BWR Geometry Enhancements for the Polaris Lattice Physics Code[1]

Matthew A. Jessee, William A. Wieselquist, Cole A. Gentry, Ugur Mertyurek

Reactor Physics Group, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN
jesseema@ornl.gov, wieselquiswa@ornl.gov, gentryca@ornl.gov, mertyureku@ornl.gov

INTRODUCTION

Polaris is a two dimensional lattice physics module in the SCALE code system [1] that is used to analyze light water reactor (LWR) fuel assemblies. Polaris provides multigroup (MG) lattice physics solutions using the method of characteristics (MOC) method for transport calculations. Self-shielding calculations are performed using the embedded self-shielding method (ESSM), and depletion calculations are performed with ORIGEN. Polaris uses ENDF/B-VII.1-based 252-group and 56-group libraries from SCALE 6.2. A detailed summary of the Polaris calculational methods is provided in Jessee et al. 2014 [2].

Polaris was initially released as part of SCALE 6.2 in April 2016. Several enhancements were implemented in the SCALE 6.2.1 update that was made available in July 2016, including the addition of P_N scattering treatment in the MOC solver, and hydrogen transport cross section correction based on the approach of Herman et al. [3]. The impacts of these calculational enhancements for full-core analysis using Polaris/PARCS are described by Xu et al. [4].

This paper describes updates to the Polaris geometry package that were implemented for modeling boiling water reactor (BWR) lattice designs. The BWR geometry enhancements are available in the SCALE 6.2.2 update released in May 2017.

POLARIS GEOMETRY

The primary function of the Polaris geometry package is to convert the user-provided description of the problem geometry, or the input, into an internal KENO-VI-based representation of the problem geometry: the geometry scene. The geometry scene is passed to the KENO-VI geometry ray-tracer, which sets up the particle tracks for the MOC transport calculations. Each region in the geometry scene is defined as an intersection of N shapes and M shape complements. Accurate computation of the region area is a fundamental requirement for the Polaris geometry package. The KENO-VI geometry package does not compute region areas analytically, so the user (i.e., Polaris) must provide the region areas.

In its original implementation, Polaris computed region areas based on the geometry data provided for the lattice design. For a simple example, consider a cylindrical fuel pin in a square-pitched flow channel with pitch p and fuel radius r. Using integer 1 for the square channel ID and 2 for the fuel cylinder ID, Polaris defines the coolant region in KENO-VI syntax as "1 − 2", i.e., inside shape 1 and not inside shape 2. In set notation, this region definition is written as "$1 \cap 2^c$" and reads, "shape 1 intersect shape 2 complement."[2] Formerly, Polaris would have calculated the area of the coolant region as $p^2 - \pi r^2$.

For pressurized water reactor (PWR) lattice designs, the region definitions are simple enough so that region areas can be computed in Polaris via a small internal library of formulas, such as the areas of rectangle, circle, triangle, circular sector, and circular segment. BWR lattice designs require modeling of control blades, curved channel boxes, and complicated internal water cross structures. These features require complicated (i.e., error-prone) logic to compute region areas. To simplify the area calculation while maintaining sufficient accuracy, a new approach was implemented in which each KENO-VI shape is modeled as a convex polygon, where the convex polygon is defined based on a counter-clockwise ordering of the polygon vertices. Given the counter-clockwise ordering of the vertices of the polygon–$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$–the polygon area is given as:

$$A = \frac{1}{2}\Big[\Big(\sum_{i=1}^{n-1}\begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1}\end{vmatrix}\Big) + \begin{vmatrix} x_n & y_n \\ x_1 & y_1\end{vmatrix}\Big]. \quad (1)$$

Convex polygon intersect operations

As stated above, Polaris must calculate the area of a region defined as an intersection of N shapes and M shape complements. This area calculation is simplified with the convex polygon approximation. There are several algorithms for computing the intersection of two convex polygons. The Sutherland-Hodgman intersection algorithm [5] is used in Polaris. If two convex polygons intersect, the resulting polygon is also convex, for which the area can be computed by Eq. (1). If a region is defined as the intersection of N convex polygons, the Sutherland-Hodgman intersection algorithm can be applied $N - 1$ times to determine the final convex polygon and region area as shown in algorithm 1. Only the final polygon and area are needed to define the region in KENO-VI.

The intersect area of a shape and a shape complement is given as:

$$A(S_1 \cap S_2^c) = A(S_1) - A(S_1 \cap S_2). \quad (2)$$

The area of the intersect of S_1 and S_2^c is equal to the area

[1]This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

[2]In set notation, "$1 \cap 2^c$" can also be written as "$1 \setminus 2$", i.e., "the relative complement of shape 2 with respect to shape 1" or as "1 − 2" (i.e., "shape 1 difference shape 2"). For this paper, "$1 \cap 2^c$" is chosen to facilitate the discussion of area calculations based on convex polygons.

Algorithm 1 Intersection of N convex polygons

function AreaPolygonSet($Q_1, Q_2, \ldots Q_n$)
  $X \leftarrow$Intersect(Q_1, Q_2) ▷ Sutherland-Hodgman
  for $i \leftarrow 3, n$ **do**
    $X \leftarrow$Intersect(X, Q_i)
  end for
  return PolygonArea(X) ▷ Eq. (1)
end function

of S_1 minus the area of the intersection of S_1 and S_2. This equation can be applied recursively to compute the intersect area of a convex polygon and M convex polygon complements. For example, consider computing the area of Y, defined as $Y = P \cap Q_1^c \cap Q_2^c$, where P, Q_1, and Q_2 are convex polygons. Substituting $X_1 \equiv P \cap Q_1^c$, the area of Y is:

$$\begin{aligned} & & A(Y) &= A(P \cap Q_1^c \cap Q_2^c) \\ X_1 \equiv P \cap Q_1^c \Rightarrow & & &= A(X_1 \cap Q_2^c) \\ & & &= A(X_1) - A(X_1 \cap Q_2) \end{aligned}$$

In this equation, $A(X_1)$ is computed from Eq. (2) as $A(P) - A(P \cap Q_1)$, and $A(X_1 \cap Q_2)$ is expanded to be $A(P \cap Q_1^c \cap Q_2)$. The area of this final term can be computed by first computing $X_2 \equiv P \cap Q_2$ and then applying Eq. (2) to compute $A(X_2 \cap Q_1^c)$.

Algorithm 2 computes the intersect area of convex polygon and M convex polygon complements. This algorithm requires $O(M^2)$ intersect calculations. This algorithm can be combined with algorithm 1 to compute the intersect area of N convex polygons and M convex polygon complements. The KENO-VI region is defined based on the final intersect polygon from algorithm 1 and the M convex polygon complements used in algorithm 2.

Algorithm 2 Intersection of a convex polygon (P) and M convex polygon complements ($Q_1, \ldots, Q_m$)

function AreaComplement($n, P, Q_1, \ldots, Q_m$)
  if m = 1 **then**
    $a \leftarrow$ PolygonArea(P) ▷ Eq. (1)
    $X \leftarrow$ Intersect(P, Q_1) ▷ Sutherland-Hodgman
    $b \leftarrow$ PolygonArea(X)
    return $a - b$
  else
    $a \leftarrow$ AreaComplement($m - 1, P, Q_1, \ldots, Q_{m-1}$)
    $X \leftarrow$ Intersect(P, Q_m)
    $b \leftarrow$ AreaComplement($m - 1, X, Q_1, \ldots, Q_{m-1}$)
    return $a - b$
  end if
end function

RESULTS AND ANALYSIS

The polygon-based area algorithms have been benchmarked against The Computational Geometry Algorithms Library (CGAL) [6]. Portions of CGAL have been implemented into Polaris for the calculation of region areas. The CGAL-based coding can be disabled or enabled based on CMake configuration settings for compiling SCALE. In early phases of polygon-based area implementation, CGAL was used to compute region areas based on the exact definition of the shapes. The CGAL-based calculations were used to determine the appropriate number of polygon sides necessary to approximate curved surfaces in Polaris. By default, Polaris uses one side per 3° of curvature, which leads to a 0.001% maximum relative error in region area computation for a BWR lattice. In later phases of development, CGAL was used to compute the region areas based on the convex polygon definitions consistent with the Polaris algorithms. In these comparisons, CGAL and Polaris computed identical region areas. Although CGAL was used for code verification, CGAL cannot be used directly in Polaris and SCALE due to licensing restrictions.

The key BWR features supported by the Polaris geometry package are provided in Fig. 1. The top row displays ATRIUM 9 × 9 and 10 × 10 fuel, the middle row displays SVEA-100 and SVEA-96 fuel, and the bottom row displays GE 8 × 8 and 9 × 9 fuel. The ATRIUM lattices display the newly supported square water box feature, which displaces a 3 × 3 location in the lattice pin map in the ATRIUM design. The SVEA lattices display the newly supported water cross option, and the GE lattices display different cylindrical water rod designs.

For the BWR channel box, the top row in Fig. 1 displays straight-edged channel box with uniform thickness. The middle row displays a curved channel box with uniform thickness. The bottom row displays a curved channel box with variable thickness.

For the BWR control blade, the top, middle, and bottom rows display OEM, Marathon, and uniform blade designs respectively. The left column displays a uniform bypass channel, and the right column displays a nonuniform channel.

Additional supported features not shown in Fig. 1 include (a) flexible control blade definition as a series of pin and/or slab sections, (b) variable control blade wing tip radius, including a straight-edged tip, and (c) a pin displacement option to translate the center of the pin locations as required by some lattice designs.

One hallmark feature of Polaris is the simplified input description. The Polaris input file contains simple input cards for `pin`, `box`, `blade`, `cross`, `pinmap`, `dxmap`, and `dymap` for defining BWR lattice models. A full input file listing is provided in Fig. 2 for a representative ATRIUM 9 × 9 design.

Fig. 3 presents k_∞ comparisons between reference continuous energy (CE) KENO-VI calculations and Polaris calculations for 362 lattice configurations composed of various lattice designs and operating conditions, as well as enrichment and gadolinium distributions and control blade insertion. 82% of these calculations were within target accuracy criteria of 200 pcm, 1% root mean squared pin power difference, and 1.5% maximum pin power difference. Most cases outside of the target accuracy are for cold zero power configurations with control blade insertion. These calculations are under further investigation.

CONCLUSIONS

The Polaris lattice physics code has been updated to support BWR lattice designs. This summary outlines the geometry modeling approach for BWR lattices with region area calculation based on convex polygons. The area calculation was verified by calculations with CGAL. Computational benchmark results presented in Fig. 3 display acceptable agreement with reference CE KENO-VI solutions for a wide range of lattice geometries and conditions.

ACKNOWLEDGMENTS

The authors would like to thank the US Nuclear Regulatory Commission Office of Research for its support in the development of Polaris.

REFERENCES

1. B. T. REARDEN and M. A. JESSEE, "SCALE Code System," Tech. Rep. ORNL/TM-2005/39, Oak Ridge National Laboratory (2016), available from Radiation Safety Information Computational Center as CCC-834.
2. M. A. JESSEE, W. A. WIESELQUIST, T. M. EVANS, S. P. HAMILTON, J. J. JARRELL, K. S. KIM, J. P. LEFEBVRE, R. A. LEFEBVRE, U. MERTYUREK, and M. L. WILLIAMS, "Polaris: A New Two-Dimensional Lattice Physics Analysis Capability for the SCALE Code System," in "PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future, The Westin Mikayo, Kyoto, Japan, September 28 - October 3, 2014," (October 2014).
3. B. R. HERMAN, B. FORGET, K. SMITH, and B. N. AVILES, "Improved Diffusion Coefficients Generated from Monte Carlo Codes," in "International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Sun Valley, ID, May 5-9, 2013," (May 2013).
4. K. XU, M. JESSEE, A. WARD, and T. DOWNAR, "Two-Step Analysis of Watts Bar Nuclear 1 Cycle 1 with SCALE/PARCS," (June 2017), to be submitted to 2017 ANS Annual Meeting.
5. I. SUTHERLAND and G. W. HODGMAN, "Reentrant Polygon Clipping," *Communications of the ACM*, **17**, 32–42 (1974).
6. THE CGAL PROJECT, *CGAL User and Reference Manual*, CGAL Editorial Board, 4.9 ed. (2016).

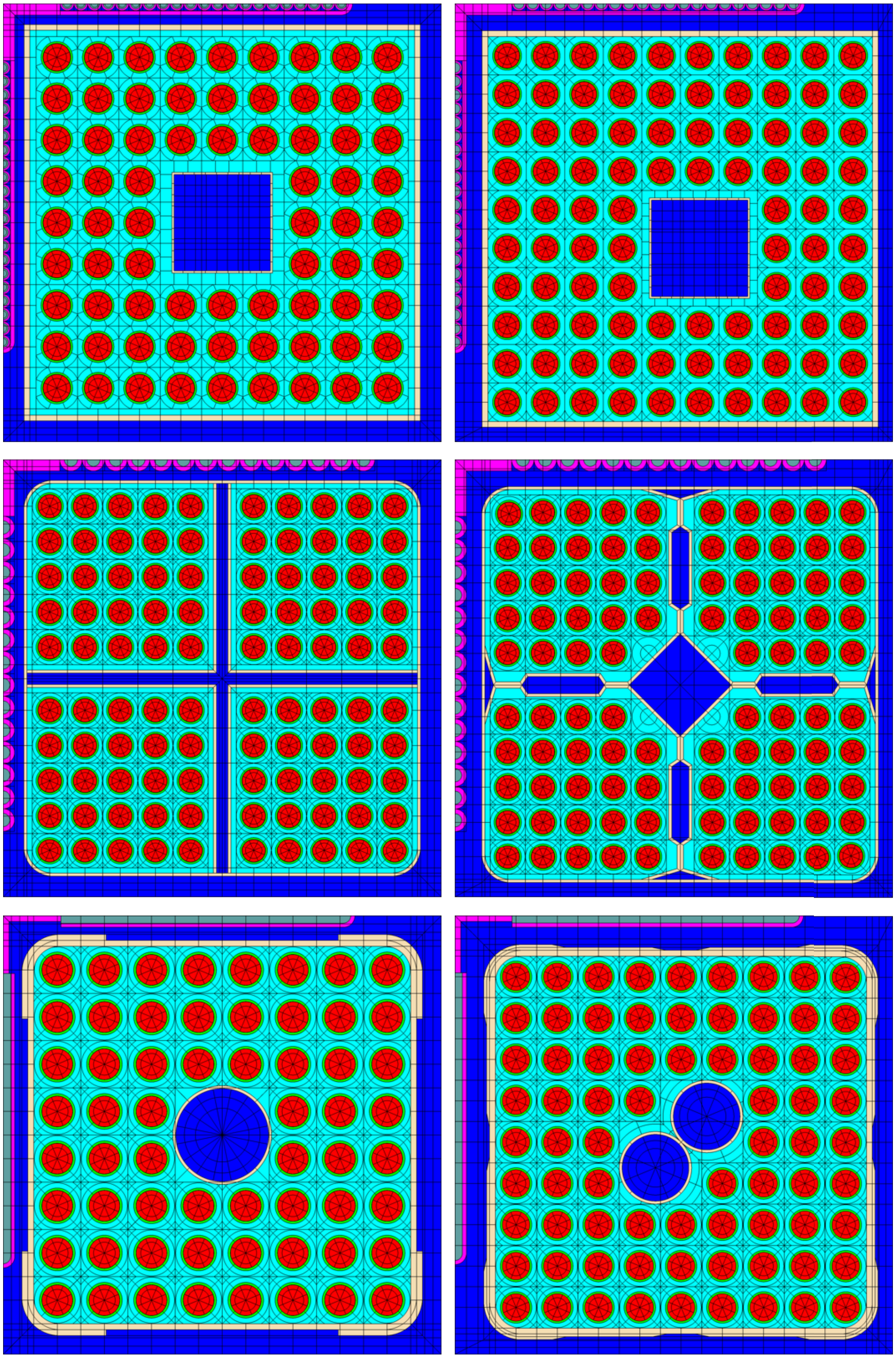

Fig. 1. BWR design examples. Top left: ATRIUM 9 × 9. Top right: ATRIUM 10 × 10. Middle left: SVEA-100, Middle right: SVEA-96. Bottom left: GE 8 × 8. Bottom right: GE 9 × 9. Lattices in the left column have uniform bypass gaps. Lattices in the right column have wide and narrow bypass gaps. The control blade inserts are Hatch OEM design, Marathon design, and uniform material in the top, middle, and bottom rows respectively.

```
=polaris_6.3
lib "broad_n"
sys BWR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% materials
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
comp c_uox : UOX 3.0
mat FUEL.1  : c_uox 10.2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% geometry
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
geom myBWR : ASSM 9 1.437
hgap 0.9
box  0.2 0.9 6.52 : 0.1 : 4.0 : 4.1
pin F : 0.475 0.4835 0.55 : FUEL.1 GAP.1 CLAD.1
pin W 3.0 : 1.68 1.75 : MOD.1 TUBE.1 : SQR SQR
pinmap
 F
 F F
 F F F
 F F F W
 F F F W W
 F F F W W W
 F F F F F F F
 F F F F F F F F
 F F F F F F F F F
mesh COOL.1 : nx=9 ny=9
mesh MOD.1  : nx=9 ny=9
mesh TUBE.1 : nx=9 ny=9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% state
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
state ALL   : temp=600
      FUEL  : temp=900
      MOD   : void=0
      COOL  : void=40
end
```

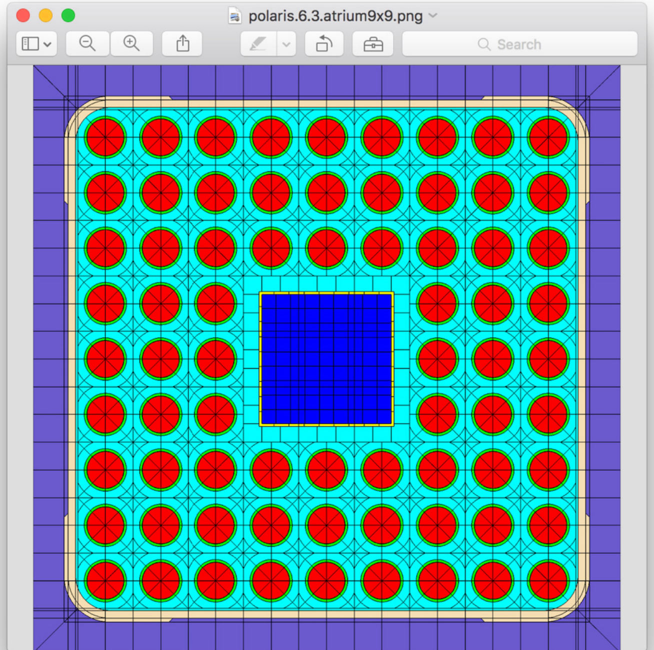

Fig. 2. Polaris input example.

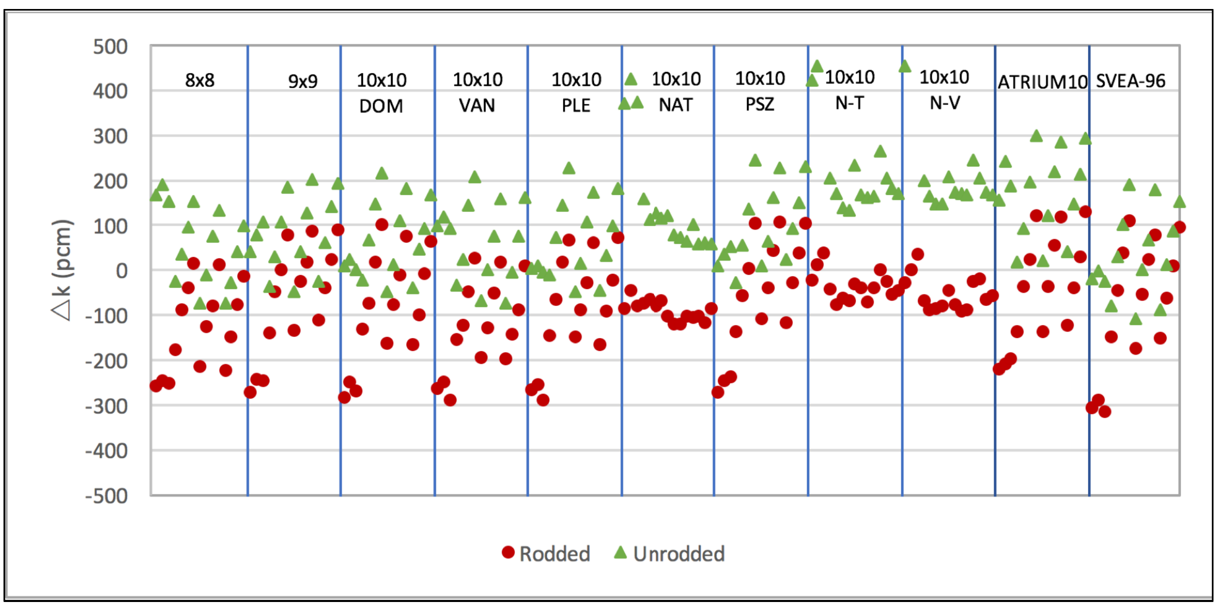

Fig. 3. k_∞ comparisons between CE KENO-VI and Polaris lattice calculations.