ANSWERS Conference Paper

Title:

GEOMETRY MODELLING AND VISUALISATION FOR THE MONTE CARLO CODE MCBEND

Authors:

Nigel Smith,	Ted Shuttleworth	Steve Carter	Mike Evans
AEA Technology	AEA Technology	AEA Technology	British Nuclear Fuels plc
Winfrith	Winfrith	Winfrith	Risley
Dorchester	Dorchester	Dorchester	Warrington
Dorset	Dorset	Dorset	Cheshire
DT2 8DH	DT2 8DH	DT2 8DH	WA3 6AS
United Kingdom	United Kingdom	United Kingdom	United Kingdom
+44 (0)305 251888	+44 (0)305 251888	+44 (0)305 251888	+44 (0) 0925 832000

Date:

January 1994

Conference:

8th International Conference on Radiation Shielding
Arlington, Texas, USA
25 - 28 April 1994

GEOMETRY MODELLING AND VISUALISATION FOR THE MONTE CARLO CODE MCBEND

Nigel Smith,	Ted Shuttleworth	Steve Carter	Mike Evans
AEA Technology	AEA Technology	AEA Technology	British Nuclear Fuels plc
Winfrith	Winfrith	Winfrith	Risley
Dorchester	Dorchester	Dorchester	Warrington
Dorset	Dorset	Dorset	Cheshire
DT2 8DH	DT2 8DH	DT2 8DH	WA3 6AS
United Kingdom	United Kingdom	United Kingdom	United Kingdom
+44 (0)305 251888	+44 (0)305 251888	+44 (0)305 251888	+44 (0) 0925 832000

ABSTRACT

In 1990, an ugrade programme for the major UK Monte Carlo particle tracking codes was initiated. Part of this programme comprised the production of a new geometry modelling scheme (subsequently called Fractal Geometry or FG) which is now included in the latest version of MCBEND. Another part of the programme has seen the production of the VISTA interactive geometry visualisation software package to support the powerful and versatile new geometry modelling capability. This paper describes the new MCBEND geometry modelling package and the associated visualisation software; the work described was performed as part of an AEA/BNFL software development collaboration.

I. INTRODUCTION

The use of the Monte Carlo method for solving particle transport problems is already well-established and will continue to grow with the availability of faster and cheaper computing facilities. One of the method's great strengths is the accuracy with which the geometry of a system can be represented. This inherent strength is exploited by Monte Carlo computer codes in a variety of ways to arrive at a sub-division of the problem space into volumes (or zones) of constant composition. There is no real theoretical limit to the geometry modelling capabilities which a Monte Carlo code could provide; the practical limitations however are set by the range of facilities provided by the code developer and the ingenuity of the code user.

During the simulation of particle transport in a Monte Carlo calculation, the position of the particle is monitored with reference to the boundaries used to describe the geometry. This can be achieved by determining which side of a given surface the particle is, or evaluating the point(s) of intersection between the particle track and a given surface. This in turn allows the code to identify the defined volume of space in which the particle is travelling and hence, the composition of that volume.

These principles, used in all major Monte Carlo codes, may be implemented in a variety of ways. The ideal form would be efficient on computer time and storage, robust in long calculations for complicated cases, and user-friendly at the input stage. The advantages and disadvantages of various options are discussed in this paper leading up to the description of a new system that is employed in the latest version of the general particle transport code MCBEND [1] and its sister code for criticality applications MONK [2]. The new system is called Fractal Geometry and it has been designed to achieve a reasonable balance between the partially conflicting requirements identified above.

In addition to a powerful and versatile geometric modelling capability, it is essential that a Monte Carlo particle transport code has a visual means of checking the specified geometry. Traditionally this has been performed by examining low-resolution two-dimensional slices through the problem geometry. However to accompany the Fractal Geometry package, interactive geometry visualisation software packages have been produced which provide a quicker and more accurate means of verifying the geometry model. These packages provide interactive high-resolution facilities within a modern and portable computing environment and enable two- and three-dimensional images of the geometry model to be displayed and manipulated. This paper also describes the development of these packages.

II. BASIC METHODS

The first stage of the development of Fractal Geometry comprised a review of available options in order to arrive at an optimised requirements specification. This section describes the various basic methods of geometry modelling that are used in major Monte Carlo particle tracking codes.

A. Surface Systems

In surface systems, zone boundaries are specified as mathematical surfaces. The requirements of efficient processing usually limit the surface repertoire to those defined by low order equations such as planes, cylinders, spheres, cones and paraboloids. The surfaces are numbered and a sign convention is typically used to indicate the volumes of space on either side of a given surface.

A zone is defined by a list of signed boundary surfaces. By introducing operators, the system can be extended to include the unions, intersections and complements of the spaces on selected sides of the surfaces. This is the basis of the geometry modelling system used in MCNP [3]. Figure 1 shows a simple application of this system in two dimensions. The shaded zone is defined by the intersection of infinite half-spaces each defined by a plane surface.

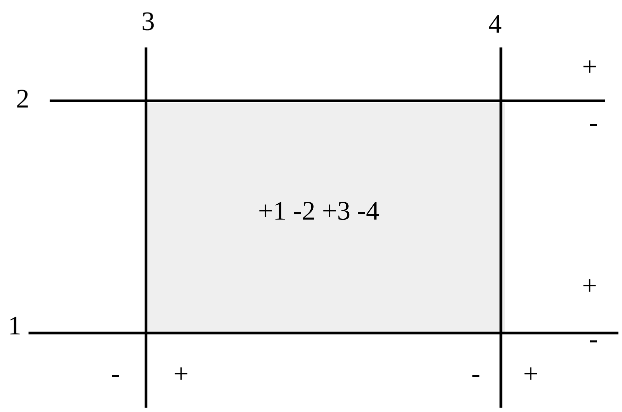

Figure 1 - Example of Surface system

B. Body Systems

In body systems, zone boundaries are specified by the surfaces of simple, solid bodies such as cuboids, cylinders, spheres, truncated cones and arbitrary polyhedra. The bodies are numbered and a sign convention is employed to indicate the volumes of space inside and outside of the bodies. A zone is defined by the intersections and differences of selected bodies. This is the basis of the geometry modelling system used in MCBEND and MORSE [4]. Figure 2 shows (in two dimensions) ways in which two bodies may be combined.

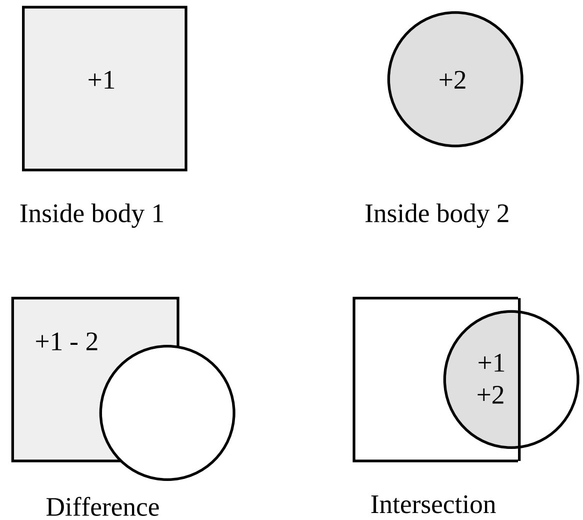

Figure 2 - Example of body Systems

C. Part Systems

The criticality code MONK contains a variation on the body system in which the model is subdivided into *parts*. Each part is a self-contained sub-system with its own local co-ordinate system. The outer surface of a part is referred to as the part *container* body. Any body, in any part, may contain a *subsidiary* part provided that the container of the subsidiary part matches the shape and size of the *parent* body. This system is specifically tailored to the repeated structures common in criticality applications.

Consider the example of a simplified flask model (see Figure 3). A fuel pin (with cladding, fuel pellets and end caps) may be defined as a part. A fuel element is a part which includes the fuel pins and an absorber pin as subsidiary parts. In this simple example, each part (E, P and A) is defined just once but included in parent parts as often as required. Common requirements are met by simple structured parts and a one-to-one correspondence between bodies and zones is maintained; this serves to simplify the user image. The structure and the local co-ordinate systems are retained into the execution stage of the calculation.

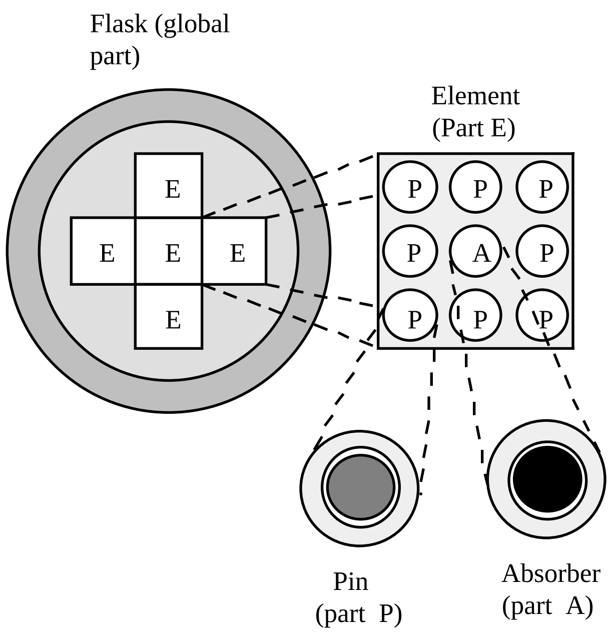

Figure 3 - Example of Part Systems

D. Woodcock Tracking

In the systems studied above, the movement of a particle is usually monitored by calculating the intersection points of the particle track with the surfaces or bodies of the defined system. This in turn involves solving equations involving the boundary surface parameters and those of the particle track. For efficient computation this limits the equations to be solved to first or second order - though many codes include a torus body with its associated quartic equation.

An alternative form of tracking (called Woodcock tracking) is available in MONK. A uniform, total cross-section is used within each special region of geometry. Its value, $\sum_t$, is the maximum (at the current particle energy) of all the materials in the region. When a collision occurs, the material at the collision site and its total cross-section, $\sum_m$, are identified. If a random number R ($0<R<1$) is less than the ratio $\sum_m/\sum_t$ then the collision is considered to be a *real* collision; otherwise it is a *null* collision and the track continues undisturbed. By this means, although the frequency of collisions is increased, the distribution of real collisions is unbiased.

Locating the material at the collision site involves testing the signs of a series of functions *evaluated* at the current particle position. This is far faster than *solving* the equations and hence there is no real limit to the order of the functions that can be used to define the geometry. In this method of tracking it is possible to model such structures as a screw feeder (see Figure 4) or a pump impeller with spiral blades. It is also efficient when modelling fine detail such as large arrays of fuel pins.

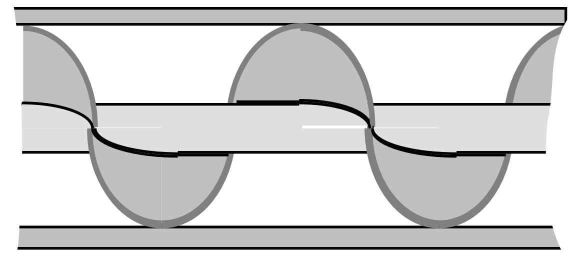

Figure 4 - Example of Woodcock Tracking

III. ADVANTAGES AND DISADVANTAGES

A. Bodies versus Surfaces

From the code user's point of view, a body system is generally easier to employ than a surface system since most physical zones to be modelled are solid engineering objects. The quantity of data necessary to define models is similar in both systems but the zone descriptions are generally much simpler to construct in a body system.

Take, for example, the containing material of a closed rectangular tank (see Figure 5). The top sketch shows a representation using the difference of two cuboidal bodies (+2 -1). The bottom pair of sketches show a representation of the same model using plane surfaces. Here it is necessary to refer to twelve surfaces in order to define the required zone. If the zone is to be defined as a single entity then elaborate input conventions are required to handle both unions and intersections within the single zone.

However body systems can encounter difficulties when two bodies meet at a common plane surface. During execution, rounding errors (however small) may cause the intersection of a particle track with the body surfaces to evaluate to different results at the common surface. Elaborate checks are then required to prevent the code concluding that the particle has left one body but not yet entered the other. It is also inefficient to repeat the calculation of the intersection point more than once where common planes exist.

The net effect is that the surface system is better for the tracking activities of the code but the body system presents a more user-friendly image. This conflict may be reconciled by introducing a stage during which a model defined using bodies is converted by the code into one based on surfaces.

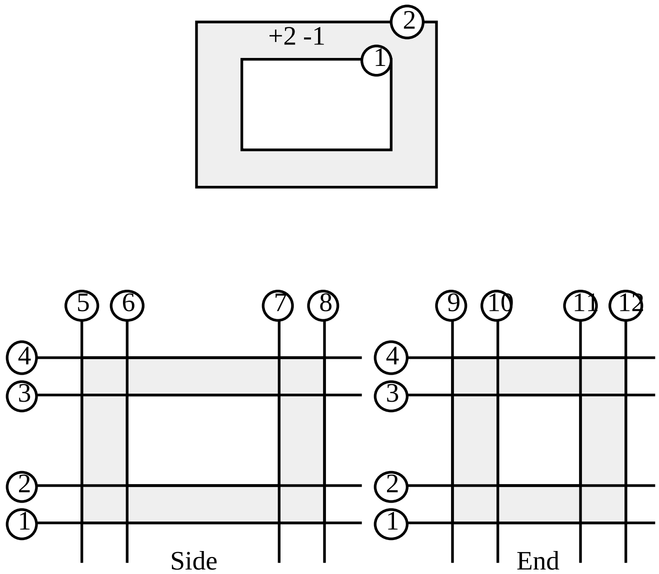

Figure 5 - Bodies versus Surfaces

A problem common to both surface and body methods is that of identifying the next zone to be entered by a particle. In a completely general system there is no simple way of establishing which zones are neighbours of any other zone. One solution is to test all other zones during the early stages of a calculation and remember which trials were successful. This list of successes for a given zone is then scanned first when a subsequent particle leaves that zone. Gradually, as the calculation proceeds, the code learns more about this list of neighbours for each zone and tracking becomes more efficient.

The problem is alleviated in part systems for many practical situations by the provision of defined structures, each limited in geometric complexity. This means that potential neighbours of each zone are identified by the part structure itself. On its own this tends to restrict the power of the modelling package, but as one component of a larger package, it has much to recommend it as an aid to both code and user efficiency.

B. Global versus Local Systems

In a part system, it is usual simply to transfer the associated local co-ordinate systems of the parts from the input to the execution stage. This is very economical on storage but, during tracking, the parameters defining the particle trajectory are constantly being revised to suit the local co-ordinates of the parts being entered. If rotations are involved in addition to translations then there is a significant computational overhead. One solution is to include a processing stage which expands the local co-ordinate part structure into a global model before tracking begins.

C. Boundary crossings versus Woodcock tracking

Woodcock tracking is a powerful algorithm but it cannot be regarded on its own as a complete alternative to the evaluation of boundary crossings. This is due to a loss of calculational efficiency that arises when the total cross-sections of the materials in the special geometry region vary by orders of magnitude. The frequency of collisions in a Woodcock tracking region is governed by the maximum cross-section of all the materials present in the region. In the presence of heavy absorbers (e.g. boron at thermal energies) the number of null events to be processed can become excessive in specific situations, although by a judicious mix of Woodcock tracking and boundary crossing this potential problem can be avoided in many practical situations. Irrespective of any such problems though, the power and ease of specification mean that Woodcock tracking is an important component in any new system.

D. User Image Issues

Even in the absence of extensive repetition there are advantages to be gained from dividing the total geometry into sub-sets. It is not uncommon for a major geometry model to include hundreds of bodies/surfaces and zones. It is a severe test of the user's memory to grasp the entirety of such a model. Communicating the model to another user or attempting modifications after the passage of time are difficult and error prone operations. The use of small, self-contained parts alleviates such problems.

This modelling structure is analogous to the engineering drawing practice of detailing components and then combining them in drawings of larger assemblies. Advantages include the simplification of exercises involving a series of related calculations (such as the evaluation of a transport flask with different loadings) and the ability to create and use libraries of standard parts. The use of local co-ordinate systems and component numbering means that there are few problems associated with the assembly of a complex model from a kit of relatively simple parts.

IV. FRACTAL GEOMETRY

Having considered the major options of existing methods, the new Fractal Geometry (FG) system has been developed satisfying the requirements for an efficient, powerful and user-friendly geometry package. The FG system is now included in the MCANO modular code scheme [5] from which the latest versions of the codes MONK and MCBEND have been formed. The FG system can be summarised as follows:

The input stage

- The geometry uses solid bodies for its basic construction units.
- The geometric model is constructed as a set of parts which may be hierarchic to any level.
- The bodies in each part are defined using a co-ordinate system local to that part.
- The zones in a part are the insides of single bodies or the differences and intersections of two or more bodies.
- Zones can contain single materials, Woodcock tracking geometries or subsidiary parts.

Intermediate processing

- The code expands the input geometric data to form a global model in which each body definition is translated and rotated to an absolute location.
- The bodies are converted to a surface system for robust and efficient tracking. Coincident surfaces are identified and discarded during this process.
- A zone relationship map is prepared to allow the code to follow the logic of the input part structure.

Tracking

- The final model is based on zones bounded by surfaces.
- The map of neighbours will be learned as the tracking process continues.
- Woodcock tracking may be used in any zone.

An advantage of Fractal Geometry is its ability to absorb the user image of established systems. A conventional MCBEND model is achieved by having a global part with no daughter parts. Additional bodies corresponding to single surfaces may be introduced to simulate a surface geometry input. The standard structures of the MONK geometry exist as a sub-set of permitted FG structures. By this means acceptable back-compatibility with existing models can be provided while at the same time providing a common way forward across different application areas. This will provide increased flexibility to user groups as well as reduced overheads associated with training, maintenance of expertise and software development.

V. GEOMETRY VISUALISATION

Geometry models for use with MCBEND have traditionally been verified by the calculation and display of two-dimensional slices through the geometry specification. These slices took the form of low-resolution, text-based pictures enabling hard copy to be produced on a conventional line printer. In order to provide a modern day equivalent to this, the interactive graphics tool VISAGE has been produced.

VISAGE is a high-resolution mouse/menu driven graphics tool for the generation, display and manipulation of two-dimensional slices through the geometry specification (see Figure 6). VISAGE has been implemented in C and uses the X-Windows and OSF/Motif tool-kits and hence is as portable as is currently possible. To date VISAGE has been implemented on a range of systems including Sun, DEC (VMS and ULTRIX) and HP, as well as IBM-compatible PC's running SCO UNIX. VISAGE images are produced using the geometry tracking routines of MCBEND and so are a genuine indication of the geometry seen by the modelling code itself. VISAGE can also be used to display MCNP geometry models.

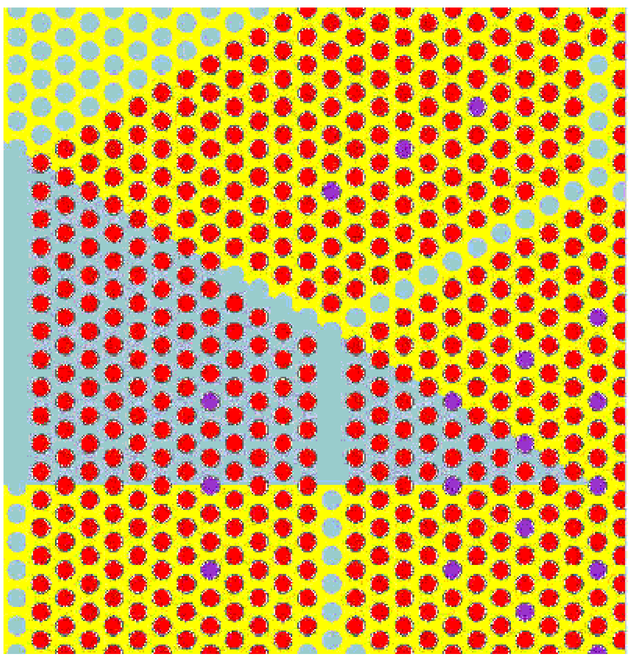

Figure 6 - VISAGE Image

Following the success of VISAGE, an advanced verification graphics tool called VISTA has been developed. VISTA performs the generation, display and manipulation of three-dimensional images by two independent means (see Figure 7).

One component of VISTA (VISTA-WIRE) exploits the portability of X-Windows, OSF/Motif and PHIGS, and the power of modern workstations, to give three-dimensional wire-frame based displays of the geometry specification. PHIGS is an international standard for three-dimensional computer graphics which has been implemented on many of the major computers currently in use. The use of PHIGS means that the images produced from this part of VISTA will not be produced using the geometry tracking routines of MCBEND. However the power and versatility of PHIGS enables a comprehensive interactive display package to be produced compatible with the power of today's computers.

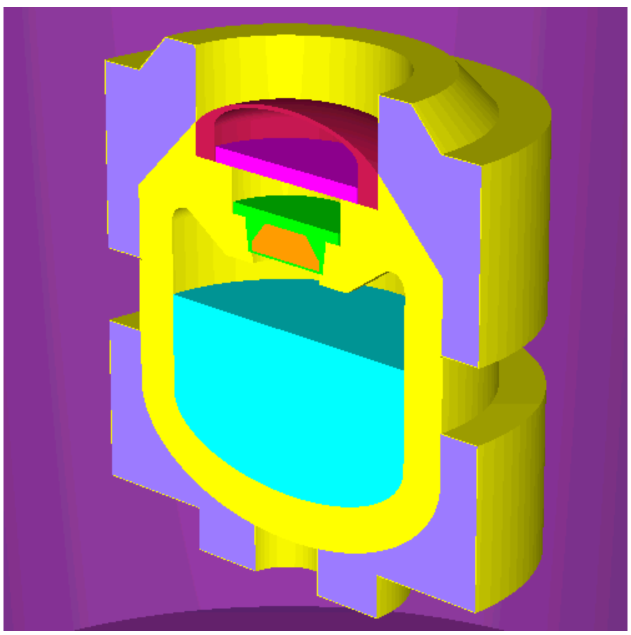

Figure 7 - VISTA Image

The second component of the VISTA package (VISTA-RAY) performs the generation, display and manipulation of three-dimensional images using particle tracking techniques. Here the images are created using the geometry tracking routines of MCBEND and therefore they are a genuine indication of the geometry seen by the modelling codes themselves. The images are produced using a simulated optical ray tracing algorithm which requires the tracking of rays from a viewing plane into the geometry model until a visible zone boundary is reached.

The user image for VISTA employs the same mouse/menu environment as VISAGE. VISTA enables materials, source bodies, dose points and splitting boundaries to be displayed and will have a range of viewing and manipulation options. In combination, VISAGE and VISTA provide a comprehensive set of geometry verification tools to accompany the Fractal Geometry package in MCBEND.

VI. CONCLUSIONS

This paper has described the creation of a new geometry modelling scheme for the Monte Carlo particle tracking computer code MCBEND. The scheme, called Fractal Geometry, is a pragmatic integration of the earlier MCBEND scheme and that of MONK and maintains the major benefits of each. At the same time judicious rationalisation has improved the overall efficiency of the Fractal Geometry package with respect to both of the earlier schemes.

Significant benefits from the new scheme will be gained by the code user community, in the form of greater flexibility and reduced overheads in related application areas. In addition code maintenance and development costs are reduced.

To accompany the Fractal Geometry package, the modern geometry visualisation software packages VISAGE and VISTA have been produced to improve the efficiency and accuracy of the geometry model verification process. These packages provide two- and three-dimensional image generation, display and manipulation facilities in a portable mouse/menu environment and in combination with the Fractal Geometry package provide a powerful and user-friendly geometry modelling and verification capability.

REFERENCES

[1] P C Miller, S J Chucas & I J Curl
'The Advanced Features of the Monte Carlo Code MCBEND' - Seminar on Advanced Monte Carlo Computer Programs for Radiation Transport, Saclay, France 1993

[2] N R Smith
'A Summary of the Current Status of the MONK6 Code and its Validation for Criticality Applications' - International Conference on Nuclear Criticality Safety ICNC'91, Oxford, UK 1991

[3] Judith F Briesmeister (Editor)

'MCNP - A General Monte Carlo Code for Neutron and Photon Transport' - LA-7396-M,Rev. 2

[4] M B Emmett
'The MORSE Monte Carlo Radiation Transport Code System' - ORNL-4972/R2

[5] N R Smith, R J Brissenden & M H Watmough
'The Development of the MONK and MCBEND Codes within the New Monte Carlo Modular Code Scheme MCANO' - Seminar on Advanced Monte Carlo Computer Programs for Radiation Transport, Saclay, France 1993