

Development of McCad as an Integrated Interface Tool for the CAD to MC Geometry Conversion

Lei Lu, Yuefeng Qiu, Christian Wegmann, Ulrich Fischer

Karlsruhe Institute of Technology (KIT), Institute for Neutron Physics and Reactor Technology,
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

Abstract -*McCad is developed as an integrated interface tool at the Karlsruhe Institute of Technology (KIT) It enables the conversion of CAD models into representations for Monte Carlo (MC) particle transport simulations applied in design analyses of fusion and fission reactors. This paper introduces the significant algorithms and functions of McCad. In addition, newly developed interfaces and a new software architecture are presented as well. The current version of McCad has been applied to various modeling applications in fusion technology demonstrating its suitability for the conversion of large and complex geometry models such as the ITER device and the EU DEMO reactor.*

I. INTRODUCTION

Most Monte Carlo (MC) codes for particle transport simulations enable the use of complex three-dimensional (3D) geometric models by adopting Boolean forms of primitive solids and algebraic half-spaces as geometry representations, namely Constructive Solid Geometry (CSG) representation. Moreover, the Unstructured Mesh (UM) is also utilized by some MC codes as a new representation for complex geometries, e.g., MCNP6 [1]. However, creating an accurate MC model manually is a tedious, time-consuming and error-prone task, especially for models with complex geometries. In addition, validating and visualizing the results are also important requirements. Therefore, efficient algorithms and tools assisting MC modeling are strongly required.

Currently, Computer Aided Design (CAD) technology and a variety of CAD interface programmes offer the possibility for creating a complex geometry model. In addition, Computer Graphics is also a mature and widely used technology which supports the capability of visualizing complex graphics and geometry. Thus an effective solution is taking the advantages of CAD software and Computer Graphics to convert complex CAD model into MC geometry and visualize the simulation results.

McCad is such a programme that is a complete open-source interface tool developed at KIT. It is provided both on the Linux and Windows platforms adopting OpenCascade (OCC) [2] as CAD kernel and the Qt4 and OpenGL libraries for the graphical user interface (GUI). McCad is primarily designed to provide the conversion of CAD into geometry models for MC codes, such as MCNP, TRIPOLI and Geant4 [3-4]. In addition, related visualization capabilities are based on the coupling of McCad with the ParaView [5] which allows the user to overlay mesh tally distributions on the CAD geometry. Besides, new McCad interfaces are being developed based on Salome [6] and FreeCAD [7], which integrate significant functions and will be a

platform for assisting MC modeling and analysis work.

II. CONVERSION ALGORITHMS

The kernel algorithms of McCad are conversions of geometry representations, which translate a BREP (Boundary-Representation) model created by a CAD system to CSG representation for most MC codes, UM model for MCNP6 or Geometry Description Markup Language (GDML) file for Geant4.

1. Brep-CSG Conversion

The BREP-CSG conversion includes two significant algorithms: solid decomposition and void filling. The solid decomposition reads input CAD solids with complex geometries and split them into a collection of disjoint and simple convex solids, which can be represented by Boolean forms of primitive solids or algebraic half-spaces in MC codes. There are three steps involved: The first is to generate a sufficient boundary set. Given a solid in boundary representation, the boundary surfaces are traversed in order to determine the definability of the solid by the available boundary support set. If the solid includes curved surfaces, the boundary surfaces might not be sufficient for the final CSG representation, thus enlarging the boundary support set is required until the definability is achieved. In practice, a sufficient boundary set of a solid can be constructed by adding auxiliary surfaces which separate the curved surfaces entirely from the solid. The second step is to distinguish the splitting surfaces in the created boundary set. Here the triangle collision technique is employed. The boundary surfaces are meshed into a number of triangles, then the original calculation of the positional relationship between a boundary surface and the solid by Boolean operation is simplified into a simple numerical calculation which only needs to calculate the relative position of vertexes of triangles and boundary surfaces. Thus it improves the stability of algorithm, and moreover reduces the required storage space and consuming

CPU times. The third step is to sort the splitting surfaces, because the sequence of splitting surfaces used for the practical decomposition affects the result intensively. Therefore, the feature recognition technique and some significant rules are employed for calculating the weight of each splitting surface and generate an optimal sequence for performing the splitting [8]. Fig.1 shows an example of solid decomposition. An input solid is decomposed into a number of simple convex solids and then written into different representations according to the syntaxes of MC codes.

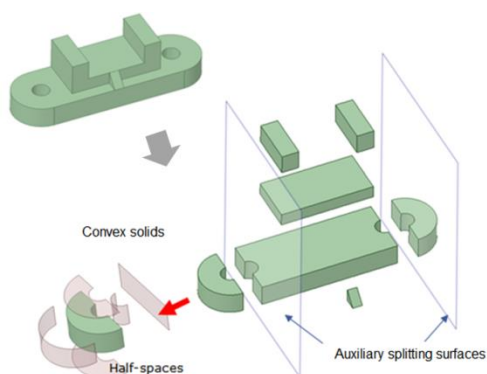


Fig. 1. BREP-CSG decomposition

The other significant algorithm of BREP-CSG conversion is the void filling, because MC geometry models need to define the whole space of the problem geometry including the void regions in and around the material solids. Such void space is usually not included in CAD models and must be created. The void filling algorithm of McCad splits the whole problem space into a number of boxes, and calculates the collided boundary surfaces of material solids with each box. Then the void spaces are described with the complements of these collided surfaces. The discrete sample points of boundary surfaces are employed here instead of Boolean operations thus making the void filling process more stable and time efficient [9].

2. Unstructured Mesh Creation

The traditional CSG approach is to adapt the models to simple geometries consisting of analytic surfaces. However, with the increase of model complexity, the CSG approach is getting more difficult and it is tedious to create the input geometry of such complex model, even with the automatic modeling tools. Therefore, some innovated approaches have been developed for finding a better way to construct geometries for the particle tracking. MCNP6 e. g. has been extended to include a new capability that permits tracking of neutrons and photons on an UM model. The mesh geometry could be created directly through Computer Aided Engineering (CAE) software using its solid modeling capabilities. And furthermore, with the new approach, the results from MCNP6

calculations can be easily exported for visualization or other physics analysis.

Therefore, a new function for converting the CAD model to MCNP6 UM geometry has been developed and integrated with McCad, which is based on fundamental functions of computer graphics which have been already intensively studied and utilized in CAE codes for discretizing the problem domains. However, the difference is that MC calculation requires describing the geometry accurately with as few cells as possible, because the number of cells influences the computing times intensively. A novel tessellation tetrahedralization meshing algorithm has been created for generating optimal UM for MC calculation [10]. It includes two steps: The first step is tessellation: an input CAD model is discretized with OCC/STL library and represented by a number of triangular facets. The second step is tetrahedralization: On the basis of the open-source meshing tool Tetgen [11], the triangular facets created in the first step are read, and then the UM are generated and then exported according to the syntax of MCNP6. Because the precision can be adjusted at the first step for controlling the differences between original and discrete models, the algorithm can guarantee a minimum difference without increasing the number of meshes (cells). Fig.2 shows a UM model created from ITER benchmark model by McCad's UM creation function.

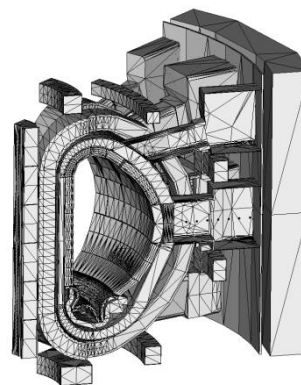


Fig. 2. UM model of the ITER benchmark model

In addition, the facets created at the first step can also be directly exported to the representation of GDML for another MC code like Geant4. A related function supporting the Geant4 modeling has been developed and integrated in McCad as well.

3. CSG-UM Hybrid Modeling

McCad supports the creation of traditional CSG and UM models. However, these two approaches have their own advantages and disadvantages. Normally, the model with medium geometry complexity can be easily represented with the CSG approach, and the computer run times can be also acceptable. Contrarily, the UM approach is

appropriate to describe a model with complex geometry or consisting of non-analytic surfaces, e.g., spline surfaces. Therefore, if there are models with detailed and complex inner structure and simple homogenous outer regions, a hybrid geometry approach can be employed. In this case, the UM approach is used to generate a mesh universe describing the inner structure and the CSG approach is used for describing the outer regions. Thus both advantages of these two representations are utilized.

Depending on the algorithms and functions of BREP-CSG conversion and UM creation, hybrid modeling function is also implemented and integrated into the McCad Salome version. The different parts of a model converted into UM and CSG can be converted separately and shared by the modeling module SMESH of the Salome platform. Then the mesh part of the model is exported to a MCNP6 mesh universe card, and the other solids are decomposed and translated into cells and surfaces cards of the CSG representation. Moreover, the information about the linking of UM and CSG is remained and final output for connecting the converted UM and CSG geometries.

III. VISUALIZATION

After the MC particle transport calculation, visualization is another highly required post-processing function for assisting the physicists for the subsequent analyses. Taking MCNP as example, versions MCNP5 and MCNP6 provided the mesh tally feature that enables the production of high resolution spatial distributions of nuclear responses during the simulation. However, MCNP restricts the visualization of such distributions to 2D representations overlaid to the MCNP geometry. Thus an interface code mt2vtk has been developed [12] which converts MCNP's mesh tally data into a file format of the Visualization Toolkit (VTK) [13]. This VTK file can be visualized in three-dimensional space by some graphics software, e.g., ParaView. In addition, if the original CAD geometry is converted into an STL file, it can be loaded by the visualisation software as well. Thus the mesh tally distributions can be overlaid to the relevant geometry and displayed together.

Recently, further improvements were added to mt2vtk to support different display modes: The unfiltered volume output displays the whole data with solid geometry; filtered volume output displays only the specified regions or components; surface output projects the data onto the given boundary surfaces; 2D cut output displays the data distribution on a given arbitrary cutting plane. Fig. 3 illustrates some of these new filter functions. Currently, mt2vtk is only an independent interface code. It will be integrated into the new McCad platform in future as a significant post-processing function.

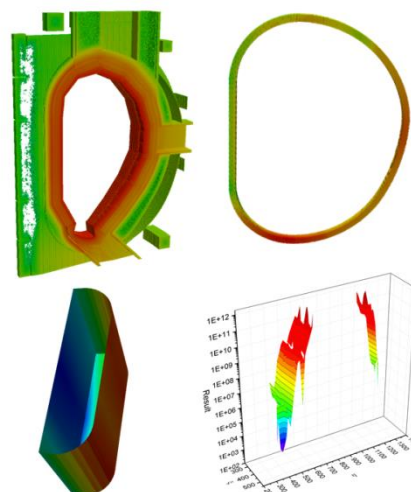


Fig. 3. Different visualization modes supplied by mt2vtk

IV. INTERFACE

The original McCad is developed based on the OpenCascade Application Framework (OCAF) and supplies a simple graphic user interface (GUI), including only the conversion functions and a material editor [8]. The new development of McCad abandoned the original GUI, and then extracted the main functions from the original codes and compiled them as independent dynamic libraries. These functions can be then integrated into other CAD systems which provide more flexible interactive graphic environments for geometry modeling, visualization and the coupling to multi-physics calculation tools. Currently, the Salome based McCad has already been developed and validated, and the user is able to complete a conversion process under this platform without using the original McCad GUI.

However, Salome is still a quite complex integrated system, which is not easy to install and run. It requires also a lot of experience about CAD/CAE systems. For this reason, a light version based on FreeCAD is currently being developed. It removes the redundant functions related to MC modeling. Furthermore, the kernel functions of McCad are compiled as modules and integrated into the FreeCAD main framework, such as the pre-processing functions including model checking, healing and simplification, CAD-MC conversion functions and other post-processing functions. The light version of McCad based on FreeCAD provides simple and clear interfaces and an easier way for the installation. It is also open source and can be compiled and run on both Linux and Windows platform. Fig.4 shows the GUI of the FreeCAD based McCad that is still under development. A preliminary version has already been released and uploaded to GitHub.

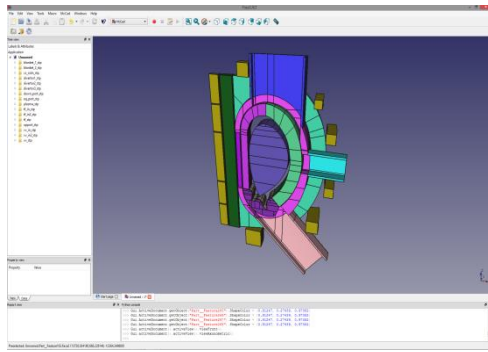


Fig. 4. GUI of the FreeCAD based McCad

V. SOFTWARE ARCHITECTURE

The future plan of McCad is to develop it as an integrated platform assisting the complete MC modeling, which includes the kernel CAD-MC conversion functions and other significant pre- and post-processing functions. Therefore, on the basis of the functions and new interfaces developed, a new software architecture of McCad has been constructed. All functions of McCad have been compiled as independent libraries which can be used as plug-ins and integrated into the main framework. The main frameworks currently are supplied by Salome and FreeCAD. In future, it could be extended to other CAD systems as well. The function modules can communicate with the main framework through defined interfaces. The new structure of the McCad code is shown in Fig. 5. Compared with the original design of McCad, the new structure is clear and each module is independent. Thus modification and updating is straightforward.

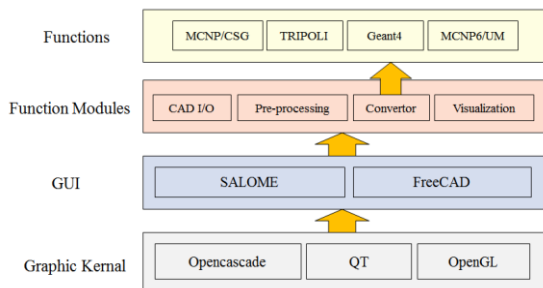


Fig.5 Software architecture of McCad

VI. APPLICATIONS

The new version of McCad including the improved functions and new interfaces has been successfully applied to different MC modeling and neutronics analysis tasks including e.g., various EU DEMO models performed in the frame of the European Power Plant Physics and Technology (PPPT), and some significant ITER tasks including the modeling of Neutral Beam Injection (NBI) system, upgraded blankets modules, Upper Launcher and In-Vessel Viewing System (IVVS) [14-17] etc. The following figures show such

significant applications. In these figures, the original CAD models are compared to the converted MCNP models (vertical cross section drawn with the MCNP plotter) for the geometry validation.

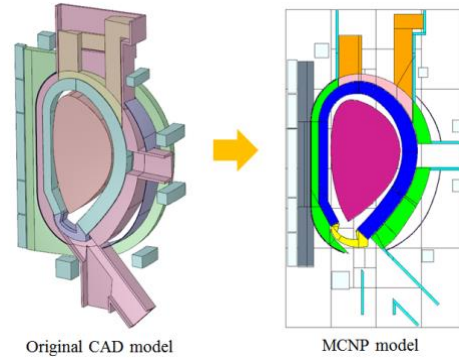


Fig. 6. EU DEMO model

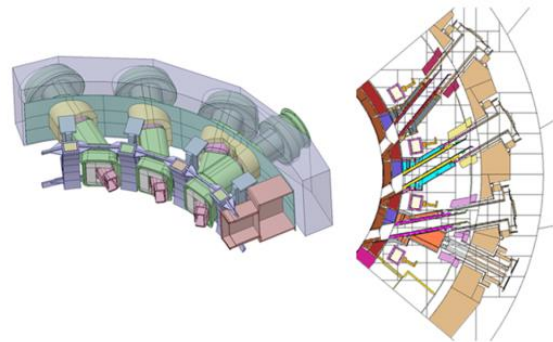


Fig. 7. ITER NBI model

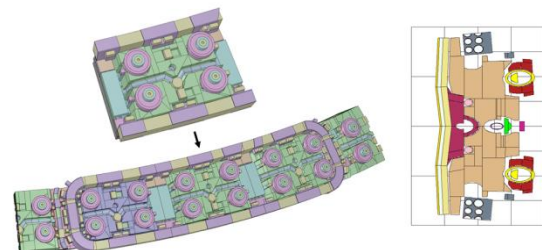


Fig. 8. ITER upgraded blanket modelling

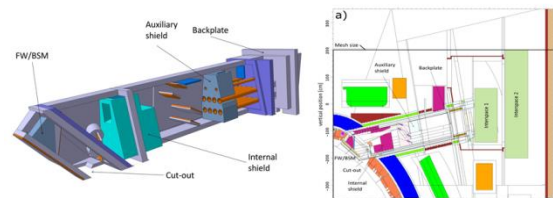


Fig. 9. ITER Upper Launcher

CONCLUSIONS

McCad is an open source geometry conversion tool developed to facilitate the MC neutronics modeling from CAD models. This paper presented the latest status of the McCad development including the improvements of Brep-CSG conversion algorithms, the newly developed UM

creation function and the visualization tool mt2vtk. Moreover, new interfaces based on Salome and FreeCAD were introduced as well. On the basis of these developments, the architecture of McCad has been optimized. In future, McCad will be developed as a special integrated platform for assisting the complete MC modeling and analysis work. The improved McCad software, including new/improved algorithms and interfaces, has been successfully applied to various important applications demonstrating its suitability for the conversion of large scale and complex models such as ITER and the EU DEMO reactor.

The next step of the McCad development will focus on two aspects: One is on the continued improvement of the efficiency of conversion algorithms and functions, the other one is on the development of a more convenient interactive environment to facilitate its use in practical applications.

ACKNOWLEDGMENTS

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed therein do not necessarily reflect those of the European Commission.

REFERENCES

1. J.F.BRIESMEISTER, MCNP a general Monte Carlo N-particle transport code, version 4C, Report LA-13709-M, Los Alamos National Laboratory, USA, 2000.
2. Opencascade, <http://www.opencascade.com/>, 2017.
3. J.C.NIMAL, T.VERGNAUD, TRIPOLI: A general Monte Carlo code, present state and future prospects. (1990) Progress in Nuclear Energy, 24 (1-3), pp. 195-200.
4. GEANT4 Collaboration, <http://geant4.cern.ch/>, 2017.
5. U.AYACHIT. The ParaView Guide: A Parallel Visualization Application, Kitware January 22, 2015

6. SALOME introduction, <http://www.salome-platform.org/>, 2017.
7. FreeCAD introduction, <http://www.freecadweb.org/>, 2017.
8. L.LU, Y.QIU, U.FISCHER, Improved Solid Decomposition Algorithms for the CAD-to-MC Conversion Tool McCad, Fusion Engineering and Design (in processing).
9. L.LU, U.FISCHER, etc., Improved algorithms and advanced features of the CAD to MC conversion tool McCad, Fusion Engineering and Design, Volume 89, Issues 9–10, October 2014, Pages 1885–1888.
10. Y.QIU, L.LU, U.FISCHER, An advanced MC modeling and multi-physics coupling system for fusion applications, M&C2015 Nashville, TN • April 19-23, 2015.
11. H.SI, TetGen, A quality tetrahedral mesh generator and three-dimensional delaunay triangulator, Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany, 2006.
12. D.GROSSE, U.FISCHER, etc., Status of the McCad geometry conversion tool and related visualization capabilities for 3D fusion neutronics calculations, F.E.D, Vol88, Issues 9–10, 2013.
13. W.J.SCHROEDER, L.S. AVILA; etc., Visualizing with VTK: a tutorial, IEEE Computer Graphics and Applications, Vol 20, Issue 5, 2000.
14. P.PERESLAVTSEV, etc., Neutronic analyses for the optimization of the advanced HCPB breeder blanket design for DEMO, Fusion Engineering and Design (in processing).
15. U.FISCHER, etc., Advanced Neutronics Simulation Tools and Data for Fusion Applications, IAEA FEC2016 conference, Kyoto, Japan, 17–22 October 2016.
16. B.WEINHORST, A.SERIKOV, Shut-down dose rate analyses for the ITER electron cyclotron-heating upper launcher, F.E.D, Volume 89, Issues 9–10, October 2014, Page 1899–1904.
17. A.TRAVLEEY, Y. QIU, Update of nuclear responses of the IVVS, Final report on the model development and nuclear analysis in the IVVS, F4E, July 2016.