

A generalization of 3D prismatic characteristics along a nonuniform projection mesh

LAJOIE Marc-André^{1,2}, FÉVOTTE François¹, and MARLEAU Guy²

¹EDF R&D

²École Polytechnique de Montréal

Large scale tri-dimensional transport calculations in unstructured meshes are limited in scale mostly by the amount of data to process (memory space limitations), the number of operations to perform (computational time limitations), or both. A method relying on a prismatic projection of non-uniform 3-D geometries is proposed, that reduces both memory space and computation time without introducing new approximations and opens new possibilities regarding large scale parallel processing.

KEYWORDS: Method of Characteristics; Prismatic geometries; 3D tracking; 2D projection; DRAGON

I. Introduction

Recent developments in computer power and advances in parallel processing have made reference full-core neutron transport calculations likely to be achievable within the next few years. Even though concerns remain regarding the optimization of the energy mesh or the generation of nuclear data (see Sanchez⁽¹⁾), such a reference calculation scheme would still provide advantages related to the validation and improvement of industry models and the analysis of complex coupling phenomenon between neutronics and thermal-hydraulics.

The main hurdles remaining in that regard concern both the quantity of information to manage and the number of computing operations to perform. Based on preceding work by LeTellier,⁽²⁾ and introducing a generalized prismatic projection of the 3-D geometry, we propose a method that purports to help solve both these problems. The Method of Characteristics (MoC)⁽³⁾ implanted in the lattice code DRAGON,⁽⁴⁾ will serve as a basis for this study.⁽⁵⁾

Similar ongoing efforts can be found in the literature and range from a 2-D/1-D flux decomposition,⁽⁶⁾ leading to an alternative MoC iterative algorithm, to unstructured meshes using straight or curved triangular prisms solved using discontinuous Galerkin finite elements.⁽⁷⁾ The algorithm proposed here presents the advantage of relying only on an alternative but exact description of the geometry, introducing no new physical approximations, to ultimately provide a solution to the regular tri-dimensional transport equation using proven and reliable legacy solvers.

Here, we propose the treatment of a full tri-dimensional geometry (relying on certain assumptions met by most nuclear reactors) using a projected representation for tracking,

combined with an extrusion process to reconstruct the tri-dimensional tracks upon flux computation, with no new approximations introduced. Preliminary results show a reduction of memory requirements by as much as 4 orders of magnitude, and a reduced computational burden in some situations. However, these results are highly dependent on the original geometry, the best results (in comparison to 3-D solvers) being observed when the geometry is relatively uniform axially, and is not too extended in the direction of projection (relatively small height to width ratio).

This new geometry processing capabilities also open new axes of parallelization for the flux solver. In analyzing previous works,⁽⁸⁾ we can expect good speedups, which, when combined with the reduced memory requirements, can provide an ideal solver for large-scale parallel deterministic 3-D transport in unstructured meshes.

This paper first presents a short review of the method and formalism used, followed by an illustration and explanation of the aforementioned modified geometry description. Two test-cases are then exposed to illustrate both the new capabilities and performances of such a solver. Finally, the short conclusion presents expected developments and new possibilities offered by this projection technique as related to parallel processing.

II. MoC transport

1. Method of Characteristics

Using the formalism presented in Hébert,⁽⁹⁾ the Boltzmann transport equation can be written as :

$$\hat{\Omega} \cdot \vec{\nabla} \phi(\vec{r}, E, \hat{\Omega}) + \Sigma(\vec{r}, E) \phi(\vec{r}, E, \hat{\Omega}) = Q(\vec{r}, E, \hat{\Omega}) \quad (1)$$

where ϕ is the neutron flux at point \vec{r} in direction $\hat{\Omega}$ with energy E , Σ is the macroscopic total cross section and Q is the neutron source that includes both fissions and diffusions. This equation can be solved using a number of methods, both deterministic and stochastic.

Here, we will consider a deterministic solution to the transport equation based on the method of characteristics. This technique relies on a transformation of eq. (1) into a series of ordinary differential equations (ODE) along tracks \vec{T} spanning the whole geometry. This process is known as tracking. As a result, the previous equation along an arbitrary track in direction $\hat{\Omega}$ can be written, using s as an index representing displacement along the track, as :

$$\frac{d}{ds}\phi(\vec{r} + s\hat{\Omega}, \hat{\Omega}) + \Sigma(\vec{r} + s\hat{\Omega})\phi(\vec{r} + s\hat{\Omega}, \hat{\Omega}) = Q(\vec{r} + s\hat{\Omega}) \quad (2)$$

The tracking operator is then defined in such a way as to be consistent with an angular quadrature that spans 4π for a 3-D calculation. Then, a plane $\Pi_{\hat{\Omega}}$ is defined for every direction $\hat{\Omega}$, upon which a surfacic density is used to generate the starting points of individual tracks \vec{T} . Thus, a global number of tracks are produced, along with their integration weights, in effect providing a 4-D numerical quadrature.

The geometry is then discretized into finite volumes in such a way as to ensure that the MoC assumptions on the spatial behaviour of the sources are met. Along each track, intersections with the discretized geometry are located, and both the chord lengths and medium indices are stored for later use within the flux solver.

Using a *step characteristic* (SC) integration scheme, whereby the source term Q is assumed constant over a discretized spatial region, we can solve analytically eq. (2) along every segment, to yield, using the optical path $\tau_g(s) = \int_0^s \Sigma_g(\vec{r} - s'\hat{\Omega}) ds'$, the following propagation relations :

$$\phi_{k+1}(\vec{T}) = e^{-\tau_k(\vec{T})}\phi_k(\vec{T}) + \frac{Q_k(\hat{\Omega})}{\Sigma_k} [1 - e^{-\tau_k(\vec{T})}] \quad (3)$$

$$\bar{\phi}_k(\vec{T}) = \frac{\phi_k(\vec{T})}{\tau_k(\vec{T})} [1 - e^{-\tau_k(\vec{T})}] + \frac{Q_k(\hat{\Omega})}{\Sigma_k} \left[1 - \frac{1 - e^{-\tau_k(\vec{T})}}{\tau_k(\vec{T})} \right] \quad (4)$$

where $\bar{\phi}_k$ is the average flux on a segment k , and ϕ_k is the flux at the intersection of segments k and $k-1$. The angular flux within each spatial region can then be obtained through an iterative process.

2. Tracking and normalization

Using the MoC formalism, we can therefore transform volume-integrated quantities into integrals over the tracking. The tracking operator mentioned previously can be expressed, for an

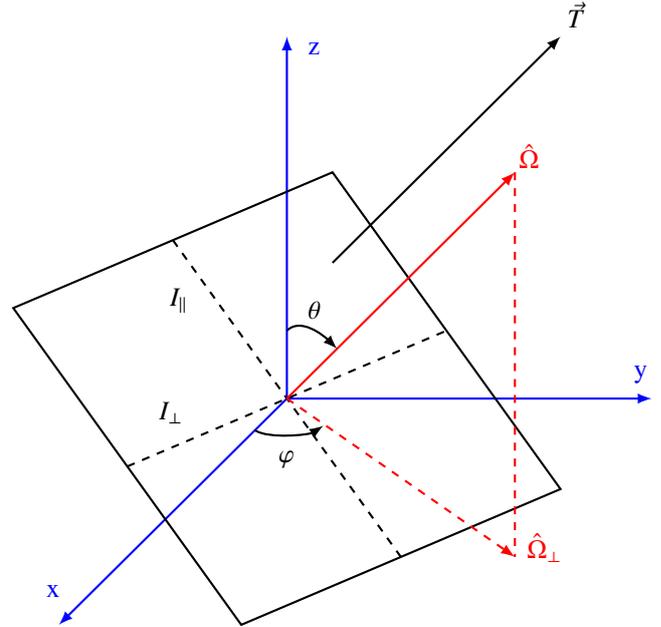


Figure 1: Decoupling of the azimuthal and polar dependencies in φ and θ ¹

arbitrary function f defined over a region R_i , as:

$$\begin{aligned} F_i(\hat{\Omega}) &= \int_{R_i} f(\vec{r}, \hat{\Omega}) d^3 r \\ &= \int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_i(s, \vec{T}) f(\vec{T}) ds d^2 p \end{aligned} \quad (5)$$

where we defined the characteristic function $\mathcal{W}_i(s, \vec{T})$ worth 1 if point s of characteristic \vec{T} is within region R_i , or 0 otherwise. In particular, these equations can be used both to obtain the angular flux ($f(\vec{r}, \hat{\Omega}) = \phi(\vec{r}, \hat{\Omega})$), or, using $f(\vec{r}, \hat{\Omega}) \equiv 1$, to provide an angular-dependent approximation to the regional volumes. Scalar quantities such as the integrated flux can then be computed as:

$$\mathcal{F}_i = \frac{1}{4\pi} \int_{4\pi} F_i(\hat{\Omega}) d^2 \Omega \quad (6)$$

These numerical volumes are important when compared to actual volumes of the region as a way to improve the quality of the results, in partially correcting the numerical error introduced by the tracking quadrature. In practice, the track lengths are modified using the numerical-to-analytical volumes ratio – that should be as close to 1 as possible.

III. Generalized prismatic projection

1. Projection

Since most nuclear reactors present a form of invariance along a certain axis for some distance (we use z in this instance, but any other reference direction can be considered assuming a simple rotation), it is possible to simplify the tracking procedure presented in the last section. The angular component of the

¹Figured inspired by LeTellier et al.⁽²⁾

tracking operator in equation (6) can be decomposed, without loss of precision, into azimuthal (φ) and polar ($\mu = \cos(\theta)$) components, such as:

$$\int_{4\pi} F(\hat{\Omega}) d^2\Omega \equiv \int_0^{2\pi} \int_{-1}^1 F(\varphi, \mu) d\mu d\varphi \quad (7)$$

Using an orthogonal basis $(I_{\perp}, I_{\parallel})_{\hat{\Omega}}$, as illustrated in **Figure 1**, we can then separate the tracking along these two components, where a displacement of p_{\perp} along I_{\perp} remains in the (x, y) plane of the projected geometry, yielding:

$$\begin{aligned} \mathcal{F}_i &= \frac{1}{4\pi} \int_{4\pi} \int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_i(s, \vec{T}) f(\vec{T}) ds d^2p d^2\Omega \\ &= \frac{1}{4\pi} \int_0^{2\pi} \int_{-1}^1 \int_{I_{\perp}} \int_{I_{\parallel}} \int_{-\infty}^{\infty} \mathcal{W}_i(s, \vec{T}) f(\vec{T}) ds dp_{\parallel} dp_{\perp} d\mu d\varphi \\ &= \frac{1}{4\pi} \int_{-1}^1 \int_{I_{\parallel}} \left[\int_0^{2\pi} \int_{I_{\perp}} \int_{-\infty}^{\infty} \mathcal{W}_i(s, \vec{T}) f(\vec{T}) ds dp_{\perp} d\varphi \right] dp_{\parallel} d\mu \end{aligned} \quad (8)$$

In this process, we also need to ensure that the directional quadrature ($d^2\Omega$) is the product of polar and azimuthal quadratures. As a result, the set of projected tracks is akin to a traditional 2-D tracking, associated with specific values of p_{\parallel} and μ , and can therefore be calculated by legacy tracking modules, with minimal modifications.

In order to use this simplified tracking procedure, one must establish a projection transformation (see **Figure 2**), where the global geometry is divided along z in N_z sections that are constant along a portion of the projection axis. The correspondence between the 3-D and 2-D geometries is then ensured – and no loss of information occurs – by creating a mapping matrix associating each 3-D region to one or more 2-D regions, depending on its corresponding position(s) in the original 3-D geometry.

In practice, the procedure starts by a series of tests to ensure that all elements are in fact prismatic, and can be projected in a single plane. The geometric elements are then combined along the whole height of the geometry, and in the same process the numbering of regions and surfaces to be used are created. Finally, the projection matrix is created, ensuring a unique correspondence between projected 2-D elements and specific z -positions. This whole procedure is not trivial except for simple geometries, and for complex cases this rapidly generates very complex 2-D geometries that also imply complex numbering and correspondence processes.

It is also worth mentioning that outer surfaces are also indexed, meaning that for a projected geometry containing N external surfaces and M regions, created from a geometry comprising N_z piecewise invariant axial regions, the dimensions of the matrix have to be $[(M + N + 1) \times (N_z + 2)]$, the two additional rows representing additional elements required to index the top and bottom outer surfaces of the original 3-D geometry. A simple example can be seen in **Table 1**, referring to the combination illustrated in **Figure 3**. Surfaces, indexed

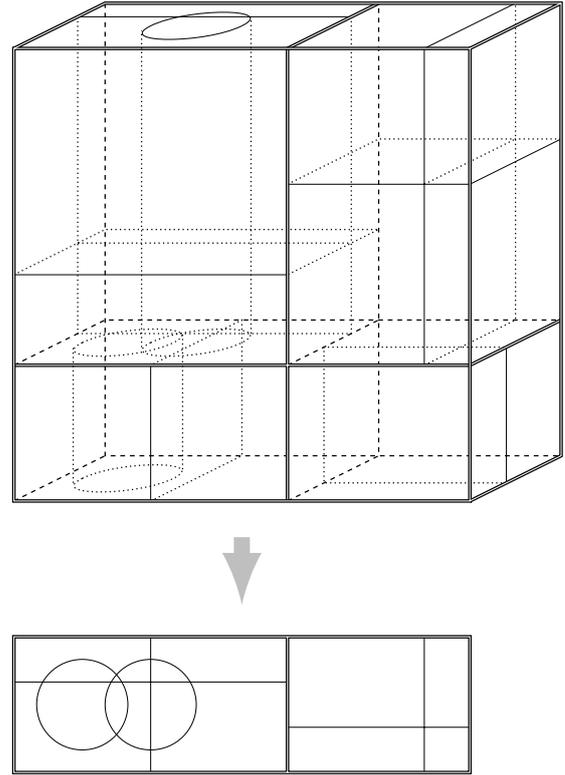


Figure 2: Creation of the 2D projected geometry

using negative indices in DRAGON, are not all indicated to preserve readability.

A traditional azimuthal 2-D tracking can then be performed on this “superdiscretized” geometry, saving disk space (3-4 orders of magnitude, depending on the geometry considered), that can afterward be used in a modified 3D flux solver.

2. Extrusion

As a consequence of this projection, the flux solver will need to “raise,” or extrude, each 2-D segment, using a polar quadrature, and the lengths of the 3-D tracks will be easily obtained as multiples of the polar angle cosine and of the 2-D lengths. The angular flux is then computed as usual along this recomposed tri-dimensional track.

However, the initial projection has produced “virtual” regions in the new geometry – that is, regions that are created when the discretization changes as we move along the projection axis, such as region 9 in **Figure 3**. These regions have no physical signification in themselves, but only in relation to the original geometry. Each of their boundaries therefore represents a boundary that will exist or not depending on the height along the projection axis, translating into either a “real” or a “virtual” frontier.

This also has to be taken into account during the reconstruction process. Thus, reconstructed tracks will need to have segments merged when they cross such a “virtual” boundary. In practice, the reconstruction process relies on the

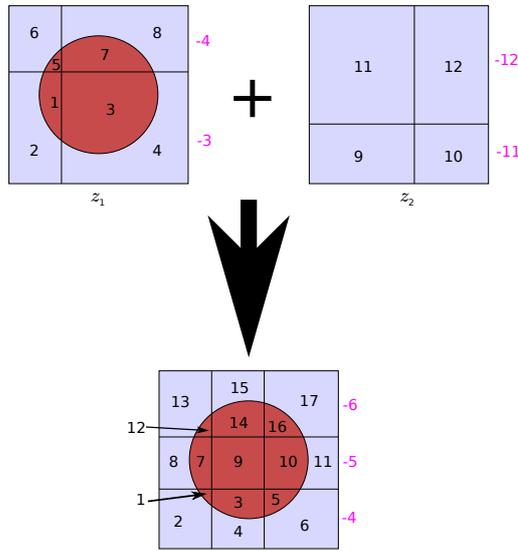


Figure 3: Numbering of the projected geometry

2-D region number	3-D region number	
	z_1	z_2
...
-6	-4	-12
-5	-3	-12
-4	-3	-11
...
1	1	9
2	2	9
3	3	9
4	4	9
...
17	8	12

Table 1: Sample projection matrix

correspondence matrix created in the original projection of the geometry. Every individual 2-D track created in the projected geometry that is raised encounters exclusively a conformal cartesian mesh. This mesh however contains some of these virtual boundaries, that can be easily identified, since the regions on each side of such a boundary are identified within the matrix to the same 3-D region number.

Upon extruding, when such a boundary is encountered, we therefore only have to merge the two track segments, adding their lengths to compute the new optical length, processing them as one, and following the progression according to eq. (4). An illustration is provided in **Figure 4**, in which the black line represents a virtual frontier, and the merged tracks are visible.

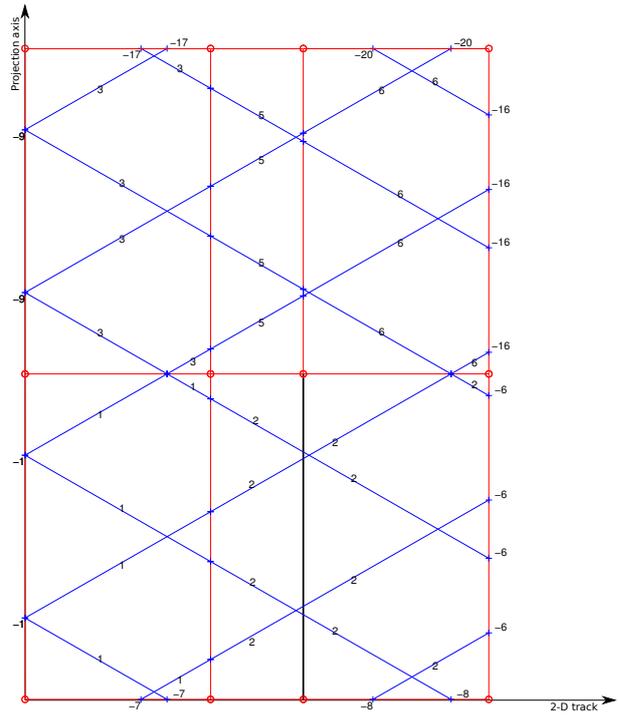


Figure 4: Extrusion of a 2-D track through a virtual boundary

3. Normalization

Finally, as mentioned in section II.2, normalization volumes are used to ensure better numerical precision. However, since in this case the tracking has been performed on the 2-D projected geometry, these ratios are unavailable to normalize individual tracks at that moment.

Moreover, as can be seen in **Figures 2 and 3**, regions in the projected geometry are generally smaller than the regions in the original geometry due to the added virtual frontiers – especially for very small axial variations in the geometry. This therefore might require a high track density to ensure that each 2-D region is adequately taken into account. Normalization using this 2-D tracking would however be more prone to numerical instabilities due to the small quantities involved, in addition to not being ideal since the flux itself is computed on the tri-dimensional tracks.

A preliminary extrusion therefore has to be performed before entering the flux solver, to calculate those values for the 3-D reconstructed regions. This has the additional advantage to provide a first evaluation of the precision of the tracking process, permitted by the comparison of the numerical volumes to the analytical ones.

IV. Applications

Two test cases will be exposed in this section. The first is a simple and small geometry to be analyzed and compared

	3-D	Prism.	Diff.
Group 1			
Integrated flux	2.59513E+01	2.59513E+01	<0.20 pcm
Collision rate	4.47792E+00	4.47797E+00	-1.12 pcm
Absorption rate	2.74911E-02	2.74997E-02	-31.28 pcm
Scat. rate within	3.46734E+00	3.46740E+00	-1.73 pcm
Scat. rate without	9.83081E-01	9.83065E-01	1.63 pcm
Group 2			
Integrated flux	6.98662E+01	6.98672E+01	-1.34 pcm
Collision rate	7.65036E+01	7.65048E+01	-1.56 pcm
Absorption rate	9.72511E-01	9.72504E-01	0.72 pcm
Scat. rate within	7.55212E+01	7.55224E+01	-1.59 pcm
Scat. rate without	9.87214E-03	9.87233E-03	-1.92 pcm
k_{eff}	0.5470143	0.5469787	6.51 pcm
CPU time (tracking + flux)	106s+1633s	1s+181s	10%
Tracking size	350208 Kb	73 Kb	0.02%
$Nb_{iter} (outer)$	5	5	0
$Nb_{tracking_sweeps}$	33	33	0

Table 2: Comparison of 3-D MoC to extruded geometries

to reference solvers using the traditional three-dimensional method of characteristics. The second illustrates the new reach of possible simulations by simulating a deformed, simplified PWR assembly.

1. Pin end

This first test-case is used as the representation of the end of a pin topped by the moderator. We describe a geometry resembling that of **Figure 3**, with an enriched uranium fuel pin and light water moderator surrounding it. This configuration enables the exploitation of the newly developed generalized prismatic capabilities, as well as enabling a valid comparison between this solver and legacy MoC solvers.

As such, the results presented in **Table 2** show an accuracy on par with the traditional 3-D method of characteristics, while providing reduced computation time and memory requirements. The calculation was performed using a 2-D step of 1mm between tracks, and then extruded using the same step along the normal to the polar extrusion angle. This translates to a 100cm^{-2} ($10\text{cm}^{-1} \times 10\text{cm}^{-1}$) density for the regular tracking, along with an equivalent quadrature providing 96 angles per octant in both cases.

As can be seen by these results, the method proves both reliable and robust, and doesn't affect the iterative process, since it only modifies the innermost MoC iterations upon integrating along individual tracks. Also worth noting is the fact that it uses both less CPU time and much less memory than a traditional 3-D computation, while delivering the same level of accuracy.

2. Deformed assembly

In this section, we describe a simplified representation of one PWR assembly composed of 3×3 pins that has been axially

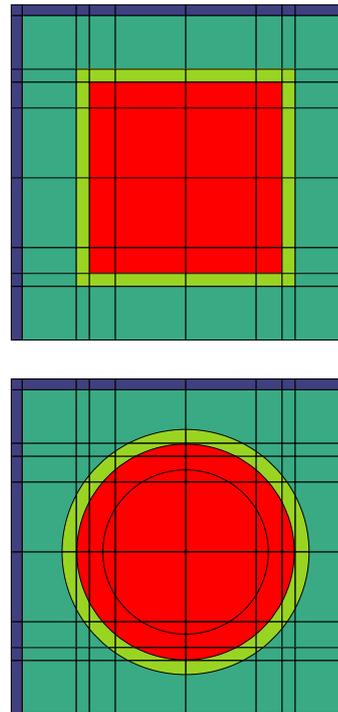


Figure 5: Cell equivalence

deformed. Such a bowed assembly will have a profile similar to that of **Figure 6**.

Since the implementation we developed does not yet provide capabilities regarding multiple cylindrical pin superposition (as illustrated in **Figure 2**) due to numbering incompatibilities of overlapping annular regions, we have used 2-D geometries to provide a transport-transport equivalence between the regular cylindrical pins and their cartesian equivalent, created such as to preserve the volumes of each region. This circumvents the problem by creating an easily treatable non-regular cartesian mesh.

As such, an SPH equivalence procedure⁽¹⁰⁾ has been performed on corner (illustrated in **Figure 5**), side and center cells. The resulting geometric sections of the rectangularized pins can be seen in **Figure 7**, with the combined projected geometry of **Figure 8** showing clearly the very small regions created by such a process – particularly near the cladding of the pins.

As expected, the tracking file resulting from such a projection is much smaller than a regular 3-D equivalent. For this comparison, we used an equivalent quadrature, in this case an azimuthal trapezoidal quadrature combined with a polar Legendre quadrature proposed by Sanchez et al.,⁽¹¹⁾ providing the same final number of tracks in the 3-D geometry and in the extrusion treatment. The size of the storage required for the traditional tracking procedure is over 3Gb, whereas the projected geometry requires only 345Kb, including the additional space to store the 2-D projection and the correspondence matrix.

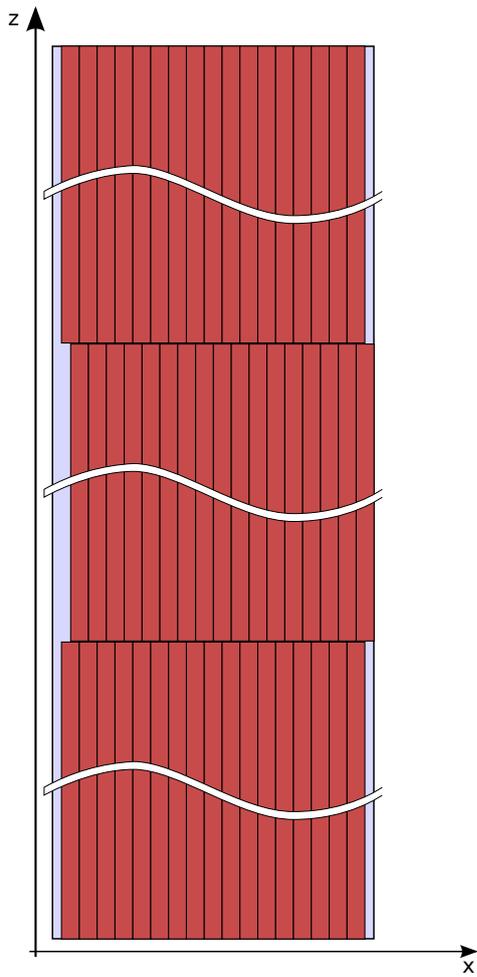


Figure 6: Bowed assembly

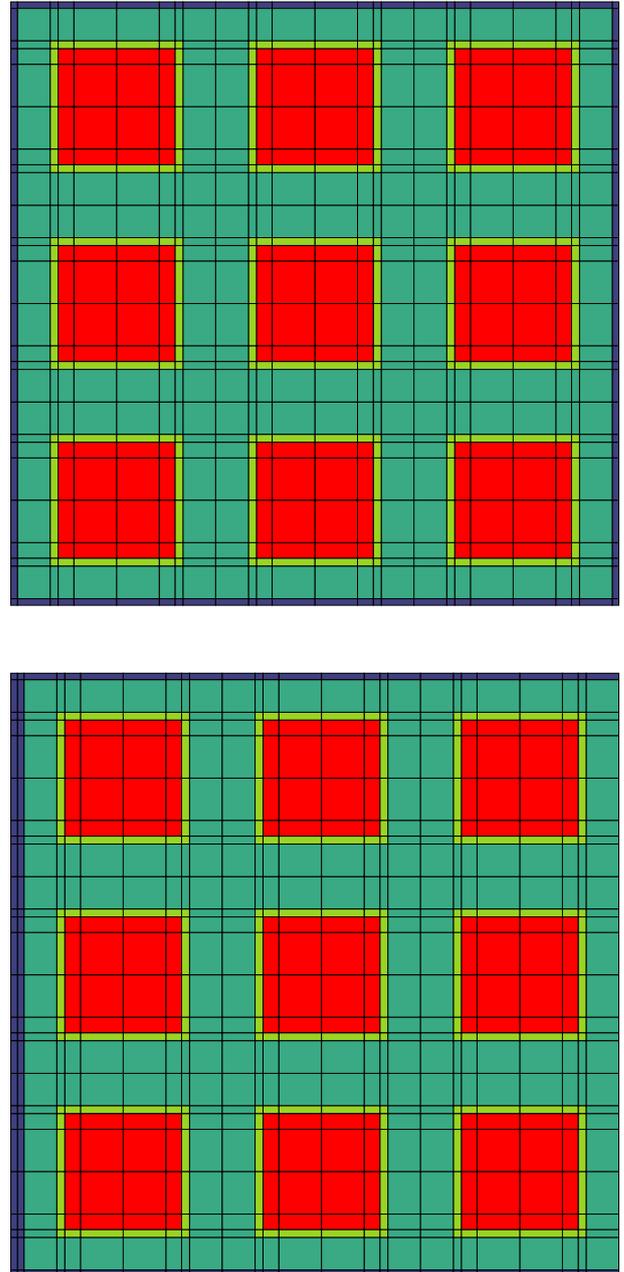


Figure 7: Normal and deformed cartesian cells

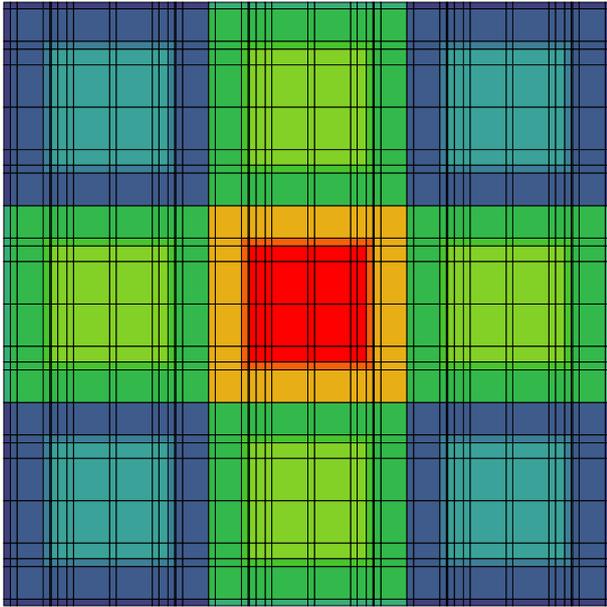


Figure 8: Projected cartesian cells

Direct comparison between the general and projected solutions are however not possible in this case, as the 3-D solver proved to have tracking and convergence issues. This only serves to prove the pertinence of the method, as apparently bigger cases can be expected to be treated using prismatic extrusion.

3. Discussion

Previous efforts⁽²⁾ also show than an improvement in overall computation time is possible using a prismatic algorithm that did not consider axial variations. While we did reduce drastically the time spent in the tracking module for the assembly case (over 30 min to about 4 seconds), the extra computational burden required to extrude the tracks seems to cancel this advantage, since the first iterations of the regular 3-D solver were faster than the prismatic ones.

In particular, we can suppose that since this test geometry is very tall in relation to its base (ten to one), the extrusion overhead cost is expected to diminish as we extend the dimensions of the projected geometry. Moreover, the preconditioning acceleration techniques available in the code are no longer compatible with the track merging process used in the extrusion approach. This could certainly explain the discrepancy observed, and can be expected, once they are adapted, to reduce the overall computation time.

Finally, considering that increasingly computer applications performances are limited by disk accesses, this time-memory trade-off is expected to be beneficial in the long run.

V. Parallelization

Considering lattice calculations, the method of characteristics is already better suited than the collision probability method for parallel computing, being that each segment contributes (relatively) independently to the final flux calculation once it has been generated through the tracking process, minimizing shared data.

In a traditional solver, parallelization may be applied at various levels, including loops over the solid angles $\hat{\Omega}$, over each individual track \vec{T} , and over the multigroup energy mesh. One of the main advantages of the projection method proposed here, is that it provides new avenues for the large scale parallelization of the solver.

In a prismatic configuration, the traditional tracking parallelization levels, which become loops over the azimuthal (2-D) angles of the projected geometry and on each of the computed tracks. To these can be compounded additional axes over the (polar) extrusion angle, and over z (as a vertical shift, used to recreate the original 4-D numerical quadrature). These then open up further parallel processing possibilities, matching the parallel options used in 3-D solvers.

We can then consider the following axes :

- the 2-D φ angles in the projected geometry ;
- each individual 2-D track in the projected geometry ;
- the 3-D $\hat{\Omega}$ directions (related to μ) upon extrusion ;
- each individual 3-D reconstructed track ;
- the energy groups ;

Furthermore, as discussed by LeTellier,⁽⁵⁾ a parallelization can be considered over the algebraic collapsing acceleration operator (ACA), reducing the time required to compute the connexion matrices. This opens up a new possibility when combined with a parallel extrusion process.

VI. Conclusion

This paper proposes a new scheme of transport calculations based on a geometric projection, reducing both memory requirements and, in certain cases, computation time, based on the method of characteristics. Two test-cases were studied, with reductions in the computational burden by as much as 90% were observed (case 1), and in the size of the tracking file for large scale problems by as much as 4 orders of magnitude.

In addition to these advantages, the procedure opens new parallelization axes offering a further broadening of the scope of applicability of transport theory in unstructured meshes to spatially larger problems, ultimately leading to a full core calculation in a non-prohibitive time.

All in all, this procedure greatly reduces the amount of storage required for large scale 3-D problems in comparison with a full 3-D calculation, even when considering the additional memory requirements of the projection matrix. Interesting cases can involve, for example, multiple PHWR misaligned clusters, partially inserted PWR control rods, or grids and reflectors that do not require the same level of discretization as the fuel rods. The larger the scale of the problem, the more advantageous the method, since these are cases that will generally be memory-bound for a transport calculation.

Some limitations do however apply, most notably regarding the range of allowed geometries. As constructed, the method forbids the inclusion of non-prismatic elements – such as pebbles in VHTR reactors – or elements that do not respect the privileged projection direction, such as CANDU control rods.

Acknowledgments

The authors would like to acknowledge the financial support from the Fonds Québécois de la Recherche sur la Nature et les Technologies, Électricité de France, and the Natural Science and Engineering Research Council of Canada, for this project.

References

- 1) R. Sanchez, “Prospects in Deterministic Three-Dimensional Whole-Core Transport Calculations,” *Nuclear Engineering and Technology*, **44**, 2, 113–150 (2012).
- 2) R. Le Tellier, G. Marleau, M. Dahmani, and A. Hébert, “Improvements of the Reactivity Devices Modeling for the Advanced CANDU Reactor,” *Ann. Nucl. Energy*, **35**, 868–876 (2008).
- 3) J. R. Askew, “A Characteristics Formulation of the neutron transport Equation in Complicated Geometries,” Technical Report AEEW-M 1108, Atomic Energy Establishment, Winfrith, United Kingdom (1972).
- 4) G. Marleau, A. Hébert, and R. Roy, “A User Guide for DRAGON Version4,” Technical Report IGE-294, Institut de Génie Nucléaire, École Polytechnique de Montréal (2006), Available freely at <http://www.polymtl.ca/merlin>.
- 5) R. Le Tellier, *Développement de la méthode des caractéristiques pour le calcul de réseau*, PhD thesis, École Polytechnique de Montréal, Montréal, 2006.
- 6) F. Févotte and B. Lathuilière, “MICADO: Parallel Implementation of a 2D–1D Iterative Algorithm for the 3D Neutron Transport Problem in Prismatic Geometries,” *Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, Idaho Falls, 2013, American Nuclear Society.
- 7) J.-Y. Moller and J.-J. Lautard, “MINARET: A deterministic neutron transport solver for nuclear core calculations,” *Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*, American Nuclear Society, 2011.
- 8) M. Dahmani, *Modélisation du calcul distribué pour des problèmes de transport de neutrons à grande échelle*, PhD thesis, École Polytechnique de Montréal, 2006.
- 9) A. Hébert, *Applied Reactor Physics*, Presses Internationales Polytechnique, Montréal (2009).
- 10) A. Hébert and A. Mathonnière, “Development of a Third-Generation Superhomogénéisation Method for the Homogenization of a Pressurized Water Reactor Assembly,” *Nucl. Sci. Eng.*, **115**, 129 (1993).
- 11) R. Sanchez, L. Mao, and S. Santandrea, “Treatment of boundary conditions in trajectory-based deterministic transport methods,” *Nuclear science and engineering*, **140**, 1, 23–50 (2002).