

Three-Dimensional Modeling of Complex Fusion Devices Using CAD-MCNPX Interface

Mengkuo Wang¹, Timothy J. Tautges², Douglass L. Henderson³, Laila El-Guebaly⁴, Xueren Wang⁵

¹ University of Wisconsin - Madison, 1500 Engineering Dr. Madison, WI, mengkuow@cae.wisc.edu

² Sandia National Laboratories, P.O Box 5800, Albuquerque, NM, tjtautg@sandia.gov

³ University of Wisconsin - Madison, 1500 Engineering Dr. Madison, WI, henderson@engr.wisc.edu

⁴ University of Wisconsin - Madison, 1500 Engineering Dr. Madison, WI, elguebaly@engr.wisc.edu

⁵ University of California – San Diego, 9500 Gilman Dr. La Jolla, CA, wang@fusion.ucsd.edu

MCNPX's [1] geometric modeling capabilities are limited to Boolean combinations of primitive geometric shapes. These capabilities are not sufficient for simulating particle transport in stellarators, whose geometric models are quite complex. We describe a CAD based implementation of MCNPX, where a CAD geometry engine is used directly for solid model representation and evaluation. The application of this code, to calculating the neutron wall loading distribution (I) in the Z and toroidal directions for the ARIES-CS[2] design, is described.

I. INTRODUCTION

For a commercial power plant fusion device, many engineering design concepts were evaluated and a design based on the compact stellarator (CS) concept has been recently developed by the ARIES team. A nuclear analysis is needed to obtain key neutronics design parameters, such as the neutron wall loading level, tritium breeding ratio, energy multiplication, and radiation damage to structural components. These parameters give guidance and recommendations on radiation protection for the toroidal field magnet, the size of a breeding blanket, and the selection of an optimal shield.

An accurate three-dimensional analysis requires a Monte Carlo simulation to estimate the overall design parameters. A Monte Carlo code that has seen widespread use at national laboratories and universities throughout the world for nuclear analysis is the MCNPX code developed by Los Alamos National Laboratory. However, the present version of the code only allows representation of bodies which can be constructed as Boolean combinations of a limited number of geometric primitives; these are difficult to use when constructing complex models, and are insufficient for representing the spline-based surfaces in the ARIES-CS design. If we use an inaccurate representation of the actual geometry, the accuracy of the Monte Carlo computational result will only provide an estimate of the true result.

Current CAD software focuses on the capabilities of geometry modeling. They have powerful abilities and

features and provide a user-friendly interface. They also provide functionalities that can evaluate the geometry such as the ray fire function and the surface area function. Interfacing to CAD directly allows the Monte Carlo transport code to take advantage of these capabilities, and any geometric models constructed with them.

II. METHODS

II.A. CAD based MCNPX

To improve the modeling capability of MCNPX, we incorporate a CAD geometry engine in the code. We use the Common Geometry Module (CGM)[3], which is based on ACIS, as the CAD geometry engine coupled with MCNPX 2.1.5, which is an extended version of MCNP[4]. Fig 1 depicts the structure of the MCNPX/CGM code.

The standard MCNPX reads in the combinatorial geometry as part of the problem initialization. During the Monte Carlo particle simulation, MCNPX determines the distance from a particle's present position to an intersection point on the boundary. The function that performs this operation is the ray object intersection function also called the ray fire function. In MCNPX/CGM, the CAD geometry and the CAD functions are initialized. In the Monte Carlo simulation, the CAD geometry engine's ray object intersection function is substituted in place of the standard ray object intersection function. With this substitution we obtain the ability to directly transport the particles through the CAD geometry. Note that this approach is not a conversion or translation from a CAD geometry model to MCNPX input but rather a direct simulation through the CAD model.

To use this code, users first construct geometry in a CAD system, and write that geometry to an ACIS file. The method currently used is to construct the geometry in CUBIT[5], a mesh generation toolkit providing advanced graphical interaction with geometric models. Alternatively, the geometry could be constructed in any CAD system able to export ACIS (e.g. SolidWorks), or export a model through the STEP standard for geometry exchange.

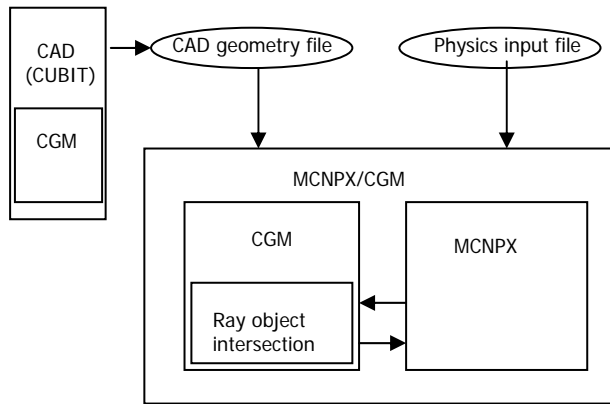


Fig. 1. Diagram of MCNPX/CGM

II.B. Advantage of CAD based MCNPX

Because complex models are usually designed with CAD systems, the CAD model of a complicated geometry usually exists before the Monte Carlo simulation is performed. Using a direct CAD interface saves the user effort required to construct that model using the standard MCNPX geometry input, which can be substantial for complicated designs. Furthermore, by eliminating the bottleneck going from CAD to MCNPX geometry input, we can take full advantage of the parametric design capabilities of modern CAD systems. In this way MCNPX calculations could be an integral part of the design process, rather than an a-posteriori tool for verifying the design.

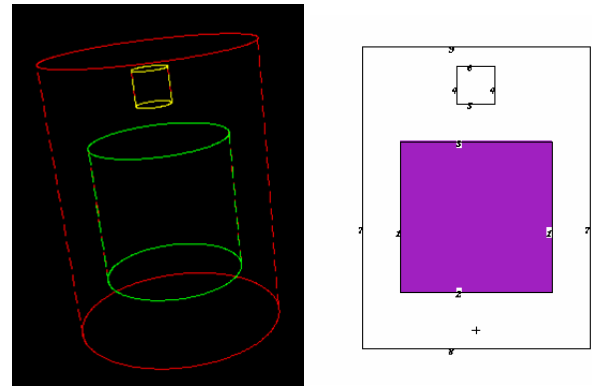
III. VALIDATION

As with any new program and code development, one of the most important exercises is validation. We have examined two cases to validate MCNPX/CGM which are described below.

III.A. Quantitative Validation (for simple geometry)

The first problem is a simple “three cylinder” nuclear analysis problem. We have an 11 MeV point neutron source that irradiates a cylindrical object from the bottom side of the object. A small cylindrical detector is located on the other side of the object to measure the gamma spectrum which is induced by interactions of neutrons and the media.

This simple case is modeled with both the standard MCNPX code and the MCNPX/CGM code. The scoring tally is the neutron flux (F1 tally) at three surfaces of the object. Fig 2 shows the geometry of this problem rendered by the standard MCNPX and the MCNPX/CGM code.



(a) Shown by CAD (b) Shown by MCNPX

Fig. 2. Plot of “three cylinder” problem

Figure 3 depicts the results of this comparison. Since only the geometry routines are different between the two codes and because no change to the sampling and physics functions were made, a particle in MCNPX/CGM will experience the exact same tracks and interactions as in the standard MCNPX. This means that if the development of MCNPX/CGM is done correctly, both codes should give the same result. This is indeed the case. The tally in MCNPX/CGM is exactly the same as for the standard MCNPX. The spectra curves from both codes coincide exactly.

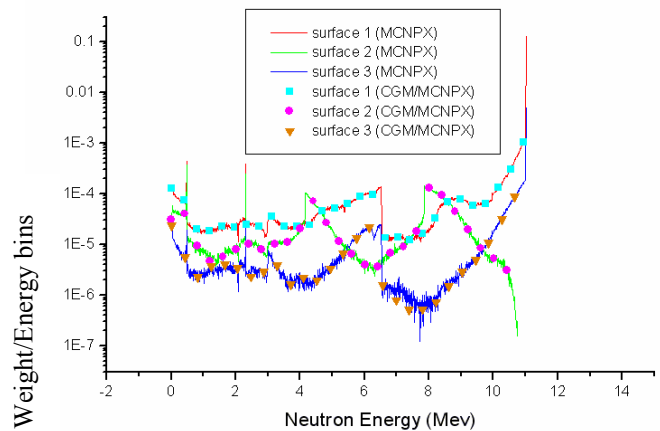


Fig. 3. Comparison of tally spectrum for the “three cylinder” problem

III.B. Visual Validation (for complicated geometry)

Fig 4 depicts a clothes pin with a match clinched between its jaws. The spring has been moved from its usual spot on purpose to better depict its complicated

geometry. Although it is not a real application, this example illustrates the modeling of complicated geometries.

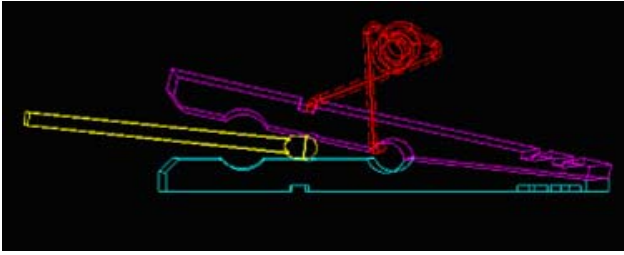


Fig. 4. Computational CAD model of clothes pin

This model would be very difficult to analyze with MCNPX alone because of the helical surface of the spring. A point gamma source is located under the paper plane and illuminates the clothes pin. The Fi5 (Pinhole image projection) tally is used to create an image of the clothes pin. Fig 5 depicts the resulting image. From this illustration we can see that MCNPX/CGM can be applied to complicated geometries.

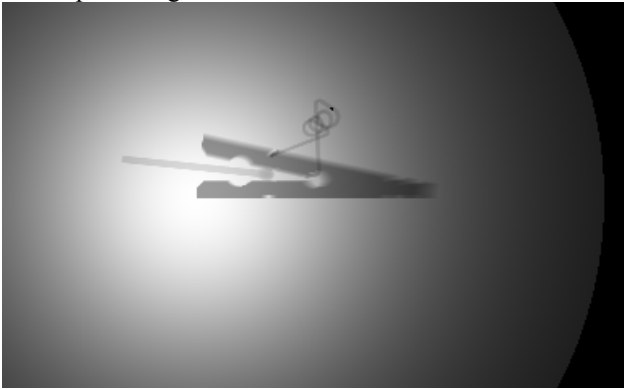


Fig. 5. Radiograph image of the clothes pin model simulated by MCNPX/CGM

IV. STELLARATOR SIMULATION

The first real application of the MCNPX/CGM is to calculate the neutron wall loading distribution (Γ) for the ARIES-CS in the poloidal and toroidal directions. The neutron source profile peaks at the geometric magnetic axis within the plasma region. The CAD model for the ARIES-CS is first generated in Pro/Engineering and converted to ACIS; Fig 6 shows this model.

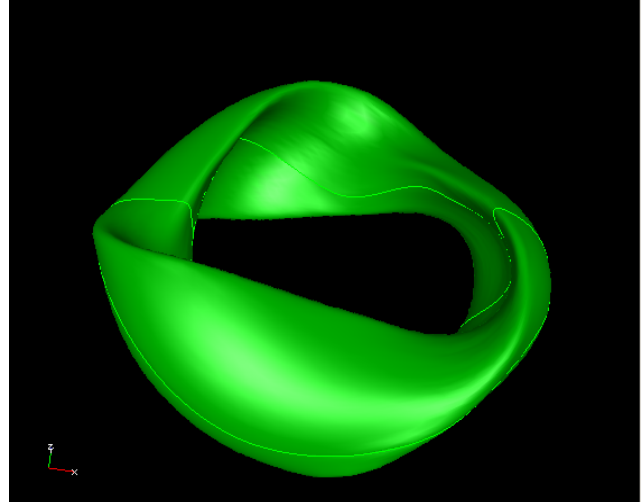


Fig. 6. The plasma region of ARIES-CS model

To construct the tally surfaces for the Monte Carlo simulation, we subdivide the plasma region into horizontal and toroidal directions. Each patch is a tally surface. The toroidal subdivision is 7.5 degree each and the horizontal subdivision is 0.5 m each. The first wall surface is only 5 cm above the plasma region. However, because we do not have a first wall model, for this simulation we use the plasma region as the tally surface. Fig 7 shows the subdivision.

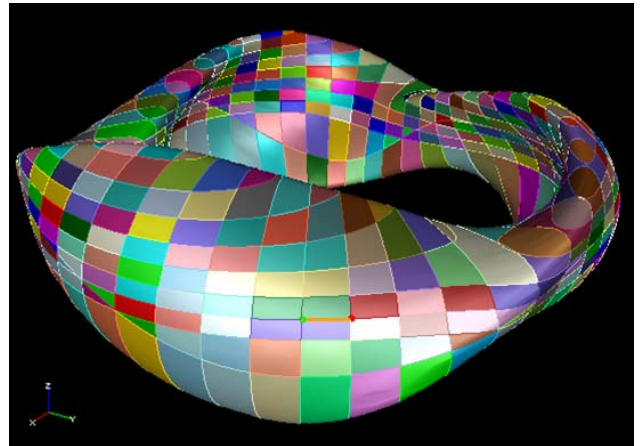


Fig. 7. The computational CAD model of the ARIES-CS

By symmetry of the ARIES-CS, we select the angular toroidal range from 0 to 60 degree. The other symmetric sections were combined with this section to construct the final tally.

Nine poloidal cross sections are provided for the toroidal positions in the angular range of 0 to 60 degrees and are depicted in Fig. 8. The inner curve is the plasma

surface. The outer curve is the magnet's winding pack center which will be added to our computation model in the future. The actual neutron source profile was used in the calculation [6].

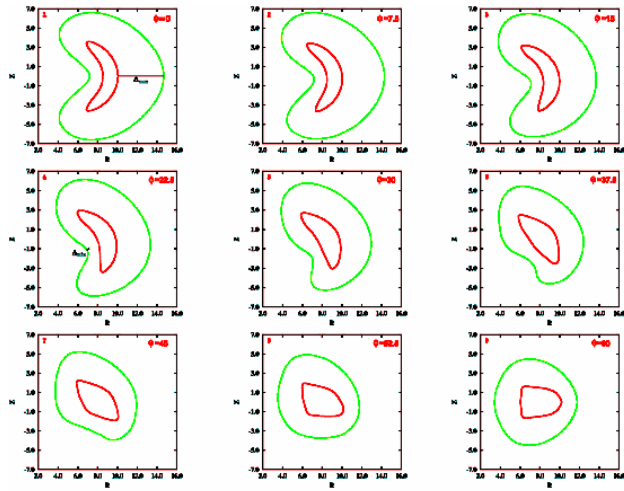


Fig. 8. The nine poloidal cross sections in the 0-60 degree toroidal cross section of the stellarator

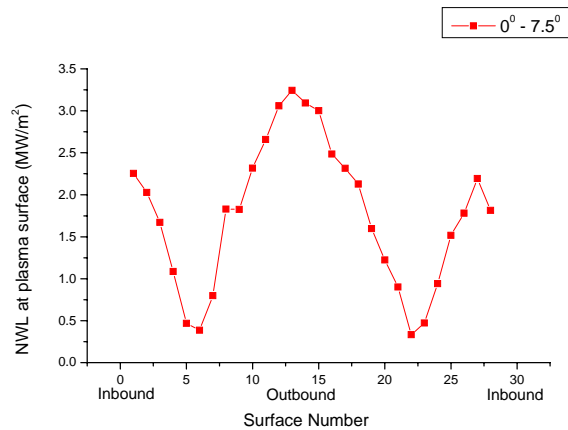


Fig. 9. The neutron wall load profile at 0 – 7.5 degree toroidal section

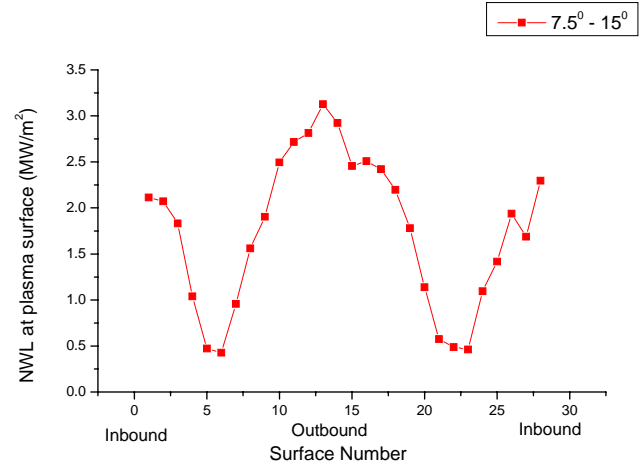


Fig. 10. The neutron wall load profile at 7.5 – 15 degree toroidal section

Figs 9 and 10 show the neutron wall loading profile at the 0-7.5 degree and 7.5-15 degree toroidal sections. We can see that the neutron wall load peak for each section is on the outboard side. To find the peak wall loading for the stellarator, we depict curves of outboard wall loadings at various toroidal positions. This is shown in figure 11. For 1600 MW of total neutron power, the average neutron wall loading is 1.985 MW/m² and the peak is 3.24 MW/m². The peak occurs at the 0-7.5 degree outboard section.

This Monte Carlo simulation was performed on a 2.4 GHz linux-based computer. The simulation time was 5 days with a relative error of about 9% ~ 10%. There are two reasons for the long run time of this simulation. The first is the source sampling routine, which is quite inefficient but can be improved. The second reason is related to the ray -fire and ray-object intersect routine and its implementation in the CAD software. We are currently investigating ways to speedup this important function. Currently, the MCNPX/CGM has a much lower computational speed than the standard MCNPX. Based on the first test problems, we estimate the speed of the MCNPX/CGM to be a factor of 10 slower than the standard MCNPX.

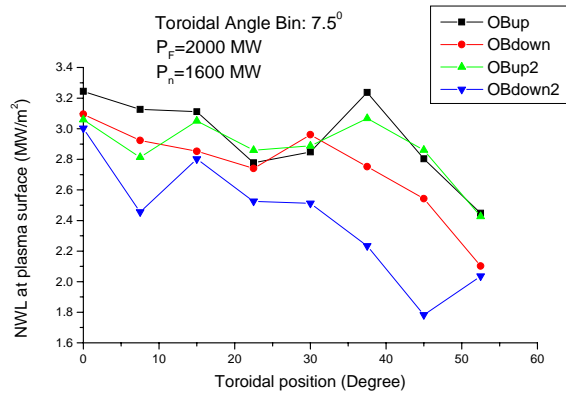


Fig. 11. Outboard side neutron wall loading

The results of this analysis will have a major impact on the ARIES-CS design. The poloidal/toroidal Γ distribution helps determine the exact size of the shield needed to protect the magnet and thus could solve a potential interference problem that has been identified for the field-period maintenance scheme when the blanket is moved toroidally out for replacement at the end of its service lifetime.

V. DISCUSSION AND CONCLUSION

CAD based MCNPX can be applied to complicated geometries. This increased geometric modeling capability can be important for radiation transport simulations in complex fusion devices, complicated shielding and reactor designs or complicated geometries in non-nuclear applications.

The MCNPX/CGM code is coupled directly to the CAD geometry (specifically the ACIS solid modeling engine). Several problems have been run in both MCNPX and MCNPX/CGM, and the results where compared have been identical. This limited testing verifies that the geometric computations in MCNPX/CGM are performing similarly to those in MCNPX. MCNPX/CGM has also been used to compute neutron wall loading for the ARIES-CS fusion device; this device cannot be modeled in MCNPX, due to the NURBS-based surfaces which describe the plasma boundary.

The MCNPX/CGM simulations are approximately a factor of 10 slower than the standard MCNPX. This is because the CAD functions are not optimized for a single function but in Monte Carlo simulations the geometry performance relies heavily on the “ray-object intersection” function.

In order to be an effective tool, ray-object intersection acceleration techniques must be used to improve the computational expense of MCNPX/CGM. That is in our future plan.

ACKNOWLEDGMENTS

We acknowledge the guidance, funding and other support of Charles Hills from Los Alamos National Laboratory. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

REFERENCES

- [1] Laurie S. Waters, Editor, “MCNPX USER’S MANUAL Version 2.1.5” Los Alamos National Laboratory TPO-E83-G-UG-X-00001 Revision 0 November 14, 1999
- [2] F. NAJMABADI, "Exploration of Compact Stellarators as Power Plants: Initial Results from ARIES-CS Study," Fusion Science & Technology, Nov 2004.
- [3] Timothy J. Tautges, “CGM: a Geometry Interface for Mesh Generation, Analysis and Other Applications”, Engineering with Computers, 17:299-314 (2001).
- [4] J.F. Briesmeister, Editor, “MCNP – A General Monte Carlo N-Particle Transport Code, Version 4B,” Los Alamos National Laboratory, LA-12625-M, March 1997
- [5] T. D. Blacker et al., 'CUBIT mesh generation environment, Vol. 1: User's manual', SAND94-1100, Sandia National Laboratories, Albuquerque, New Mexico, May 1994, http://endo.sandia.gov/cubit/release/doc-public/Cubit_UG-4.0.pdf
- [6] Private communication: Dr. Laila El-Guebaly: Fusion Technology Institute, University of Wisconsin – Madison, May 2004