

GENERATING ALBEDO BOUNDARY CONDITIONS FOR NODAL DIFFUSION CODES USING SERPENT 2

Ville Valtavirta¹, Jaakko Kuopanportti², and Antti Rintala¹

¹VTT Technical Research Centre of Finland Ltd
P.O. Box 1000, FI-02044 VTT, FINLAND

²Fortum Power and Heat Ltd
P.O. Box 100, FI-00048 FORTUM, FINLAND

ville.valtavirta@vtt.fi, jaakko.kuopanportti@fortum.com, antti.rintala@vtt.fi

ABSTRACT

We use the Serpent Monte Carlo code to produce total and partial albedo boundary conditions that can be used to model the Loviisa NPP VVER-440 core with the nodal neutronics tools of Fortum. The albedo generation process is described in detail. The dependence of the generated albedos on boron content and water density is investigated and a clear distinction is noted in water density dependence between regions containing mostly water and those containing mostly structural materials. The Serpent generated albedos are currently used in production calculations for modeling the Loviisa reactors at Fortum.

KEYWORDS: Serpent, albedo, nodal diffusion, boundary condition, Apros

1. INTRODUCTION

Reflector modelling is a necessary part of core level nodal calculations as it provides a boundary condition for the neutronics solution in the active core. The axial boundary conditions affect the axial buckling of the flux, whereas the radial boundary conditions affect the radial buckling/tilt of the flux and have a significant effect on the power distribution at the core periphery.

The power distribution at the periphery of the core is important to the neutron dose of the reactor pressure vessel (RPV), especially in cores lacking heavy reflector or large water volume between the active core and the RPV (e.g. VVER-440 reactors). New heavy radial reflector designs (EPR, NuScale) aiming at improving the neutron economy of the reactor bring their own importance to the proper modelling of the radial reflector. Various approaches for reflector modelling exist, from a full nodal solution in the reflector to simplified pre-calculated boundary conditions applied at the core–reflector interface.

One approach is the use of an albedo boundary condition, where the reflector is represented using an albedo matrix formulation that allows the calculation of the core-returning neutron current distribution based on the core-exiting neutron current distribution.

The elements of the albedo matrix describe which fraction of neutron current that exits the core in energy group g returns as neutron current in energy group g' . The elements of the albedo matrix

have to be pre-calculated and are usually calculated and stored separately for different types of reflector nodes.

We describe the use of the continuous energy Monte Carlo particle transport code Serpent 2 [1] for evaluating the axial and radial albedos for a VVER-440 type reactor core. As an example of production level calculations, Serpent is used to generate the albedo boundary conditions needed for modelling the Loviisa NPP VVER-440 reactor core with the nodal neutronics tools of Fortum (HEXBU-3D [2] and Apros* [3]). The albedos are generated as functions of coolant temperature (and density) and boron content and the dependence of the albedos on these state variables is discussed.

2. ALBEDO BOUNDARY CONDITIONS

Several different albedo formulations can be used as boundary conditions at the boundary between the active core and the reflector. Here, we consider the albedos used by the nodal neutronics tools of Fortum, which are divided into *total albedos* and *partial albedos*. In the following sections, we will give concise descriptions of both albedo types.

2.1. Total Albedos

Generally, the albedo approach is applied on each boundary surface between an active core node and a reflector node and is based on using known core-exiting neutron currents (e) in each group to evaluate core-returning neutron currents (r) through surfaces (s) in each group.

$$\begin{pmatrix} J_1^{r,s} \\ J_2^{r,s} \end{pmatrix} = \begin{pmatrix} \alpha_{1 \rightarrow 1}^{\text{tot}} & \alpha_{2 \rightarrow 1}^{\text{tot}} \\ \alpha_{1 \rightarrow 2}^{\text{tot}} & \alpha_{2 \rightarrow 2}^{\text{tot}} \end{pmatrix} \begin{pmatrix} J_1^{e,s} \\ J_2^{e,s} \end{pmatrix}. \quad (1)$$

In the general case, the total albedos ($\alpha_{g \rightarrow g'}^{\text{tot}}$) are different for the different core-facing partial surfaces of the reflector node.

2.2. Partial Albedos

The partial albedo model was originally developed to represent the side-to-side couplings of VVER-440 control elements. It was soon used also in the modelling of the radial reflector regions in the VVER-440 reactors. The model has additional details compared to the total albedo model as it allows different faces of reflector nodes to have different albedos. In essence, the partial albedos represent the fraction of core-exiting current \vec{J}^e that returns to core through the same partial surface (A-albedos), the adjacent partial surfaces (B- and F-albedos), the next-to-adjacent partial surfaces (C- and E-albedos) or the opposite surface of the reflector node (D-albedos). These partial currents are illustrated in Fig. 1.

Due to the symmetries of the VVER-440 control elements where this model was first applied, the model assumes that F-albedos are equal to B-albedos and E-albedos are equal to C-albedos.

*Apros is jointly owned by Fortum and VTT.

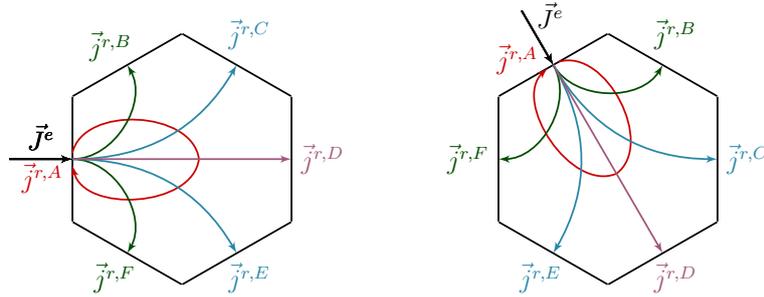


Figure 1: The core-exiting total neutron current \vec{J}^e can lead to several partial currents returning to the core. The partial albedo model labels these based on the returning side relative to the exiting side.

The partial albedo model evaluates the core-returning currents using effective albedos instead of total albedos

$$\begin{pmatrix} J_1^{r,s} \\ J_2^{r,s} \end{pmatrix} = \begin{pmatrix} \alpha_{1 \rightarrow 1}^{\text{eff},s} & \alpha_{2 \rightarrow 1}^{\text{eff},s} \\ \alpha_{1 \rightarrow 2}^{\text{eff},s} & \alpha_{2 \rightarrow 2}^{\text{eff},s} \end{pmatrix} \begin{pmatrix} J_1^{e,s} \\ J_2^{e,s} \end{pmatrix}. \quad (2)$$

where the effective albedos are calculated based on the core-exiting current distribution and the partial albedos via

$$\alpha_{g \rightarrow g'}^{\text{eff},s} = \alpha_{g \rightarrow g'}^A + \frac{1}{J_g^{e,A}} [(J_g^{e,B} + J_g^{e,F}) \alpha_{g \rightarrow g'}^B + (J_g^{e,C} + J_g^{e,E}) \alpha_{g \rightarrow g'}^C + J_g^{e,D} \alpha_{g \rightarrow g'}^D]. \quad (3)$$

This means that the core-returning current to one fuel node can actually depend on core-exiting currents in the other fuel nodes that are adjacent to the same reflector node. In practice, calculating the partial albedos α^A , α^B , α^C and α^D in a consistent manner requires the tallying of core-exiting total currents and core-returning side-to-side and group-to-group partial currents for the whole reflector region.

3. GENERATING ALBEDOS WITH SERPENT

The `set alb` input card of Serpent allows the user to specify a bounding surface for which albedos are generated. With this card, Serpent will automatically tally the entering and exiting currents for each side of the surface and each energy group. Serpent uses the tallied currents to evaluate total albedos

$$\alpha_{s,g \rightarrow g'}^{\text{tot}} = \frac{J_{g'}^{r,s}}{J_g^{e,s}}. \quad (4)$$

and partial albedos

$$\alpha_{s \rightarrow s',g \rightarrow g'}^{\text{part}} = \frac{j_{g \rightarrow g'}^{s \rightarrow s'}}{J_g^{e,s}}. \quad (5)$$

for each side of the surface. The partial albedos evaluated by Serpent correspond to a response matrix representation of the system bounded by the surface.

The automatic albedo calculation of Serpent works well in most cases, but certain peculiarities of the partial albedo model require additional tools. The partial albedo model binds core-returning partial currents to core-exiting partial currents, but the core-exiting energy group is only available in Serpent results if the neutron current entered the surface from a fuel region. The data is lost for neutrons entering from adjacent reflector nodes.

It would be possible to construct a response matrix representation for the whole reflector region and solve for the partial currents using a response matrix solver, but the approach taken here was to use the *detector flagging* and *conditional scoring* capabilities of Serpent. Detector flagging is based on the ability to set or reset certain flags for a particle each time it contributes to a detector tally. On the other hand, detectors can be scored conditionally based on whether certain flags are set or unset for a particle. This procedure allows the user to, e.g. only score a certain current detector if a particle has exited the core through a specific partial surface and in a specific energy group, which can be used to evaluate the group-to-group partial currents between all core boundary surfaces.

4. CALCULATION MODEL

Table 1: Water density parametrization used in the albedo generation and corresponding water temperatures. The state points are divided into powered core states and non-powered core states.

Water density (g/cm ³)	Water temperature (°C / K)	Powered state?
0.20	326.6 / 599.75	powered
0.40	326.6 / 599.75	powered
0.60	326.6 / 599.75	powered
0.70	308.9 / 582.05	powered
0.71	303.0 / 576.15	powered
0.75	285.0 / 558.15	powered
0.78	267.0 / 540.15	powered
0.79	260.0 / 533.15	zero power
0.90	174.6 / 447.75	zero power
1.00	33.1 / 306.25	zero power

The albedos were parametrized with respect to

1. The water density in the reflector region. The ten density points are shown in Table 1.
2. The boron content in the water (0 ppm, 500 ppm, 1000 ppm, 1500 ppm and 2275 ppm).

In order for the albedos to cover both operating and shutdown conditions, the water density state points were divided into powered and zero power states, which was reflected in the simulation model as changes in the coolant axial temperature and density distributions, fuel xenon distribution and fuel temperature distribution used in the calculations. The multi-physics capabilities of Serpent

allow the user to overlay material temperature and density distribution on top of the geometry model without requiring any further discretization of the geometry model [1].

As the parametrization needed to be applied separately for the axial reflector and the radial reflector, the two sets of albedos were generated using two separate models.

4.1. Axial Model

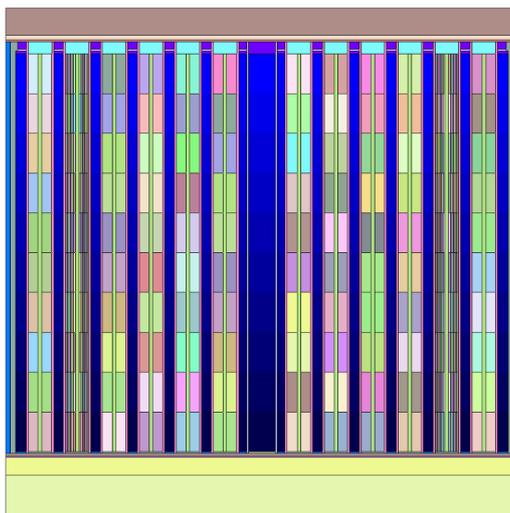


Figure 2: Vertical cut (not to scale) from the diagonal of the hexagonal single assembly 3D geometry model used for generating the axial total albedos for the upper and lower reflectors. The fuel in each rod is divided axially into 10 depletion regions for equilibrium xenon-calculation. In powered state points, a realistic coolant temperature/density distribution is applied in the active core.

The albedos for the top and bottom reflector were generated in a radially infinite single-assembly 3D geometry (Fig. 2). The reflector regions closest to the active fuel were modeled in detail, whereas regions more distant were homogenized while conserving the masses of water and structural materials. The lower reflector consisted of five separate axial regions, whereas the upper reflector was constructed of six separate axial regions.

The albedos were generated using the `set alb` input card for Serpent with hexagonal prism surfaces surrounding the whole of the upper and lower reflectors. While the same model was used to generate both lower and upper albedos, in some state points only one (either upper or lower) of the generated albedos were used. This is caused by the fact, that some of the water temperatures cannot be achieved in one axial end of the reactor during its operation.

4.2. Radial Model

In order to calculate the partial albedos and total albedos for each radial reflector node, several neutron currents through the radial reflector need to be tallied. Considering Fig. 3, we need to tally the exiting total currents in each group through all of the red-numbered partial surfaces between the core and the reflector as well as the returning partial currents from each partial surface to each partial surface and from each group to each group.

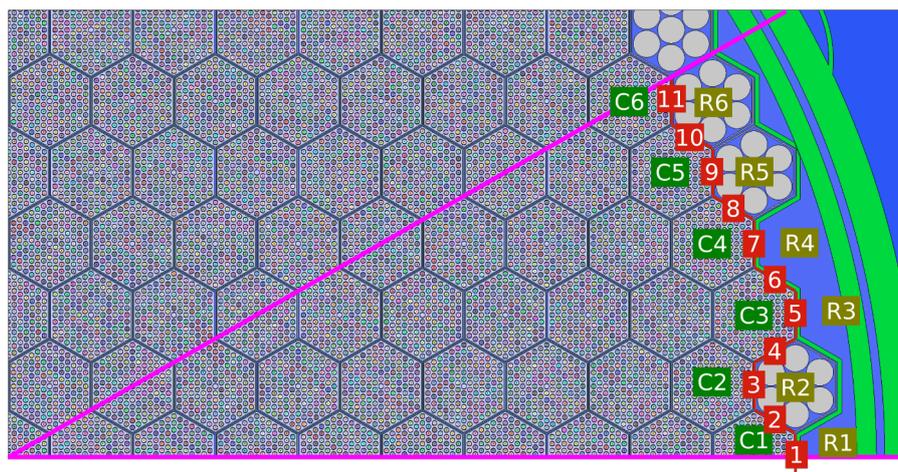


Figure 3: Annotated core reflector boundary showing also several shielding assemblies at the core periphery (R2, R5, R6). The core boundary nodes are indicated with C_x on a green background. The partial surfaces on the core reflector boundary are indicated with numbers on a red background. The reflector boundary nodes are indicated with R_x on a yellow-green background. The magenta lines indicate planes of reflective boundary condition.

In the past, the partial albedos for the radial reflector nodes were generated using MCNP in a multi-stage approach. First, surface source distributions were generated for each core-boundary partial surface (and group) in a criticality source simulation. Second, each surface source was used in a separate fixed source calculation to calculate the returning partial currents due to that exiting current. Now, the same partial currents can be evaluated with a single Serpent calculation using detector flagging and conditional scoring. For example, in order to calculate the partial current $j_{1 \rightarrow 2}^{r, 2 \rightarrow 7}$ at the core boundary, we need to score neutrons that exited the core from node C1 through partial surface 2 to reflector node R2 in the fast group and returned from R4 to C4 through surface 7 in the thermal group. The flagging and scoring procedure relies on several groups of detectors:

1. Out-of-core detectors that track and flag the core node the particle last exited and the particles exiting energy group.
2. In-to-reflector detectors that track and flag the reflector node the particle first entered.
3. Out-of-reflector detectors that track and flag the reflector node the particle last exited.
4. In-to-core detectors that track and flag the core node the particle first returned to and the particles returning energy group.

The results from these detectors could be used in evaluating the partial albedos for each of the reflector nodes.

5. RESULTS

5.1. Dependence of Total Albedos on State Variables

In analyzing the dependence of the total albedos on state variables, we focus on radial reflector nodes R4 and R5 (see Fig. 3). Node R4 contains a large amount of water, whereas node R5 contains a shielding element with a large amount of stainless steel and a reduced water volume.

The behavior of the total albedos of the upper and lower axial reflector are closer to that observed in node R4 than that observed in node R5.

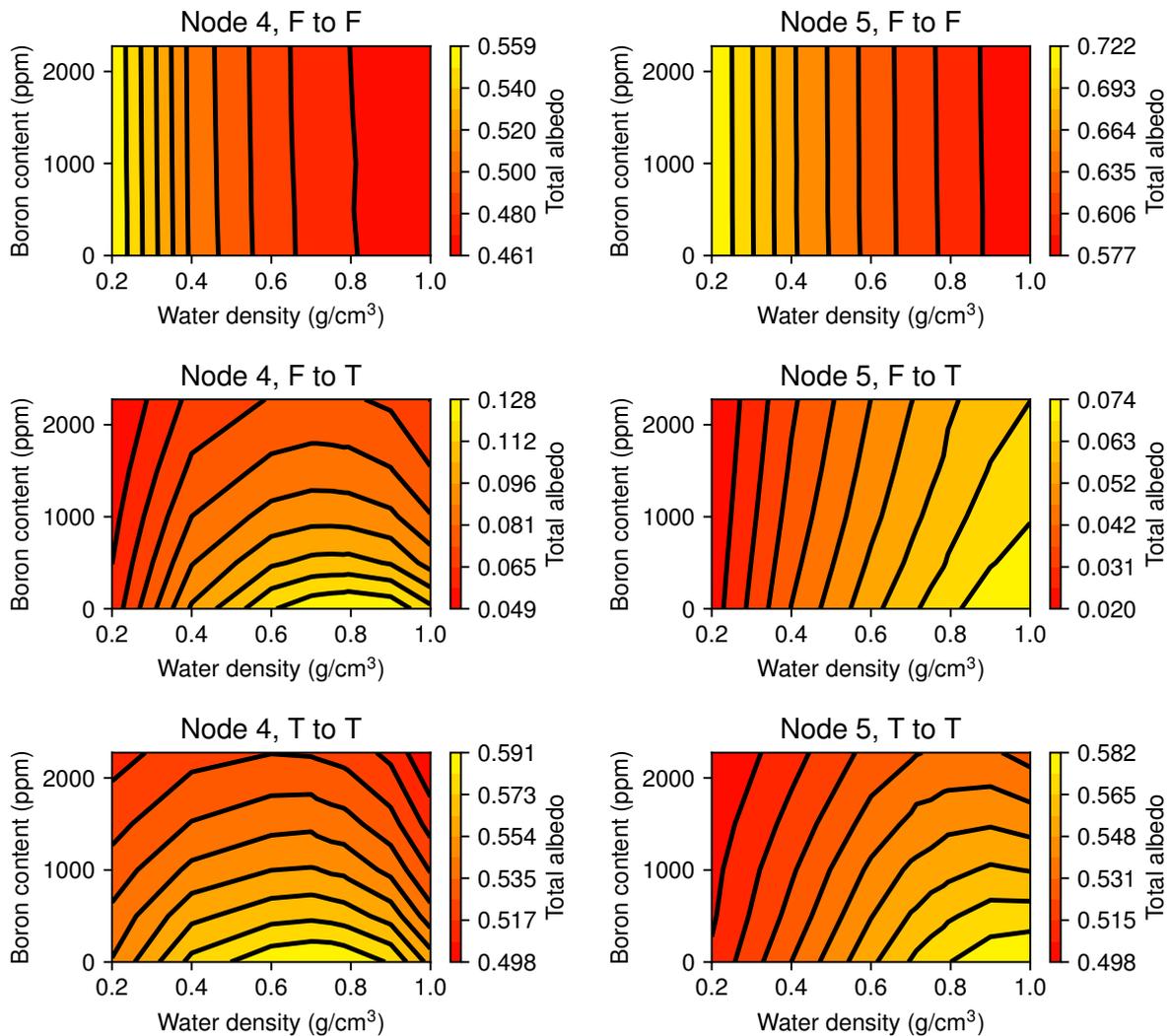


Figure 4: Dependence of fast-to-fast, fast-to-thermal and thermal-to-thermal total albedos on water density and boron content in the radial reflector nodes R4 and R5. 9 contours (10 shades) uniformly spaced between the minimum and maximum value.

The fast-fast albedos shown in Fig. 4 (top) were largely insensitive to the boron content, being only dependent on the water density regardless of the reflector node type (upper, lower or radial reflector). The water density dependence of the fast-fast albedo was slightly more linear in the steel-containing shielding elements than in the regions that contained more water.

The fast-thermal albedos (downscattering albedos) were sensitive both to the boron content and the water density. In non-shielding-element albedo nodes that contained significant amounts of water, a clear and complex behavior can be seen (Fig. 4 middle left). The shielding elements, however show only the left side of the behavior (Fig. 4 middle right).

The upscattering albedos are effectively zero and are not investigated further.

The thermal-thermal albedos show a complex behavior depending on both water density and boron content (Fig. 4 bottom) not unlike that observed in the downscattering albedos. Again, the behavior in the shielding elements (e.g. node R5) only captures the left side of the contour curves, which can be expected based on the reduced amount of water in the node.

For axial albedos, accounting correctly for coolant density is more important in the upper reflector region, as the changes in coolant density are much larger in the upper parts of the reflector in power operation.

5.2. Burnup Dependence of Albedos

The burnup dependence of albedos was analyzed when generating the axial albedos by depleting the 3D assembly model and calculating separate albedos for the different burnup points. The burnup calculation for the 3D model required slightly less than 16 GB of RAM using the least memory consuming optimization mode of Serpent (`set opti 1`). The burnup dependence of the albedos was not significant and the albedos were not further parametrized against burnup.

6. CONCLUSIONS

Axial and radial albedos for the Loviisa VVER-440 reactors were generated using the Serpent 2 Monte Carlo code. The work utilized both the automatic albedo calculation of Serpent and the manual albedo calculation made possible by the detector flagging and conditional scoring capabilities of Serpent. The dependence of albedos on water density and boron content was investigated and showed only a density dependence for fast-to-fast albedos whereas fast-to-thermal and thermal-to-thermal albedos showed a complex behavior depending on both variables. The burnup dependence of the albedos was investigated in relation to the axial albedos and was found minor. The Serpent generated albedos are currently used in production calculations for modeling the Loviisa reactors at Fortum.

REFERENCES

- [1] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho. “The Serpent Monte Carlo code: Status, development and applications in 2013.” *Annals of Nuclear Energy*, **volume 82**, pp. 142 – 150 (2015). URL <http://www.sciencedirect.com/science/article/pii/S0306454914004095>.
- [2] T. Lahtinen, S. Saarinen, and M. Antila. “Effect of HEXBU-3D feedback and boundary condition models on the calculated reactor core characteristics of Loviisa NPP” In *Proc. 18th symposium of AER on VVER Reactor Physics and Reactor Safety*, pp. 787–798. Eger, Hungary (2008).
- [3] “Apros - Dynamic Process Simulation Software for Nuclear and Thermal Power Plant Applications.” URL <http://www.apros.fi/>. Visited 23 Oct. 2019.