Use of Tetrahedral Mesh Geometry to import a converted CAD file for Shielding and Criticality calculations with MONK and MCBEND

Thomas Barker (University of Birmingham)
Adam Bird, Roger Thetford (Serco)
Andrew.Cooper (Sellafield Ltd)

The ANSWERS Software Service, Serco, A32, Winfrith Technology Centre, Dorchester, Dorset, DT2 8DH, United Kingdom.
email: Adam.Bird@sercoassurance.com

Purpose *Reactor components (e.g. fuel assemblies) and other equipment (e.g. transport flasks) are designed using Computer Aided Design (CAD) packages. It is useful to be able to directly use these geometries in Monte Carlo calculations; Monte Carlo tracking using these geometries requires processing of complex surfaces which may slow the calculation. Here we describe a simplified approach of converting the CAD file into an Ansys ASCII-format tetrahedral mesh, and importing this into ANSWERS codes. A key advantage of this approach is that it can use the same finite element model as other analysis codes (e.g. stress analysis).*

Methods and Materials: *A new development has enabled multiple mesh files to be used freely in conjunction with all other geometry capabilities of MONK and MCBEND. The meshing program assigns 'body' names to groups of meshes which become material names in MONK and MCBEND allowing simple cross reference to material compositions. We report recent work on optimization and testing.*

Results: *Test calculations agree with models created using the standard geometry capabilities of MONK and MCBEND. Large mesh files, 1 – 2 million tetrahedra, can increase the initial load time by up to 10 minutes and require more memory than a standard calculation. Run times are less dependent on mesh size and can be the same as a standard calculation.*

Conclusions: *A route for importing CAD files, via a meshing program, into the Monte Carlo codes MONK and MCBEND is described. The route gives accurate results, is simple to use and requires little additional processing power.*

I. INTRODUCTION

Reactor components (e.g. fuel assemblies) and other equipment (e.g. transport flasks) are designed using Computer Aided Design (CAD) packages. It is useful to be able to directly use these geometries in Monte Carlo calculations. The root of the problem is as follows. MCBEND[1] and MONK[2], in common with other Monte Carlo programs, use Constructive Solid Geometry (CSG) a geometry system based on simple geometric bodies, for example cuboids, cylinders, ellipses and tori, whereas CAD programs generally use Boundary Representation (B-rep) consisting of higher-order shapes, typically represented as Non Uniform Ration B-Spline (NURBS) surfaces. There is no general method of accurately converting a B-rep model into a CSG model.

There are two broad approaches to this problem. One is to use the CAD package itself to perform the particle tracking and form a link between the Monte Carlo code and the CAD package. This can tie the system to a particular CAD package and be detrimental to the performance of the Monte Carlo calculation. A second method is to implement NURBS geometry tracking routines in the Monte Carlo program. This is a non-trivial exercise and is the subject of current work under the Nuclear Code Development partnership (NCD) between Serco and Sellafield Ltd (Ref.3).

The development described here is the result of looking for a simplified approach to the problem of tracking through CAD geometries in Monte Carlo codes.

II. OVERVIEW OF METHOD

The approach chosen is to convert the CAD model into a format that is easier to track in Monte Carlo codes. The format chosen is a tetrahedral mesh representation. There is a considerable amount of software available on the market for converting CAD models into meshes of tetrahedra. Serco has used the ICEM-CFD workbench to produce meshes. ICEM-CFD is part of the CFX modelling package, which has powerful meshing capabilities including functions for cleaning 'dirty' or 'bad' CAD.

II.A. The tracking process

Once the mesh model has been read, the requirements of MONK and MCBEND are simply to be able to query the mesh and find the material that is present at a point **p**. Rather than needing to calculate distances to material boundaries, which is the usual requirement of particle tracking, a process called Woodcock tracking is employed. The basis of Woodcock tracking is to give the geometry an artificial mean free path equal to the shortest mean free path of any material within it. Thus there is no requirement to find at what distance along the current track the material cross section will change. Finding the tetrahedron that contains point **p** is relatively straightforward. For each of its four faces, the dot product of **p** and the normal to the face must have the same sign as the dot product of the normal and one of the edges that does not bound the face. The search process simply scans a list of candidate tetrahedra until one is found that fits the criterion.

II.B. Mapping tetrahedra to material compositions

Any CAD import system for Monte Carlo has to determine how to reconcile a position in the model with the correct material composition. The ANSYS meshing package will assign a name to groups of tetrahedra; this name might be associated with the original entity in the CAD model or created by the user during the meshing process. In MCBEND and MONK these names in the mesh file become 'alpha-numeric' named materials in the MONK or MCBEND input file.

II.C. Optimisations

Despite being a simple calculation, searching millions of tetrahedra every time a point is sampled is prohibitively slow. To optimise the calculation the volume containing the mesh is divided into voxels, which are equal-sized cubes aligned with the model axis. First the voxel containing point **p** is determined then the tetrahedra that overlap that voxel are tested to determine which tetrahedron and hence which material is at point **p**. This greatly reduces the number of tetrahedra to search but requires a setup stage to construct the list of the tetrahedra that overlap each voxel.

III. INITIAL RESULTS

Calculations were performed on a Dell Optiplex GX520 PC with Intel Pentium Dual CPU 2.8Ghz processor and 3.49 GB of RAM.

The test model was a stainless steel and lead flask typically used for transporting gamma-ray sources. For the Criticality tests, using MONK, a uranium fissile source was used. This involved a simplification of the source region, replacing sources and support structure with smeared Uranium. A mesh containing 58,702 tetrahedra was created. The MONK calculations run to a requested standard deviation so comparison of run times indicates the relative performance of the tracking systems. The results showed a good agreement with the equivalent MONK/MCBEND geometry (FG) case; however the run times were considerably different. The tetrahedral mesh case took 11 minutes to set up and 5 minutes to run, whereas the FG case took 2 seconds in total.

For the shielding tests, using MCBEND, a smeared gamma-ray source was used. Energy deposition was scored at locations through the flask and gamma-ray dose was scored outside the flask. The mesh and FG models used the same Monte Carlo acceleration so comparing the samples/second is from each case is a meaningful indication of the performance. For the mesh case the setup time was the same as MONK. Again the calculational results from the mesh and FG cases compared well with the largest differences seen near the boundary of the smeared source and the lead shield, this is understandable given the approximation to the surface that that was achieved using this relatively coarse tetrahedral mesh. The difference in efficiency of the calculations was not as large as the MONK case because MCBEND is doing more acceleration work transporting gamma-rays through the lead shield so differences in the tracking performance have less impact on the overall calculation. The FG case achieved 22,700 samples/ second against the mesh case that gave 14,607.

Further tests using a more detailed mesh of 120,000 tetrahedra took 40 minutes to setup and ran 3,800 samples/second and going to a 1 million tetrahedra mesh the set did not complete in 48 hours.

The conclusions were that the method had potential because the simple calculations worked but the performance needed to be improved before it would be of practical use. Note that it is notoriously hard to construct a good tetrahedral mesh for thin curved shells. Bodies with a different shape — particularly if the faces were flat — would be likely to give good results with fewer tetrahedra.

IV. FURTHER WORK ON OPTIMISATION

Many methods of optimisation were tried. The successful ones are listed here.

- Improved memory management enabling a re-design of the search routine that constructs lists of tetrahedra in each voxel.
- Use of the Separation of Axis Theorem to optimise the detection of a tetrahedron overlapping a voxel.
- Pre-process the surface normals for each tetrahedron.
- Order the tetrahedra in each voxel by the overlap volume so the largest volume is tested first.
- Provide user control of the 'Voxel Ratio', this is a value that defines how many average volume tetrahedra will fit into the volume of 1 voxel. It defines the size of the voxel grid.

- Allow a model to be constructed from multiple tetrahedral mesh files. This means that areas of detail the will be represented by many small tetrahedra will have their own voxel grid that is more suitable to the detail of the mesh.

V. RESULTS

The changes mentioned in the last section vastly improved the efficiency of the calculation. The setup times are now only the order of a few minutes even for large meshes containing two million tetrahedra and the calculation is typically 25% slower than the FG equivalent. The voxel ratio was found to have a large impact on the run time, with smaller values giving a faster run time but increasing the memory requirements.

The test calculations were performed on a Dell Latitude D830 PC with Intel Core2 Duo CPU 2.6Ghz processor and 3.5 GB of RAM.

Currently the new system has only been tested for shielding using MCBEND. The same test calculation was used except this time the gamma-ray source rods and the supports were modelled explicitly. Figure 1 shows a picture if the FG model generated by the tracking routines using Visual Workshop[4]. Three mesh files containing 120,957 (mesh 1) 919,386 (mesh 2) and 2,052,067 (mesh 3) tetrahedra were used. Figure 2 shows images of the FG and mesh models showing the relative detail of the source basket. Figure 3 indicates the location of the scoring regions.

Graph 1 shows the agreement between the FG model and the mesh models for the energy deposition results. The average C/E for the 3 mesh models are 0.932, 0.928 and 1.07 and variances typically between 1% and 5% for the four calculations.

Graph 2 shows the agreement between the FG model and the mesh models for the gamma-ray dose results. The average C/E for the 3 mesh models are 0.970, 0.941 and 0.949 and variances typically between 0.5% and 2% for the four calculations. There is a notable difference between the mesh calculations and FG near the top of the flask.

The approximations introduced by meshing appear to only have a small effect on the result, although for this particular calculation using a more detailed mesh has made little difference.

More importantly from the point of view of this trial is the performance of the calculations. The FG model achieved 41443 samples/s, the mesh models achieved 38,158, 28,032, 32,314 samples/second respectively. These calculations used the default Voxel Ratio of 0.25. It should be noted that the calculation speed is no longer related to the size of the mesh, it is instead related to the Voxel Ratio. Table 1 shows the affect of changing the Voxel Ratio on the calculation for mesh 2. Graph 3 shows the setup time, memory use and calculation speed on a normalised scale. It is clearly seen that successively halving the Voxel Ratio gives an approximately linear increase in speed, with an exponential increase in memory use and setup time.

VI. CONCLUSION

A route for importing CAD files, via a meshing program, into the Monte Carlo codes MONK and MCBEND is described. The route gives accurate results, is simple to use and requires little additional processing power.

The next stage of work is to gain experience on the sensitivity of calculations to the approximation of representing the model as a tetrahedral mesh. The tetrahedral mesh import facility is currently available in BETA versions of MONK and MCBEND.

ACKNOWLEDGMENTS

The authors wish to acknowledge the other members of the MCBEND & MONK development team: Ted Shuttleworth, Malcolm Armishaw, Geoff Dobson, Christopher Dean, Ray Perry, (all Serco), David Bendall, Simon Connolly, Richard Neal and Keith Searson (all Sellafield Ltd).

REFERENCES

1. P. COWAN, E. SHUTTLEWORTH, A. BIRD and A. COOPER, "The Launch of MCBEND 10", *Proc. 10th International Conference on Radiation Shielding (ICRS-10) and 13th Topical Meeting on Radiation Protection and Shielding (RPS-2004),* Funchal, Madeira Island, Portugal (May 2004)

2. M. J. ARMISHAW and A. J. COOPER, "Current Status and Future Direction of the MONK Software Package", *Proc. 8th International Conference on Nuclear Criticality Safety*, St.Petersburg, Russia, May 28 – June 1 2007, (Vol II p264)

3. K. SEARSON, R. NEAL and A. J. COOPER, “Importing CAD data for Shielding and Criticality calculations with MONK and MCBEND”. *Proc. 11th International Conference on Radiation Shielding (ICRS-11) and 14th Topical Meeting on Radiation Protection and Shielding (RPS-2008),* Georgia, USA, April 13-18 2008

4. D. DEWAR, A. J. COOPER, A. J. BIRD and P. COWAN, “Visual Workshop as the single model viewer for the ANSWERS Shielding and Criticality Codes”. *Proc. 11th International Conference on Radiation Shielding (ICRS-11) and 14th Topical Meeting on Radiation Protection and Shielding (RPS-2008),* Georgia, USA, April 13-18 2008

Figure 1. The FG model

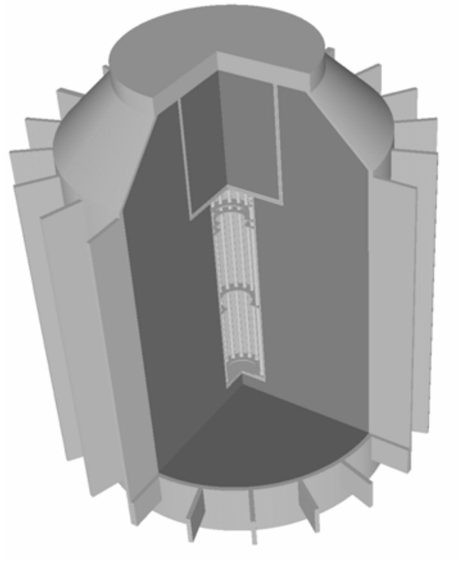
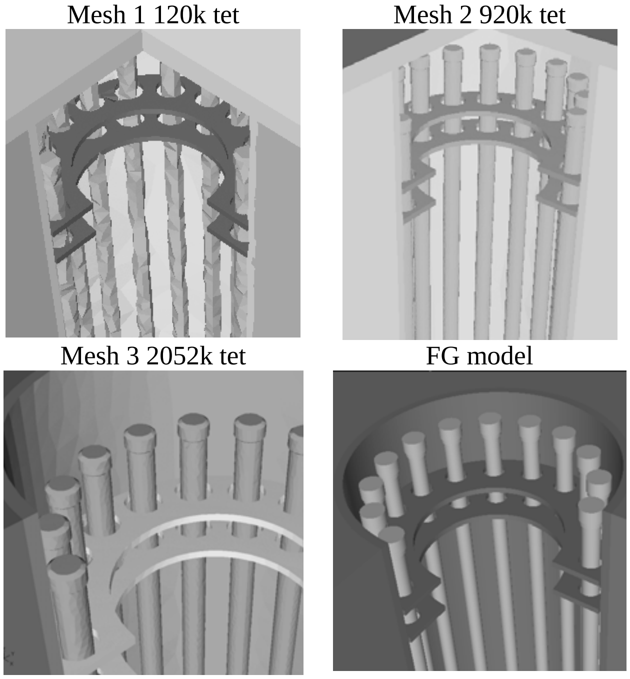


Figure 2 Representations of the geometry.



Mesh 1 120k tet

Mesh 2 920k tet

Mesh 3 2052k tet

FG model

Figure 3 Location of the scoring regions.

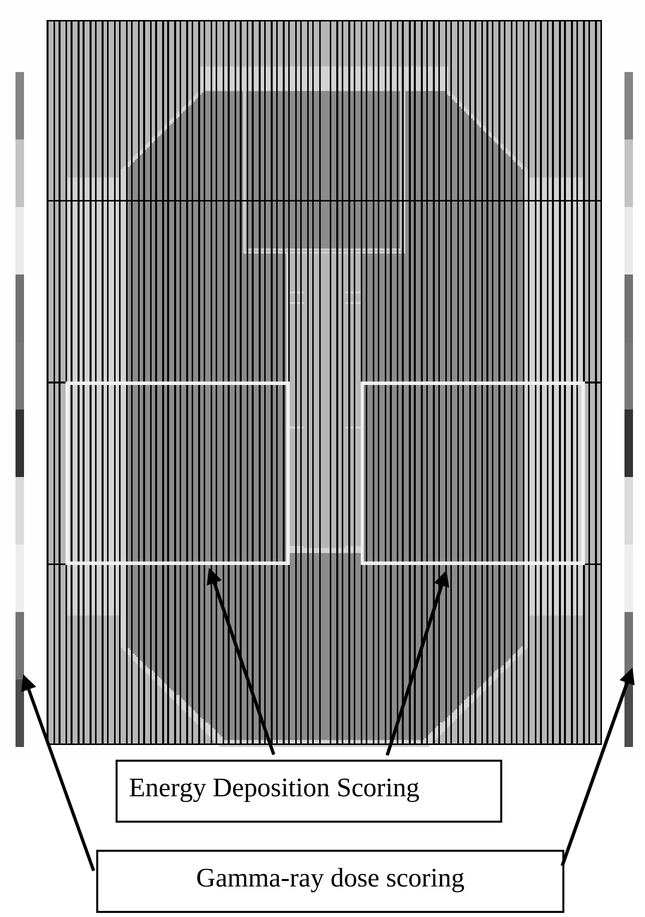
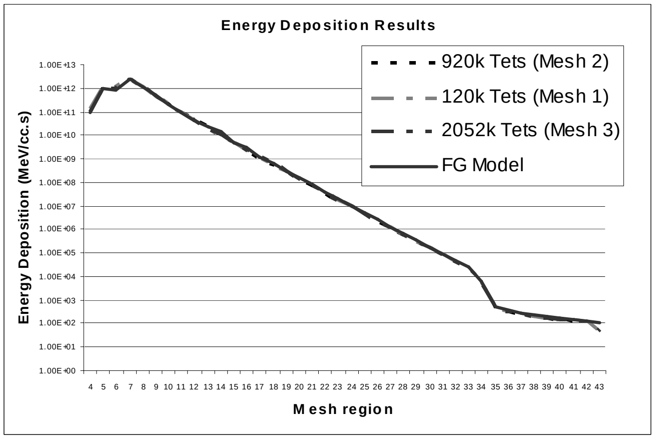


Table 1. Effect of the Voxel Ration on the efficiency of the calculation.

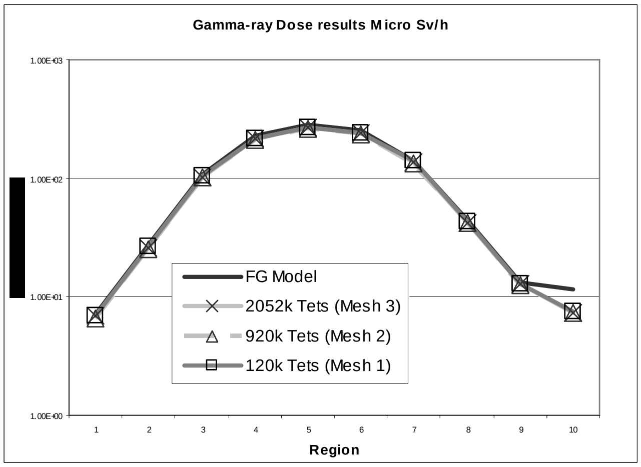
Voxel Ratio	0.03125	0.0625	0.125	0.25	0.5	1	2	4	FG
Set Up Time (s)	88	51	31	20	13	9	7	5	0
Samples/s	40081	37090	32422	28006	22154	16252	12285	8658	41443
Memory use (Mb)	1636	893	514	320	221	171	144	131	21

.

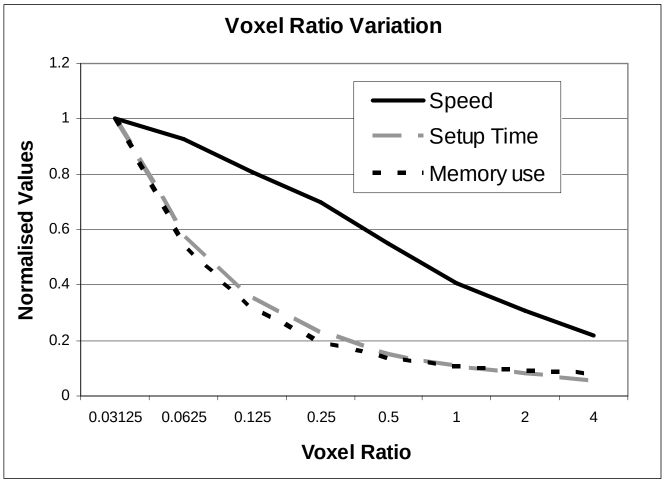
Graph 1 Energy deposition results.



Graph 2 Gamma-Ray dose results



Graph 3 Affect of Voxel Ratio on Speed, Memory use and Setup time.


Voxel Ratio Variation
Speed
Setup Time
Memory use
Normalised Values
Voxel Ratio