

THE ANSWERS CODE MONK – A NEW APPROACH TO SCORING, TRACKING, MODELLING AND VISUALISATION

M.J. ARMISHAW, N.DAVIES, A.J. BIRD

Serco Technical Consulting Services,
Kimmeridge House, Dorset Green Technology Park,
Winfrith Newburgh, Dorchester, Dorset DT2 8ZB, UK
malcolm.armishaw@serco.com, nigel.davies2@serco.com, adam.bird@serco.com

A.J. COOPER

Sellafield Ltd,
Hinton House, Risley,
Warrington, Cheshire, WA3 6AS, UK
andrew.j.cooper@sellafieldsites.com

ABSTRACT

The Monte Carlo code MONK[®] has a proven track record of application to the whole of the nuclear fuel cycle and is well-established in the UK criticality community. MONK is part of a continuing development programme comprising the code, nuclear data libraries, validation, guides, geometry model visualisation tools and productivity tools. This paper describes many developments that have been made or are being made to the MONK package, including:

- Enhancements to the geometry modelling capabilities such as CAD import using both IGES models and tetrahedral mesh models.
- Run-time Doppler broadening for the BINGO collision processor, to allow calculations to be performed at any temperature.
- Continuous-energy depletion to further enhance the depletion capabilities of the code in the Burnup Credit application area.
- Parallelisation.
- Extensions to the parameter and formula input facilities to simplify model specification;
- Unified tally option to enable fluxes to be scored in user defined volumes by materials/energy groups and events to be scored in these user defined volumes by reaction type/material/nuclide/energy group.
- Updates to the BINGO continuous-energy nuclear data libraries including JEFF3.1, ENDF/B-VII and CENDL3.1.
- The Integrated Development Environment package Visual Workshop 2 allowing input data to be created and edited, geometries to be displayed and checked, and results displayed over the geometry.
- Productivity Tools - the on-going development of LaunchPad (within Visual Workshop), automatic output checking (MONKcheck), and a facility for launching MONK across a network of computers (CODEMORE).

Key Words: MONK, Doppler, Monte Carlo, Depletion, CAD

1 INTRODUCTION

This paper describes a range of enhancements to the MONK [1] Monte Carlo code that are either complete or under development for the next code release. The programme of continuous development includes not only developments to the code, but also nuclear data improvements, expansion of the validation database, and developments to supporting tools such as Visual Workshop.

The Monte Carlo code MONK has been under continuous development for over 26 years as part of the ANSWERS suite of codes, and for many years before that. MONK is a well-established criticality tool which has a proven track record in applications covering the whole nuclear fuel cycle. MONK also incorporates a broad-group microscopic depletion capability to enable core

depletion calculations to be done. The development of MONK has been carried out as a collaboration between Serco (through the ANSWERS Software Service) and Sellafield Ltd.

The MONK package comprises the code, a range of nuclear data libraries, validation data, supporting documentation, visualisation tools, other productivity tools, and a user support hotline service. The service also includes a range of standard and bespoke training courses given either at Serco premises or at those of the customer. The developments planned for each release of MONK are based not only on input from the collaboration partners, but also on feedback from customers of the ANSWERS Software Service. A wide range of input is sought to provide a balanced short and long-term development programme that meets the needs of the user community.

The current release of MONK, Version 9A(RU1) issued in August 2007, was the first to include the new BINGO continuous energy nuclear data library for criticality calculations, and the revised Datagram broad group nuclear data library for an improved depletion capability. New geometry features for combining and rotating simple bodies were introduced, and the HOLE geometry package was extended to include several new geometry types including COIL, RANDOM RODS, USER and BENT PINS. Auto-Settling and Auto-ZONEMAT options were also introduced to free the user from the need specify the number of settling stages and to identify where the fissile material is located.

Since the release of version 9A(RU1) the MONK package has undergone significant development aimed at the next major code release.

2 CODE DEVELOPMENTS

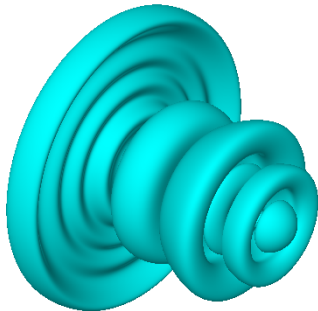
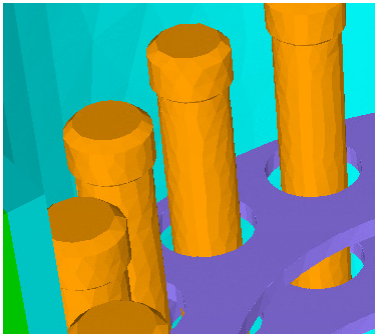
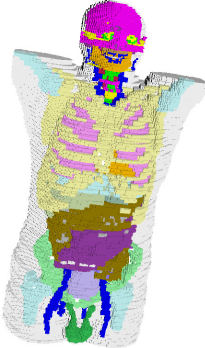
The code developments comprise the introduction of a CAD import capability, changes to the HOLE geometry, collision processing, results tallying, sensitivity analysis, and the depletion capability. There are also changes to the input syntax for MONK to provide the user with greater flexibility when specifying their input data: this includes input parameters, formulae and the looping capability. Other changes include the use of MPI to enable MONK to run across a cluster of computers.

2.1 CAD Import

The development of the CAD import capability (Figures 1-3) includes the option to import an IGES format file using software developed through the Serco/Sellafield Ltd collaboration. The IGES CAD model is handled as if it were a MONK simple body, allowing the user to combine their CAD model with other MONK simple bodies and HOLE geometries to create a more complete model. The software will process the CAD model and, where possible, convert it to Constructive Solid Geometry (CSG) objects, to maximize the tracking speed. The use of 'IGES trimmed NURBS' can also speed up tracking.

Two other routes are available for processing CAD models, but both depend on converting the CAD model to tetrahedral meshes. Such a tetrahedral mesh based data file can be read either by a special simple body geometry, or by the TETMESH HOLE. As above, several such bodies or HOLES can be used in a single model.

The final option for importing models is the VOXEL HOLE. This uses the concept of small rectangular cuboids called voxels (from volume pixels) for modelling geometry and is often used when processing scan data of a human 'phantom'. The use of voxels on a simple XYZ grid in principle allows models to be tracked rapidly, however the quantity of data supplied in a typical voxel model requires data compression and decompression to be used. Methods were developed for the VOXEL HOLE both to enable the large quantity of data in a voxel model to be compressed, and to minimize the impact on tracking speed of the decompression process during tracking through a VOXEL HOLE.

Example imported geometries		
		
Figure 1: IGES	Figure 2: Tetrahedral Mesh	Figure 3: Voxel Hole

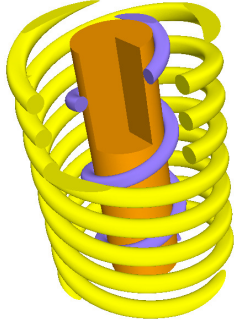
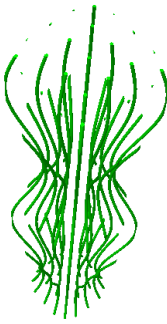
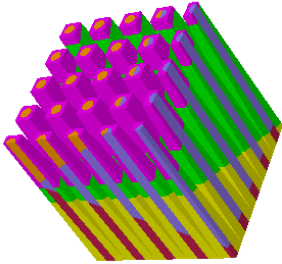
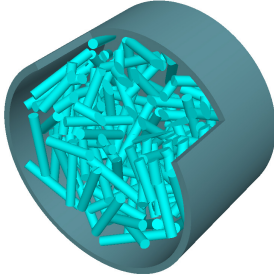
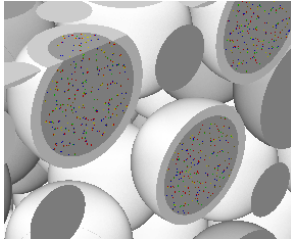
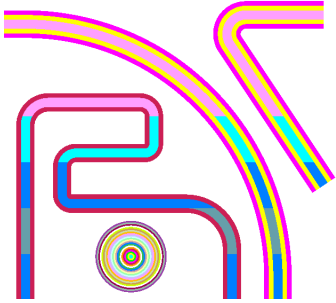
2.2 HOLE Geometries

HOLE geometries use Woodcock tracking [2] to enable complicated arrangements of surfaces to be represented by simple functions. HOLE geometries have been available in MONK for several decades, and have proved robust in providing either all the model geometry, or in combination with simple bodies to provide the additional detail required to accurately model a complex system. MONK includes a range of 18 generic HOLE geometries for simple geometric arrangements such as PLATE, HELIX, SQUARE, LATTICE and TRIANGLE. These generic HOLES, and combinations of them, provide the capability to model in detail most of the systems we see today, however there are cases where new HOLES must be developed to meet specific needs.

Previous MONK papers [1] have described such specific HOLE geometries (Figures 4-9) for modelling PBMR reactors in great detail, RANDOM RODS for modeling cut fuel pins in dissolvers or dropped fuel rods, the COIL HOLE for modeling coiled heat exchangers and the poison wires around fuel pins, and the BENTPINS HOLE for modeling arrangements of pins (some may be missing) that have undergone bending or buckling. The USER HOLE is perhaps the most flexible HOLE geometry available in MONK because it uses a FORTRAN-style interface to enable the user to define the HOLE behaviour during tracking. The USER HOLE example shows a model comprising an array of square pins with central cylindrical rods, with materials varying across the model differently in the pins and rods. The HOLE geometry developments continue to provide new tools to enable the user to model complex systems with ease.

The latest HOLE geometry to be introduced is the PIPES HOLE. This HOLE geometry allows the user to specify in three dimensions any number of pipes with any number of radii, these can be joined automatically either using straight or curved junctions. Subsidiary holes can then be placed within any of these pipes to add further geometric detail to the model. The PIPES HOLE is aimed at those users who wish to remove as many geometric approximations as practical from models of rooms containing complex pipework, pumps and valve systems. An example of the sort of geometry that can be easily modeled using the PIPES HOLE is shown below, showing a subsidiary PLATE HOLE located in all the pipes to set notional fluid levels, the automatic link process (all the curved sections are code-generated), and an example of a cylinder comprising 25 annuli, each of which can contain further geometric detail.

Geometrical simplifications can have undefined effects on the final result, hence tools such as this that allow the user quantify the effect of their assumptions can be used either to justify such assumptions, or to provide an estimate of the bias.

Example Hole geometries		
		
Figure 4: Coil Hole	Figure 5: Bent Pins Hole	Figure 6: USER Hole
		
Figure 7: Random Rods Hole	Figure 8: PBMR Hole	Figure 9: Pipes Hole

2.3 Collision Processing

The MONK code includes three main collision processing packages: DICE point energy, BINGO continuous energy and WIMS broad group, each with their corresponding nuclear data libraries. The WIMS package is used mainly for depletion calculations, though recent developments have enabled BINGO to be used for the tracking phase of a depletion calculation.

The DICE and BINGO packages are those used most frequently for criticality calculations, with the focus of current and future development being BINGO. Since the release of BINGO in version 9A of MONK the BINGO collision processor has undergone a major rewrite. As part of this process the run-time Doppler-broadening capability [3] for treating temperature variations in material data was implemented. This technique does not pre-process the cross-section data for the specific material temperature, but at each collision event evaluates the effect of temperatures on the resonances and derives an appropriate set of cross-section data and event outcomes. Whereas temperature interpolation is appropriate in broad group data, it is considered less so for data tabulated at energy points because the area below the resonance remains constant but moves from the peak to the wings. In the current development version activating the run-time Doppler broadening has a significant impact on execution speed, being a factor of ~8 times slower than running the same calculation with a pre-broadened library. However, run-time Doppler-broadening provides the capability to follow temperature variation throughout a reactor core and throughout its lifetime, opening links to thermal hydraulics feedback.

With this method, the nuclear data library contains, for each nuclide, cross-section data at a relatively small number of specific base temperatures. The nearest lower temperature data to the

user specified temperature is selected, and those data used as the basis for the run-time broadening process. It is thus possible to model any material temperature within the overall temperature range present on the library (currently 293.6K to 80,000K). The accuracy of the run-time Doppler-broadening decreases with distance from the base temperature so a single base temperature cannot be used. Instead extra base temperatures are provided on the library and used before significant error can build up. The distance between base temperatures increases as the cross section becomes smoother with raised temperature.

2.4 Results Tallying

The current version of MONK allows tallying in the ZONES or REGIONS produced by the intersections of the bodies used to construct the model. This places a significant restriction on the user, forcing them to model their geometry specifically to support the tallies required, and to change the geometry if different tallies are necessary. This is poor practice and impacts on the peer review and QA of models.

To address this problem the Unified Tally (UT) input unit has been developed. This new unit allows the user to specify one or more independent meshes that are overlaid on the model geometry. Each of the meshes may have its own energy group structure or share it with other meshes, and each mesh can be subdivided if additional scoring detail is required. In practice several meshes would be defined, each sharing or with their own group structure, and they would overlap other meshes and be subdivided to tally in greater detail. Thus to change the data tallied and where they are tallied, only the UT input needs to be changed and not the underlying geometry.

The Action Tally (AT) input for MONK is used to provide data on the reactions that have occurred during tracking. These are not only by material, energy and nuclide, but also by REGION. These latter tallies are again restricted to the geometry, so the option is being included to allow these tallies to use one or more of the UT meshes to increase the flexibility of AT, and provide multiple tallies from a single calculation.

The meshes supplied to UT are also used by the new depletion capability, again separating the depletion requirements from the geometry requirements. It also provides an independent mesh for monitoring the stability of the flux results by stage using Information Theory processes such as Shannon Entropy.

The aim of this approach is to enable the user to model their geometry in the most efficient way, without having to consider or later modify the geometry to support tallying, depletion or any other operation that can be handled separately.

2.5 Sensitivity Analysis

A first-order sensitivity option for k-effective to cross-section data has been available in MONK for the DICE point-energy nuclear data since version 8B. The approach implemented used Derivative Operator Sampling, as used in MONK's sister shielding code MCBEND, but with the addition of the Limited Weight Recycling method pioneered in MONK version 6.

The sensitivity option in MONK is being further developed so it will be available for use with the continuous energy BINGO library. The sensitivity option in MONK with BINGO will support the sensitivity of k-effective to material density, one or more reaction cross-sections in a specific nuclide in a material, or all available reactions excluding specific listed reactions. A user defined energy group scheme can also be specified for calculating the sensitivity of k-effective to reactions by energy group.

Work is in progress to provide tools to combine these sensitivity results with tables of covariance data to estimate better the uncertainties on k-effective arising from the nuclear data.

2.6 Depletion Capability

As introduced earlier, MONK has a microscopic depletion capability which uses the WIMS nuclear data library for tracking through the geometry and providing the data for solving the depletion equations for the nuclides used. The WIMS nuclear data was chosen because it already included both the yield and the chain data required for a depletion calculation. The rating (in MW) and step length (in days) were used to define the iteration cycles, and the MONK code cycled over each depletion iteration, printing out the k-effective and nuclide number densities after each cycle, until the required number of cycles were completed.

The method available in the released versions of MONK required the user to significantly modify their geometry to support the subdivision of the model into separate burnable regions. Thus a simple geometry could become complicated due to these changes, making input preparation and QA checking more difficult and expensive. This was not considered appropriate and a major development programme was undertaken to separate the depletion requirements from those of the geometry input. This new approach also included the requirement for thermal hydraulic feedback to modify the temperatures and densities of the components.

The new approach allows the user to set up their geometry without considering the depletion or thermal hydraulics requirements, thus existing criticality models can be used with this approach. Once the geometry input has been completed, the new Unified Tally input is used to define a subdivided mesh that can be used to identify the burnable regions, and optionally a second subdivided mesh can be defined that will match the requirements of a thermal hydraulics code, thus providing a mesh for temperature and density feedback into MONK. This approach allows the user easily to modify one or both meshes depending on the calculations being done, leaving the geometry unchanged. With the previous method the geometry would need to be changed, along with the QA consequences of doing so.

A further enhancement has been the addition of the continuous energy BINGO nuclear data library and collision processor to improve the accuracy of the tracking process compared to the 172-group WIMS nuclear data currently used. The increased accuracy of the BINGO method, and the use of the latest JEFF3.1 or ENDF/B-VII libraries, will result in better estimates of both k-effective and the reaction rate data tallied in each cycle of the depletion calculation. The improvements to the reaction rate estimates will provide better data for use in solving the depletion equations and calculate better estimates of the nuclide number densities for the subsequent depletion cycle.

2.7 Named Materials, PARTs and HOLES

MONK developments have been systematically transforming the component numbering system to a component naming system. Historically materials, PARTs and HOLES would need to be numbered so they could be referenced from elsewhere in the input. MONK also includes 'embedded files', these are separate files that include parts of the geometry, material or scoring data, and can have labelled sections to be included at different sections of the main input deck, or not at all. The need to number components has limited the use of these embedded files, so an ongoing programme has been in place to provide an alternative scheme based on named components.

In the new version the user can use named materials, PARTs and HOLES, and by using these in combination with embedded files they allow the construction of a library of such files, each with a QAd section of the model and associated data (e.g. material compositions and subsidiary HOLE definitions). Also, making major changes to a model based on numbered components can become complex, or at best untidy: this process becomes significantly easier when named components can be used and re/numbering is unnecessary.

2.8 Input Syntax

MONK, and the ANSWERS codes MCBEND and RANKERN, include a rich input syntax to provide the user with a range of tools to simplify the specification of sophisticated models. These tools include parameters, formulae, looping and IF...ENDIF tests.

The previous version of MONK introduced double precision parameters and formulae: the new version includes logical and character parameters. The logical parameters enable the user to construct sophisticated TRUE/FALSE expressions for controlling IF tests, while the character parameters give the user greater control over processing labelled data from embedded files described earlier.

The range of functions available in formulae continue to be expanded, including general power functions, the full range of trigonometric functions, log, exp, roots, and an INT function. These can be combined with user defined parameters and values to create sophisticated multi-line formulae, and the result either used directly in the input or assigned to a parameter for further processing. The use of formulae is beneficial because it removes the need to use pre-processing methods that need separate QA: by putting all the calculation steps into the MONK input, along with suitable comments, a fully traceable and QA compliant input can be produced.

The looping option was a powerful addition to the MONK syntax, thus any parameters that are specified with a range of values will cause the code to automatically loop using each of the values in turn. This was improved in previous versions of MONK by the addition of nested looping, and has been taken further in the new version by removing the need to explicitly list each parameter value by using forms such as a:b:c, thus looping between parameter values a and b in steps of length c. Parameters defined earlier in the input, and even formulae, can be included in the list of values to be looped over.

These looping cases run sequentially, thus for cases comprising many loops the total execution time can be significant. The total execution time could be reduced if a large network of computers is available by running all the loops concurrently. The CODEMORE tool (see section 5) is designed to do this and will automatically split up such looping input files into a set of separate files, and then submit each of these files to separate computers over a local network.

With such a powerful looping capability the IF...ELSE...ENDIF test has been expanded. The test now includes the ELSEIF construct, and these IF tests can now be nested to any depth. The expressions used in the IF or ELSEIF tests are no longer just simple expressions. Enhancements to the syntax now include AND, OR and NOT logical operators, parentheses for modifying the expression hierarchy, and loop counters, so specific loops can be identified. As described earlier, logical parameters can be used to create expressions that assign TRUE or FALSE to a parameter that will be used in the IF or ELSEIF tests.

Figure 10 is a small section of input showing the use of nested IF blocks to change the bodies used in a simple model. The loop level numbers (#0 and #1) are available to control loop-specific choices. Figure 11 shows logical parameters being defined then used in conjunction with logical operators, loop parameters and a simple formula to control the program flow through the input.

<pre> @#2out=1;2 @#1in =1;2 begin material geometry part 1 nest IF @out=1 IF @in =1 sphere m1 5 5 5 4 ELSE xrod m1 1 5 5 4 8 ENDIF ELSE IF #0 = 3 AND #1 =1 yrod m1 5 1 5 4 8 ELSE zrod m1 5 5 1 4 8 ENDIF ENDIF BOX m0 0 0 0 10 10 10 end </pre>	<pre> @#2out=1;2 @#1in =1;2 @G=1.;0.;1.;0. @L1= /NOT (#0=2 OR #0=3 OR 4 = #0) @L2 = / #2 = 1 AND 2=#1 @L3 = /@out = 2 AND @in=1 @S = [@G*sin(60.0d0)] begin material geometry part 1 nest IF @L1 AND @S=0.8660253 sphere m1 5 5 5 4 ELSEIF #2=1 AND ((#0=1 OR #0=2) AND (#0=2 OR #0=3)) xrod m1 1 5 5 4 8 ELSEIF @L3 AND NOT @L2 yrod m1 5 1 5 4 8 ELSE zrod m1 5 5 1 4 8 ENDIF BOX m0 0 0 0 10 10 10 end </pre>
Figure 10: Nested IF blocks	Figure 11: Logical Parameters

2.9 Parallelisation

The need to parallelise MONK has been driven by the development of the new depletion capability in MONK, the slowdown associated with the run-time Doppler broadening capability, and the increasing availability of networked machines capable of rapid data transfer.

The MPI library was used to develop a criticality version of MONK that can run over a distributed network of computers. This provided a test of the speed improvements that could be achieved with MONK on such systems, and enabled the prototyping of the capability with the new depletion version of MONK. Thus a MONK depletion calculation that may take several days to run the 61 cycles required to simulate the life of a core, would run in several hours on a 32 core cluster. Larger clusters can be used to run the same calculation in less time, or to calculate reaction data more accurately.

3 NUCLEAR DATA

The nuclear data used during MONK calculations is available in broad group WIMS (172 groups), point energy DICE (13,193 groups), and continuous energy BINGO formats, corresponding to each of the three collision processors discussed in section 2. The point energy libraries will no longer be produced for new evaluations, with the emphasis being on the BINGO and WIMS format libraries. The WIMS broad group libraries used by MONK are identical to those used by the reactor physics code WIMS, and thus include all the data required for depletion calculations.

The continuous energy BINGO libraries are the preferred libraries for MONK criticality calculations because they are considered to provide the available nuclear data in the most detailed form to enable accurate calculations to be performed. This is particularly true with the higher actinides, where the new BINGO methods are superior to the old DICE methods.

Four BINGO libraries are currently available for use with MONK, based on JEF2.2, JEFF3.1, ENDF/B-VII.0 and CENDL3.1 evaluated nuclear data respectively. The BINGO nuclear data library comprises for each nuclide a set of cross-section files at specific temperatures. The JEFF3.1,

ENDF/B-VII.0 and CENDL3.1 libraries incorporate the new run-time Doppler broadening feature [3] and each comprises 11 temperature files for each nuclide covering the range from 293.6K to 80,000K. These temperature files are used by the run-time Doppler-broadening to provide a series of points from which to broaden the cross-sections for intermediate temperatures relative to the nearest lower temperature point.

Table I shows the effect of using the run-time Doppler broadening in the KRITZ experiment [3, 4], compared to calculations using libraries processed for the specific temperatures used in the models. The MONK calculations were run to achieve convergence of 20 pcm. These results show excellent agreement with the fixed temperature libraries for uranium, but for the mixed oxide fuel there is a small discrepancy, which is discussed in [3].

Table I: Effect of run-time Doppler broadening

Core	Fuel	T (°C)	MONK K_{eff}		
			Fixed T	DBF	pcm
2:1	UO ₂	19.7	0.9972	0.9973	-10
		248.5	0.9954	0.9953	10
2:13	UO ₂	22.1	0.9995	0.9993	20
		243.0	0.9980	0.9976	40
2:19	PUO ₂	21.1	1.0020	1.0015	50
		235.9	1.0000	1.0011	-110

4 VALIDATION

The Validation of MONK has been expanded from just the DICE library to include all the BINGO libraries currently available. The MONK validation database comprises analyses based on experimental data, mostly taken from ICSBEP, amounting to around 800 configurations. The systems modelled comprise U, Pu and mixed, in the form of metals, solutions and compounds. The purpose of the database is to provide the user with a measure of the accuracy of MONK for the range of systems analysed.

MONK includes a simple categorization scheme based on six properties that are used to differentiate between system types. The six properties used are: type of fissile nuclide, non-fuel absorption, leakage, resonance absorption, fast fission and hydrogen fuel content. These properties are tallied during tracking, and are thus derived from the neutronic behaviour of the system. The six properties are combined to give a categorization number, and these are listed in the MONK User Guide Validation Section for each configuration analysed. The analyst can use the categorization number from their own calculations with the index supplied in the MONK User Guide to identify quickly comparable systems, from which they can then choose validation cases that most closely match their calculation.

The latest BINGO nuclear data library to be prepared is based on CENDL3.2, and this is being run through the MONK automated test tool. This test tool automatically runs all the MONK verification and validation test cases, compares the results with a set of expected results, and tabulates the outcome of this process on a series of intranet web pages. The comparison also includes a range of diagnostic prints that MONK produces to aid the analyst. The automated testing is designed to take the tedious job of submitting calculations and sorting through the numerous output files away from the analyst, who can then concentrate on just the results and diagnostic prints: the original output files are archived in case they are needed by the analyst.

Tables II to IV list the average calculated/expected (C/E) results for compound, solution and metal systems, for each of the fissile species, with each of the four BINGO libraries discussed

above. The OESD is the Observed External Standard Deviation and shows the spread in C/E values. The tables also list the number of cases in each of the four standard deviation partitions, using the experimental uncertainties, and show that in general the JEFF3.1 library has improved the accuracy of the calculated results compared to the industry standard JEF2.2 library.

Table II: Uranium Systems

		Avg C/E	OESD	<1σ	>1σ <2σ	>2σ <3σ	>3σ
COMP	JEF2.2	1.0010	0.0055	144	86	30	26
	JEFF3.1	1.0003	0.0040	197	66	15	8
	ENDF/B-VII.0	1.0011	0.0039	208	53	7	18
	CENDL3.1	0.9979	0.0042	130	102	40	14
SOL	JEF2.2	1.0036	0.0069	35	35	14	8
	JEFF3.1	0.9992	0.0069	43	28	11	10
	ENDF/B-VII.0	0.9993	0.0064	44	28	12	8
	CENDL3.1	0.9989	0.0065	50	23	12	7
MET	JEF2.2	0.9971	0.0131	18	17	11	27
	JEFF3.1	0.9987	0.0055	27	22	7	17
	ENDF/B-VII.0	1.0003	0.0054	40	12	11	10
	CENDL3.1	0.9974	0.0064	23	20	15	15

Table III: Plutonium Systems

		Avg C/E	OESD	<1σ	>1σ <2σ	>2σ <3σ	>3σ
COMP	JEF2.2	1.0128	0.0098	1	0	0	5
	JEFF3.1	1.0178	0.0055	1	0	0	5
	ENDF/B-VII.0	1.0206	0.0050	0	1	0	5
	CENDL3.1	1.0193	0.0112	1	0	0	5
SOL	JEF2.2	1.0044	0.0077	81	59	8	25
	JEFF3.1	1.0048	0.0077	75	64	10	24
	ENDF/B-VII.0	1.0053	0.0070	76	64	10	23
	CENDL3.1	1.0149	0.0064	0	23	60	90
MET	JEF2.2	0.9989	0.0066	10	4	3	3
	JEFF3.1	1.0008	0.0053	8	8	2	2
	ENDF/B-VII.0	1.0021	0.0067	8	3	5	4
	CENDL3.1	1.0044	0.0077	3	6	5	6

Table IV: Mixed Pu/U Systems

		Avg C/E	OESD	<1σ	>1σ <2σ	>2σ <3σ	>3σ
COMP	JEF2.2	0.9992	0.0086	47	24	3	7
	JEFF3.1	1.0021	0.0084	61	12	1	7
	ENDF/B-VII.0	1.0034	0.0077	58	16	0	7
	CENDL3.1	1.0073	0.0086	27	32	13	9
SOL	JEF2.2	0.9987	0.0063	37	15	8	16
	JEFF3.1	1.0001	0.0063	34	22	5	15
	ENDF/B-VII.0	1.0005	0.0062	40	17	7	12
	CENDL3.1	1.0077	0.0068	19	21	18	18
MET	JEF2.2	1.0001	0.0021	19	4	0	0
	JEFF3.1	1.0004	0.0023	20	3	0	0
	ENDF/B-VII.0	1.0028	0.0019	12	9	2	0
	CENDL3.1	1.0056	0.0026	4	8	10	1

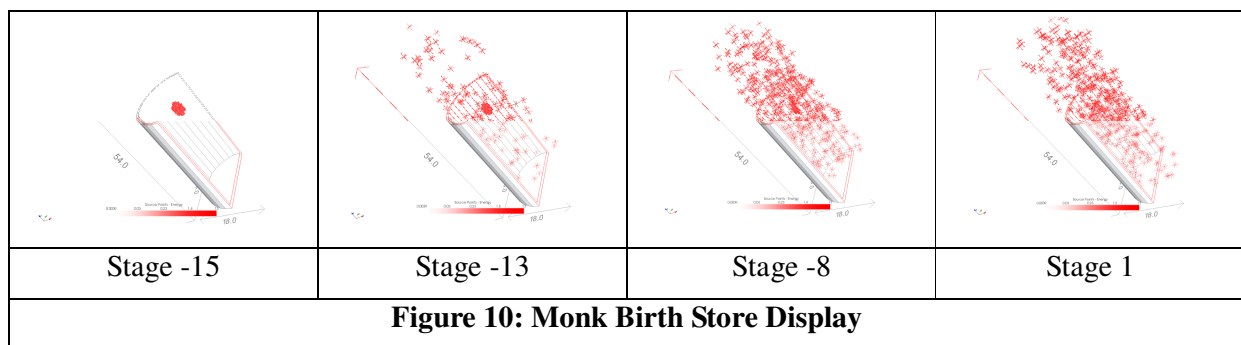
5 VISUALISATION AND PRODUCTIVITY TOOLS

Visual Workshop is a productivity tool used in conjunction with ANSWERS codes such as MONK. It is used by analysts to display and check their MONK geometry input data interactively, and to run either one or many MONK calculations using the built-in platform independent queue tool. It includes a fully featured editor for modifying the input, and will automatically redraw an image when the data is saved. The latest version underdevelopment includes results and log file displays.

The job launch and queue tool capabilities simplify the submitting of calculations to each of the platforms Visual Workshop can be runs on, which includes Windows PC and linux. The space required to run the code and the nuclear data library can be selected: the material composition input of MONK is generic, hence different libraries can be used with a single input file.

Visual Workshop includes 2D and 3D ray-trace capabilities, and a 3D wire-frame display option. The ray-trace options use the tracking routines from the MONK code version used for the later analysis, and produces a display based on what the MONK tracking process sees. There are also options to search for specific geometry modelling user errors, and to display volumes of interest. The ray trace options also give excellent rendering of HOLE geometries as seen earlier in section 2.

The latest version of Visual Workshop has undergone major developments since the previous release. These developments include: improvements to the Wire Frame display with the addition of results visualisation features including 3D and 2D contouring of results; display of particle tracks and event locations, allowing the display of source points at the end of each stage can be displayed and used to monitor the source convergence process using the animate feature. The use of the MONK tracking algorithms in the ray trace options enables the user to visualise CAD models imported into MONK through the IGES route, the tetrahedral HOLE or PART, and the voxel HOLE routes.



Two separate tools are also available for directly running and processing MONK output data when large numbers of calculations must be run. The first is a code called CODEMORE that is used to take a single MONK input file that uses looping, and to split it up into separate input files and submit them to a cluster of networked computers. This enables calculations comprising several hundred loops to be run concurrently, significantly reducing the elapsed time for the user to receive their results. The second tool is called MONKcheck, and is used to parse folders of output files, extracting the results, and searching for error and warning messages that can be used to determine whether the calculations were performed correctly. The output from this tool is a simple file that quickly identifies to the analyst those cases that need further investigation.

6 CONCLUSIONS

The MONK Monte Carlo criticality and reactor physics code has undergone a significant number of developments over the last few years. These are part of the continuing support for MONK as one of the major codes in the ANSWERS Software Service of Serco. The code has a varied and powerful geometry specification package, comprising simple bodies, HOLES, and now CAD import. MONK can use a range of nuclear data libraries, and this enables the effect of different evaluations on k-effective and other estimators/tallies to be determined.

The input syntax available to the analyst is also undergoing development to simplify specifying ever more sophisticated models and data: the aim is to not only reduce the user effort in preparing such models, but to reduce the QA burden by making models easier to check.

The validation database supplied by ANSWERS is continuously reviewed and updated. The aim is to provide the user with the latest data on the accuracy of the MONK code for a wide range of system types.

The sophisticated visualization, input editor, results viewer, and job launch tool provides the user with methods for preparing, checking, running and viewing their MONK data. The input editor and visualization options not only ensure that an input is syntactically correct, but also enable the user easily to identify and deal with any geometry data problems. The job launch tool provides a simple interface to enable the user to launch one or more cases, and to monitor the job as it runs. The output display capabilities can be used to monitor the source convergence process in MONK, as well as view tallies accumulated during tracking.

The codes CODEMORE and MONKcheck provide the user with tools to make use of clusters of networked computers to launch large numbers of calculations, and then to parse the outputs from those calculations both to extract the required results and to search for keywords associated with error and warning messages: all these data are put in a single file for easy analysis by the user.

7 ACKNOWLEDGMENTS

This acknowledges the significant input from Geoff Dobson, Ray Perry, Simon Richards, David Hanlon and Christopher Dean who have all contributed to the development of the MONK Monte Carlo criticality and reactor physics code and its associated data libraries. Also, this acknowledges the major contribution from Sellafield Ltd to the development of the BINGO and CAD options of MONK, their contributions to Visual Workshop development, and for their codes MONKcheck and CODEMORE, all as part of the collaboration with Serco.

8 REFERENCES

1. Armishaw, M J, et al "Current Status and Future Direction of the MONK Software Package", *Proceedings: 8th International Conference on Nuclear Criticality Safety, ICNC 2007, St Petersburg, Russia (May 2007)*.
2. Woodcock, E R, et al "Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry," *Proceedings: Conference on the Application of Computing Methods to Reactor Problems, ANL-7050, pp.557-579, Argonne National Laboratory (1965)*.
3. Dean, C J et al "Validation of Run-time Doppler Broadening in MONK with JEFF3.1", *Proceedings for the International Conference on Nuclear Data for Science and Technology, Korea, 2010*.
4. Johansson, E "Data and Results for KRITZ Experiments on Regular H₂O/Fuel PIN Lattices at temperatures up to 245C", STUDSVIK/NS-90/133, JEF/DOC-329.