M&C 2023 - The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering · Niagara Falls, Ontario, Canada · August 13 – 17, 2023

Distributed Neutron Transport based on Discontinuous Galerkin and Spherical Harmonics Approximations

K. Assogba[1,2], L. Bourhrara[1], and G. Allaire[2]

[1]Université Paris-Saclay, CEA, Service d'Études des Réacteurs et de Mathématiques Appliquées
91191 Gif-sur-Yvette, France

[2]CMAP, École Polytechnique, Institut Polytechnique de Paris
91128 Palaiseau, France

kenneth.assogba@cea.fr, lahbib.bourhrara@cea.fr, gregoire.allaire@polytechnique.fr

ABSTRACT

High-order transport solutions are required for precise lattice and reactor core analysis. In this work, we use a combined discontinuous Galerkin – spherical harmonics method to solve the neutron transport equation on unstructured, non-conforming and curved meshes. The matrices resulting from this approximation are mainly block diagonal, moreover cells identical up to a translation yield to the same elementary matrices. We leverage this benefits to build a distributed assembly-free transport solver based on an algebraic domain decomposition. The use of non-blocking communications allows to overlap computation and communication. Numerical experiments on reactor cores and fuel assemblies have been carried out up to 10^3 CPU-cores. The results obtained show that the method is robust and scalable. In particular, we obtain the eigenvalue of the 3D-C5G7 core with an error of 12 pcm in 1 minute, reaching a 19x speedup regarding shared-memory calculations.

KEYWORDS: parallel neutron transport; distributed Krylov method; MPI; OpenMP; asynchronous communications

1. INTRODUCTION

Modern computing clusters are made up of independent computing nodes linked together by a fast interconnection network. Programmatic access to these – so-called distributed – systems can be achieved through MPI (Message Passing Interface). The MPI standard defines a library of functions that allow parallel programming by exchanging messages. Within each host device having continuous memory addressing (also referred to as NUMA node) it is possible to spread a task across threads through OpenMP compiler directives.

From an algorithmic point of view, domain decomposition methods (DDM) such as Schwarz iterative algorithm are widely used, and have proven they parallel efficiency for reactor physics problems [1]. In this work we exploit an algebraic approach of domain decomposition. After discretization of the neutron transport equation, one obtains a linear system or a generalized eigenvalue problem:

$$Ax = b \quad \text{or} \quad Ax = \lambda Q x. \tag{1}$$

Once the spatial mesh has been partitioned and shared over independent processes, each process owns a sub-matrix of the global matrix, a part of the right hand side and a part of the unknown vector. In this way, the global system can be solved concurrently on all processes. When the problem is

solved by a Krylov method, one uses the term *distributed Krylov subspace method* to refer to the method described above. Very popular softwares in the scientific computing community supporting distributed Krylov subspace methods are PETSc and Trilinos.

In this note, we describe the implementation of this method in the transport solver NYMO [2], [3], [4]. Developed inside the CEA reactor physics industrial platform APOLLO3 ®, NYMO is based on discontinuous Galerkin approximation in space and spherical harmonics or P_N method for the angular variable discretization. The support of unstructured, non-conforming and curved meshes enables the modeling of a wide range of reactor physics problems. The results presented here are from the standalone version of NYMO using the SPMD (single program multiple data) pattern.

The aim of this paper is to highlight that this algebraic domain decomposition approach is well suited for DG discretized transport problems. A parsimonious choice of basis functions allows to solve the resulting linear system without global matrix assembly. The coupling between subdomains is easily carried out during the matrix-vector products. Moreover, the positioning at the algebraic level permits to consider all meshes, even non-conforming ones. Finally, the implementation is minimally intrusive and the parallel execution requires no additional effort from the end user.

In the remaining part of this paper, we describe the important steps of the algorithm and evaluate its strong scalability behavior.

2. DISTRIBUTED KRYLOV SOLVER

The NYMO numerical scheme for the linear Boltzmann equation has been introduced in [2], [3] and [4]. Let us here describe the parallelization strategy from an algebraic viewpoint.

The move to distributed systems requires effective design of data distribution and communication between processes. A straightforward advantage is that it allows extremely large problems to be solved, since each process holds only a small part of the original data. The implementation is simplified in the context of discontinuous Galerkin approximation. In fact, each mesh element is only coupled with its neighboring cells through a face. Thus the matrices resulting from this approximation are essentially block diagonal, with off-diagonal blocks coupling neighboring cells. As a corollary, after data distribution, each domain only needs to fetch the flux from the cells that share a face with its local cells.

Let us denote by n_d the number of MPI process in the global communicator. The parallel resolution start by partitioning the initial mesh into n_d subdomains. Two partitioning strategies are proposed, a *simple* one based on mesh numbering and a *geometric* one based on Hilbert space-filling curves [5], Figure 1.

Next, each process collects the cross-sections and cells having a part of their stencil straddling two domains, also referred to as ghost or halo cells. Then the sub-vectors and sub-matrices corresponding to the subdomain are built in an assembly-free fashion. When solving the linear system (1) inside the power iteration algorithm, all process work concurrently. The neighboring domains are coupled, and exchange upwind flux on their common interfaces. At the algebraic level, this exchange is done at each sparse matrix-vector product (SpMV). The use of non-blocking communications allows communication time to be covered by computing time. The asynchronous sparse matrix-vector product (SpMV) attached to the Krylov solver takes place in three stages.

- Phase (i): The coupling between the different domains is done via the ghost cells. The transmission of the upwind flux at the interface is done at each SpMV in a non-blocking manner.

- Phase (ii): Since the communications at Phase (i) are asynchronous, the calculation continues and the part of the product involving only local degrees of freedom is calculated.
- Phase (iii): It is then necessary to check the end of Phase (i), and then to calculate the part of the product involving the received data.

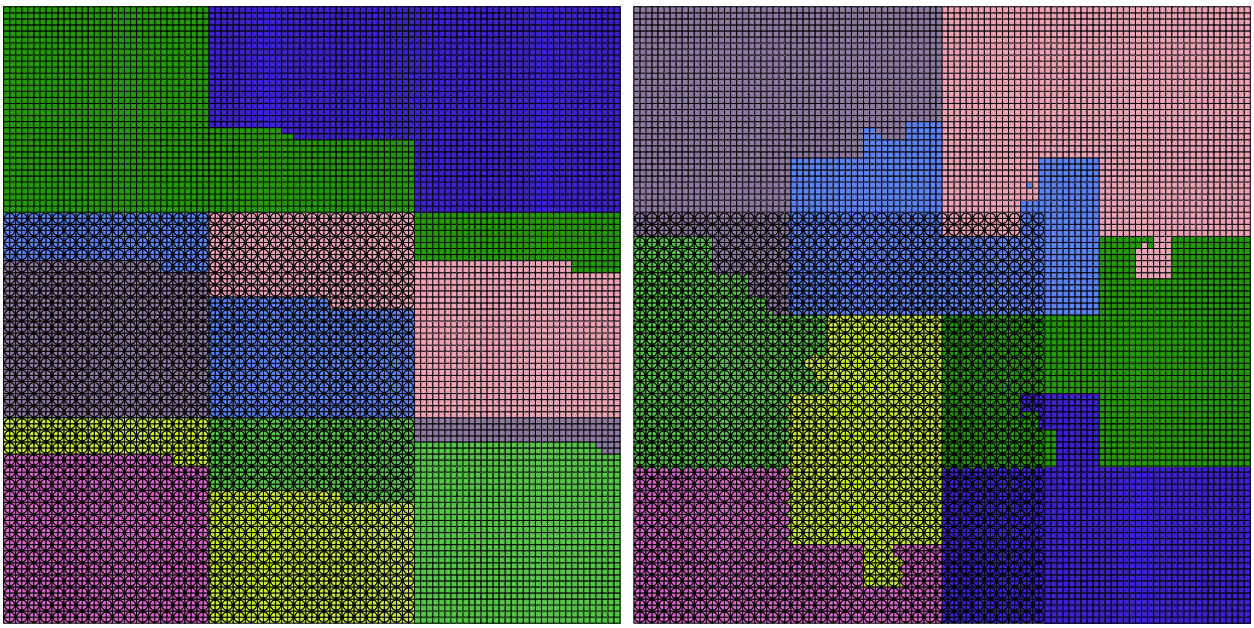

Figure 1: Simple (left) and geometric (right) partitioning of one axial slice of the C5G7 core into 8 domains.

3. NUMERICAL EXPERIMENTS

All calculations are performed on the Rome partition of the CEA cluster Orcus. Each Rome computing node houses two AMD EPYC 7352 2.3 Ghz sockets equipped with 24 cores each. The shared-memory only calculation are performed on one socket. For hybrid MPI + OpenMP jobs, each MPI process is binded to one socket.

The cases used previously to evaluate the capabilities of the `NYMO` solver [3] [4] have been rerun to evaluate the parallel efficiency of the approach described in Section 2. We consider the Rodded B case of the C5G7 3D core in order to test the strong scalability and compare the *simple* and *geometric* partitioning, Figure 1. The Krylov solver that has been moved to distributed memory is BiCGSTAB.

Table 1 shows that we obtain a reduction of the calculation time to 1 min 46 sec which corresponds to a speedup of 18.7 and an efficiency of 58%. Thus the speedups obtained are very encouraging. The method is robust, in the sense that the variations on the solution obtained when the number of processes is increased are of the order of machine precision. This is because the calculations performed are exactly the same, only the order of operations changes. When dealing with a large number of domains, most of the latency comes from the global synchronization operations (`MPI_Allreduce`). Indeed, they constitute a barrier, which acts as a bottleneck. In particular, this case being small (15028 cells per radial plane), the limits of possible gain are reached very

quickly. Actually, the computation time is already very low, so the total time is dominated by the communication time. A larger test case would allow a better evaluation of the parallel efficiency.

Table 1: Strong scaling experiment on up to 32 domains (768 CPU-cores). 0* means shared memory only calculation.

		Simple partitioning			Geometric partitioning		
n_d	#core	time (s)	speedup	efficiency (%)	time (s)	speedup	efficiency (%)
0*	24	1982	-	-	1982	-	-
2	48	978	2.1	101	1038	1.9	95
16	384	193	10.3	64	268	7.4	46
32	768	106	18.7	58	116	17.1	53

4. CONCLUSIONS

In this paper, we have described a domain decomposition method suitable for solving the transport equation in distributed memory. We also present a mesh partitioning strategy based on Hilbert space-filling curves. The method is well adapted for DG based transport discretizations and is minimally intrusive in terms of implementation.

Numerical experiments show that the method is very robust, as we observe little variation in the solutions obtained when increasing the number of computational processes. Moreover, it enjoys a good scalability. These advances are also valuable from the point of view of the end-user, as the use of this tool requires no additional effort on their part.

Future work involves the evaluation of this approach on larger test cases, in particular the direct calculation of a whole core without energy condensation. In addition, it would be valuable to integrate distributed and scalable preconditioners and acceleration strategies.

REFERENCES

[1] E. Jamelot and P. Ciarlet. "Fast Non-Overlapping Schwarz Domain Decomposition Methods for Solving the Neutron Diffusion Equation." *Journal of Computational Physics*, **volume 241**, pp. 445–463 (2013).

[2] L. Bourhrara. "A New Numerical Method for Solving the Boltzmann Transport Equation Using the PN Method and the Discontinuous Finite Elements on Unstructured and Curved Meshes." *Journal of Computational Physics*, **volume 397** (2019).

[3] K. Assogba, L. Bourhrara, I. Zmijarevic, G. Allaire, and A. Galia. "Spherical Harmonics and Discontinuous Galerkin Finite Element Methods for the Three Dimensional Neutron Transport Equation: Application to Core and Lattice Calculation. (Submitted)." (2022).

[4] K. Assogba, L. Bourhrara, I. Zmijarevic, and G. Allaire. "Precise 3D Reactor Core Calculation Using Spherical Harmonics and Discontinuous Galerkin Finite Element Methods." In *Proceedings of International Conference on Physics of Reactors 2022 (PHYSOR 2022)*, pp. 1224–1233. American Nuclear Society, Pittsburgh, PA, United States (2022).

[5] D. Hilbert. "Über die stetige Abbildung einer Linie auf ein Flächenstück." In D. Hilbert, editor, *Dritter Band: Analysis - Grundlagen der Mathematik - Physik Verschiedenes: Nebst Einer Lebensgeschichte*, pp. 1–2. Springer, Berlin, Heidelberg (1935).