

Non-Intrusive Coupling of Neural Network-Based Local Models to Explicit Dynamics Solvers

A. Pulikkathodi¹, E. Lacazedieu², L. Chamoin^{1,3}, J.P.B. Ramirez⁴, L. Rota⁵, M. Zarroug⁵, A. Zammouri⁶

¹ Université Paris-Saclay, Centrale Supélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de Mécanique Paris-Saclay, {afsal.pulikkathodi, ludovic.chamoin}@ens-paris-saclay.fr

² LAMIH, elisabeth.lacazedieu@uphf.fr

³ IUF, Institut Universitaire de France

⁴ Altair Engineering France, jpbramirez@altair.com

⁵ Stellantis, {laurent.rota,malek.zarroug}@stellantis.com

⁶ EPF School of Engineering, amin.zammouri@epf.fr

Abstract —

In solving large structural problems with multiple complex localized behaviors, current industry practices rely on highly simplified Finite Element Method (FEM) models, leading to inaccurate failure predictions. Various domain decomposition techniques have emerged to tackle these challenges, with non-intrusive coupling methods standing out in industrial applications due to their flexibility. Despite their advantages, these methods typically require numerous iterations between local and global problems, posing significant computational challenges. To address this issue, the article proposes substituting the FEM-based local problem with a Neural Network-based Reduced Order Model (ROM). The work focuses on non-intrusive coupling, integrating the ROM with an explicit dynamic solver, resulting in significant reduction in computational time. The effectiveness of the proposed method is shown through two numerical examples, each increasing in complexity, to showcase the methods adaptability and scalability.

Keywords — Local/Global Coupling, Neural Network-based Model Order Reduction, Transient dynamics, Spotwelds simulation.

1 Introduction

Crash-worthiness analysis of full vehicle containing multiple complex localised features like spot welds and bolted joints, poses significant challenges. This is mainly due to the requirement of fine mesh for representing local feature and its associated small time step to meet Courant-Friedrichs-Lewy (CFL) condition in explicit method. Due to this reason, at present in industries the full vehicle FEM models are highly simplified, leading to inaccurate failure predictions.

Over the last couple of decades, various methods based on domain decomposition techniques, such as the primal BDD method, dual FETI method, and the mixed Latin scheme [1], have emerged to solve these challenges. In this approach, the refined localized features, referred to as local problem, and the simplified large-scale structures, known as the global problem, are solved separately in their respective time steps. Among these methods, non-intrusive coupling [2, 3, 4] stands out due to its ability to connect a simplified global model to a detailed local one, using features readily available in commercial software. However, a significant drawback of this non-intrusive coupling is the need for a large number of iterations between local and global problems. To reduce this iteration count, various acceleration techniques are also implemented in similar works.

When it comes to non-intrusive coupling in transient dynamics, [5] proposed a method where the global computation occurs only once per global time step, while the local problem is solved multiple times within each global time step. Building upon this foundation, our work introduces a novel concept: replacing the FEM-based local problem with a Neural Network-based Reduced Order Model (ROM). This innovation significantly reduces the overall computational time by reducing the cost of the local problem in the online phase.

The work comprises two main components: the development of a data-driven ROM for the local

problem and the formulation of a non-intrusive local/global coupling method to integrate the ROM with a transient dynamic solver. This article focuses mainly on the latter aspect, focusing on our novel approach. For a detailed study on the ROM development, readers are referred to [6]. The ROM uses a Physics-Guided Architecture of Neural Networks (PGANN) [7] to map interface velocities to interface reaction forces, incorporating additional relevant physical variables into the neural network architecture.

The paper is organised as follows. In Section 2, we explain our research method starting with the non-intrusive local/global coupling development, where we discuss how Neural Networks and global models are integrated. In Section 3, we demonstrate our methodology through two numerical examples: an academic 2D elastic model with localized stiffness variations, and an industrial 3D case involving spot welded plates, implemented in an industrial software. Finally, in the Section 4, we summarize the key findings, discuss the implications of our novel approach, and outline ideas for future research.

2 Non-intrusive coupling and model reduction in transient dynamics

2.1 Problem setup

In this study, we briefly present a method for addressing the non-intrusive local/global coupling in explicit dynamics. To illustrate the method, we consider the problem configuration depicted in Figure 1a. As shown in the figure, the overall domain, denoted as Ω , is partitioned into two subdomains: the local region, denoted as Ω_l , and the complementary region, denoted as Ω_c . The local region Ω_l may contain fine geometric features or large gradients, which requires a refined mesh in both space and time scales. Conversely, the complementary region Ω_c only requires a coarse mesh in both space and time. The interface between the two regions is denoted as Γ . In this problem, the displacement u_D is prescribed on the Dirichlet boundary $\partial\Omega_u$, the external force f_{ext} is applied on the Neumann boundary $\partial\Omega_f$, and the body force f_Ω may be applied in the domain Ω .

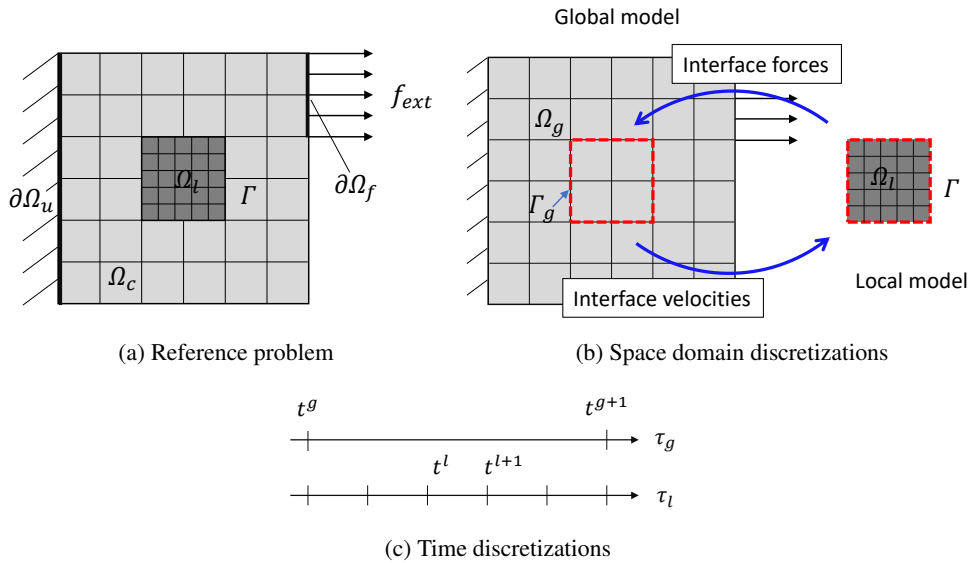


Figure 1: An illustration of the non-intrusive local/global coupling method.

Using a displacement-based finite element method with the heterogeneous spatial discretization described in Figure 1b (left), the problem can be expressed as follows.

$$\begin{aligned}
 \mathbf{M}\ddot{\mathbf{U}} &= \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}} & \text{over } & \Omega_c \cup \Omega_l \times [t_0, t_{\text{end}}] \\
 \mathbf{U} &= \bar{\mathbf{U}} & \text{along } & \partial\Omega_u \times [t_0, t_{\text{end}}] \\
 \{\mathbf{U}, \dot{\mathbf{U}}\} &= \{\mathbf{U}_0, \mathbf{V}_0\} & \text{over } & \Omega_c \cup \Omega_l|_{t_0}
 \end{aligned} \tag{1}$$

where \mathbf{M} is the lumped mass matrix, \mathbf{F}^{ext} and \mathbf{F}^{int} are, respectively, the external and internal force vectors, and $\Omega_c \cup \Omega_l$ represents the union of two different homogeneous finite element discretizations

that span the entire domain. \mathbf{U} represents the vector containing the nodal displacement ($\ddot{\mathbf{U}}$ and $\dot{\mathbf{U}}$ are the associated acceleration and velocity, respectively). The initial and final times are denoted t_0 and t_{end} .

2.2 Non-intrusive local/global coupling

In a non-intrusive local/global coupling framework, the global model extends over the whole structure with a global mesh and never changes (Figure 1b). The local analysis is carried out with a more refined mesh where the boundary conditions are derived from the global problem. Two different time steps Δt_g and Δt_l are applied in the two partitions Ω_c and Ω_l (Figure 1c). The additional nodal forces \mathbf{r} required to make global problem identical to that of reference problem is given by the local problem. The expression for \mathbf{r} can be written as:

$$\mathbf{r}_\Gamma = \mathbf{r}_{l,\Gamma} - \mathbf{r}_{c,\Gamma} \quad (2)$$

where :

$$\begin{aligned} \mathbf{r}_{l,\Gamma} &= \mathbf{F}_{l,\Gamma}^{\text{ext}} - \mathbf{M}_{l,\Gamma} \ddot{\mathbf{U}}_{l,\Gamma} - \mathbf{F}_{l,\Gamma}^{\text{int}} \\ \mathbf{r}_{c,\Gamma} &= \mathbf{F}_{c,\Gamma}^{\text{ext}} - \mathbf{M}_{c,\Gamma} \ddot{\mathbf{U}}_{c,\Gamma} - \mathbf{F}_{c,\Gamma}^{\text{int}} \end{aligned} \quad (3)$$

The global problem is then re-solved by applying the reaction forces from the local problem at the interface. This process is repeated until the difference in reaction forces e reaches a predefined tolerance \bar{e} . The quantity e is defined as follows:

$$e = \|\mathbf{r}_{l,\Gamma} + \mathbf{r}_{c,\Gamma}\| \quad (4)$$

Algorithm 1: Local/Global Coupling in Explicit Dynamic Simulation

```

Data: Initial conditions, time step, total simulation time,  $\bar{e}$ 
while  $t_g < t_{end}$  do
    // Solve the global problem
     $\mathbf{M}_g \ddot{\mathbf{U}}_g = \mathbf{F}_g^{\text{ext}} - \mathbf{F}_g^{\text{int}}$ ;
    while  $e > \bar{e}$  do
        // Local computation using interface velocities of global problem
         $\mathbf{M}_l \ddot{\mathbf{U}}_l = \mathbf{F}_l^{\text{ext}} - \mathbf{F}_l^{\text{int}}$ ;
        // Compute residual and global correction acceleration
         $\mathbf{r}_\Gamma = \mathbf{r}_{l,\Gamma} - \mathbf{r}_{c,\Gamma} = (\mathbf{F}_{l,\Gamma}^{\text{ext}} - \mathbf{M}_{l,\Gamma} \ddot{\mathbf{U}}_{l,\Gamma} - \mathbf{F}_{l,\Gamma}^{\text{int}}) - (\mathbf{F}_{c,\Gamma}^{\text{ext}} - \mathbf{M}_{c,\Gamma} \ddot{\mathbf{U}}_{c,\Gamma} - \mathbf{F}_{c,\Gamma}^{\text{int}})$ 
         $\mathbf{M}_{g,\Gamma} \ddot{\mathbf{U}}_{g,\Gamma}^{\text{corr}} = \mathbf{r}_\Gamma$ ;
        // Update global acceleration at the interface
         $\ddot{\mathbf{U}}_{g,\Gamma} = \ddot{\mathbf{U}}_{g,\Gamma} + \ddot{\mathbf{U}}_{g,\Gamma}^{\text{corr}}$ 
        Compute  $e$  ;
    end
    // Update time
     $t_g = t_g + \Delta t_g$  ;
end

```

2.3 Neural Network-based local model

In this section, we briefly discuss the proposed NN model. In non-intrusive local/global coupling, the local problem takes u_Γ, v_Γ as boundary conditions and the reaction force $\mathbf{r}_\Gamma = \mathbf{r}_{l,\Gamma} - \mathbf{r}_{c,\Gamma}$ is computed. Due to the lack of a robust input-output relationship, it is challenging to train a NN only using input and output. This is primarily due to their noisy nature and dependence on other model parameters, such as material parameters and geometric parameters. In order to address this issue, a new layer containing informations such as displacement u_l , total plastic strain $\epsilon_{p,l}$ of the local problem is inserted between the input and output layers.

The proposed NN architecture corresponds to the local problem is displayed in Figure 2. It takes the displacement u_Γ , velocity v_Γ , and rotational velocity ω_Γ at the interfaces as inputs to predict the corresponding reaction force r_Γ and reaction moment m_Γ at the interfaces. The terms ω_Γ and m_Γ are relevant only for 2D problems with rotational degrees of freedom activated. The NN architecture starts

with the reconstruction of the local domain displacement u_l from these inputs. Subsequently, an Autoencoder is employed to extract the most critical features of u_l into a Latent Vector (LV). In parallel, another Autoencoder is used for extracting most significant aspects of the effective total plastic strain, $\varepsilon_{p,l}$, and encoding them into its respective LVs. The evolution of these LVs is modeled through a Long Short-Term Memory (LSTM) network. Finally, the predicted displacement at the next global time step is mapped into interface reaction forces and moments.

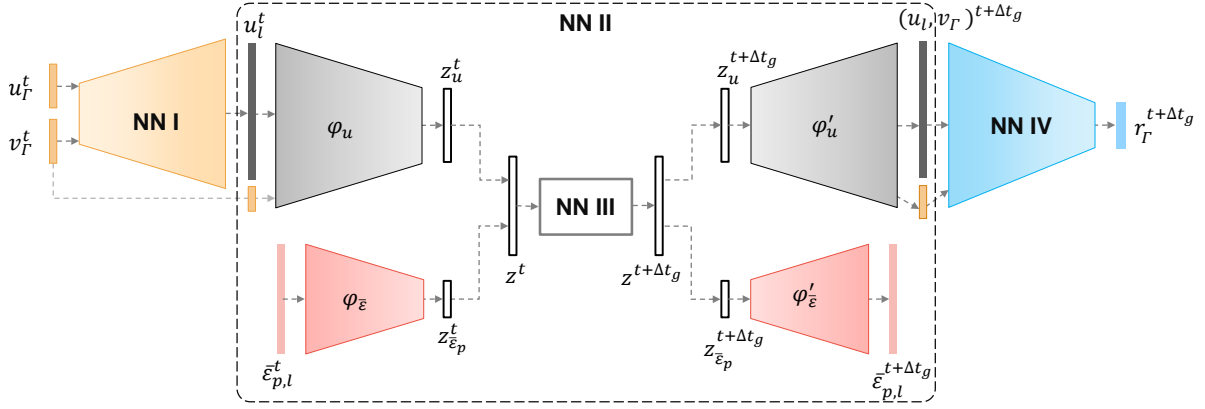


Figure 2: Schematic representation of the proposed NN architecture.

3 Numerical examples

The methodology outlined in the previous subsection is implemented in two distinct scenarios: firstly, on a simple academic 2D elastic example, where a specific local area exhibits a different level of stiffness compared to the global region. Secondly, it is applied to a 3D example using industrial software, characterized by elasto-plastic material properties, with 3D solid elements in the local domain and shell elements in the global problem.

3.1 Academic case : 2D structure

3.1.1 Data generation

Let us consider an elastic rectangular heterogeneous structure as shown in Figure 3a. The left half of the domain is taken as local region with Young modulus E_L . Similarly the right half is taken as global complementary region with Young modulus E_G . The structure is clamped on its left side $\partial\Omega_L$. A quadratic monotonic tensile load $f_{ext}(t)$ varying with time is applied on its right edge $\partial\Omega_R$. Both the local and global regions are discretized using a mesh of 4-node quadrilateral elements of identical size.

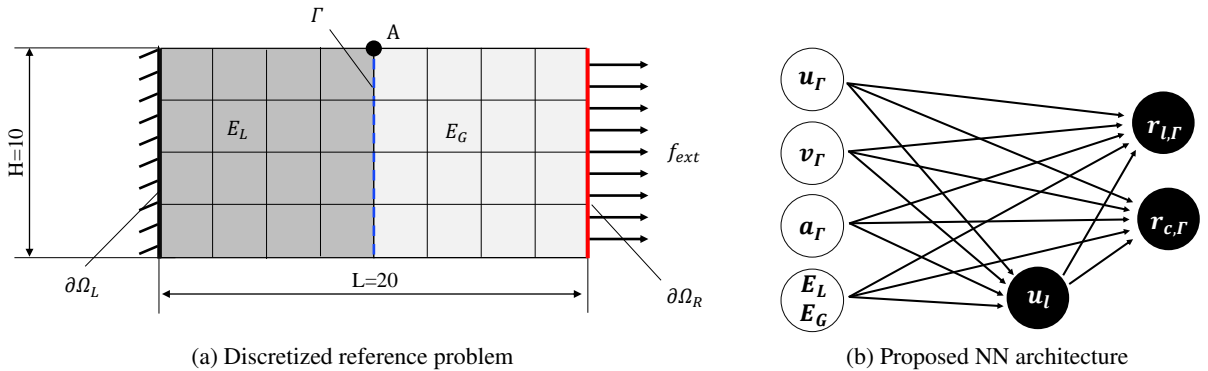


Figure 3: (a) Discretized reference problem. (b) Schematic architecture of the proposed Neural Network. Inputs are denoted by white circles, while outputs are represented by black circles. Arrows are used to denote neural network links.

The fixed parameters of the problem are : density $\rho = 7800 \text{ kg/m}^3$; Poisson's ratio $\nu = 0.3$; length of domain $L = 20 \text{ m}$; and height of domain $H = 10 \text{ m}$. The simulation progresses from an initial time

$t_0 = 0$ s to a final time $t_{\text{end}} = 0.01$ s. The time step Δt is chosen in accordance with the estimation of the critical time step using CFL condition. For the sake of simplicity a constant time stepping is used, so that the time interval $[t_0, t_{\text{end}}]$ is partitioned in N time steps Δt of the same size, with $t_{\text{end}} - t_0 = N\Delta t$. Here the value computed is $\Delta t = 5 \times 10^{-5}$, which makes $N = 200$.

For creating ROM of the local problem, snapshots are created by varying following parameters: Young moduli E_L and E_G , and the maximum value of f_{ext} at final time t_{end} . It is assumed here that the value of E_G is always greater than the value of E_L . The range and increment of the parametric spaces \mathcal{S}_{E_G} , \mathcal{S}_{E_L} and $\mathcal{S}_{f_{\text{ext}}^{\text{max}}}$ are shown in the Table 1. In total, 364 simulations are generated using explicit dynamic solver by combining these parameter values. From each simulation result, the following quantities are saved at each time step : the displacement of local domain $u_l = (u_{l,x}, u_{l,y})$, the interface displacement $u_\Gamma = (u_{\Gamma,x}, u_{\Gamma,y})$, interface velocity $v_\Gamma = (v_{\Gamma,x}, v_{\Gamma,y})$, interface acceleration $a_\Gamma = (a_{\Gamma,x}, a_{\Gamma,y})$, and the interface reaction force of local domain $\mathbf{r}_{l,\Gamma}$. To generate data of interface reaction force of global complementary problem $\mathbf{r}_{c,\Gamma}$, another set of 36 simulations are performed with $E_L = E_G$.

Parameter	Range	Increment
E_L	[200 500] GPa	25
E_G	[200 500] GPa	25
$f_{\text{ext}}^{\text{max}}$	{10, 30, 60, 90} GN	-

Table 1: Parameters of the problem.

3.1.2 NN-based reduced model

In this section, we shall elaborate on the proposed NN model. The inputs of proposed NN are the boundary conditions and values of Youngs Moduli E_L and E_G ; the outputs are $\mathbf{r}_{l,\Gamma}$ and $\mathbf{r}_{c,\Gamma}$. A schematic representation of proposed NN architecture is shown in Figure 3b. Introducing such additional variables does not have significant effect in this example, but it is significant in case of complex problems, such as in the subsequent 3D case study presented.

The proposed architecture contains two sub networks, the first network corresponds to a forward problem. It takes $u_\Gamma, v_\Gamma, a_\Gamma, E_L, E_G$ as input and predict its corresponding displacement of local domain u_l . The second sub network takes this predicted displacement and inputs of first sub network as inputs and predicts $\mathbf{r}_{l,\Gamma}$ and $\mathbf{r}_{c,\Gamma}$. The main hyper-parameters of the NN to tune include : the number of layers, the number of neurons, the choice of activation functions in both sub networks. The loss function type, the learning rate, and the batch size are other general parameters to optimize. The procedure used to tune these hyper parameters is detailed in [6]. The input-output pairs correspond to this architecture are generated using the data described in previous section. Each input-output pair undergoes scaling and shuffling to optimize training performance. The complete dataset is then partitioned into a training dataset, comprising 80% of the total data, and a validation dataset, consisting of the remaining 20%. Tensorflow library of Python is used for training the NN.

The trained NN contains 4 dense layers in each sub network with 40-100 neurons in each layer. After 10,000 epochs, the Normalized Mean Absolute Error (NMAE) of the validation dataset is found to be 8.5×10^{-5} . To assess the accuracy of the trained NN model on unseen parameters, two new simulations are generated with previously unseen parameters, as shown in Table 2. It is evident that the NN performs well in predicting unseen parameters.

Case	E_g (MPa)	E_l (MPa)	f_{ext} (N)	NMAE
I	480	390	5×10^{10}	9.1×10^{-5}
II	385	265	2×10^{10}	8.2×10^{-5}

Table 2: NMAE on NN training for various test cases.

3.1.3 Non-intrusive coupling of global model and reduced local model

The next step is to replace the local problem in Algorithm 1 with the previously developed NN-based local model. For that, at each time step, the global problem is first computed with homogeneous domain with Young's modulus E_G , then the NN is evaluated using the boundary extracted from global problem. Finally the reaction force given by NN is used to update the acceleration in global interface nodes. The effect of these updates is taken into account in the next time step of global problem.

For validating the accuracy of developed method, we choose same test cases shown in the Table 1. Figure 4 shows the velocity at point A using the NN-coupled model and the reference model for the two test cases mentioned in Table 2. The velocity at point A displayed in Figure 3a of the NN-coupled model closely matches with its corresponding reference solution. Figure 5 provides a comparison of displacement in x and y direction between the NN-coupled solution and the reference solution for case I, further affirming the model accuracy.

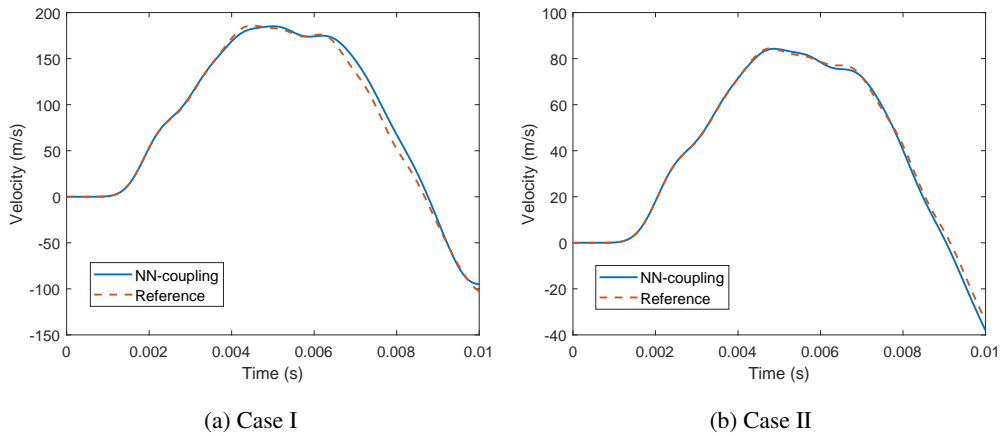


Figure 4: Comparison of velocity at point A between the NN-coupled model and the Reference model for two test cases from Table 2.

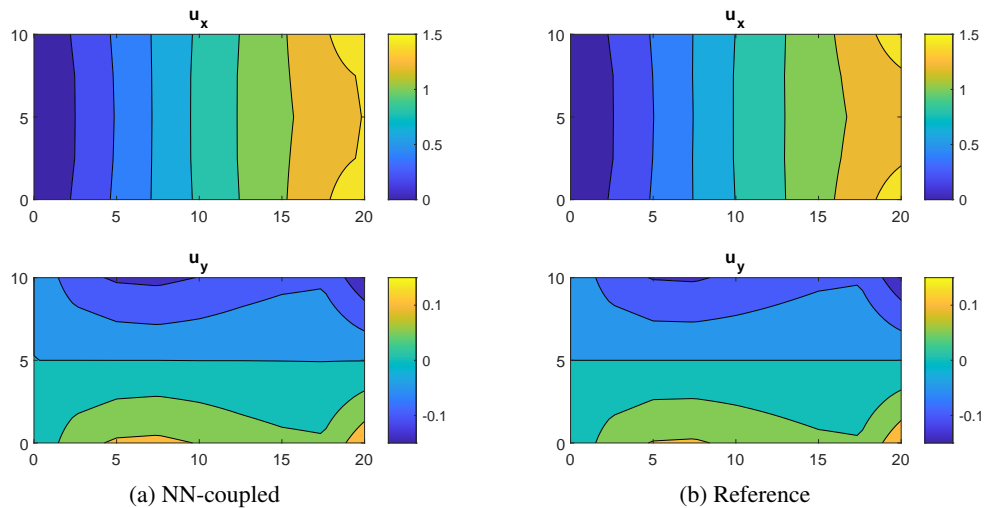


Figure 5: Comparison of displacement between the NN-coupled model and the reference solution for case I.

3.2 Industrial case : Spotwelded plates

In this section, we extend the application of previously discussed techniques to a practical case involving spotwelded plates with 3D elements. The reference model, depicted in Figure 6a, uses 3D elements for modeling the region near the spotweld, while the remaining structure is represented with 2D shell elements. The simplified version of this model is shown in Figure 6b, where 2D shell elements are used

to represent the plates and a 1D spring element for the spotweld. This simplified model is considered as the global problem within the non-intrusive coupling framework.

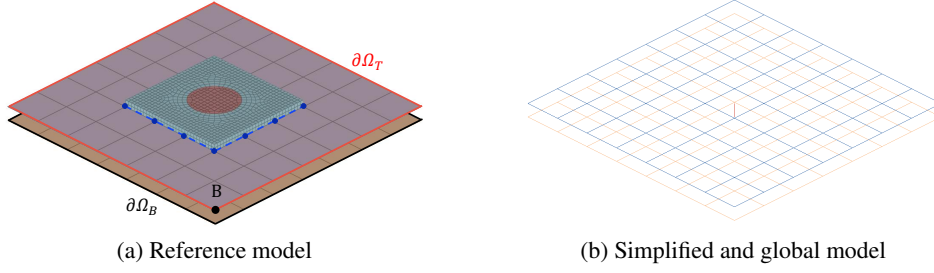


Figure 6: The reference and simplified model used in spotwelded plate example.

The data required for training of the local problem is generated by solving the reference problem. The simulation setup involves clamping the outer edges of the bottom plates $\partial\Omega_B$ and applying a load on the outer edges of the top plate $\partial\Omega_T$ over a duration of 1.5 ms. The time step size is determined using the CFL condition, with the reference and simplified models having time steps of $\Delta t_{ref} = 5 \times 10^{-5}$ and $\Delta t_g = 5 \times 10^{-4}$, respectively. The material is characterized by a density of $\rho = 7800 \text{ kg/m}^3$. An isotropic elastoplastic material model, specifically the Johnson-Cook material model, is used to describe the material behavior, ignoring the effects of temperature or strain rate. The stress expression is defined as:

$$\sigma = a + b\varepsilon_p^n \quad (5)$$

where ε_p denotes the plastic strain. For the purposes of simulations, the values are set as $a = 0.792 \text{ GPa}$, $b = 0.51 \text{ GPa}$, and $n = 0.26$. As in the previous example, both the magnitude and direction of the applied loads are parameterized to generate training dataset, resulting a total of 60 simulations. The details on NN training methods used for this example are not discussed in this article. Interested readers are encouraged to refer to our earlier publication [6].

The main difference in this example compared to the previous one lies in the implementation of coupling. Here, the trained NN model is coupled with an industrial solver, OpenRadioss. Its UserWindow feature is used to facilitate communication between the NN and the global model. The coupling process is executed in three stages within each cycle. First, the UserWindow extracts the required input for NN at predefined interface nodes. Following this, it evaluates the trained NN model with this input to obtain the output interface forces and moments. Finally, the UserWindow transfers these calculated forces and moments to the global model.

For validating the accuracy of developed method, a new test case is generated. In this test case, a monotonic linear force applied on $\partial\Omega_T$ in the direction z , simultaneously a monotonic linear moment is applied on $\partial\Omega_T$ along the axis z , with maximum values $20kN$ and $100N.m$ respectively. Figure 7 shows the velocity at point B of the simplified global model, the reference model and NN-coupled model (simplified global model with NN enrichment) of the two test case. The velocity at point B of the NN-coupled model closely matches with its corresponding reference solution. Figure 8 shows a comparison of resultant displacement between the NN-coupled solution and the reference solution for case I, further affirming the model accuracy.

4 Conclusion

This study presented a novel method for solving large structural problems with complex localized behaviors, employing non-intrusive local/global coupling in explicit dynamics. The method integrates a reduced model of the localized feature, developed using a Physics-Guided Architecture of Neural Networks (PGANN), with an Explicit solver. This integration is achieved through a non-intrusive coupling strategy, aiming to overcome challenges posed by fine mesh requirements and small time steps. The key innovation lies in replacing the Finite Element Model (FEM) at the local scale with a Neural Network-based Reduced Order Model (ROM). This replacement significantly reduces computational time required for multiple iteration required for complex phenomena.

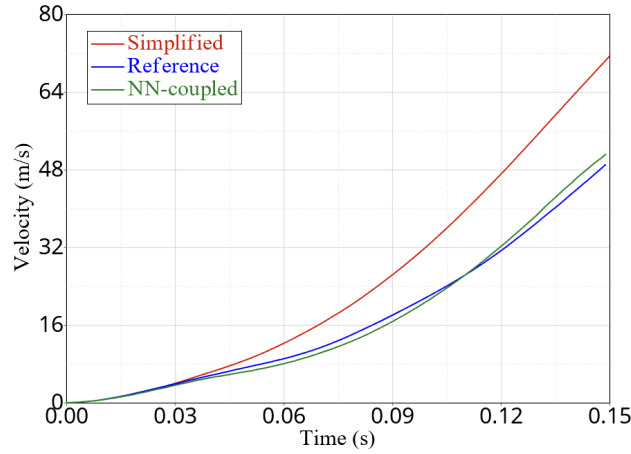


Figure 7: Comparison of velocity at point B between the NN-coupled model and the Reference model.

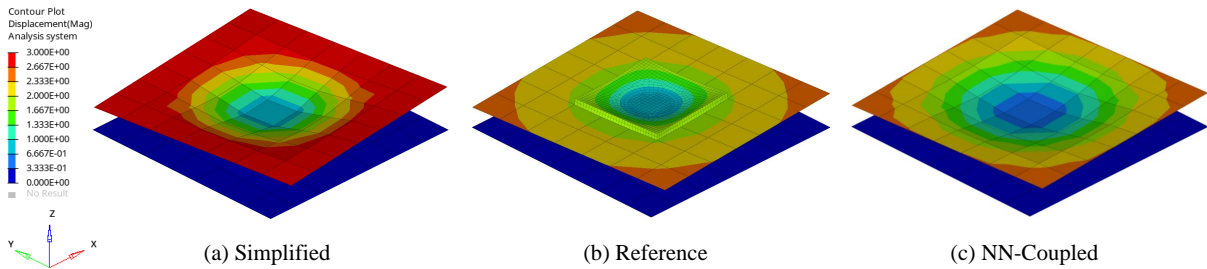


Figure 8: Comparison of displacement between the Simplified model, Reference and NN-coupled model.

An academic and industrial example demonstrates the idea of the proposed method. The Neural Network is trained to predict interface reaction forces based on input parameters, and the trained model is used in the non-intrusive coupling strategy, showing good agreement with the reference solution.

Future work will further validate and extend the proposed method to more complex examples, particularly coupling Neural Networks with commercial software for applications like plates with multiple spotwelds.

References

- [1] P. Ladevèze, O. Loiseau, and D. Dureisseix. A micro–macro and parallel computational strategy for highly heterogeneous structures. *Int. J. Numer. Meth. Engng.*, 52: 121-138 (2001).
- [2] L. Gendre, O. Allix, and P. Gosselet. Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Comput Mech.*, 44: 233–245 (2009).
- [3] G. Guguin, O. Allix, P. Gosselet, On the computation of plate assemblies using realistic 3D joint model: a non-intrusive approach. *Adv. Model. and Simul. in Eng. Sci.*, 3: (2016).
- [4] T. Chantrait, J. Rannou, A. Gravouil. Low intrusive coupling of implicit and explicit time integration schemes for structural dynamics: Application to low energy impacts on composite structures, *Finite Elements in Analysis and Design*, 86: 23-33 (2014).
- [5] O. Bettinotti, O. Allix, U. Perego, V. Oancea, B. Malherbe. A fast weakly intrusive multiscale method in explicit dynamics. *International Journal for Numerical Methods in Engineering* 100 (8), 577-595 (2014).
- [6] A. Pulikkathodi, E. Lacazedieu, L. Chamoin, J. P. Berro Ramirez, L. Rota, and M. Zarroug, A Neural Network-Based Data-Driven local modeling of spotwelded plates under impact, *Mechanics & Industry* 24, 34 (2023).
- [7] JS. Read, X. Jia, J. Willard, AP. Appling, et al. Process-guided deep learning predictions of lake water temperature. *Water Resources Research* (2019).