Domain decomposition method for 2d and 3d transport calculations using hybrid mpi/openmp parallelism

R. Lenain, E. Masiello, F. Damian, R. Sanchez

ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method • Nashville, TN • April 19-23, 2015, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2015)

DOMAIN DECOMPOSITION METHOD FOR 2D AND 3D TRANSPORT CALCULATIONS USING HYBRID MPI/OPENMP PARALLELISM

R.Lenain, E.Masiello, F.Damian, R.Sanchez
CEA Saclay – DEN/DANS/DM2S/SERMA,
F-91191 Gif-sur-Yvette, France

roland.lenain@cea.fr; emiliano.masiello@cea.fr; frederic.damian@cea.fr; richard.sanchez@cea.fr

ABSTRACT

In this paper we analyze the efficiency of the Domain Decomposition Method associated to the Coarse Mesh Finite Difference acceleration. We evaluate the effectiveness of the algorithm for shared memory parallelism. We also present the advantages of the Hybrid OpenMP/MPI parallelism to perform high-fidelity large scale calculation. We show that CPU time for a best-estimate 2D whole core calculation can be reduced from several days to few minutes on computer cluster. Finally high-fidelity 3D full assembly color-set calculation is compared to Monte Carlo simulation. This particular test shows the challenges for advanced neutron transport simulations in 3D. This work is part of the development program for the neutronic code APOLLO3®.

Key Words: Domain-Decomposition, Coarse-Mesh Finite Difference, Hybrid parallelism, Discrete Ordinates

1 INTRODUCTION

Massively parallel computing is the nowadays solution to extend the range of solvable numerical problems. However, massively-parallel high-performance machines are not available for daily calculation and this is why intermediate solution would be useful. The objective of this work is to realize a proficient use of daily computational resources for a “quick” best-estimate transport calculation to overcome infinite lattice approximation. The latter approximation, which is generally employed for the generation of the cross section libraries used in the reactor analysis, is a critical source of error in the core simulations. The computational resource we consider as a starting point is a desk computer with multi-core shared memory architecture. In this case the number of cores is of the order of 10 with a limited amount of memory available which is a notable constraint for the problem we aim to solve. For 2D full core depletion, the amount of memory required is about 1TB due to the huge number of material involved. Hybrid MPI/OpenMP parallelism has been developed to give access to distributed memory on computer cluster.

A best-estimate deterministic calculation needs hundreds of energy groups, a sizable number of directions and a fine description of the geometry. In the case of a typical PWR 2D assembly calculation, we use 281 energy groups, 40 directions and around 300 regions. A single processor calculation of this assembly needs tens to hundreds of seconds to converge and around 1Gb of memory. The amount of memory is mainly due to numerical coefficients and cross section library. The most important costs regarding CPU time consumption are the transport

sweep and the fissions and, especially, the scattering sources computation. Indeed for the latest, the whole multigroup flux and transfer cross sections are embedded in the routine performing this part. If the size of the cross section library and the flux is larger than the processor cache, it leads to memory swap and pagination, resulting in a slowdown of the runtime.

These observations lead us to consider a Domain Decomposition Method (DDM) where the global geometry, i.e. a color-set up to the whole core, is split in several sub-geometries where a full multigroup source problem is solved. This is based on the assumption that for a given computer architecture, we can find an optimal problem size to use most efficiently a single processor. It results in a set of non-overlapping subdomains coupled by the interface angular flux. The exchange of the interface angular fluxes guarantees the neutron conservation and the preservation of the quality of the global solution. In this DDM, the size of the subdomains has to be chosen regarding both the optimization of the cache of the processors and the convergence ratio. It is known that DDM without overlapping suffers from a poor convergence ratio. As a consequence, the Coarse Mesh Finite Difference (CMFD) acceleration method at the global level has been implemented to speed-up the communication between subdomains. More than using efficiently computer's resources, the DDM offers the possibility to parallelize the subdomain calculations. Indeed, each calculation unit is independent from the others. One can also note that the independent multigroup iteration in each subdomain limit the communications between subdomains and thus minimize the number of synchronizations between the parallel processes. The new incoming interface flux is only set once the whole multigroup problem has been solved for each subdomain. The DDM for full multigroup subdomain problems accelerated by the CMFD seems to be a good compromise to perform large scale transport calculation using efficiently computer's resources. This work is partially inspired by the research of professor Aragones and his research team. [1] [2]

Firstly, we present the efficiency of the DDM+CMFD regarding the use of CPU and subdomain size with single processor tests. The subdomain calculations have been parallelized with shared memory parallelism (OpenMP) which is the most common for optimization on standard workstations (load balance, exchanges between subdomains and CMFD acceleration). Scalability tests have been performed on a standard workstation. The drawbacks of shared memory parallelism (lack of memory and number of cores) lead us to consider Hybrid MPI/OpenMP parallelism. The benefits are highlighted by realistic 2D whole core transport calculation where both CPU time and memory are constraints for high fidelity depletion calculation. Finally, we present here a 3D full assembly color-set calculation which permits to face present challenges for neutron transport simulation.

2 CALCULATION ALGORITHM

The method consists in dividing the entire problem domain into N non-overlapping subdomains with $\alpha=1,\ldots,N$ denoting the subdomain index. The interfaces between subdomains are reduced to shared interface surfaces [3] [4]. The angular flux is transmitted between neighboring subdomains to guarantee the conservation of the solution of the global problem. Each subdomain is considered as a boundary-value multigroup problem with boundary and volume fixed source.

Our iteration strategy consists in 4 steps detailed in [5]:

- For the initialization, the subdomain boundary fluxes and volume sources are initialized from un-converged infinite lattice calculation. $Q_\alpha = \frac{F_\alpha \phi_\alpha}{\lambda_\alpha}$ is the fission sources with a locally computed eigenvalue.

- The second step consists in solving a multigroup fixed-source problem in each subdomain with the fine transport:

$$\begin{cases} (\mathrm{L}-\mathrm{H})_\alpha \Psi_\alpha(x) = Q_\alpha(x), & x \in \mathrm{X}_\alpha \,, \\ \Psi_{\alpha-}(x) \,, & x \in \partial \mathrm{X}_{\alpha-} \,. \end{cases} \quad (1)$$

The operator $(L-H)_\alpha \equiv \Omega.\nabla + \Sigma_\alpha(x) - \int dx' \Sigma_\alpha(x' \to x)$ is locally inverted to provide a fine transport solution in each subdomain. $\Psi_{\alpha-}(x)$ is the angular entering flux at subdomain boundary. The boundary conditions of each subdomain are given either by the outgoing flux of neighbor subdomain (here noted β) or by the boundary conditions given on the boundary of the domain,

$$\Psi_{\alpha-}(x) = \begin{cases} \Psi_{\beta+}(x), & x \in \partial \mathrm{X}_\alpha \cap \partial \mathrm{X}_\beta \,, \\ \Psi_{-}(x), & x \in \partial \mathrm{X}_\alpha \cap \partial \mathrm{X} \quad . \end{cases} \quad (2)$$

- The local transport solution feeds a global whole-domain non-linear diffusion operator. This is the CMFD operator defined on a coarser phase space Y with fewer energy groups and homogenized regions. The inversion of the coarse operator permits to stabilize both scattering and fission source distribution over the whole domain and provides new eigenvalue for the local fine transport calculations.

- Finally, the iteration ends by the correction of the local transport volume and boundary fluxes as follow:

$$\Psi_\alpha(x) = \Psi_\alpha(x) f_\alpha \quad ; \quad \Psi_{\alpha+}(x) = \Psi_{\alpha+}(x) f_\alpha \quad ; \quad f_\alpha = \frac{\phi_{cmfd}(y)}{\int_{\mathrm{y}} dx \Psi_\alpha(x)} \quad (3)$$

The eigenvalue in each subdomain is provided by the CMFD solution $\lambda_\alpha = \lambda_{cmfd}$ in order to compute the new volume source $Q_\alpha = \frac{F_\alpha \phi_\alpha}{\lambda_\alpha}$.

Convergence is checked for eigenvalue, the fission source distribution and the interfaces currents between subdomains as described in [5].

The fine transport problem is solved with the IDT S_N solver [6] [7] [8]. The transport operator is discretized with the method of short characteristic on Heterogeneous Cartesian Cells which allows pin-by-pin transport calculation without homogenization. We presented the accuracy of the DDM implemented in IDT in previous work [5].

3 EFFICIENCY OF THE DDM ON 3X3 COLORSET

A numerical experiment has been set on a 3x3 color-set based on 16x16 pin-cells PWR-like assemblies, as presented in Fig. 1. We try to find the best compromise to optimize the utilization of the cache of the processor. Moreover, we try to experiment the gain offered by using the parallel algorithm on desk computer.

The color-set does not present any spatial symmetry due to the asymmetric burn-up distribution. The assembly cross sections are self-shielded and discretized with 281 energy group

based on the SHEM discretization and a P1 scattering kernel. [9] The cross sections are self-shielded at the assembly level and regrouped in 7 types of pin-cells for each burn-up step as described in in Figure 1b. The total number of region is 13392 for the cluster geometry. We use linear spatial expansion and 40 directions and 3 sub-surfaces per pin-cell for the discretization of the interface angular flux.

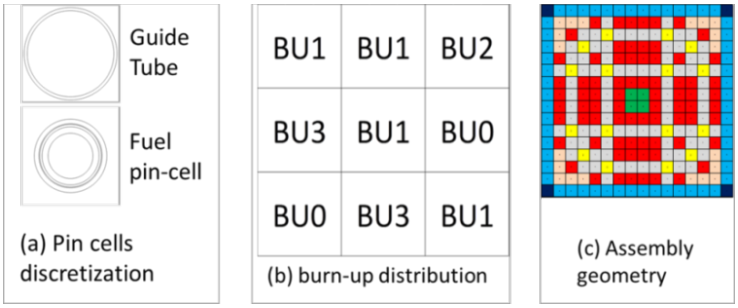

Figure 1. Description of the cluster

3.1 Intrinsic Efficiency of DDM: Single Processor Run

This test aims to evaluate the performances of the DDM regarding the use of processor cache and memory pagination. For this purpose, the number of floating-point operations to perform transport calculations is fixed in 5 transport sweeps per each group and in 10 multigroup iterations. We defined one material per spatial region, i.e. 13392 materials and associated sets of cross sections. This accounts for about 4.6 GB of memory. The calculations are run on a 6-proc Intel(R) Xeon(R) X2620 2.00GH 15.3 MB of cache per processor, 64 GB of memory which is one of the standard workstation available in our laboratory. The same calculation is performed with different Domain Decompositions (DD) going from 1 single domain up to 2304 subdomains (1 pin cell per subdomain). The calculations are run on a single processor (no parallelism). The test is oriented to search a set DD in which the method guarantees the resources optimization in terms of memory utilization and CPU work. The efficiency, defined as the ration between the elapsed time of the direct calculation (1 single domain) and the elapsed time of the calculation performed with n subdomains, $E(n) = T(1)/T(n)$, has been measured and showed in Figure 2a). The picture shows that the efficiency is above 1 for subdomains sized from 16x16 pin-cells to 4x4 pin-cells respectively composing 9 to 144 subdomains. The over-efficiency has a maximum of 7% at 12x12-pin-cell sized subdomains. The total time has been decomposed in the *inner-time*, which includes 1 spatial and angular transport sweep for each energy group, the *source time* which accounts for multigroup scattering source computation, the *exchange time* which takes into account the exchange of boundary fluxes between subdomains and, finally, the *CMFD time* which includes the CMFD matrix construction, the CMFD solver iterations and the update of the transport fission integrals.

In Figure 2b), we analyze the single behavior of the above mentioned elapsed times. The computationally demanding parts of the calculation are the inner iterations and the scattering sources computation. The first depends especially on the number of spatial regions per subdomain while the second depends on the size occupied by the transfer matrix cross sections. Both times benefits from DD: as the Figure 2b) shows, the inner iteration time starts to decrease from 9 subdomains while the scattering source computation starts to decrease from 16 subdomains. The inner transport sweep is mainly a floating-point demanding area whereas the source computation stresses the memory utilization.

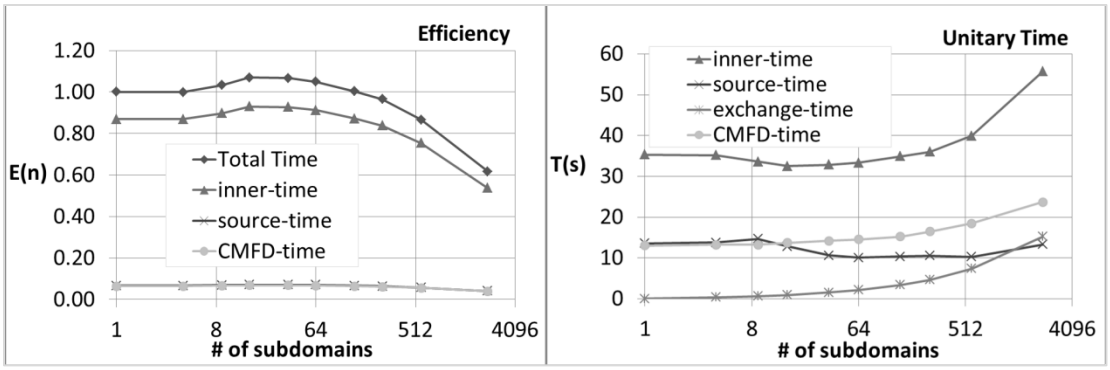

Figure 2. (a) Efficiency of the DDM. Fig (b) Multigroup time analysis

A deeper understanding of the benefits of the DDM for the source computation is given in Table I. In this table we estimate the average memory occupation for the scattering source computed for one arrival group and the memory occupation of the multigroup flux with P1 expansion. The total memory occupation of the subroutine devoted to such calculation embeds both multigroup flux and the transfer matrix together with several auxiliary arrays. The memory occupation becomes lower than the cache size from 16-subdomain. One can see the same kind of behavior for the transport sweep with a gain about the 10% of computational time (Figure 2a)). The pictures show that the calculation time increases when the subdomains are too small. The processor loads too small data in its cache while communications increase. The test suggests us that an optimal configuration for the DDM can be run from 256-pin-cell per subdomain down to 16-pin-cell per subdomain for a reference 2D calculation.

Table I. Estimation of memory occupation by 1 subdomain during scattering source computation in MB (evaluation for 1488 regions, 3 spatial moments, 6 angular moments)

# of subdomains	1	4	9	16	36	64	144
# of regions / subdomain	13392	3348	1488	837	372	204	93
Memory per arrival group for transfer matrix / subdomain (MB)	80.4	20.1	8.9	5.0	2.2	1.2	0.6
Memory for multigroup flux / subdomain (MB)	67.0	16.7	7.4	4.2	1.9	1.0	0.5
Total amount of memory per arrival group / subdomain (MB)	156.2	39.1	17.4	9.8	4.3	2.4	1.1

3.2 Performances of CMFD Acceleration Applied to DDM: Single Processor Run

The DDM without overlapping suffers from poor convergence ratio. We attempt to mitigate this lack of convergence with CMFD acceleration at global level [10]. In previous work [5] we presented some optimization of the CMFD parameters in term of the size of the coarse spatial regions and the size of the coarse energy groups. In this study we find out that an optimal 2x2 homogenized pin cells coarse mesh and a 26-group coarse energy mesh can efficiently accelerate the DDM iterations. In the following test we run the previous DD set of calculations until convergence on a single processor. The convergence criteria are 10^{-5} for the eigenvalue, 10^{4} for the fission source and 10^{-3} for the interface currents.

The Figure 3 shows the elapsed time for the 3x3 colorset for different DD from 1 single domain to pin-by-bin DD. We set 2 coarse spatial meshes; the first is a pin-by-pin homogenization while the second is a 2x2 pin-cells homogenization. As the picture shows, the CMFD mesh influences the convergence of the method. From 2x2 subdomains up to 12x12 subdomains, the DDM calculation time is almost identical to the direct calculation time. This implies the possibility to achieve the ideal scalability. In this range, the CMFD stabilizes the number of outer power iterations despite of the number of subdomains.

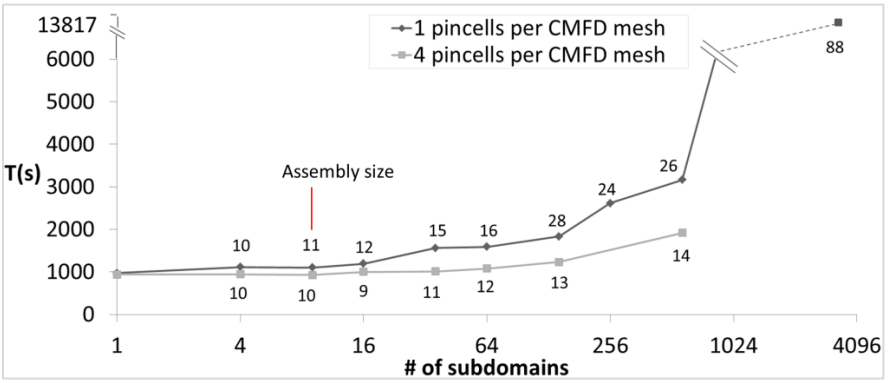

Figure 3. Elapsed time vs. # of subdomains. The number of power iterations is indicated for each calculation.

It is also interesting to note that when subdomain is too small, the number of iterations increases. As the size of the subdomains decreases, the local scattering sources are more influenced by the entering angular flux from the neighbor subdomains. We found an optimal subdomain size around the typical PWR assembly size where the slowing down of the neutrons is very important in the water. That could be quite different for other kind of reactors where the optical thickness is lower. 2D PWR core would imply o(200) assemblies. Since the DDM is efficient down to 36 pin-cells per subdomain, which counts for 8 subdomains per assembly, the number of subdomains for a full-core DDM calculation would be o(1000). This allows for a proficient application of the DDM to large scale parallel computer before reaching the scalability limit.

4 OPENMP SCALABILITY OF DDM+CMFD

A series of weak and strong scaling studies were performed with shared memory parallelism (OpenMP). These tests have been performed on a 12-proc Intel(R) Xeon(R) X5645 2.40GH 12.3 MB of cache per processor, 24 GB Memory. We fixed the number of operations to perform the transport calculations (5 angular-spatial sweeps and 10 multigroup iterations) to focus on the parallel performances. The number of threads for both weak and strong scaling varied from 1 to 24. For the weak scaling, the amount of work per thread is constant. The number of threads is equal to the number of subdomains. The geometry is composed of *n* assembly-sized subdomains, with *n* going from 1 to 24. For the strong scaling, the size of the problem is always the same (3x3 colorset) but the number of subdomains increases from 1 to 24. The number of threads is always equal to the number of subdomains. The speed-ups for weak scaling and strong scaling are respectively defined as:

$$S_{weak}(n) = n\,T(1)/T(n) \quad and \quad S_{strong}(n) = T(1)/T(n) \qquad (4)$$

Here *n* is the number of threads.

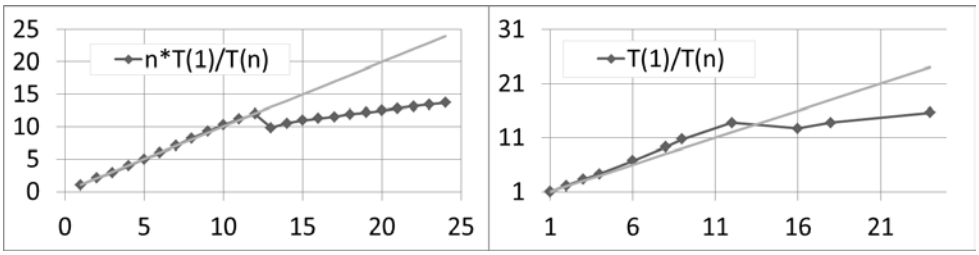

Figure 4. (a) weak scaling speed-up and (b) Strong scaling speed-up

The Figure 4a) and Figure 4b) show respectively the speed-up for the weak and the strong scaling. The processor has only 12 cores and the hyper-threading is activated, this justifies the gap occurring beyond 12 threads. In both cases, the linearity is respected until hyper-threading is reached. The reason for super-linear scaling is certainly explained by the better use of cache, highlighted in the previous chapter. When the hyper-threading is activated, even if the curves deviate from the ideal line, the speed-up reaches 15 for 24 threads. This behavior of shows us that efficient daily calculations can run with OpenMP parallelism on desk computer when the total amount of memory can be supported by a single node.

5 2D FULL CORE CALCULATION – OPENMP/MPI HYBRID PARALLELISM

In [5] we presented a 2D configuration of a large LWR with heavy reflector at BOC [11], the calculation has been parallelized using the OpenMP library. In that case we limited the number of materials to about 300. For a full depletion of a 2D core, one needs to store almost half a million of materials which needs 281 energy groups and P1 expansion for a total of 1TB. This large amount of memory is not available on a single workstation. The limitations of the shared memory parallelism lead us to consider a Hybrid OpenMP/MPI parallelism which gives access to this huge amount of memory. As depicted in Figure 5 the core is loaded with 241 fuel assemblies surrounded by a heavy baffle (homogeneous media including water and steel). All cross sections were self-shielded at the assembly level with the lattice code APOLLO2 and stored in 281-groups cross-section libraries without homogenization. There are 9 types of assemblies: 5 UOx assemblies and 4 UOx-Gd assemblies at different burn-up.

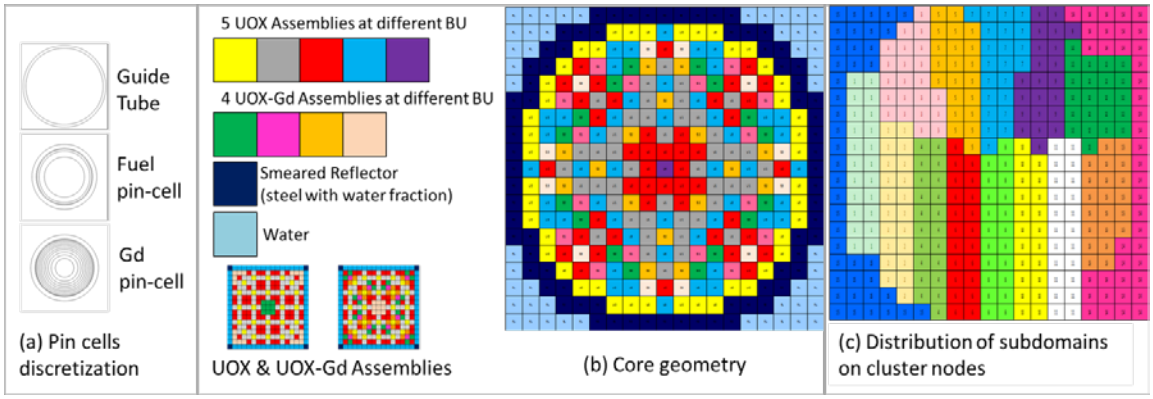

Figure 5. Description of the a large LWR core (heavy reflector) and the mapping between subdomains and nodes

The UOx pins were discretized with 6 regions per pin cell: 4 self-shielded rings for the fuel and 2 materials respectively for the clad and the coolant. A total of 130 materials were

defined for the 5 UOx assemblies. In the UOx-Gd assemblies, the gadolinium pin-cells were discretized with 11 self-shielded rings in the fuel and 2 regions in the clad and the coolant for a total of 160 materials for the 4 assemblies. The reflector baffle has been computed by smearing 5% of water in the steel, the cross sections have been self-shielded in a 1D simplified geometry composed by a fuel assembly and a slab of reflector. The whole geometry is described with 439,542 regions. The reference calculation was obtained by running the Monte Carlo code TRIPOLI-4® [12]. The results compared to IDT-DDM are synthetized in Table II.

Table II. Error of IDT-DDM vs. reference TRIPOLI-4® calculation

IDT – DDM	P1
Reactivity Error	-49 pcm
Max pin-by-pin Power Error	2.10%
Min pin-by-pin Power Error	-1.10%
Fraction of pin-cells in 3σ	54%

IDT-DDM calculations were run in 3 configurations. The first is an OpenMP coarse grained parallelism run on 6-proc Intel(R) Xeon(R) X2620 2.00GH 15.3 MB of cache per processor whit 64 GB of memory, and parallelized on 12 threads. In this calculation the number of materials was limited to 300 for a total of 35GB of memory. A second calculation benefits from the hybrid parallelism and is run on a 15 nodes cluster. Each node is 2x10-proc Intel(R) Xeon(R) E5-2680 V2 2.80GH with 128 GB of memory. In this calculation we use both MPI and OpenMP directives to parallelize the DD sweep. The number of materials is the same as for the previous calculation. This first comparison illustrates the memory and communication overheads due to the distributed memory architecture. The third calculation is a full depletion of the core where we stored 1 material per spatial region. The later demands the storage of 1TB in P1 expansion. We implemented the hybrid solution as a 2-level DD. The first one consists in allocating groups of subdomains on several MPI processes. The Figure 5c) shows the mapping between subdomains and nodes for the 2D core calculation. Each color corresponds to 1 MPI process. Then at the second level, the OpenMP directives solve in parallel the subdomains associated to the node. This solution is a first attempt to save some benefits from shared memory parallelism while eliminating its drawbacks. We attempted to have a good load balance by considering two assembly-sized chunks of reflector as one fuel assembly.

Table III. 2D Full core calculation. Case A: OpenMP parallelism. Case B: hybrid calculation. Case C: hybrid core depletion calculation (*includes MPI communication)

Designation	A-OpenMP	B-Hybrid	C-Hybrid
# of materials (including duplication)	292	2984	~$4x10^5$
Memory (GB)	20	20	1,000
Number of threads – Number of Nodes	12 – 1	300-15	300-15
Total elapsed time	2h35	11 min	17 min
Time for transport solver	83%	39%	44%
Time for CMFD (coef* + solv.)	15% (13.5+1.5)	60% (38+22)	50% (37+13)
Time for flux exchange*	< 1.3%	13.30%	5%
# of outer iterations	22	18	18
Total CPU time	1.25 days		

Table III shows the performances of the 3 calculations for the P1 expansion of the scattering. We analyzed the number of materials, the total memory occupation, the total elapsed time and the partial times spend in the transport sweeps, in the CMFD solver and in the interface flux exchanges.

A first comparison is made between the OpenMP calculation (columns A in the Table III) and the hybrid implementation (column B). Memory duplication is observable for the hybrid calculation dues to the copy of the transport coefficients and the cross sections. In this case, the memory overhead is negligible with respect to the storage that is needed for the flux. The hybrid parallelism seems to preserve the ideal scalability of the transport iterations. However, for the hybrid implementation, the fraction of time for flux exchanges and for the CMFD acceleration becomes important. In fact, the CMFD time occupies 60% of the total elapsed time compared to only 15% when only the shared memory is used. Moreover the time for flux exchange is not negligible anymore. We underline that, at the present stage of the implementation, the CMFD solver is not parallelized and a considerable reduction of its computational time can be gained from this. On the other hand, the absolute time for flux exchange remains constant between the OpenMP and the hybrid parallelization. Column C shows the capability to run a full depletion of the core in a reasonable amount of time. 15 nodes offer the enough memory amounts to store the cross sections for half a million of materials for a total of 1 TB. The calculations time for one burn-up steps becomes also interesting running over 17 minutes. Because of the full depletion, the time required for the computation of the transport coefficients is not negligible in this case. It takes about 100 seconds per assembly parallelized on 10 threads.

This hybrid parallelism preserves the OpenMP qualities within each node. By this way, we save the good scalability performances obtained with shared memory architecture. We save memory, avoiding copy of shared data belonging to the same node while preserving a fast interface flux communication between subdomains. The CMFD acceleration is solved by the master MPI process. This is a bottleneck since local CMFD coefficients have to be sent from all process to the master (MPI_REDUCE) before solving the diffusion system, and then the solution have to be spread to all processes from the master (MPI_BCAST).

6 3D FULL ASSEMBLY 3X3 CLUSTER CALCULATION

The Figure 6 depicts 3-D color-set geometry. It is composed by 9 PWR assemblies: 4 UOx assemblies having 3 different burn-up, 4 UOx-Gd assemblies having 4 different burn-up and one central UOx assembly with a control rod partially inserted. The assemblies are modeled in their full height, i.e. a total height of 400 cm for the fuel pins plus 20 cm of reflector at the top and at the bottom of the geometry (mix between water and steel for the reflector) for a total of 440 cm. The boundary conditions are the specular reflection on the radial boundaries while the vacuum on the top and bottom surfaces. A control rod is inserted in the central assembly at half of the total height (cf. Figure 6b)). The cross section libraries are the same as for the 2D core calculation which represents about 30 materials per assembly and 281 energy groups with P1 or P3 expansion. An additional library was generated for the rodded assembly. The latter is composed by fresh UOx fuel pins, with a B4C absorber inserted in guide tubes. The absorber pin has a coarse model composed by 2 materials, the absorbed and its cladding. A water gap between the absorber cladding and the tube has also been modeled. The upper and lower reflectors have been computed by smearing 50% of water in the steel, the cross sections have been self-shielded in a 1D simplified geometry similarly to the heavy baffle for 2D core calculation.

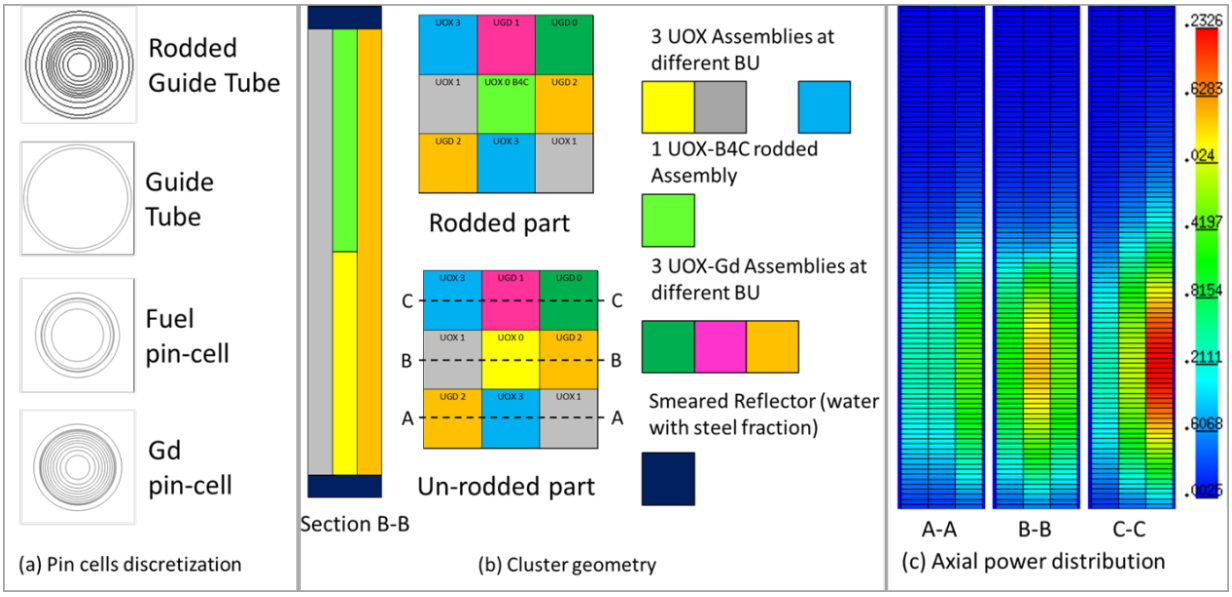

Figure 6. (a) 2D spatial discretization for the pin-cells used in the 3D Heterogeneous Cartesian Cells. (b) color-set configuration. (c) Axial power distribution per assembly

We compared our deterministic solution to a Monte Carlo simulation obtained with the TRIPOLI-4® code (version 9). For the latter calculation, the continuous-energy cross sections have been used while the statistics has been done by running ~2500 independent simulations (independent random number seed). Each simulation is characterized by 2200 batches having 10^4 particles per batch (the first 200 batches are discarded to stabilize the fission source and the eigenvalue). The runtime of a single Monte-Carlo simulation is around 1 day. A fist comparison between IDT-DDM and TRIPOLI4® in term of reactivity error and runtime is sketched in Table IV where the comparison is done using the P1 and the P3 libraries.

Table IV. IDT-DDM vs. TRIPOLI-4®, reactivity error and elapsed time

	IDT – P1	**IDT – P3**
Nb threads – Nb Nodes	198 – 11	198 – 11
K-eff (Delta Reac T4 (pcm))	1,05815 (28)	1,05816 (29)
# of outer iterations	27	26
Total time (elapsed time)	4h19	7h47
Time for transport solver	81%	87%

A deeper understanding of the quality of the solution was obtained by calculating the Monte-Carlo power distributions on three different Cartesian-meshes corresponding to various scales (fuel assembly and pin-by-pin). The first one is devoted to the average radial pin-power distribution. The whole height of the cluster is integrated while the radial mesh is discretized with 51x51 Cartesian cells at pin-cell scale. The second one is used to compare the axial power distribution. It is composed by 100 axial planes; each plane has 4 cm in height and is radially integrated over the cluster assemblies. Finally, the third one is the 3D power distribution combining the first radial description with the second axial discretization. As depicted in Figure 6 (c), the axial power distribution undergoes a 3-order of magnitude excursion between the active part and the rodded part. This kind of decoupled system is difficult to simulate for both deterministic and probabilistic codes.

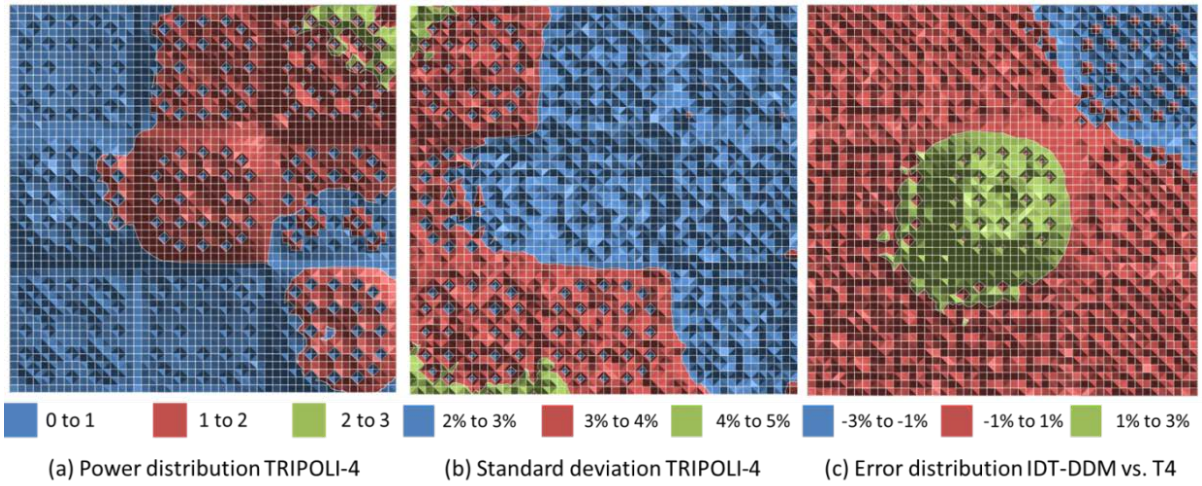

Figure 7. IDT-DDM vs. TRIPOLI-4® on the 3D rodded color-set problem

Figure 7 (a), (b), (c) depicts respectively the average radial power-distribution for the TRIPOLI4® simulation, the standard deviation associated to the calculation and the relative error on the pin-by-pin power distribution with respect to the IDT-DDM calculation. We observe that the standard deviation associated to radial power distribution of the TRIPOLI-4® calculation goes up 4.3%. With such quality of the Monte-Carlo solution, the 100% of the pin-cells fits the standard deviations of the Monte-Carlo simulation.

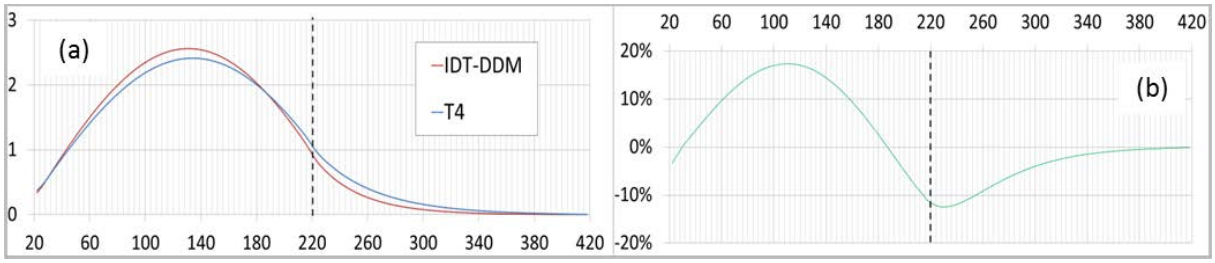

Figure 8. IDT-DDM vs. TRIPOLI-4®, on the 3D rodded color-set problem. (a) axial power distribution, (b) Relative pin Power per-cent Error (RPE)

A bigger discrepancy between the deterministic and the Monte-Carlo simulation is observed by analyzing the axial power distribution. Figure 8 (a) and (b) show, respectively, the axial power distribution for both TRIPOLI-4® and IDT-DDM and the Relative pin Power per-cent Error (RPE). The RPE is defined as:

$$\varepsilon_{p,k} = \varepsilon_k * \frac{P_k}{P_{avg}} \qquad (5)$$

Here ε_k is the relative per cent error between TRIPOLI-4® and IDT-DDM while P_k and P_{avg} are respectively the IDT-DDM average power in the layer "k" and the global average power. The RPE gives an estimation of the total amount of error for the pin power distribution calculation. Such error has a maximum of 17% in the active part and a minimum of -12% in the rodded part. Analyses are in progress to determine the origin of such discrepancies. For the deterministic simulation, the 2D self-shielding model used for the cross section library generation is not sufficient to handle the 3D axial effect at the interface between the B4C rodded assembly and the fresh UOx-Gd assembly. On the other hand, we are presently analyzing the statistics of the Monte Carlo simulation, in particular the convergence of the sources involved in

the power iterations. In this particular test, the rodded and the active parts are quite decoupled. The analysis at the pin-cells scale of the axial statistical fluctuations of the source shows that the convergence of the sources distribution has to be improved. An analysis is in progress to evaluate the optimum number of discarded batches per simulation.

7 CONCLUSIONS AND PERSPECTIVES

This work has been devoted to the synergy between the DDM and the CMFD acceleration by implementing a hybrid parallelism. The numerical tests point out the strength and the weakness of the method. Certainly, the DDM+CMFD method can be tuned for finding the optimum use of both CPU and cache. A shared memory parallelism shows a very good scalability: high fidelity calculation can be obtained in a reasonable amount of time with standard workstation using only the OpenMP directives, as for the 2D whole core calculation. However, such calculation also points out the weakness of the shared memory parallelism: lack of memory for a full depletion of the core. We mitigate this drawback by introducing an MPI layer in order to get the possibility to run calculation on computer clusters. Using the hybrid parallelism, full core 2D calculation can be run in roughly ¼ of hour using only 15 nodes, guaranteeing high-fidelity calculations in a reasonable amount of time and resources. The optimization of the MPI communications together with the parallelization of the CMFD solver still have to be performed to present a conclusive scalability study. The extent of memory provided by the distributed memory parallelism allows for a full depletion of the core by depleting up to 400,000 different materials.

We have also explored a challenging 3D color-set with a control-rod-in configuration, where we have compared the IDT-DDM solution with the Monte-Carlo TRIPOLI4® results. This realistic test case has revealed as a computationally demanding problem for both deterministic and Monte-Carlo codes. The quality of the solutions has to be improved in both sides. The deterministic code needs a more consistent 3D self-shielding model for the cross section generation at the axial interface between heterogeneous materials. Moreover, we are investigating an increasing discretization of the interface angular flux and quadrature formulas. On the other hand, the Monte-Carlo solution needs to be improved in order to face the source convergence challenge in a 3D geometry that represents hundreds of neutron mean-free-paths in the axial direction. A partial mitigation of this issue can be obtained by increasing both the number of particles and the number of discarded batches. Concerning the application of the DDM to 3D problems, we have not yet explored the tuning of the method in terms of the optimal axial and radial discretization of the subdomain mesh. This study will be part of our future works.

The DDM+CMFD method opens the door to many numerical research fields such as approximated 3D transport solutions and the 2D/1D fusion method. This method allows for the combination of locally defined discretization: the memory load for the discretization of each subdomain is locally managed by the processor. Regarding our future works about hybrid parallelism, we would like to explore different strategies for the load distribution over the MPI nodes, with a particular attention to asynchronous algorithms.

8 ACKNOWLEDGMENTS

This work is part of the development program for the neutronic code APOLLO3®. We gratefully acknowledge AREVA and EDF for their long term partnership and their support to the R&D. APOLLO3® and TRIPOLI-4® are registered trademark of CEA

9 REFERENCES

[1] J. M. ARAGONES and C. AHNERT, "A Linear Discontinuous Finite Difference Formulation for Synthetic Coarse-Mesh Few-Group Diffusion Calculations," *Nucl. Sci. Eng.*, vol. 94, pp. 309-324, 1986.

[2] J. J. HERRERO, C. AHNERT and J. M. ARAGONES, "3D Whole Core Fine Mesh Multigroup Diffusion Calculation by Domain Decomposition through Alternate Dissections," in *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)*, 2007.

[3] H. SCHWARZ, "Uber einige Abbildungsaufgaben," *Ges. Math. Abh*, vol. 11, p. 65–83, 1969.

[4] P. L. LIONS, «On the Schwarz alternating method III: a variant for nonoverlapping subdomains,» chez *In Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1990.

[5] R. LENAIN, E. MASIELLO, F. DAMIAN and R. SANCHEZ, "Coarse-grained parallelism for full core transport calculations," in *Physor2014*, Kyoto,Japan, 2014.

[6] I. ZMIJAREVIC, "Multidimensional Discrete Ordinates Nodal and Characteristics Methods for APOLLO2 Code," in *Proc. Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid,Spain, 1999.

[7] R. SANCHEZ, I. ZMIJAREVIC, M. COSTE-DELCLAUX and et al., "APOLLO2 Year 2010," *Nuclear Engineer and Technology*, vol. 42, no. 5, 2010.

[8] E. MASIELLO and I. ZMIJAREVIC, "New Numerical Solution with the Method of Short Characteristics for 2-D Heterogeneous Cartesian Cells in the APOLLO2 Code: Numerical Analysis and Tests," *Nucl. Sci. Eng.*, vol. 161, p. 257, 2009.

[9] N. HFAIEDH and A. SANTAMARINA, "Determination of the optimized SHEM mesh for neutron transport calculation," in *M&C2005 Topical Meeting*, Avignon, France, 2005.

[10] E. MASIELLO, B. MARTIN and J.-M. DO, "Domain Decomposition and CMFD Acceleration Applied to Discrete-ordinates methods for the solution of the Neutron Transport Equation in XYZ Geometry," in *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil, 2011.

[11] I. ZMIJAREVIC, H. GOLFIER, S. SANTANDREA et Z. STANKOVSKI, «Use of 2D whole core reactor models for PWR analysis with APOLLO2,» chez *Proc. Int. Conf. on Physics of Reactors, "Nuclear Power: A Sustainable Resource," (PHYSOR 2008)*, Interlaken, Switzerland, 2008.

[12] T-4. P. Team, «TRIPOLI-4®, CEA, EDF and AREVA Reference Monte Carlo Code,» chez *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013)*, Paris, 2013.