POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Approximation of multiscale problems using domain decomposition methods and model order reduction

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Viola Brenzone

Advisor: Paolo Zunino

Co-advisors: Nunzio Dimola, Piermario Vitullo

Academic year: 2023-2024

1. Introduction

Multiscale problems are challenging because they involve phenomena occurring at multiple spatial and/or temporal scales. These problems are prevalent in various scientific and engineering fields, where processes at the microscopic level influence and interact with macroscopic behavior. A multiscale problem is characterized by significant interactions across different scales. This study tackles the main challenge of numerically solving multiscale problems, which arises from the complexity involved in discretizing, assembling, and solving the multiscale discrete operator. To address these challenges, this study combines domain decomposition techniques with model order reduction methods, providing an effective solution to this computational hurdle.

The theoretical foundation is based on domain decomposition techniques, which partition the domain into subdomains for the direct solution of the problem. To avoid assembling the global problem, iterative domain decomposition methods, such as Schwarz methods, are used. However, these methods incur significant computational costs because of repeated local problem solution steps.

To mitigate this computational burden, model order reduction techniques are employed, significantly reducing the cost of parameterized problems. Special attention is given to non-intrusive, data-driven techniques that integrate classical approaches with neural network-based approximations to create efficient surrogate models, bypassing the need for projection into the original problem.

The primary objective is to integrate domain decomposition methods with non-intrusive model reduction techniques to devise an algorithm for approximating the global solution of multiscale problems through iterative solution of local problems. These local problems are approximated using reduced models, with training conducted exclusively on a local scale. The proposed methodology is applied to multiscale problems that arise in the interaction of microvascular networks with biological tissues. This application is particularly critical in scenarios where modern technologies generate highly complex networks, surpassing current simulation capabilities for associated physical phenomena, such as diffusion transport equations. The proposed methodology effectively addresses these inherent limitations.

2. Methodologies

Regarding the domain-decomposition techniques, the Schwarz method requires the subdomains to overlap and this could be a severe restriction and a waste of effort in the regions shared by two or more subdomains. This is why, for the purpose of this work, a variant for non overlapping subdomains presented in [3] is chosen. The main idea behind this method is that at each step the same equation is solved in each subdomain passing from each subdomain to the others a convex combination of Dirichlet and Neumann data: in particular this yields a Robin type boundary condition on each interface. Regarding the construction of the ROM, a non-intrusive reduced basis method combining Proper Orthogonal Decomposition for reduced basis generation and artificial neural networks for interpolation is employed [2]. This approach is suitable for quickly and accurately solving complex nonlinear PDEs with non-affine parameter dependence. Despite their significant expressivity, plain neural network (NN) architectures based on dense layers encounter major limitations, especially with high-dimensional data. As dimensionality increases, dense architectures become intractable, harder to optimize, and prone to overfitting, necessitating more training data. To address this challenge, a class of sparse architectures known as Mesh-Informed Neural Networks (MINNs) is utilized. Proposed in [1, 5], MINNs are a class of architectures specifically tailored to handle mesh-based functional data, and thus are of particular interest for reduced order modeling of parametrized Partial Differential Equations(PDEs). The driving idea behind MINNs is to embed hidden layers into discrete functional spaces of increasing complexity, obtained through a sequence of meshes defined over the underlying spatial domain. The approach leads to a natural pruning strategy that enables the design of sparse architectures that are able to learn general nonlinear operators.

2.1. DD-FOM

Following the method described in the previous paragraph, this section provides a rigorous mathematical formulation of the domain decomposition method applied to multiscale problems. Consider a coupled 3D-1D problem with the following general formulation: given Ω, the global 3D domain subdivided into $m \geq 0$ subdomains Ω_i such that $\overline{\Omega} = \cup_{i=1}^{m} \overline{\Omega_i}$ and $\gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, and given Λ, the 1D domain. The global 3D solution u on Ω and the global 1D solution $u_\circledast$ on Λ are provided by solving:

$$\begin{cases} \mathcal{L}_\Omega u + \mathcal{M}(u, u_\circledast) = f & \text{in } \Omega, \\ \mathcal{L}_\Lambda u_\circledast + \mathcal{M}(u, u_\circledast) = F & \text{in } \Lambda, \\ \mathcal{B}_\Omega u = g & \text{on } \partial\Omega, \\ \mathcal{B}_\Lambda u_\circledast = G & \text{on } \partial\Lambda, \end{cases} \tag{1}$$

To avoid the computational cost and complexity of subdividing the 1D domain, the 3D problem is first solved using a domain decomposition method. Once the global 3D solution u is obtained by assembling the local solutions u_i, the 1D problem is solved using updated information from the domain Ω.
The strategy involves the following steps at each iteration $k > 0$:

- **Step 1**: Solve the local 3D problem on each of the m subdomains.
- **Step 2**: Assemble the global solution u^k from the local solutions u_i^k.
- **Step 3**: Solve the global 1D problem using the updated 3D solution to compute $u_\circledast^k$. Repeat from Step 1.

Step 1
The first step requires, at each iteration $k > 0$, solving the following local problem on each subdomain Ω_i:

$$\begin{cases} \mathcal{L}_{\Omega_i} u_i^k + \mathcal{M}(u_i^k, u_\circledast^{k-1}) = f_i & \text{in } \Omega_i, \\ \mathcal{B}_{\Omega_i} u_i^k = g & \text{on } \partial\Omega_i \backslash \gamma_{ij}, \\ \mathcal{B}_{loc} u_i^k = g_{i,loc}^{k-1} & \text{on } \gamma_{ij}, \end{cases} \tag{2}$$

where $\mathcal{B}_{loc}$ represents the combination of the Neumann and Dirichlet boundray conditions at the interfaces between the local domains Ω_i. The function g_{loc}^{k-1} computes $\frac{\partial u_j^{k-1}}{\partial n_{ij}} + \lambda_{ij} u_j^{k-1}$ on γ_{ij} for each j such that $\partial\Omega_i \cap \partial\Omega_j \neq \emptyset$. Choosing g_{loc}^{k-1} as the known term results in a Jacobi-type method, while g_{loc}^k results in a Gauss-Seidel-type method. For the purpose of this work Jacobi-type iterations are chosen.

Step 2
The global solution u^k is assembled from the local solutions u_i^k using a proper definded operator $\mathcal{T} : \Omega_1 \times \Omega_2 \times ... \times \Omega_m \to \Omega$:

$$u^k = \mathcal{T}(u_1^k, u_2^k, \dots, u_m^k) \tag{3}$$

Step 3
The updated 3D solution u^k is used to solve the 1D global problem and compute $u_{\circledast}^k$:

$$\begin{cases} \mathcal{L}_\Lambda u_{\circledast}^k + \mathcal{M}(u^k, u_{\circledast}^{k-1}) = F & \text{in } \Lambda, \\ \mathcal{B}_\Lambda u_{\circledast}^k = G & \text{on } \partial\Lambda, \end{cases} \tag{4}$$

At the end of each iteration, a control criterion is calculated to assess the convergence of the method. The iteration process ends once the convergence criterion falls below a chosen tolerance.

2.2. DD-ROM

Solving problem 2 with classical discretization methods involves significant computational costs due to the repeated resolution of local problems, which can be complex. Additionally, assembling the multiscale problem on the domain of interest can be infeasible due to its complexity. To address these limitations, this work proposes combining domain decomposition methods and reduced order model (ROM) techniques, referred to as DD-ROM.
Unlike the standard Finite Element Discretization method described earlier, the DD-ROM algorithm addresses the 3D local problem 2 using a properly trained ROM. This approach focuses on non-intrusive, data-driven techniques that combine classical methods with neural network-based approximation techniques to build efficient surrogate models. These models do not require projection into the original problem's discrete subspace, significantly reducing computational costs and effort, especially for complex multiscale problems.
After defining a suitable parameter set μ, a ROM operator $\mathcal{L}_{\text{ROM}}$ is trained so that the solution u_{ROM} approximates the solution of the local Full-Order Model (FOM) 2:

$$u_{\text{ROM}}(\mu) \approx u_{\text{FOM}}(\mu) \tag{5}$$

This ROM is trained exclusively on local scales, introducing some approximation error, but significantly reducing computational costs and time compared to solving the FOM for each subdomain Ω_i.
Given initial guesses u^0 and $u_{\circledast}^0$, each local solution u_i^k for subdomains Ω_i and iteration $k > 0$ is obtained by solving the following steps:

- **Step 1**:

  $$u_i^k = \mathcal{L}_{\text{ROM}}(\Omega_i, \Lambda_i, u\circledast^{k-1}, g_{i,loc}^{k-1}, g) \tag{6}$$

  Here, $\mathcal{L}_{\text{ROM}}$ is a ROM trained with snapshots computed by solving the global problem on local subdomains. The information from Ω_i, Λ_i, and $u\circledast^{k-1}$ is aggregated into a continuous function, $\mathbf{d}^+$, which combines the Euclidean distance between nodes of the two domains weighted by the 1D solution values at those nodes: $\forall x \in \Omega$:

  $$\mathbf{d}^+(x) = \left(1 - \min_{y \in \Lambda} \|x - y\|_2\right) * u_{\circledast}^{k-1}(y)$$

  This strategy reduces the number of inputs to the ROM. $g_{i,loc}^{k-1}$ encodes the Robin boundary conditions at the subdomain interfaces. The local problem at each iteration is:

  $$u_i^k = \mathcal{L}_{\text{ROM}}(\mathbf{d}^+(\Omega_i, \Lambda_i, u_{\circledast}^{k-1}), g_{i,loc}^{k-1}, g) \tag{7}$$

  Remark: the methodology of training the Reduced Order Model (ROM) on a local subdomain to solve the global Full Order Model (FOM), and then employing this ROM to approximate the local problem 2, is consistent and coherent. This is due to the fact that the operators $\mathcal{B}_{\Omega_i}$ and $\mathcal{B}_{loc}$ are equivalent. This is exactly what happens for the reference problem on which this methodology will be applied.
- **Step 2 and Step 3**: The same described for the DD-FOM algorithm.

Therefore, the DD-FOM and DD-ROM algorithms are equivalent, with the main difference being how the local 3D solutions are computed. Applying a ROM provides a consistent and efficient alternative to solving the full-order model (FOM).

3. Application of the DD-ROM to a problem with microstructure

To demonstrate the efficacy of the DD-ROM algorithms, it is applied to a problem with microstructure. In particular, the ultimate goal of

this thesis would be to apply the new algorithm developed on a problem that models blood flow and transport at the microcirculation level. The model is proposed by [4] and is a coupled 3D-1D problem in which the unknown variables are the pressure p^V in the vessels and the pressure p^t in the tissue in which the vessels are embedded. Thus, p^V represents the variable $u_\circledast$ in problem 1, 2, 4, while p^t plays the role of the variable u in problems 1, 2.
The 3D domain in which the method is tested is a cube $(-1, 1)^3$ with a $40 \times 40 \times 40$ mesh. This domain is subdivided into eight unit cubes, each with a $20 \times 20 \times 20$ mesh.
The vascular networks are represented as metric graphs, with vertices being points in the 3D space.
The POD-MINN architecture tailored for this work has the following structure:

$$\mathcal{M}_{rb}(\mu; \theta_{rb}) = \mathcal{M}_{rb,m}(\mu_m; \theta_m)\mathcal{N}_{rb,M}(\mu_M; \theta_M),$$

In particular it's the combination of two neural networks:

- MINN for Geometrical Input Data: $\mathcal{M}_{rb,d}(\mathbf{d}^+; \theta_m)$. The function $\mathbf{d}+$ encodes the information coming from the geometric 3D and 1D domains and the value of the 1D solution at the nodes.
- Dense Neural Network for Physical Parameters : $\mathcal{N}_{rb,M}(\mu_M; \theta_M)$. μ_M represents the value of the boundary conditions prescribed on each local subdomain.

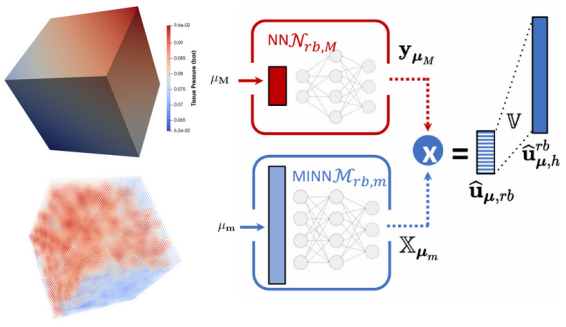


Figure 1: Sketch of the MINN architecture

The ROM has been trained on 1200 snapshots computed solving the 3D-1D FOM, varying the spatial configuration of the vascular networks and the values of the boundary conditions. In particular, the linear system related to the global problem is solved using PETSc (Portable, Extensible Toolkit for Scientific Computation), a widely used library for solving linear and nonlinear systems, as well as optimization problems, in the FEniCs framework. The solver type used is GMRES (Generalized Minimal Residual), and the preconditioner employed is HYPRE with the BoomerAMG (Algebraic Multigrid) method. After 500 epochs, the averaging ROM error is 0.70%.

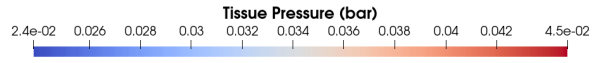
Algorithm 1 DD-ROM Algorithm

```
Given m, tol
Subdivide the 3D domain in m subdomains
Initialize the global 3D and 1D solutions
while control_criteria < tol do
    Generate the function d^+ for each subdomain
    Split the global 3D solutions into m local solutions
    Compute the boundary conditions μ for each subdomain
    for each Subdomain do
        Compute the new local solutions u_i = ROM(d^+, μ)
    end for
    Assemble the updated global 3D solution
    Apply a smoother operator to the global 3D solution
    Solve problem 4 to update the global 1D solution
    Compute the control_criteria
end while
```

With respect to the DD-ROM algorithm described in the previous section, a smoother operator is added to reduce the discontinuities at the interfaces between the subdomains.

4. Numerical Results

To analyze the functioning of the iterative algorithm in more detail, the figures 2, 3 show the difference between two iterations (specifically the tenth and the sixty-sixtieth). By examining the changes between these iterations, it is possible to gain insights into the convergence behavior and accuracy of the algorithm.



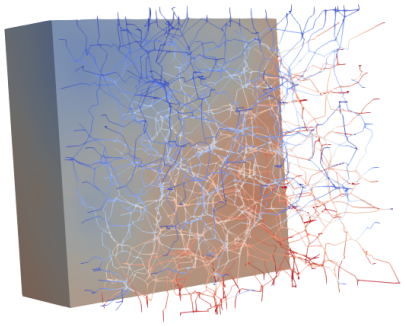


Figure 2: DD-ROM section at the 10^{th} iteration

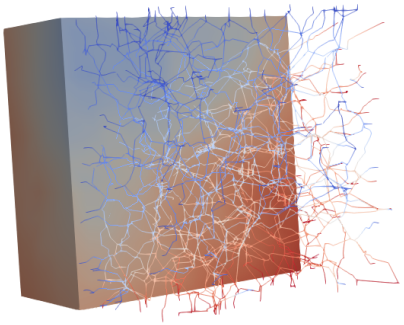


Figure 3: DD-ROM section at the 60^{th} iteration

4.1. Convergence Analysis

The parameters that modulate the influence of Dirichlet and Neumann data in the Robin boundary conditions play an essential role in determining whether the iterative DD methods would converge to a feasible solution or not. Since the parameter that governs the influence of Neumann data in the reference problem is a physical parameter (and for simplicity define it as α) and, therefore, it is fixed, three different values of the parameter β, that modulates the influence of Dirichlet data, have been used to create three different datasets on which train the NN and test the DD-ROM algorithm. After testing these values, it was observed that the method does not converge to a feasible solution when the ratio between the two parameters $\frac{\beta}{\alpha} \geq \approx 100$.

$\frac{\beta}{\alpha}$	Convergence
$\approx 10^5$	NO
$\approx 10^2$	NO
≈ 10	YES

Table 1: DD-ROM Convergence

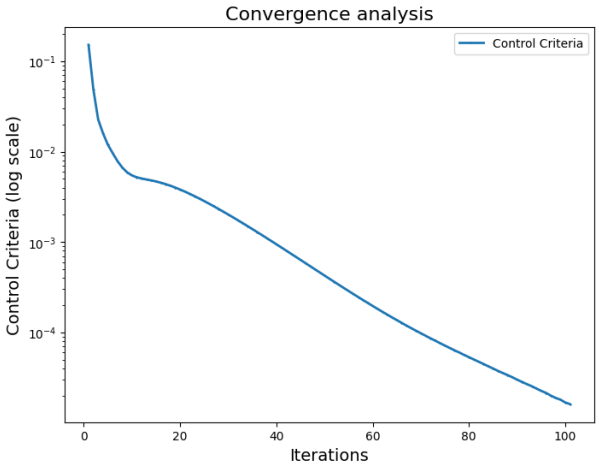


Figure 4: Convergence Analysis for $\frac{\alpha}{\beta} = 10$

4.2. Comparison with the Full Order Model

To further assess the reliability of this new methodology a comparison with the Full Order Model solved with the classical Finite Element Discretization is provided. The FOM approximation is obtained solving 1 using PETSc library in the FEniCs framework. The solver type used is GMRES and the preconditioner employed is HYPRE with the BoomerAMG method.
Overall the behavior of the solution is well captured from the DD-ROM methodology as it is possible to understand from figure 5. However, as can be deduced from the graphic 6, there is still a quite significant error between the 2 solutions. This is, for sure, partly due to the propagation of the POD and ROM approximation error throughout all the iterations. Some important developments to reduce this error are explained in the next session.

5. Conclusion and Further Developments

The thesis aims to combine domain decomposition techniques with non-intrusive model reduc-

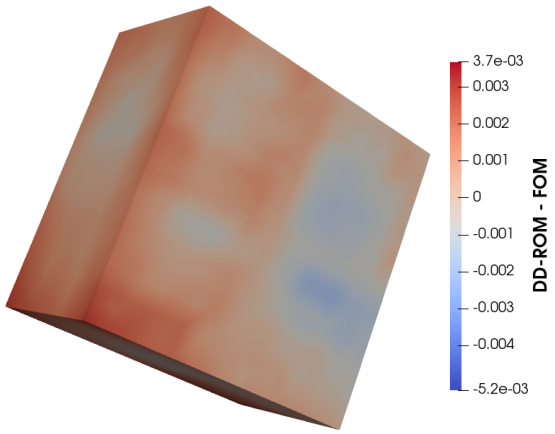


Figure 5: Qualitative Difference between DD-ROM and FOM solutions

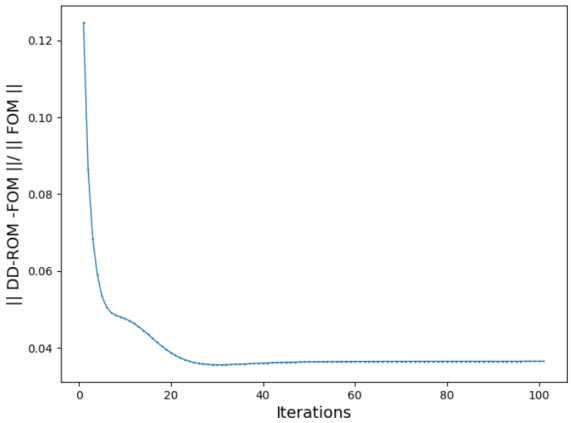


Figure 6: Normalized difference between the FOM and DD-ROM solution

tion techniques to develop an algorithm for approximating global solutions of multiscale problems. This is achieved through the iterative solution of local problems approximated with reduced models, trained exclusively at the local scale.

The research introduces a domain-decomposition ROM (Reduced-Order Model) method for parameter-dependent problems with microstructure in a stationary regime, showing promise for simulating complex physical phenomena like diffusion transport equations. This method overcomes the limitations of traditional global methodologies.

The innovative Mesh-Informed Neural Network (MINN) architecture is central to the approach, although its current accuracy and computational cost are suboptimal. To improve the ROM approach, it is suggested to incorporate a closure model to reduce the approximation error by using high-dimensional microscale parameters for accurate spatial corrections. Furthermore, integrating the parameter β as input to the neural network and generating multiple snapshots of the Full Order Model (FOM) could enhance flexibility and real-time adaptation.

For the domain decomposition algorithm, refining the assembly function to homogenize local solutions near the interfaces is crucial to eliminating discontinuities and accelerating convergence. Furthermore, increasing computational efficiency through parallelization is essential for handling complex domains and numerous subdomains. These improvements would make the DD-ROM method a reliable, efficient, and accurate solution for multiscale parameter-dependent problems in complex and large domains.

References

[1] Nicola Rares Franco, Andrea Manzoni, and Paolo Zunino. Mesh-informed neural networks for operator learning in finite element spaces. *Journal of Scientific Computing*, 2023.

[2] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.

[3] P.L Lions. Third international symposium on domain decomposition methods for partial differential equations. *Society for Industrial and Applied Mathematics*, pages 202–223, 1990.

[4] Ettore Vidotto, Timo Koch, Tobias Koppl, Rainer Helmig, and Barbara Wohlmuth. Hybrid models for simulating blood flow in microvascular networks. *Multiscale Modeling & Simulation*, 17(3):1076–1102, 2019.

[5] Piermario Vitullo, Alessio Colombo, Nicola Rares Franco, Andrea Manzoni, and Paolo Zunino. Nonlinear model order reduction for problems with microstructure using mesh informed neural networks. *Finite Elements in Analysis and Design*, 229:104068, 2024.