

A Non Overlapping Parallel Domain Decomposition Method Applied to The Simplified Transport Equations

LATHUILIERE Bruno and BARRAULT Maxime

EDF R&D

bruno.lathuiliere@edf.fr; maxime.barrault@edf.fr

RAMET Pierre and ROMAN Jean

INRIA Bordeaux - Sud-Ouest and LaBRI UMR5800

ramet@labri.fr; roman@labri.fr

1 Introduction

The exploitation of a PWR based nuclear plant requires to change the fuel each year in the reactor. It is done while ensuring the safety and the productivity of the plant in service. More precisely, EDF uses the numerical simulation to achieve this goal. In our context, the neutron transport inside a nuclear reactor have to be simulated. So, EDF has developed a fast sequential solver [2] based on the simplified transport equations SPN [4]. We report on Table 1 its performances, obtained while considering the pin by pin IAEA-3D benchmark [3] with the SP1/SP3 model and the RT0/RT1 finite element approximation [6]:

Type of computation	SP1-RT0	SP1-RT1	SP3-RT0	SP3-RT1
DOF Number	31 948 950	254 747 720	63 897 900	509 495 440
Memory consumption	1.5 Go	13.4 Go	2.5 Go	23 Go
Time	143.55 s	×	347.59 s	×

Table I. Performance of the sequential solver for different computations with the same convergence parameters (see section 3 for a description of the target cluster). × indicates that the computation can't be run because of memory consumption.

This sequential code suffers of two limitations. On the one hand, we are not able to run some industrial cases due to the memory consumption and/or the computational time. On the other hand, it is necessary to refine a large part of the mesh as soon as a better numerical approximation is needed in a local part of the core. So, we propose a non overlapping domain decomposition to tackle these problems. In the following, the domain decomposition method is presented. Then, numerical results on the IAEA-3D benchmark are provided.

2 Description of the domain decomposition method

2.1 Presentation of the problem

The simplified transport equations SPN for a reactivity calculation lead to the following algebraic problem:

find the highest k_{eff} such that

$$AX = \frac{1}{k_{eff}}FX \quad (1)$$

where A and F are the transport and fission matrices. To solve problem (1), a Generalized Power Inverse Iteration Algorithm is used [7] where a linear system

$$AX = S \quad (2)$$

has to be solved at each iteration. In order to benefit from the pattern of A , we consider the following iterative strategy to solve (2) :

1. to solve (2), we use an iterative block Gauss-Seidel algorithm ;
2. for each linear system involving the diagonal blocks A_g of A , we use an iterative block Gauss-Seidel algorithm* ;
3. for each linear system involving the diagonal blocks $A_{g,l}$ of A_g , we use an iterative block Gauss-Seidel algorithm ;
4. we solve exactly each linear system involving the diagonal blocks $A_{g,l}^d$ of $A_{g,l}$ with a cholesky factorization as $A_{g,l}^d$ is a band matrix (tridiagonal for RT0, pentadiagonal for RT1).

So, the overall algorithm is made of four iterative imbricated loops. For our applications, we obtain the best performance in terms of CPU time for a given accuracy while fixing the number of iterations of all Gauss-Seidel loops to one.

2.2 The domain decomposition method

The domain decomposition method we propose is based on the introduction of Lagrange multipliers [8] to deal with different numerical approximations between two adjacent subdomains. At each inverse power iteration, for a partition composed of two subdomains, (2) is replaced by:

$$\begin{pmatrix} A_{d=1} & 0 & C_{\Lambda \rightarrow 1} \\ 0 & A_{d=2} & C_{\Lambda \rightarrow 2} \\ C_{\Lambda \rightarrow 1}^t & C_{\Lambda \rightarrow 2}^t & 0 \end{pmatrix} \begin{pmatrix} X_{d=1} \\ X_{d=2} \\ \Lambda \end{pmatrix} = \begin{pmatrix} S_{d=1} \\ S_{d=2} \\ 0 \end{pmatrix} \quad (3)$$

where X_d contains the unknowns of the subdomain d . Λ contains the unknowns associated with the Lagrange multipliers for the matching of the unknowns X_d at the interface. The matrices $C_{\Lambda \rightarrow d}$ come from the coupling between Λ and X_d .

To solve (3) a Krylov based Uzawa algorithm was proposed [1]. In practice, for our context of inner-outer iteration, this Krylov algorithm is less efficient than the Algorithm 1 based on the Minimal Residual algorithm [5].

*In SP1 case, this step is skipped.

Algorithm 1: Interface algorithm

with the following notations :

$$\begin{aligned} \Lambda &= \Lambda_0; \\ X &= A_{DD}^{-1}(S - C\Lambda); \\ r &= C^t X; \\ \text{while } ! \text{Convergence do} \\ & \quad \hat{X} = A_{DD}^{-1}(Cr); \\ & \quad M_r = C^t \hat{X}; \\ & \quad \rho = -\frac{\langle r, M_r \rangle}{\langle M_r, M_r \rangle}; \\ & \quad \Lambda = \Lambda + \rho r; \\ & \quad r = r + \rho M_r; \\ & \quad X = X - \rho \hat{X}; \\ \text{end} \end{aligned} \quad \begin{aligned} C &= \begin{pmatrix} C_{\Lambda \rightarrow 1} \\ C_{\Lambda \rightarrow 2} \end{pmatrix}; \\ A_{DD} &= \begin{pmatrix} A_{d=1} & 0 \\ 0 & A_{d=2} \end{pmatrix}; \\ S &= \begin{pmatrix} S_{d=1} \\ S_{d=2} \end{pmatrix}. \end{aligned}$$

To compute the product by A_{DD}^{-1} , we use in parallel the sequential solver in each subdomain. Compared with the sequential monodomain algorithm, we finally add one loop to get convergence of the minimal residual algorithm. For this loop, we consider a fixed number of iterations in the same way as the sequential monodomain solver.

3 Experimental results

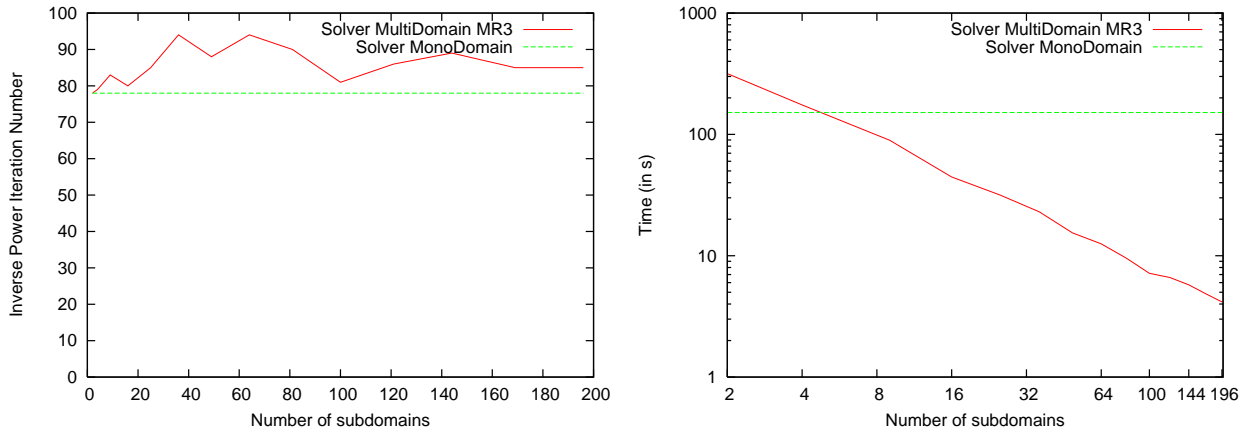


Figure 1. Behavior of the Minimal Residual algorithm with 3 iterations (MR3)

To validate the proposed method, we consider the two groups homogeneous IAEA-3D benchmark [3]. For a SP1-RT0 pin by pin computation, the computational mesh used is made of $289 \times 289 \times 38$ cells. The inverse power algorithm has the following convergence criteria:

$$\frac{\|S_n - S_{n-1}\|}{\|S_n\|} < 10^{-6} \quad \text{and} \quad \frac{|\lambda_n - \lambda_{n-1}|}{|\lambda_n|} < 10^{-6}$$

where S_n and λ_n are the source of fission and the estimated eigenvalue at the iteration n .

The tests have been performed on a cluster with 208 nodes (each with 2 Intel Xeon processors, 3.40GHz, 2MB Cache and with 4 GB PC3200 DDR2). The MPI Implementation used is MPICH 1.2.7 with an

Infiniband(openib-2.0.5) network. During batch submissions, one node is assigned to each subdomain to avoid concurrency for memory access.

We have fixed to three the maximum number of iterations of the interface solver (algorithm MR3). So, the domain decomposition method brings to four more amount of computations than the sequential monodomain solver at each power iteration. Consequently, the best efficiency we can expect is 25%. We have performed the parallel domain decomposition method from 2 to 192 subdomains.

First, we have checked that the domain decomposition method brings to the same solution as the sequential solver for the requested accuracy. Secondly, we report on Figure 1, for each subdomains partition, the number of power iterations and the execution time. The execution time of the sequential monodomain solver is reported also. On the one hand, the introduction of the domain decomposition leads to a small increase ($< 20\%$) in the number of power iterations. On the other hand, the efficiency of the method is constant and nearly equal to 21%, which is rather good compared to the best we can expect. This ensures the good quality of the parallel implementation. In other words, five nodes are required to get a parallel domain decomposition solver faster than the sequential monodomain one. But once this price is accepted, we get a good parallel algorithm : the execution time decreases proportionally with the number of processors (See Figure 1 on right with log/log scale).

We obtain similar results while using the RT1 finite element approximation. Besides, the method allows us to make computations we are not able to run with the sequential monodomain solver due to memory consumption. Finally, we expect to confirm the efficiency of the domain decomposition method for SP1/SP3 homogeneous and heterogeneous computations coming from industrial applications.

4 Conclusions

The proposed domain decomposition method in the difficult context of an approximate resolution of each linear system at each power iteration reveals very reliable and efficient on SP1 homogeneous pin by pin computations. In the future, we plan first to reach on the limitations of the method while using a machine with more nodes like BlueGene P. On the other hand, it is necessary to benefit from all the cores of a node. So, we propose to use preliminary work made by our team on the improvement of the sequential solver on a node made of some cores or Graphical Processors Units (GPU). Thus, we hope to reach a speed up of one hundred on a two groups SP1 homogeneous pin by pin 3D computation.

Secondly, we plan to develop a new domain decomposition method where the loop on the subdomains is located at the lowest level of the power algorithm (that is to say under the loop on the current directions). So, we can solve exactly the interface problem as it is made of many decoupled 1D interface problems, and consequently ensure to do exactly the same computations as the sequential solver at each power iteration. We hope to improve too the speed up of the domain decomposition method.

References

- [1] Maxime Barrault, Bruno Lathuilière, Pierre Ramet, and Jean Roman. A domain decomposition method applied to the simplified transport equations. In *Proceedings IEEE CSE'08, 11th IEEE International Conference on Computational Science and Engineering, 16-18 July 2008, São Paulo, SP, Brazil*, pages 91–97, 2008.

- [2] A. Ponçot L. Plagne. Generic programming for deterministic neutron transport codes. In *In Proceedings of Mathematical and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications, Palais des Papes, Avignon, France*, page 10 p., M & C 2005.
- [3] B. Micheelsen. 3D IAEA Benchmark Problem. Technical report, IAEA, 1977.
- [4] C.G. Pomraning. Asymptotic and variational derivation of the simplified pn equations. *American Nuclear Energy*, 20, 9, pages 623–637, 1993.
- [5] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.
- [6] D. Schneider. *Éléments finis mixtes duaux pour la résolution numérique de l'équation de la diffusion neutronique en géométrie hexagonale*. PhD thesis, Université Paris VI, 2001.
- [7] Henk A. van der Vorst. Computational methods for large eigenvalue problems. In J.L. Lions RG. Ciarlet, editor, *Handbook of Numerical Analysis*, volume 8, pages 3–179. North-Holland (Elsevier), 2002.
- [8] B.I. Wohlmuth. *Discretization methods and iterative solvers based on domain decomposition*. Springer, 2001.