Adaptive Mesh Refinement of the Domain Decomposition+L^2-Jumps Method applied to the Neutron Diffusion Equation on 3D Structured Meshes

Mario Gervais [1,*], François Madiot[1], Minh-Hieu Do[1], Patrick Ciarlet[2]

[1]Université Paris-Saclay, CEA, Service d'Études des Réacteurs et de Mathématiques Appliquées, 91191, Gif-sur-Yvette, France;
[2]POEMS, CNRS, INRIA, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France.

[leave space for DOI, which will be inserted by ANS]

ABSTRACT

A deterministic solver for neutron calculation is classically based on the two step calculation scheme: the 2D lattice step to find the homogenized cross sections and the 3D core step usually based on the neutron diffusion equation for the whole core domain. In general, this equation can be recast in a mixed variational form, and then discretized by using the Raviart-Thomas-Nédélec Finite Element. More importantly, the neutron diffusion equation usually admits low regularity solution due to heterogeneous coefficients. This requires Adaptive Mesh Refinement (AMR) to improve the accuracy of the solution. In order to have independent local refinement on each subdomain, the AMR strategy is combined with the domain decomposition+L^2 jumps as illustrated in [1] with two subdomains on 2D numerical test cases. The goal of this work is to extend the above strategy to 3D structured meshes with multiple subdomains which leads to a more optimal refinement for the full 3D core calculation.

Keywords: diffusion equation, mixed formulation, adaptive mesh refinement, domain decomposition.

1. INTRODUCTION

Simulations with the neutron transport equation at the core level is costly since it requires to solve a complex equation with several variables such as space, direction and energy variables. In industrial applications, the neutron flux at the core level is usually modeled by the neutron diffusion equation.

In this work, we would like to focus on the neutron deterministic code APOLLO3®[2], a shared platform among CEA, EDF and FRAMATOME. In general, this multi-purpose code is based on the two-step calculation scheme: the (2D) lattice calculation and the (3D) core calculation. The lattice calculation is usually performed at the assembly scale in order to get a few groups homogenized cross sections and the full core calculation step is usually performed with the diffusion equation with the homogenized cross sections coming from the previous lattice step. In particular, our context is the development of the MINOS solver based on the mixed Raviart-Thomas-Nédélec finite element discretization of the multigroup diffusion and simplified transport equation (SP$_N$), implemented on Cartesian and hexagonal grids [3].

Therefore in this study, we focus on the one group diffusion equation given by

[*] mario.gervais@cea.fr

$$\begin{aligned} \operatorname{div} \mathbf{p} + \Sigma_a \phi &= S_f \quad \text{in } \mathcal{R}, && \text{(1a)}\\ \mathbb{D}^{-1}\mathbf{p} + \nabla\phi &= 0 \quad \text{in } \mathcal{R}, && \text{(1b)}\\ \phi &= 0 \quad \text{on } \partial\mathcal{R}, && \text{(1c)} \end{aligned}$$

where $\mathcal{R}$ is the core calculation domain, $\mathbb{D}$ is the diffusion coefficient, Σ_a is the absorption cross section, $\mathbf{p}$ is the current ϕ is the flux, and S_f is the neutron source.

Since the cross sections and the diffusion coefficients are usually homogenized to be constant on each cell or assembly in the lattice step, we may have low regularity solution for the neutron diffusion equation which limit the precision and convergence of the solution [4]. Moreover, according to [5, 6, 7, 8], it is very interesting to point out that one of the most effective methods to deal with this problem is mesh subdivision (h-refinement).

Domain Decomposition (DD) refers to the technique of solving partial differential equations in the whole domain by using subroutines that solve the problems on subdomains. Therefore, this strategy is well-suited for parallel computing and received a great attention in scientific computing. In order to perform the domain decomposition calculation, it is essential to ensure the continuity between the subdomains. One of the most popular method is iterative procedure such as the classical overlap iterative Schwarz method or the non-overlap one with the Robin transmission condition [9]. Another way to have the continuity at the interface is using the direct procedure such as the Lagrange multiplier techniques. However, we have to take into account the fact that we have new unknown for the Lagrange multipliers. Among those domain decomposition methods, the domain decomposition+L^2-jumps method (or DD+L^2-jumps method) is proposed in [4]. The successful applications of the DD+L^2-jumps method to criticality calculations is presented in [10] .

In the framework of reactor core simulations, we are interested in combining the Adaptive Mesh Refinement (AMR) strategies to the DD+L^2-jumps method in order to have independent local refinement in each subdomain. This allows to reduce the computational cost and improve the accuracy of the solution. In fact, some preliminary results of an Adaptive Mesh Refinement (AMR) strategy on the domain decomposition+L^2 jumps for neutron diffusion equation are discussed in [1] on 2D test cases and two subdomains.

The purpose of this work is to extend the analysis to 3D structured meshes with several subdomains. In particular, this article is organized as follows. In Section 2, we briefly present the DD+L^2-jumps method and its discretization. Next, the AMR strategy based on a posteriori error estimators is explained in Section 3. Section 4 is dedicated to the numerical test cases in 3D. Finally, some conclusions and perspectives are discussed in Section 5.

2. VARIATIONAL FORMULATION AND DISCRETIZATION

In this section, we present the domain decomposition+L^2-jumps method applied to the problem (1) as introduced in [1].

We assume that the domain $\mathcal{R}$ is a bounded, connected and open subset of $\mathbb{R}^d$ for $d = 2, 3$, having a Lipschitz boundary which is piecewise smooth. Let us define a partition $\{\mathcal{R}^*_{i^*}\}_{1\leq i^* \leq \overset{*}{N}}$ of $\mathcal{R}$. For $\psi \in L^2(\mathcal{R})$, we will use the notation

$$\psi_{i^*} := \psi|_{\mathcal{R}^*_{i^*}},$$

for $1 \leq i^* \leq \overset{*}{N}$. We will note $\Gamma_{i^* j^*}$ the interface between two subdomains of $\mathcal{R}^*_{i^*}$ and $\mathcal{R}^*_{j^*}$, for $i^* \neq j^*$. We define the interface Γ by

$$\Gamma := \cup_{i^*=1}^{\overset{*}{N}} \cup_{j^*=i^*+1}^{\overset{*}{N}} \overline{\Gamma_{i^*j^*}}.$$

We now define the following spaces

$$\begin{aligned}
PH^1_0(\mathcal{R}) &:= \{\psi \in L^2(\mathcal{R}) | \, \psi_{i^*} \in H^1(\mathcal{R}^*_{i^*}), \psi|_{\partial\mathcal{R}^*_{i^*}\setminus\Gamma} = 0, 1 \leq i^* \leq \overset{*}{N}\},\\
\mathbf{PH}(\mathrm{div}\,, \mathcal{R}) &:= \{\mathbf{q} \in L^2(\mathcal{R}) \mid \mathbf{q}_{i^*} \in \mathbf{H}(\mathrm{div}\,, \mathcal{R}^*_{i^*}), 1 \leq i^* \leq \overset{*}{N}\},\\
M &:= \{m = (m_{i^*j^*})_{i^*<j^*} \in \prod_{i^*<j^*} L^2(\Gamma_{i^*j^*})\},\\
\overset{*}{\mathbf{Q}} &:= \{\mathbf{q} \in \mathbf{PH}(\mathrm{div}\,, \mathcal{R}) \mid [\mathbf{q}\cdot\mathbf{n}] \in M\},\\
\mathtt{W} &:= \overset{*}{\mathbf{Q}} \times L^2(\mathcal{R}) \times M,
\end{aligned}$$

where $[\mathbf{q}\cdot\mathbf{n}]$ is called the global jump of the normal component and is defined by

$$[\mathbf{q}\cdot\mathbf{n}]|_{\Gamma_{i^*j^*}} := \mathbf{q}_{i^*}\cdot\mathbf{n}_{i^*} + \mathbf{q}_{j^*}\cdot\mathbf{n}_{j^*}, \text{ for } 1 \leq i^* < j^* \leq \overset{*}{N}.$$

The DD+L^2-jumps method writes,

Find $(\mathbf{p}, \phi, \ell) \in \overset{*}{\mathbf{Q}} \times PH^1_0(\mathcal{R}) \times M$ such that:

$$\begin{aligned}
\mathrm{div}\,\mathbf{p}_{i^*} + \Sigma_{a,i^*}\phi_{i^*} &= S_{f,i^*} \quad \text{in } \mathcal{R}^*_{i^*}, \text{ for } 1 \leq i^* \leq \overset{*}{N}, && \text{(3a)}\\
\mathbb{D}^{-1}_{i^*}\mathbf{p}_{i^*} + \mathbf{grad}\phi_{i^*} &= 0 \quad \text{in } \mathcal{R}^*_{i^*}, \text{ for } 1 \leq i^* \leq \overset{*}{N}, && \text{(3b)}\\
\phi_{i^*} &= \ell \quad \text{on } \partial\mathcal{R}^*_{i^*} \cap \Gamma, \text{ for } 1 \leq i^* \leq \overset{*}{N}, && \text{(3c)}\\
[\mathbf{p}\cdot\mathbf{n}]_{\Gamma_{i^*j^*}} &= 0 \quad \text{for } 1 \leq i^* < j^* \leq \overset{*}{N}. && \text{(3d)}
\end{aligned}$$

The flux and the current respectively belong to the broken spaces $PH^1_0(\mathcal{R})$ and $\mathbf{PH}(\mathrm{div}\,, \mathcal{R})$. The space M is the space of the Lagrange multipliers defined on the interface Γ. They are introduced in order to impose the constraint (3d). The key observation in the domain decomposition+L^2-jumps method is that the jump of the normal component of the current belongs to M. The associated variational formulation writes

$$\text{Find } \mathtt{u} = (\mathbf{p}, \phi, \ell) \in \mathtt{W} \text{ such that } \forall\, \mathtt{w} = (\mathbf{q}, \psi, m) \in \mathtt{W}, \quad c_S(\mathtt{u}, \mathtt{w}) = \int_{\mathcal{R}} S_f\psi, \tag{4}$$

where

$$c_S(\mathtt{u}, \mathtt{w}) := c((\mathbf{p}, \phi), (\mathbf{q}, \psi)) + \int_\Gamma [\mathbf{p}\cdot\mathbf{n}]m - \int_\Gamma [\mathbf{q}\cdot\mathbf{n}]\ell.$$

Let $(\mathcal{T}_h)_h$ be a family of meshes, made for instance of simplices, or of rectangles ($d = 2$), resp. cuboids ($d = 3$), indexed by a parameter h equal to the largest diameter of elements of a given mesh. We introduce Raviart-Thomas-Nédélec approximation spaces indexed by h as follows: $\mathbf{Q}_{i^*,h} \subset \mathbf{H}(\mathrm{div}\,, \mathcal{R}^*_{i^*})$ and $L_{i^*,h} \subset L^2(\mathcal{R}^*_{i^*})$, for $1 \leq i^* \leq N^*$. Introducing the discrete space of Lagrange multipliers $M_h \subset M$, we then set

$$\mathbf{Q}^*_h = \prod_{i^*=1}^{N^*} \mathbf{Q}_{i^*,h}, \quad L^*_h = \prod_{i^*=1}^{N^*} L_{i^*,h}, \quad \mathtt{W}_h = \mathbf{Q}^*_h \times L^*_h \times M_h,$$

The discrete variational formulation writes,

$$\text{Find } \mathtt{u}_h = (\mathbf{p}_h, \phi_h, l_h) \in \mathtt{W}_h \text{ such that } \forall \mathtt{w}_h = (\mathbf{q}_h, \psi_h, m_h) \in \mathtt{W}_h,\ c_S(\mathtt{u}_h, \mathtt{w}_h) = \int_{\mathcal{R}} S_f \psi_h. \qquad (5)$$

A priori estimates have been derived for low regularity solutions in [4].

3. ADAPTIVE MESH REFINEMENT STRATEGY

3.1. Generalities

In this paper, we aim to illustrate an AMR strategy for the problem (5). The general method generates a sequence $\mathcal{T}_{h_k}$ from the initial mesh $\mathcal{T}_{h_0}$ by using the following iterative loop, which is divided into four modules as presented in Figure 1.

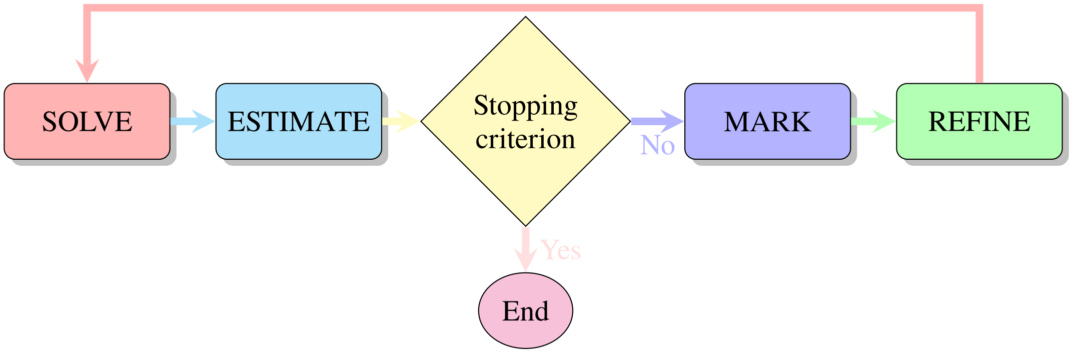


Figure 1. AMR process

The module **SOLVE** amounts to solving the source problem (5). In module **ESTIMATE**, the local error indicator is computed on each element $K \in \mathcal{T}_h$ based a posteriori error estimate for the discrete solution [11]. It is defined as

$$\eta_K = \left(\eta_{r,K}^2 + \sum_{K' \in N^*(K)} \eta_{f,K}^2 \right)^{1/2}, \qquad (6)$$

with $\overset{*}{N}(K) := \{K' \in \mathcal{T}_h \mid \dim_H(\partial K' \cap \partial K) = d - 1)\} \cap \mathcal{R}_K^*$, where $\dim_H$ is the Hausdorff's dimension $\mathcal{R}_K^*$ is the subdomain which contains K,

$$\eta_{r,K} := \|\Sigma_a^{-1/2}(S_f - \operatorname{div} \mathbf{p}_h - \Sigma_a \tilde{\phi}_h)\|_{L^2(K)} \quad \text{and} \quad \eta_{f,K} := \|\mathbb{D}^{1/2}(\mathbb{D}^{-1}\mathbf{p}_h + \mathbf{grad}\, \tilde{\phi}_h)\|_{L^2(K)},$$

where the reconstruction $\tilde{\phi}_h$ is defined as in 3.2.

The stopping criterion is defined as $\max_{K \in \mathcal{T}_{h_k}} \eta_K \leq \varepsilon_{\text{AMR}}$ for a user-defined $\varepsilon_{\text{AMR}} > 0$. In Section 4, we use a relative stopping criterion $\varepsilon_{\text{AMR}} = \varepsilon_{\text{AMR, rel}} \|\phi_h\|_{L^2(\mathcal{R})}$, where $\varepsilon_{\text{AMR, rel}} > 0$.

The purpose of the module **MARK** is to select a set of elements with large error to be refined. For a user-defined parameter θ, the marking strategy consists in finding for all $1 \leq i^* \leq N^*$ an optimal set of cells S_{i^*} such that one has

$$\eta(S_{i^*}) \leq \theta_{i^*}\, \eta(\mathcal{T}_{h,i^*}), \quad \text{where} \quad \eta(S_{i^*}) := \left(\sum_{K \in S_{i^*}} \eta_K^2 \right)^{1/2},$$

and $\theta_{i^*} > 0$ is a user-defined parameter. According to [12, Section 6], an efficient strategy which preserves the Cartesian structure of the mesh is the *direction* marker strategy. One selects for each direction $\mathbf{e}_x$, $x = 1, \dots, d$, the smallest set of lines $L_{x,i^*} \subset \mathcal{T}_{h,i^*}$ such that $\eta(L_{x,i^*}) \geq \theta_{i^*}\eta(\mathcal{T}_{h,i^*})$. The resulting selected set is $\cup_{1\leq i^*\leq N^*, x=1,\dots,d} L_{x,i^*}$.
Finally, the module **REFINE** refines, for all $1 \leq i^* \leq N^*$, the mesh $\mathcal{T}_{h,i^*}$ if the stopping criterion is not reached locally i.e. $\max_{K\in\mathcal{T}_{h,i^*}} \eta_K > \varepsilon_{\text{AMR}}$.

3.2. Reconstruction

By construction $\phi_h \in L_h^*$, it is likely that $\phi_h \notin H^1(\mathcal{R})$. In order to ensure the reliability and efficiency of the estimators, we introduce a *reconstruction* of the discrete solution $\mathrm{u}_h = (\mathbf{p}_h, \phi_h, \ell_h)$, denoted $\tilde{\zeta}_h = (\tilde{\mathbf{p}}_h, \tilde{\phi}_h(\phi_h, l_h))$, where

- $\tilde{\mathbf{p}}_h = \mathbf{p}_h$;
- $\tilde{\phi}_h \in H_0^1(\mathcal{R})$.

In order to reconstruct the neutron flux when dealing with a Cartesian conformal mesh, one can use the so-called *averaging* method, described in [12, Section 5.1]. However, for non-conformal meshes, non-conformities need to be tackled. The method we use here is an extension of the averaging method. Let us describe it in the case of a RTN$_0$ discretization. For $1 \leq i^* \leq N^*$, let $\mathcal{V}_h(\mathcal{T}_{h,i^*})$ be the set associated to the $\mathbb{Q}^1$ Lagrange finite elements on the Cartesian mesh. Let also $\mathcal{V}_{h,\text{disc}} = \cup_{i^*=1,\dots,N^*}\mathcal{V}_h(\mathcal{T}_{h,i^*})$ be the set of nodes associated to the degrees of freedom of the $PH_0^1(\Omega)$-conforming Lagrange Finite Element space $\mathbb{Q}^1(\mathcal{T}_h)$. We compute $\phi_{h,\text{disc}} \in PH_0^1(\Omega)$ such that

$$\forall a \in \mathcal{V}_{h,\text{disc}}^1, \quad \phi_{h,\text{disc}}(a) = \begin{cases} \dfrac{1}{|\mathcal{E}_a|} \displaystyle\sum_{E\in\mathcal{E}_a} l_{h|E}(a) & \text{if } a \in int(\Gamma_S), \\ \dfrac{1}{|\mathcal{T}_a|} \displaystyle\sum_{K\in\mathcal{T}_a} \phi_{h|K}(a) & \text{otherwise.} \end{cases}$$

The reconstruction $\tilde{\phi}_h \in V_h^1$ is then defined such that

$$\forall 1 \leq i^* \leq N^*, \forall a \in \mathcal{V}_h^1, \quad \tilde{\phi}_h(a) = \begin{cases} \dfrac{1}{|\mathcal{E}_a|} \displaystyle\sum_{E\in\mathcal{E}_a} l_{h|E}(a) & \text{if } a \in int(\Gamma_S), \\ \phi_{h,\text{disc}}(a) & \text{otherwise.} \end{cases}$$

For the practical implementation, an algorithm has been proposed in [1, Section 4.2]. Although the example has been taken in 2D, the 3D extension extension can be easily performed. The main difference lies in the fact that in 3D, the 2D-mesh of the interface contains vertices which are not included in any mesh. Indeed, in order to have a conformal 2D-mesh at the interface, one needs to construct a Cartesian mesh from a list of vertices contained in a 2D-plane. This construction generates new vertices, as shown in Figure 2.

4. NUMERICAL RESULTS

4.1. Mono-group 3D test-case

The test-case is a 3D extension of the test presented in [13]. We use RTN$_0$ discretization. For the DD+L^2-jumps method, the Lagrange multiplier space is built as the sum of the trace of the normal component of the

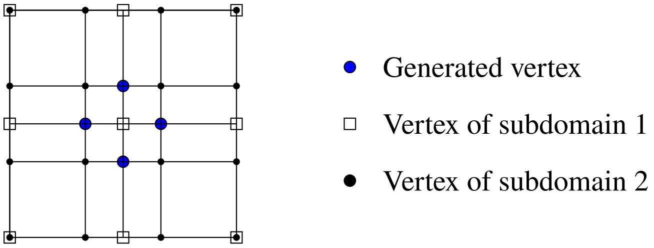


Figure 2. 2D-mesh of the interface for a given configuration. Black vertices belong to one of the two subdomains, blue vertices are generated to have a Cartesian 2D-mesh of the interface.

discrete current [4, Section 5.2.1] using Gram-Schmidt orthogonalization.

We let $\mathcal{R} = [0, 30]^3$ the domain of computation, and $\mathcal{D} = [10, 20]^3$. The geometry of the test case is shown in Figure 3 and the value of the physical parameters are represented in Table I.

Data	$\mathbf{x} \in \mathcal{D}$	$\mathbf{x} \in \mathcal{R} \setminus \mathcal{D}$
$\Sigma_a(\mathbf{x})$	1/2	1.9
$\mathbb{D}(\mathbf{x})$	1/3	1/6
$S_f(\mathbf{x})$	1	0

Table I. Physical parameters for the 3D Mono-group test-case

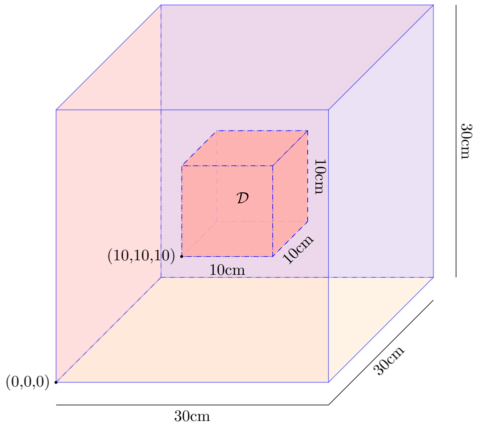


Figure 3. Geometry of test case.

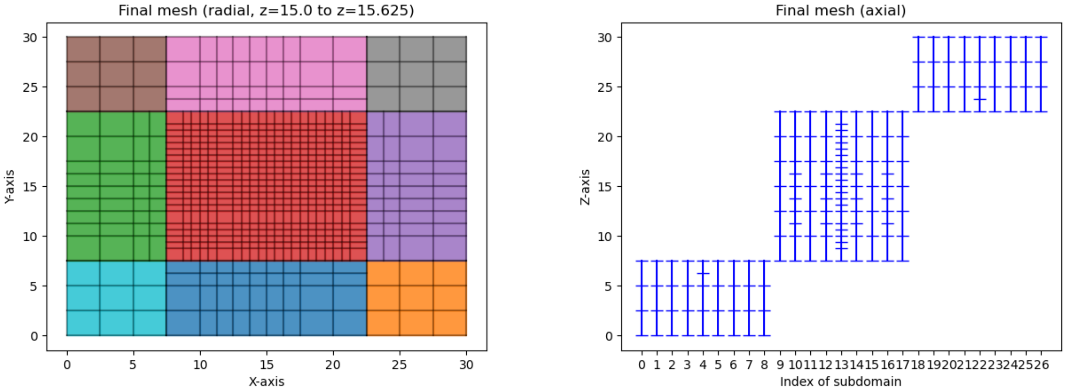
The so-called reference solution is computed on a uniform 192×192×192 Cartesian mesh. For the test-case, the initial mesh configuration is a uniform 12 × 12 × 12 Cartesian mesh. We compared the AMR process between three different strategies : uniform refinement, AMR with a monodomain refinement, AMR with the DD+L^2-jumps method. The domain decomposition is performed on an uniform Cartesian 3×3×3 grid ($N^* = 27$). The stopping criterion is set to $\varepsilon_{\text{AMR,rel}} = 0.015$, which gives $\varepsilon_{\text{AMR}} \approx 0.8$.
We apply the AMR process for the DD+L^2-jumps method detailed in Section 3 .
In the MARK module, we fix $\theta_{i^*} = 0.5$ for all $1 \leq i^* \leq N^*$, except in the subdomain j^* located at $(7.5, 22.5)^3$ where we set $\theta_{j^*} = 0.3$. The choice is based on numerical results obtained in [11].
Correspondingly, we apply the AMR proces for the monodomain formulation. The ESTIMATE module is

based on the estimators defined in [12, Section 5.1], and the averaging method described in [12, Section 5.2]. In the REFINE module, we fix $\theta = 0.5$.
All these methods are compared to the reference flux in Table II.

Iter	Uniform			Monodomain			DD+L^2-jumps		
	N_h	$\max_{K\in\mathcal{T}_h} \boldsymbol{\eta_K}$	L^2-error	N_h	$\max_{K\in\mathcal{T}_h} \boldsymbol{\eta_K}$	L^2-error	N_h	$\max_{K\in\mathcal{T}_h} \boldsymbol{\eta_K}$	L^2-error
0	1728	3.356	0.266	1728	3.356	0.266	1728	3.054	0.266
1	13824	1.249	0.173	3375	2.916	0.245	3469	2.799	0.220
2	46656	0.659	0.125	8000	1.271	0.157	5484	1.269	0.139
3	-	-	-	17576	0.665	0.098	12001	0.614	0.098

Table II. Comparison between meshes, estimators and relative L^2-error with respect to the reference solution for different the different mesh configurations where N_h represents the number of mesh elements in $\mathcal{T}_h$.

The AMR strategy for the DD+L^2-jumps method requires approximately 4 times less elements than the uniform refinement to reach the stopping criterion. It also requires 30% less elements that the monodomain AMR. The final mesh is represented in Figure 4. For the radial component, we chose the slice where $z = 15$. We clearly see that the AMR procedure mainly refined the central mesh where the source and the interface between the two materials is located. It is also the subdomain where the regularity of the flux is the lowest.



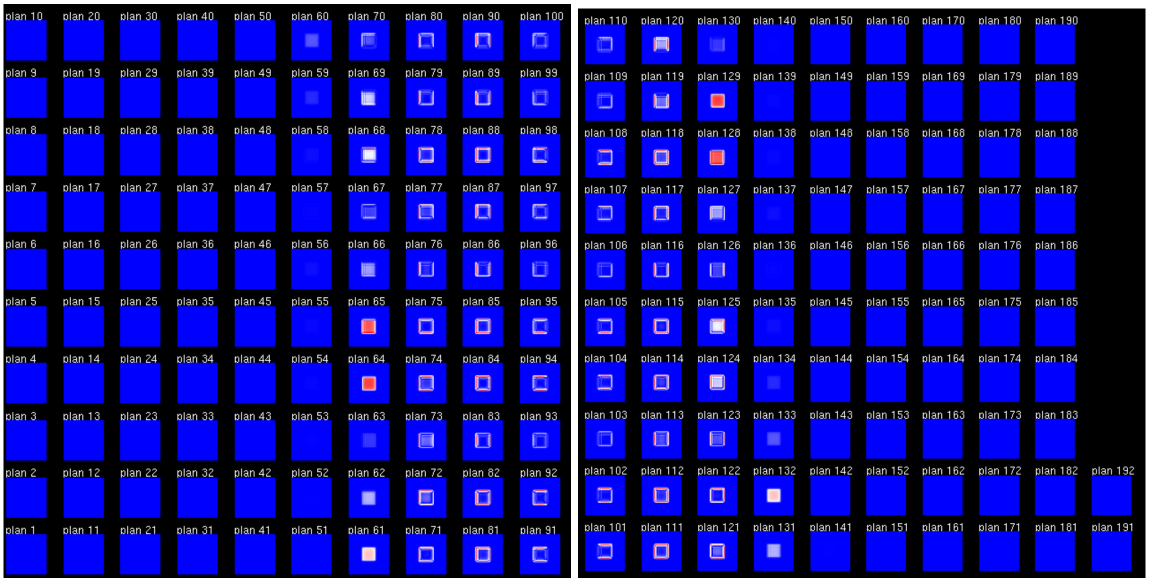
(a) Radial mesh at $z = 15$

(b) Axial mesh. Starting from the bottom south west, the subdomains are indexed from west to east, from south to north, then from bottom to top.

Figure 4. Final mesh for the DD+L^2-jumps method.

The sparsity ratio is defined as the number of non zero-elements in a matrix divided by the total number of elements in the matrix. It is shown in Table III that the sparsity ratio is very similar between the monodomain and the DD case.

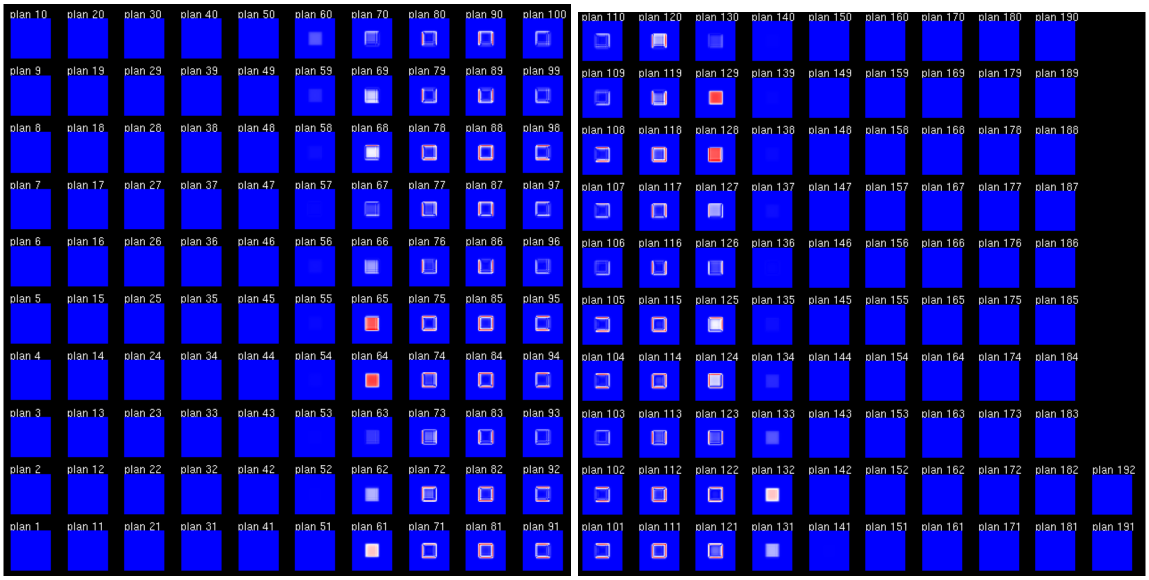
The absolute error of the AMR solutions (for the monodomain and the domain decomposition problems) is shown in Figures 5 and 6. Its peak is also located at the interface between the materials and in $\mathcal{D}$, indicating that our method was able to refine the mesh in the regions of interest thanks to rigorous a posteriori estimators.



(a)

(b)

Figure 5. Spatial distribution of the L^2-absolute error for the final monodomain mesh. The peak of the the error is located at the interface between the materials and in $\mathcal{D}$. The values range from 2.10^{-14} to 0.350.



(a)

(b)

Figure 6. Spatial distribution of the L^2-absolute error for the final DDM mesh. The peak of the the error is located at the interface between the materials and in $\mathcal{D}$. The values range from 2.10^{-14} to 0.350.

Iteration	Uniform refinement	Monodomain	DD+L^2-jumps
0	$7.13.10^{-2}$	$7.13.10^{-2}$	$5.20.10^{-2}$
1	$9.41.10^{-3}$	$3.73.10^{-2}$	$2.83.10^{-2}$
2	$2.84.10^{-3}$	$1.6.10^{-2}$	$1.94.10^{-2}$
3	-	$7.43.10^{-3}$	$9.80.10^{-3}$

Table III. Comparison between the different sparsity ratios of the matrix of the system for the different mesh configurations (in ‰).

5. CONCLUSIONS

In this work, we applied on 3D structured meshes an adaptive mesh refinement strategy for the domain decomposition+L^2 jumps method for the neutron diffusion equation, which relies on a posteriori error estimators for the source problem. We have shown numerically that this approach is more effective than in the monodomain case and the uniform refinement.
Future work will be dedicated to the extension of this method to the SP$_N$ model.

ACKNOWLEDGEMENTS

APOLLO3® is a registered trademark of CEA. We gratefully acknowledge EDF and Framatome for their long-term partnership and their support.

REFERENCES

[1] M. Gervais, F. Madiot, M.-H. Do, and P. Ciarlet, Jr. “Adaptive solution of the domain decomposition+L^2-jumps method applied to the neutron diffusion equation on structured meshes.” *EPJ Web Conf*, **volume 302**, p. 02011 (2024).

[2] P. Mosca, L. Bourhrara, A. Calloo, A. Gammicchia, F. Goubioud, L. Lei-Mao, F. Madiot, F. Malouch, E. Masiello, F. Moreau, S. Santandrea, D. Sciannandrone, I. Zmijarevic, G. Garcias-Cervantes, E. Y.and Valocchi, J. Vidal, F. Damian, A. Brighenti, B. Vezzoni, P. Laurent, and A. Willien. “APOLLO3®: Overview of the new code capabilities for reactor physics analysis.” In *Proceeding of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, M&C 2023* (2023).

[3] A.-M. Baudron and J.-J. Lautard. “MINOS: A Simplified P$_N$ Solver for Core Calculation.” *Nuclear Science and Engineering*, **volume 155**(2), pp. 250–263 (2007).

[4] P. Ciarlet, Jr., E. Jamelot, and F. Kpadonou. “Domain decomposition methods for the diffusion equation with low-regularity solution.” *Computers & Mathematics with Applications*, **volume 74**(10), pp. 2369–2384 (2017).

[5] Y. Wang, W. Bangerth, and J. Ragusa. “Three-dimensional h-adaptivity for the multigroup neutron diffusion equations.” *Progress in Nuclear Energy*, **volume 51**(3), pp. 543–555 (2009).

[6] Y. Wang and J. Ragusa. “Application of hp adaptivity to the multigroup diffusion equations.” *Nuclear Science and Engineering*, **volume 161**(1), pp. 22–48 (2009).

[7] M. Vohralík. "A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations." *SIAM Journal on Numerical Analysis*, **volume 45**(4), pp. 1570–1599 (2007).

[8] M.-H. Do, P. Ciarlet, Jr., and F. Madiot. "Adaptive solution of the neutron diffusion equation with heterogeneous coefficients using the mixed finite element method on structured meshes." In *EPJ Web of Conferences*, volume 247, p. 02002. EDP Sciences (2021).

[9] E. Jamelot and P. Ciarlet, Jr. "Fast non-overlapping Schwarz domain decomposition methods for solving the neutron diffusion equation." *Journal of Computational Physics*, **volume 241**, pp. 445–463 (2013).

[10] L. Giret, P. Ciarlet, Jr., and E. Jamelot. "Criticality Computation with Finite Element Method on Non-Conforming Meshes." In *Proceeding of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, M&C 2017* (2017).

[11] P. Ciarlet, Jr., M.-H. Do, M. Gervais, and F. Madiot. "A posteriori error estimates for the DD+L^2 jumps method on the Neutron Diffusion equations." (2024). Working paper or preprint.

[12] P. Ciarlet, Jr., M. H. Do, and F. Madiot. "A posteriori error estimates for mixed finite element discretizations of the Neutron Diffusion equations." *ESAIM: M2AN*, **volume 57**(1), pp. 1–27 (2023).

[13] J. I. Duo, Y. Y. Azmy, and L. T. Zikatanov. "A posteriori error estimator and AMR for discrete ordinates nodal transport methods." *Annals of Nuclear Energy*, **volume 36**, pp. 268–273 (2009).