Bringing service to life

NEW FEATURES

A Guide to the New Features of MONK Version 9A

ANSWERS/MONK/REPORT/004

MONK is developed and maintained through a collaboration between Serco Assurance and British Nuclear Group Sellafield Limited

Issue	Date	Description
1	Jul 2006	Original issue for MONK version 9A

Issued by: M J Armishaw

1 Introduction

MONK Version 9A is now available from the ANSWERS Software Service. This new version incorporates a range of new and enhanced features. This report provides an introduction to their use and is directed at existing MONK code users. Further information for all the features introduced here can be found in the MONK User Guide for version 9A [1].

The new features of MONK Version 9A (as described in section 2) comprise:

1. FG - Window Part
2. FG - OVERLAP Part
3. FG - Zone Complement
4. FG - OR Operator
5. FG - CONTAINED and CONFINED
6. FG - Rotation Options
7. FG - Relative Body Origins
8. Unified Source
9. ZONEMAT – Named Parts and Materials
10. Updated Material Specification Module
11. New Datsets (Paths)
12. HOLE GEOMETRY Input Unit
13. USER Hole
14. COIL Hole
15. RANDOM RODS Hole
16. BENT PINS Hole
17. Triangle Hole Character Map
18. Named Holes and Named Materials in Holes
19. STDV First Stage Check
20. RUNID
21. JOB LOG
22. Expected Value Option
23. SHOWK
24. INFLOWS Tally
25. AUTO SETTLING
26. AUTO ZONEMAT
27. Syntax Improvements
28. New Formula Functions and Options
29. BINGO
30. CFREE and CBOUND
31. Datagram
32. Burnup Predictor-Corrector
33. NGOTO
34. Flux Scaling
35. Revised COMPONENT MULTIPLICATION

2 Guide to New Features

2.1 FG - WINDOW PART

The current PART option in MONK requires that the body containing the part include no other structures within it. The 'E' material option can be used in some cases to model system where such overlapping is necessary, but it is not always clear how to implement it correctly. If there was a mismatch between the components this was not always easy for the code or user to determine.

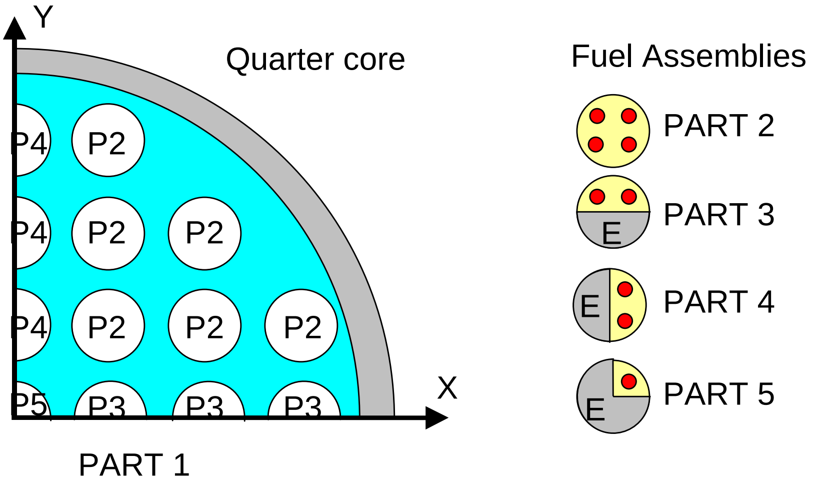
The new WINDOW PART is designed to overcome these limitation by allowing the subsidiary PART to be partially obscured by zones in its parent PARTs, and by removing the need for the 'E' material improve the ability of the code to check for errors. The input to define a WINDOW PART is made when the PART is referenced – i.e. in its parent PART.

An example input would be of the form:

```
BOX W2 0.0 0.0 0.0 100.0 200.0 200.0
```

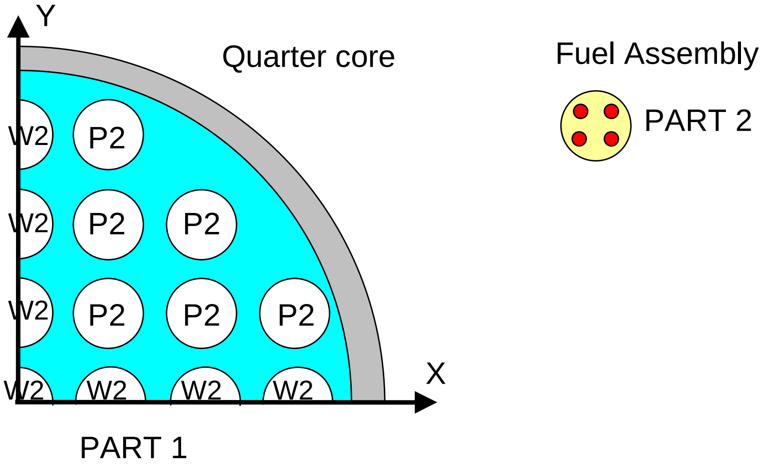
In this case PART 2 will be inserted into the BOX as a WINDOW PART (W2 instead of the usual P2), thus allowing any other zones in this PART (or parent PARTs) to obscure some of its contents.

As an example of moving from using material 'E' to the WINDOW PART, consider modelling a quarter core using normal parts:



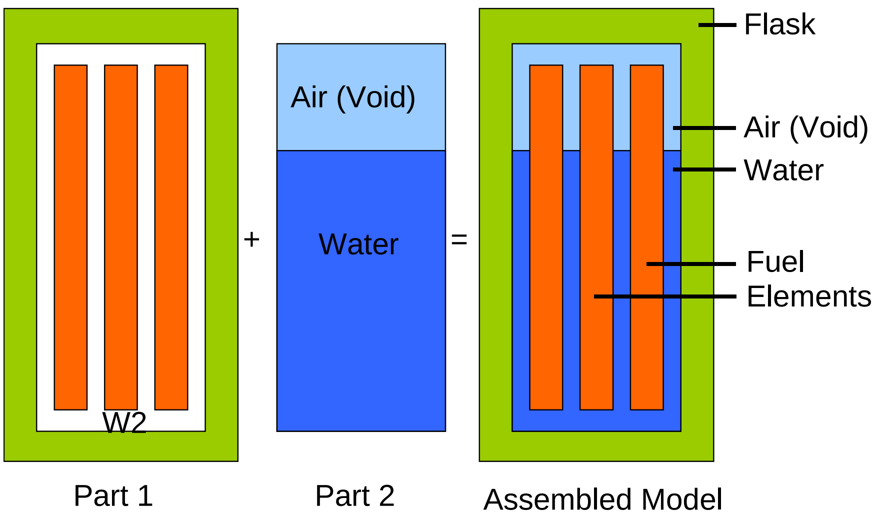
In this case it is necessary to define four PARTs to handle the visible portions of the fuel assemblies, three of which require material 'E'. These additional PARTs are an overhead and each is a possible source of error.

The WINDOW PART removes the requirement to create additional PARTs and thus simplifies both the model and its subsequent checking. Using the WINDOW PART the model becomes:



Now there is a single subsidiary PART for the fuel assembly that can be used in all locations within the quarter core.

Another example could be to model water levels in a flask. The usual solution in these cases is either to use the GENERAL PART or to use a PLATE HOLE. The WINDOW PART provides an alternative that will prove to be a better option in some cases:



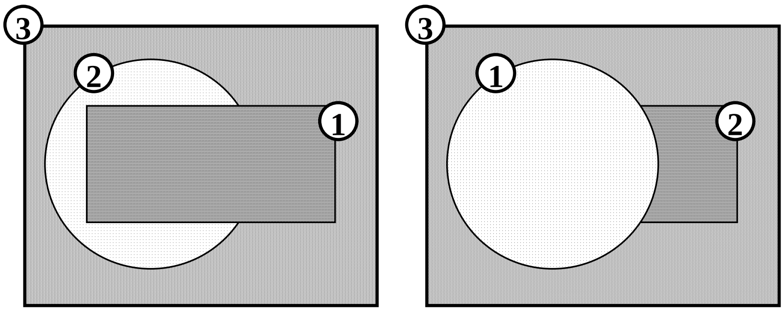
In this example the Fuel Elements in the parent PART obscure the Air and Water in the subsidiary PART.

See Chapter 9 Geometry Modelling, Section 4.2 for further details.

2.2 FG – OVERLAP PART

This is a new structure in the same category as a NEST or CLUSTER. Each body in an overlap part is overlapped by all bodies earlier in the list of definitions. Bodies are defined in a specific order. The last body defined thus becomes the part container. The surfaces of the internal bodies may touch one another and may overlap.

Some examples (in two dimensions) are sketched below:



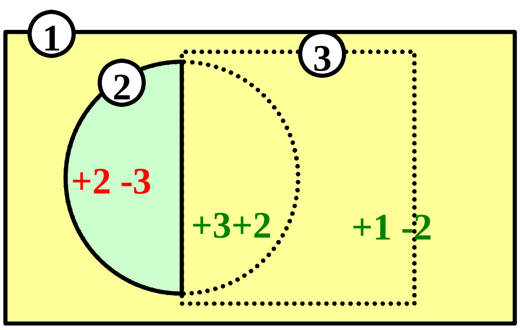
Each example contains three bodies – the container being body 3. The bodies are identical, but they are defined with the sequence indicated above, resulting in different zones. This illustrates the significance of sequence number when defining bodies.

The zones in an overlap part are automatically defined as the difference between the current body and all previously specified bodies (or the inside of the first body). Since there is an inherent one-to-one correspondence between bodies and zones in an overlap part, it is not necessary to number the bodies and zones independently.

See Chapter 9 Geometry Modelling, Section 3.3 for further details.

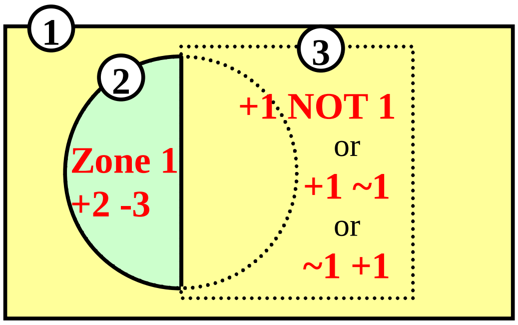
2.3 FG - ZONE COMPLEMENT

This new option of FG allows complete zones to be excluded from the definitions of other zones. This is designed to simplify the specifications of zones comprising many bodies, typically the interstitial volume that remains after all material zones are defined. For example consider the following system comprising three bodies:



If we define zone 1 to be +2 –3, the remaining area must normally be defined as two distinct zones (+3+2 and +1-2). It would be convenient to say that the remaining area is ‘everywhere that is NOT zone 1’. The NOT (or ~) operator allows us to do this.

An example of using the NOT operator on the above example would give:



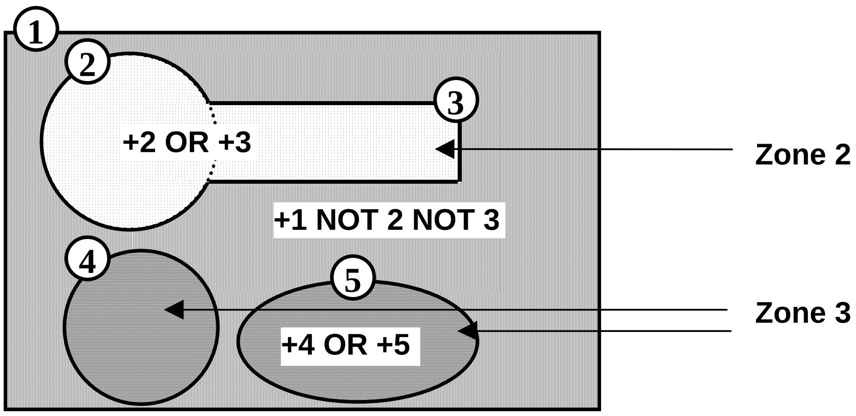
So, as above, zone 1 is defined as +2 –3, but now we can define the remaining area as +1 NOT 1 (or as +1 ~1, or even ~1 +1). Note that the number after the NOT (or the ~) refers to a previously defined *zone*.

Note that there is a known problem when using this option with the TORUS and SECTOR bodies – refer to the appropriate error memo for details.

See Chapter 9 Geometry Modelling, Section 3.4 for further details.

2.4 FG – OR OPERATOR

The OR operator allows a zone to be defined as the *union* of two or more bodies. The example below illustrates its use:



Within the code, the OR operator effectively generates two (or more) distinct zones with common properties: content, region number, sequence number etc. It is a form of legitimate multiple definition. A zone defined with the OR operator cannot contain a *body hole*, a *subsidiary part* or a *window part*. (There is no single, identifiable body to act as a container for these contents.)

See Chapter 9 Geometry Modelling, Section 3.4 for further details.

2.5 FG – CONTAINED AND CONFINED

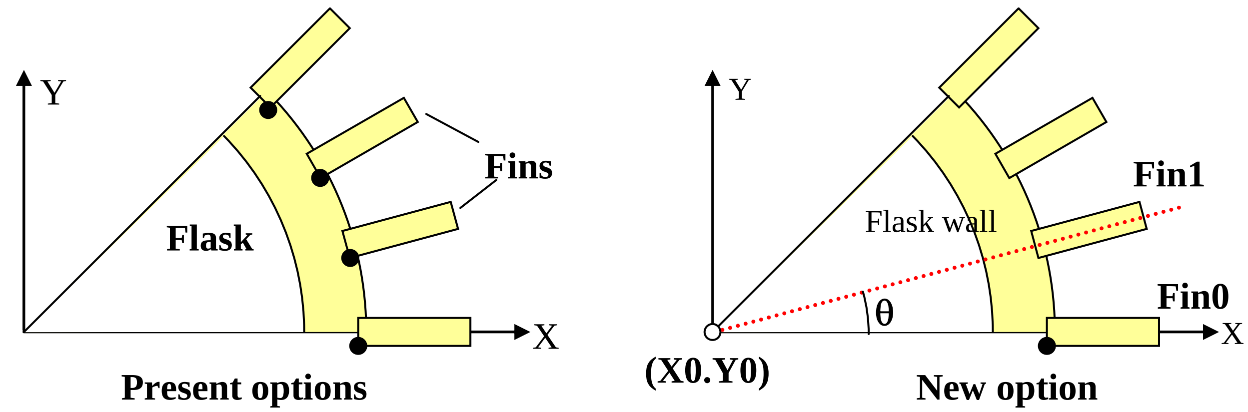
When included in a general part, the CONTAINED qualifier declares that all inner bodies are intended to be within the confines of the container body. The code checks analytically that this condition is met.

When included in a general part, the CONFINED qualifier declares that all zones are to be restricted to be within the container body. If the container body has sequence number n: all zone definitions are automatically extended by the characters '+n'. Consequently the zones cannot contain a *subsidiary part*, although they can contain a *window part*.

See Chapter 9 Geometry Modelling, Section 3.4 for further details.

2.6 FG - ROTATION OPTIONS

This option allows a body to be rotated about point other than its origin.
In the left hand sketch below, the origin of each fin must be computed explicitly.
In the right hand sketch, all fins have the same initial position as fin0 and are rotated about the centre of the flask



The option may only be used in conjunction with simple rotation about a co-ordinate axis. E.g.

body data (shape, position etc.)

```
ZROT 15 ABOUT 3.6 9.8
```

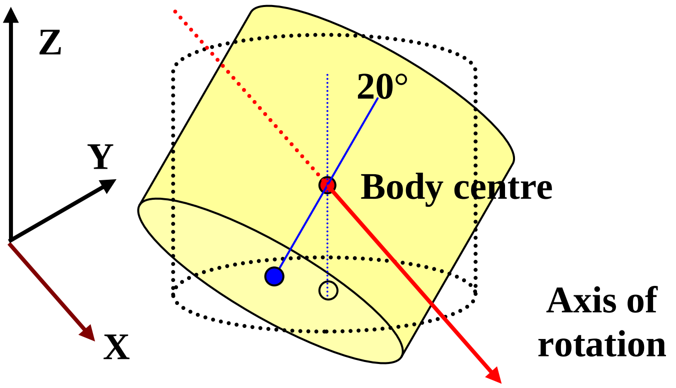
This requests a 15° rotation about an axis parallel to the Z axis and passing through a point with co-ordinates X=3.6; Y=9.8

If the new data ABOUT… are omitted then the original default (rotate about the body origin) applies.

A variation allows the centre of rotation to be the geometrical centre of the body, e.g.

```
XROT 20 ABOUT CENTRE
```

It may only be used for bodies with an unambiguous centre. (e.g. not for a CONE, PRISM or HEMI)

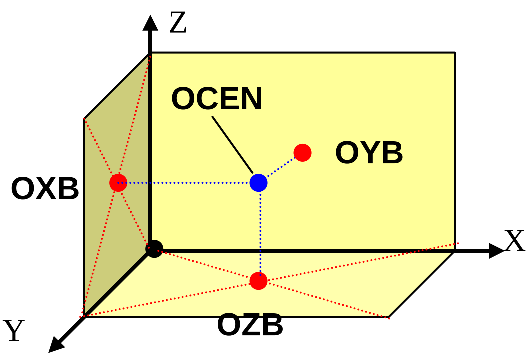


2.7 FG – RELATIVE BODY ORIGINS

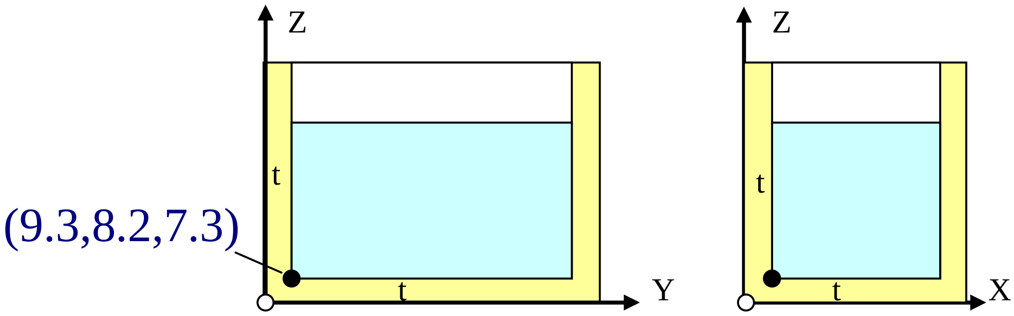
Arrangements of bodies in FG PARTS currently have their origins relative to some local point within the local PART. In some situations it is necessary to change several origins to make a minor rearrangement of the model. This new option allows the user to specify the origin of a body relative to the previously entered body. This it is necessary to edit only one origin – the 'relative' bodies will then move with it.

The origin data may be prefixed with a keyword to reference alternative points in the body.

OXB	point at centre of minimum X face
OYB	point at centre of minimum Y face
OZB	point at centre of minimum Z face
OCEN	point at geometrical centre of body.



Thus the origin of a body may be defined by its position relative to that of the previous body, the keyword prefix for such an origin is OREL
The following example uses a NEST to define a water tank with wall thickness t (=0.5, say)



The input for the above model would be:

```
BOX      9.3 8.2 7.3 etc. water volume
BOX OREL 3*0.0       etc. Air volume with same origin
BOX OREL 3*-0.5      etc. Tank (container)
```

Some points to note about this new option are:

- The recorded origin of a body remains in its conventional place. E.g. for a BOX it is always the corner.
- The keyword prefix for a relative origin (OREL) may be used for any shape of body.
- keyword prefix for a central origin (OCEN) may be used for any body with an unambiguous centre. Thus not for a CONE, PRISM or HEMI

2.8 UNIFIED SOURCE

A new Unified Source specification input unit has been included that combines the capabilities of both the Simple and Complex source input units. The function of the Unified Source module is to select the starting parameters of each new source particle: its position, energy, direction and weight. It is likely that the existing SOURCE GEOMETRY ZONEMAT option will continue to provide most of the functionality required in MONK, however this new option is available if required.

The input data supplied by the user define the distribution of source in the problem from which these parameters may be sampled. The data divides into the following broad categories.

2.8.1 Preface

This section includes items such as seeds for the random number generator. It may often be omitted.

2.8.2 Geometry

This section defines the shapes and positions of the sources using a set of geometrical volumes or surfaces: cylinders, cuboids, discs etc. Source bodies may be subdivided into smaller volumes (or areas) within which the source can be assumed constant.

The source geometry has close links with the Fractal Geometry (FG) module that is used for defining the distributions of materials in the problem. The source geometry can therefore be defined by reference to Fractal Geometry PARTS, BODIES and/or ZONES.

2.8.3 Energy

This section defines the group schemes used for specifying the variation of source intensity with energy. Note that libraries of fission and activation product spectra are not yet available in the Unified Source module.

2.8.4 Angles

In most cases, source particles are emitted isotropically from a selected point. This section is required exceptionally to specify an angular quadrature in which anisotropic sources may be defined.

2.8.5 Intensity

The above sections essentially divide space, energy and direction into a set of *cells* within which the source can be considered constant. This section defines the source intensity within each cell.

2.8.6 Weighting

It is often fruitful to bias the sampling of source parameters to give preference to those that are more likely to contribute to the scored results. This (optional) section assigns a relative importance to each source cell. Note that the energy group schemes do not need to be compatible to use automatic source weighting.

2.8.7 Examples

An example input that references previously defined FG zones could be:

```
BEGIN UNIFIED SOURCE DATA
  GEOMETRY                        ! specify the source geometry:
  FGZONE 3 IN PART 1              ! the source bodies refer to
  FGZONE 4 IN PART 1              ! the FG zones/parts
  FGZONE 5 IN PART 1
  FGZONE 6 IN PART 1
  ENERGY                          ! specify the energy data:
  SOURCE GROUPS                   ! using a histogram…
  14.6 5.0 1.0 0.1 0.0674         ! then the energy boundaries
  SPECTRA                         ! and the source spectrum
  0.1 3*100.0
  INTENSITY                       ! and the source intensity
```

```
      BODY 1                        ! for each body listed above
      COMPONENT 100.0
      BODY 2
      COMPONENT 80.0
      BODY 3
      COMPONENT 60.0
      BODY 4
      COMPONENT 30.0
      WEIGHTING AUTOMATIC          ! use automatic source weighting
    END
```

The example shows FG zones 3 to 6 of PART 1 being identified as source bodies (BODY 1, 2, 3 and 4 further down the input unit). After the source bodies are identified the energy boundaries and spectra are defined (in this example a single histogram is defined). Then, for each source body defined, the source intensity is entered. Finally automatic source weighting is specified to enable the code to concentrate on the more important source regions.

This second example makes use of more of the features in the Unified Source. The example begins by using the PREFACE section to provide seeds to the random number generator. In the GEOMETRY section the bodies are explicitly defined and include source body subdivisions. To assist in specifying the subdivisions the **I**, **F** and **P** operators are available to allow the user to specify the **I**ntervals between boundaries, **F**ractions of the total width between boundaries, or the boundary **P**ositions explicitly. The ENERGY data comprises a line spectrum

```
BEGIN UNIFIED SOURCE DATA
   Preface seeds 12345 54321     ! define the RNG seeds
   Geometry
   +ZHEMI One 10.0 10.0 10.0 4.5 ! define the first body ‘One’
   R 2.5 P                       ! and its subdivisions
   THETA 4*90.0 I DEGREES
   -ZHEMI Two 10.0 10.0 10.0 4.5 ! and then body ‘Two’
   R 2.5 P
   THETA 45 90 135 180 225 270 P DEGREES
   ENERGY                        ! Specify the energy data
   LINES 6.0                     ! for a line spectrum
   SPECTRA 1.0
   INTENSITY                     ! and the source intensities
   BODY one S 1                  ! for the subdivisions in each body
   COMPONENT R 0.0 1.0
   COMPONENT THETA 1.0 2.0 0.0 3.0
   Body two S 1
   COMPONENT R 0.0 1.0
   COMPONENT THETA 1.0 3.0 2.5 4.0 0.0 5.0 3.0
END
```

Refer to Chapter 4 Source Options, Section 3.5.1 in the MONK version 9A User Guide for detailed information on the Unified Source module.

2.9 ZONEMAT – NAMED PARTS AND MATERIALS

The SOURCE GEOMETRY ZONEMAT option has been enhanced to allow the use of named parts and named materials. Thus a typical format input string could be:

```
ZONE 3 PART 2 / MATERIAL 17
```

This can now be entered (assuming named parts and materials are used) as:

```
ZONE 3 PART fuel_element / MATERIAL fuel_u235
```

Note that as described in the user guide both the PART names and material names are restricted to 20 characters, they must contain only the characters a-z, A-Z, 0-9 and '_' (underscore), and the name must not start with a digit (0-9).

It is recommended that the SOURCE GEOMETRY input unit always follows the MATERIAL SPECIFICATION and MATERIAL GEOMETRY input units when named materials and parts are used.

2.10 UPDATED MATERIAL SPECIFICATION MODULE

The library materials have undergone minor revisions, principally the composition of Air. It is still the recommended advice that if a library material is to be used, then the composition of the material (listed in the user guide) is checked to ensure it is consistent with the composition required. In some cases (e.g. some concrete types) there is no universally agreed composition.

For certain materials the bound hydrogen data has been changed from HINH2O to HINCH2 to reflect the dominant binding nuclide.

If named materials are used it is recommended that the MATERIAL SPECIFICATION input unit be the first input unit in the input file.

2.11 NEW DATSETS (PATHS)

The capability of the DATSETS method of identifying input, output and library files to MONK has been enhanced to allow the definition of PATHS. A PATHS file can now be set up which allows the user to assign paths to single character variables. These variables can then be referenced in the DATSETS file for the specific cases to be run. The advantage of this is that any change in location of input, output or libraries can be trivially handled by a corresponding change to the PATHS file. For example the PATHS file might be called P1.lis and contain:

```
I=/mctest/M2/IN/
O=/mctest/M2/OUT/
L=/answers/data_libraries/
```

And in the datsets file we have:

```
INPUT  $Itest01.dat
OUTPUT $Otest01.out
MATDB  $LMONK_matdb.dat
DICE   $Ldicedat1c.dat
```

When the code is run the command line instruction ‘paths=P1.lis’ is added (in the same way as for space=) to identify to MONK where the variables are defined. Thus by simply changing the entry in the PATHS file the outputs from several runs can be directed to different directories. Alternatively three PATHS files could be set up to fully define both the libraries and the correspondingly named output file directories for each nuclear data library (JEF, JENDL, ENDF), thus allowing a single set of inputs and datsets to be set up and run with the output for each nuclear data library appearing in its own directory.

2.12 HOLE GEOMETRY INPUT UNIT

The old HOLE DATA input unit has been deprecated and the HOLE GEOMETRY input unit written to replace it. The input of the new input unit is similar to that of the old input unit, but now any material must be preceded by an M, and any subsidiary hole preceded by an H. The use of negative values to identify subsidiary holes will generate an error in this input unit.

There is a special keyword, NOMORH (no M or H), that can be used in the SQUARE, LATTICE and TRIANGLE holes to allow the old style input when inputting the array of pin contents only – it does NOT make the old option applicable throughout the hole.

Consider a list of materials in the old style input:

```
  1   2   4   0  -3   4  -6
M 1 M 2 M 4 M 0 H 3 M 4 H 6
```

these would now become:

The space between the M or H and its following number is not mandatory.

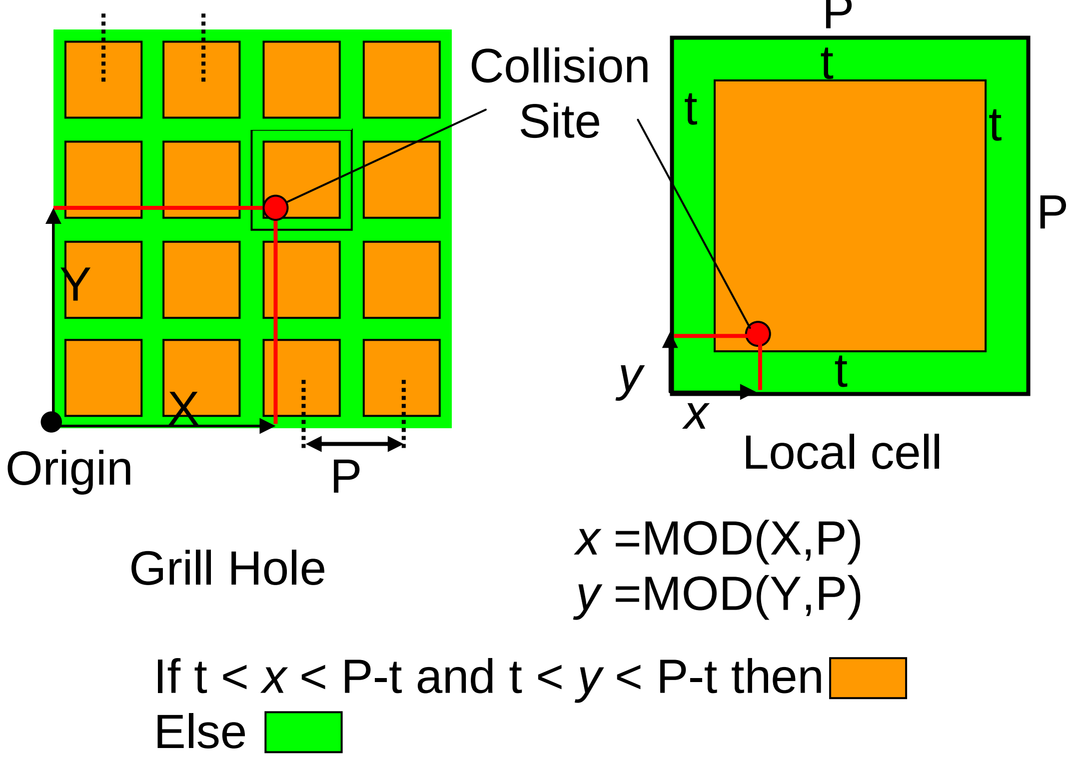
Future hole developments will only be made to the HOLE GEOMETRY input unit, thus the HOLE DATA facility will not contain the latest hole types. This has already occurred for the USER, COIL, Random Rods and Bent Pins holes described below.

2.13 USER HOLE

The USER hole uses a BASIC-like syntax to define a 'designer hole' for specific tasks. As a simple example a PLATE HOLE could be simulated using:

```
BEGIN HOLE GEOMETRY
  HOLE 1 USER
  If Z < 6 Then
    M = 1
  ELSE
    M = 0
  ENDIF
END
```

Intermediate variables are available, and there are reserved variables for the current X, Y and Z coordinates: @.X, @.Y and @.Z. As an example consider the following model:



This could be modelled in a simple form by using the following input:

```
BEGIN HOLE GEOMETRY
HOLE 1 USER
M 2                                 !set duct material 2 as the default
P = 2                               !Pitch – define a local variable P
T = 0.2                             !Half thickness of web
U = [MOD(@.X,@.P]                   !Local x,y in grid cell, reference P
V = [MOD(@.Y,2]
IF U < T OR  U >= [@.P-@.T] OR
   V < T OR  V >= [@.P-0.2] THEN
    M 1                             !use web material 1
ENDIF
END
```

Local variables are referenced by preceding them with ‘@.’ (at then dot) instead of just ‘@’. IF..ELSE..ENDIF is available and can be nested. Functions available include INT(), MOD() and ATAN() – the user guide lists all those available.

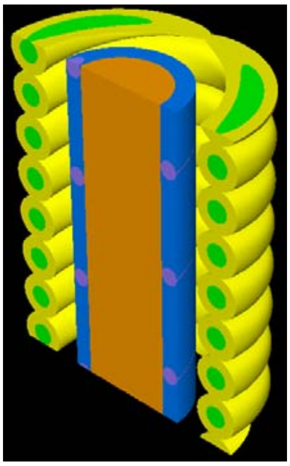
A note of warning regarding the speed of the USER hole is necessary. Due to the interpreted nature of this hole, systems that use it can run much more slowly than if they were modelled without the user hole. Thus it is important to recognise this limitation, and to use this hole when no other option is available.

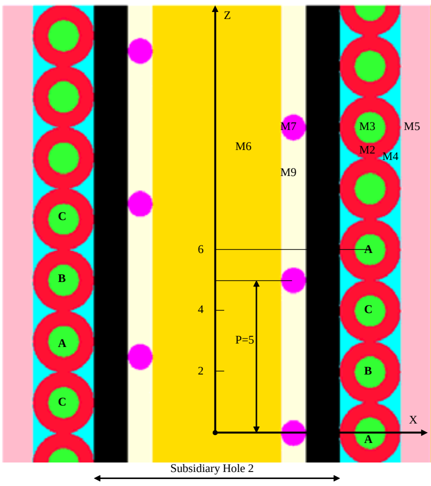
Perhaps more importantly is a warning regarding the QA of any USER hole inputs written. Since the user of the code is writing the input for this hole, it is their responsibility to ensure that their input produces the correct model. To achieve this tools such as VISAGE and VISTA-RAY must be used to check that the hole functions as the user wishes.

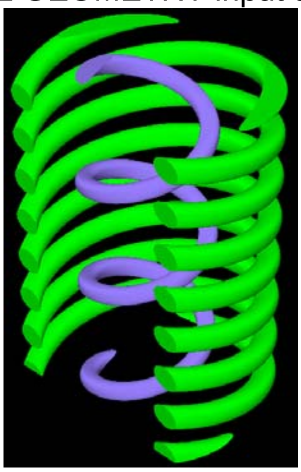
Refer to Chapter 3 Geometry Modelling, Section 13.19 of the MONK version 9A User Guide for more details on using the USER hole.

2.14 COIL HOLE

The COIL hole allows the user to model sets of one or more helically coiled pipes. The pipes can be either left or right handed (the default), and can be either solid or have an outer layer. It is only available from the HOLE GEOMETRY input unit.







An example input for a coil hole could be:

```
HOLE 1 COIL
 ORIGIN 7 7 1
 LEFT HANDED
 MEAN  5.0
 INNER 0.5
 OUTER 1.0
 COILS 3
 PITCH 6
 MATERIALS H2 M2 M3 M4 M5
```

This is a left handed coil with a mean radius of 5.0cm. The pipes have an inner radius of 0.5cm and outer of 1.0cm. There are three coils in the hole and they have a pitch of 6cm. The MATERIALS section demonstrates how to enter the material data and the option to include subsidiary holes.

Refer to Chapter 3 Geometry Modelling, Section 13.2 of the MONK version 9A User Guide for more details on using the COIL hole.

2.15 RANDOM RODS HOLE

The RANDROD (random rods) hole allows the user to model the distribution of rods in a cylindrical container. The container is filled with rods all of the same length, radius and composition. The rods can be made up of a number annuli of different materials. The rods will not be cut by the container surfaces.

The model requires the dimensions of the cylinder and the rods (radius and axial length). The default orientation for the container is with the axis aligned with the Z axis. It is likely that the RANDROD hole will be contained in an FG ZROD.
The number of rods is also specified, as is the desired packing fraction. From this the code calculates the height of the distribution of rods, and the rods selected will not extend above this height.

The rods may comprise a number of materials specified in a number of concentric annuli of increasing radius. An interstitial material is also specified.

An example input for this hole could be:

```
HOLE 1 RANDROD
    CYLINDER 30.0 50.0  !Specify dimensions of the container
    RODS 5000           !Number of rods
    0.5 5.0             !Radius and length
    MATERIALS           !Keyword to introduce materials in annuli
    M1 0.45             !Fuel material and radius
    M2 0.5              !Clad material and radius
    PACK 0.2            !packing fraction
    SEED 9876543        !Seed to reproduce arrangement
    TRIALS 100000000    !Number of trials to achieve packing fraction
    COOLANT             !Keyword to introduce interstitial material
    H2                  !Hole representing interstitial material
HOLE 2 …                !Definition of interstitial material hole
```

This input will model 5000 rods of radius 0.5 cm and length 5 cm. Each rod will comprise material 1 out to a radius of 0.45 cm, with an outer annulus, 0.5 cm thick of material 2. A packing fraction of 0.2 has been specified, and so this will correspond to a height of 34.7 cm. So the 5000 rods will be randomly distributed within a cylindrical volume 30 cm in radius and 34.7 cm high.

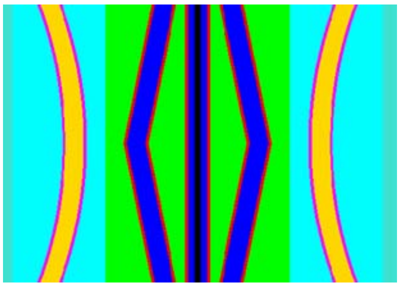
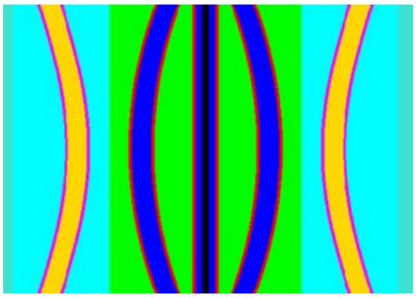
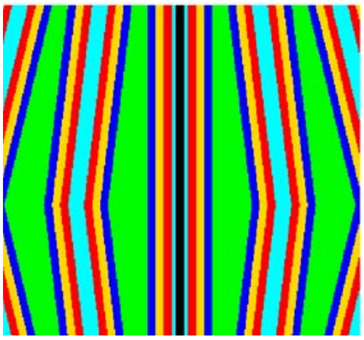
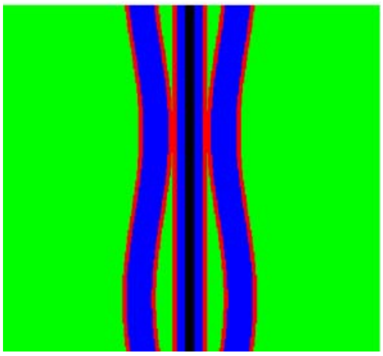
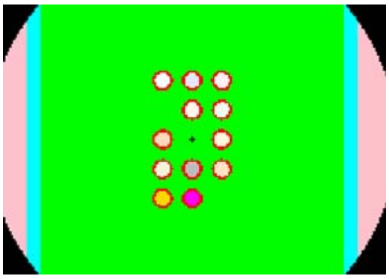
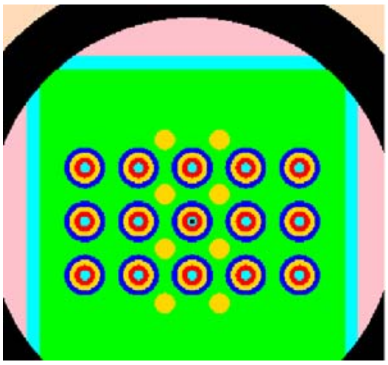
Please contact ANSWERS for more details on using the RANDROD hole.

2.16 BENTPINS HOLE

There is an increasing need to model arrays of bent pins to determine whether there is an arrangement that could exceed some pre-defined criticality limit. Modelling such arrangements currently requires combinations of hole geometries or the FG torus body. The BENTPINS hole allows the user to model such arrays of pins (example images are shown below).

The hole is similar to the SQUARE hole, but includes additional input for specifying the type of curvature, the minimum and maximum pitch values and their corresponding z-coordinates. These parameters fully define the type and extent of the curvature in the hole. Curvature types for the pins currently available are sine, circle and kinked, with each pin capable of having one or more annuli. The option to exclude one or more pins is available by setting the rod material of the corresponding pin to N (instead of M or H).

This new hole will only be accessible from the HOLE GEOMETRY input unit, and thus will allow the used of named holes and named materials.

It is important to ensure that the curvature of the pins is not excessive, simplifications used in the algorithms may affect such systems and the user should ensure that these do not affect their model.

Refer to Chapter 4 of the MONK version 9A User Guide for details on entering the data for the BENTPINS hole.

2.17 TRIANGLE HOLE CHARACTER MAP

The TRIANGLE hole character map option allows the user to display the contents of a TRIANGLE hole as alphanumeric characters on an hexagonal grid. To invoke the option the value after PINS is set negative. Thus if the original input was `PINS 17`, then if it is set to `PINS -17` then a character map will be displayed. Note that only the rod contents relevant to the current hole will be displayed, no details from within any subsidiary holes will be displayed.

As an example consider a TRIANGLE hole with 17 pins along a side. After the normal output for the TRIANGLE hole this option if invoked would add:

```
Character Mapping for this hole only:

                A B B A C D B B C E C E A B E D A
               E C C B B E E C D E E A D E D B C C
              C E D A C E D E B B C A A F B B B C E
             B D A B E B E E D A E F D B A C B D D D
            A D B A B D C D B A B E C E A A B A C D A
           D E E B B D E C B B D C F E D B D B B D B F
          D F A A B F C A F D A D C E C A C E E C A E E
         A B F A C C A B D E B F D B F B B D C D E F D A
        E E F D C A E F B D D A A D B D D E C C E D A B F
       C B B A C D F A D B A E E C F C D C D A C F B B B D
      A B C C D C A D D C D D B A B E A A E B C B A B B C E
     A B A D E C A A B D C E E D E E A C D E E D D A B A D E
    E F D A B C F C E D F D A B C E A A A D E A F C A D D C A
   B A E D E A A D D C F F C D B C D B E D A C D A A F A E A C
  C C E C A C D A A F A E D B B E C B D D F D F D A B E A C F A
 D D B C E D D D E B D E B D B D C F C A D D F E C C E B E F C E
C B B A C E F E D C E B B B C C E A E A A A C F A D C F E B C E A
 A E F B D A E D D E C E B E C C E B A C B D A D D D D A B B D C
  A D E A D A D B C A C B B A B E F C D F A A C C A F A C E B F
   C D C C D E D C C D C C C A B A E D C C E B D D D A B E E A
    F F A B E D E D E E E D A A D F C E C C D E A E E B C F C
     E A B C B C F C A E C D D A B B C D C E F B C B D B B B
      A A A D D C D A C A E E E E E A F A D C C F E C F D A
       C B D E E D B E A A B B A A D B E B C B D C D E A A
        E C A B D C B E D A B A F A B A A C D B F A A D C
         B A C C F F B B E D C E E A F B F A D C A D D A
          D C F B D E E B F A D D A D C A F C A C E C A
           D D E F A D A B B E D A B D E C F E E F B D
            E A C B B B B B E B A B C C E C F C C C F
             E A E E C F B F D E D D E C F E D F D C
              C D D B B D C B A D C A A E A A E E B
               F A B D C A E E A E D E A D B A B E
                E A D B A C E A B E D A C F A D F

 Statistics for this hole:
 Number of lines =  33
 Number per side =  17
 Character mappings for this hole only:
     A       -3
     B       -6
     C        4
     D       -5
     E       -2
     F       -1
```

This option is available in both the HOLE DATA and HOLE GEOMETRY input units, however the HOLE DATA version has a limit of 35 distinct character mappings (A-Z and 1-9) while the HOLE GEOMETRY version has a limit of 61 (A-Z, a-z and 1-9). The above example is from the HOLE DATA version hence the negative values signifying subsidiary holes. For the HOLE GEOMETRY version M and H would precede the numerical value to signify whether it is a material or a subsidiary hole so the above table would appear as:

```
 Character mappings for this hole only:
     A     H    3
     B     H    6
     C     M    4
     D     H    5
     E     H    2
     F     H    1
```

Refer to Chapter 4 of the MONK version 9A User Guide for further details of the Lattice Hole.

2.18 NAMED HOLES AND NAMED MATERIALS IN HOLES

The new HOLE GEOMETRY input unit described previously supports the use of named holes and named materials in holes. The syntax for naming holes it to simply put the hole name after the keyword HOLE. Thus a typical input might look like:

```
HOLE fuel_type1 GLOBE …
HOLE fuel_type2 TRIANGLE…
```

Where in each case the remainder of the input is as usual for the hole type.

The use of named materials and named subsidiary holes is simply to replace the material number by the name, or the subsidiary hole by its name – noting that in the HOLE GEOMETRY input unit materials are preceded by M and holes by H. Thus and input of the form:

`M1  M  2  H2  H  4` might become:
`M  fuel1  M  water  H  fuel_type1  H  fuel_type2`

Noting that the spaces separating the M and H from their corresponding names are mandatory (material and hole numbers do not require a separating space).

It is recommended that the HOLE GEOMETRY input unit follows the MATERIAL SPECIFICATION input unit if named materials are used.

2.19 STDV FIRST STAGE CHECK

Some calculations complete within the ten stages required for the Sampling Guidance to analyse the case for possible settling problems. This option has been provided to allow the user to specify the number of scoring stages that must be completed before a target STDV is checked. The keyword STDVSTAGE is entered in the CONTROL DATA input unit, followed by the stage number at the end of which the STDV is first checked. Thus the input might be of the form:

```
BEGIN CONTROL DATA
       SEEDS 123 6789
       STAGES -10 100 1000 STDV 0.0030
       STDVSTAGE 10
END
```

In this example the STDV will not be checked until the end of stage 10.

Note that stage numbers other than 10 can be used to specify when the checking begins – the above example shows one reason for its use.

2.20 RUNID

At the end of a MONK run the code will write a unique identifier to the normal output file for each run unless the RUNID keyword is specified in the CONTROL DATA input unit. The default identifier is based on the date and time of the run. The RUNID keyword allows the user to specify their own identifier for the run. An example input is:

```
BEGIN CONTROL DATA
RUNID Run:case1
STAGES -10 100 1000 STDV 0.0020
END
```

Here RUNID to gives the case the user defined name "`Run:case1`".

2.21 JOB LOG

The Job Log option allows the user to force the code to write a log of the current calculation to a separate file, on a channel called JOBLOG, specified in the datsets file. The request for a Job Log is entered via the CONTROL DATA input unit, and includes the sub-option keyword DESCRIPTION to enable the user to specify a header to be written to the Job Log file.

An example input using these keywords would look like:

```
BEGIN CONTROL DATA
JOB LOG
DESCRIPTION
  Simple UO2 pincell in Water
  Purpose to find optimum pitch
  at different temperatures.
  BINGO data \
STAGES -10 100 1000 STDV 0.0020
END
```

2.22 EXPECTED VALUE OPTION

The Expected Value option allows the user to specify one or more sets of results along with a short string identifying them. On completion of the calculation the these results are compared with the K(THREE) value and a table of C/E and (C-E)/E printed. The keyphrase for the option is EXPECTED VALUE, which is followed by a text string (a short description), the expected k-effective and its standard deviation. An example input would be of the form:

```
BEGIN CONTROL DATA
 STAGES -20 100 1000 STDV 0.0014
 EXPECTED VALUE EXPERIMENT 0.7802 0.0062
 EXPECTED VALUE MONK8B-UKNDL 0.7849 0.0010
 EXPECTED VALUE MONK8B-JEF2.2 0.7788 0.0010
 EXPECTED VALUE MONK8B-ENDF/B-VI.3 0.7791 0.0010
 EXPECTED VALUE MONK8B-JENDL3.2 0.7790 0.0010
END
```

Here the k-effective from an experiment and expected from each of the DICE nuclear data libraries supported by MONK is entered. The output for this input would be:

```
qv:     COMPARISON WITH EXPECTED K(THREE) VALUES ENTERED
qv      ================================================
qv
qv      Expected  Expected  Present   Present  Difference   C/E      (C-E)/E    Library
qv      K(THREE)    STDV    K(THREE)   STDV     SD units             SD units
qv       0.7802    0.0062    0.7785   0.0014     -0.26     0.9979     -0.34    EXPERIMENT
qv       0.7849    0.0010    0.7785   0.0014     -3.72     0.9919     -4.75    MONK8B-UKNDL
qv       0.7788    0.0010    0.7785   0.0014     -0.16     0.9997     -0.20    MONK8B-JEF2.2
qv       0.7791    0.0010    0.7785   0.0014     -0.33     0.9993     -0.43    MONK8B-ENDF/B-VI.3
qv       0.7790    0.0010    0.7785   0.0014     -0.28     0.9994     -0.35    MONK8B-JENDL3.2
```

Note that each line of this table is preceded by ‘qv’, with the first line preceded by ‘qv:’, to simplify finding the table with search tools.

The text string can be up to 50 characters long, and up to 100 ‘expected values’ can be entered.

2.23 SHOWK

This option is selected in the CONTROL DATA input unit and copies the stage k-effective to the screen (FORTRAN channel zero). An example of its use would be:

```
BEGIN CONTROL DATA
       STAGES -20 100 1000 STDV 0.0020
       SHOWK
END
```

During a run this option causes the code to write to the screen the K(THREE) values evaluated at the end of each stage: for unix or linux the screen would be the command or shell window used to launch the job; for a WINDOWS PC it would be the DOS window used to launch the job; for LaunchPad it will be the LaunchPad window containing the details of the run.

2.24 INFLOWS TALLY

A new Action Tally has been added better to log the transfer of samples between FG regions. The existing boundary crossing tally gives little information on the actual flow across boundaries. The new option presents a table listing the number of samples in each energy group travelling form one region into another and between FG parts.

The option is activated using the keyword INFLOWS in the ACTION TALLY input unit. The output for a case comprising 3 groups, three regions and two parts is of the form:

```
1INFLOWS FOR STAGES 1 TO   1 NORMALISED TO 10000 SAMPLES
 =======================================================

    REGION TO REGION INFLOWS
    ------------------------


    GROUP      1    ( 100.0000 KEV TO  15.0000 MEV)
    ----------------------------------------------
    FROM:      1      2      3
     TO:
      1        0      0    754
      2        0      0      0
      3     8744      0      0
    Leak       0      0   7990

    GROUP      2    (   0.4000  EV TO 100.0000 KEV)
    ----------------------------------------------
    FROM:      1      2      3
     TO:
      1        0      0    151
      2        0      0      0
      3      452      0      0
    Leak       0      0    302
```

```
GROUP      3   (   0.0000  EV TO   0.4000  EV)
----------------------------------------------
FROM:      1      2      3
 TO:
  1        0      0      0
  2        0      0      0
  3        0      0      0
Leak       0      0      0

SUMMED OVER ALL ENERGIES
------------------------
FROM:      1      2      3
 TO:
  1        0      0    905
  2        0      0      0
  3     9196      0      0
Leak       0      0   8291

PART TO PART INFLOWS
--------------------


GROUP      1   ( 100.0000 KEV TO  15.0000 MEV)
----------------------------------------------
FROM:      1      2      3
 TO:
  1        0    754      0
  2     8744      0      0
  3        0      0      0
Leak       0   7990      0

GROUP      2   (   0.4000  EV TO 100.0000 KEV)
----------------------------------------------
FROM:      1      2      3
 TO:
  1        0    151      0
  2      452      0      0
  3        0      0      0
Leak       0    302      0

GROUP      3   (   0.0000  EV TO   0.4000  EV)
----------------------------------------------
FROM:      1      2      3
 TO:
  1        0      0      0
  2        0      0      0
  3        0      0      0
Leak       0      0      0

SUMMED OVER ALL ENERGIES
------------------------
FROM:      1      2      3
 TO:
  1        0    905      0
  2     9196      0      0
  3        0      0      0
Leak       0   8291      0
```

Note that the part-to-part table shows an extra (third) part – this has no effect on the results and can be ignored - it will be corrected in the next release of the code.

2.25 AUTO SETTLING

A new AUTO SETTLING capability has been added to MONK. This option allows the user to specify the number of superhistories per stage and a settling and final standard deviation (STDVs). The code then monitors k-effective during the run and if the settling STDV is exceeded it clears the tallies (as done at the end of settling for a normal run).

The input is of the form:

```
AUTO SETTLING  0.0025 0.0025
STAGES  0  0  5000
```

The current recommendation is to use the final STDV value for the settling STDV.

An output table prints the number of times the tallies have been cleared because k-effective has become either too high or too low (referred to as 'ups and downs'). A large number of 'ups and downs' suggests that insufficient superhistories per stage were specified.

The AUTO SETTLING option was intended for use with cases that might demonstrate slow convergence due possibly to a large number of reactive components of slightly differing reactivity. An example could be an array of fuel elements in a storage pond, where the interactivity is slight and the elements experience different boundary conditions such as water or concrete walls.

It is still necessary for the user to fully review the output from an AUTO SETTLING calculation to convince themselves that the case has run correctly. This option can improve the behaviour of the code when a normal run would under-settle, however if this is a novel model then it important that more than one calculation be run to build confidence in the modelling.

2.26 AUTO ZONEMAT

If the AUTO SETTLING option is used and no source geometry input unit is specified, then the code will look through the user specified material compositions to identify the fissile materials. MONK will then look for these materials in the geometry and generate a ZONEMAT specification.

Note that it is likely that this will not be the 'best' source description available, and that a user should be able to specify a better ZONEMAT description and thus reduce the settling time.

2.27 SYNTAX IMPROVEMENTS

There are a range of syntax improvements, including an enhanced sequence of numbers and an enhanced group-repeat option. These can all be found in Chapter 2 Input Description, Section 2.2 of the MONK version 9A User Guide.

The group-repeat enhancement is an extension to the group repeat option that allows a different value to be used in each pass of the repetition. The syntax for this is:

```
N*{A B (C1 C2 C3 . . CN) D}
```

`N` = number of repetitions
`A B D` = values used in each repetition
`C1` = value used in first pass
`C2` = value used in second pass
...
`CN` = value used in last pass

As an example of using this option consider a sequence of concentric ZROD bodies with increasing radius and different materials. The origin and height are common

`5*{ZROD M (1 2 1 2 1) 0 0 0 (2.1 4.2 6.3 8.4 10.5) 5.3}`

```
=
ZROD M 1 0 0 0  2.1 5.3
ZROD M 2 0 0 0  4.2 5.3
ZROD M 1 0 0 0  6.3 5.3
ZROD M 2 0 0 0  8.4 5.3
ZROD M 1 0 0 0 10.5 5.3
```

The number of items between each pair parentheses must match the value of the group repetition count. Parameters or formulae may be used for a given item but the colon sequence may not.

A semicolon may be used as a null value if a particular pass does not require an item.

2.28 NEW FORMULA FUNCTIONS AND OPTIONS

The triplet of inverse trigonometric functions have been added to the input processor of MONK, these are: ASIN, ACOS and ATAN.

The input processor has also bee modified such that:

- Any input using double precision numbers uses double precision arithmetic in the formula
- Thirty levels of parentheses are now available
- A generalised power function (^) of the form (a+-*/b)^(c+-*/d) that can be used with numbers and parameters, along with the existing ^2, ^3, SQRT and CBRT functions.

2.29 BINGO

The new BINGO continuous-energy nuclear data and collision processor capability has been included in MONK version 9A for evaluation purposes. This library uses JEF2.2 nuclear data, but with improved nuclide-dependent energy structure better to represent the behaviour.

BINGO also includes temperature data to enable calculations at elevated temperatures to be undertaken. In this version of BINGO there are distinct sets of temperatures for each nuclide and the code will select the nearest temperature available: thus it will not interpolate temperatures.

Also note that some MONK tallies are not available with BINGO. The most relevant of these include: ICSBEP, sensitivity, AUTO ZONEMAT and the Tabular Output.

2.30 CFREE AND CBOUND

The BINGO nuclear data library now includes bound carbon for correctly modelling graphite systems. To ensure that the user does not accidentally use the free carbon model in graphite with BINGO, the nuclide identifier C has been removed. Thus in BINGO it is necessary to specify either CFREE or CBOUND explicitly for free and bound (in graphite) carbon respectively.

With DICE nuclear data libraries the nuclide identifier C remains available and its use is unchanged.

2.31 DATAGRAM

The original WIMS-E based libraries used in MONK version 8B and earlier for depletion calculations have been replaced with the new Datagram format nuclear data library. The library used is the same as that found in WIMS version 9A except there is a single U235 fission spectrum used throughout the library. This form of the library is called the ‘CHI’ library, and should be the only WIMS library used with MONK version 9A. Use of the non-CHI library is likely to result in the code selecting inappropriate data for the fission spectrum.

The library is still accessed via the WIMS channel in datsets, or through the pulldown menus in LaunchPad.

2.32 BURNUP PREDICTOR-CORRECTOR

An simple predictor-corrector option has been coded into MONK version 9A to improve the behaviour of burnup calculations with long time steps. This version is available for evaluation purposes.

The option is activated using the keyword PREDCOR in the BURNUP input unit.

2.33 NGOTO

This new option for burnup cycling calculations has been added to allow the user to tell the code not to jump to the specified label for the specified cycles. This is to remove the need for GOTO 2 2-100 statements (GOTO label 2 for cycles 2 to 100 inclusive). Such a statement can be replaced with NGOTO 2 1. This statement informs the code NOT to go to label 2 for cycle 1. The earlier version has caused problems when more cycles were specified and the user failed to update all the relevant GOTO statements.

2.34 FLUX SCALING

The flux scaling used for the WIMS route has been changed to be the same as that used for the DICE/BINGO route. The flux is now either scaled to 10000 samples tracked (the default normalisation) or output without scaling (if NONORM specified). The original WIMS route always scaled to 100 samples tracked and thus made intercomparison with other methods difficult.

2.35 REVISED COMPONENT MULTIPLICATION

The COMPONENT MULTIPLICATION OPTION has undergone a minor revision to support the use of named parts. The change comprises the need to terminate the list of parts with a /. Thus a MONK8B input that looks like:

    COMPONENT MULTIPLICATION FOR PART 2 4 7

Must now be entered in MONK9A as

    COMPONENT MULTIPLICATION FOR PART 2 4 7 /

The change is to support named parts in situations such as

    COMPONENT MULTIPLICATION FOR PART fuel_A rod_B rod_C /

Where fuel_A, rod_B and rod_C are named parts.

3 Summary

This report has introduced the main new features of MONK version 9A that provide a range of additional options to assist the criticality analyst and assessor:

- FG - Window Part
- FG - OVERLAP Part

- FG - Zone Complement
- FG - OR Operator
- FG - CONTAINED and CONFINED
- FG - Rotation Options
- FG - Relative Body Origins
- Unified Source
- ZONEMAT – Named Parts and Materials
- Updated Material Specification Module
- New Datsets (Paths)
- HOLE GEOMETRY Input Unit
- USER Hole
- COIL Hole
- RANDOM RODS Hole
- BENT PINS Hole
- Triangle Hole Character Map
- Named Holes and Named Materials in Holes
- STDV First Stage Check
- RUNID
- JOBLOG
- Expected Value Option
- SHOWK
- INFLOWS Tally
- AUTO SETTLING
- AUTO ZONEMAT
- Syntax Improvements
- New Formula Functions and Options
- BINGO
- CFREE and CBOUND
- Datagram
- Burnup Predictor-Corrector
- NGOTO
- Flux Scaling
- Revised COMPONENT MULTIPLICATION

A major programme of pre-release testing has been completed which has not only exercised the new features of the code but also exercised a large set of older models to ensure that existing features have not been disturbed. This test programme has also included contributions from the user community through external beta code versions. Further details on the new features are contained within the user guide issued to accompany the release of MONK version 9A. Additional advice and information is available via the usual ANSWERS customer support channels.

4 References

[1] The ANSWERS Software Service
MONK User Guide for Version 9A
ANSWERS/MONK/REPORT/005