3D heterogeneous Cartesian cells for transport-based core simulations

Emiliano Masiello, Roland Lenain, Wesley Ford

Version of Record: https://www.sciencedirect.com/science/article/pii/S0306454920300621
Manuscript_d1906bf668feaa3a5a0392dd8aef4f0b

3D Heterogeneous Cartesian Cells For Transport-Based Core Simulations: Hybrid Coarse Parallelism and Tests

Emiliano Masiello[1], **Roland Lenain**[1] **and Wesley Ford**[1]
[1]DEN-Service d'études des réacteurs et de mathématiques appliquées (SERMA),
CEA, Université Paris-Saclay, F-91191, Gif-sur-Yvette, France
emiliano.masiello@cea.fr, roland.lenain@cea.fr, wesley.ford@cea.fr

ABSTRACT

In this work, we present an alternative discrete-ordinates method to perform 3D PWR core simulations. The numerical technique takes profit of the Cartesian modular construction of the geometry based on Heterogeneous Cartesian Cells (HCC). HCCs are basic geometrical patterns delimited by a box and having an arbitrary number of locally-extruded heterogeneous regions. The source is spatially expanded by piece-wise linear approximation in each region. Furthermore, the faces of the box, composing the boundary of the HCC, are discretized with a uniform Cartesian mesh. This surface sub-mesh is the support of a piece-wise linear representation of the interface angular flux. The linear expansion of the sources allows for a considerable reduction of the number of regions. Because of the Cartesian nature of the geometry, the method uses the effective spatial sweeping based on progression by front. Results on three-dimensional core simulations show accurate power distribution while minimizing the number of degrees of freedom. As a preliminary test for the accuracy of the method, results on the C5G7 MOX benchmark and the problem #4 of the VERA benchmark.

In this paper, we also summarize the latest application of the domain decomposition method (DDM) on the Integro-Differential Transport solver of APOLLO3®, namely IDT. In particular, we will focus on the strong scalability test by running the solver up to O(1000) cores of the HPC Cobalt cluster of the "Très Grand Centre de Calcul (TGCC)" in the "Centre de Calcul Recherche et Technologie" (CCRT) of the CEA. Initial results show good scalability performances: the full-core simulation of the EOLE nuclear reactor facility can be performed with a reference P3 281-group cross section library in 45min.

KEYWORDS: Neutron Transport Equation; Discrete-ordinates; Linear Short Characteristics; Domain Decomposition Method; APOLLO3®

1. Introduction

In this work, the Linear Short Characteristics (LSC), [1], [2] and [3], are extended to 3D Heterogeneous Cartesian Cells (HCC), [8]. The HCC allows for modeling fuel pin-cells in their exact shape without spatial homogenization. The discretized linear equations are obtained via a Galerkin projection of the integral transport equation. This leads to a response-matrix formalism similar to that used in the Interface-Current Collision Probability method, but in this case, because of the

discrete-ordinates approximation, the system solves for the angular flux using angular dependent matrices. Furthermore, the method allows for the solution of an arbitrary anisotropy order of the scattering kernel. The presented method preserves the typical fast solution of the LSC, allowing for a HCC-by-HCC sweeping algorithm.

We take advantage of the regularity of the geometry to model the pin-cell in a modular basic pattern. The HCC has an external box boundary and can contain an arbitrary number of concentric, heterogeneous cylinders. The cylinders can be not centered in the box and can intersect the external surface of the surrounding box. This is done to model the regular patterns as fuel pins and spacer grid and irregular patterns as the reflector boundary of the reactor. The faces of the surrounding box are subdivided in an arbitrary number of equally dimensioned surface meshes, this minimizes the numerical diffusion and provides a accurate propagation of the angular flux. The source is expanded on linear/bilinear bases on each volume of the HCC, while the boundary fluxes entering and exiting the cell are similarly approximated along surfaces composing the boundary of the HCC. The mathematical formalism, which is based on the projection of the integral transport equation, allows for an explicit linear system solving the spatial moments of the angular flux simultaneously for all regions and surfaces of the HCC.

The discretized transport equation consists of two linear systems per HCC formally similar to those used for homogeneous cells.

The 3D HCC implementation is an upgrade of the XYZ solver, namely IDT (Integro Differential Transport) [4], of in the code APOLLO3®. IDT has been already benchmarked with realistic 3D full-core simulations, the results of which have been published in reference [13]. In this work, we are more concerned about the mathematical framework of the HCC coefficients.

2. Linear Short Characteristics and Heterogeneous Cartesian Cells (HCC)

The IDT solver is a discrete-ordinates neutral-particle transport code based on XYZ geometry. In the past years, IDT has been extended to Heterogeneous Cartesian Cells (HCC) to model fuel pins of nuclear reactor in their exact geometries without need of spatial homogenization, [6]. The HCC model has proven accurate results while saving the number of spatial meshes and, thus, computational time and memory. Furthermore, because the 3D extension is based on locally extruded geometry, it allows for the modeling of non extruded 3D geometries, as the assembly top/bottom nozzles, plugs and grids.

Because of the discrete ordinates, the angular integrals are approximated by quadrature formula composed of N_d directions,

$$\frac{1}{4\pi}\int_{4\pi} d\mathbf{\Omega}\; f(\mathbf{\Omega}) \simeq \sum_{d=1,..,N_d} f(\mathbf{\Omega}_d) w_d, \tag{1}$$

where w_d is the angular weight, while $\mathbf{\Omega}_d$ is the direction unit vector. The distribution of particle flux in direction d is computed using source iterations, which consists in solving for the uncollided flux by updating the self scattering source at each iteration. For a fixed energy group, the iterative

scheme for the angular flux $\psi_d^{(l+1)}(\mathbf{r}) = \psi^{(l+1)}(\mathbf{r}, \mathbf{\Omega}_d)$ solves the equation

$$
\begin{aligned}
[\mathbf{\Omega}_d \cdot \nabla + \Sigma(\mathbf{r})]\psi_d^{(l+1)}(\mathbf{r}) &= q_d^{(l)}(\mathbf{r}) \text{ for } (\mathbf{r}, \mathbf{\Omega}_d) \in \mathbb{R}^3 \times S_N^2, &(2)\\
\psi_d^{-,(l+1)}(\mathbf{r}) &= (\beta\psi^{+,(l)})(\mathbf{r}, \mathbf{\Omega}_d) \text{ for } (\mathbf{r}, \mathbf{\Omega}_d) \in \mathbb{R}_-^2 \times S_N^2, &(3)
\end{aligned}
$$

where $\psi^{(l+1)}(\mathbf{r}, \mathbf{\Omega})$ and $q^{(l)}(\mathbf{r}, \mathbf{\Omega})$ are the angular flux at iteration $(l+1)$ and the source computed at previous iteration (l), respectively. When homogeneous boundary conditions are present, the current incoming boundary flux $\psi^{-,(l+1)}(\mathbf{r}, \mathbf{\Omega})$ is updated by using the outgoing boundary flux $\psi^{+,(l)}(\mathbf{r}, \mathbf{\Omega})$ at previous iteration. If an external boundary incident flux is present, then Eq. (3) is replaced by the identity $\psi^-(\mathbf{r}, \mathbf{\Omega}) = S_{in}(\mathbf{r}, \mathbf{\Omega})$**,** where $S_{in}(\mathbf{r}, \mathbf{\Omega})$ is the incident source.

The iteration ends with the updating of the angular moments and, thus, of the source,

$$
\begin{aligned}
\phi_{k,l}^{(l+1)}(\mathbf{r}) &\simeq \sum_d w_d A_{k,l}(\mathbf{\Omega}_d)\psi_d^{(l+1)}(\mathbf{r}), &(4)\\
q_d^{(l+1)}(\mathbf{r}) &= \sum_k (2k+1)\Sigma_k \sum_{|l|\leqslant k} A_{k,l}(\mathbf{\Omega}_d)\phi_{k,l}^{(l+1)}(\mathbf{r}) + q_{ext,d}(\mathbf{r}), &(5)
\end{aligned}
$$

where the angular moment $\phi_{k,l}^{(l+1)}(\mathbf{r})$ in Eq. (4) is computed thanks to the discrete ordinates approximation (1). As usual, in Eq. (4), the $A_{k,l}$ are the real-valued spherical harmonics of order k and degree l while Σ_k is the scattering cross section of order k.

The one-group transport operator

$$\mathcal{L}_d \equiv [\mathbf{\Omega}_d \cdot \nabla + \Sigma(\mathbf{r})]$$

is inverted within each HCC using linear short characteristics . Without any geometrical restriction, the method can be presented in an unstructured mesh context encapsulated in a Cartesian grid. In the case of HCC application, the Cartesian cells are the surrounding boxes of the HCC. The HCC could contains an arbitrary number of concentric rings representing the fuel pin, air gap, clad and the moderator. The integral form of transport equation is:

$$\psi_d(\mathbf{r}_t^- + x\mathbf{\Omega}_d) = \psi_d(\mathbf{r}_t^-)e^{-\tau_t(x,0)} + \int_0^x q_d(\mathbf{r}^- + y\mathbf{\Omega}_d)e^{-\tau_t(x,y)}dy \quad (6)$$

with the optical thickness $\tau_t(x, y) = \int_y^x dz\ \Sigma(\mathbf{r}_t^- + z\mathbf{\Omega}_d)$. If we restrict our interest to a single HCC we can specialize Eq. (6). Let us use the following indexes:

$\alpha, \beta, \gamma, ...$: indexes of the regions within the HCC,

s, s' : local boundary surface mesh indexes of the HCC,

$S^{\pm}(\mathbf{\Omega}_d)$: set of the outgoing/incoming local surface indexes with respect to the direction $\mathbf{\Omega}_d$,

t : trajectory index,

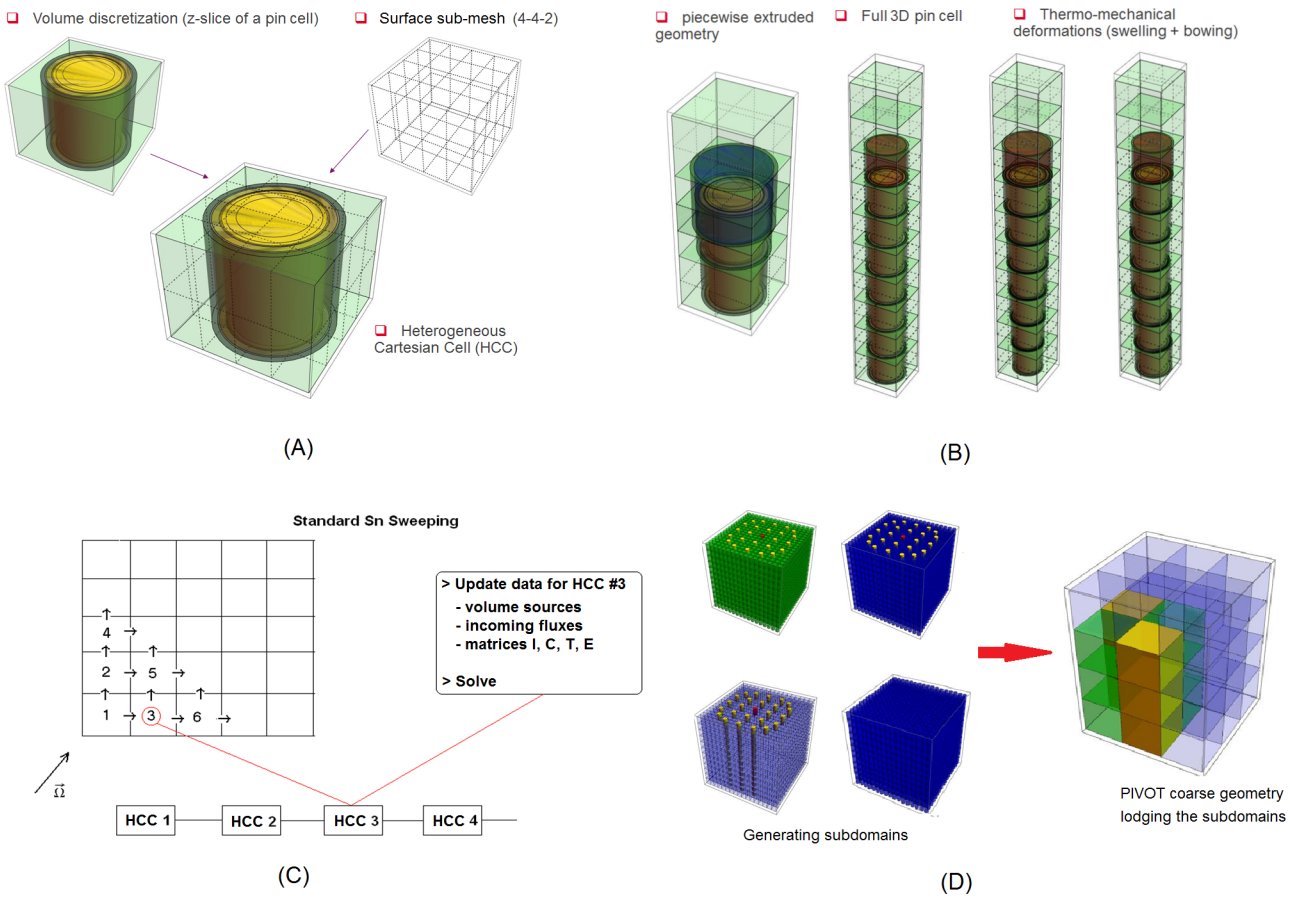

Figure 1: (A) Spatial discretization of an HCC with volume and surface mesh for a pin cell Z-slice. (B) Examples of models for deformed pin cells. (C) XY illustration of the HCC front-based spatial sweeping. (D) Subdomain geometry of the C5G7 MOX benchmark.

$T(\mathbf{\Omega}_d)$: set of indexes of all the trajectories associated to a given direction $\mathbf{\Omega}_d$ for a given HCC geometrical pattern,

$T_\alpha(\mathbf{\Omega}_d)$: set of indexes of trajectories intersecting the region α of the HCC,

$T_s(\mathbf{\Omega})$: set of indexes of trajectories intersecting the boundary surface s of the HCC,

i, j, k : indexes of the chords of a trajectory,

$I_t(\mathbf{\Omega}_d)$: set of indexes of chord lengths of trajectory t,

$I_{t,\alpha}(\mathbf{\Omega}_d)$: set of indexes of chord lengths of trajectory t that intersects region α, $I_{t,\alpha} \in I_t$,

$r(i)$: integer map such that $r : i \to \alpha$, giving for a chord i the region index α,

$\mathbf{r}_t^-$ and $\mathbf{r}_t^+$: incoming and outgoing points, respectively, of trajectory t.

We define the total cross section of the chord i as $\Sigma_i = \Sigma_{r(i)}$ and the optical paths $\tau_{t,i}(x,0)$ and $\tau_{t,i}(x,x_j)$, for x being the coordinate along the trajectory t associated to a point in the chord i, as

$$\tau_{t,i}(x,0) = \begin{cases} \Sigma_i x & \text{for } i = 1, \\ \sum_{j=1}^{i-1} \tau_{t,j} + \Sigma_i (x - x_{t,i-1}) & \text{for } i > 1, \end{cases}$$

$$\tau_{t,i}(x,x_j) = \begin{cases} 0 & \text{for } i = 1 \text{ or } j = i-1, \\ \sum_{k=j+1}^{i-1} \tau_{t,k} + \Sigma_i (x - x_{t,i-1}) & \text{for } i > 1, \end{cases}$$

with $x_0 = 0$, where the optical thickness of the chord is defined as $\tau_{t,i} = \Sigma_i (x_{t,i} - x_{t,i-1})$.

Applying the HCC indexes to Eq. (6), one obtains

$$\begin{aligned} \psi_d(\mathbf{r}_t^- + x\mathbf{\Omega}_d) = {} & \psi_d(\mathbf{r}_t^-) e^{-\tau_{t,i}(x,0)} + \\ & \sum_{j<i} e^{-\tau_{t,i}(x,x_j)} \int_{x_{j-1}}^{x_j} q_{d,r(j)}(\mathbf{r}^- + y\mathbf{\Omega}_d) e^{-\Sigma_j (x_j - y)} dy + \\ & \int_{x_{j-1}}^{x} q_{d,r(i)}(\mathbf{r}_t^- + y\mathbf{\Omega}_d) e^{-\Sigma_j (x-y)} dy \end{aligned} \tag{7}$$

Few comments are necessary for this formula. The second term on the RHS of the Eq. (7) takes into account the contribution from chord j to chord i. Because of the nested geometry of the HCC, the trajectory can exit and re-enter in the same ring. Thus, the trajectory map $r(j)$ may assume the

same region index as $r(i)$ for a specific chord i. This characteristic reflects the fact that Eq. (7) exactly and explicitly inverts the transport operator inside the HCC. Note also that, because of the integral transport equation, the interface angular flux along the inner curved surfaces of the HCC is implicitly solved without approximation. We introduce the spatial approximation of the source as

$$q_{d,\alpha}(\mathbf{r}) = \sum_{\alpha}\sum_{c} P_{\alpha,c}(\mathbf{r}) q_{d,\alpha,c} = \sum_{\alpha} \mathbf{P}_{\alpha}(\mathbf{r}) \cdot \mathbf{q}_{d,\alpha} \tag{8}$$

where c is indexing 4 spatial components, i.e. one constant and three linear moments, while $\mathbf{P}_{\alpha}(\mathbf{r})$ is the linear base

$$\mathbf{P}_{\alpha}(\mathbf{r}) = \begin{bmatrix} 1 \\ \mathbf{r} - \mathbf{r}_{\alpha} \end{bmatrix}$$

with $\mathbf{r}_{\alpha}$ being the center of mass of α. The spatial coordinate $\mathbf{r}$ is defined with respect to the center of mass of the HCC box. Because the linear moments of $\mathbf{P}_{\alpha}(\mathbf{r})$ are orthogonal to the constant moment but not necessary orthogonal to each others, the mass matrix $\underline{\mathbf{M}_{\alpha}} = (\mathbf{P}_{\alpha}, \mathbf{P}_{\alpha})$ is generally of the form

$$\underline{\mathbf{M}_{\alpha}} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & (\mathbf{r} - \mathbf{r}_{\alpha}, \mathbf{r} - \mathbf{r}_{\alpha}) \end{bmatrix}$$

and the spatial moments of the source are then defined by the relation

$$\mathbf{q}_{d,\alpha} = \frac{[\underline{\mathbf{M}_{\alpha}}]^{-1}}{V_{\alpha}} \int_{D_{\alpha}} d\mathbf{r}\ \mathbf{P}_{\alpha}(\mathbf{r}) q_d(\mathbf{r}) = [\underline{\mathbf{M}_{\alpha}}]^{-1} (\mathbf{P}_{\alpha}, q)\,. \tag{9}$$

The particular approximation characterizing short characteristics applied to HCC is the linear approximation of the boundary angular flux. As specified before, here we intend the boundary as the external surfaces of a single HCC. As shown in Fig. 1-A, each face of the surrounding box has a conformal Cartesian mesh. The angular flux is then expanded locally on each surface mesh. By noting with s the surface index, the flux is represented as the linear combination

$$\psi^{\pm}_{d,t} = \psi_d(\mathbf{r}^{\pm}_t) = \sum_{s \in S^{\pm}(\mathbf{\Omega}_d)} \sum_{b} P_{s,b}(\mathbf{r}^{\pm}_t) \psi^{\pm}_{d,s,b} = \sum_{s \in S^{\pm}(\mathbf{\Omega})} \mathbf{P}_s(\mathbf{r}^{\pm}_t) \cdot \psi^{\pm}_{d,s} \tag{10}$$

where $\mathbf{P}_s(\mathbf{r}^{\pm}_t)$ is the linear base associated to the surface s, while b is the index of the three spatial components, one constant plus two linear moments,

$$\mathbf{P}_s(\mathbf{r}^{\pm}_t) = \begin{bmatrix} 1 \\ \mathbf{r}^{\pm}_t - \mathbf{r}_s \end{bmatrix},$$

where $\mathbf{r}_s$ is the center of mass of the surface s with respect to the local system of coordinates. The mass matrix of the surface s, defined as $\mathbf{M}_s = \langle \mathbf{P}_s, \mathbf{P}_s \rangle^{\pm}$, is diagonal because of the 2D Cartesian shape of the surface. The spatial moments of the angular flux, $\psi^{\pm}_{d,s}$, are defined by the projection

$$\underline{\mathbf{M}_s} \psi^{\pm}_{d,s} = \langle \mathbf{P}_s, \psi_d \rangle^{\pm} = \frac{|\mathbf{\Omega}_d \cdot \mathbf{n}_s|}{A_s} \int_{A_s} d\mathbf{r}^{\pm} \mathbf{P}_s(\mathbf{r}^{\pm}) \psi_d(\mathbf{r}^{\pm}), \tag{11}$$

where A_s is the surface area. Using the spatial expansion (8) for the source, in a point $\mathbf{r}'$ along the trajectory, and the expansion (10) for the incoming angular flux, in a point $\mathbf{r}^{-}_t$ within a surface s',

the integral transport equation (7) in a point x of a chord i of a trajectory t becomes

$$\begin{aligned}\psi_{d,t,i}(\mathbf{r}_t^- + x\mathbf{\Omega}_d) &= e^{-\tau_{t,i}(x,0)}\mathbf{P}_{s'}(\mathbf{r}_t^-)\cdot\psi_{d,s'}^- + \sum_{j<i} e^{-\tau_{t,i}(x,x_j)}\mathbf{F}_{d,t,r(j),j}(x_j)\cdot\mathbf{q}_{d,r(j)} + \\ &\quad \mathbf{F}_{d,t,r(i),i}(x)\cdot\mathbf{q}_{d,r(i)}\end{aligned} \tag{12}$$

where the 4-component exponential integral vector $\mathbf{F}_{d,t,r(i),i}(x)$ is defined, for $r(i)=\alpha$, as

$$\mathbf{F}_{d,t,\alpha,i}(x) = \int_{x_{t,i-1}}^{x} \mathbf{P}_\alpha(\mathbf{r}_t^- + y\mathbf{\Omega}_d)e^{-\Sigma_\alpha(x-y)}dy \text{ and} \tag{13}$$

$$\mathbf{F}_{d,t,\alpha,i} = \mathbf{F}_{d,t,\alpha,i}(x_{t,i}) = \int_{x_{i-1}}^{x_i} \mathbf{P}_\alpha(\mathbf{r}_t^- + y\mathbf{\Omega}_d)e^{-\Sigma_\alpha(x-y)}dy. \tag{14}$$

Equation (12) can be specialized for an outgoing boundary point $\mathbf{r}_t^+ = \mathbf{r}_t^- + X_t\mathbf{\Omega}_d$, where X_t is the total chord length of the trajectory t

$$\psi_{d,t,i}(\mathbf{r}_t^+) = e^{-\tau_t}\mathbf{P}_s(\mathbf{r}_t^-)\cdot\psi_s^- + \sum_{i\in I_t(\mathbf{\Omega}_d)} e^{-\tau_{t,i}^>}\mathbf{F}_{d,t,r(i),i}\cdot\mathbf{q}_{r(i)}, \tag{15}$$

here τ_t is the total optical thickness of the trajectory defined as $\tau_t = \sum_{i\in I_t}\tau_{t,i}$. It can also be decomposed as

$$\tau_t = \tau_{t,i}^< + \tau_{t,i} + \tau_{t,i}^>$$

for each chord i. The previous relation define the up-stream optical path $\tau_{t,i}^< = \sum_{j<i}\tau_{t,j}$, that is non-zero except for $i=1$, and the down-stream optical path $\tau_{t,i}^> = \sum_{j>i}\tau_{t,j}$, that is non-zero except for $i = N(I_t)$ where $N(I_t)$ is the number of total chord lengths associated to t.

The projection of the integral equation does not need the explicit expansion of the angular flux within the volumes. However, the updating of the source needs the computation of the angular moments which are approximated by discrete-ordinates integration (4). The source expansion (8) "hides" the spatial expansion of the angular moments. The computation of the spatial moments of $\phi_{k,l}(\mathbf{r})$ are obtained by applying the definition (9) to the angular moments as

$$\phi_{\alpha,k,l} = \frac{[\underline{\mathbf{M}_\alpha}]^{-1}}{V_\alpha}\int_{4\pi} d\mathbf{\Omega} A_{k,l}(\mathbf{\Omega})\int_{D_\alpha} d\mathbf{r}\ \mathbf{P}_\alpha(\mathbf{r})\psi(\mathbf{r},\mathbf{\Omega}). \tag{16}$$

Because the angular integration is approximated by discrete-ordinates quadrature formula and the spatial integration is angular dependent (since is approximated by discrete trajectories), the spatial moments of $\phi_{k,l}^{(i+1)}(\mathbf{r})$ are computed as the sum

$$\phi_{\alpha,k,l} \simeq [\underline{\mathbf{M}_\alpha}]^{-1}\sum_d w_d A_{k,l}(\mathbf{\Omega}_d)\left(\mathbf{P}_\alpha,\psi\right)_d, \tag{17}$$

that is used for updating the source as shown in (4). The numerical integration (17) helps to define the spatial moments of the angular flux as

$$\psi_{d,\alpha} = [\underline{\mathbf{M}_\alpha}]^{-1}\left(\mathbf{P}_\alpha,\psi\right)_d. \tag{18}$$

In the previous definition, the mass matrix $\underline{\mathbf{M}_\alpha}$ is not angular dependent and is computed once and for all by analytical integration. Instead, the scalar product $(\mathbf{P}_\alpha,\psi)_d$ of Eq. (17) defines the numerical projection of the angular flux as

$$(\mathbf{P}_\alpha,\psi)_d = \frac{1}{V_{\alpha,d}} \sum_{t\in T(\mathbf{\Omega}_d)} w_t^\perp \sum_{i\in I_{t,\alpha}(\mathbf{\Omega}_d)} \Delta_i \, (\mathbf{P}_\alpha,\psi)_{d,t,i} \,, \tag{19}$$

where $w_t^\perp$ is the transverse weight of the trajectory, $V_{\alpha,d}$ is the numerical volume of region α for direction d, while the integral $(\cdot,\cdot)_{d,t,i}$ defines the projection on the chord length, as

$$(\mathbf{P}_\alpha,\psi)_{d,t,i} = \frac{1}{\Delta_i} \int_{x_{i-1}}^{x_i} dx \; \mathbf{P}_\alpha(\mathbf{r}_t^- + x\mathbf{\Omega}_d)\psi(\mathbf{r}_t^- + x\mathbf{\Omega}_d), \tag{20}$$

where $\Delta_i = (x_i - x_{i-1})$ is the length of the chord. Equation (20) includes for the contribution of each trajectory to the spatial moments of the angular flux. Substituting the integral equation (12) in (20), one obtains the explicit solution for the moments $(\mathbf{P}_\alpha,\psi)_{d,t,i}, viz.$,

$$\begin{aligned} (\mathbf{P}_\alpha,\psi)_{d,t,i} \;\; = \;\; & [\mathbf{G}_{d,t,\alpha,i} \times \mathbf{P}_s(\mathbf{r}_t^-)]e^{-\tau_{t,i}^<}\psi_s^- + \\ & \mathbf{G}_{d,t,\alpha,i} \times \sum_{j=1}^{i-1} e^{-\tau_{i,j}} \mathbf{F}_{d,t,r(j),j}\mathbf{q}_{r(j)} + \underline{\mathbf{H}}_{d,t,\alpha,i}\mathbf{q}_\alpha. \end{aligned} \tag{21}$$

The 4-components exponential integral vectors $\mathbf{G}_{d,t,\alpha,i}$ and $\mathbf{F}_{d,t,\alpha,i}$ appearing in (21) are

$$\mathbf{G}_{d,t,\alpha,i} \;\; = \;\; \int_{x_{i-1}}^{x_i} \mathbf{P}_\alpha(\mathbf{r}_t^- + x\mathbf{\Omega}_d)e^{-\Sigma_\alpha(x-x_{i-1})}dx, \tag{22}$$

$$\mathbf{F}_{d,t,\alpha,i} \;\; = \;\; \mathbf{F}_{d,t,\alpha,i}(x_i), \tag{23}$$

while the integral exponential matrix $\underline{\mathbf{H}}_{d,t,\alpha,i}$ is

$$\underline{\mathbf{H}}_{d,t,\alpha,i} = \int_{x_{i-1}}^{x_i} \mathbf{P}_\alpha(\mathbf{r}_t^- + x\mathbf{\Omega}_d) \times \mathbf{F}_{d,t,\alpha,i}(x) \; dx. \tag{24}$$

When introducing the Eq. (21) into (19) and (19) into (18), one obtains the linear system explicitly solving the angular flux spatial moments of the region α ,

$$\psi_{d,\alpha} = \sum_{s\in S^-(\mathbf{\Omega})} \underline{\mathbf{I}}_{d,\alpha,s}\psi_{d,s}^- + \sum_{\beta\neq\alpha} \underline{\mathbf{C}}_{d,\alpha,\beta}\mathbf{q}_{d,\beta} + \underline{\mathbf{C}}_{d,\alpha,\alpha}\mathbf{q}_{d,\alpha} \tag{25}$$

where

$$\underline{\mathbf{I}}_{d,\alpha,s} = \frac{[\underline{\mathbf{M}_\alpha}]^{-1}}{V_{\alpha,d}} \sum_{t\in T_\alpha(\mathbf{\Omega}_d)} w_t^\perp \sum_{i\in I_{t,\alpha}(\mathbf{\Omega}_d)} [\mathbf{G}_{d,t,\alpha,i} \times \mathbf{P}_s(\mathbf{r}_t^-)]e^{-\tau_{t,i}^<}, \tag{26}$$

is the incoming matrix, while

$$\underline{\mathbf{C}}_{d,\alpha,\beta} = \frac{[\underline{\mathbf{M}_\alpha}]^{-1}}{V_{\alpha,d}} \sum_{t\in T_\alpha(\mathbf{\Omega}_d)} w_t^\perp \sum_{i\in I_{t,\alpha}(\mathbf{\Omega}_d)} \sum_{\substack{j<i \\ j\in I_{t,\beta}(\mathbf{\Omega}_d)}} [\mathbf{G}_{d,t,\alpha,i} \times \mathbf{F}_{d,t,\beta,j}]e^{-\tau_{i,j}}, \tag{27}$$

and

$$\underline{\mathbf{C}}_{d,\alpha,\alpha} = \frac{[\underline{\mathbf{M}}_\alpha]^{-1}}{V_{\alpha,d}} \sum_{t\in\ T_\alpha(\mathbf{\Omega}_d)} w_t^\perp \sum_{i\in I_{t,\alpha}(\mathbf{\Omega}_d)} \big(\underline{\mathbf{H}}_{d,t,\alpha,i} + \sum_{\substack{j<i \\ j\in I_{t,\alpha}(\mathbf{\Omega}_d)}} [\mathbf{G}_{d,t,\alpha,i} \times \mathbf{F}_{d,t,\alpha,j}] e^{-\tau_{i,j}}\big), \tag{28}$$

are the collision matrices.

Equation (25) needs a transmission equation to compute the outgoing angular flux transmitted by continuity to the next HCC cell. We use definition (11) to construct the equation for the spatial moments of the boundary surface flux. The outgoing angular flux is explicitly computed with the integral transport equation (15). Because of the numerical spatial integration, the integral in (11) is computed as

$$\psi_{d,s}^+ = \frac{[\underline{\mathbf{M}}_s]^{-1}}{A_{s,d}} \sum_{t\in\ T_\alpha(\mathbf{\Omega}_d)} w_t^\perp \mathbf{P}_s(\mathbf{r}_t^+)\psi_d(\mathbf{r}_t^+) \tag{29}$$

since the trajectory transverse weight is equal to

$$w_t^\perp = |\mathbf{\Omega}_d{\cdot}\mathbf{n}_s|\,\Delta A_{s,t} = |\mathbf{\Omega}_d{\cdot}\mathbf{n}_{s'}|\,\Delta A_{s',t}$$

where s and s' are respectively the outgoing surface index and the incoming surface index intersected by trajectory t and defined by the segment $(\mathbf{r}_t^-, \mathbf{r}_t^+)$. Using the equation for the outgoing flux, i.e. Eq. (15), in the definition (29), one obtains the transmission equation

$$\psi_{d,s}^+ = \sum_{s'\in S^-(\mathbf{\Omega}_d)} \underline{\mathbf{T}}_{d,s,s'}\psi_{d,s'}^- + \sum_\alpha \underline{\mathbf{E}}_{d,s,\alpha}\mathbf{q}_{d,\alpha}\ , \tag{30}$$

that gives the spatial moments of the outgoing angular flux $\psi_{d,s}^+$ as a function of the sources and of the incoming fluxes $\psi_{d,s'}^-$. The matrices $\underline{\mathbf{T}}_{s,s'}$ and $\underline{\mathbf{E}}_{s,\alpha}$ are, respectively, the transmission matrix

$$\underline{\mathbf{T}}_{d,s,s'} = \frac{[\underline{\mathbf{M}}_s]^{-1}}{A_{s,d}} \sum_{t\in\ T_\alpha(\mathbf{\Omega}_d)} w_t^\perp [\mathbf{P}_s(\mathbf{r}_t^+) \times \mathbf{P}_{s'}(\mathbf{r}_t^-)] e^{-\tau_t}, \tag{31}$$

and the escape matrix,

$$\underline{\mathbf{E}}_{d,s,\alpha} = \frac{[\underline{\mathbf{M}}_s]^{-1}}{A_{s,d}} \sum_{t\in T_s(\mathbf{\Omega}_d)} w_t^\perp\ [\mathbf{P}_s(\mathbf{r}_t^+) \times \sum_{i\in I_{t,\alpha}(\mathbf{\Omega}_d)} e^{-\tau_{t,i}^>}\mathbf{F}_{d,t,\alpha,i}]. \tag{32}$$

Coefficients (26), (31) and (32) are computed only for those surfaces and regions shadowing each other along the direction. Because of the re-entrant geometry of the rings, the collision coefficients (27) are generally non-zero for all couple of indexes (α, β).

3. Sweeping algorithm

Because the HCCs compose a conformal XYZ grid, the spatial sweeping algorithm is based on the standard sweeping-by-front of homogeneous XYZ grids. Each HCC transfers its outgoing fluxes to the down-stream HCCs by using the continuity condition of the angular flux on each mesh of the outgoing surface. Figure 1-C sketches the spatial solution algorithm. The HCC are ordered

by forming a front, as depicted in the figure. Then, the volume sources of regions covered by the HCC are loaded together with the incoming boundary fluxes. Those fluxes can come from boundary conditions or from up-stream HCCs. Because the HCC surface mesh is uniform on each face, the overall surface mesh is conforming in each direction so that the flux is transmitted without approximation. As illustrated in Fig. 1-B, HCC-based mesh can easily handle piecewise-extruded geometry for the modeling of the assembly deformation.

By noting with n the HCC order number during the sweeping along the direction d, the continuity of the angular spatial moments of the flux on an incoming surface s' of the HCC #n is ensured by

$$\psi^{-}_{n,d,s'} = \psi^{+}_{n',d,s} \tag{33}$$

where n' is the up-steam HCC order number, while s is the index of the outgoing surface of n' (which is the same global surface defined by s' of n). Equations (25) and (30)are then solved for all the regions and outgoing surfaces of the HCC.

$$\psi_{n,d,\alpha} = \sum_{s' \in S^{-}(\mathbf{\Omega})} \underline{\mathbf{I}}_{n,d,\alpha,s'} \psi^{-}_{n,d,s'} + \sum_{\beta \in R_n} \underline{\mathbf{C}}_{n,d,\alpha,\beta} \mathbf{q}_{d,\beta} + \underline{\mathbf{C}}_{n,d,\alpha,\alpha} \mathbf{q}_{d,\alpha} \text{ for } \alpha \in R_n \ ,$$

$$\psi^{+}_{n,d,s} = \sum_{s' \in S^{-}_{n}(\mathbf{\Omega})} \underline{\mathbf{T}}_{n,d,s,s'} \psi^{-}_{n,d,s'} + \sum_{\alpha \in R_n} \underline{\mathbf{E}}_{n,d,s,\alpha} \mathbf{q}_{d,\alpha} \text{ for } s \in S^{+}_{n}(\mathbf{\Omega}),$$

which we re-propose here by introducing the HCC order number n, are then solved for all the regions and outgoing surfaces of the HCC.

4. The trajectory tracking requirements

Numerical integration is performed by a cell-based modular tracking. The HCC are classified by their geometries, a unique HCC geometrical pattern defines a unique trajectory module. Thus, because pin-cell patterns are repeated high number of times, the memory imprint of the 3D HCC tracking is negligible. Numerical integrals are performed using a local coordinates system centered in the center of mass of the surrounding box.

Furthermore, because the tracking is local to each HCC mesh, the numerical integration takes care of geometrical discontinuities that are normally neglected by most popular MOC tracking. This last aspect, together with the use of Gauss points, makes the HCC matrix integration accurate and robust. The local 3D tracking is based on the factorization among two 2D ray-tracing.

The local 3D tracking is based on the factorization among two 2D ray-tracing. As depicted in Fig. (2), the trajectory tracing starts from the projection of 2D discontinuities. In this step, also the geometrical discontinuities induced by the surface mesh of the HCC are taken into account (the red dots in the figure). Then, a set of equally spaced trajectory are distributed along the 2D transverse plane, the trajectory spacing is defined by the user. At this stage, the trajectories (blue and black lines) defines a set of 2D slices. The 2D trajectories are then generated by distributing Gauss points along the transverse side of each slice. The Gauss quadrature is adapted to the slice transverse thickness following a user-defined number of points. Generally, 0.05 cm trajectory spacing in XY with three or five Gauss points and 0.07 in Z with 3 Gauss points are sufficient to ensure very accurate integration. The same algorithm is then repeated for each 3D slice defined by

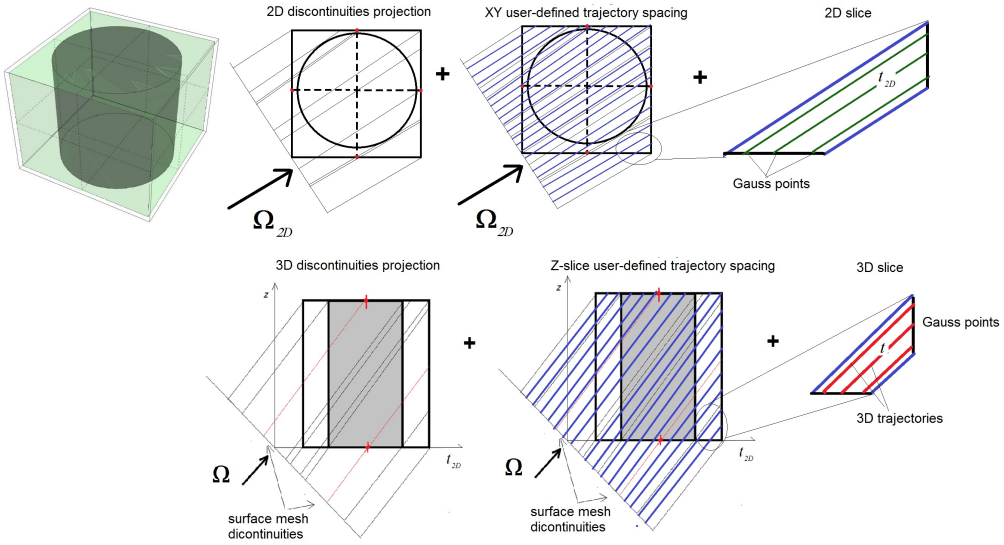

Figure 2: HCC modular tracking illustration.

the 2D trajectory and the Z axis. The same algorithm is then repeated for each 3D slice defined by the 2D trajectory and the Z axis. Then, the spatial weight is obtained by the product of the weight of the 2D trajectory in the XY plane and the weight of the "real" 3D trajectory in the $Z - t_{2D}$ plane, that is

$$w_t^{\perp} = w_{t_{2D}}^{\perp} w_z^{\perp}.$$

The storage requirement for each trajectory consists in: the transverse weight $w_t^{\perp}$, the outgoing and incoming surface points $\mathbf{r}_t^{\pm}$ with respect to the local coordinate system of the HCC, the local indexes of the surfaces, i.e. s and s', intersected by the trajectory, the total length X_t, the number of chords, the chord-to-region mapping $r(i)$, the chord lengths. These data are stored for each angle belonging to half of the unit sphere. Because of the reciprocity property of the integral transport equation, the computation of the matrices (26), (27), (28), (31) and (32) in the second half of the sphere is performed by using symmetry relations among matrices of opposite directions.

$$\begin{aligned}
\underline{\mathbf{M}_{\alpha}}\underline{\mathbf{C}}_{\alpha,\beta}(-\mathbf{\Omega}_d) &= \underline{\mathbf{S}_{vv}}(-\mathbf{\Omega}_d)[\underline{\mathbf{M}_{\beta}}\underline{\mathbf{C}}_{\beta,\alpha}(\mathbf{\Omega}_d)]^T, \\
\underline{\mathbf{M}_{s'}}\underline{\mathbf{T}}_{s',s}(-\mathbf{\Omega}_d) &= \underline{\mathbf{S}_{s's}}(-\mathbf{\Omega}_d)[\underline{\mathbf{M}_{s}}\underline{\mathbf{T}}_{s,s'}(\mathbf{\Omega}_d)]^T,
\end{aligned}$$

and

$$\underline{\mathbf{M}_{\alpha}}\underline{\mathbf{I}}_{\alpha,s'}(-\mathbf{\Omega}_d) = \underline{\mathbf{S}_{vs}}(-\mathbf{\Omega}_d)[\underline{\mathbf{M}_{s}}\underline{\mathbf{E}}_{s,\alpha}(\mathbf{\Omega}_d)]^T,$$

where the matrices $\underline{\mathbf{S}_{vv}}(-\mathbf{\Omega}_d)$, $\underline{\mathbf{S}_{vs}}(-\mathbf{\Omega}_d)$ and $\underline{\mathbf{S}_{s's}}(-\mathbf{\Omega}_d)$ are integer symmetric matrices giving

the sing assumed by the monomials $x^i y^j z^k$ of the base. In particular,

$$\underline{\mathbf{S}_{vv}}(-\mathbf{\Omega}_d) = sign \begin{bmatrix} 1 & x & y & z \\ x & x^2 & xy & xz \\ y & xy & y^2 & yz \\ z & xz & yz & z^2 \end{bmatrix}$$

is for the volumes,

$$\underline{\mathbf{S}_{s's}}(-\mathbf{\Omega}_d) = [\underline{\mathbf{S}_{ss'}}(-\mathbf{\Omega}_d)]^T = sign \begin{bmatrix} 1 & x_s & y_s \\ x_{s'} & x_{s'}x_s & x_{s'}y_s \\ y_{s'} & y_{s'}x_s & y_{s'}y_s \end{bmatrix}$$

is for the surfaces and

$$\underline{\mathbf{S}_{vs}}(-\mathbf{\Omega}_d) = [\underline{\mathbf{S}_{sv}}(-\mathbf{\Omega}_d)]^T = sign \begin{bmatrix} 1 & x_s & y_s \\ x & xx_s & xy_s \\ y & yx_s & yy_s \\ z & zx_s & zy_s \end{bmatrix}$$

for volume-to-surface or the surface-to-volume. A particular case, which is actually the default model for a pin-cell, is presented by symmetric HCCs, having the pin centered in center of mass of the box. In this case, because of the geometrical symmetry, the numerical integration for matrix coefficients, i.e. Eqs. (26), (27), (28), (31) and (32), is applied only on the first quadrant, while the rest is computed by symmetry relations.

4.1. Outer power iterations and thermal iterations

The k-effective eigenvalue problem can be formally presented as

$$\begin{cases} (L-H)\psi(x) = \frac{1}{k} F\psi(x) & \text{for } x \in X \\ \psi^-(x) = 0 & \text{for } x \in \partial X^-, \end{cases} \tag{34}$$

where X is the phase space with its entering $(+)$ and exiting $(-)$ boundaries $\partial X^{\pm}$, while L, H and F are respectively the transport, the scattering and the fission operators. Without loss in generality, only the vacuum boundary condition is taken into account. The phase space is then defined by

$$\begin{aligned} X &\equiv (\mathbf{r} \in D,\ \mathbf{\Omega} \in S^2,\ E \in \mathbb{R}^+_G), \\ \partial X^{\pm} &\equiv (\mathbf{r} \in \Gamma^{\pm}(\mathbf{\Omega}),\ \mathbf{\Omega} \in S^2,\ E \in \mathbb{R}^+_G), \\ \Gamma^{\pm}(\mathbf{\Omega}) &\equiv (\mathbf{r} \in \partial D,\ \mathbf{n}_+(\mathbf{r}) \cdot \mathbf{\Omega} \gtrless 0), \end{aligned} \tag{35}$$

where D represents the domain partitioned into N computational regions, while S^2 and $\mathbb{R}^+_G$ are respectively the discrete-ordinate support for the angle and the multigroup discretization for the energy.

The discrete version of Eq. (34) is solved by power iterations. The scheme starts by fixing the initial guess for the eigenvalue and the fission source, $k^{(0)}$ and $F\psi^{(0)}$ respectively, then the equation

is solved in each group by inverting the operator $(L-H)$,

$$\begin{aligned}(L-H)\psi^{(i+1)}(x) &= \frac{1}{k^{(i)}}F\psi^{(i)}(x) \quad \text{for } x \in X, \\ \psi^{-,(i+1)}(x) &= 0 \qquad\qquad\qquad \text{for } x \in \partial X^{-}.\end{aligned} \tag{36}$$

The inversion process entails a set of thermal iterations for converging the scattering source in each energy group. These are the Gauss-Seidel thermal iterations

$$L\psi^{(t+1)}(x) = H_{dw}\psi^{(t+1)} + H_{up}\psi^{(t)} + q_{fs},$$

where (t) is the iteration index, while H_{up} and H_{dw} are the uper-triangular and the lower-triangular part of the scattering matrix, respectively. At each thermal iteration, a set of inner iterations is run for solving the spatial-angular distribution in each energy group by Eqs. (2) to (5).

Once the new flux is available, the fission source is updated and a new eigenvalue is computed

$$k^{(n+1)} = k^{(n)}\frac{(w, F\psi^{(n+1)})}{(w, F\psi^{(n)})}, \tag{37}$$

where $(\cdot,\cdot)$ is a scalar product acting on the discrete phase space X, while w is a weight-function. The algorithm stops when

$$\begin{aligned}\varepsilon_{\,k} &= \left|1 - \frac{k^{(n+1)}}{k^{(n)}}\right| < \epsilon_{\,k} \\ \varepsilon_F(\mathbf{r}) &= \left|1 - \frac{(1, F\psi^{(n+1)})(\mathbf{r})}{(1, F\psi^{(n)})(\mathbf{r})}\right| < \epsilon_F \quad \mathbf{r} \in D,\end{aligned}$$

where $\epsilon_{\,k}$ and $\epsilon_F(\mathbf{r})$ are respectively the tolerance on the eigenvalue error and the tolerance on the point-wise error of the fission source.

5. Non-overlapping Domain Decomposition

The domain decomposition consists in splitting the global domain into several overlapping or non-overlapping spatial subdomains. [17] Several successful application of the DDM has been applied to diffusion, [20], and to the PN transport equation, [19] [21]. IDT implements non-overlapping subdomains to minimize the mutual exchange of data, [18]. The phase space $X \equiv \bigcup_{u=1,..,U} X_u$ is decomposed in subspaces as

$$\begin{aligned}X_u &\equiv (\mathbf{r} \in D_u,\ \mathbf{\Omega} \in S^2,\ E \in \mathbb{R}^+_G) \\ \partial X^{\pm}_u &\equiv (\mathbf{r} \in \Gamma^{\pm}_u(\mathbf{\Omega}),\ \mathbf{\Omega} \in S^2,\ E \in \mathbb{R}^+_G) \\ \Gamma^{\pm}_u(\mathbf{\Omega}) &\equiv (\mathbf{r} \in \partial D_u,\ \mathbf{n}_{u+}(\mathbf{r}) \cdot \mathbf{\Omega} \gtrless 0)\end{aligned}$$

with the subdomain index varying from $u = 1, \ldots, U$. The whole geometry mesh is a 2-level partition: the first level is the decomposition of the geometry in U sub-geometries, i.e. $D \equiv$

$\bigcup_{u=1,U} D_u$, while the second level consists in the meshing of the sub-geometries, i.e. $D_u \equiv \bigcup_{n\in u}\bigcup_{\alpha\in n} D_{n,\alpha}$. Also, the flux $\psi(x)$, solution of Eq. (34), is decomposed as

$$\psi(x) = \sum_u \chi_u(x)\psi_u(x), \tag{38}$$

where $\chi_u(x)$ is the characteristic function of subdomain u, which is equal 1 if $x \in \partial X_u$ and 0 elsewhere. The continuity of the interface flux takes place as particle conservation condition at the interface boundaries of the subdomains, as

$$\psi_u^-(x) = \psi_v^+(x) \;\; x \in \partial X_u^- \cap \partial X_v^+ \text{ for all } v \cap u, \tag{39}$$

where the statement $v \cap u$ implies that v is neighbor of u, or more precisely, the subdomain v shares a part of its outgoing surface $\Gamma_v^+(\mathbf{\Omega})$ with the incoming surface of subdomain u.

The transport Eq. (34) is then split into U independent multigroup source problems of the type

$$\begin{cases} (L-H)_u\psi_u(x) = q_u(x) & x \in X_u\ , \\ \psi_u^-(x) = \psi_v^+(x) & x \in \partial X_u^- \quad \text{for all } v \cap u\ , \\ \psi_u(x) = 0 & x \in \partial X_u^- \cap X^-\ , \end{cases} \tag{40}$$

for $u = 1, \ldots, U$.

The fixed source $q_u(x)$ is given by the normalized fission production

$$q_u(x) = \frac{F_u\psi_u(x)}{k}. \tag{41}$$

Continuity condition (39) guarantees that the original solution $\psi(x)$ is preserved by the re-composition of the fluxes $\psi_u(x)$, i.e. Eq. (38), which are solutions of U independent problems of type (40). The problems (40) and the fission source update (41) are iteratively solved by power iterations. The process starts with an initial guess for the eigenvalue $k^{(0)}$, for the interfaces incoming boundary fluxes $\psi_u^{-,(0)}(x)$ and the fission distribution $q_u^{(0)}(x) = \frac{F_u\psi_u^{(0)}(x)}{k^{(0)}}$. One iteration solves U boundary-source problems of the type

$$\begin{cases} (L-H)_u\psi_u^{(i+1)}(x) = q_u^{(i)}(x) & x \in X_u\ , \\ \psi_u^{-,(i+1)}(x) = \psi_v^{+,(i)}(x) & x \in \partial X_u^- \quad \text{for } v \cap u\ , \\ \psi_u^{-,(i+1)}(x) = 0 & x \in \partial X_u^- \cap X^-. \end{cases} \tag{42}$$

The inversion of the operator $(L-H)_u$ requires local multigroup thermal iterations, which are performed in each subdomain to converge the scattering source. This is a crucial point of the algorithm: the subdomains do not transfer their boundary solutions until the end of local multigroup thermal and inner iterations. As it will be displayed in next sections, this characterizes the coarse-grained parallelism implemented in IDT and the consequent minimization of flux exchanges.

Once the local problems are solved, the eigenvalue, as well as the fission source are updated,

$$k^{(i+1)} = k^{(i)} \frac{\sum_{u=1,U} (w, F_u \psi_u^{(i+1)})}{\sum_{u=1,U} (w, F_u \psi_u^{(n)})} , \tag{43}$$

$$q_u^{(i+1)}(x) = \frac{F_u \psi_u^{(i+1)}(x)}{k^{(i+1)}} . \tag{44}$$

The algorithm stops if the following conditions are satisfied,

$$\varepsilon_{\,k} = \left| 1 - \frac{k^{(i+1)}}{k^{(i)}} \right| < \epsilon_{\,k},$$

$$\varepsilon_F = \left| 1 - \frac{\sum_g (F_u \psi_u^{(i+1)})^g}{\sum_g (F_u \psi_u^{(i)})^g} \right| < \epsilon_F \quad u = 1, U \;\; \mathbf{r} \in D_{u,R},$$

$$\varepsilon_\psi(x) = \left| 1 - \frac{\int_{2\pi^-} d\mathbf{\Omega} \;\; |\mathbf{n} \cdot \mathbf{\Omega}| \, \psi_u^{-,(i+1)}(x)}{\int_{2\pi^-} d\mathbf{\Omega} \;\; |\mathbf{n} \cdot \mathbf{\Omega}| \, \psi_u^{-,(i)}(x)} \right| < \epsilon_\psi \quad u = 1, U \;\; x \in \partial X_{u-}.$$

which are respectively the conditions to be met for k-effective, the fission integral and the boundary interface incoming currents.

It is important to notice that the DDM implementation in IDT preserves the original transport solution through the continuity of the interface angular flux spatial moment per each energy group g, i.e.

$$\psi^-_{u,g,n,d,s'} = \psi^+_{v,g,n,d,s}. \tag{45}$$

6. The Coarse-Mesh Finite Differences

The CMFD is a nonlinear method based on the conservation of the neutron balance. [22] [23] [24] In the present work, the finite-difference diffusion equation is applied on a coarse phase space to speed up the convergence of DDM outer iterations. As mentioned, the original spatial XYZ grid, that lodges the HCCs, is homogenized in space and coarsened in energy. In IDT, the spatial nodes of the CMFD mesh may contain one or more HCCs. Indexing by c the coarse node and by h the coarse energy group, the balance equation for the couple (c, h) is

$$\sum_{k \in c} \frac{A_{c,k}}{V_c} J^h_{c,k} + \Sigma^h_c \phi^h_c - \sum_{h \neq h'} \Sigma^{hh'}_{k,c} \phi^{h'}_c = \frac{1}{\lambda} \sum_{h \neq h'} (\chi \upsilon \Sigma)^{hh'}_{f,c} \phi^{h'}_c , \tag{46}$$

where ϕ^h_c is the average scalar flux of the node and $J^h_{c,k}$ is the net current on the face k of the node, while λ is the eigenvalue of the problem. By symbol $k \in c$ we intend the faces of the node, which have area equal to $A_{c,k}$. The cross section are obtained by flux-weighted homogenization. Indexing with $(i + 1/2)$ the transport flux and using the symbol $n \in c$ to refer the HCCs composing node c,

the CMFD cross sections are

$$\Sigma_c^h = \frac{\sum_{g\in h}\sum_{n\in c}\sum_{\alpha\in n} V_{n,\alpha}(\Sigma^g_{n,\alpha} - \Sigma^g_{n,\alpha,0})\phi^{g,(i+1/2)}_{n,\alpha}}{\sum_{g\in h}\sum_{n\in c}\sum_{\alpha\in n} V_{n,\alpha}\phi^{g,(i+1/2)}_{n,\alpha}}, \tag{47}$$

$$\Sigma^{hh'}_{k,c} = \frac{\sum_{g\in h}\sum_{g'\in h'}\sum_{n\in c}\sum_{\alpha\in n} V_{n,\alpha}\Sigma^{gg'}_{k,n,\alpha,0}\phi^{g',(i+1/2)}_{n,\alpha}}{\sum_{g'\in h'}\sum_{n\in c}\sum_{\alpha\in n} V_{n,\alpha}\phi^{g',(i+1/2)}_{n,\alpha}}, \tag{48}$$

$$(\chi\upsilon\Sigma)^{hh'}_{f,c} = \frac{\sum_{g\in h}\sum_{g'\in h'}\sum_{n\in c}\sum_{\alpha\in n} V_{n,\alpha}\sum_{\iota\in\alpha}\chi^g_\iota(\upsilon\Sigma)^{g'}_{\iota,\alpha}\phi^{g',(i+1/2)}_{n,\alpha}}{\sum_{g'\in h'}\sum_{n\in c}\sum_{\alpha\in n} V_{n,\alpha}\phi^{g',(i+1/2)}_{n,\alpha}}. \tag{49}$$

In equations (47) to (49), the transport scalar flux $\phi^{g,(i+1/2)}_{n,\alpha}$ is the zero-order spatial component of the vector $\phi^{(i+1/2)}_{n,\alpha,k=0,l=0}$. This is possible because the zero-order component is orthogonal to linear components, since the coordinate system of each region is centered in its center of mass. Furthermore, IDT uses the fission matrix defined by Eq. (49) because of the collapsing of the fissile isotopes index, indicated in the formula by ι. This preserves the accuracy of the reference transport cross section library that contains the $\upsilon\Sigma$ cross section per each fissile isotope.

The CMFD solution is obtained by solving the balance equation (46) for the flux. The currents $J^h_{c,k}$ are computed by an artificial finite-difference Fick's law, i.e.

$$J^h_{c,k} = -d^h_{c,k}(\phi^h_{c,k} - \phi^h_c) + \widehat{d}^h_{c,k}(\phi^h_{c,k} + \phi^h_c), \tag{50}$$

where the net current $\phi^h_{c,k}$ is the average interface scalar flux of the face of the node. The coefficient $d^h_{c,k}$ is the finite-difference diffusion coefficient given by

$$d^h_{c,k} = \frac{2}{3\Delta^\perp_{c,k}\Sigma^h_{c,tr}} \tag{51}$$

with $\Sigma^h_{c,tr}$ as the transport cross section, $\Sigma^h_{c,tr} = \Sigma^h_c - \Sigma^h_{c,s,1}$, and with $\Delta^\perp_{c,k}$ as the node thickness perpendicular to face k. The interface flux $\phi^h_{c,k}$ is eliminated from equation (50) by imposing the continuity of the current with the contiguous cells c'

$$J^h_{c,k} = -J^h_{c',k} \quad \text{for all } c' \cap c. \tag{52}$$

(The sign of the current in the continuity equation changes since the current $J^h_{c,k}$ in the balance (46) becomes positive/negative if it is outgoing/incoming from/into node c.) Then, using Eq. (50) into Eq. (52), for the interface flux, and a back-substitution of (52) into (50), a classical finite-difference form takes place expressing the current as a linear combination of two fluxes of contiguous nodes,

$$J^h_{c,k} = \frac{(d^h_{c',k} - \widehat{d}^h_{c',k})(d^h_{c,k} + \widehat{d}^h_{c,k})}{(d^h_{c,k} - \widehat{d}^h_{c,k}) + (d^h_{c',k} - \widehat{d}^h_{c',k})}\phi^h_c - \frac{(d^h_{c,k} - \widehat{d}^h_{c,k})(d^h_{c',k} + \widehat{d}^h_{c',k})}{(d^h_{c,k} - \widehat{d}^h_{c,k}) + (d^h_{c',k} - \widehat{d}^h_{c',k})}\phi^h_{c'}. \tag{53}$$

Preserving the transport current is the key point for equivalence. The non-linear diffusion will respect the particle balance if it will be capable of reproducing the transport currents. To this end,

the coefficient $\widehat{d}_{c,k}^{h}$ is computed on-the-fly to iteratively adjust Eq. (50) in order to comply with the transport balance equation. This task is achieved by the formula

$$\widehat{d}_{c,k}^{h} = \left[\frac{J_{c,k}^{h,(i+1/2)} + d_{c,k}(\phi_{c,k}^{h,(i+1/2)} - \phi_{c}^{h,(i+1/2)})}{(\phi_{c,k}^{h,(i+1/2)} + \phi_{c}^{h,(i+1/2)})} \right] \tag{54}$$

that forces Eq. (50) to preserve the transport current at the convergence.

6.1. Specific aspects of IDT implementation

During transport iterations, IDT computes and stores three partial angular moments of the flux on each node interface. They are: the interface partial scalar flux,

$$\phi_{c,k}^{+,h,(i+1/2)} = \sum_{g \in h} \frac{\sum_{n \in c} \sum_{s \in k} A_{n,s} \sum_{d \in 2\pi_s^+} w_d \, \psi_{n,d,s,0}^{+,(i+1/2)}}{\sum_{n \in c} \sum_{s \in k} A_{n,s}}, \tag{55}$$

the outgoing current

$$J_{c,k}^{+,h,(i+1/2)} = \sum_{g \in h} \frac{\sum_{n \in c} \sum_{s \in k} A_{n,s} \sum_{d \in 2\pi_s^+} w_d \, |\mathbf{\Omega}_d \cdot \mathbf{n}_s| \, \psi_{n,d,s,0}^{+,(i+1/2)}}{\sum_{n \in c} \sum_{s \in k} A_{n,s}} \tag{56}$$

and the first component of the second-order angular moment, namely $\Theta_{c,k}^{+,(i+1/2),h}$,

$$\Theta_{c,k}^{+,(i+1/2),h} = \sum_{g \in h} \frac{\sum_{n \in c} \sum_{s \in k} A_{n,s} \sum_{d \in 2\pi_s^+} w_d \, |\mathbf{\Omega}_d \cdot \mathbf{n}_s|^2 \, \psi_{n,d,s,0}^{+,(i+1/2)}}{\sum_{n \in c} \sum_{s \in k} A_{n,s}} \tag{57}$$

The net quantities are then reconstructed as

$$\phi_{c,k}^{h,(i+1/2)} = \phi_{c,k}^{+,h,(i+1/2)} + \phi_{c',k'}^{+,h,(i+1/2)}, \tag{58}$$

$$J_{c,k}^{h,(i+1/2)} = J_{c,k}^{+,h,(i+1/2)} - J_{c',k'}^{+,h,(i+1/2)}, \tag{59}$$

$$\Theta_{c,k}^{h,(i+1/2)} = \Theta_{c,k}^{+,h,(i+1/2)} + \Theta_{c',k'}^{+,h,(i+1/2)}, \tag{60}$$

for each $c' \cap c$, such that the area $A_{c',k'} \equiv A_{c,k}$.

The second-order angular moment $\Theta_{c,k}^{h,(i+1/2)}$ is used to enhance the CMFD effectiveness. This is done by modifying the diffusion coefficient of Eq. (51) in the following way,

$$d_{c,k}^{h} = f^{(i+1/2)} \frac{\Theta_{c,k}^{h,(i+1/2)}}{\phi_{c,k}^{h,(i+1/2)}} \frac{2}{\Delta_{c,k}^{\perp} \Sigma_{c,tr}^{h}}, \tag{61}$$

where $f^{(i+1/2)}$ is a stability parameter computed on-the-fly by transport quantities. A detailed description of the computational algorithm for $f^{(i+1/2)}$ is contained in [9]. The factor $f^{(i+1/2)} \frac{\Theta_{c,k}^{h,(i+1/2)}}{\phi_{c,k}^{h,(i+1/2)}}$

is such that it goes to $\frac{1}{3}$ in diffusive regime. Moreover, in thick diffusive problems, $f^{(i+1/2)}$ amplifies the diffusion coefficients for avoiding instabilities. The stability parameter $f^{(i+1/2)}$ ensures that the CMFD is effective and stable for a wide range of transport regimes.

IDT computes and stores interface quantities by using two steps: the first is averaging the HCC boxes, the second is averaging the necessary number of HCCs composing the CMFD node. This two-steps homogenization allows for the automatic generation of an arbitrary CMFD mesh.

The first step is performed inside the transport sweep, where currents are averaged on each interface of the finer grids, i.e. the HCC grid. The second is performed while constructing the CMFD coefficients. Then, the interface quantities are averaged on the face of the node. An illustration of the 2-step homogenization used for computing the CMFD interface angular moments is given in Fig. (3). In the example, a single node of the CMFD homogenizes $2 \times 2 \times 2$ HCCs.

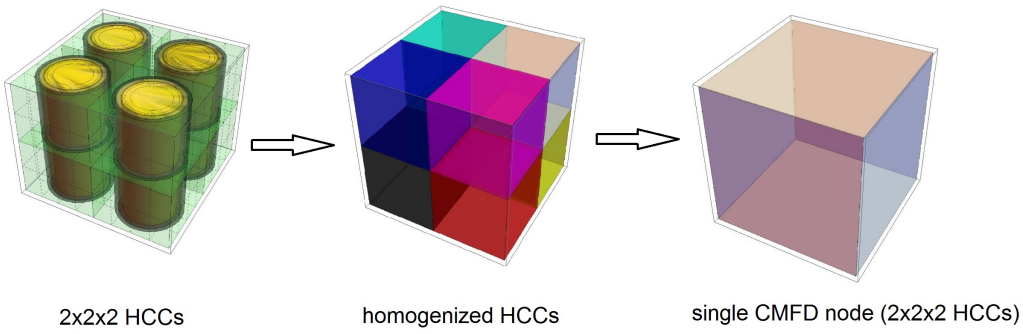

Figure 3: Example of a 2-step homogenization: the interface angular moments are accumulated over the faces of the HCCs, then they are re-homogenized on the faces of the CMFD node.

The motivation justifying the storage of partial quantities is threefold. First, IDT uses the intermediate grid, i.e. the one defined by homogenized HCCs, to accelerate by the CMFD inner transport iterations.

Second, IDT allows for the utilization of 4 types of CMFD operators, each CMFD method has a specific current-flux closure relation. The default CMFD method is the MCHN (Moon, Cho, Noh and Hong) that is characterized by Eq. (50) that we report hereafter,

$$J^h_{c,k} = -d^h_{c,k}(\phi^h_{c,k} - \phi_c) + \widehat{d}^h_{c,k}(\phi^h_{c,k} + \phi^h_c). \tag{62}$$

The MCHN uses the net current as well as the AFC (average flux correction), that is the second available method,

$$J^h_{c,k} = -d^h_{c,k}(\phi^h_{c,k} - \phi_c) + \widehat{d}^h_{c,k}\phi^h_c. \tag{63}$$

While the third and the fourth, which are respectively the pCMFD (partial-current CMFD) and the AFC-pCMFD, use partial currents, that is

$$J^{+,h}_{c,k} = -d^h_{c,k}(\phi^h_{c,k} - \phi_c) + \widehat{d}^h_{c,k}(\phi^h_c + \phi^h_c), \tag{64}$$

and

$$J_{c,k}^{+,h} = -d_{c,k}^{h}(\phi_{c,k}^{h} - \phi_c) + \widehat{d}_{c,k}^{h}\phi_c^h. \tag{65}$$

The third motivation is due to the domain decomposition. In fact, the synergy among transport and CMFD operators relies on the same spatial decomposition. As explained in the next section, the reconstruction of updated interfaces quantities requires the knowledge of partial currents among subdomain interfaces.

6.2. DDM for the CMFD

As anticipated, the transport and the CMFD operator share the same spatial decomposition. The CMFD mesh is automatically generated by user's inputs. As depicted in Fig. (4), the user specifies the number of coarse mesh in each subdomain, then the mesh generator adapts user's instructions to the *pivot* grid. The local CMFD mesh is generated by coarsening the HCC grid. The mesh generator takes care of the conformity of the mesh among the subdomains.

With reference to Fig. (4), the net current on interfaces of the subdomain (red lines in the picture) is computed by the difference

$$J_{u,c,k}^{h,(i+1/2)} = J_{u,c,k}^{+,h,(i+1/2)} - J_{u,c',k'}^{+,h,(i+1/2)},$$

for two contiguous nodes c and c' belonging to the same subdomain u, while the net current on interfaces among subdomains (internal black lines in the picture) are computed by

$$J_{u,c,k}^{h,(i+1/2)} = J_{u,c,k}^{+,h,(i+1/2)} - J_{v,c',k'}^{+,h,(i+1/2)},$$

for two contiguous nodes belonging to the two neighbor subdomains, u and v, respectively. For outer boundary surfaces, the net current is computed using the local boundary conditions as incoming current of the node.

For each outer transport loop, a CMFD multigroup problem is established. In IDT, the user can coarsen the energy mesh as well. The multigroup CMFD problem is then solved by global power iterations. At each CMFD power iteration, subdomains are computed independently. Indeed, as has been done for the transport, a local multigroup fixed-source problem is established in each subdomain. The scattering source of the CMFD operator is converged by thermal Gauss-Seidel iterations and by using a BiCGStab Krylov solver for the solution of the one-group problems, [26]. A detailed description of the DDM iteration scheme is presented in Algorithm 1 (see Section 5.3).

6.3. Automatic construction of the *pivot* grid, the CMFD mesh and the subdomain-to-process mapping

The automatic construction of the grids relies on the Cartesian nature of the IDT mesh, which is established on a XYZ fine grid lodging the HCCs.

The algorithm generating the fine-to-coarse map is the following.

Indicating by N the number of fine steps on a generic axis of the finer grid and by M the number of desired coarse steps, which is user's input such that $M \leqslant N$, the initial number of fine steps per

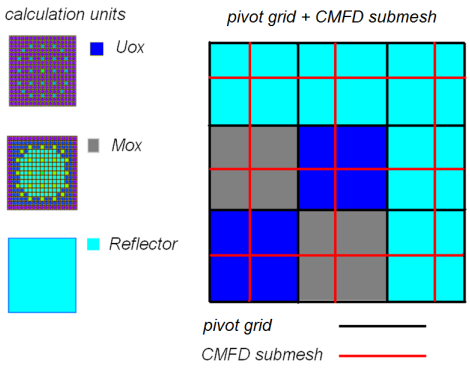

Figure 4: Example of a 2D coarse mesh applied to the C5-G7 MOX benchmark: the CMFD mesh is generated by the union of the pivot grid (black lines) and the user-defined subdomain grid (red line). The mesh generator assures the conformity of the CMFD grid among the subdomains. In this example, the CMFD grid is composed of 2x2 CMFD nodes per each subdomains.

coarse steps is fixed by

$$n = \max(1, \left\lfloor \frac{N}{M} \right\rfloor)$$

then the number of fine steps per each coarse step m is computed as

$$N_m = \begin{cases} n+1 & \text{for } m = 1, ..., N(\mod M) \\ \\ n & \text{for } m = N(\mod M)+1, ..., M \end{cases} \tag{66}$$

so that the first $N(\mod M)$ coarse steps will receive sequentially $n+1$ steps, while the remaining will contain only n. This algorithm is applied to each axis of fine XYZ grid, and, by extending the notation to X, Y and Z, the subdomain (m_x, m_y, m_z) of the *pivot* grid will contain $N_{m_x} N_{m_y} N_{m_z}$ HCCs.

The algorithm also applies to automatically generate the coarse mesh of the CMFD in each subdomain. It is re-iterated on a second spatial level inside the subdomain grid. Thus, the initial number of fine steps per CMFD step is fixed to

$$n_m = \max(1, \left\lfloor \frac{N_m}{C_m} \right\rfloor),$$

with C_m as the number of coarse steps of the CMFD (user's datum) associated to m, while

$$N_{m,c} = \begin{cases} n_m + 1 & \text{for } c = 1, ..., N_m(\mod C_m) \\ \\ n_m & \text{for } c = N_m(\mod C_m)+1, ..., C_m \end{cases} \tag{67}$$

is the effective number of fine steps for each coarse step c relative to the subdomain step m . Thus, the number of HCCs in the CMFD node (c_x, c_y, c_z) of the subdomain (m_x, m_y, m_z) will be $N_{m_x,c_x} N_{m_y,c_y} N_{m_z,c_z}$.

The data distribution needs the mapping among subdomains and MPI processes. The number of subdomains per process, U_p, is obtained by the same algorithm. Here,

$$s = \max(1, \left\lfloor \frac{U}{P} \right\rfloor)$$

is the initial value of the subdomain per process, with $U = M_x M_x M_x$ as the total number of subdomains and P as the number of processes defined by the user, so that

$$U_p = \begin{cases} s+1 & \text{for } c = 1, ..., U(\mod P) \\ s & \text{for } c = U(\mod P) + 1, ..., P \end{cases} \tag{68}$$

is the number of subdomains associated to the process p. The spatial distribution of the subdomain is done as depicted in the Fig. (5). The subdomains are assigned by sweeping the *pivot* grid in a *serpentine* path, as shown in the picture. This geometrical mapping guarantees that the subdomains assigned to a process share at least one face, limiting, thus, the inter-node MPI communications.

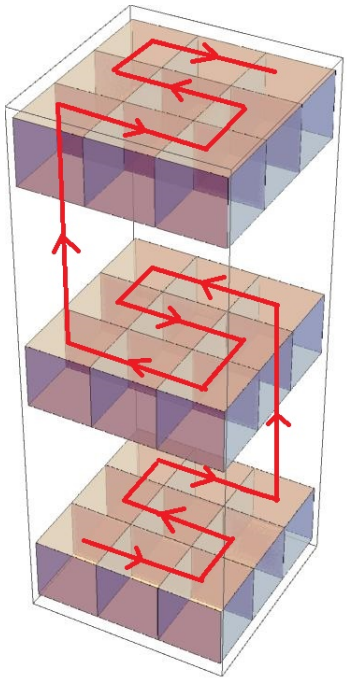

Figure 5: Illustration of the *serpentine* sweeping algorithm on a $3 \times 3 \times 3$ pivot grid.

As an example of the subdomain splitting algorithm, Figure (6) illustrates an application on a 3x3x3 *pivot* grid distributed on 4 MPI processes.

6.4. Automatic identification of repetitive geometrical patterns

Once the subdomains geometries constructed, the IDT interface automatically identifies repetitive patterns of the geometrical decomposition. The subdomains that share the same spatial mesh and

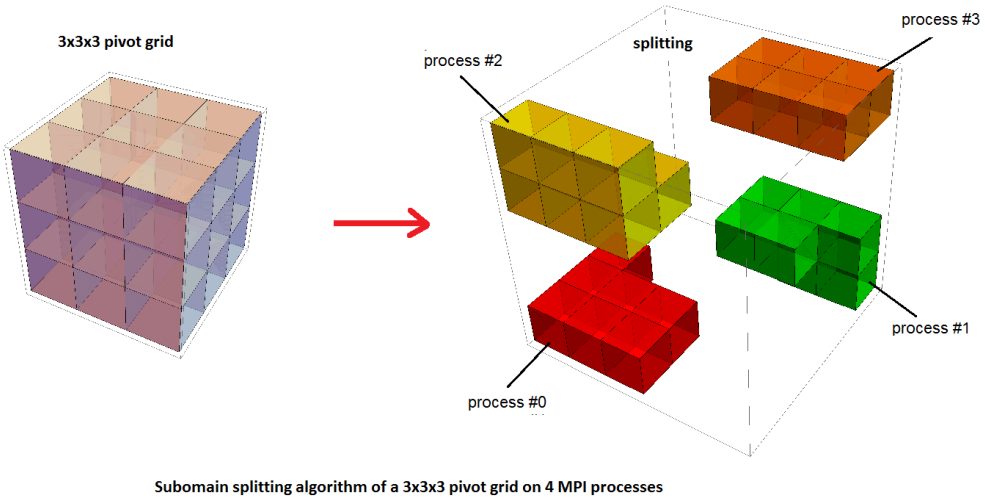

Figure 6: Example of the automatic splitting algorithm applied to a $3 \times 3 \times 3$ subdomain grid distributed on 4 MPI processes.

the same spatial distribution of the material are grouped in Generating Calculation Units (GCU). A GCU contains a unique mesh and material distribution and also a unique set of options, as the angular and spatial discretization, the HCC surface mesh option, the tracking options and inner acceleration options. As will be explained in the next section, this ensures an optimization of the data distribution. The GCU are then assigned over the *Pivot* Calculation Unit (PCU) and this allows for the reconstruction of the global geometry. The PCU lodges the *pivot* geometry and the GCU mapping and also the options for constructing the CMFD operator.

An example of the automatic generation of the GCU on the geometry of the C5G7 Mox benchmark is depicted in Fig. 1-D.

6.5. Hybrid parallelism algorithm

As anticipated, the parallelism within IDT is based on the definition of the Generating Calculation Unit (GCU) and the Effective Calculation Unit (ECU). As shown, once the geometry is decomposed, the subdomains sharing the same geometry and materials define a unique GCU. The GCU is devoted to the storage and the computation of non-mutable data as, for example, the cross section library and the transport HCC matrices ($\underline{\mathbf{I}}, \underline{\mathbf{C}}, \underline{\mathbf{T}}, \underline{\mathbf{E}}$). Moreover, the GCU is used to initialize the flux in each subdomain by running one or several multigroup iterations with infinite-lattice boundary conditions. The definition of the GCU allows for the minimization of data for the subdomains that share the same MPI process. Indeed, in case of hybrid MPI-OpenMP parallelism, the IDT solver uses a map that gives the correspondence among the GCUs and the MPI process. This map is obtained by combining the map giving the correspondence among the subdomains and MPI process, and the maps giving the distribution of the GCU on the *pivot* geometry. Setting g as the GCU index and $\mathcal{G}$ as the GCU-to-*pivot* mapping

$$\mathcal{G} : (m_x, m_y, m_z) \rightarrow g$$

and by

$$\mathcal{P} : (m_x, m_y, m_z) \rightarrow p$$

the subdomain-to-process map, the map assigning the GCU to the process is obtained as

$$\mathcal{G} \circ \mathcal{P}^{-1} : p \rightarrow g.$$

This map is used for setting the GCU data on the process and run a first initialization step. The GCU together with the PCU are distributed and allocated on the MPI processes. In particular, the PCU is copied in each processes. Then, each GCU generates a number of Effective Calculation Units (ECU) on the process that is equal to the number of times the GCU is assigned to the process. In this manner a single ECU is generated per each subdomain associated to the process. The ECUs associated to a GCU share the same non-mutable memory of the reference GCU but allocate the mutable part of the memory, as the memory for the flux and the CMFD operator. This means that the memory occupied by coefficients and cross sections is defined by the number of GCUs assigned to a process, while the memory occupied by the flux is determined by the number of ECUs. A schematic example of the memory organization is presented in Fig. (7). The communications among MPI processes engages only subdomains of the same operator: the MPI send/receive directives are embedded in the power iterations of the two operators and are called only by the subdomains that share one or more faces with subdomains belonging to other processes. Instead, the data transfer among the transport and the CMFD operators is done on the same MPI process because both operators share the same spatial decomposition.

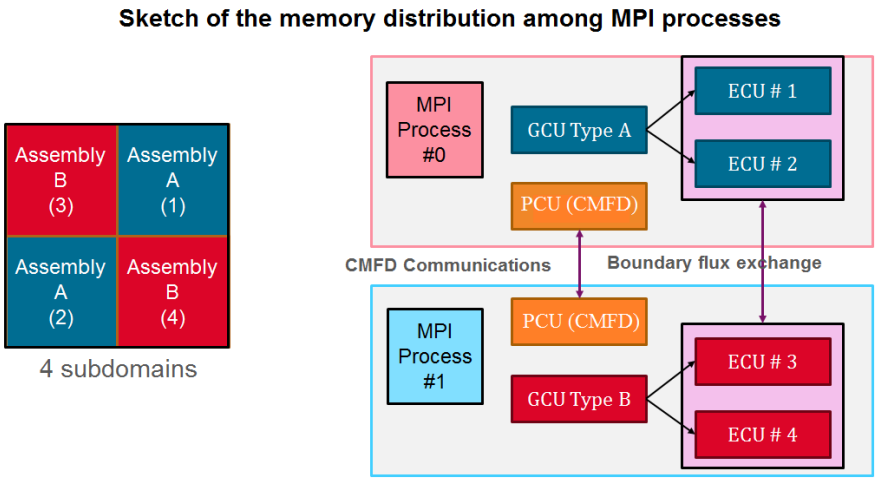

Figure 7: Example of a domain-splitting on a 2x2 colorset composed by 2 GCUs (blue and red in the picture). The red one is loaded in process #0 and the blue one on the process #1. The communications among MPI processes engages only subdomains of the same operator. Instead, because the CMFD and the transport share the same domain decomposition, the data transfer among the transport and the CMFD operators is done on the same MPI process.

In IDT, the hybrid parallelism is realized by partitioning the global list of subdomains, defined as

$L = \{u \in \mathbb{N} : u = 1, ..., U\}$, in a set of sub-lists L_p, such that

$$L = \bigcup_{p=1,P} L_p$$

with P being the number of MPI processes. Each sub-list L_p has a number of subdomains equal to U_p (which is computed thanks to Eq. (68)), the subdomain are then affected to L_p following the serpentine path depicted in Fig. (5). Each list L_p is solved on shared-memory parallelism thanks to OpenMP directives. Algorithm 1 shows the DDM iteration algorithm seen by a generic MPI process. This algorithm supposes that the number of processes is less then the number of subdomains.

In the particular case of a single element per list, i.e. when the number of nodes is equal to the number of subdomains, the solver uses shared-memory parallelism to perform the angular transport sweeping inside the subdomain.

7. Description of the resources

The Cobalt cluster is composed of nodes interconnected by Infiniband, [28]. Each node contains an amount of CPUs plugged on sockets. The sockets are interconnected by a QuickPath Interconnect (QPI). Each CPU contains a certain amount of cores, while each core is equipped of SIMD computational units. The Cobalt cluster is subdivided in 4 partitions. In this test suite, IDT runs on the Broadwell partition. Main characteristics of the partition are:

- 1412 nodes,
- 2×14-cores CPU Intel Broadwell@2,4GHz (AVX2),
- 28 cores / node for a total of 39 536 cores,
- 128 GB of memory per node.

The IDT sources compiled within the hybrid MPI/OpenMP framework of Cobalt using the Gnu 4.8.5 compiler and the OpenMPI library version 1.8.8. Typical instructions for generating the `idt.exe` file are

```
-bash -4.1 mpif90 -O3 -qopenmp -o mpi_omp_tes mpi_omp_test.for.
```

8. 3D benchmarks: the 3D C5G7 MOX benchmark and the CASL Problem #3 3x3 PWR colorset

Tables 1 shows the results of the three-dimensional C5G7 MOX benchmark, [15], [16]. Simulations have been obtained by modeling the geometry by 56258 regions. The radial discretization of each pin cell is obtained by using 3-region HCC having 1 cylinder for the fuel and 1 ring in the moderator, and a surface discretization 3-3-2, which means a 3x3 surface grid on the top and bottom faces and 3x2 grid on the remaining lateral faces. The axial step is 3.57 cm for each HCC,

Algorithm 1 DDM eigenvalue iteration algorithm seen by process $\#p$

Initialization step $i = 0$: $\psi_u^{(0)}$, $\psi_u^{-,(0)}$, $k^{(0)}$ for all for $u \in L_p$
for i=0; $i = i + 1$; until the convergence of ψ_u, ψ_u^- and k **do**
  MPI send/receive: update b.c. $\psi_u^{-,(i+1)} = \psi_{u'}^{+,(i)}$ for $u' \notin L_p$ and $u' \cap u$
  OMP parallel loop
  for each subdomain $u \in L_p$ **do**
    update b.c.: $\psi_u^{-,(i)} = \psi_{u'}^{-,(i)}$ for $u' \in L_p$ and $u' \cap u$
    update fission source:

$$q_u^{(i)} = \frac{F_u \psi_u^{(i)}}{k^{(i)}}$$

    solve:

$$(L - H)_u \psi_u^{(i+1/2)} = q_u^{(i)}$$

    homogenize cross sections
    compute the CMFD operator for u.
  end for
  MPI send/receive (synchronized): update $J_u^{+,(i+1/2)} = J_{u'}^{-,(i+1/2)}$ for $u' \notin L_p$ and $u' \cap u$
  Initialization step of the CMFD solver $j = 0$:

$$\lambda^{(0)} = k^{(i)}, \text{ and } \phi_u^{CMFD,(0)} = \int \psi_u^{(i+1/2)}$$

  for j=0; $j \leftarrow j + 1$; until convergence of ϕ_u^{CMFD}, and λ: **do**
    MPI send/receive for exchanging the flux $\phi_{u'}^{CMFD}$ for $u' \notin L_p$ and $u' \cap u$
    OMP parallel loop
    for for each subdomain $u \in L_p$ **do**
      finalize the CMFD operator A_u^{CMFD} on the boundary
      update the fission source:

$$q_u^{CMFD,(j)} = \frac{F_u \phi_u^{CMFD,(j)}}{\lambda^{(j)}}$$

      exchange $\phi_{u'}^{CMFD}$ for all $u' \cap u$ and $u' \in L_p$
      solve:

$$A_u \phi_u^{CMFD,(j+1)} = q_u^{CMFD,(j)}$$

    end for
    update: $\lambda^{(j+1)} = \lambda^{(j)} \frac{\sum_u (w, F_u \phi_u^{CMFD,(j+1)})}{\sum_u (w, F_u \phi u^{CMFD,(j)})}$
  end for
  update the eigenvalue: $k^{(i+1)} = \lambda$
  update the flux:

$$\psi_u^{(i+1)} = \psi_u^{(i+1/2)} \frac{\phi_u^{CMFD}}{\int \psi_u^{(i+1/2)}} \text{ and } \psi_u^{\pm,(i+1)} = \psi_u^{\pm,(i+1/2)} \frac{\phi_u^{CMFD}}{\int \psi_u^{(i+1/2)}}$$

end for

Table 1: Reactivity and pin power error analysis of the 3D C5G7 MOX benchmark.

		Unrodded	Rodded A	Rodded B
K-effective		1.14336	1.12827	1.07753
Reactivity error (pcm)		24	-19	-22
Max pin power error (%)		1.5	1.2	1.3
Min pin error (%)		-0.9	-0.4	-0.8
RMS (%)		0.3	0.4	0.45
central UOX (%)	Bottom	-0.13	-0.04	-0.18
	Midle	-0.13	-0.24	-0.53
	Top	0.06	-0.44	0.06
flat UOX (%)	Bottom	0.06	0.14	0.21
	Midle	0.03	0.12	0.03
	Top	0.12	0.10	0.12
MOX (%)	Bottom	0.04	-0.19	-0.73
	Midle	0.04	0.08	0.03
	Top	0.27	0.03	0.31
# of power iterations		25	25	24
# of inner iterations		665	666	652
Elapsed Time (sec.)		942	945	920

except for the top reflector interface which has three floors of 1.19 cm step. The angular quadrature is a modified S8 Level-Symmetric quadrature formula that flattens the directions along the Cartesian axis. The total memory footprint is 19 GB. The calculation has been accelerated by the multigroup CMFD using the homogenized pin-cell grid as the coarse mesh. Furthermore, calculations have been run in parallel framework using a domain decomposition consisting in 3x3x4 subdomains, see Fig. 1-D. All simulations have been run on 9 threads (3 in hyper-threading) in OpenMP shared-memory parallelism on a standard workstation, i.e. Intel Xeon E5-2620, 2.00GHz, 64 GB of memory, 6 cores. The error tolerance is 1 pcm on the k-effective and on the multigroup flux distribution, while 10 pcm on the fission source and on the interface current. The method exhibits a maximum pin-power error of 1.5% over the three configurations with respect to the reference Monte Carlo solution. HCC model guarantees high fidelity 3D calculation while minimizing the number of regions and angles. Indeed, the calculation uses 10 to 100 times less unknowns than the most popular Method Of Characteristic (MOC), while the runtime is less then 16 minutes for all configurations, as shown in the last row of Tab. 1.

The second test is the problem #4 of the VERA benchmark specification [14]. This problem has the presence of spacer grids, PYREX rods and a central B4C control rod. The domain splitting used in this case is of 250 subdomains distributed on a $5 \times 5 \times 10$ pivot grid. Each subdomain is composed on the average by $6 \times 6 \times 12$ HCCs, having transverse dimension 1.26×1.26 cm2 and an average height of 4 cm. The surface mesh used on the HCC surface is a 3-3-2 configuration with 9 sub surfaces for the horizontal faces and 6 sub surfaces for the vertical ones, with a total of 42 sub surfaces per HCC. The CMFD grids were obtained by coarsening the original $29 \times 29 \times 120$

Table 2: Reactivity, pin power error and performances of the 3D VERA problem #4.

k-effective	0.99901
Reactivity error	54 pcm
Pin-power error	0.81%
RMS	0.21%
Total DoF	2.134 321
# of Threads/Node	25
# Node	10
average Memory/Node	65 GB
Elapsed time	1h.09min
# of outer DDM iterations	8
# of outer CMFD iterations	342
% Time for Tansport	42 %
% Time for CMFD	46 %
% Time for HCC coef.	12 %

lattice grid, with and explicit description of the water gap, to a $24 \times 24 \times 120$ grid, where the water gap is smeared with the neighbor cells. The groups of the original P3 cross section library are also collapsed from 47 to 8. The reactivity errors and the relative errors for the 3D assembly-wise power distribution are compared in Table 2 to the McCARD reference solution provided by the Department of Nuclear Engineering, Seoul National University, thanks to the collaboration with Professor H. G. Joo in 2017, [13]. Table 2 contains also the total number of degrees of freedom and the execution time.

9. Strong Scalability up to 1000 cores

This section shows the results of the strong-scalability test preformed on the rodded-B configuration of OCDE/NEA 3D C5-G7-MOX benchmark, [27].

As illustrated in Tab. (3), the benchmark has only 7 energy groups and P0 approximation of the scattering kernel. However, the discretization of the spatial and angular variables has been increased to 375 642 regions and 168 directions to achieve an error of 10 pcm on the k-effective with respect to the MCNP reference and less than 1% of error on the reactivity. For the huge amount of regions and directions, this test stresses particularly the IDT routines devoted to the one-group transport sweep. Moreover, because of the local multigroup iterations performed in each

Table 3: Problem size.

Anisotropy order	P0
Number of regions	375 642
Number of outer boundary surfaces	15 606
Number of groups	7
Number of HCC	10
Number of outer boundary surface meshes	216 000
Number of directions per octant	10
Total number of directions	80
Number of angular flux spatial moments	4
Number of interface angular flux spatial moments	3
Number of angular moments for the source	16
Number of materials	8
Number of surfaces per HCC	6
Number of CMFD coarse cells	241 875
Number of CMFD coarse energy groups	24
Total memory on a single node (GB)	19

subdomain, the scalability of the DDM implementation is expected to grow with the number of groups since, the more time is spent in local iterations, the more the parallelization will be effective. Furthermore, for reasons that will be detailed later on, the initialization step will proportionally count more as the number of groups decreases. For those reasons, the C5-G7-MOX is a sort of "*stress-test*" for the DDM implementation in IDT. In other words: if IDT proves its scalability for 7-group P0 C5G7 MOX benchmark, it will certainly scale better for higher number of groups and for higher anisotropy orders.

Table (4) shows the subdomain grids used to decompose the domain and the resources allocated on the Cobalt cluster. In the runs, the number of processors matches always the number of subdomains. Moreover the number of MPI processes is equal to the number of nodes in each run. For example, for the $3 \times 3 \times 3$ configuration, three nodes are allocated for three MPI process, in this manner each node computes 9 subdomains using 9 threads. This particular implementation guaranties that two neighbor cells sharing the same node exchang the interface boundary flux without communications, i.e. without MPI send/receive directives.

In this test suite, the number of nodes is always equal to the number of MPI processes. We prevent that certain configurations can be penalized because of the high number of nodes allocated: for

Table 4: Subdomains subdivision and resources associated for the strong scalability test.

# of Subdomains	nodes	allocated threads/node	cores
$2 \times 2 \times 2$	2	4	8
$3 \times 3 \times 3$	3	9	27
$4 \times 4 \times 4$	4	16	64
$5 \times 5 \times 5$	5	25	125
$6 \times 6 \times 6$	8	27	216
$8 \times 8 \times 8$	32	16	512
$9 \times 9 \times 9$	27	27	729
$10 \times 10 \times 10$	40	25	1000

example the $8 \times 8 \times 8$ *pivot* grid has been run on more nodes than the $9 \times 9 \times 9$ configuration.

In the following analysis, it is considered:

- the elapsed time T_E as the time elapsed between the beginning and the end of the calculation.
- the initialization time T_{INI} as the time elapsed in the initialization step,
- the solution time T_{DDM} as the time spent in the DDM iterations, which takes into account the transport and the CMFD iterations.

As a consequence of the definition, the relation

$$T_E = T_{INI} + T_{DDM}$$

holds. The time T_{DDM}, that is also the solution time, is decomposed as

$$T_{DDM} = T_{TRA} + T_{CMFD},$$

where T_{TRA} is the time elapsed in the DDM transport iterations while T_{CMFD} is the time elapsed in the global CMFD operator. The time has been measured with the time diagnostics embedded in the IDT solver. In particular, the work of each processor is monitored using the FORTRAN directive `CPU_TIME` that returns the elapsed CPU time in seconds.

9.1. Nominal scalability

Configurations of Tab. (4) has been run fixing the number of inner and outer iterations. In particular, it has been considered

- 10 outer iterations, i.e. 10 DDM power iterations,
- 1 thermal iteration per subdomain and per external,

- 10 inner iterations per group and per subdomain,
- 5 power iterations in the CMFD per each DDM outer iteration,
- 5 thermal iterations per each CMFD power iteration,
- 10 Krylov iterations in the BiCGStab routine that solves the one-group CMFD equation.

This test represents the *nominal scalability* because the work amount is the same in all configurations. Indeed, the energy groups, the space and directions are equally swept the same number of times. The interest of this test is to analyze the impact of the communications on the speed up and the impact of the initialization phase.

By definition, the speed up is the ratio

$$S_n = \frac{T_{E,1}}{T_{E,n}},$$

where $T_{E,1}$ is the time elapsed using a single processor, while $T_{E,n}$ is the runtime with n processors. As usual, the parallel efficiency is defined as $E_n = \frac{S_n}{n}$. Since the serial calculation on a single processor has not been performed, the speed up is measured by the ratio

$$S_n = \frac{8T_{E,8}}{T_{E,n}},$$

that takes as the reference the elapsed time of configuration $2 \times 2 \times 2$. Figure (8) shows the ideal speed-up line (in light blue) compared to the measured speed up (the gamboge line). The total speed up has been decomposed in partial speed up for the DDM iterations, i.e. transport and CMFD, and the transport alone. The simulations show an elapsed time running from 2860 seconds, with 8 processors, down to 43 seconds with 1000 processors. The global speed-up have a maximum of 58% of the ideal speed up 1000 processors. Analyzing the partial speed-up relative to the DDM solution algorithm, represented by the green line in Fig. (8), the IDT solver shows almost ideal scalability. In this part, the deviation from the ideal speed-up gives the idea of how communications affect the solution time. The communications comprise the flux exchanged among transport subdomains and the flux exchanged among the CMFD subdomains. Results show a maximum difference of 10% from the ideal speed-up lost in flux exchanges. This result is confirmed also by Fig. (9) that shows the amount of the total time spent in exchanging the flux among the subdomains. As espected, the trend of Fig. (9) suggests that the time spent in flux exchange is proportionally affected by the number of nodes and, thus, by the number of calls to MPI send/receive directives.

As shown in Fig. (8), the global scalability dumps to 58% of the ideal speed up: such result is highly affected by the low scalability performances of the initialization part. Figure (10) plots the time spent in the initialization (blue line) and the time spent for solving (the gamboge line). Because of the excellent scalability of the solution algorithm, the weight of the initialization part grows dramatically: it is emblematic the run on 1000 processors, where 118 seconds have been spent for initializing, while only 21 seconds for solving.

The poor scalability performances of the initialization phase is mainly caused by two reasons: the growing number of HCC matrices, that grows with the number of generating subdomains (GCU),

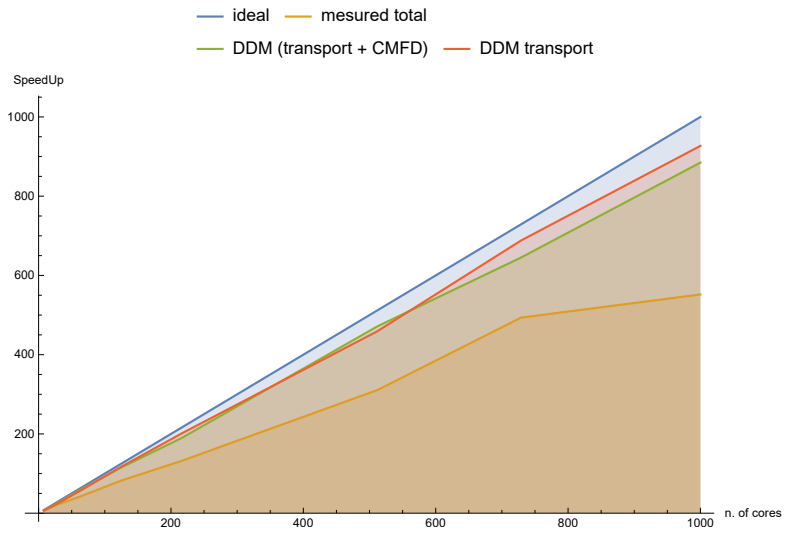

Figure 8: Speed-up from 8 to 1000 subdomains = cores

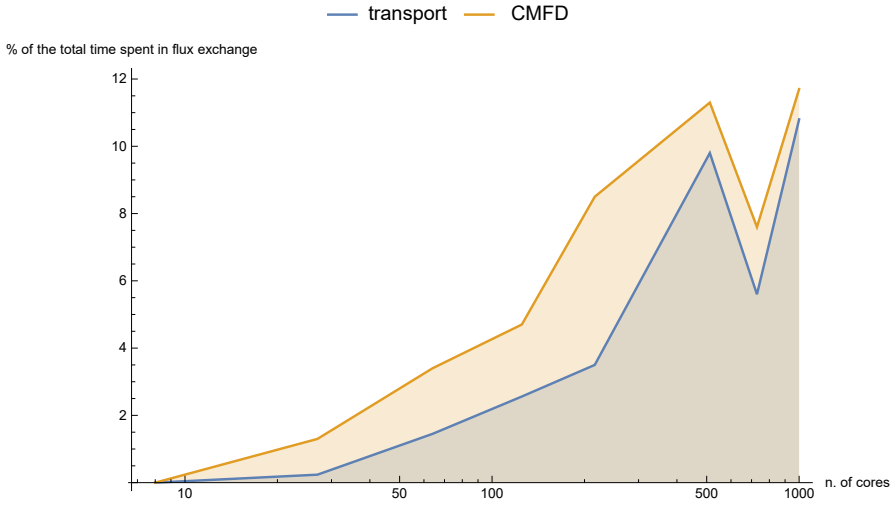

Figure 9: Per-cent of the solution time spent in flux exchange among subdomains. The two lines represent the amount of transport and CMFD communications.

and the shared-memory parallelism implemented in this phase. Moreover, all these issues are amplified by the (simple) nature of the problem. We will try to analyze the above mentioned issues by comparing the $2 \times 2 \times 2$ configuration to the $10 \times 10 \times 10$ configuration. The initial point is that only 10 HCC types are necessary to construct the global C5-G7 geometry. This correspond to 10 HCC sets of coefficients if the problem was solved on a single processor. In the $2 \times 2 \times 2$ subdomain grid, there are 8 GCU and 8 ECU. In this case, because of repetition, the total number of HCC types is 50. This number grows to 613 HCCs for 1000 subdomains. In this case, 200 GCUs are defined for generating 1000 ECUs. As en example of such overhead affecting the $10 \times 10 \times 10$ configuration, the HCC representing the UOX pin cell is computed 75 times! Of course, this extra cost is particularly important in benchmark problem, as the one analyzed, but it is irrelevant when all spatial meshes contains different materials, as for depletion and multi-physics applications.

The second reason is merely linked to the OpenMP parallelism used in this calculation phase: the shared-memory parallelism is applied on the discrete-ordinates for the calculation of coefficients. Instead, the subdomains, are computed sequentially on the node. Each thread is in charge of the computation of HCC coefficients along one direction. Because of the symmetry, the coefficients are computed only for the directions of the first octant. In our particular test suite, there are only 10 directions per octant. This means that the $2 \times 2 \times 2$, where 4 threads/node are allocated, uses all the available resources while initializing. Instead, in the $10 \times 10 \times 10$ configuration, only 10 of the 25 available threads are used . Of course, such an issue fades away when the number of discrete directions is greater or equal to the number of threads.

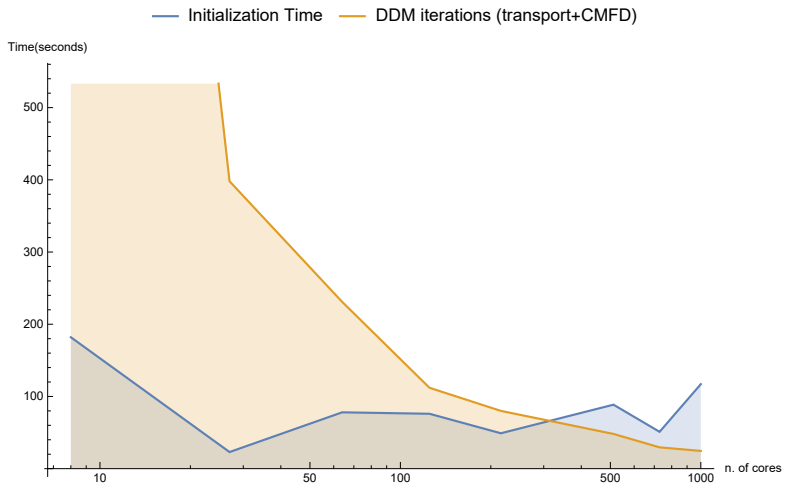

Figure 10: Time elapsed in the DDM iterations (transport + CMFD) and the time elapsed in the initialization.

9.2. Effective scalability

In this test, the configurations of Tab. (4) have been run until the convergence. The error tolerance is 1 pcm on the eigenvalue, 10 pcm on the fission integral distribution and 100 pcm on the interface flux exchanged among the subdomain boundaries. This test is of particular interest to mesure the

effective speed up for a given problem solution.

In Figures (11) and (12) are respectively plotted the elapsed time and the number of outer power iterations versus the number of subdomains. As expected from the theory, the configuration of the domain decomposition determines the rate of convergence of the iterative scheme. In fact, for a given problem, the more the domain is subdivided, the more iterations will be needed to convergence. Also, the synergy among transport and CMFD operators is also affected by the way the domain is decomposed. Because the 'propagation' of the transport solution is slowed down by the increasing number of subdivisions, also the 'quality' of the CMFD operator will be affected. Moreover, because the CMFD is solved with the same spatial decomposition of the transport, also the number of CMFD power iterations increases with the number of subdivisions.

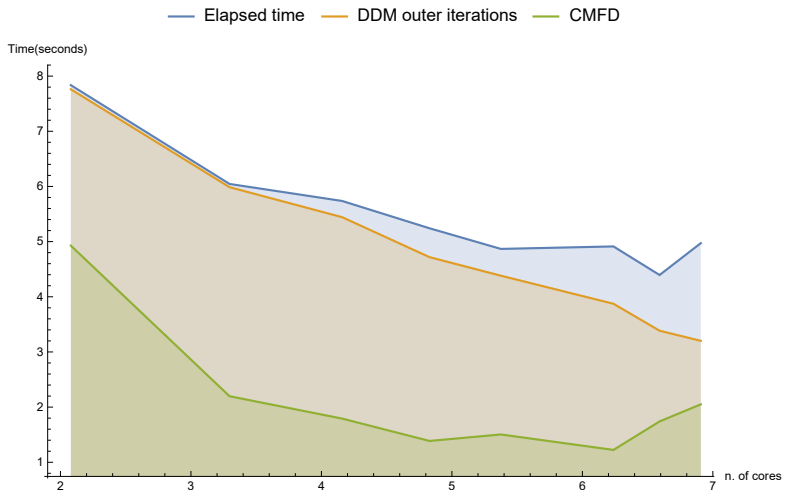

Figure 11: Elapsed time versus the number of subdomains (= n. of cores).

When the number of subdomains increases, the fraction of time spent in the CMFD grows substantially because of the growing of the number of iterations needed to converge the CMFD problem. For an insight of the CMFD time consumption, Fig. (13) shows the ratio of the elapsed time of the CMFD with respect to the total runtime and with respect to the time spent in transport iterations, while Figs. (14) and (15) respectively depict the amount of CMFD outer and inner iterations versus the number of subdomains. The CMFD time is less then 10% of the total time until the 300 subdomains threshold is reached, as illustrated in Fig. (13). Then, the cost of the CMFD solution becomes predominant, reaching about the 60÷70% of the total time. Such increasing in time is produced by the huge amount of inner and outer CMFD iterations necessary to converge those configurations. Such particular behavior of the CMFD has also been observed in the reactor solver MPACT, [**?**].

Figure (16) depicts the global parallel efficiency (blue line), the DDM-iteration parallel efficiency (gamboge line) and the transport iteration parallel efficiency (green line). The overall results show a minimum parallel efficiency of 54%. The results, although affected by the poor scalability of the initialization step, show a great parallel efficiency for the solution step, which demonstrates that the solver is effective. Indeed, the parallel efficiency of the solution part stands at around 80%÷90%.

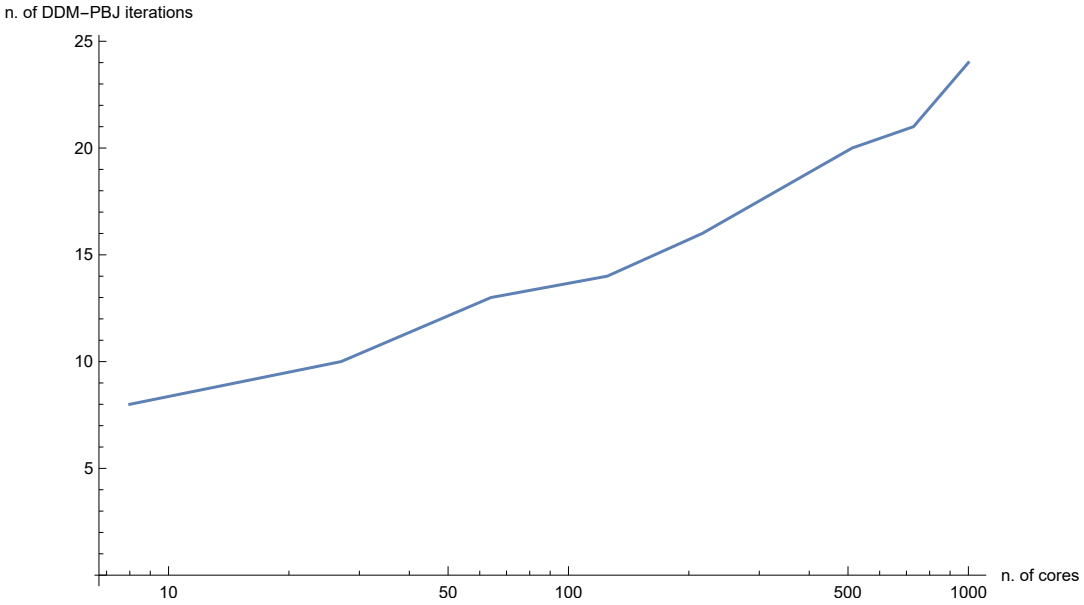

Figure 12: Number of DDM Parallel-Block Jacobi outer iterations versus the number of subdomains = n. of cores.

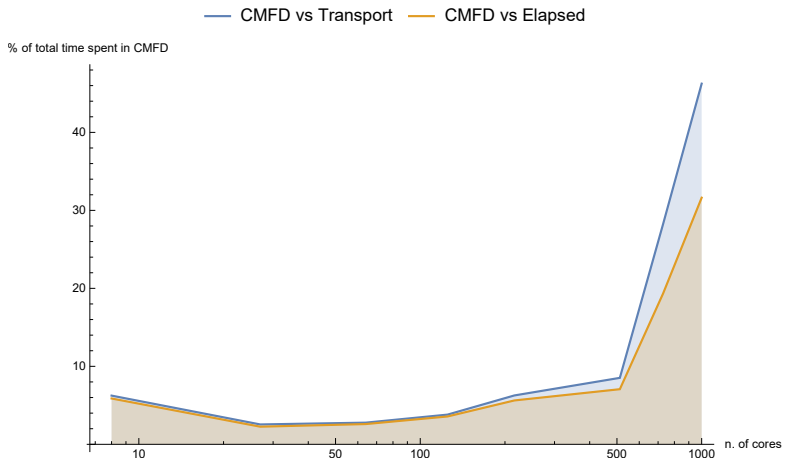

Figure 13: Fraction of the total time spent in the CMFD iterator: with respect to the total elapsed time, with respect to the transport time.

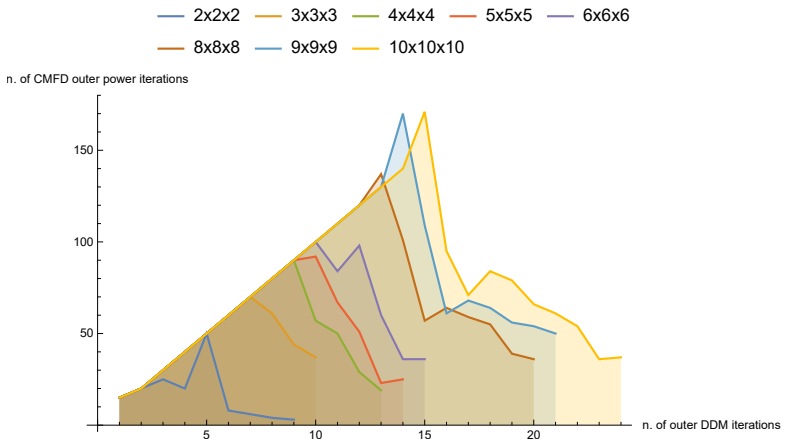

Figure 14: Outer power iterations of the CMFD operator done per each outer power iteration of the DDM transport operator.

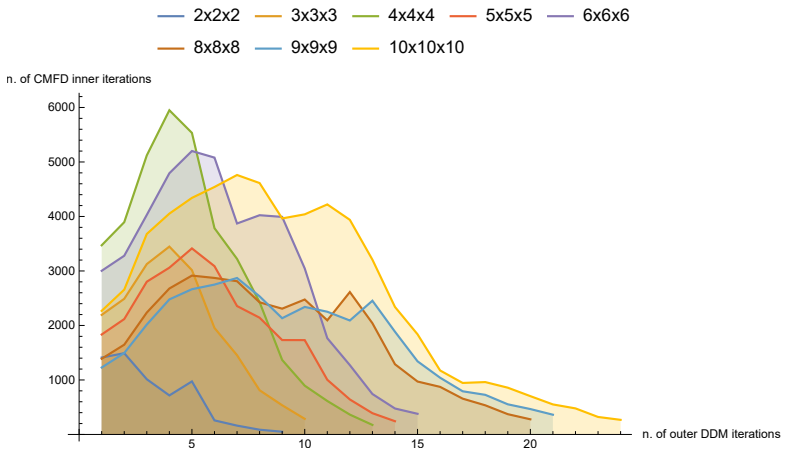

Figure 15: Number of inner Krylov iterations of the CMFD solver per each outer power iteration of the DDM transport operator.

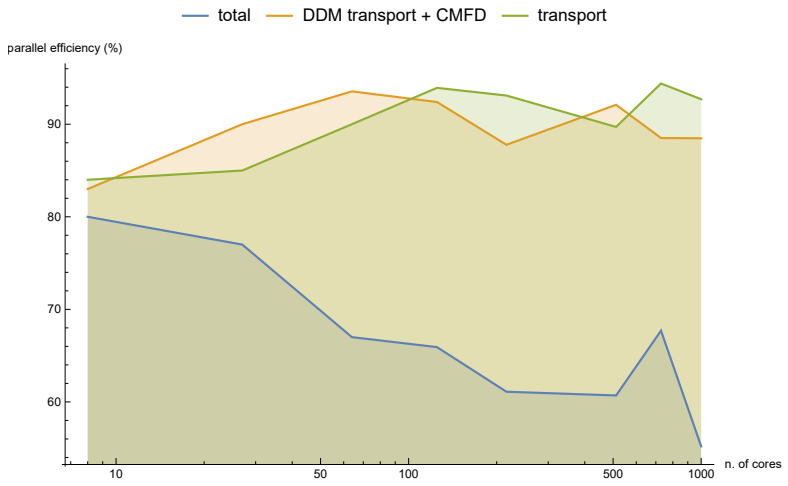

Figure 16: Parallel efficiency.

10. Whole-core simulation: the Eole reactor simulation on 6250 processors

In this test, the IDT DDM algorithm is benchmarked with the Eole reactor simulation. The APOLLO3 data file used for generating the IDT input file and cross sections is derived from the one used in presented in [29], in particular here, the nominal configuration with half of the total geometry is analyzed. This calculation is actually 4 times bigger than the former APOLLO3 simulation performed in reference [29]. Except for the surrounding reflector, the geometry of the reactor is modeled exactly. The boundary conditions are zero incoming boundary flux everywhere except for the bottom surface which has a reflective boundary conditions simulating the bottom half of the reactor. The surface meshing of the HCCs is a uniform 2×2 grid on each face of the boxes. The angular discretization is performed using a S12 level-symmetric quadrature. The CMFD mesh consists in 241 875 regions obtained by homogenizing each HCC.

The test has been run using 6250 processors for 6250 subdomains. The resources allocated on the Cobalt cluster are 250 nodes with 25 cores/node. The spatial subdomain grid, that composes the *pivot* geometry, comprises $25 \times 25 \times 10$ zones. The size of the subdomains is $3 \times 3 \times 5$ HCCs/subdomain for the first 1875 *pivot* zones and $3 \times 3 \times 4$ HCCs/subdomain for the remaining 4375 zones. Although the reactor configuration comprises only 100 materials, the test has been performed by duplicating the cross sections of the fuel materials in each HCC containing fuel. This makes the number of materials grow up to 158 647. Table (5) shows the discretization options and the overall size of the problem. The total memory occupation is 10 TB, the 56% of which is occupied by transport coefficients. The IDT solver has 3 strategies to handle coefficients: 1/ the coefficients are computed in the initialization step and stored in memory once and for all, 2/ they are computed on-the-fly for each group before the angular-spatial transport sweep, 3/ they are compressed and reconstructed on-the-fly by using a linear Taylor expansion around well-chosen values of the total cross-sections. In this particular test, we have privileged the performances at the expense of memory cost by choosing the first option.

The 35% of memory is used for storing the flux. This memory comprises the multigroup angular moments, the interface boundary angular flux, and the fission integral. But the huge part of the flux storage is occupied by the interface boundary angular flux which is exchanged among the subdomains.

In any event, as shown in Tab. (6), the average memory/node is 40 GB against 128 GB available, which means about only the 33% of the total available memory. The deviation to the mean value is about 5 GB. The detailed data decomposition among the memories of the nodes is depicted in Fig. (17). The figure shows a maximum memory occupation of 63 GB/node, which is less of the half of the memory of the node.

Table 7 shows the performances of the run. The overall elapsed time is satisfactory: the calculation runs in 47 minutes with only 15% of the time spent in the initialization, that involves the calulation of the HCC's matrices. Furthermore, considering that 1/4 of the same calculation took 5 days on a single node with 16 processors without DDM (as it has been shown in the work of [29]), one can roughly estimate a speed-up of 153.

The analysis of the elapsed times shows a great performance of the transport solver that uses only 7 minutes, i.e. the 11% of the total time. Instead, as predicted by the scalability test, the 71% of the runtime is spent by the CMFD. More precisely, just the 6% is used for CMFD iterations while the 65% of the total time is used to iteratively update the CMFD coefficients. The most expensive step in the CMFD is the construction of the broad cross sections, in particular, the scattering matrix where the 281-group P3 matrices are homogenized and collapsed to 24 groups in a P0 matrix in each coarse cell.

Another positive result is also the reduced time spent in communication which is only 16 seconds, that is negligible.

In Table 7, the total number of transport iterations could seems huge at a first glance, but if one divides by the number of groups and by the number of outer DDM iterations, than the average number of transport sweep in a subdomain is more reasonably $3 \div 4$ iterations per group.

The Figures (18) and (19) show the time distribution per subdomain and the time distribution per threads. The average time spent in a subdomain is 500 seconds, that is almost equal to the average time spent by a thread, which is 471 seconds. But the deviation from the average is 500 seconds for the time per subdomain and 300 seconds for the time per thread. Such spread of the time distribution is caused by two main reasons: the memory imprint of the subdomains is not the same and, above all, the spectral radius of the local multigroup iterations varies from subdomain to subdomain. This last aspect, which is deeply influenced by the material composition of the subdomain, has a consequence on the number of iterations and, thus, on the computational cost of the subdomain. In conclusion, subdomains having the highest spectral radius (as for example, those in the water reflector) consume the most of the computational time. This means that the domain decomposition layout affects the iterative properties of the calculation, as it has been shown in the strong scalability test.

Nevertheless, as depicted in Fig. 20, the CPU time cumulated per node shows a good balance.

Table 5: Problem size of the EOLE-UH1.2 simulation with 6250 subdomains = 6250 cores.

Pivot grid	$25 \times 25 \times 10$
Number of Subdomains	6250
Anisotropy order	P3
Number of regions	648 434
Number of outer boundary surface meshes	24 150
Number of interface boundary surface meshes (DDM)	1 095 000
Number of groups	281
Number of HCC types	78 344
Number of outer boundary surface meshes	216 000
Number of directions per octant	21
Total number of directions	128
Number of spatial moments of the volume flux	4
Number of spatial moment for the interface flux	3
Number of angular moments for the source	16
Number of materials	158 647
Number of CMFD coarse cells	241 875
Number of CMFD coarse energy groups	24
Memory for coefficients (GB)	5 632
Memory for cross sections (GB)	395
Memory for flux	3 548
Total Memory (GB)	9 881

Table 6: Allocated resources and memory occupation per node.

Number of Nodes	250
Number of cores / node	25
Total number of cores	6 250
Number of Subdomains	6 250
Average memory per node (GB)	40
σ(Memory per node) (GB)	5
Max. memory in a node (GB)	63
Min. memory in a node (GB)	36

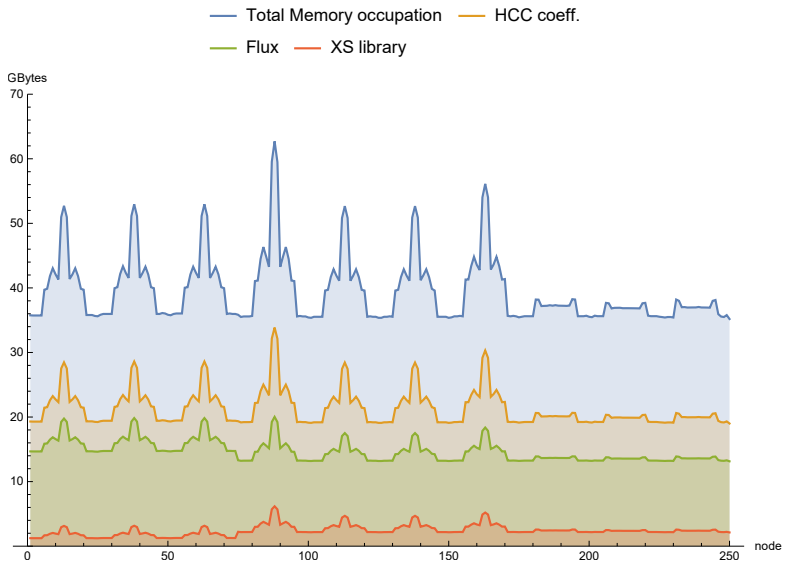

Figure 17: Memory occupation per node.

Table 7: Iterations and time analysis of the Eole reactor simulation. Errors on the k-effective and on the fission distribution are given with respect to pin-by-pin experimental measures.

Eigenvalue	1.00434
eigenvalue error (pcm)	66
max. fission source error (%)	2.1
RMS (%)	0.56
Total DDM outer iterations	40
Inner transport it./subdomain	36105
CMFD Outer iterations	12105
CMFD Inner it./subdomain	46316
Total Elapsed Time (seconds)	**2894**
Initialization step (seconds)	**433**
DDM construction (seconds)	**2**
DDM outer loop (seconds)	**2458**
Total in transport Subdomain loop (seconds) CMFD solver (seconds) CMFD Coef. (seconds) total in fluxes exchange (seconds)	341 213 1 889 16

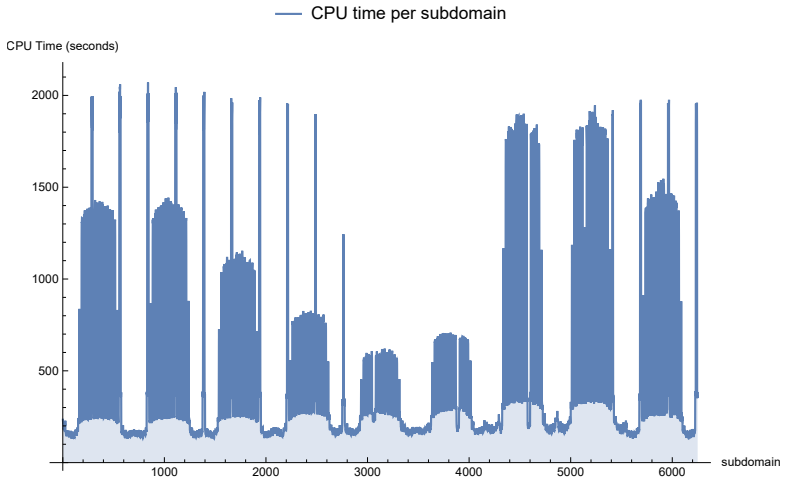

Figure 18: Distribution of the CPU time per subdomain.

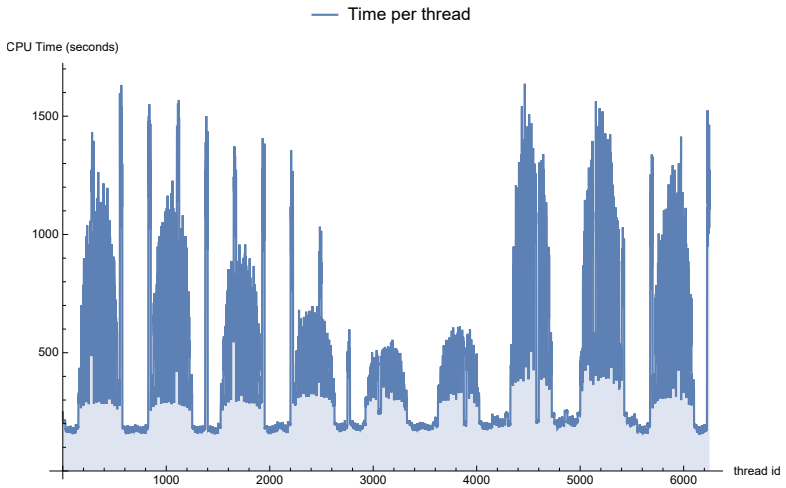

Figure 19: Distribution of the CPU time per threads.

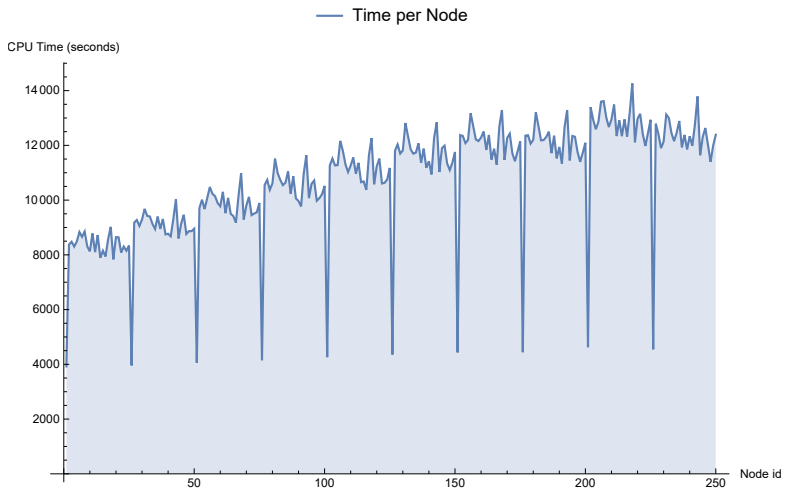

Figure 20: Cumulative CPU time per node.

11. Conclusions and further remarks

In this paper, we presented a numerical archetype for 3D whole-core transport simulations. The HCC provides an accurate representation of the flux gradients within the pin cell. The numerical tests show that high-fidelity transport calculations in regular assembly geometries can be efficiently obtained without need for homogenization. The LSC applied to HCC allows for a considerable reduction of the number of regions without a relevant loss in accuracy. The HCC can provide a consistent alternative to the most popular MOC-based solvers, as in [**?**] and [**?**]. The tests shown in this paper demonstrate the capabilities to preform HPC simulations up to o(1000) processors of a reactor core using the HCC discretization. Furthermore, the DDM multigroup iterative algorithm has shown good scalability performances reaching 60% of global efficiency up to 6250 processors. Furthermore, the simulation of the Eole reactor shows accurate results on a 3D full-core in a reasonable amount of time. Indeed, those tests also display that a depletion step of a reactor can be obtained in less than 1 hour, thus, one can foresee full-core depletion calculation in about 1 day of calculation on a HPC machine.

Considering that this work is a first step for porting IDT on HPC machines, we can argue that IDT has a margin for improvements:

- The CMFD options, as the number of coarse groups or the size of the coarse cells, are not tuned. The tuning of the CMFD can sensibly affect the acceleration effectiveness and, thus, the calculation time.
- There is an ongoing work for the improving of the CMFD non-linear acceleration: an experimental version of IDT is already equipped with a non-linear acceleration library which solves the CMFD system by the PETSc library. Early comparison among the current CMFD and the CMFD of the library has shown very encouraging results since the latter is 40% faster than the former.

- A refactoring of the HCC coefficient is also in the pipeline, this will guarantee a sensible reduction of the trajectories and, thus, of the computational cost of the initialization step.
- The initialization step also has to be reorganized for a more effective usage of the parallel resources.
- The domain splitting, that especially influences the CMFD, can be improved by implementing other strategies taking into account the spectral radius of the subdomain and the spatial load, i.e. the number of regions in the HCCs.
- Finally, the build-chain used in this work is based on the `gnu/4.8.5` distribution, for favoring the error diagnostic, which is a conservative decision. We have not yet built an executable with the recommended Intel compiler which allows for vectorization, in particular, the AVX2 directives.

This work served as a first real-scale benchmark for validating IDT and its capability of efficiently running on HPC machines.

Acknowledgements

One of the author, E.M., would like to express sincere gratitude to R. Sanchez, for its never-ending support, and Y. S. Ban and Professor H. G. Joo from the Department of Nuclear Engineering of the Seoul National University, for the fruitful and precious collaboration that gives a deep help to the V&V of the IDT solver. Also, authors would like to thank B. Martin and J.-M. Do, for the DDM implementation undertaken during the internship in SERMA. Finally, the authors thank Électricité de France (EDF) for partial financial support.

REFERENCES

[1] Gol'din, V. Ya., 1960. Characteristic Difference Scheme for Non-Stationary Kinetic Equation. *Soviet Mathematics* - Doklady, **1**, 902.

[2] Takeuchi, K., 1969. A Numerical Method for Solving the Neutron Transport Equation in Finite Cylindrical Geometry. *J. Nucl. Sci.*, **6**, 446.

[3] Larsen, E. W., Alcouffe, R.E., (1981)."The Linear Characteristics Method for Spatially Discretizing the Discrete Ordinates Equations in (x ,y)-geometry," *Proc. Int. Topl. Mtg. on Advances in Math. Methods for the Solution of Nucl. Eng. Problems*, Munich, Germany, April 27-29, 1981, American Nuclear Society.

[4] Zmijarevic, I., 1999. Multidimensional Discrete Ordinates Nodal and Characteristics Methods for APOLLO2 Code. *Proc. Mathematics & Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 1999, 705.

[5] Masiello, E., Zmijarevic, I., 2006. Short Characteristics Method for Two Dimensional Heterogeneous Cartesian Cells. *Int. Top. Mtg. Advances in Reactor Physics, Proc. of Int. Conf. PHYSOR'06*, Vancouver, Canada, September 12-15, 2006.

[6] Masiello, E., Sanchez, R., and Zmijarevic, I., 2009. New Numerical Solution with the Method of Short Characteristics for 2-D Heterogeneous Cartesian Cells in the APOLLO2 code: Numerical Analysis and Tests. Nuclear Science and Engineering, 161, 1-22.

[7] Masiello, E., Martin, B., Do, J.-M., 2011. Domain Decomposition and CMFD Acceleration Technique applied to Neutron Discrete-Ordinates Transport Equation in XYZ Geometries. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)* Rio de Janeiro, RJ, Brazil, May 8-12, 2011, on CD-ROM, Latin American Section (LAS) / American Nuclear Society (ANS), ISBN 978-85-63688-00-2

[8] Masiello, E., 2008. Analytical Stability Analysis of the Coarse-Mesh Finite Difference Method. *International Conference on the Physics of Reactors 'Nuclear Power: A Sustainable Resource'* , Proc. of Int. Conf. PHYSOR'08 , September 14-19.

[9] Masiello, E., 2018 " On-the-fly stabilization of the Coarse-Mesh Finite Difference acceleration for multidimensional discrete-ordinates transport calculations", *Journal of Comp. Physics,* Volume ***373***, , Pages 1-27.

[10] Lenain, R., 2015. Amélioration des méthodes de calcul de cœurs de réacteurs nucléaires dans APOLLO3: décomposition de domaine en théorie du transport pour des géométries 2D et 3D avec une accélération non linéaire par la diffusion. Physique Numérique [physics.comp-ph]. *Université Paris Sud-Paris XI*, 2015. Français. https://tel.archives-ouvertes.fr/tel-01224873

[11] Lenain, R., Masiello, E., Damian, F., Sanchez, R., 2013. A Parallel Full Core Transport Calculation Based On Domain Decomposition Method. *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA+MC2013)*, La Cité des Sciences et de l'Industrie, Paris, France, October 27–31.

[12] Lenain, R., Masiello, E., Damian, F., Sanchez, R., 2015. Domain Decomposition Method For 2D And 3D Transport Calculations Using Hybrid MPI/OpenMP Parallelism. *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 .

[13] Ban, Y. S., Masiello, E., Lenain, R., Joo H. G., Sanchez, R., 2018. Code-to-code comparisons on Spatial Solution Capabilities and Performances between nTRACER and the standalone IDT solver of APOLLO3®. *Annals of Nuclear Energy*, Volume **115**, May, pg. 573-594.

[14] Godfrey, A.T., 2014. Vera Core Physics Benchmark Progression Problem Specifications, Revision 3, CASLU2012-0131-003, CASL, March 31.

[15] Lewis, E.E., Smith, M.A., Tsoulfanidis, N. , Blomquist, R.N., 2005. Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation. https://www.oecd-nea.org/science/docs/2005/nsc-doc2005-16.pdf

[16] Lewis, E.E., Smith, M.A., Na, B.C., 2004. "OECD/NEA Benchmark on (MOX-fuelled Core) Transport Calculations without Spatial Homogenisation," Guest Editors, *Progress in Nuclear Energy*, Volume **45**, Issues 2-4.

[17] Schwarz, H., 1869. Uber einige Abbildungsaufgaben. *Ges. Math. Abh.* 11, 65–83, (1869).

[18] Lions, P.-L., 1990. On the Schwarz alternating method III: a variant for nonoverlapping subdomains. *In Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia , pp. 202–223. SIAM.

[19] de Oliveira, C. R. E., Pain, C. C., and Goddard, A. J. H., 1995. Parallel domain decomposition methods for large-scale finite element transport modelling. *In Proc. Int. Conf. Math. Comp. Reactor Physics and Environmental Analysis of Nuclear System*, Portland. Oregon.

[20] Guerin, P., Baudron, A.-M., and Lautard, J.-J., 2007. Domain decomposition methods for core calculations using the MINOS solver. *In Proc. Joint International Topical Meeting on Mathematics & Computation, Supercomputing in Nuclear Applications (M & C + SNA 2007)*, April 15-19 Monterey, CA, U.S.A.

[21] Van Criekingen, S., Nataf, F., Have, P., 2011. PARAFISH: a Parallel FE-PN Neutron Transport Solver based on Domain Decomposition. *Annals of Nuclear Energy*, Vol. 38 (1), pp. 145-150.

[22] Smith, K. S. , Henry, A. F. , Lorentz, R. , 1980. Determination of Homogenized Diffusion Theory Parameter for Coarse-Mesh Nodal Analysis. *Proc. ANS Topl. Mtg. Advanced in Reactor Physics and Shielding*, Sun Valley, Idaho, September 14-19, 1980, 224, ANS.

[23] Smith, K. S. , 1983. Nodal Method Storage Reduction by Non-Linear Iteration, *Trans. Am. Nucl. Soc.*, 44, 265.

[24] Aragones, J. M. , Ahnert, C. , 1986. A Linear Discontinuous Finite Difference Formulation for Synthetic Coarse-Mesh Few-Group Diffusion Calculations. *Nucl. Sci. Eng.* 94, 309-324.

[25] Anistratov, D. Y., 2006. Nonlinear Quasidiffusion Acceleration Methods with Independent Discretization. *Trans. Am. Nucl. Soc.*, 95, 553-555.

[26] Saad, Y., 2001*Iterative Methods for Sparse Linear Systems*, Dover Publication, New York.

[27] Smith, M.A., E.E. Lewis, Na, B.-C., 2006. Benchmark on deterministic 3-D MOX fuel assembly transport calculations without spatial homogenization. *Progress in Nuclear Energy,* Volume **48**, Issue 5, July 2006, Pages 383-393·

[28] http://www-hpc.cea.fr/fr/complexe/tgcc.htm

[29] Palau, J.-M., et al., 2018. 3D Analysis of the UH1.2 Mock-up Experiment Using the APOLLO3 IDT Method and Comparisons Against Tripoli4 Monte Carlo Reference Calculations. *Proc. of Int. Conf. Top. Mtg. Advances in Reactor Physics*, PHYSOR'18, Cancun April 23rd.