# Tracking on periodic lattices for the method of characteristics

François Févotte[a,*], Simone Santandrea[a], Richard Sanchez[a]

[a] *Commissariat à l'Énergie Atomique (CEA-Saclay) DEN/DANS/DM2S/SERMA/LTSD,
91191 Gif-sur Yvette Cedex, France*

## Abstract

A technique has been developed, allowing the reduction of the storage size requirements for the tracking in the method of characteristics on periodic lattices. This technique takes advantage of repetitions and symmetries in the geometry to compute and store only few minimal tracking data, which can later be fetched and recombined on-the-fly during the sweep to generate all the tracking information needed. This does not affect the results in any way, and has negligible impact on the sweeping efficiency.

The implementation of this method has shown that the storage space savings can be as high as 80% on a cluster of 9 assemblies, while incurring no noticeable slow down of the computation.

## 1. Introduction

In the past years, the Method of Characteristics (MOC) has become a popular tool for the numerical solution of the neutron transport equation (Halsall, 1980; Hong and Cho, 1998; Sanchez et al., 1988; Smith and Rhodes, 2000). The MOC accurately accounts for transport within the regions by means of an analytical integration of the neutron flux along a set of trajectories.

In most practical implementations of the MOC, the trajectories are defined over the whole geometric domain, and the data relative to their intersections with the regions are usually computed and stored beforehand, during the "tracking" phase of the computation. Although most core designs involve periodic arrays of identical cells (e.g. fuel cells in an assembly, or assemblies in the core), very few implementations of the MOC take advantage of repetitions in the geometry to speed-up the tracking process and decrease the amount of data storage needed for each neutron trajectory (Kosaka and Saji, 2000). Moreover, to the authors' knowledge, these implementations have always been limited to simple square lattices for core calculations.

In this work, we show that the neutron trajectories on a lattice of identical cells can be inferred from a well-chosen set of trajectories on only one cell of the lattice. Furthermore we show that this can be done for any reasonably regular cell shape, and in particular for all usual core designs involving square, rectangular or hexagonal lattices. Implementing such a method could lead to a dramatic decrease of the amount of data storage needed for the neutron trajectories.

First, we briefly describe the method of characteristics to emphasize tracking-related issues. We then study the implications of the lattice symmetries on the tracking, which will allow us to state the conditions that a local tracking on a cell must verify to generate a full global tracking. In the fourth section, we pro-

[*] Corresponding author, francois.fevotte@cea.fr
Phone: (+33) 1 69 08 86 09; fax: (+33) 1 69 08 94 90

pose a practical method to build such a local tracking for the usual lattice geometries. We further discuss implementation issues by studying the more general and useful cases in which the geometry may also include non-periodic parts (e.g. a reflector surrounding the assemblies). We then present some numerical examples. We end with a brief conclusion.

## 2. The Method of Characteristics

The method of characteristics provides a solution for the multigroup, $S_N$-discretized formulation of the transport equation in a geometric domain $\mathcal{D}$:

$$\begin{cases} (\mathbf{\Omega}.\nabla + \Sigma)\psi = q, & \forall(\mathbf{r},\mathbf{\Omega}) \in \mathcal{D} \times S_N, \\ \psi = \beta\,\psi + \psi_0, & \forall(\mathbf{r},\mathbf{\Omega}) \in \partial\mathcal{D} \times S_N^-, \end{cases} \tag{1}$$

where $\psi(\mathbf{r},\mathbf{\Omega})$ represents the angular flux at position $\mathbf{r}$ in direction $\mathbf{\Omega}$, $\Sigma(\mathbf{r})$ represents the total cross-section and $q(\mathbf{r},\mathbf{\Omega})$ is the emission density. $\beta$ is an albedo operator on the domain boundary and $\psi_0$ represents an angular flux entering through the boundary ($S_N^-$ is the subset of entering angles from the angular quadrature formula). For the sake of simplicity, we omitted the multigroup indices.

In the MOC, the spatial discretization is achieved by defining an unstructured mesh over the domain and introducing approximated representations for the fluxes within the regions and on the region boundaries. The angular flux, cross-sections and neutron sources are supposed to be constant within each region:

$$\begin{cases} \Sigma(\mathbf{r}) = \Sigma_i, & \\ \psi(\mathbf{r}) = \psi_i, & \forall\mathbf{r} \in \text{region } i. \\ q(\mathbf{r}) = q_i, & \end{cases} \tag{2}$$

Under these assumptions, integrating equation (1) on a straight trajectory $t$ of direction $\mathbf{\Omega}$ crossing region $i$ yields the following transmission and balance equations:

$$\begin{aligned} \psi_i^+(t) = {} & e^{-\Sigma_i\,R_i(t)}\,\psi_i^-(t) \\ & + \frac{1 - e^{-\Sigma_i R_i(t)}}{\Sigma_i}\,q_i(\mathbf{\Omega}), \end{aligned} \tag{3}$$

$$\psi_i(t) = \frac{\psi_i^-(t) - \psi_i^+(t)}{\Sigma_i\,R_i(t)} + \frac{q_i(\mathbf{\Omega})}{\Sigma_i}, \tag{4}$$

with the notations of figure 1: $\psi_i^\pm(t)$ are the angular fluxes entering ($-$) and leaving ($+$) region $i$ along the trajectory $t$, $\psi_i(t)$ is the average angular flux in region $i$ along $t$, and $R_i(t)$ is the intersection length of the trajectory within the region.

Then, starting with a given initial boundary condition and repeatedly applying equations (3) and (4) on a line intersecting the geometric domain, we can compute the average angular fluxes along each line segment intersecting a homogeneous region.

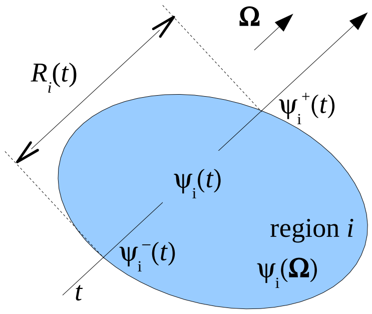

Fig. 1. Transmission of the angular flux along a characteristic line across a region.

The first step in a MOC calculation is thus to define a set of characteristic lines over the geometric domain. In the case of open domains with vacuum boundary conditions, one usually chooses a constant-step mesh of lines in directions given by an arbitrarily chosen quadrature formula. However in the case of closed domains with reflexion or translation boundary conditions, much care must be devoted to the choice of the angular quadrature formula and the characteristic lines, in order to ensure that the whole domain is covered by characteristic lines, while allowing exact treatment of the boundary conditions. This problem has been widely addressed in the literature (Sanchez et al., 2002).

For each characteristic line, the series of intercepted regions and the corresponding intersection lengths are then computed and stored. When the geometric domain is composed of a lattice of identical cells, one could expect the neutron trajectories to show some periodicity, of which we could take advantage to reduce the amount of data storage required for the tracking. Put the other way round, this problem reduces to the following question: a set of trajectories on a cell being given, under which conditions is it possible to reconstruct a set of trajectories on a periodic lattices composed of similar cells?

## 3. Local tracking symmetries

In this section, we consider a lattice composed of identical cells. Let $\mathcal{C}_0$ be the base cell of the lattice; each other cell in the lattice is built applying a translation to $\mathcal{C}_0$, possibly composed with an isometry under which the shape of $\mathcal{C}_0$ is invariant (for example, a reflexion or rotation in the case of a square-shaped cell). Let us call $\mathcal{G}_T$ the group of translations used to build the lattice, $\mathcal{G}_{int}$ the group of isometries corresponding to the internal symmetries of the shape of $\mathcal{C}_0$, and $\mathcal{G}$ the group generated by their union: $\mathcal{G} = \langle \mathcal{G}_T \cup \mathcal{G}_{int} \rangle$.

Considering a trajectory $\mathcal{T}$ over the lattice, we can see it as a series of straight segments intersecting the lattice cells. Since each cell of the lattice can be mapped to any other through an isometry, we may define an equivalence relation between trajectory segments:

$$\begin{aligned} \mathcal{T} \cap \mathcal{C} \equiv \mathcal{T}' \cap \mathcal{C}' \quad & \text{if and only if} \\ \exists m \in \mathcal{G};\ \mathcal{C} = m\mathcal{C}' & \text{ and } \mathcal{T} = m\mathcal{T}', \end{aligned} \tag{5}$$

where $\mathcal{T}$ and $\mathcal{T}'$ are two trajectories and $\mathcal{C}$ and $\mathcal{C}'$ two cells intercepted by these trajectories.

In the following, we will try to describe a global trajectory using only equivalent pieces from a "local tracking" on $\mathcal{C}_0$. A local tracking will be said to be "complete" if it can generate a full "global tracking", i.e. a set of global trajectories on the lattice.

Moreover, since the objective is a decrease in tracking storage, the local tracking needs to contain only a finite number of segments. We therefore choose an angular quadrature formula generating cyclic trajectories.

### 3.1. Translation invariance

For the sake of simplicity, let us first consider an infinite lattice $\mathcal{L}_\infty$, only built with translations. If $\mathcal{T}$ is a trajectory on this lattice, let us study the required equivalent local tracking segments required to describe it:

$$\forall t \in \mathcal{G}_T, \begin{cases} t\,\mathcal{C}_0 \in \mathcal{L}_\infty, \\ \mathcal{T} \cap (t\,\mathcal{C}_0) \equiv \left(t^{-1}\,\mathcal{T}\right) \cap \mathcal{C}_0. \end{cases}$$

A complete local tracking must therefore contain the set of intersection between the base cell $\mathcal{C}_0$ and the orbit of $\mathcal{T}$ under the action of $\mathcal{G}_T$:

$$\mathcal{G}_T\mathcal{T} = \{t\,\mathcal{T};\ t \in \mathcal{G}_T\}.$$

Moreover, since the direction $\mathbf{\Omega}$ of $\mathcal{T}$ itself is defined by a translation $t \in \mathcal{G}_T$ (cf. the cyclic condition in Sanchez et al. (2002)), then we can show that this orbit may also be obtained under the action of a group generated by only one translation:

$$\begin{aligned} &\exists t_0 \in \mathcal{G}_0; \\ &\qquad \mathcal{G}_T\mathcal{T} = \left\{t_0^i\,\mathcal{T};\ i \in \mathbb{Z}\right\}. \end{aligned}$$

$\mathcal{G}_T\mathcal{T}$ is thus a set of parallel lines, with a constant step $\Delta_{\mathbf{\Omega}} = t_0 \cdot \mathbf{\Omega}_\perp$ between them.

### 3.2. Internal symmetries invariance

Now, let us consider the more general and useful case of a lattice $\mathcal{L}$ with boundary conditions and/or rotated or mirrored cells. Let us study a global trajectory $\mathcal{T}$ on this lattice:

$$\forall (t, m) \in \mathcal{G}_T \times \mathcal{G}_{int}, \begin{cases} t \circ m\,\mathcal{C}_0 \in \mathcal{L}, \\ \mathcal{T} \cap (t \circ m\,\mathcal{C}_0) \\ \qquad \equiv \left(m^{-1} \circ t^{-1}\,\mathcal{T}\right) \cap \mathcal{C}_0. \end{cases}$$

In this case, not only must the local tracking be invariant under the action of $\mathcal{G}_T$, but also $\mathcal{G}_{int}$.

As a conclusion, a complete local tracking must be the intersection of $\mathcal{C}_0$ with a set of lines stable under the action of the group generated by $t_0$ and under the action of $\mathcal{G}_{int}$. This last constraint additionally imposes the stability of the angular quadrature formula under the action of $\mathcal{G}_{int}$.

## 4. Method Implementation

Such a complete local tracking is not easy to build because of the numerous internal symmetries to ensure (e.g. six rotations and six reflexions for a hexagonal cell). One simple method could consist of first tracking a segment, and then recursively build its orbit under the action of $\mathcal{G}$. The main drawback of this technique is the lack of control on the obtained local tracking (and therefore on the generated global tracking). In particular, we do not know the final number of segments, nor the minimal distance between two consecutive parallel segments. This can be seen for example on figure 2 where the local tracking does not cover the base cell uniformly*, which could lead to accuracy issues in the computation.

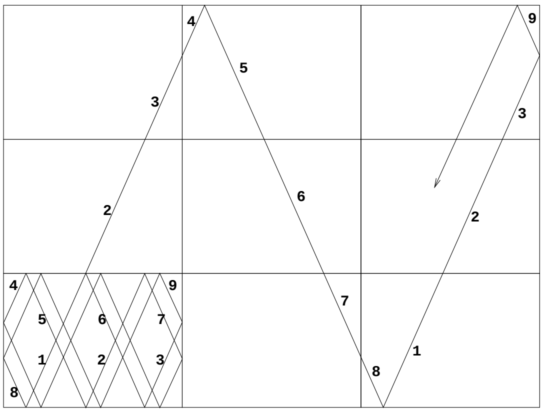

Fig. 2. Lattice of rectangular cells with specular reflexion on the boundaries.

### 4.1. *Construction of the local tracking*

In this section we propose a technique allowing construction of a complete local tracking with a constant step between consecutive segments in the same direction. We need the additional hypothesis that all $\mathcal{G}_{int}$ isometries have a common fixed point $O$. While theoretically a big loss of generality, this is in practice always the case for commonly used geometries (square, rectangles or hexagons).

Let us consider an axis originating from $O$ in direction $\boldsymbol{\Omega}_0$, and define an abscissa $x$ on it. For each direction $\boldsymbol{\Omega} \neq \boldsymbol{\Omega}_0$ in the quadrature formula, we track all the segments originating from the points of abscissae

$$x^{\pm}_{\boldsymbol{\Omega},i} = \pm \left(i + \frac{1}{2}\right) \underbrace{\frac{\Delta_{\boldsymbol{\Omega}}}{\boldsymbol{\Omega}_{\perp} \cdot \boldsymbol{\Omega}_0}}_{\Delta x_0}, \quad i \in \mathbb{N}.$$

As shown in figure 3, the resulting local tracking does not depend on the direction $\Omega_0$ chosen for the axis.

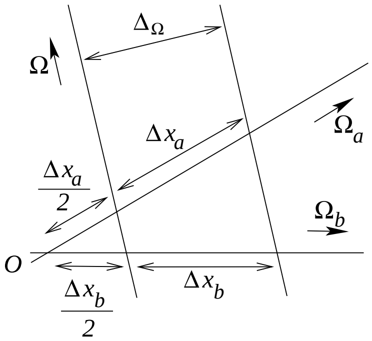

Fig. 3. Thales' theorem shows that the local tracking is independent from the direction $\boldsymbol{\Omega}_a$ or $\boldsymbol{\Omega}_b$ which is chosen as an axis.

By construction, such a local tracking is clearly invariant under the action of $\mathcal{G}_T$. Moreover, it can easily be proved that, since the angular quadrature formula is stable under the action of $\mathcal{G}_{int}$, the local tracking also is.

### 4.2. *Example of local trackings*

If we consider the very simple example of a 2D rectangular cell of size $(a \times b)$ as an example, the above method reduces to the following steps:

1. The internal symmetries of the base cell are the reflexions with respect to the $x$ and $y$ axes. The center of the cell is a fixed point, which allows us to use the technique. The quadrature formula must be stable under these transformations, which means it must be defined over a quadrant and replicated by symmetry.
2. The cyclic angles are associated to the translations defining the lattice (Sanchez et al., 2002):
$$\begin{aligned} t_{n,m} &= n\, e_x + m\, e_y, \\ \tan\phi_{n,m} &= \frac{m\, b}{n\, a}, \\ \Delta_{n,m} &= \frac{a}{|m|} |\sin\phi_{n,m}|. \end{aligned}$$
3. We choose to use the $x$ axis as base for the tracking:
$$\begin{aligned} \Delta x_{n,m} &= \frac{\Delta_{n,m}}{|\sin\phi_{n,m}|} = \frac{a}{|m|}, \\ x^{i,\pm}_{n,m} &= \pm\left(i + \frac{1}{2}\right)\frac{a}{|m|}, \quad i \in \mathbb{N}. \end{aligned}$$

[*] We can see that the distance between two consecutive parallel segments is not homogeneous. Some parts of the geometric domain thus remain less extensively explored than others, which can cause a loss of accuracy.

Thus we need to compute the local tracking starting from the points of abscissae $x^{i,\pm}_{n,m}$ in the directions given by $\phi_{n,m}$ and $\pi - \phi_{n,m}$.

In conclusion, the tracking procedure is very similar to that of the base cell with cyclic boundary conditions. The only additional step is to ensure that the tracking step divides $\Delta x$.

Using this method, one obtains local trackings similar to those shown on figure 4. We can clearly see the additional complexity arising from the need to account for the internal symmetries of the square (rotations by $k\frac{\pi}{2}$) and the hexagon (rotations by $k\frac{\pi}{3}$).

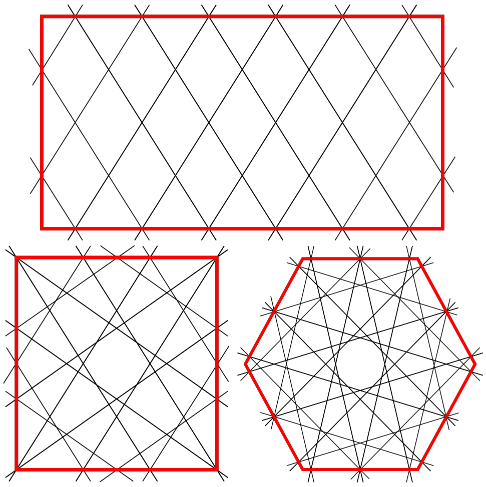

Fig. 4. Local trackings on the usual lattice cell shapes.

| | |
|---|---|
| rectangle: | $(n, m) = (1, 3)$ |
| square: | $(n, m) = (2, 3)$ |
| hexagon: | $(n, m) = (1, 3)$ |

### 4.3. *Interfaces between periodic lattice and non-periodic parts*

In general the geometric domain is not only composed a periodic lattice but also contains a general, non-periodic part (for example a reflector surrounding the assemblies, fig. 5). In this case, we have to deal with interfaces between periodic parts of the trajectories (which can be described using a local tracking) and non-periodic ones (which must be entirely stored).

A first solution could be to store the non-periodic parts of the trajectories along with the local tracking, and then fetch the tracking segments needed on-the-fly, during the sweep. Such a solution requires the ability to determine, for any given position and direction in the geometry, whether we enter a periodic or non-periodic part, and in either case, which segment of the tracking must be followed to resume the sweep. Although this can be computed easily enough, it would entail some unacceptable slow down.

A more flexible technique, although slightly more space-consuming, consists of pre-computing the series of tracking segments that one needs to sweep. This information can be stored in a macro-tracking and be used very efficiently during the sweep. This is the solution that we used: a macro-tracking stores the non-periodic parts of the tracking, interleaved with pointers to the local tracking segments. This only requires few modifications to the traditional sweeping algorithm, while preserving most of its efficiency.

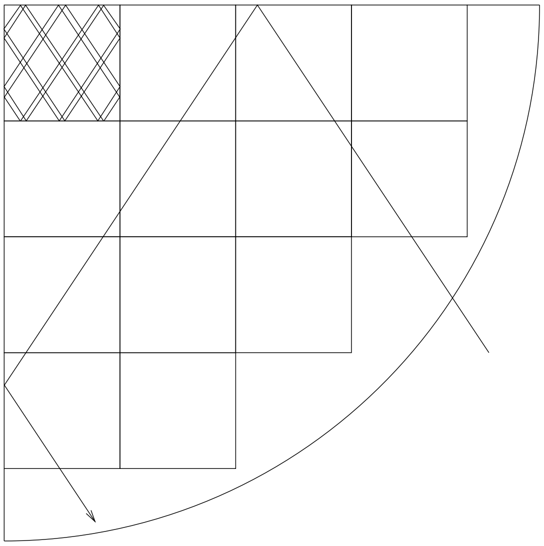

Fig. 5. Example of a non entirely periodic geometry. Some parts of the trajectories can be fetched in a local periodic tracking; some others must be stored extensively.

### 4.4. *Numerical tests and results*

We have implemented this tracking technique in the TDT solver and present here some of the results obtained for different geometry configurations.

#### 4.4.1 *Cluster of cells*

As a very first test-case, we studied a cluster of nine simple cells, such as presented on figure 6. Table 1 shows a comparison of the tracking size and the tracking/sweeping times for the traditional “flat” technique and the periodic tracking technique presented in this paper. In both cases, the results in terms of angular flux and multiplication factor were exactly identical, since the tracking data actually used during the sweep are strictly the same. For such a simple cell, the results are not much in favour of the periodic tracking: the gain in tracking storage is diminished by the importance of the macro-tracking size. Moreover, the additional complexity in the sweeping algorithm makes it just not worth using the periodic tracking on such small scales.

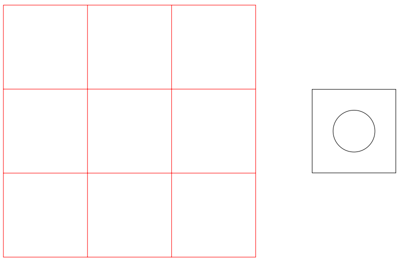

Fig. 6. Cluster of 3 × 3 cells: lattice geometry (left) and detail of the cell geometry (right).

Table 1
Comparison between both techniques on the cell cluster presented on figure 6.

| | Flat | Periodic | rel. diff. |
|---|---|---|---|
| Tracking size (b) | 6 192 | 5 160 | -16.67% |
| Tracking time (s) | 0.18 | 0.26 | +44.44% |
| Sweeping time (s) | 0.89 | 1.15 | +29.21% |

### 4.4.2 *Cluster of assemblies*

The gains become significant for more complex geometries such as the cluster of nine assemblies presented on figure 7, which contains 8 000 regions. The results collected in table 2 show that the tracking size decreases by as much as 81%. The loss with respect to the maximal theoretical gain (which would be $8/9 = 88.9\%$) is due to the additional storage needed for the macro-tracking.

Besides, there aren't any loss with respect to sweeping efficiency (there is even some gain in the sweeping time, probably due to the decreased need of memory swapping).

### 4.4.3 *Non entirely periodical domain*

On a geometric domain which is only partly composed of a lattice, the non-periodic part (such as for example the reflector) must be stored entirely, which limits the maximal theoretical gain in terms of tracking storage size. Still, we obtain advantageous enough storage gains, such as 77% on the geometry presented on figure 8 (in a more realistic computation, the discretization of the reflector would probably limit the gains to approximately 70%).

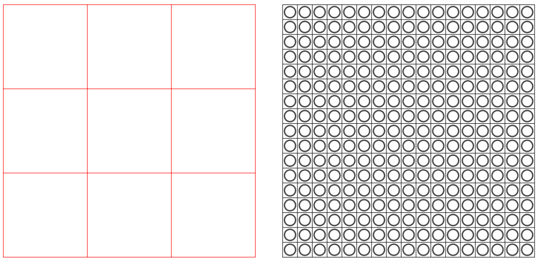

Fig. 7. Cluster of 3 × 3 assemblies: lattice geometry (left) and detail of the assembly geometry (right).

Table 2
Comparison between both techniques on the assembly cluster presented on figure 7.

| | Flat | Periodic | rel. diff. |
|---|---|---|---|
| Tracking size (b) | 3 557 760 | 671 388 | -81.13% |
| Tracking time (s) | 14.43 | 5.32 | -63.13% |
| Sweeping time (s) | 369.42 | 356.58 | -3.48% |

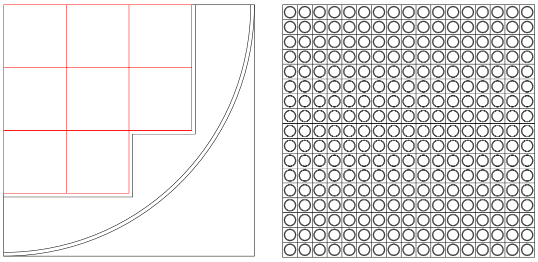

Fig. 8. Lattice of assemblies surrounded by a reflector: global geometry (left) and detail of the assembly (right).

Table 3
Comparison between both techniques on the geometry presented on figure 8.

| | Flat | Periodic | rel. diff. |
|---|---|---|---|
| Tracking size (b) | 3 255 288 | 758 832 | -76.69% |
| Tracking time (s) | 16.78 | 6.33 | -62.28% |
| Sweeping time (s) | 773.78 | 762.79 | -1.42% |

## 5. Conclusion

We have developed and implemented a tracking technique for the method of characteristics that takes advantage of repetitions and symmetries in the geometry to reduce the tracking storage size. Such a technique is useful for geometries which are composed, at least partly, of a lattice of identical cells.

Our numerical examples show that this method can lead to dramatic decreases in the tracking storage requirements (up to 80% for example on a cluster of nine assemblies). It is however unadapted to cases where the periodic cell is too simple (such as lattices of fuel rod cells). In any case, the obtained results are strictly identical to those which would come from a traditional full-tracking computation.

In our current implementation, only square and rectangular cell shapes are supported[†]. The global shape of the geometric domain is also currently limited to rectangles. We plan to remove these limitations to allow computation of hexagonal lattices or of eighth of core geometries.

## References

Halsall, M. J., 1980. CACTUS, a characteristics solution to the neutron transport equations in complicated geometries, Technical Report AEEW-R 1291, Atomic Energy Establishment, Winfrith, Dorchester, Dorset, United Kingdom.

Hong, S. G. and Cho, N. Z., 1998. ‘CRX: a code for rectangular and hexagonal lattices based on the method of characteristics’, *Annals of Nuclear Energy* **25**, 547–565.

Kosaka, S. and Saji, E., 2000. ‘Transport theory calculation for a heterogeneous multi-assembly problem by characteristics method with direct neutron path linking technique’, *Journal of Nuclear Science and Technology* **37**(12), 1015–1023.

Sanchez, R., Mao, L. and Santandrea, S., 2002. ‘Treatment of boundary conditions in trajectory-based deterministic transport methods’, *Nuclear Science and Engineering* **140**, 23–50.

Sanchez, R., Mondot, J., Stankovski, Z., Cossic, A. and Zmijarevic, I., 1988. ‘APOLLO II: a user-oriented, portable, modular code for multigroup transport assembly calculations’, *Nuclear Science and Engineering* **100**, 352.

Smith, K. S. and Rhodes, J. D., 2000. ‘CASMO-4 characteristics methods for two-dimensional PWR and BWR core simulations’, *Trans. Am. Nucl. Soc.* **83**, 294–296.

[†] Although the local tracking technique is independent from the cell geometry, the macro-tracking construction method must be adapted to the number of coupled angles in the angular quadrature formula as well as to other geometry-dependant parameters.