

Recent Developments to the ANSWERS[®] Monte Carlo Codes MONK[®] and MCBEND[®]

Jessica Fildes^{*ID}, Simon Richards^{**}, Adam Bird^{***}, Andrew Cox, Timothy Fry, David Hanlon, Brian Jones^{ID}, David Long, Francesco Tantillo, George Wright, and Richard Hiles

ANSWERS[®] Software Service, Jacobs, Kings Point House, Queen Mother Square, Dorchester DT1 3BW, UK

Received: 14 June 2024 / Received in final form: 30 October 2024 / Accepted: 15 November 2024

Abstract. MONK[®] and MCBEND[®] are two advanced Monte Carlo codes developed by the ANSWERS[®] Software Service, which have been used in over 30 countries worldwide for a range of applications, and have been well-established in the UK criticality and shielding communities respectively over the course of several decades. These codes continue to be actively developed to meet the needs of their users. We introduce a number of new developments which have been implemented in recent releases of these codes, or will be implemented in the next release. As MONK and MCBEND share a significant amount of source code, some of these new developments are in common between the two codes and some are specific to each. Improvements to the physics include stochastic mixing of bound thermal scattering data, Doppler broadening rejection correction, improved support for low temperature analyses in MONK, and modelling of photonuclear reactions in MCBEND. A novel approach for transferring irradiated geometry components between models has been developed for burn-up credit applications in MONK. A list mode facility has been added for subcritical analyses, including a model for the full distribution of neutron multiplicity. The geometry capabilities have been enhanced by the addition of novel stochastic geometry algorithms for modelling highly disordered heterogeneous mixtures and random distributions of shapes. IGES-formatted CAD geometries may be imported into MONK and MCBEND and approximated as a set of triangular surface polygons in a new Fractal Geometry body. The new IGES import feature additionally offers dedicated particle tracking routines that process the model geometry within the IGES file exactly, allowing the geometry to be used in MONK and MCBEND with no approximation. New estimators of the adjoint flux in both codes and adjoint-weighted kinetics parameters in MONK have been implemented, in addition to a pulse height distribution detector resolution model in MCBEND. A brief discussion of parallel processing in both codes is presented. New developments to the tools available in the accompanying Visual Workshop integrated development environment are also described.

1 Introduction

MONK[®] and MCBEND[®] are two advanced Monte Carlo codes which are part of the ANSWERS[®] codes suite. ANSWERS codes are used globally for modelling various scenarios encountered across the nuclear fuel cycle and beyond, including but not limited to: modelling of physics in nuclear reactor types such as AGR, BWR, CANDU, HTR, MAGNOX, MSR, RBMK, PBMR, PWR, SMR, VVER and experimental reactors; dosimetry assessments; radiation shielding; medical physics applications; and oil well logging.

MONK is an advanced Monte Carlo neutronics code for the solution of criticality safety and reactor physics

problems. It has a proven track record of application to the whole of the nuclear fuel cycle and is well established as the *de facto* standard criticality code in the UK criticality community. Furthermore, it is increasingly being developed and used for reactor physics applications. The current version of MONK is MONK12A [1], which was released in early 2024. The previous release was MONK11A [2], which was released in 2021 after an extensive programme of enhancements over the previous versions. MONK continues to be actively developed, with new features and enhancements, including some of those described in the current work, being incorporated into the next release, MONK12B.

MONK calculates the neutron multiplication factor, k_{eff} , for the system modelled using a staged (or iterative) approach, with each stage consisting of a fixed number of neutron *superhistories* [3]. A neutron superhistory is the set of tracks followed by a neutron and its fission progeny from birth to absorption or leakage, through a

* e-mail: jessica.fildes@global.amentum.com

** MONK Design Authority: e-mail: simon.richards@global.amentum.com

*** MCBEND Design Authority: e-mail: adam.bird@global.amentum.com

fixed number of fission generations. Examples of application cases for MONK include: fission transport container design; spent fuel dry storage; fuel fabrication; uranium enrichment; fuel dissolution; waste treatment and handling; reactor core loading assessments; thermal reactor analysis; and burn-up credit analysis.

MCBEND is well established in the UK radiation shielding community and globally as a powerful Monte Carlo code for general radiation transport analysis in shielding and dosimetry applications. The current version of MCBEND is MCBEND12A [4], released in 2021, and the next anticipated release will be MCBEND12B. MCBEND can be applied to a wide range of problems and has previously been used for the following applications: reactor plant design and shielding; fuel transport flask design; design of reprocessing facilities; design of fusion devices; analysis and interpretation of measurements in operating plants and in experimental facilities; calculation of personnel dose levels; and design of food irradiation and medical equipment sterilization facilities.

These codes remove both share the same advanced geometry modelling and detailed, continuous-energy collision treatment, providing realistic and flexible 3D models for accurate simulations of particle behaviour. They additionally utilize a powerful accompanying integrated development environment, Visual Workshop [5], which provides geometry visualisation, results display and a powerful toolset for further analysis of constructed models across codes in the ANSWERS suite.

The following sections briefly describe a number of recent developments to MONK and MCBEND, with the aim of informing the criticality safety, shielding and Monte Carlo communities of the latest capabilities of these well-established codes. Some developments apply individually to each code while some are shared. Further details of some of the recent developments specific to MONK are given in [6–8].

2 Physics

2.1 Improved temperature representation

The continuous-energy collision processor employed in MONK and MCBEND, referred to as BINGO, features run-time Doppler broadening to give an accurate representation of the effect of temperature on neutronic behaviour, with a stochastic mixing algorithm [9] to interpolate the secondary direction and energy in bound thermal scattering. The Doppler broadening rejection correction [10] is applied to remove the asymptotic approximation in the scattering kernel used in many Monte Carlo codes, improving the treatment of epithermal resonances in heavy nuclides at elevated temperatures. This requires the elastic scattering cross-section at zero Kelvin, which needed to be supplied in a separate data file in previous versions of MONK, but which MONK11A is now able to read directly from recent versions of the BINGO nuclear data libraries. These libraries also have a base temperature of 200 K, and bound scattering data for hydrogen and oxygen in ice below 273.15 K, and water above that tem-

perature, to facilitate criticality safety calculations below room temperature. These developments allow MONK, with its associated nuclear data libraries, to accurately model materials at any practical temperature without the need for any additional tools for generating Doppler broadened nuclear data.

2.2 Fission neutron multiplicity

The standard approach in Monte Carlo codes is to use bounding integer sampling for ν , the number of secondary neutrons from fission reactions. For most criticality and reactor physics applications it is sufficient to reproduce the correct average number of neutrons, $\bar{\nu}(E)$. However, the full multiplicity distribution becomes important for neutron noise analysis of subcritical systems (see Sect. 6.1). The Terrell algorithm [11] for the full multiplicity distribution has therefore been implemented in MONK as a user-selectable option.

2.3 Photo-neutron capability

MCBEND12A introduces modelling of the production of neutrons from photonuclear reactions (γ, n). The probability of photonuclear reactions occurring increases with photon energy and has a typical lower threshold of 5–6 MeV, with precise thresholds depending on the nuclide. While these reactions are uncommon, they may be important in some shielding applications. Options are available to increase the stochastic probability of neutron-producing reactions and to select either forward or isotropic emission. A library of photonuclear data for 43 nuclides, based mainly on ENDF/B-VII.0 data, supports the capability.

3 Geometry

Geometry in MONK and MCBEND models is specified according to a system known as Fractal Geometry (FG). In the material geometry section of the model input, one *part*¹ is specified at a time, where a *part* is a set of: simple *bodies*; *zones*, which are volumes of space created by the unions and intersections of bodies; and the contents of those zones. The zone contents may be: homogeneous materials; a *subsidiary part*, i.e. a part placed within a zone of a parent part; or a heterogeneous region containing further geometric complexity in the form of *hole* geometries (HG) employing Woodcock tracking [12].

A part may contain subsidiary parts nested to an arbitrary number of levels, and holes can also contain subsidiary holes, so a single part may in fact contain a complex hierarchy of geometric detail freely mixing conventional surface tracking and Woodcock tracking.

¹ The use of italic text in this section indicates specific terminology used in MONK and MCBEND.

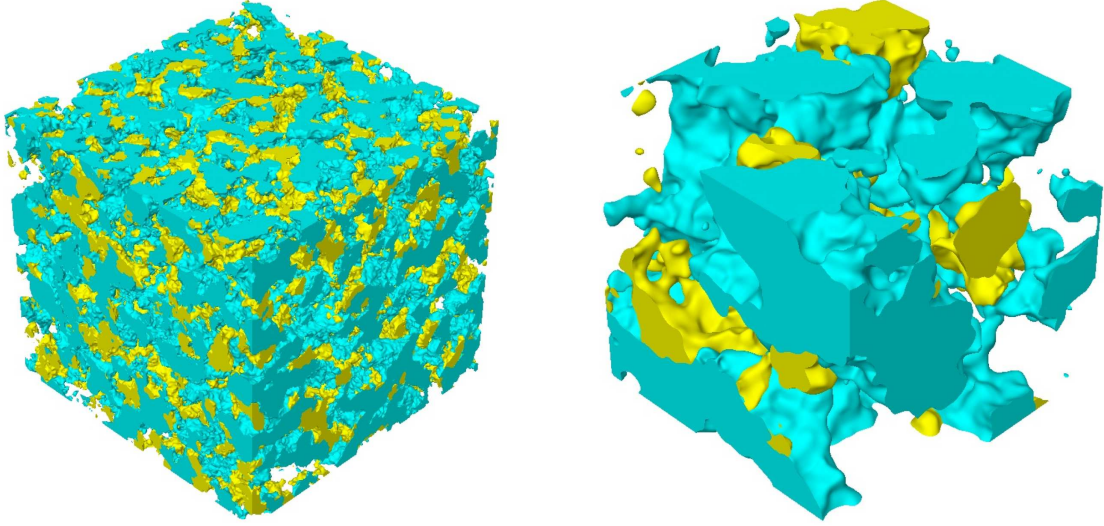


Fig. 1. Randomized geometries using Perlin noise with scale factors 2.0 (left) and 10.0 (right).

3.1 Stochastic geometries

Several novel stochastic hole geometry capabilities have been developed [6,7] for use in MONK and MCBEND. This expands MONK's flexible and detailed modelling capabilities while allowing realistic modelling of select systems which do not have a precisely known material distribution. The simulation of a large number of random realizations of a particular system produces a distribution of k_{eff} values, which may be used to estimate the mean k_{eff} for the system in addition to a maximum likely value.

3.1.1 Highly disordered mixtures

Highly disordered mixtures of two or more materials are simulated in the HET hole using an algorithm based on Perlin noise [13]. This is computationally efficient and produces randomized geometries which closely resemble the systems of interest. An important characteristic of this method is that it conserves the requested volume fractions of constituent materials in each random realization within the specified container. Containers currently available include cuboids, polygonal prisms, cylinders with an optional annulus, spheres and spherical shells. This approach represents the current state-of-the-art in modelling randomized heterogeneous mixtures in Monte Carlo radiation transport codes.

Properties of the distribution can be easily and intuitively altered while conserving requested material volume fractions, which conveniently allows analyses to be performed concerning criticality in highly disordered heterogeneous systems. Figure 1 demonstrates how a random distribution may be varied by changing the Perlin scale parameter, which alters the size of heterogeneous material chunks in the distribution.

The HET hole was used to perform calculations in one benchmark study which aimed to investigate the impact on the neutron multiplication factor of randomizing an intact fuel assembly model with a heterogeneous random

model [6]. Figure 2 demonstrates how k_{eff} varies as a function of the Perlin noise scale parameter when modelling this randomized assembly. Changes in the scale parameter can be related to the average material chord length Λ of the system, as displayed on the abscissa axis. A new diagnostic feature in MONK and MCBEND, to be documented in the next releases of the codes, allows distribution tables and average chord length estimates to be calculated for individual materials in the model, in addition to chord length distributions and averages over all materials. Small steps are taken along the path between randomly selected pairs of points in the model and changes in sampled material are monitored. If a section of the path is determined to lie completely within one material, the length of this path is registered as a chord for that material.

A particular feature of interest in Figure 2 is that a randomized PWR fuel assembly containing MOX fuel with an average material chord length of 1.5 cm achieved a reactivity which was approximately 5000 pcm higher than that of the intact fuel assembly, and approximately 21000 pcm higher than a distribution formed with a small Λ , which approaches a homogeneous distribution. Increasing the average chord length of the system additionally increases the spread of the k_{eff} distribution, as displayed by the error bars and shading. These results highlight the value of modelling multiple stochastic geometry realizations to aid estimation of the highest potential value of k_{eff} in random systems.

3.1.2 Random arrangements of debris

The RANDSHAPE hole models random distributions of discrete shapes with random sizes and orientations, as shown in Figure 3. It is intended for modelling fuel manufacturing debris and swarf, post-irradiation fuel element debris, and other miscellaneous waste streams. Any number of supported shapes may be defined by material and probability distribution, in addition to their mean

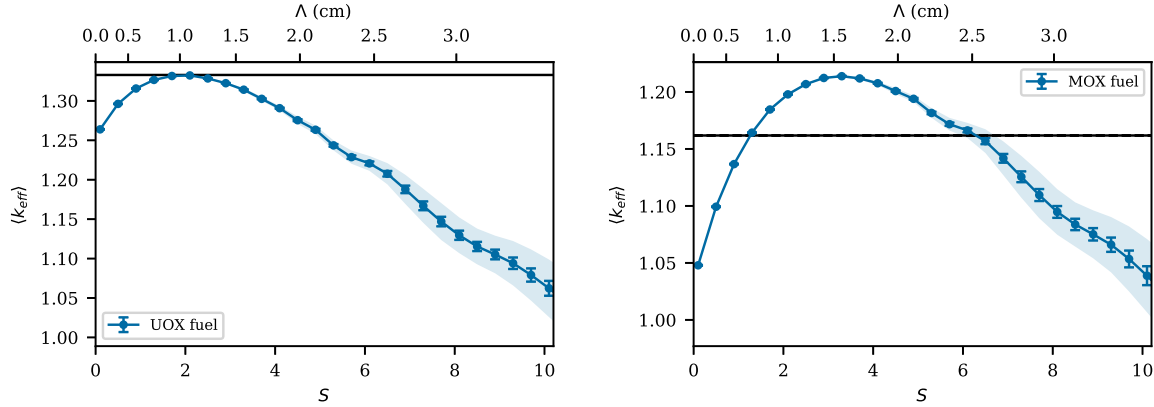


Fig. 2. Variation of k_{eff} with Perlin noise scale factor S and chord length Λ in a randomized PWR assembly containing UOX fuel (left) and MOX fuel (right). Each point represents the mean of 15 random realizations. The error bars represent the standard error on the mean of the ensemble, while the shading displays the standard deviation of the ensemble. The horizontal black line represents k_{eff} for the corresponding intact benchmark fuel assembly, before randomization.

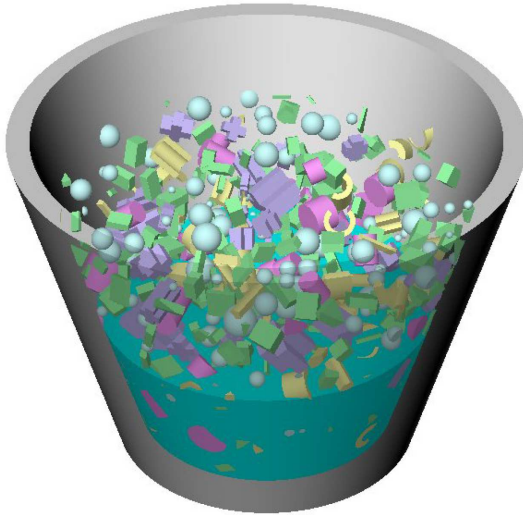


Fig. 3. An example of a RANDSHAPE hole containing a distribution of randomly sized and oriented shapes inside a cylindrical container which is partially filled with water.

dimensions and dimension uncertainties. Rather than storing the positions, dimensions and orientations of a potentially large number of shapes, this novel algorithm repeatedly regenerates the shapes on-the-fly during particle tracking.

These shapes are placed randomly into a container to a specified packing fraction. Shapes may be permitted to overlap to achieve higher packing fractions, or they may be arranged in the container such that they do not intersect. The latter process utilizes the Gilbert-Johnson-Keerthi (GJK) distance algorithm [14], explained in more detail in Section 5.1.1, which detects shape intersections. Additionally, there are options to allow shapes overlapping the container boundary to be either rejected or truncated. Figure 3 displays an example of the RANDSHAPE hole bounded by a cylindrical container.

3.1.3 Particulate fuel

The PEBBLE hole, which was an existing capability for modelling the random distributions of TRISO fuel particles in a spherical fuel pebble in a pebble bed reactor, has been further developed to allow a random distribution of intact, multicoated fuel grains in a cube, which can be tessellated to fill large volumes without the need to store an excessive number of grain centres. Grains which overlap the boundaries of the basic cube “wrap around” to the opposite face, so that when the cubes are tessellated a continuous distribution of intact fuel grains is modelled. In an alternative implementation for the modelling of randomly packed multi-coated fuel grains of equal radius, which is in the late stages of development and testing at the time of writing, high grain packing fractions of up to 62% can be achieved in a cuboid container while ensuring no sphere truncation at container boundaries. Random close packing can be applied in a variety of container shapes in the development version of this feature, which were chosen to offer a wide range of options for modelling of particulate fuel and all prevent unphysical sphere truncation at the container boundary. These container shapes include cuboids, polygonal prisms with user-specified polygon vertices, cylinders with an optional annulus, spheres and spherical shells. Figure 4 displays examples of the PEBBLE and PEPPER hole geometry features.

3.2 POLY body

Recent releases of MONK and MCBEND have introduced the polygon surface, or POLY body, which is an FG body defined by an arbitrary number of triangular polygons. It is arbitrarily re-entrant and parts of the same body may even be completely separate volumes of space. A single

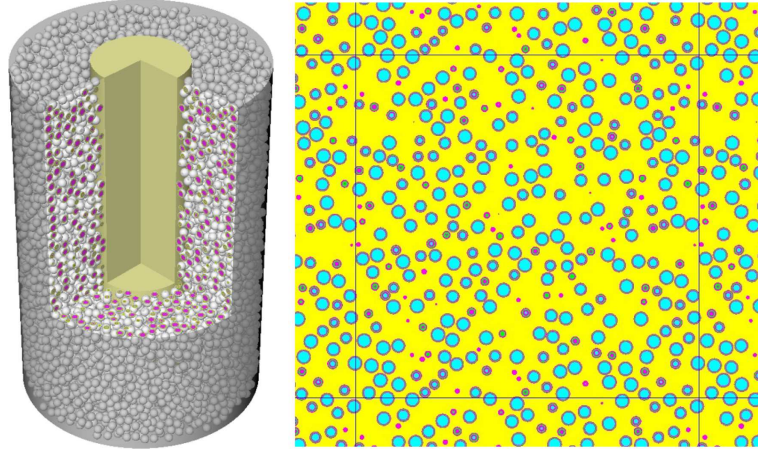


Fig. 4. Visual Workshop images showing a 3D ray trace visualisation of the PEBBLE hole in an annular container (left), and a 2D ray visualisation of the PEPPER hole demonstrating how spheres which overlap the cube boundary are handled (right).

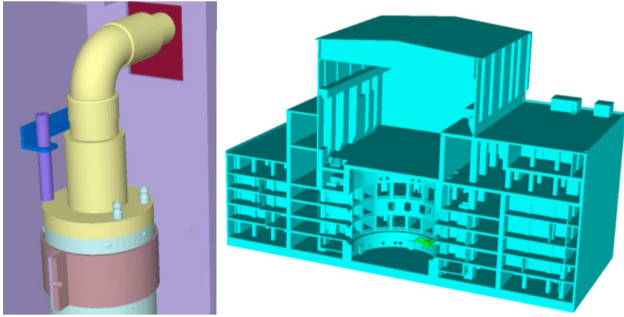


Fig. 5. Visual Workshop images showing examples of POLY body geometries: a small section of a larger piece of machinery imported from an OBJ (Wavefront .obj) file (left); and a section through the model of a building imported from an STL (Stereo Lithography) file as a single POLY body (right).

POLY body may be an entire building with multiple rooms, floors, doors and staircases (Fig. 5).

There is a range of options for inputting the polygon mesh directly as MONK or MCBEND input, including extrusion of a surface, rotation of a profile and options to “grow” the points on a curved surface to conserve volume. While defining more complex POLY bodies manually can be impractical, it is expected that the facility will be used by importing files of POLY definitions from external sources, such as tetrahedral mesh models or relevant surfaces from CAD packages. Visual Workshop [5] provides a translation tool to convert a tetrahedral mesh or an OBJ file to a set of POLY bodies. Faceted geometries are generally available from many sources. However, they will always involve an element of approximation when curved surfaces are represented. The IGES (Initial Graphics Exchange Specification) import feature described in Section 3.3 uses a dedicated set of particle tracking routines that process the geometry within the IGES file exactly, with no conversion or approximation, at the cost of increased run time as a result of the additional computation required. Table 1 displays the increase

in runtime when using IGES tracking instead of FG for different version of MCBEND. In a development version of MCBEND12B, which is yet to be released, IGES tracking is 4.8 times slower than FG tracking for a transport flask model, which is a significant improvement from 84.8 times slower in MCBEND11A.

In common with the tetrahedral mesh geometry, each POLY body uses an underlying voxel acceleration mesh which means that a traced ray efficiently steps through the voxel mesh and only needs to calculate intersections with a small number of triangular polygons that overlap a given voxel. Because only the surfaces are used, a POLY body representation of a model is more storage and computationally efficient than a tetrahedral mesh. Unless one requires results in each tetrahedron, a POLY body representation should be used.

3.3 CAD geometries

Computer-Aided Design (CAD) geometries which have been converted into a tetrahedral mesh representation or polygon surface can be imported into MONK and MCBEND via the TETMESH hole geometry or POLY body features. An additional CAD import feature has been introduced which enables a user to import a CAD geometry model in IGES format without having to convert it manually to a MONK or MCBEND geometry input. Tracking in the IGES model is provided by a dedicated set of particle tracking routines which process the exact CAD geometry, with no approximation required.

It should be noted that even though an IGES CAD model is treated like an FG body in both codes, the body contains the whole IGES model. In addition, there is no requirement to explicitly model voids within the CAD because these automatically become the defined interstitial material for that imported CAD object.

An IGES model consists of subfigure definitions and subfigure instances (these are realizations, and there could be many, of a particular subfigure definition). A subfigure definition is a shape that could be simple, comprising

Table 1. Performance improvements in IGES tracking between MCBEND11A and MCBEND12B, for a range of different models. The performance metric compared is the average number of particle histories per second.

Model	Performance ratio (FG/IGES)			Speedup factor from 11A to 12B
	MCBEND11A	MCBEND12A	MCBEND12B	
Transport flask	84.8	7.7	4.8	17.8
Fuel flask	283.3	9.4	3.6	79.6
Barrel room	26.6	6.2	5.5	4.8
CSG complex	17.3	2.8	3.5	5.0
Detector	22.5	3.1	1.9	12.0
Laby	26.1	4.9	1.9	13.5
Pipeworks	101.2	27.1	22.7	4.5
Triso	57.1	4.7	2.3	24.8
Water flask	48.4	6.2	2.5	19.8

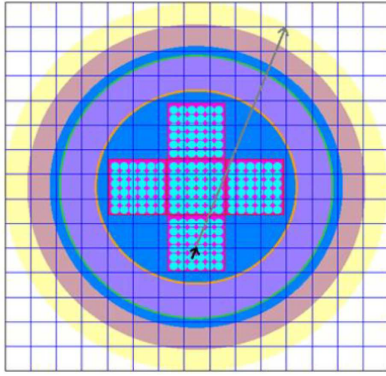


Fig. 6. Voxel acceleration grid overlaid on IGES geometry.

a few components, or complex, comprising many components, depending on how the CAD modeller has created it. MCBEND12A provides increased user control over allocating materials to parts of an IGES model. The user may define the compositions of each subfigure instance, and may optionally specify each component within a subfigure instance separately. Compositions may contain solid materials or, if desired, hole geometries. Placing FG bodies or parts in the IGES model will “blank out” parts of the IGES model, effectively enabling the placement of parts of the FG model inside the IGES model, allowing further modelling flexibility. Recent developments have made it possible to import multiple IGES files or a subset of an IGES file so that parts of a larger IGES model or models can be used as part of a MCBEND or MONK geometry. In addition, note that an IGES file imported into an FG part will be replicated, translated and rotated with that part as the model is assembled.

While these enhancements have made the CAD import capabilities a powerful addition to the geometry capabilities of MONK and MCBEND, much development effort has focused on improving the runtime performance of the IGES import. MCBEND12A has introduced a number of enhancements to the IGES import, with further enhancements to be introduced in MCBEND12B.

Table 1 shows the relative performance improvements since MCBEND11A [15].

The descriptions of the models in Table 1 and the performance of MCBEND11A have previously been reported [16]. The largest performance improvement was obtained by reducing the number of unnecessary ray intersection calculations. When an FG model is built, there is a reasonable understanding of which entities are near other entities, and what is contained in those entities. This means that after, say, a collision event there is a limited number of surface intersections that need to be evaluated to determine the next volume of space entered. In addition, FG learns which volume of space is likely to be entered from another and prioritizes these in the search for the next volume to be entered. For an IGES geometry, relying on testing the bounding boxes of the components in the subfigure definitions was not efficient as can be seen by the performance of MCBEND11A IGES compared to FG. For MCBEND12A, a voxel acceleration system, not unlike the one used for the POLY body, was implemented. A voxel grid is overlaid on the IGES geometry and a pre-processing step identifies the IGES components overlapping each voxel. It is this development that has accounted for the majority of the improvements between MCBEND11A and MCBEND12A. As an illustration the slice through the fuel flask model in Figure 6 shows fuel pins in a cruciform. FG knows which parts in the array that forms the fuel element are in its vicinity and calculates intersections to those first, whereas the IGES geometry, with no knowledge of proximity of shapes, needed to calculate intersections to all surfaces to the edge of the geometry, as shown by the arrow in Figure 6, before deciding which one was encountered.

4 Results tallies

4.1 Adjoint flux

A new Iterated Fission Probability (IFP) based method has been added into MONK for tallying the adjoint flux.

This method has been utilized as part of novel developments to calculate the effective kinetics parameters [17], and in generalized sensitivities calculations [8]. Iterated fission probability methods calculate the adjoint flux using its equivalence to the importance of a neutron released at a particular point in phase space spanning energy, position, time and direction.

A neutron generation is defined as the sequence of events which occur between the birth of a neutron during fission and its capture in a subsequent fission event. Calculation of the adjoint flux considers an original neutron generation α , and an asymptotic neutron generation γ . The asymptotic generation represents the point at which the user believes any given sequential neutron path should form part of the representative set of important sequential neutron paths which contribute to the steady state power output of the system. The adjoint flux may be estimated for a specific spatial region r and energy bin E by scoring two quantities: the total weight of neutrons released in original generation α with energy E from position r ; and the total of descendant weights directly caused by those neutrons released in generation α occurring in asymptotic generation γ . An estimate of the adjoint flux per unit weight $\Phi^\dagger$ for a number of neutrons $N(r, E)$ is therefore given by

$$\Phi^\dagger(r, E) \sim \frac{\sum_{n=1}^{N(r, E)} \sum_{d=1}^{N_d^\gamma} w_d^\gamma}{\sum_{n=1}^{N(r, E)} w_n^\alpha}, \quad (1)$$

where w_n^α is the weight of a neutron n released in the original generation α ; w_d^γ is the weight of a descendent d neutron released in the asymptotic generation γ from an ancestral neutron n released in original generation α ; and N_d^γ is the total number of neutrons in the asymptotic generation γ descended from the initial neutron.

Scoring of these quantities is made simple in MONK due to its use of superhistory powering. Neutron superhistories contain sets of tracks followed by a neutron and its fission progeny from birth to absorption or leakage though a specified number of fission generations. This allows MONK to intrinsically know which fission neutron the current tracked neutron is descended from, for any generation of the superhistory, without needing to pass a large number of additional tags along through different calculation stages. For a given superhistory with 10 generations, the denominator of equation (1) is scored in generation 1 of the superhistory, designated original generation α , by summing the neutron weights for a particular model region and energy group. The numerator of equation (1) is scored by summing the descendant neutron weights in the final generation, designated asymptotic generation γ . A number of generations, referred to as latent generations, lie between the first and final generations. This allows IFP estimators of the adjoint flux to converge to an accurate result. Currently there is no automated way of selecting an appropriate number of latent generations to ensure convergence for any given calculation. However, the user is able to increase the number of latent generations per superhistory in MONK if desired.

4.1.1 Effective kinetics parameters

Once estimated, the adjoint flux may be used to calculate effective kinetics parameters commonly used in reactor physics calculations. MONK can currently calculate estimates of the effective delayed neutron fraction β_{eff} and the effective neutron generation time Λ_{eff} , in addition to their associated statistical uncertainties. Calculation of stochastic uncertainties for these quantities is non-trivial since various quantities used in the calculation of these effective kinetics parameters are strongly correlated, and therefore have significant covariance terms which must be accounted for in order to accurately estimate the uncertainty. Values of these parameters obtained in MONK show good agreement when compared with experimental data comprising fourteen different delayed critical experiments [17].

4.1.2 Generalized sensitivities

In addition to effective kinetics parameters, the adjoint flux may also be used to calculate sensitivities of various parameters to perturbations in the underlying nuclear data. Typically sought are sensitivities of k_{eff} to nuclear data, however more recent extensions to these perturbation methods have been developed to calculate sensitivities of other parameters to nuclear data perturbations. Such methods are referred to as generalized perturbation methods. In theory, sensitivities may be calculated for any response R that can be represented in the form

$$R = \frac{\langle \Phi^* | A | \Phi \rangle}{\langle \Phi^* | B | \Phi \rangle}, \quad (2)$$

where A and B are arbitrary operators, $\langle \Phi^* |$ represents the adjoint flux and $|\Phi\rangle$ represents the forward flux which is typically scored in MONK by summing the lengths of all particle tracks crossing a given volume and dividing by that same volume [8].

Currently in MONK, sensitivities may be calculated using this generalized perturbation method for the effective kinetics parameters mentioned in Section 4.1.1, various reaction rates, and the power shape [18]. The stochastic uncertainties of these quantities are also reported alongside the quantities themselves.

4.2 Pulse height distribution detector resolution model

Detectors generally produce a Gaussian-like peak in the output spectrum for incident gamma-rays of the same energy. This distribution of the output signal occurs due to fluctuations in the number of excitations and ionizations in the detector material and in response times of the detector electronics. MCBEND now has a detector resolution model to simulate this effect. Calibration of a detector may characterize the broadening effect via the full width at half maximum (FWHM). For the majority of detectors the FWHM will increase near-linearly with energy. The implementation assumes a Gaussian distribution for the resolution, where the standard deviation of the distribution is related to the energy-dependent FWHM.

The detector resolution model can be applied during runtime or as a post-processing step or both. Runtime

evaluation is more accurate, but requires more computation. The post-processing option is a computationally cheaper option and provides a more direct comparison with alternative post-processed methods.

5 Burn-up and burn-up credit

A microscopic burn-up capability based on multigroup data in 172 energy groups has been available in MONK for several decades and is still available in MONK12B. More recent versions, including MONK12B, have additionally included a continuous-energy burn-up methodology based on the concept of *artificial materials*². A mesh is overlaid on the geometry and a unique artificial material is generated for each real material in each cell of the mesh, and it is the compositions of these artificial materials which are modified as a result of burn-up. A Monte Carlo method is used to estimate the volumes of the artificial materials in the first time step in order to correctly normalize the reaction rates before solving the depletion equations. The updated artificial material compositions are written to an *archive file* at the end of each time step. The use of an overlaid mesh greatly simplifies the task of managing the spatial discretization of materials required in burn-up and coupled multiphysics calculations, and means that the fidelity of the spatial discretization can easily be changed without needing to change the underlying geometry model.

A facility, known as COWL materials, is available for importing irradiated material compositions from a reactor model (the *donor* model) to a post-irradiation criticality model (the *receiver* model), by specifying the user-defined material name or number and the cell of the overlaid burn-up mesh in which it appears. This also provides control over which actinides and fission products are included in each material to simplify the application of actinide only, or actinide and specific fission product, burn-up credit. The COWL materials feature has been available since MONK10 and is still available in MONK12B.

For burn-up credit applications we typically wish to model irradiated fuel assemblies in ex-core geometries such as spent fuel transport flasks, spent fuel ponds and dry storage facilities. This requires both the geometry model of the fuel assemblies and the compositions of the irradiated materials to be transferred from the original reactor model to a subsequent criticality model. While the discretized material geometry could be modelled explicitly in the criticality model, and COWL materials used to import the irradiated material compositions, this is somewhat laborious. The artificial material method means that the spatial discretization of the materials does not need to be represented in the model geometry, greatly simplifying the model design. However, the transfer of irradiated fuel assemblies to a burn-up credit model requires a novel approach.

In the reactor model a particular type of fuel assembly might be modelled once, and then replicated in many

different locations in the core, potentially with different reflections and rotations. Each of these will have a unique set of artificial materials representing the spatial discretization of materials, and the positions of the material boundaries may be different in each case as a result of their translations, reflections, and rotations relative to the mesh.

5.1 Importing artificial materials

To import a specific *part* (see Sect. 3 for an explanation of geometry in MONK and MCBEND models) from a reactor donor model into a burn-up credit receiver model it is first necessary to identify which specific instance of that part is required, and then to identify the set of artificial materials associated with that part. This means that the set of mesh cells containing the required part must first be identified.

In the general case a part could be at any position within the mesh and at any orientation with respect to the mesh. Depending on the mesh fidelity a part might lie wholly within a mesh cell, or it may overlap many cells, some of which lie wholly within the part and some of which may be partially overlapped. The identification of cells fully or partially overlapping the part in three dimensions is therefore not a trivial problem.

The outer body of a part, or the *part container* can be any of the simple bodies from which the MONK geometry is constructed, including: cuboids; circular and elliptical cylinders; spheres; hemispheres; triangular, trapezoidal and hexagonal prisms; cylindrical sectors; tori; cones; and rotated ellipses. We begin by considering a body-aligned bounding box containing the part. The set of cells associated with this bounding box is sufficient to contain all of the cells associated with that part and its subsidiary parts. We also consider an axis-aligned bounding box which wholly contains all cells which could possibly be associated with the required part. This leads to a subset of cells which need to be tested for intersection with the body-aligned bounding box.

5.1.1 The Gilbert-Johnson-Keerthi algorithm

Testing for the intersection of a bounding box of arbitrary orientation with a cell in a mesh is essentially a form of collision detection. A number of different approaches could be considered, but the approach adopted in this work uses the Gilbert-Johnson-Keerthi (GJK) distance algorithm [14] which is widely used for real time collision detection in physics engines for video games. The characteristics of efficiency and robustness which make the algorithm suitable in that application similarly apply to the current application.

Defining two convex objects A and B as a set of points in three-dimensional space, the Minkowski sum $A + B$ of the two objects is found by adding every point in A to every point in B . The Minkowski difference of A and B is determined by computing the Minkowski sum of A and $-B$, where $-B$ denotes the reflection of B about the

² The use of italic text in this section indicates specific terminology used in MONK.

origin³. If and only if A and B intersect it follows that there exists a point in common to A and B , from which it follows that the Minkowski difference contains the origin if and only if A and B intersect. The goal of the GJK algorithm is therefore to determine whether the Minkowski difference contains the origin.

In fact, the GJK algorithm does not actually need to calculate the Minkowski difference directly. Instead, it iteratively modifies simplices (a point, a line segment, a triangle or a tetrahedron defined by 1, 2, 3, or 4 points respectively) within the Minkowski difference until a simplex is either found to contain the origin or it is found that the origin cannot be enclosed. This is usually achieved within a small number of iterations.

5.2 Tracking in donor parts

The process of tracking neutrons in a Monte Carlo code fundamentally requires the answers for two questions: “what is the material at the current particle position?”; and “what is the distance to the next material boundary in the direction of travel?” (the answer to the second question not being needed within a hole material in the case of Woodcock tracking). When this tracking is performed in a burn-up calculation using artificial materials the material at each point must be mapped to the corresponding artificial material for the mesh cell containing the point, and distance to the next material boundary must take account of the boundaries between artificial materials which are imposed by the burn-up mesh.

In order to answer these two questions when tracking in donor parts we must transform the particle position to the coordinate system of the donor model in order to determine in which cell of the burn-up mesh the point would have been located in the burn-up calculation. The basic algorithm is described by the pseudo-code shown in Algorithm 1.

5.3 Editing archive files

The import of irradiated material compositions from a reactor model to a criticality model includes, by default, all properties of the materials, including the full composition of isotopes and their fission, capture and decay products, and their temperatures. For burn-up credit applications we may wish to change some of the material properties, for example limiting the isotopic compositions to certain actinides and fission products for actinide only, or actinide plus specified fission product burn-up credit studies, or changing the material temperature.

In previous versions of MONK, modification of individual material properties in archive files had to be performed by the user and required the text-based archive file to be manually edited to remove unwanted nuclides and alter material temperatures. MONK12A therefore introduces a new archive file editing capability which allows

Algorithm 1 Algorithm for tracking in donor parts

```

while not at end of current track do
   $xyz \leftarrow$  current position vector
   $uvw \leftarrow$  current direction vector
   $nextp \leftarrow$  next part to be entered by track
   $nextm \leftarrow$  next material to be entered by track
   $d_1 \leftarrow$  distance from  $xyz$  to next material boundary in
  direction  $uvw$ 
  if  $nextp$  is a donor part then
     $dxyz \leftarrow xyz$  transformed to donor model coordinates
     $duvw \leftarrow uvw$  transformed to donor model coordi-
    nates
     $ijk \leftarrow$  index of donor mesh cell containing  $dxyz$ 
     $d_2 \leftarrow$  distance from  $dxyz$  to next mesh boundary in
    direction  $duvw$ 
    if  $d_2 < d_1$  then
       $d_1 \leftarrow d_2$ 
       $nextm \leftarrow$  donor material number for material
       $nextm$  in cell  $ijk$ 
    end if
  end if
  current position vector  $\leftarrow xyz + uvw * d_1$ 
  (Continue with standard Monte Carlo tracking and col-
  lision processing...)
end while

```

these material changes to be performed in a simple and general way via the MONK input file. Given the file path of an archive file, changes can be specified for real and artificial materials using MONK keywords. MONK produces a new archive file with the specified changes immediately after reading this section of the MONK input so that the resulting archive file can be used immediately in a burn-up calculation if desired. This streamlines the user experience and reduces the capacity for user error.

5.4 Results

We restrict ourselves here to presenting results showing the successful transfer of spatially discretized materials from a reactor donor model to a burn-up credit receiver model. This is the core functionality of the method. Results from a full simulation of a burn-up credit application are beyond the scope of this article, and will be presented in a future publication.

The geometry visualization in Visual Workshop [5] features 2D and 3D ray trace views which utilize the same subroutines as the Monte Carlo tracking code, giving assurance that the visualized geometry is the same as that simulated by the Monte Carlo code. Here we use the 2D ray trace to demonstrate the results of the part transfer capability in a completely fictitious example.

Figure 7 shows a nominal reactor core containing 25 nominal fuel assemblies, each of which contains 264 fuel rods in a 17×17 array with 24 guide tubes and a central instrumentation tube. An asymmetric loading of 8 burnable absorbers is present in one corner of each assembly. The nominal fuel assembly is modelled once in the MONK input as a part with multiple subsidiary parts, and this common assembly model is replicated 25 times with

³ Note that the Minkowski difference may alternatively be defined as the inverse operation of the Minkowski sum, using set complements, but that definition is not used here.

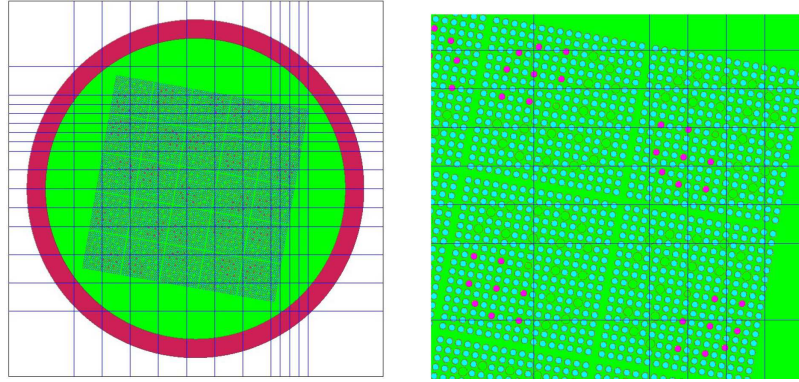


Fig. 7. A fictitious reactor geometry containing 25 copies of the same fuel assembly with various rotations, with an overlaid burn-up mesh (top); and an expanded detail of the upper right corner of the core showing the asymmetric absorber positions in rotated assemblies (bottom).

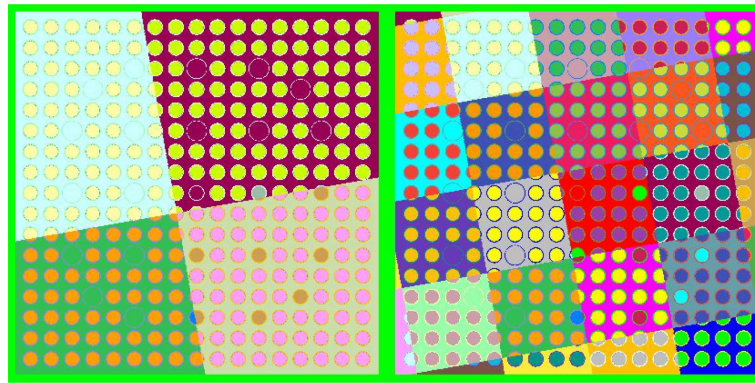


Fig. 8. The discretized materials resulting from applying the part transfer capability to the bottom left and top right fuel assemblies from the core shown in Figure 7.

different rotations in multiples of 90 degrees, in order to fill the core. A non-uniform mesh is then overlaid over this model in order to define the spatial discretization of materials for the burn-up calculation. An example of a further complication is provided by rotating the whole core relative to the mesh. The various geometry features present in this example are not intended to be representative of a real system, but rather to illustrate the flexibility of the part transfer methodology.

Figure 8 shows the result of using the new part transfer methodology to transfer two of the fuel assemblies (the lower left and the upper right) from the reactor model to a nominal spent fuel transport model. The fuel assemblies are modelled in the spent fuel model in exactly the same way as in the reactor core; the input lines are literally copied and pasted *verbatim*. It is not necessary to represent the spatial discretization of the materials in the model input, since this is all handled automatically by the part transfer method. The figure shows how the original materials are replaced by the irradiated artificial materials which have been imported from the burn-up calculation. The different spatial discretizations of the two assemblies are clearly demonstrated, accounting for the different positions and rotations of the assemblies in the reactor model, and the differing mesh fidelity in those positions.

The rotations of the assemblies, rotations of the core relative to the mesh, the non-uniform mesh, and the independence of the mesh and underlying geometry all combine in this example to produce a spatial discretization of the materials which would be extremely challenging to model explicitly. This therefore demonstrates the power of the part transfer methodology in greatly simplifying the modelling of such systems.

5.5 Future work

Currently the artificial material approach is applied over a grid formed from a box with arbitrary subdivisions in the x , y and z directions. For some systems an alternative form of mesh could improve efficiency. Therefore, a future development is planned to implement part-based meshes which allow greater flexibility and efficiency.

6 Subcritical analysis

6.1 List mode output

A new list mode capability in MONK simulates detector signals for the analysis of subcritical systems using

neutron noise techniques in fixed-source calculations. Detectors are defined as a geometric volume within which detections are recorded, together with a nuclide-reaction (MT number) pair, such as U235 18 for a ^{235}U fission chamber. Each neutron detected in a list mode detector produces an entry in a file, recording: a unique sample number for the initial source particle; the source region; the detector number; and the age of the neutron. Note that the detected neutron may either be the initial source neutron or any of its progeny. The age is the total time between the birth of the initial source neutron and the detection of a neutron in the detector, with the addition of a delay time for each delayed neutron within the associated fission chain.

Initial validation of the list mode capability in MONK is based on a benchmark evaluation [19,20] of subcritical experiments performed in the Inherently Safe Subcritical Assembly (ISSA) facility at LLNL between June 2017 and February 2018. The benchmark experimental measurements include time-based count data registered in ^3He -filled detectors analysed using the Feynman variance-to-mean methodology [21]. A count distribution is constructed based on fixed width time bins, and these data are used to form a frequency distribution. This would be a Poisson distribution if the source was purely random. The effect of correlated counts from fission chains is to cause a deviation from the Poisson distribution. A measure of this deviation is obtained by calculating the reduced factorial moments of the count distribution and then from these the first two Feynman moments Y_{2F} and Y_{3F} (which for a Poisson distribution are both zero).

The MONK results for this benchmark are compared with the results achieved by the benchmark evaluators, using the COG code, in Table 2. Cases 1 to 5 correspond to experimental configurations containing: one; two; four; six; and nine fuel assemblies respectively. From the MONK calculations it is possible to estimate the total multiplication, T , of each configuration using the number of samples started and the total number of samples followed. While the multiplication is not a benchmark parameter it is worth noting that the calculated values are in good agreement with the expected values of between 2 and 10 for these systems [20]. MONK also provides estimates of k_{eff} from the fixed source calculation which are consistent with these multiplication values, noting that the total multiplication is related to k_{eff} by

$$T = \frac{1}{1 - k_{\text{eff}}}. \quad (3)$$

The MONK results show reasonably good agreement with those of the experiment and the COG code used by the benchmark evaluations in the benchmark report analysis, noting that the MONK calculations were performed using JEFF-3.1.2 nuclear data while COG used ENDF/B-VII.1. The level of agreement between MONK and the experimental results is very similar to that obtained by the benchmark evaluators using COG.

These cases, especially Case 5 (containing nine assemblies) are sensitive to the modelling detail and very useful for validation of subcritical modelling methods. A calculation performed with the standard bounding integer

sampling approach for the number of neutrons per fission gives Y_{2F} and Y_{3F} values of around 3.44 and 24.50 respectively, compared with the results shown in Table 2 using the Terrell model. Further assessment of the suitability of the Terrell algorithm is planned to determine whether more detailed, but computationally more expensive, treatments of fission neutron multiplicity, such as the Fission Reaction Event Yield Algorithm (FREYA) [22], are required.

Further work is planned to validate the list mode capability using other benchmarks and to investigate the application of neutron noise techniques to the assessment of subcritical margins in different core configurations.

7 Parallel processing

MONK and MCBEND, while being closely related codes with a great deal of shared source code, have different requirements for parallel processing and therefore take different approaches.

MONK is parallelized using MPI (Message Passing Interface) which works well on both distributed memory and shared memory architectures, albeit at the expense of having multiple copies of the data on a single node. To address this point recent work has focused on the potential use of shared memory MPI to reduce the memory use on each node. To improve the parallel scaling further, recent developments have introduced a task-stealing algorithm to optimize load balancing.

MCBEND on the other hand, being a pure fixed-source code, is embarrassingly parallel so does not need MPI parallelism. Instead, it has a so-called “grid” capability which allows results from multiple serial calculations to be combined to produce the final results. This has optimal parallel scaling but also suffers from the problem of data duplication if multiple parallel processes are run on the same node. Therefore, MCBEND also employs shared memory parallelism using OpenMP. Typically, a MCBEND calculation with large memory requirements would use a combination of grid and OpenMP capabilities.

8 Visual Workshop

Visual Workshop [5] is the integrated development environment for MONK, MCBEND and other ANSWERS codes, which allows users to edit input files, display the model geometry as a wireframe and 2D/3D ray trace, run the physics codes (including submitting jobs to HPC scheduling systems), view the output files, display results on the model geometry, and perform various analyses. An important characteristic of the 2D/3D ray trace in Visual Workshop is that it uses exactly the same subroutines as MONK and MCBEND use to track particles through the geometry, giving users confidence that the displayed geometry is exactly the same as the geometry in which the particles are tracked.

For criticality calculations it is necessary to ensure that all sources of bias and uncertainty associated

Table 2. MONK results for subcritical measurements of water-moderated, highly enriched UO_2 , and comparisons between experimental and calculated results from MONK and COG [20].

Case	MONK Results				$(C - E)/E$			
					MONK		COG	
	T	k_{eff}	$Y_{2\text{F}}$	$Y_{3\text{F}}$	$Y_{2\text{F}}$	$Y_{3\text{F}}$	$Y_{2\text{F}}$	$Y_{3\text{F}}$
1	1.74	0.4255 ± 0.0002	0.04593 ± 0.00061	0.00442 ± 0.00086	0.3%	-33.2%	0.3%	-21.5%
2	2.24	0.5536 ± 0.0002	0.10481 ± 0.00082	0.02344 ± 0.00162	-6.9%	-34.4%	-4.3%	-25.1%
3	3.61	0.7230 ± 0.0002	0.37675 ± 0.00239	0.30242 ± 0.01392	-13.2%	-31.8%	-9.1%	-24.6%
4	4.97	0.7991 ± 0.0001	0.83636 ± 0.00266	1.49415 ± 0.02125	-15.9%	-32.5%	-9.2%	-20.8%
5	10.23	0.9025 ± 0.0001	4.32338 ± 0.01870	39.03251 ± 0.45234	-12.0%	-23.4%	-11.6%	-22.8%

with the criticality code used are incorporated in the estimation of the maximum calculated k_{eff} . Visual Workshop therefore includes a number of tools to assist the criticality analyst. In addition, Visual Workshop hosts further uncertainty quantification tools for broad use across multiple codes, including MONK and MCBEND.

8.1 Similarity index tool

MONK has a first order sensitivity method which can estimate the energy and nuclide dependent sensitivity of k_{eff} to nuclear data. Comparing the sensitivities of two systems provides an objective measure of how similar they are. Such a measure can provide strong justification for the use of selected benchmark experiments in providing validation evidence for a particular application.

Visual Workshop contains a Similarity Tool for use in MONK calculations which calculates a similarity index based on the comparison of sensitivities in application cases and selected validation cases. This was originally developed to calculate two separate sensitivity measures, referred to as D_{sum} and E_{sum} , based on the approach described in [23].

A new C_k similarity index method has been added to the existing D_{sum} and E_{sum} similarity indices. This represents the correlation between the nuclear data uncertainties in one system with another based on the definition given in [23]. The use of C_k seems to be preferred in some organizations. However, it does introduce a dependence on the covariance data, which means the C_k value has a greater dependence on the choice of nuclear data evaluation than the similarity measures based on sensitivities alone.

A posterior uncertainty estimator using a new Bayesian updating tool has also been added to the Similarity Tool. This can be used to assess the best estimate plus uncertainty values for an application case based on comparison with available validation cases, with due consideration to the similarity of those cases. This uses the Bayesian updating scheme employed in AREVA's MOCABA code [24].

8.2 Validation database analysis tool

The MONK Validation Tool in Visual Workshop gives users direct, searchable access to the results of the MONK validation benchmarks. A new Analysis Tool has been added which allows the estimation of application-specific bias and uncertainty values.

In order to demonstrate criticality safety, it is necessary to show that the maximum k_{eff} of the system does not exceed some safely subcritical value. The maximum value of k_{eff} incorporates not just the calculated value, but additional contributions from biases and uncertainties associated with the calculation and the system. The bias and the uncertainty on the bias must be derived from the code validation evidence. In the UK regulators do not impose a methodology on licensees for calculating these, merely stating that consideration of validation, and uncertainties in calculated values, is necessary. Therefore, a number of different methodologies have been employed in the UK in the derivation of the criticality safety criterion (CSC) used in criticality assessments.

The selection of the most appropriate method to use is left to the criticality analyst. The Analysis Tool however provides the necessary statistical analyses of user-selected validation experiments in order to generate a report containing: the result of a test of whether the supplied set forms a normal distribution; and a series of measures that can be used to identify the systematic bias and uncertainty associated with the type of system analysed.

Three different methods of calculating a suitable CSC are considered in the current Analysis Tool: the E_{PD} , or "error in program and data" method; the systematic bias and uncertainty method; and the Lichtenwalter upper subcritical limit method [25]. As these are standard methods used by criticality analysts they are not described further here.

8.3 Uncertainty tool

The Uncertainty Tool allows the user to investigate the impact on physical results caused by uncertainties in model input parameters in MONK and MCBEND calculations. Given a parameterized input file, the Uncertainty Tool will produce multiple model inputs, varying

the specified parameters according to specified probability distribution functions. After running this series of model inputs, Visual Workshop returns a set of files reporting the details of the run and any results of interest. Visual Workshop is additionally able to plot on a graph the results obtained from an Uncertainty Tool calculation. Recent improvements include allowing the tool to be run in batch mode from the command line, more detailed reporting and automatic results extraction for values such as k_{eff} in MONK and certain tallied response results in MCBEND.

9 Conclusions

Recent developments to the MONK and MCBEND Monte Carlo codes have been introduced. Developments specific to MONK include: an iterated fission probability method for estimating the adjoint flux and adjoint weighted kinetics parameters; a generalised perturbation method of calculating the sensitivity of a range of quantities to nuclear data; a novel algorithm for transferring irradiated geometry from a reactor model to support burn-up credit applications in MONK; and list mode outputs for use in the analysis of subcritical systems using neutron noise techniques. Developments specific to MCBEND include a new pulse height distribution detector resolution model. Developments shared between codes include: enabling imports of IGES CAD geometries; an FG POLY body capable of modelling polygonal surfaces; improvements to the physics modelling; and novel algorithms for modelling stochastic geometries. Developments to the accompanying Visual Workshop software have also been described, including new and improved tools for assisting criticality safety analysts in selecting suitable validation experiments and estimating systematic biases in application cases.

All developments to the ANSWERS codes are customer-driven. Customer feedback is welcomed and forms a key part of our mission to continuously advance our software.

Funding

This work did not receive any specific funding.

Conflicts of interest

The authors declare that they have no competing interests to report.

Data availability statement

This article has no associated data.

Author contribution statement

All listed authors have contributed to the work described in this paper. While Jessica Fildes is the lead author of this paper, Simon Richards and Adam Bird, who are the design authorities of MONK and MCBEND respectively, also made significant contributions to the written content. Editing of this paper was performed by Simon Richards, Adam Bird and Andrew Cox.

References

1. ANSWERS Software Service, MONK: A Monte Carlo Program for Nuclear Criticality Safety and Reactor Physics Analyses. User Guide for Version 12A. Tech. rep., ANSWERS/MONK/REPORT/016, 2024
2. ANSWERS Software Service, MONK: A Monte Carlo Program for Nuclear Criticality Safety and Reactor Physics Analyses. User Guide for Version 11A. Tech. rep., ANSWERS/MONK/REPORT/014, 2021
3. R.J. Brissenden, A.R. Garlick, Biases in the estimation of k_{eff} and its error by Monte Carlo methods, *Ann. Nucl. Energy* **13**, 63 (1986)
4. ANSWERS Software Service, MCBEND: A Monte Carlo Program for General Radiation Transport Solutions. User Guide for Version 12A. Tech. rep., ANSWERS/MCBEND/REPORT/011, 2021
5. ANSWERS Software Service, VISUAL WORKSHOP: The ANSWERS product to prepare and verify models, launch jobs and visualise results for MONK, MCBEND, RANKERN, WIMS and CRITEXUK. User Guide for Version 4B. Tech. rep., ANSWERS/VISUALWORKSHOP/REPORT/008, 2023
6. J.A. Fildes, R.P. Hiles, B.J. Jones, S.D. Richards, New capabilities in MONK for modelling stochastic media, in *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Niagara Falls, Canada, August 2023* (Canadian Nuclear Society (CNS), 2023)
7. B.J. Jones, J.A. Fildes, S.D. Richards, J. Sakurai-Hale, Neutron multiplication and reactor kinetics parameters in stochastic media based on randomized noise function, *Ann. Nucl. Energy* **192**, 109943 (2023)
8. A.J. Cox, C.H. Murphy, S.D. Richards, J.G. Hosking, P.N. Smith, Generalised sensitivities calculations utilising superhistory powering in MONK, in *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Niagara Falls, Canada, August 2023* (Canadian Nuclear Society (CNS), 2023)
9. S.D. Richards, Stochastic mixing of bound thermal scattering data in MONK, *Ann. Nucl. Energy* **136**, 107052 (2020)
10. B. Becker, R. Dagan, G. Lohnert, Proof and implementation of the stochastic formula for ideal gas, energy dependent scattering kernel, *Ann. Nucl. Energy* **36**, 470 (2009)
11. J. Terrell, Distributions of fission neutron numbers, *Phys. Rev.* **108**, 783 (1957)
12. E.R. Woodcock, T. Murphy, T. Hemmings, P.J. Longworth, Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry, in *Proceedings of ANL-7050* (Argonne National Laboratory (ANL), 1965)
13. K. Perlin, An image synthesizer, *SIGGRAPH Comput. Graphics* **19**, 287 (1985)
14. E.G. Gilbert, D.W. Johnson, S.S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space, *IEEE J. Robotics Autom.* **4**, 193 (1988)
15. ANSWERS Software Service, MCBEND: A Monte Carlo Program for General Radiation Transport Solutions. User Guide for Version 11A. Tech. rep., ANSWERS/MCBEND/REPORT/008, 2013

16. A. Bird, A. Kyrieleis, Experience using models imported from cad software for shielding calculations in MCBEND, in *Proceedings of 12th International Conference on Radiation Shielding (ICRS-12) and 17th Topical Meeting on Radiation Shielding (RPSD-2012)*, Nara, Japan (Atomic Energy Society of Japan (AESJ), 2012)
17. A.J. Cox, S.D. Richards, G. Dobson, P.N. Smith, Effective kinetic parameter estimation in MONK using an Iterated fission probability method with superhistory powering, in *Proceedings of International Conference on Physics of Reactors, Pittsburgh, PA, USA, May 2022* (American Nuclear Society (ANS), 2022)
18. A.J. Cox, R.P. Hiles, S.D. Richards, P.N. Smith, Calculating the change in power shape due to localised perturbations to nuclear data using a generalised sensitivities method in MONK, in *Proceedings of International Conference on Physics of Reactors, San Francisco, CA, USA, April 2024* (American Nuclear Society (ANS), 2024)
19. OECD-NEA, *International Handbook of Evaluated Criticality Safety Benchmark Experiments* (Nuclear Energy Agency (NEA), Paris, 2019)
20. A. Nelson, S. Kim, J. Verbeke, W. Zywiec, Subcritical Measurements of Water-Moderated Highly Enriched Uranium Oxide MTR Type Fuel. FUND-LLNL-ALPHAN-HE3-MULT-001. Tech. rep., NEA/NSC/DOC/(95)03/IX, 2019
21. R.P. Feynman, F. de Hoffman, R. Serber, Dispersion of the neutron emission in U-235 fission, *J. Nucl. Energy* **3**, 64 (1956)
22. J.M. Verbeke, J. Randrup, R. Vogt, Fission reaction event yield algorithm, FREYA – for event-by-event simulation of fission, *Comput. Phys. Commun.* **191**, 178 (2015)
23. B.L. Broadhead, B.T. Reardon, C.M. Hopper, J.J. Wagschal, C.V. Parks, Sensitivity and uncertainty based criticality safety validation techniques, *Nucl. Sci. Eng.* **146**, 340 (2017)
24. A. Hoefer, O. Buss, M. Hennebach, M. Schmid, D. Porsch, MOCABA: A general Monte Carlo – Bayes procedure for improved predictions of integral functions of nuclear data, in *Proceedings of International Conference on Physics of Reactors, Kyoto, Japan, September 2014* (Japan Atomic Energy Authority (JAEA), 2014)
25. J.J. Lichtenwalter, S.M. Bowman, M.D. DeHart, C.M. Hopper, Criticality benchmark guide for light-water reactor fuel in transportation and storage packages. Tech. Rep., NUREG/CR-6361, 1997

Cite this article as: Jessica Fildes, Simon Richards, Adam Bird, Andrew Cox, Timothy Fry, David Hanlon, Brian Jones, David Long, Francesco Tantillo, George Wright, and Richard Hiles. Recent Developments to the ANSWERS[®] Monte Carlo Codes MONK[®] and MCBEND[®], EPJ Nuclear Sci. Technol. **10**, 27 (2024)