omega^{0,G} = \Omega^G \setminus \cup_{s>0} \Omega^{s,G}$. Note that this zone may not exist if the global model is fully covered by patches. Our reference is the assembly of all the local models and of the complement zone: $\Omega^{0,G} \cup (\cup_{s>0} \Omega^{s,L})$, see Figure 1.

We note $\Gamma^s = \partial\Omega^s \cap \Omega$ the interface of the patches, and $\Gamma^G = \cup_{s>0} \Gamma^s$ the global interface. We introduce the trace operators $(\mathbf{T}^G, \mathbf{T}^{s,L})$ that extract the interface degrees

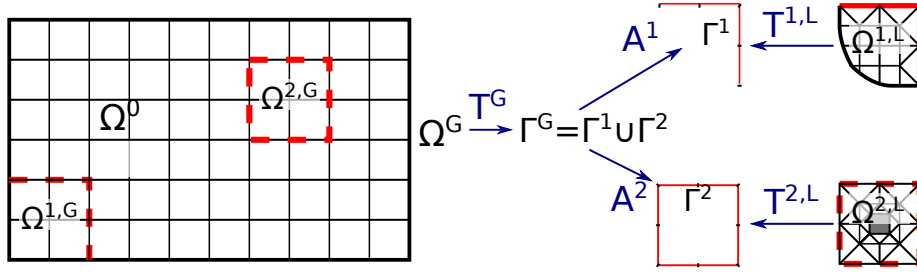


Figure 2: Global problem and topological operators (traces and assembly – Use the transpose to follow the opposite direction).

of freedom. We also introduce the injection operators \mathbf{A}^s which position Γ^s with Γ^G , see Figure 2.

We consider Dirichlet problems on the patches with the imposed displacement inherited from the global problem. Let Subscript i denote *internal* degrees of freedom of the patches and Subscript Γ *interface* degrees of freedom.

$$\text{Dirichlet condition: } \mathbf{u}_\Gamma^{s,L} = \mathbf{A}^{sT} \mathbf{T}^G \mathbf{u}^G$$

$$\text{Internal nonlinear problem: Find } \mathbf{u}_i^{s,L} \text{ such that } \mathbf{f}_{int,i}^{s,L} \begin{pmatrix} \mathbf{u}_i^{s,L} \\ \mathbf{u}_\Gamma^{s,L} \end{pmatrix} + \mathbf{f}_{ext,i}^{s,L} = 0 \quad (3)$$

$$\text{Interface post-processing: } \boldsymbol{\lambda}^{s,L} = -\mathbf{f}_{int,b}^{s,L} \begin{pmatrix} \mathbf{u}_i^{s,L} \\ \mathbf{u}_\Gamma^{s,L} \end{pmatrix} - \mathbf{f}_{ext,b}^{s,L}$$

$\boldsymbol{\lambda}^{s,L}$ is the vector of nodal reaction associated with the imposed displacement $\mathbf{A}^{sT} \mathbf{T}^G \mathbf{u}^G$, it can also be obtained as the Lagrange multiplier which imposes the given displacement.

At this point, we have carried out a classical sub-modeling approach. This approach is known to potentially brings lots of errors because the effect of the local models is not brought back on the global model, which in particular prevents interactions between patches.

The residual \mathbf{r}_Γ can be measured as the lack of balance between the refined patches and the subdomain 0:

$$\mathbf{r}_\Gamma = - \left(\mathbf{A}^0 \boldsymbol{\lambda}^{0,G} + \sum_{s=1}^N \mathbf{A}^s \boldsymbol{\lambda}^{s,L} \right). \quad (4)$$

The reaction on subdomain 0 is defined as $\boldsymbol{\lambda}^{0,G} = -\mathbf{f}_{int,b}^{0,G} \begin{pmatrix} \mathbf{u}_i^{0,G} \\ \mathbf{u}_\Gamma^{0,G} \end{pmatrix} - \mathbf{f}_{ext,b}^{0,G}$. A formula which avoids the management of Subdomain 0 will be given soon.

The global/local coupling simply consists in doing a Richardson iteration to try to reduce the residual. To do so, we introduce the (initially null) corrective load $\mathbf{T}^{GT} \mathbf{p}_\Gamma^G$ applied to the global model. The update of the load and its application in the corrected global problem can be written as:

$$\begin{aligned} \mathbf{p}_\Gamma^G &\leftarrow \mathbf{p}_\Gamma^G + \omega \mathbf{r}_\Gamma, \\ \mathbf{f}_{int}^G(\mathbf{u}^G) + \mathbf{f}_{ext}^G + \mathbf{T}^{GT} \mathbf{p}_\Gamma^G &= 0. \end{aligned} \quad (5)$$

where $\omega > 0$ is a relaxation parameter.

Remark 1. Other implementations of the global-local coupling, like the one in [16], prefer to use \mathbf{u}^G as the main unknown to be updated with relaxation at each iteration. When the global problem is linear, \mathbf{u}^G depends on \mathbf{p}_Γ in an affine manner and the two approaches are strictly equivalent. The approaches may differ slightly if the global problem is nonlinear,

and we prefer our approach where the unknown has the same physical dimension (and vector size) as the “natural” residual which is the lack of balance between the models. Like in any other domain decomposition method, it highlights the crucial role played by the interface.

The existence of a non-empty interval of values of the relaxation coefficient for which the algorithm converges is proved for an elastic global model coupled with monotonic patches (e.g. visco-plasticity with positive hardening). In practice the convergence is observed in more general cases, like with an elasto-visco-plastic global model or softening (but stable) patches. Moreover, all fixed-point accelerators can be used to improve the convergence.

Once the iteration has converged, we recover the *reference* model constituted by the complement zone $\Omega^{0,G}$ and the patches ($\Omega^{s,L}$) where the displacement is continuous at the interface and the nodal reaction are balanced, that is to say all interactions between the complement zone and the patches have been counted for.

Algorithm 1 recalls the classical global-local coupling method.

Algorithm 1: Non-invasive stationary iterations with relaxation

Arbitrary initialization $\mathbf{p}_{\Gamma,0}$

for $j \in [0, \dots, j_{\max}]$ **do**

Global solve: Find \mathbf{u}_j^G such that:

$$\mathbf{f}_{int}^G(\mathbf{u}_j^G) + \mathbf{f}_{ext}^G + \mathbf{T}^{G^T} \mathbf{p}_{\Gamma,j}^G = 0$$

Global post-process of $\boldsymbol{\lambda}_j^{0,G}$ (if it exists):

$$\boldsymbol{\lambda}_j^{0,G} = -\mathbf{T}^{0,G} \left(\mathbf{f}_{int}^{0,G}(\mathbf{u}^{0,G}) + \mathbf{f}_{ext}^{0,G} \right)$$

Local solves: $\forall s > 0$, find $\mathbf{u}_j^{s,L}, \boldsymbol{\lambda}_j^{s,L}$ such that:

$$\begin{cases} \mathbf{T}^{s,L} \mathbf{u}_j^{s,L} = \mathbf{A}^{s^T} \mathbf{T}^G \mathbf{u}_j^G \\ \mathbf{f}_{int}^{s,L}(\mathbf{u}_j^{s,L}) + \mathbf{f}_{ext}^{s,L} + \mathbf{T}^{s,L^T} \boldsymbol{\lambda}_j^{s,L} = 0 \end{cases}$$

Assembly of residual: $\mathbf{r}_{\Gamma,j} = - \left(\mathbf{A}^0 \boldsymbol{\lambda}_j^{0,G} + \sum_{s=1}^N \mathbf{A}^s \boldsymbol{\lambda}_j^{s,L} \right)$

if $\|\mathbf{r}_{\Gamma,j}\|$ *small enough* **then** Break

Update: $\mathbf{p}_{\Gamma,j+1} = \mathbf{p}_{\Gamma,j} + \omega \mathbf{r}_{\Gamma,j}$

end

Remark 2. The post-processing of $\boldsymbol{\lambda}^{0,G}$ is sometimes difficult to implement in commercial software. A workaround consists in extracting the global version of the patches, solve the same Dirichlet problem as their refined counterparts and obtain the global nodal reactions on the patches $\boldsymbol{\lambda}^{s,G}$. Since the extra load \mathbf{p}_{Γ} introduces a stress discontinuity on the global model, we have:

$$\sum_{s \geq 0} \mathbf{A}^s \boldsymbol{\lambda}^{s,G} = \mathbf{p}_{\Gamma}, \quad (6)$$

and thus

$$\begin{aligned}\mathbf{A}^0 \boldsymbol{\lambda}^{0,G} &= \mathbf{p}_\Gamma - \sum_{s>0} \mathbf{A}^s \boldsymbol{\lambda}^{s,G} \\ \mathbf{p}_\Gamma &\leftarrow (1 - \omega) \mathbf{p}_\Gamma + \omega \sum_{s>0} \mathbf{A}^s (\boldsymbol{\lambda}^{s,G} - \boldsymbol{\lambda}^{s,L})\end{aligned}\tag{7}$$

2.1 Use of inexact solvers

In the case of nonlinear problems, the classical solution strategy is to use a Newton-Raphson or a quasi Newton iterative procedure. In that case, a stopping criterion must be used. Typically, the global solver can be written as: Find \mathbf{u}^G such that:

$$\mathbf{f}_{int}^G(\mathbf{u}^G) + \mathbf{f}_{ext}^G + \mathbf{T}^{G^T} \mathbf{p}_\Gamma^G = \boldsymbol{\rho}^G, \quad \text{with } \|\boldsymbol{\rho}^G\| < \varepsilon_r^G.\tag{8}$$

Another possible control is the use the length of the last correction to the unknown $\|\delta \mathbf{u}^G\| < \varepsilon_\delta^G$. ($\varepsilon_r^G, \varepsilon_\delta^G$) are the convergence thresholds (in term of residual and in term of variation). Those thresholds are often written in a relative way by introducing some normalization (e.g. the norm of the external load $\|\mathbf{f}_{ext}^G\|$). A classical question [15, 17] is to try to tune the values of the thresholds in agreement with current outer loop residual, in our case $\|\mathbf{r}_\Gamma\|$, in order to avoid oversolving while preserving the good convergence rate. In this work, the authors make use of the Abaqus convergence controls [1], which correspond to relative convergence criteria $\varepsilon_r^G = 1$ (the default value being 0.005) and $\varepsilon_\delta^G = 1$ (the default value being 0.01), where the normalizing factors are respectively the average flux norm and the norm of solution increment.

The fundamental equation to drive the thresholds, is the evaluation of the residual:

$$\mathbf{r}_{\Gamma,j} = - \left(\mathbf{A}^0 \boldsymbol{\lambda}_{j-1}^{0,G} + \sum_{s=1}^N \mathbf{A}^s \boldsymbol{\lambda}_j^{s,L} \right) + \left(\mathbf{A}^0 \boldsymbol{\rho}_{\Gamma,j}^G + \sum_{s>0} \mathbf{A}^{s,L} \boldsymbol{\rho}_{\Gamma,j}^{s,L} \right)\tag{9}$$

In order for the global-local residual to be correctly evaluated, it is important to ensure that the residuals of the global and local solves remain negligible in comparison. If we use the notation Δ for the variation of quantities from one global-local iteration to another, we have:

$$\mathbf{p}_{\Gamma,j+1} = \mathbf{p}_{\Gamma,j} - \omega \left(\mathbf{A}^0 \Delta_j \boldsymbol{\lambda}^{0,G} + \sum_{s=1}^N \mathbf{A}^s \Delta_j \boldsymbol{\lambda}^{s,L} \right) + \omega \left(\mathbf{A}^0 \boldsymbol{\rho}_{\Gamma,j}^G + \sum_{s>0} \mathbf{A}^{s,L} \boldsymbol{\rho}_{\Gamma,j}^{s,L} \right)\tag{10}$$

A good control appears to be $\|\boldsymbol{\rho}_{\Gamma,j}^G\| \leq \alpha \|\Delta_j \boldsymbol{\lambda}^{0,G}\|$, with $\alpha \ll 1$, and the same relation for the patches. A simpler control is to use $\|\boldsymbol{\rho}_{\Gamma,j}^G\| \leq \alpha \|\mathbf{r}_{\Gamma,j-1}\|$, and the same relation for the patches.

2.2 Accelerators

The Richardson iteration of Algorithm 1 is the simplest example of fixed-point methods. Its convergence can be slow and it may even lack robustness, failing to converge on too stiff problems. This section investigates the use of 3 standard accelerators to overcome these issues.

For simpler presentation, we rephrase the coupling in terms of abstract operators:

$$\begin{aligned}\mathbf{u}_\Gamma^G &= \mathcal{S}^{G^{-1}}(\mathbf{p}_\Gamma) && \text{obtained from eq. (5)} \\ \mathbf{r}_\Gamma &= \mathcal{S}^L(\mathbf{u}_\Gamma^G) && \text{obtained from eq. (3) and (4)}\end{aligned}\tag{11}$$

The standard fixed-point system relies on the sequential Gauss-Seidel execution of solvers:

$$\left(\mathcal{S}^L \circ \mathcal{S}^{G^{-1}}\right)(\mathbf{p}_\Gamma) = \mathbf{p}_\Gamma \quad (12)$$

\mathcal{S}^G and \mathcal{S}^L refer to the computation of one global time increment and one local time increment respectively. The notations are further simplified by introducing the operator $H = \mathcal{S}^L \circ \mathcal{S}^{G^{-1}}$ and omitting the index Γ in the unknown \mathbf{p} :

$$H : \mathbb{R}^n \rightarrow \mathbb{R}^n, H(\mathbf{p}) = \mathbf{p} \quad (13)$$

In the reminder of this section, the lower j denotes the iteration index to solve the fixed-point equation. $\tilde{\mathbf{p}}_j$ denotes the j -th Picard iterate $\tilde{\mathbf{p}}_j = H(\mathbf{p}_j)$. For some problems, such as Fluid-Structure Interaction with moderate/large density ratios or structural multi-scale problems, a moderate to strong physical coupling bonds the subdomains. More elaborated algorithms such as the Aitken's relaxation or Quasi-Newton algorithms [14, 45], referred as accelerators, must be employed in order to get a stable solution and good convergence properties. The cost associated with this additional acceleration step is negligible compared to the Picard iteration itself, especially for cases with surface coupling. Once the Picard iteration is computed, a correction is applied using accelerators:

$$\mathbf{p}_j \xrightarrow{\text{Picard}} \tilde{\mathbf{p}}_j \xrightarrow{\text{Acc.}} \mathbf{p}_{j+1} \quad (14)$$

The classical residual can then be written as $\mathbf{r}_j = \tilde{\mathbf{p}}_j - \mathbf{p}_j$.

2.2.1 Aitken's relaxation

The simplest accelerators are based on relaxation techniques:

$$\mathbf{p}_{j+1} = \mathbf{p}_j (1 - \omega_j) + \tilde{\mathbf{p}}_j \omega_j \quad (15)$$

A relaxation method with constant parameters might be sufficient for certain problems involving weak coupling. However, Aitken's relaxation is often used to improve the efficiency with a very limited additional computation cost. The Aitken's dynamic relaxation factor is provided by:

$$\omega_j = \omega_{j-1} \frac{\mathbf{r}_{j-1}^T (\mathbf{r}_j - \mathbf{r}_{j-1})}{\|\mathbf{r}_j - \mathbf{r}_{j-1}\|_2^2} \quad (16)$$

2.2.2 Multi-secant methods

In [18], Fang and Saad present multi-secant methods to solve the nonlinear problem $\mathbf{r}(\mathbf{p}) = H(\mathbf{p}) - \mathbf{p} = \mathbf{0}$. The fixed-point problem is formally inverted in order to expose Picard iterations outputs:

$$\mathbf{r}(\mathbf{p}) = H(\mathbf{p}) - \mathbf{p} \approx \mathbf{0} \Rightarrow \tilde{\mathbf{r}}(\tilde{\mathbf{p}}) = \tilde{\mathbf{p}} - H^{-1}(\tilde{\mathbf{p}}) \approx \mathbf{0} \quad (17)$$

Introducing $\Delta\tilde{\mathbf{p}}_j = \tilde{\mathbf{p}}_{j+1} - \tilde{\mathbf{p}}_j$ and the Jacobian $J_{\tilde{\mathbf{r}}}(\tilde{\mathbf{p}}_j)$, the transition from the Picard step to the subsequent iterate in the Newton-Raphson update is articulated as follows:

$$J_{\tilde{\mathbf{r}}}(\tilde{\mathbf{p}}_j) \Delta\tilde{\mathbf{p}}_j = -\tilde{\mathbf{r}}_j(\tilde{\mathbf{p}}_j) \quad (18)$$

$$\mathbf{p}_{j+1} = \tilde{\mathbf{p}}_j + \Delta\tilde{\mathbf{p}}_j \quad (19)$$

Approximate of the interface Jacobian $J_{\tilde{\mathbf{r}}}$ is computed using input-output information:

$$\begin{aligned}\mathbf{W}_j &= \left[\Delta \tilde{\mathbf{p}}_0, \Delta \tilde{\mathbf{p}}_1, \dots, \Delta \tilde{\mathbf{p}}_{j-1} \right], \quad \text{with} \quad \Delta \tilde{\mathbf{p}}_j = \tilde{\mathbf{p}}_{j+1} - \tilde{\mathbf{p}}_j \\ \mathbf{V}_j &= \left[\Delta \mathbf{r}_0, \Delta \mathbf{r}_1, \dots, \Delta \mathbf{r}_{j-1} \right], \quad \text{with} \quad \Delta \mathbf{r}_j = \mathbf{r}_{j+1} - \mathbf{r}_j\end{aligned}\tag{20}$$

Using the two datasets W_j and V_j , the multi-secant equation for the inverse interface Jacobian reads:

$$J_{\tilde{\mathbf{r}}}^{-1}(\tilde{\mathbf{p}}_j) \mathbf{V}_j \approx \mathbf{W}_j\tag{21}$$

Anderson acceleration [6] We seek for the minimum of the approximate inverse Jacobian norm while satisfying the multi-secant equation constraint. Let L be the associated Lagrangian:

$$L(J_{\tilde{\mathbf{r}}}^{-1}, \lambda) := \frac{1}{2} \|J_{\tilde{\mathbf{r}}}^{-1}\|_F^2 + \lambda (J_{\tilde{\mathbf{r}}}^{-1} \mathbf{V}_j - \mathbf{W}_j)\tag{22}$$

λ is the Lagrange multiplier associated with the multi-secant constraint. The solution $(\bar{\lambda}, \bar{J}^{-1})$ is obtained by making stationary the Lagrangian. The inverse of the Jacobian can be written as:

$$\bar{J}^{-1} = \mathbf{W}_j (\mathbf{V}_j^T \mathbf{V}_j)^{-1} \mathbf{V}_j^T\tag{23}$$

The Anderson acceleration does not provide an approximation for the entire inverse Jacobian matrix; it focuses solely on the pertinent directions leading to the zero residual. We seek for the vector α minimizing the norm:

$$\alpha_j = \arg \min_{\alpha} \|\mathbf{V}_j \alpha + \mathbf{r}_j\|_2\tag{24}$$

The same direction is then applied to \mathbf{W}_j to get the prediction in terms of input variations:

$$\mathbf{p}_{j+1} = \tilde{\mathbf{p}}_j + \mathbf{W}_j \alpha_j\tag{25}$$

Information from past increments can be used to improve the performance. However, the optimal number of reused time increments for the Anderson acceleration is highly dependent on the problem at stake. During the iterative process, dependent data within the data-sets \mathbf{W}_j and \mathbf{V}_j are removed using vector orthogonalization based on QR algorithms. Algorithm 2 lists the details of the Anderson accelerator.

Broyden's quasi-Newton method The Broyden's quasi-Newton method approximates the full inverse Jacobian matrix by minimizing the difference with the Jacobian at the previous increment under the multi-secant equation constraint:

$$L(J_{\tilde{\mathbf{r}}}^{-1}, \lambda) := \frac{1}{2} \|J_{\tilde{\mathbf{r}}}^{-1}(\mathbf{p}_j) - J_{\tilde{\mathbf{r}}}^{-1,(N)}(\mathbf{p}_j)\|_F^2 + \lambda (J_{\tilde{\mathbf{r}}}^{-1} \mathbf{V}_j - \mathbf{W}_j)\tag{26}$$

Information from previous time increments are captured implicitly throughout this approach. The inverse of the Jacobian minimizing the Lagrangian function reads:

$$\bar{J}^{-1} = J_{\tilde{\mathbf{r}}}^{-1,(N)} + \left(\mathbf{W}_j - J_{\tilde{\mathbf{r}}}^{-1,(N)} \mathbf{V}_j \right) (\mathbf{V}_j^T \mathbf{V}_j)^{-1} \mathbf{V}_j^T\tag{27}$$

Storing the complete inverse Jacobian matrix is impractical for real problems. Therefore, the inverse Jacobian storage is implemented through a truncated Singular Value Decomposition (SVD) to mitigate memory consumption. The SVD update is carried out using an incremental SVD [44] to minimize computational costs. By using such an approach, the full rank SVD is never computed.

Algorithm 2: Anderson acceleration

Extrapolation: $\mathbf{p}_0 = f(\mathbf{p}^n, \mathbf{p}^{n-1}, \dots)$ Constant relaxation: $\tilde{\mathbf{p}}_0 = H(x_0)$, $\mathbf{r}_0 = \tilde{\mathbf{p}}_0 - \mathbf{p}_0$ $\mathbf{p}_1 = \mathbf{p}_0 + \omega_0 \mathbf{r}_0$ **for** $j \in [0, \dots, j_{\max}]$ **do** Compute Picard iterate and residual: $\tilde{\mathbf{p}}_j = H(\mathbf{p}_j)$, $\mathbf{r}_j = \tilde{\mathbf{p}}_j - \mathbf{p}_j$ **if** (*!converged*) **then** Build \mathbf{V}_j and \mathbf{W}_j input-output datasets: $\mathbf{V}_j = [\Delta \mathbf{r}_0, \dots, \Delta \mathbf{r}_{j-1}]$, $\Delta \mathbf{r}_i = \mathbf{r}_{i+1} - \mathbf{r}_i$ $\mathbf{W}_j = [\Delta \tilde{\mathbf{p}}_0, \dots, \Delta \tilde{\mathbf{p}}_{j-1}]$, $\Delta \tilde{\mathbf{p}}_i = \tilde{\mathbf{p}}_{i+1} - \tilde{\mathbf{p}}_i$ Perform QR decomposition of \mathbf{V}_j : $\mathbf{V}_j = \mathbf{Q}\mathbf{U}$ Minimize $\|\mathbf{V}_j \alpha + \mathbf{r}_j\|_2$ using a least-square method: $\mathbf{U}\alpha = -\mathbf{Q}^T \mathbf{r}_j$ Compute the correction and the accelerated solution: $\Delta \tilde{\mathbf{p}}_j = \mathbf{W}_j \alpha$ $\mathbf{p}_{j+1} = \tilde{\mathbf{p}}_j + \Delta \tilde{\mathbf{p}}_j$ **end****end**

The solution correction $\Delta \tilde{\mathbf{p}}_j$ is obtained using the Newton-Raphson update formula. The predicted (or the accelerated) value reads:

$$\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{p}}_j + \Delta \tilde{\mathbf{p}}_j = \tilde{\mathbf{p}}_j - J_{\tilde{\mathbf{r}}}^{-1} \mathbf{r}_j \quad (28)$$

The pseudo-code of the Broyden's algorithm is detailed in Algorithm 3.

Remark 3. The GLIC is implemented in Abaqus through co-simulation. With this approach, the co-simulation coordinates two Abaqus concurrent simulations and their communications between each other. The quantities communicated in the GLIC case are displacements from global to local and reaction forces from local to global. Automatic mapping enables couplings between finite element interfaces with mismatching nodes, with a linear interpolation in space. If the simulations run with different increment sizes, a linear interpolation in time occurs. However, within this paper, only matching nodes and increment sizes have been considered. All the three accelerators described in Section 2.2 are implemented in the Abaqus co-simulation engine. From the practical point of view, beside running the global and local simulations concurrently, a so-called configuration file needs to be written and run in order to drive the Abaqus co-simulation engine. Within the configuration file, the user selects the accelerator technique and associated parameters, defines the field quantities that are exchanged and the pairs of outputs/inputs for, provides convergence and communication controls. Last but not least, it is important to note that users can substitute one of the two Abaqus simulations in the GLIC with their own third-party code, making use of special co-simulation APIs.

Remark 4. It is important to state that, despite all the efforts to accelerate the convergence of the global-local iterations and optimize the number of global and local Newton-Raphson iterations, the GLIC will never be more efficient in terms of computing time than the monolithic finite elements model direct simulation, that will be taken as reference for accuracy measurements in what follows. More precisely, the GLIC will be slower by the factor close to the average number of global-local iterations during a given co-simulation run. However, for most of industrial cases, the computing time from simulations is often negligible in comparison to the time spent by engineers in building finite element models and, most importantly, is scalable with computing resources. The convenience of a global-local approach resides in the mitigation of the modeling burdens for creating multiple complex finite element models for multiple load cases or geometry configurations.

Algorithm 3: Broyden's quasi-Newton method

```
Extrapolation:  $\mathbf{p}_0 = f(\mathbf{p}^n, \mathbf{p}^{n-1}, \dots)$ 
Constant relaxation:
 $\tilde{\mathbf{p}}_0 = H(x_0)$ ,  $\mathbf{r}_0 = \tilde{\mathbf{p}}_0 - \mathbf{p}_0$ 
 $\mathbf{p}_1 = \mathbf{p}_0 + \omega_0 \mathbf{r}_0$ 
for  $j \in [0, \dots, j_{\max}]$  do
    Compute Picard iterate and residual:  $\tilde{\mathbf{p}}_j = H(\mathbf{p}_j)$ ,  $\mathbf{r}_j = \tilde{\mathbf{p}}_j - \mathbf{p}_j$ 
    if (!converged) then
        Build  $\mathbf{V}_j$  and  $\mathbf{W}_j$  input-output datasets:
         $\mathbf{V}_j = [\Delta \mathbf{r}_0, \dots, \Delta \mathbf{r}_{j-1}]$ ,  $\Delta \mathbf{r}_i = \mathbf{r}_{i+1} - \mathbf{r}_i$ 
         $\mathbf{W}_j = [\Delta \tilde{\mathbf{p}}_0, \dots, \Delta \tilde{\mathbf{p}}_{j-1}]$ ,  $\Delta \tilde{\mathbf{p}}_i = \tilde{\mathbf{p}}_{i+1} - \tilde{\mathbf{p}}_i$ 
        Perform QR decomposition of  $\mathbf{V}_j$ :  $\mathbf{V}_j = \mathbf{Q}\mathbf{U}$ 
        Compute  $\mathbf{Z} = (\mathbf{V}_j^T \mathbf{V}_j)^{-1} \mathbf{V}_j^T \mathbf{r}_j = \mathbf{U}^{-1} \mathbf{Q}^T \mathbf{r}_j$ 
        Update the approximate inverse Jacobian:
         $J_{\tilde{\mathbf{r}}}^{-1} \mathbf{r}_j(\tilde{\mathbf{p}}_j) = J_{\tilde{\mathbf{r}}}^{-1, (n)} + (\mathbf{W}_j - J_{\tilde{\mathbf{r}}}^{-1, (n)} \mathbf{V}_j) \mathbf{Z}$ 
        Compute the correction and the accelerated solution:
         $\Delta \tilde{\mathbf{p}}_j = -J_{\tilde{\mathbf{r}}}^{-1}(\tilde{\mathbf{p}}_j) \mathbf{r}_j$ 
         $\mathbf{p}_{j+1} = \tilde{\mathbf{p}}_j + \Delta \tilde{\mathbf{p}}_j$ 
    end
end
```

Indeed, the GLIC proposes a unique global finite element model that is used for multiple simulations, whereas multiple specialized local models are applied where required, while keeping the global model unchanged. This is especially practical and a powerful approach in the aeronautic industry, where multiple levels of abstraction are defined for a given aircraft. The Authors refer to this concept as non-intrusive modeling.

3 Use cases

A set of generic use-cases was defined to mimic situations met by Airbus where a direct unidirectional sub-modeling approach is not satisfactory:

- significant material non-linearity occurs locally in an area that is not modeled at global scale (plate with hole)
- significant material non-linearity occurs locally in an area that is modeled at global scale, but through a macro-element (elementary bolt)

3.1 Holed plate with localized elasto-plasticity

The first case under study is a cylindrical holed plate section with localized elasto-plastic behavior. Figure 3 shows the monolithic finite element model of reference. The plate has width 10 cm along the X direction, length 20 cm along the Z direction, thickness 1 mm and a cylindrical curvature along the X direction of radius 0.2 m. The hole is perfectly centered with respect to the plate and measures 6 mm of radius. A mesh of solid elements is applied all around the hole to an outer radius of 12 mm, with 8-node hexahedral elements with reduced integration of variable size counting 8 times through the thickness, 56 times in the tangential direction through the full circle and 8 times through the radial direction, for a total of 3584 solid elements. The rest of the plate is

meshed with 4-node shell elements with Simpson reduced integration up to a mesh size of 6 mm and connected to the solid elements with a shell-to-solid coupling (see [1] for details on this type of connection).

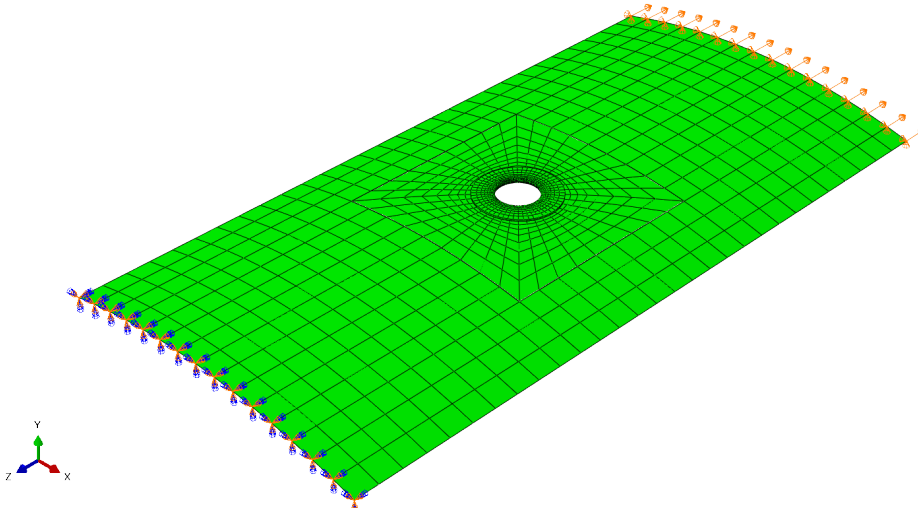


Figure 3: Holed plate finite element model of reference.

The plate is clamped at the bottom end at $Z = 0.2$ m along the full width and pulled at the top end at $Z = 0$ m with an imposed displacement of 1 mm along the Z direction. An elastic material is applied to the full model with a Young modulus of 210 GPa and Poisson coefficient of 0.3. The plastic material behavior is defined linearly interpolating the tabular values of Table 1 and is applied to the solid elements around the hole. A nonlinear static analysis under the assumption of finite strains is therefore run with Abaqus, starting with a increment 50 times smaller than the full step size and making use of the default convergence controls that adjust the increment size in function of convergence rates up to a maximum increment size 10 times smaller than the full step size.

Stress (MPa)	Plastic strain
400	0.0
420	0.02
500	0.2
600	0.5
625	0.6
650	0.8

Table 1: Plastic material behavior on the solid elements around the hole.

The contour plot in Figure 4 shows results in terms of displacement magnitude projected to the deformed full mesh. All nodal displacements magnitudes vary between 0 mm along the clamp and 1 mm at the other end of the plate.

The contour plot in Figure 5 shows results in terms of equivalent plastic strain projected to the solid elements mesh around the hole, which was the only part of the model subjected to elasto-plastic material behavior. Equivalent plastic strains reach a maximum plastic strain value of 0.04486 on the side faces of the hole, where stresses concentrate.

The solution in Figures 4 and 5 has been achieved with 26 increments with a number of Newton-Raphson iterations varying between 2 (at the beginning of the simulation) and 6 (when yield stress is reached).

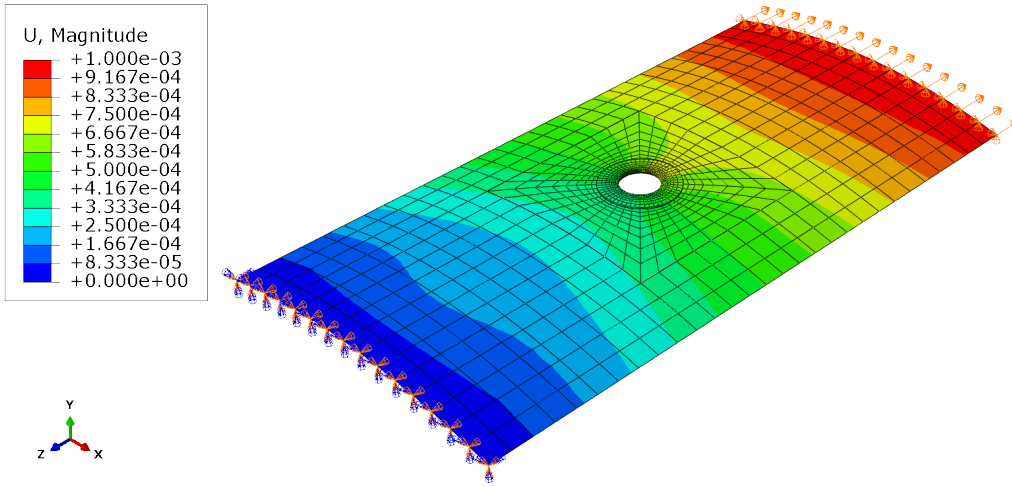


Figure 4: Results in terms of displacement magnitude (mm) for the holed plate finite element model of reference.

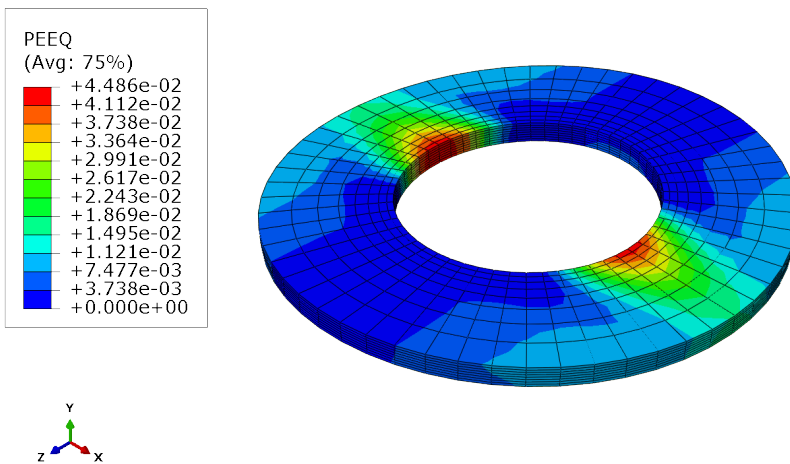


Figure 5: Results in terms of equivalent plastic strain for the holed plate finite element model taken as reference.

The GLIC approach is applied here with coupling a coarse global model with uniform mesh of quadrilateral shell elements of size 6 mm without considering the presence of the hole to a local mesh that is identical to the mesh of the reference simulation on the squared patch of size 60 mm around the hole. Automatic convergence controls of the global-local iterations have been implemented, namely, an increment-focused absolute criterion, an increment-focused relative criterion and a step-focused relative criterion, that have been set to 10^{-3} s, 10^{-3} s and 10^{-4} s, respectively, in what follows. This use case is challenging for the GLIC algorithm as the local model is different from the overlapped global region on multiple sides: element type, mesh size, material definition and evolution during the simulation. As the simple fixed-iteration coupling scheme converges too slowly for practical usage, only the accelerator techniques can be considered for this use case.

To appreciate the theoretical aspects of the GLIC, the contour plots from Figure 6 show Von Mises stress fields in the global and local analyses for the co-simulation run with Aitken’s relaxation (GLIC-Aitken). The contour plot color values vary in the range [500, 1250] MPa in order to qualitatively highlight the stress variation at the global-local interface. The overlapped region in the global model can be nicely identified with the larger jump in Von Mises stress values at the 9×9 elements squared patch over the center of the model: this is due to the correction forces in the global analysis that applied all along the global-local interface in the global model. On the other hand, the Von Mises stress values at the elements along the boundary of the local analysis are in accordance with the Von Mises stress values in the non-overlapped region of the global analysis.

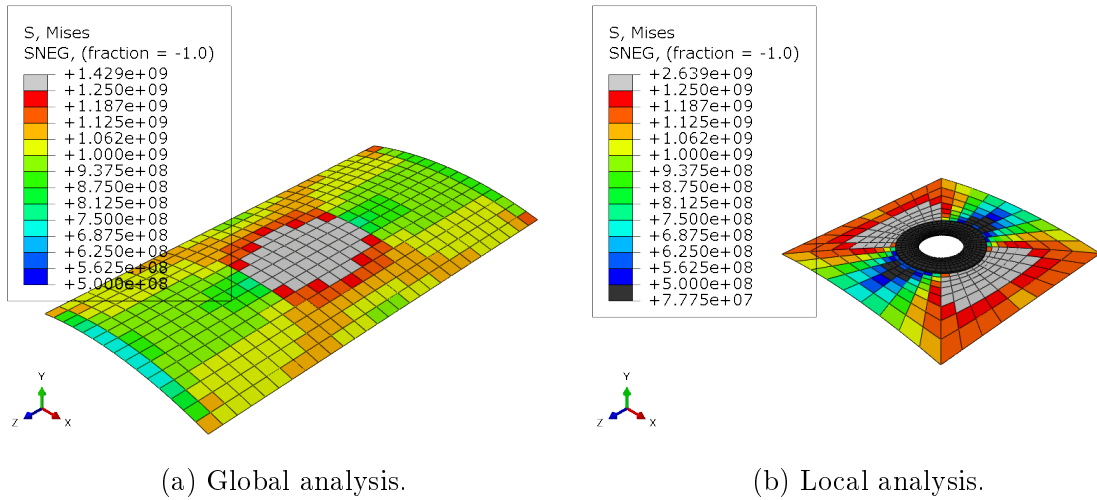
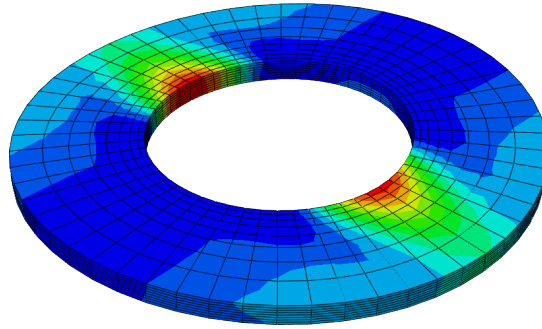
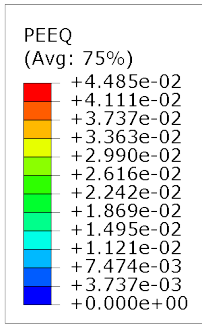


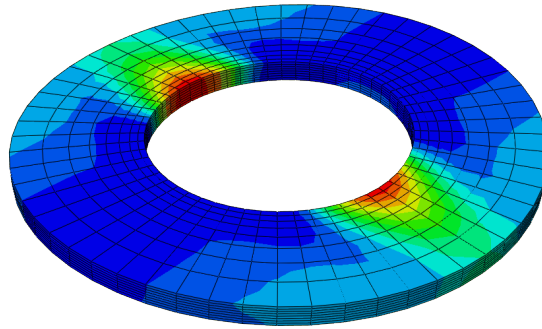
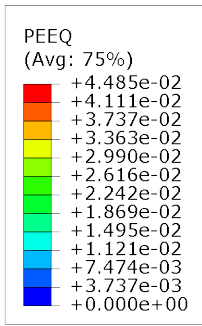
Figure 6: Results in terms of Von Mises stress (MPa) from the GLIC-Aitken simulation.

In terms of accuracy, the contour plots in Figure 7 show results in terms of equivalent plastic strains obtained with Aitken’s relaxation (GLIC-Aitken), Anderson acceleration technique (GLIC-Anderson) and Broyden quasi-Newton method (GLIC-Broyden) in Figures 7a, 7b and 7c, respectively. Such results are identical for engineering purposes and consistently show a maximum peak in equivalent plastic strain of 0.04485, which is equivalent to an error of 0.02% due to the same global-local convergence threshold for all three simulations.

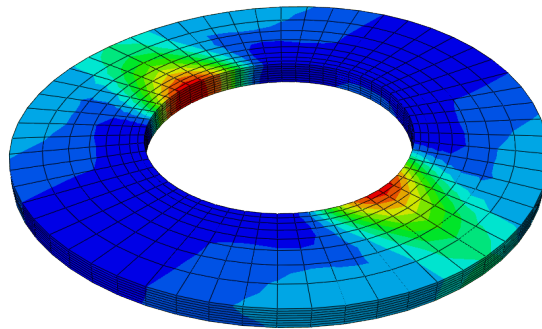
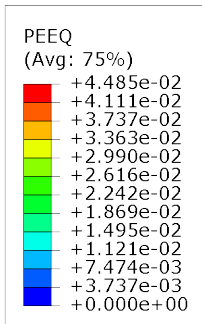
In terms of performance, the numbers of iterations in time are reported in Figure 8. The plot on the top shows the total number of Newton-Raphson iterations in the global analysis (elastic finite strain model), the plot in the middle shows the total number of Newton-Raphson iterations in the local analysis (elasto-plastic finite strain model with coarse uniform mesh) and the plot on the bottom shows the number of iterations for the global-local coupling, which is a common multiplier of both global and local total



(a) GLIC-Aitken simulation.



(b) GLIC-Anderson simulation.



(c) GLIC-Broyden simulation.

Figure 7: Results in terms of equivalent plastic strain for the GLIC simulations.

Newton-Raphson iterations. The blue squares, the red circles and the black triangles are on curves out of the GLIC-Aitken simulation, GLIC-Anderson and GLIC-Broyden simulation, respectively. Whereas curves from the GLIC-Anderson and GLIC-Broyden simulations are perfectly overlapped with most of the iterations happening in the first half of the analysis, the GLIC-Aitken simulation curve behaves differently, with a sudden peak after time instant 0.4 s, a few increments after a cutback in the increment size is triggered by the solver to optimize convergence.

The Authors want to note that the curves out of the global analysis quickly converge at each increment and global-local iteration, as the model is elastic.

Overall, the GLIC-Aitken simulation is the fastest, independently from the sizes of global and local models. As highlighted in Table 2, the total number of global-local iterations N^{GL} is 1.5 times smaller in GLIC-Aitken than in GLIC-Anderson and GLIC-Broyden. As a consequence, also in terms of total number of Newton-Raphson iterations in the global analysis N_{iter}^G and in the local analysis N_{iter}^L , GLIC-Aitken is 1.57 times faster than both GLIC-Anderson and GLIC-Broyden in the global analysis and 1.51 times in the local analysis. Due to convergence control heuristics, the number of increments in the global analysis N_{inc}^G and in the local analysis N_{inc}^L are different. Namely, they are both smaller in the GLIC-Aitken than in both GLIC-Anderson and GLIC-Broyden, because of faster convergence on average. Last but not least, also the total number of global-local iterations is 1.52 times smaller in GLIC-Aitken than in both GLIC-Anderson and GLIC-Broyden.

	N_{inc}^G	N_{iter}^G	N_{inc}^L	N_{iter}^L	N^{GL}
GLIC-Aitken	32	163	35	338	65
GLIC-Anderson	41	256	44	513	99
GLIC-Broyden	41	256	44	513	99

Table 2: Comparison of total performance values between GLIC-Aitken, GLIC-Anderson and GLIC-Broyden.

The Newton-Raphson convergence controls employed in the simulation above have been set to the values that are proposed by default in the Abaqus solver. The readers are encouraged to consult the Abaqus manual [1] for comprehensive details on such convergence control values. To verify the inexact solver strategy introduced in Section 2.1 (GLIC-IS), the following simulation have been run relaxing the convergence controls. Namely, the convergence criterion for the ratio of the largest residual to the corresponding average flux norm for convergence and the convergence criterion of the ratio of the largest solution correction to the largest corresponding incremental solution value have been increase from 0.005 and 0.01 to 1 and 1, respectively, for both global and local analyses. Contour plots in Figure 9 show the comparison in terms of equivalent plastic strain in the local model between the GLIC-Aitken and GLIC-IS-Aitken simulations.

Maximum values for the equivalent plastic strains are 0.04485 for the GLIC-Aitken simulation and 0.04486 for the GLIC-IS-Aitken simulation, with a negligible degradation in accuracy of approximately 0.02%. On the other hand, in terms of performance, the plots in Figure 10 show the comparison of the number of Newton-Raphson iterations during the global and local analyses between GLIC-Aitken and GLIC-IS-Aitken simulations. Although the curves have similar values in terms of iteration numbers on the vertical axes, the total number of iterations is lower in the relaxed case, due to larger increments, resulting in less equation solves. Indeed, the time increment is automatically increased or cut off based on convergence rate, which is faster with relaxed parameters.

As a more accurate performance analysis, the same numbers considered for the comparison between GLIC-Aitken, GLIC-Anderson and GLIC-Broyden are considered here

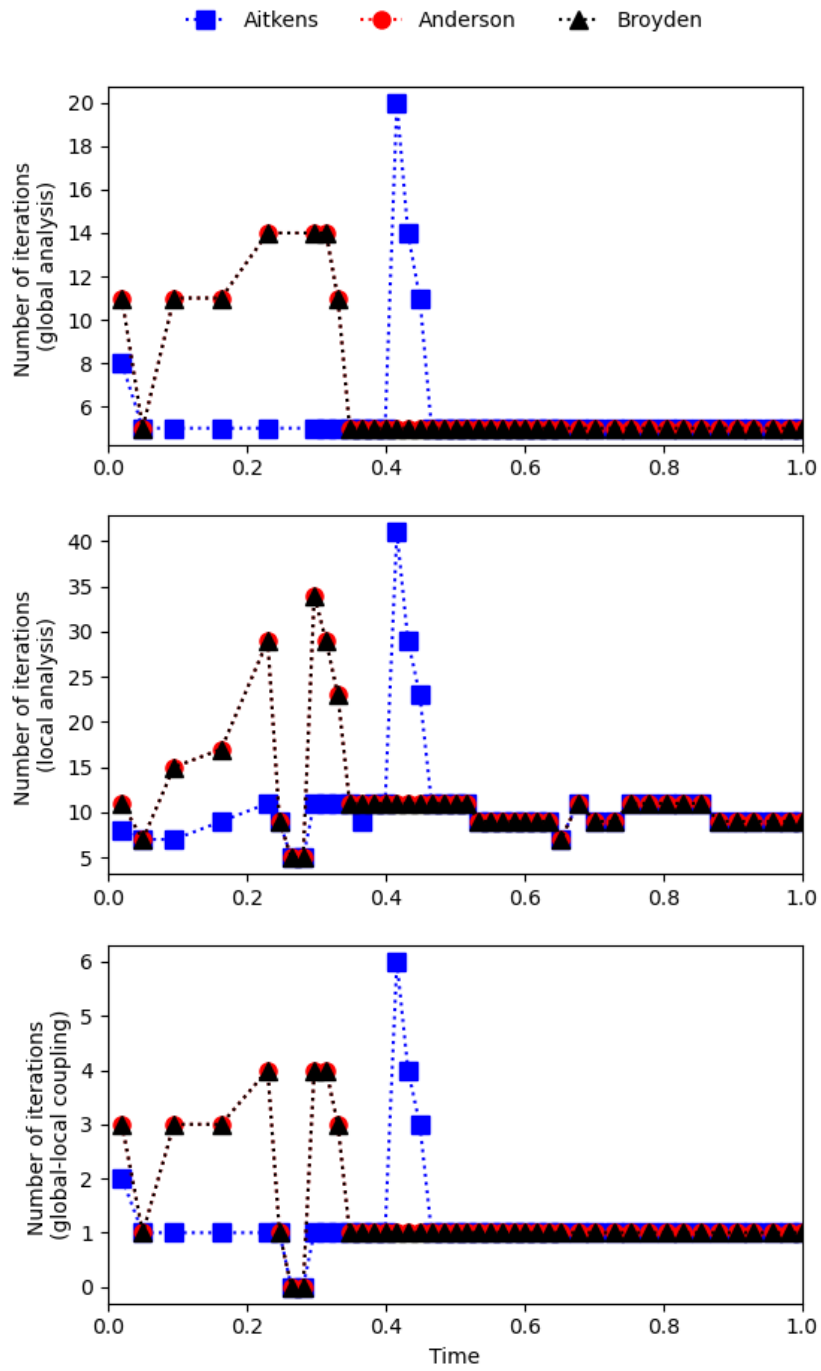


Figure 8: Comparison of the number of iterations in time between the GLIC-Aitken, GLIC-Anderson and GLIC-Broyden simulations.

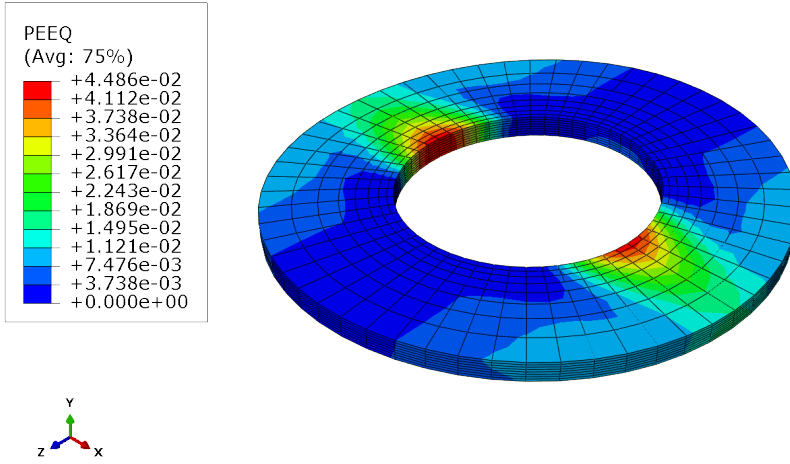


Figure 9: Results in terms of equivalent plastic strain in the local analysis from the GLIC-IS-Aitken simulation.

in Table 3, comparing GLIC-Aitken and GLIC-IS-Aitken. Whereas the total number of global-local iterations N^{GL} is very similar in both simulations, as well as the total number of Newton-Raphson iterations in the global analysis N_{iter}^G , which is counterintuitively 3% smaller in GLIC-Aitken than in GLIC-IS-Aitken. The total number of Newton-Raphson iterations in the local analysis N_{iter}^L is 11% smaller in GLIC-IS-Aitken than in GLIC-Aitken, due to small numbers of increments in both global and local analyses, N_{iter}^G and N_{iter}^L , respectively.

	N_{inc}^G	N_{iter}^G	N_{inc}^L	N_{iter}^L	N^{GL}
GLIC-Aitken	32	163	35	338	65
GLIC-IS-Aitken	24	168	27	304	64

Table 3: Comparison of total performance numbers between GLIC-Aitken and GLIC-IS-Aitken.

This time, the convenience in terms of performance depends on the size of the global and local models. As the number of variables in the local model is 16269, much larger than the number of variables in the global model, that is 2880, then in this case GLIC-IS-Aitken is more convenient than GLIC-Aitken.

Similar behaviors are encountered relaxing convergence criterion values with the Anderson acceleration and Broyden's method.

3.2 Structure with bolted joints

The second case under study is a bolted joint connecting two planar plates together. Even if derived into a simplistic use-case (Figure 11), the situation explored here is representative of actual industrial concerns where elementary bolts are represented by connector elements at global scale and may meet critical loading conditions, which requires further refined investigations with local models of higher physical representability. The full model is an assembly composed of 3 parts in contact between each other: the two plates and the bolt. Figure 11 shows the reference finite element model applied to the geometry. The two plates are identical, of width 40 mm along the X direction, length 100 mm along the Y direction, thickness 2 mm along the Z direction and are stacked in the Z direction. As for the holed plate, the bolt is perfectly centered across the two plates, its head has a diameter 10 mm and height 3 mm, its body has a diameter 6.35 mm and a height 7 mm.

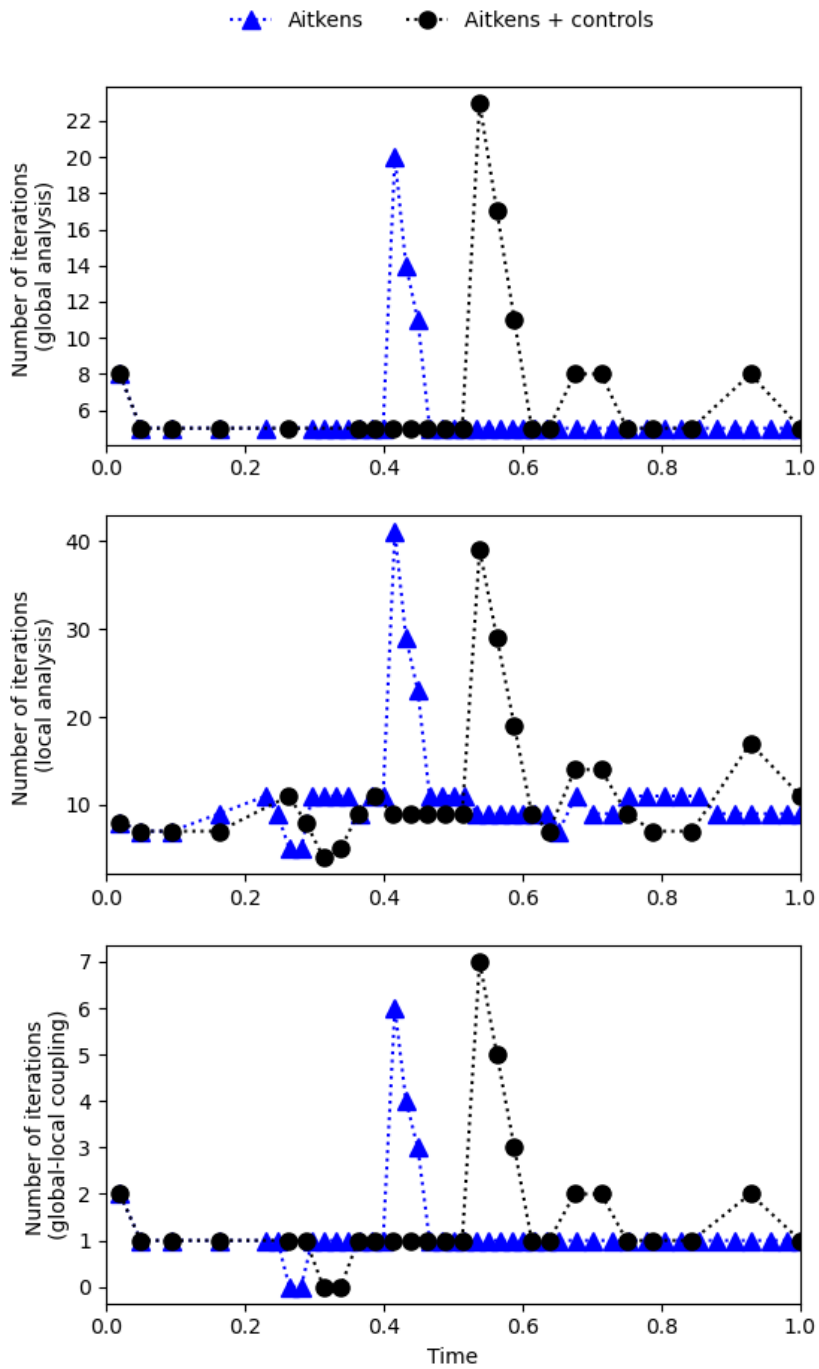


Figure 10: Comparison of the number of iterations in time between the GLIC-Aitken and GLIC-IS-Aitken simulations.

A mesh of solid elements is applied around the bolt up to a diameter 15 mm, with 4-noded tetrahedric elements of variable size, with smaller ones over the surfaces in contact. As for the holed plate, the rest of the assembly is meshed with 4-noded shell elements with Simpson reduced integration up to a mesh size 2.5 mm.

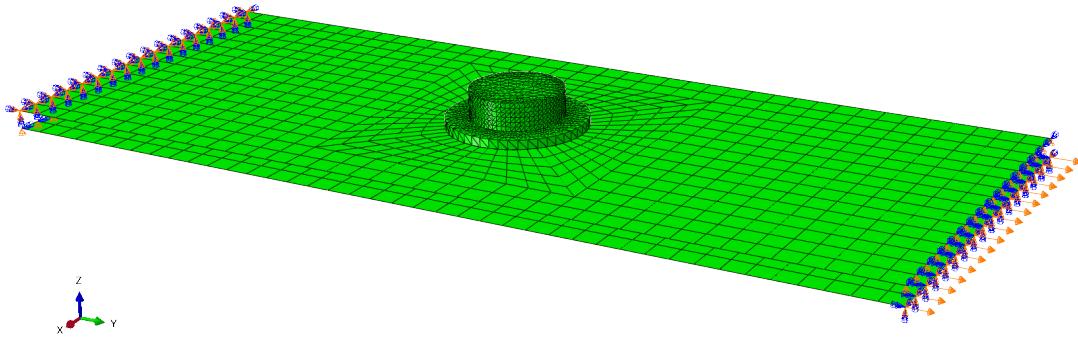


Figure 11: Bolted joint finite element model of reference.

As one of the plates is clamped on the bottom edge and simply supported in the Z direction on the top edge, the other plate is simply supported in the Z direction on the bottom edge and pulled up in the Y direction on the top edge for a displacement of 2. Elastic materials are applied to the plates and the bolt with Young moduli 70 GPa and 110 GPa, respectively, and a Poisson coefficient 0.3 in both cases. Small sliding contact with friction 0.2 and slip tolerance 0.005 mm is defined between the plates and between the bolt and the plates. A nonlinear static analysis under the assumption finite strains is run with Abaqus in two steps. In the first step, a pre-load of 1 kN is applied to the bolt as compression in the Z direction through bushing connectors of elasticity 100 kN/mm for axial forces against translations and 10^5 for moments against rotations. In the second step, one of the plate is pulled and loads the bolt with shear stress through contact.

The contour plot in Figure 12 shows results in terms of displacement magnitude projected to the deformed full mesh. All nodal displacements magnitudes vary between 0 mm along the clamp and 2.058 mm.

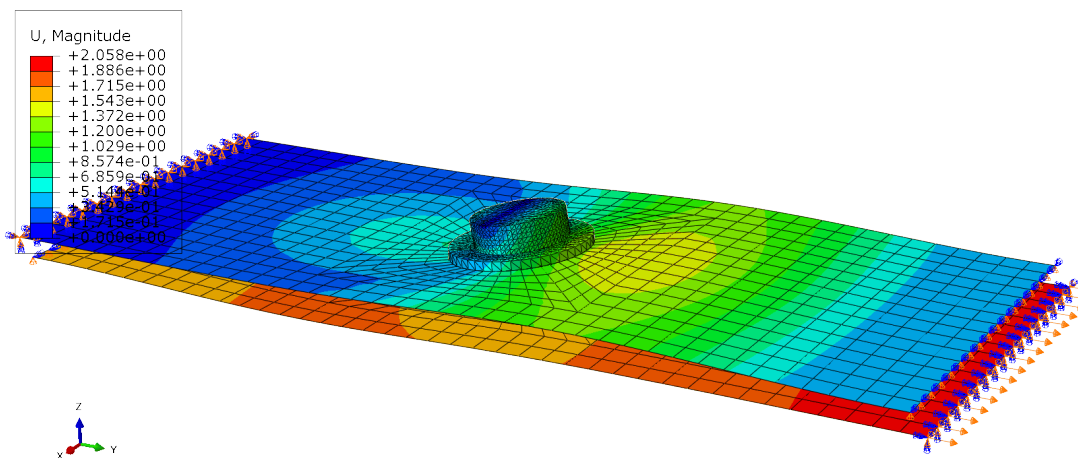


Figure 12: Results in terms of displacement magnitude (mm) for the bolted joint assembly finite element model of reference.

The contour plot in Figure 13 shows results in terms of contact pressure and contact shear, respectively, projected to the solid elements mesh of the bolt. Contact pressure goes up to an absolute value of $6.96 \cdot 10^3$ MPa in the middle of the bolt body in an almost symmetric manner. Contact shear participates with values ranging between $-1.25 \cdot 10^3$ MPa and $1.37 \cdot 10^3$ MPa.

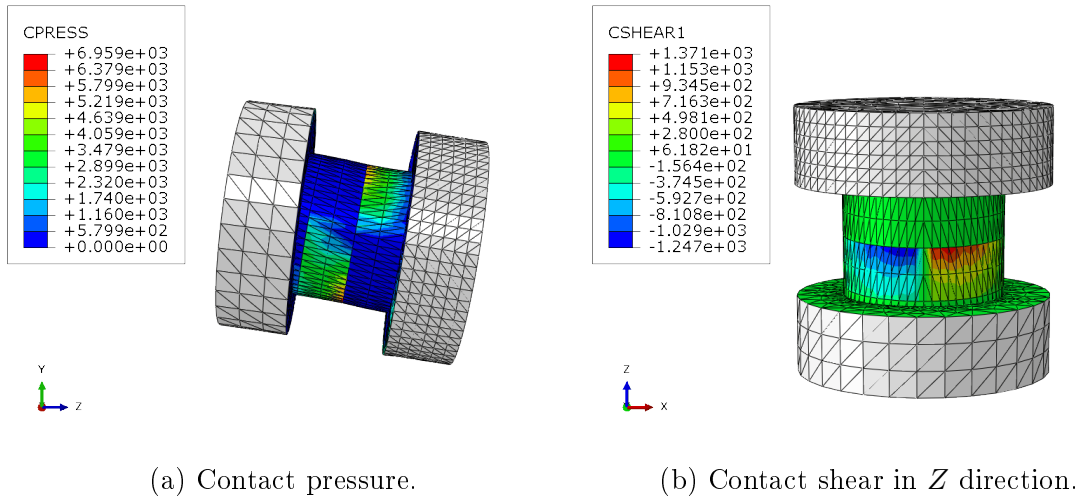


Figure 13: Results in terms of contact interaction (MPa) on the bolt finite element model taken as reference.

The solution in this case has been achieved with 15 increments per step. The first step ran with a number of Newton-Raphson iterations varying between 1 and 7, on average $2.1\bar{3}$, the peak of 7 caused by contact severe discontinuities. The second step ran with a number of Newton-Raphson iterations varying between 5 and 9, on average $6.2\bar{6}$, in part due to contact, in part due to geometrical nonlinearities.

In accordance to the GLIC strategy and similarly to the first use case described in Section 3.1, the global model is meshed with a uniform coarse mesh of 4-noded shell elements of mesh size 2.5 mm, whereas the local model is defined on the central 30×30 mm² squared patch that covers the bolt and the immediate area of the two plates, meshed partly with 4-noded shell elements and partly with 3-noded tetrahedric elements, connected with shell-to-solid coupling. The overlap of the local model to the global model exactly represents what was defined in the reference model of Figure 11, as the goal of this verification exercise is to obtain identical results to reference. Automatic convergence control tolerances of 10^{-4} , 10^{-3} and 10^{-3} have been used to stop global-local iterations as increment-focused absolute criterion, increment-focused relative criterion and step-focused relative criterion, respectively.

As the bolt is fully modeled in 3D in the local model, its behavior is captured in the global model with a bushing connector with elastic behavior of 275 kN/mm in X and Y directions and $348 \cdot 10^3$ kN/mm in the Z direction.

This case is particularly challenging for the fixed-point GLIC algorithm as the local model is much more refined and globally stiffer than the global overlapped region and the spectral radius of the amplification matrix might likely be larger than 1, which would lead to divergence. Indeed, also in this case, the fixed-iteration GLIC algorithm without any acceleration failed to converge. Therefore, only the results obtained with accelerator techniques are detailed in what follows.

In terms of accuracy, Figure 14 shows contour plot results in terms of contact pressure and contact shear in the Z direction for co-simulation runs with GLIC-Aitken, Anderson accelerator technique and GLIC-Broyden. Results are qualitatively identical between

each other and are close enough to the reference solution, as detailed on maximum and minimum values with Table 4. This possibly indicates that the error is chiefly due to the GLIC convergence criteria rather than the chosen accelerator methodology.

	Max CPRESS	Max CSHEAR1	Min CSHEAR1
Reference	6.959	1.371	-1.247
GLIC-Aitken	7.01	1.377	-1.252
GLIC-Anderson	7.01	1.377	-1.252
GLIC-Broyden	7.01	1.377	-1.252

Table 4: Contact interaction results comparison (values expressed in GPa).

In terms of performance, the numbers of iterations in time are reported in Figure 15 for the second step of the analysis only. The plot on the top shows the total number of Newton-Raphson iterations in the global analysis (elastic finite strain model with coarse uniform mesh), the plot in the middle shows the total number of Newton-Raphson iterations in the local analysis (where contact occurs) and the plot on the bottom shows the number of iterations for the global-local coupling. The blue squares, the red circles and the black triangles are results from GLIC-Aitken, GLIC-Anderson and GLIC-Broyden, respectively. This time, the red and black curves are not perfectly overlapped but are roughly on top of each other, whereas the blue curve is consistently on the bottom part of the plot.

This means that the GLIC-Aitken turns out to be the faster accelerator technique when counting the number of total iterations during the analysis. More precisely, as evaluated with numbers in Table 5, global and local Newton-Raphson convergence rates are similar, but the global-local iteration is faster with GLIC-Aitken than with GLIC-Anderson and GLIC-Broyden. The number of increments is identical in the three cases, meaning that the convergence control heuristics work in the same way. The total number of Newton-Raphson iterations in the global analysis N_{iter}^G is 1.49 times smaller in GLIC-Aitken than in GLIC-Anderson and 1.59 times than GLIC-Broyden. The total number of Newton-Raphson iterations in the local analysis N_{iter}^L is 1.4 times smaller in GLIC-Aitken than in GLIC-Anderson and 1.51 times than in GLIC-Broyden.

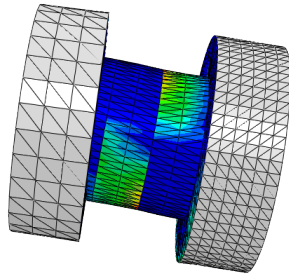
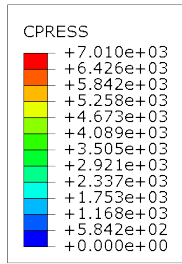
	N_{inc}^G	N_{iter}^G	N_{inc}^L	N_{iter}^L	N^{GL}
GLIC-Aitken	14	434	14	1093	151
GLIC-Anderson	14	647	14	1533	220
GLIC-Broyden	14	691	14	1648	234

Table 5: Comparison of total performance numbers between GLIC-Aitken, GLIC-Anderson and GLIC-Broyden for the bolted-joint use case.

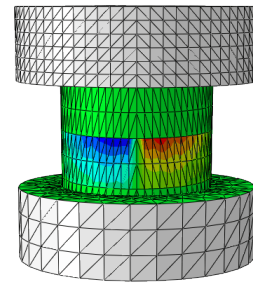
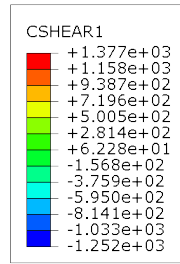
Relaxing the convergence criteria in the very same way of the first use case of Section 3.1 once more does not affect accuracy, as contact pressure and contact shear in Z direction do not change, but improve performance.

Figure 16 shows contour plots of contact interaction results in terms of contact pressure and contact shear in Z direction. Values are in line with reference results, maximum and minimum values being in between reference and co-simulation with Abaqus default convergence criteria.

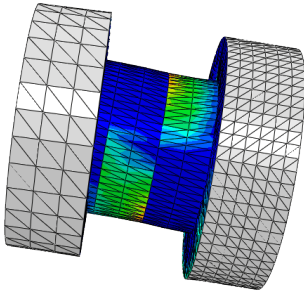
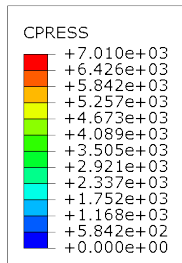
In terms of performance, Figure 17 shows the comparison in the total number of Newton-Raphson iterations in time between the Abaqus default convergence criteria and the relaxed convergence criteria. The performance gain using the latter strategy are



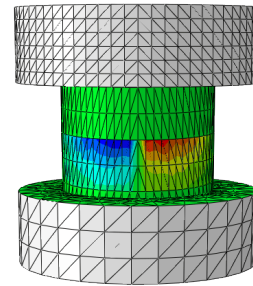
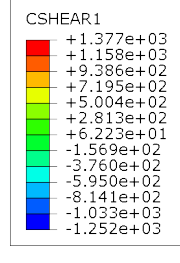
(a) Contact pressure with GLIC-Aitken.



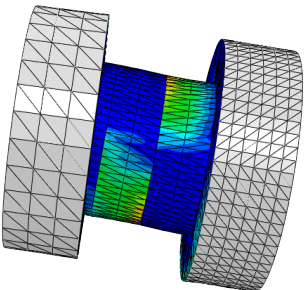
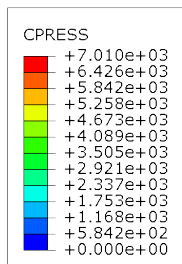
(b) Contact shear (Z dir.) with GLIC-Aitken.



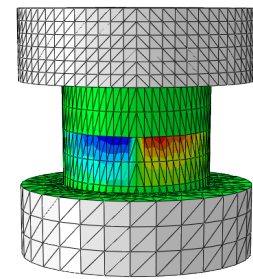
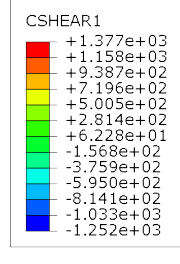
(c) Contact pressure with GLIC-Anderson.



(d) Contact shear (Z dir.) with GLIC-Anderson.



(e) Contact pressure with GLIC-Broyden.



(f) Contact shear (Z dir.) with GLIC-Broyden.

Figure 14: Results in terms of contact interaction (MPa) on the bolt for the GLIC simulations.

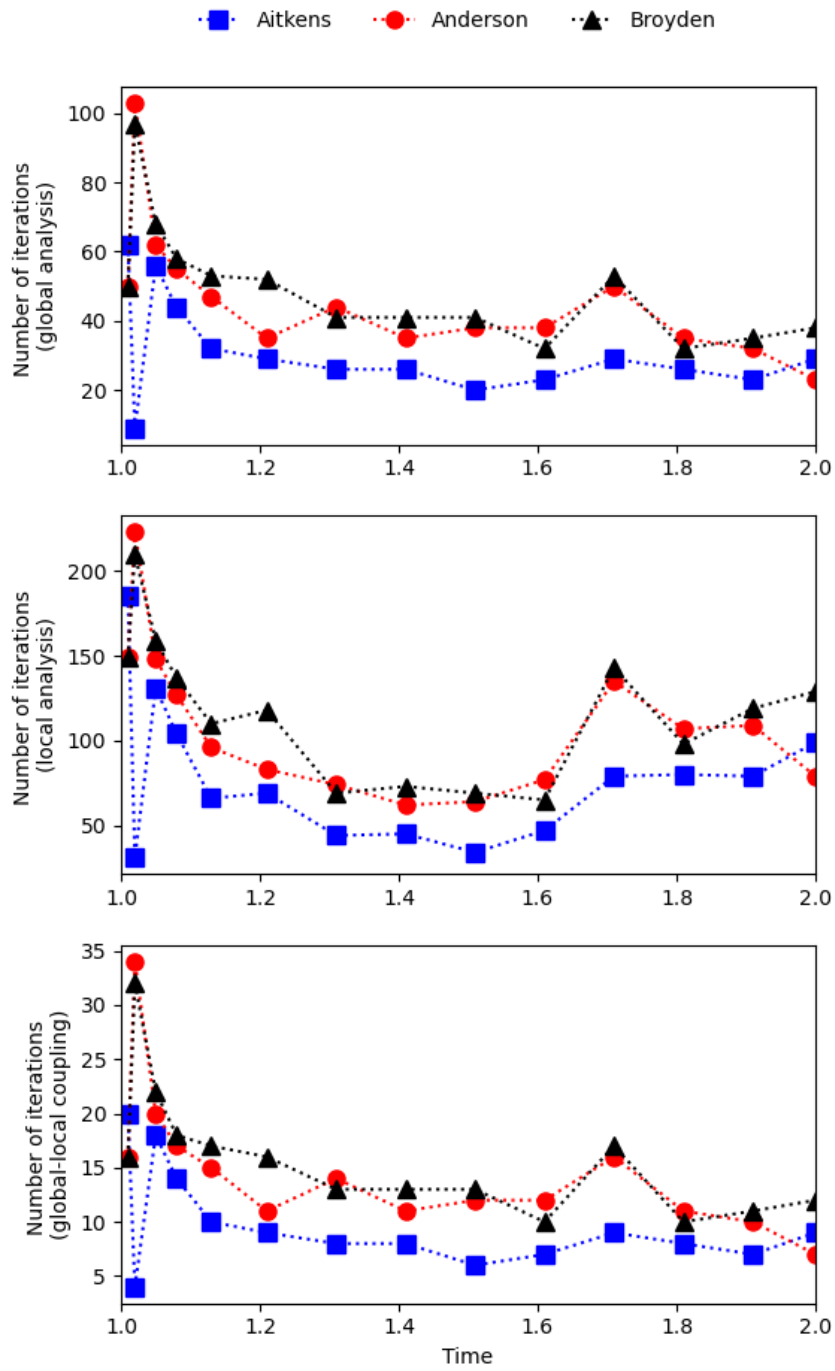


Figure 15: Comparison of the number of iterations in time between the GLIC-Aitken, GLIC-Anderson and GLIC-Broyden simulations for the bolted joint use case.

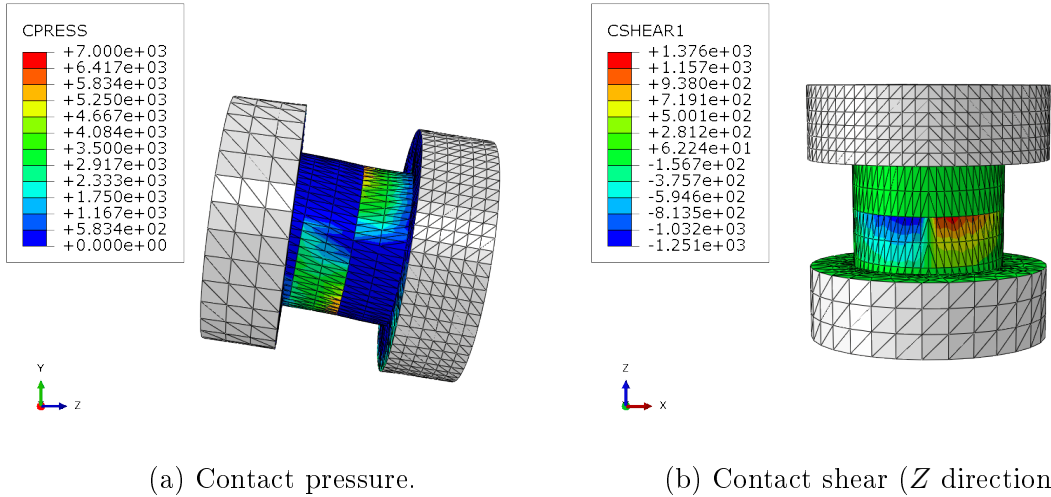


Figure 16: Results in terms of contact interaction (MPa) with the GLIC-IS-Aitken simulation.

appreciated chiefly on the local analysis, especially in the second half of the analysis, whereas there are no remarkable gains in the global analysis.

Also in this case, more precise numbers are listed in Table 6, comparing GLIC-Aitken and GLIC-IS-Aitken. As curves of global Newton-Raphson iterations and global-local iterations were almost completely overlapped, the total number of global-local iterations N^{GL} and the total number of Newton-Raphson iterations in the global analysis N_{iter}^G are also in this case slightly smaller in GLIC-Aitken than in GLIC-IS-Aitken, namely, 2.6% and 2.5% smaller, respectively. On the other hand, the total number of Newton-Raphson iterations in the local analysis N_{iter}^L is 1.5 times smaller in GLIC-IS-Aitken than in GLIC-Aitken. Also in this case, the number of increments N_{inc}^G and N_{inc}^L are identical in GLIC-Aitken and GLIC-IS-Aitken.

	N_{inc}^G	N_{iter}^G	N_{inc}^L	N_{iter}^L	N^{GL}
GLIC-Aitken	14	434	14	1093	151
GLIC-IS-Aitken	14	445	14	727	155

Table 6: Comparison of total performance numbers between GLIC-Aitken and GLIC-IS-Aitken for the bolted joint use case.

As also in this case the local model (33393 variables) is much larger than the global one (8412 variables), the GLIC-IS-Aitken approach is more convenient.

4 Conclusions and future works

Recent research works have proven the GLIC in conjunction with acceleration techniques as a robust and accurate solution for bridging multiple levels of modeling abstraction in computational structural mechanics, which is often a necessary approach when models are complex, in order to reduce the high and non-scalable modeling times and burdens. However, the simulation performance of the GLIC has always been reason of concern for the daily engineering practice. Within this article, the Authors study the GLIC implemented in Abaqus through co-simulation and compare acceleration techniques such as the Aitken's relaxation, Anderson acceleration and Broyden's quasi-Newton method. For the two highly nonlinear use cases under study, the Aitken's relaxation proved slightly more effective than the other techniques, as long as straightforward on implementation concerns

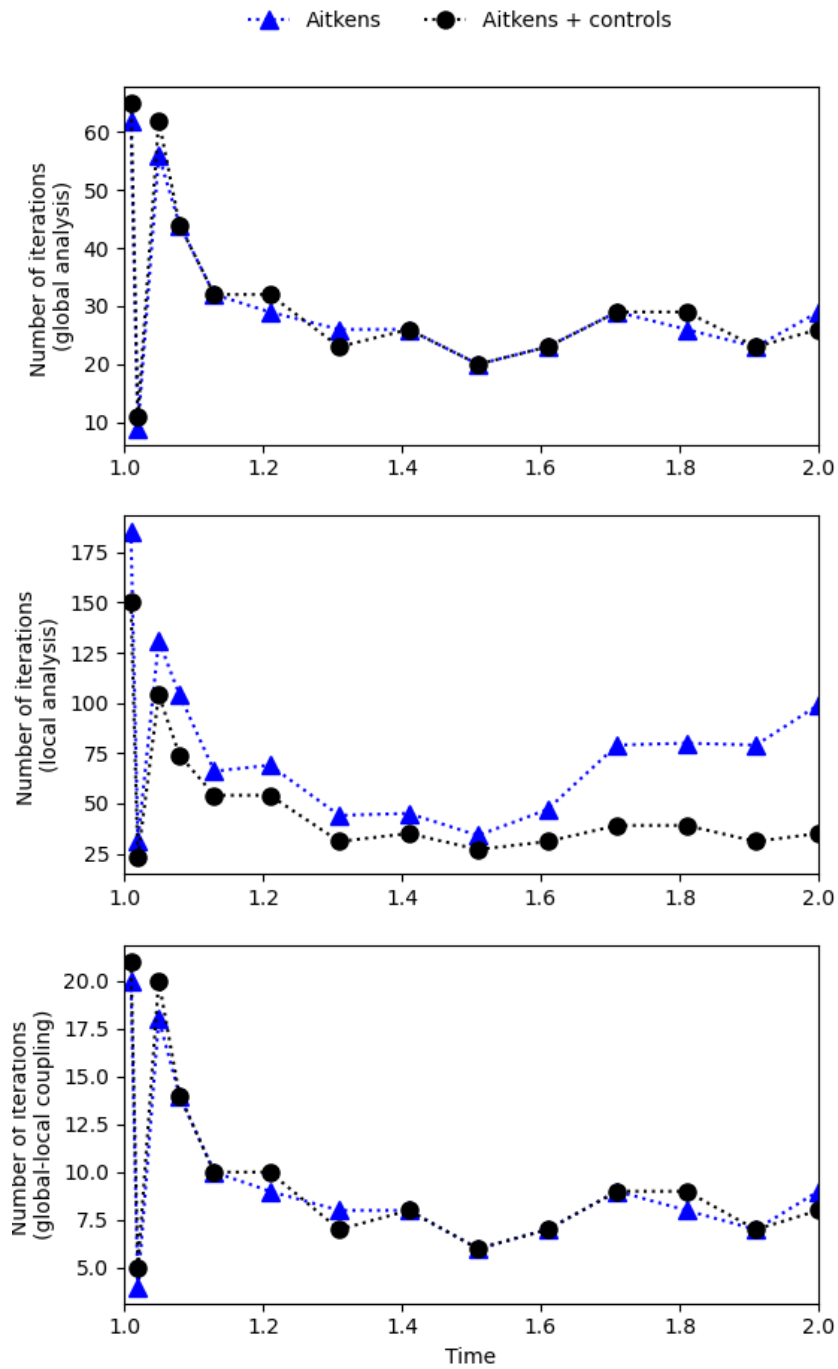


Figure 17: Comparison of the number of iterations in time between the GLIC-Aitken and GLIC-IS-Aitken simulations for the bolted joint use case.

The coupling reached acceptable levels of convergence with approximately 10 iterations, in line with the most challenging multi-physics coupling strategies. Furthermore, the Authors could prove the employment of inexact solver strategy (GLIC-IS) more effective, with negligible loss of accuracy for engineering purposes, improving performance leveraging loosened tolerance values on the Newton-Raphson iteration schemes of the global and local analyses.

As the mitigation of the robustness and performance issues of the GLIC improves the appeal of the methodology for an effective engineering practice, the full promise of non-intrusive modeling is not fully met yet: in the presented works, the authors needed to manufacture local models with solid-to-shell coupling in order to align global and local interfaces to the facets of global elements, in order to avoid to cut through global element with the local patch. This issue was successfully tackled with a re-meshing strategy in the local model by [10], translating modeling intrusivity to implementation intrusivity. The authors would also like to explore new opportunities in future works such as the employment of virtual elements [47] or enhanced elements [36] where global elements are cut by the coupling interface.

Acknowledgements

The Authors thanks Albert Kurkchubasche and Sethu Subbarayalu for their precious help on the implementation tasks within the Abaqus co-simulation tools.

References

- [1] *Abaqus User Documentation, Johnston RI, Dassault Systèmes SIMULIA, 2023, 2023.*
- [2] Margarita Akterskaia, Eelco Jansen, Stephen R. Hallett, Paul Weaver, and Raimund Rolfes. Analysis of skin-stringer debonding in composite panels through a two-way global-local method. *Composite Structures*, 202:1280–1294, 2018.
- [3] Fadi Aldakheel, Nima Noii, Thomas Wick, Olivier Allix, and Peter Wriggers. Multilevel global-local techniques for adaptive ductile phase-field fracture. *ArXiv*, abs/2103.02377, 2021.
- [4] Fadi Aldakheel, Nima Noii, Thomas Wick, and Peter Wriggers. A global-local approach for hydraulic phase-field fracture in poroelastic media. *Comput. Math. Appl.*, 91:99-121, 2021.
- [5] Olivier Allix, Maxime Blanchard, and Pierre Gosselet. *Current Trends and Open Problems in Computational Mechanics*, chapter A non-intrusive Global/Local Cycle-Jumping Techniques: Application to Visco-Plastic Structures, pages 23–30. Springer International Publishing, Cham, 2022.
- [6] Donald G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the Association for Computing Machinery*, 12(4):547–560, 1965.
- [7] O. Bettinotti, O. Allix, and B. Malherbe. A coupling strategy for adaptive local refinement in space and time with a fixed global model in explicit dynamics. *Computational Mechanics*, 53:561-574, 2014.
- [8] O. Bettinotti, O. Allix, U. Perego, V. Oancea, and B. Malherbe. A fast weakly intrusive multiscale method in explicit dynamics. *International Journal for Numerical Methods in Engineering*, 100(8):577-595, 2014.

- [9] Omar Bettinotti, Olivier Allix, Umberto Perego, Victor Oancea, and Benoît Malherbe. Simulation of delamination under impact using a global–local method in explicit dynamics. *Finite Elements in Analysis and Design*, 125:1-13, 2017.
- [10] Maxime Blanchard, Olivier Allix, Pierre Gosselet, and Geoffrey Desmeure. Space-time global-local noninvasive coupling strategy: Application to viscoplastic structures. *Finite Elements in Analysis and Design*, 156:1-12, 2019.
- [11] A. Combescure and A. Gravouil. A numerical scheme to couple subdomains with different time-steps for predominantly transient linear analysis. *Computer Methods in Applied Mechanics and Engineering*, 191(1112):1129-1157, 2002.
- [12] F. Confalonieri, A. Corigliano, M. Dossi, and M. Gornati. A domain decomposition technique applied to the solution of the coupled electro-mechanics problem. *International Journal for Numerical Methods in Engineering*, 93(2):137-159, 2012.
- [13] Joris Degroote, Klaus-Jürgen Bathe, and Jan Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. *Computers & Structures*, 87(11):793-801, 2009.
- [14] Joris Degroote, Robby Haelterman, Sebastiaan Annerel, Peter Bruggeman, and Jan Vierendeels. Performance of partitioned procedures in fluid–structure interaction. *Computers & Structures*, 88(7):446–457, 2010.
- [15] Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2):400-408, 1982.
- [16] Mickaël Duval, Jean-Charles Passieux, Michel Salaün, and Stéphane Guinard. Non-intrusive coupling: recent advances and scalable nonlinear domain decomposition. *Archives of Computational Methods in Engineering*, 23(1):17–38, March 2016.
- [17] Stanley C Eisenstat and Homer F Walker. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996.
- [18] Haw-ren Fang and Yousef Saad. Two classes of multiseant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, 2009.
- [19] C. Farhat, L. Crivelli, and F.-X. Roux. A transient FETI methodology for large-scale parallel implicit computations in structural mechanics. *International Journal for Numerical Methods in Engineering*, 37(11):1945-1975, 1994.
- [20] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205-1227, 1991.
- [21] V. Faucher and A. Combescure. A time and space mortar method for coupling linear modal subdomains and non-linear subdomains in explicit structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 192:509-533, 2003.
- [22] J. Fish. The s-version of the finite element method. *Computers and Structures*, 43:539-547, 1992.
- [23] J. Fish and V. Belsky. Multigrid method for periodic heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 126:1-38, 1995.

- [24] J. Fish, V. Belsky, and M. Pandheeradi. Composite grid method for hybrid systems. *Computer Methods in Applied Mechanics and Engineering*, 135:307-325, 1996.
- [25] J. Fish, A. Suvorov, and V. Belsky. Hierarchical composite grid method for global-local analysis of laminated composite shells. *Applied Numerical Mathematics*, 23:241-258, 1997.
- [26] L. Gendre, O. Allix, and P. Gosselet. A two-scale approximation of the Schur complement and its use for non-intrusive coupling. *International Journal for Numerical Methods in Engineering*, 87:889-905, 2011.
- [27] L. Gendre, O. Allix, P. Gosselet, and F. Comte. Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Computational Mechanics*, pages 233-245, 2009.
- [28] Tymofiy Gerasimov, Nima Noii, Olivier Allix, and Laura De Lorenzis. A non-intrusive global/local approach applied to phase-field modeling of brittle fracture. *Advanced Modeling and Simulation in Engineering Sciences*, 5, 2018.
- [29] R. Glowinski and P. LeTallec. Augmented lagrangian interpretation of the non-overlapping Schwartz alternating method. *International Symposium on Domain Decomposition Methods, SIAM, Philadelphia*, pages 224-231, 1990.
- [30] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational methods in Engineering*, 13(5):515-572, 2006.
- [31] Pierre Gosselet, Maxime Blanchard, Olivier Allix, and Guillaume Guguin. Non-invasive global-local coupling as a Schwarz domain decomposition method: acceleration and generalization. *Advanced Modeling and Simulation in Engineering Sciences*, 5:1-23, 2018.
- [32] A. Gravouil and A. Combescure. Multi-time-step explicit-implicit method for nonlinear structural dynamics. *International Journal for Numerical Methods in Engineering*, 50:199-225, 2001.
- [33] A. Gravouil and A. Combescure. Multi-time-step and two-scale domain decomposition method for nonlinear structural dynamics. *International Journal for Numerical Methods in Engineering*, 58(10):1545-1569, 2003.
- [34] G. Guguin, O. Allix, P. Gosselet, and S. Guinard. Nonintrusive coupling of 3D and 2D laminated composite models based on finite element 3d recovery. *International Journal for Numerical Methods in Engineering*, 98(5):324-343, 2014.
- [35] Guillaume Guguin, Olivier Allix, Pierre Gosselet, and Stéphane Guinard. On the computation of plate assemblies using realistic 3d joint model: a non-intrusive approach. *Advanced Modeling and Simulation in Engineering Sciences*, 3:1-18, 2016.
- [36] Paul Hauseux, Emmanuel Roubin, and Jean-Baptiste Colliat. The embedded finite element method (E-FEM) for multicroacking of quasi-brittle materials. In *Porous Rock Fracture Mechanics with Application to Hydraulic Fracturing, Drilling and Structural Engineering*, pages 177-196. Elsevier, 2017.
- [37] H. Li, J. Avecillas-Leon, N. Shauer, and C.A. Duarte. A non-intrusive iterative generalized finite element method for multiscale coupling of 3-d solid and shell models. *Computer Methods in Applied Mechanics and Engineering*, page 115408, 2022.

- [38] H. Li, P. O’Hara, and C.A. Duarte. Non-intrusive coupling of a 3-d generalized finite element method and abaqus for the multiscale analysis of localized defects and structural features. *Finite Elements in Analysis and Design*, 193:103554, 2021.
- [39] P. Lions. On the Schwarz alternating method iii. a variant for non-overlapping sub-domains. *Proceedings of Domain Decomposition Methods for Partial Differential Equations*, pages 202-223, 1990.
- [40] J. Mandel. Balancing domain decomposition. *International Journal for Numerical Methods in Biomedical Engineering*, 9(3):233-241, 1993.
- [41] Nima Noii, Fadi Aldakheel, Thomas Wick, and Peter Wriggers. An adaptive global–local approach for phase-field modeling of anisotropic brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 2020.
- [42] Anthony Nouy and Florent Pled. A multiscale method for semi-linear elliptic equations with localized uncertainties and non-linearities. *ESAIM: M2AN*, 52(5):1763-1802, 2018.
- [43] J. Plews, C. A. Duarte, and T. Eason. An improved nonintrusive global-local approach for sharp thermal gradients in a standard FEA platform. *International Journal for Numerical Methods in Engineering*, 91(4):361-397, 2012.
- [44] Klaudius Scheufele and Miriam Mehl. Robust multiseccant quasi-newton variants for parallel fluid-structure simulations—and other multiphysics applications. *SIAM Journal on Scientific Computing*, 39(5):404–433, 2017.
- [45] Benjamin Walter Uekermann. *Partitioned Fluid-Structure Interaction on Massively Parallel Systems*. Phd thesis, Technische Universität München, 2016.
- [46] J. D. Whitcomb. Iterative global/local finite element analysis. *Computers and Structures*, 40(4):1027-1031, 1991.
- [47] Peter Wriggers and Blaz Hudobivnik. Virtual element formulation for gradient elasticity. *Acta Mechanica Sinica*, 39:722306, 2023.