

ANSWERS/MONK(98)7

An Introduction to the New Features of MONK8A

February 1998

Title	An Introduction to the New Features of MONK8A
Customer	For Criticality Customers of the ANSWERS Software Service
Customer reference	
Confidentiality, copyright and reproduction	Copyright AEA Technology plc 1998 Distributed only on the basis of strict confidentiality. Confidentiality to be maintained. Enquiries about copyright and reproduction should be addressed to the Commercial Manager, AEA Technology plc.
File reference	
Reference number	ANSWERS/MONK(98)7

The information which this report contains is accurate to the best knowledge and belief of AEA Technology, but neither AEA Technology nor any person acting on behalf of AEA Technology make any warranty or representation expresses or implies with respect to the accuracy, completeness or usefulness of this information, nor assume any liabilities with respect to the use of, or with respect to any damages which may result from the use of any information, apparatus, method or process disclosed in this report.

AEA Technology plc
The ANSWERS Software Service Manager
Winfrith Technology Centre
Dorchester
Dorset
DT2 8DH
Telephone 01305 202352
Facsimile 01305 202746

AEA Technology is the trading name of
AEA Technology plc

AEA Technology is certified to ISO9001

Author	Name	M J Armishaw
	Checked by	Name
Approved by	Signature	
	Date	
	Name	I J Curl
	Signature	
	Date	

Executive Summary

Version 8A is the latest release of the MONK Monte Carlo code. This new version incorporates a range of additional features, including the capability to perform both criticality and reactor physics calculations. This report provides an introduction to the use of each of the additional features of the new code and is directed at existing MONK7B code users. Further information on all the features introduced here can be found in the MONK8 User Guide.

Contents

1 INTRODUCTION	1
2 GUIDE TO NEW FEATURES	2
2.1 REACTOR PHYSICS CAPABILITY	2
2.1.1 WIMS Multigroup Data	2
2.1.2 Weighted Tracking	2
2.1.3 Microscopic Burnup	3
2.1.4 WIMS interface	4
2.1.5 EDIT Options	4
2.1.6 Adjoint Tracking	4
2.1.7 Point Estimator	5
2.1.8 Fixed Source Tracking	5
2.2 HEMISPHERE AND HEXAGONAL PRISM BODIES	5
2.3 ALBEDO MATERIALS	8
2.4 SIMPLE ROTATION	8
2.5 K-EFFECTIVE HALT	8
2.6 QUICK DICE	9
2.7 SPLITTING AND ROULETTE	9
2.8 EMBEDDED FILES	10
2.9 DATSETS AND INTFACE FILES	12
3 SUMMARY	14
4 REFERENCES	15

Appendices

APPENDIX 1	REACTOR PHYSICS EXAMPLES
APPENDIX 2	HEMISPHERE AND HEXAGONAL PRISM EXAMPLES
APPENDIX 3	ALBEDO MATERIAL EXAMPLE
APPENDIX 4	SIMPLE ROTATION EXAMPLE
APPENDIX 5	K-EFFECTIVE HALT EXAMPLES
APPENDIX 6	SPLITTING AND ROULETTE EXAMPLE
APPENDIX 7	EMBEDDED FILE EXAMPLE
APPENDIX 8	CONVERSION OF MONK5W INPUT TO MONK8A

1 Introduction

The development and subsequent release of MONK7A was the culmination of a major renewal programme for the MONK criticality codes. This comprised a comprehensive software re-engineering programme to bring MONK into line with the successful modular system employed for the shielding codes MCBEND and RANKERN. This has provided customer and custodian benefits in terms of shared facilities and developments and more cost-effective and efficient maintenance. Version 7B was later released and included a range of extra features to extend the capabilities of the code [1].

Version 8A is now available from the ANSWERS Software Service and is the first version to provide the capability to perform both criticality and reactor physics calculations. In addition several other features have been developed and are included in version 8A. This report provides an introduction to their use and is directed at existing MONK7B code users. Further information for all the features introduced here can be found in the MONK8 User Guide [2].

The main new features of MONK8A are

- *Reactor physics capability* - the ability to perform reactor physics calculations, including depletion cycles
- *New geometry bodies* - the HEMISPHERE and the HEXAGONAL prism
- *Albedo materials* - the capability to approximate the treatment of the albedo process in a range of materials without having to track through the material
- *Simple rotation option* - this provides a user friendly input for rotating bodies about the X, Y or Z axis
- *K-effective halt* - the option to halt the calculation if k-effective falls outside user-defined limits
- *Removal of unnecessary material input* - removal of the need to enter the number of materials and the number of nuclides prior to entering the material composition data
- *Quick DICE* - an improved package for reading and processing the DICE nuclear data
- *Splitting and Roulette* - a mechanism for encouraging samples to concentrate in specified areas within the model.
- *Embedded Files* - an option which allows sections of the input data to be stored in files that are embedded within the main input file.

For those users who are unfamiliar with MONK7 there are details in Appendix 8 describing the conversion of input data from the MONK5W format to that of MONK8.

The following section describes each new feature and where appropriate provides an example of its use.

2 Guide to New Features

2.1 REACTOR PHYSICS CAPABILITY

The most significant development in MONK8A is the inclusion of a reactor physics capability derived from the standalone MONK5W code. The options included into MONK8A for the reactor physics capability include:

- WIMS multigroup and subgroup data
- weighted tracking
- microscopic burnup
- WIMS interface
- EDIT options
- Adjoint tracking
- Point Estimator
- Fixed source tracking

A brief description of each of the above options is given below with examples listed in Appendix 1.

2.1.1 WIMS Multigroup Data

MONK8A now includes the capability to process and use WIMS group dependent nuclear data processed from the WIMS library. This data can be used as both conventional broad group pre-shielded cross sections in a defined group structure or as sub-group data prepared by the NOVICE method [3]. In both cases the data can be imported into MONK8A using a WIMS7 interface which can now contain both conventional and sub-group data. This data is prepared by running WIMS to generate the interface files, while the sub-group data can also be prepared using the NOVICE routines, which have been coded into MONK8A, to read the WIMS library directly. These routines produce a set of interface files which, like the option of reading WIMS interfaces, are then read by the tracking process.

2.1.2 Weighted Tracking

The MONK5W absorption collision survival algorithm has been incorporated into MONK8A. In this algorithm, the neutron weight is modified to account for the probability of absorption rather than treating the collisions explicitly as in the process used in MONK7 calculations. This modified algorithm can be shown to be a generalisation of the analogue tracking treatment used in MONK7. The algorithm uses 3 parameters which correspond to the average weight, the minimum weight and the maximum weight of a particle at a collision. If the minimum weight is larger than the actual weight after a collision then the code uses a roulette procedure to see if the particle continues. If the particle continues it has the average weight. On the other hand, if the weight after a collision exceeds the maximum weight then the particles splits.

The analogue treatment would use unity for the 3 parameters controlling the fate of a particle at a collision. Thus this new algorithm gives the user extra flexibility which has been found to improve convergence in some calculations. Particular examples are the cases where other

parameters, such as power distribution is required, in addition to the standard estimate of reactivity required for criticality calculations.

Further weighted tracking is achieved by using splitting and roulette by region and energy originally developed for use in shielding calculations. The splitting and roulette option is described in Appendix 6.

2.1.3 Microscopic Burnup

The MONK8A code includes a microscopic burn-up capability which will enable the code to treat whole core burn-up in any reactor type and determine the effects of radiation on the fuel material. The burnup process requires special treatment because the material compositions vary with time and are dependent on their environment. In a real reactor core the material compositions would be changing continuously with time during burnup. To treat this process in MONK8 an extension of the methods used in deterministic calculations of burnup are used. Monte Carlo calculations are used to estimate the flux at a given time point and this flux is used to estimate the reaction rates of the materials in the problem. Thus the nature of the Monte Carlo component of the code, which assumes steady state conditions and then requires time to converge and calculate both the value of k-effective and the various reaction rates, is the same as for non-burnup calculations. These reaction rates are then used to calculate the new material compositions by solving the depletion equations. Therefore the method adopted in MONK8A simulates the continuous change in material compositions by breaking down the process into discrete steps comprising a tracking phase followed by a burnup phase, then repeating these two phases until the required time period has been spanned. In a burnup calculation each cycle corresponds to a real burnup time which is specified in the input data for the burnup calculation.

The tracking phase uses fixed material compositions and calculates the fission and absorption reaction rates for the steady state system. These results are written to interface files for use by the BURNUP module built into MONK8A. The tracking phase in a burnup calculation is similar to a normal MONK calculation but is able to handle the extended number of materials required to model the changing material compositions in each cycle due to depletion.

The burnup phase reads the interface files produced above and, for each material in each region uses the supplied reaction rates and the actual burnup time to calculate the new nuclide number densities. These new number densities are written to a set of interface files to be used in the tracking phase of the next cycle. Each cycle corresponds to a time step in the burnup process and during this time step constant reaction rates are assumed. The solution of the depletion equations is most accurate with short time steps and so the user has the option to sub-divide the time step for greater accuracy. However it is normally the practise to use short time steps between Monte Carlo flux calculations so that only one time step is required in the depletion calculation.

A typical burnup case requires several cycles to complete with the differing requirements of each cycle (e.g. burnup step size) controlled by special cycle control keywords such as QUAL, MISS, NCYCLE and FINISH. The NCYCLE keyword is used at the beginning of the input file and is used to set the first and last cycle number. The QUAL/MISS keywords are used within the body of the input file to determine which lines are selected/ignored for each cycle. The FINISH keyword defines the end of the current NCYCLE block. Appendix 1 gives examples of how to use these keywords.

2.1.4 WIMS Interface

At the end of a reactor physics calculation in MONK8A, the default option is for the code to write the flux or reaction rates to a WIMS format interface. This enables modules within WIMS to read and process the data.

2.1.5 EDIT Options

The following edit options have been included in MONK8A:

- Collision density flux estimator
- Reaction edit by region, group, material, hole
- Migration area
- Buckling
- Collision probabilities
- Flux tilts and spatial components
- Harmonic components of flux
- Reaction rates for burnup

The collision density flux estimator (and the associated reaction edits by region, group, material and hole) are an alternative to the standard track length estimator used in MONK7.

The migration area and the buckling edits are useful in reactor physics calculations where estimates of both leakage and its effect on reactivity are required. These edits mimic the edits available from deterministic calculations in codes such as WIMS.

Collision Probabilities are now available from MONK. These values can be compared with deterministic estimates with the added advantage that complex geometries can be used in MONK. Thus this option gives MONK an extra validation function that will be useful in a reactor physics context.

Flux tilts and the associated tilts in power are quantities used by assessors to estimate peak powers and hence peak temperatures in a reactor core. With this edit available it is possible to use a Monte Carlo calculation to validate calculations which previously could only be carried out in a deterministic calculation

Higher components of the flux such as the current can now be obtained from MONK using the spherical harmonics components of the flux. Initially these values were used to study leakage across regions of a lattice cell, but there are now increasing uses of the higher components of the flux in reactor physics. These include their use to calculate discontinuity factors for nodal theory calculations and also to calculate corrected reflector data. Further, these data can be used in the perturbation theory estimate of the leakage component of reactivity changes. In the MONK5W code there was a post processor called PERTH that utilised these components of the flux to estimate small reactivity changes using perturbation theory.

2.1.6 Adjoint Tracking

The calculation of the adjoint flux is required both for perturbation calculations and for point estimation of flux or reaction rates. The MONK8A method uses the same multigroup tracking techniques as the forward MONK solution but with the scattering and yield data transposed. In addition a bias function may be input to improve the convergence of the calculation by adjusting the basic cross-section data. The bias function for an ADJOINT calculation is normally the normal flux solution.

2.1.7 Point Estimator

The calculation of flux or reaction rates at a point (or in a small volume) using the Monte Carlo technique can be both difficult and time consuming due to the low probability of tracks passing through the specified point or volume. The Point Estimator method was developed to solve this problem by making use of a kernel to calculate the required score.

The kernel is calculated at the start of the run by using adjoint tracking with the target point as the source and running a user specified number of samples. During the adjoint tracking the fissions of the samples in the surrounding regions are tallied and it is these values which comprise the kernel. The kernel is thus the probability that a fission in a given region in a problem will cause a subsequent collision or reaction at the point of interest.

When the kernel has been calculated for all the target points a forward calculation will be performed in which the fission rates are scored in the same regions. At the end of the calculation the scored values by region are folded into the kernel to calculate the required score at the target point(s). This method gives results with significantly greater accuracy than is possible with the standard scoring method in a similar run time.

2.1.8 Fixed Source Tracking

This option will calculate the multiplication and flux due to a fixed source in a problem with fissile materials. Multiple sources comprising any combination of points, lines, laminae and volumes can be used, each with a user specified relative probability of being sampled. In addition each of the sources can have either a user defined spectrum or the default fission spectrum. At the start of each stage the source is obtained from the source distribution and energy spectrum supplied by the user.

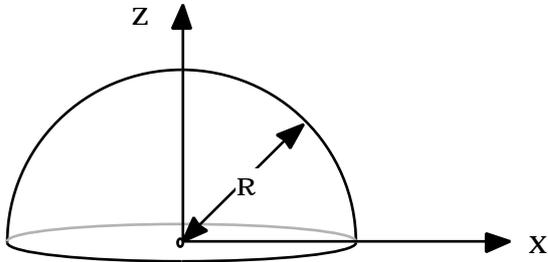
The superhistory option can be used with fixed sources for those problems which are close to critical. In such cases, after the specified number of generations have been completed, a truncation procedure is used to estimate the contribution from those generations which would otherwise have been tracked.

2.2 HEMISPHERE AND HEXAGONAL PRISM BODIES

Two new bodies are available for use in MONK8A: the hemisphere and the hexagonal prism. The bodies support the full range of FG facilities for part construction and geometry checking. They may be used as the global part container with albedo boundary conditions. In addition a new part type - A Hexagonal array has been included to provide a means of tessellating the Hexagon body.

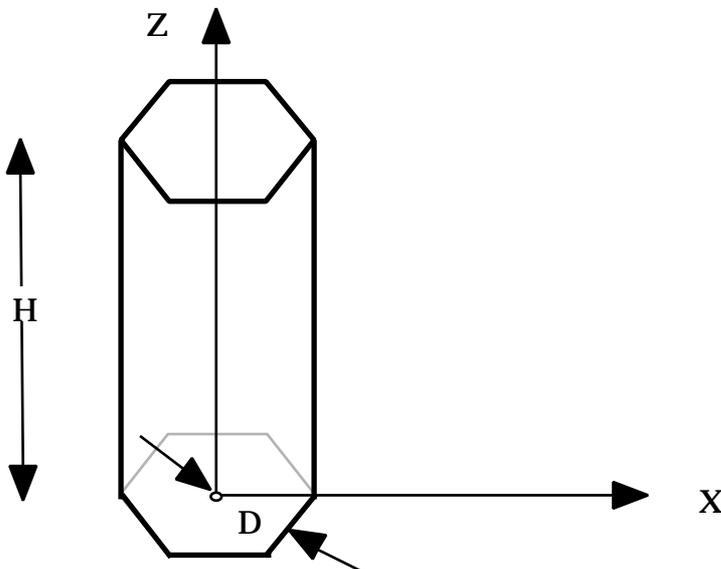
Hemisphere

The origin located at the centre of the plane base. Additional input parameter is the radius (R). The shape exists in three main variants: XHEMI, YHEMI and ZHEMI with '+' and '-' options to indicate the direction in which the hemisphere lies with reference to its plane surface (above or below). The example below is +ZHEMI, specified as **+ZHEMI Body origin R**.



Hexagonal Prism

The origin located at the centre of the base. Additional input parameters are the half-width across the hexagonal cross-section (D) and prism height (H). The hexagonal prism exists in three variants: XHEX, YHEX, ZHEX. The example shown below is a ZHEX, specified as **ZHEX Body origin D H**.



Hexagonal Prism Arrays

An array of hexagonal prisms does not form a regular FG body and so it cannot be dealt with in exactly the same way as boxes in the existing array part. Therefore, the part formed by an array of hexagons will have a hexagonal outer boundary and an interstitial material to fill in the gaps. Each hexagonal prism in the array will contain a subsidiary hexagonal prism part (similar to the use of the existing array part) - of course, any number of hexagonal prisms in the array may contain the same subsidiary part. The interstitial material can be either a real material or a hole

material (specified using Hn of BHn where n is the hole serial number). The hexagonal array formed can then be included in a parent hexagonal body in another part and hence located in the model in its correct position - this is consistent with the use of the existing array part. The hexagonal array will exist in X, Y and Z variants with the subsidiary hexagonal prism parts being XHEX, YHEX and ZHEX respectively. Note the automatic rotation performed on the hexagonal prisms making up the array shown in the picture below.

To obtain the full array shown below, the specification would be:

ZHEXARRAY

M(int) D H

N

3N(N-1)+1 part numbers

Keyword

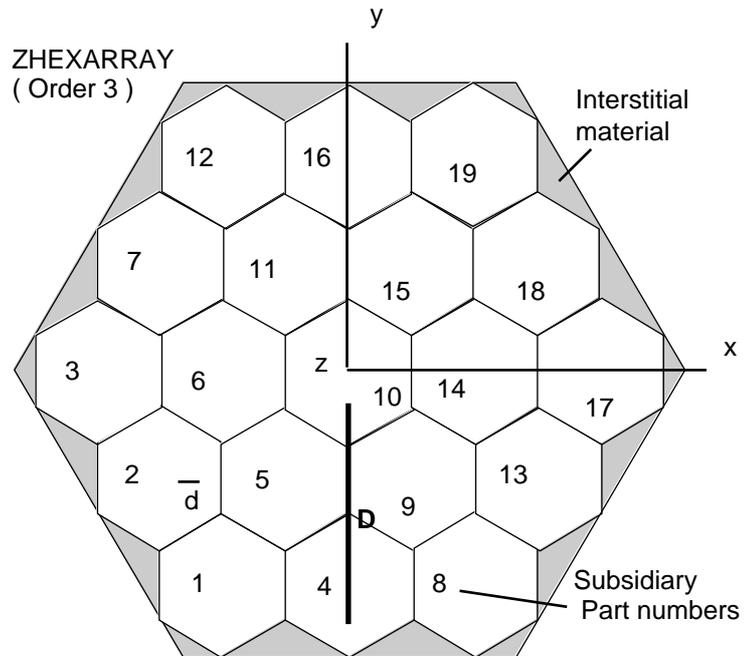
Interstitial material, followed by the size of the enclosing hexagonal array

Number of hexagons on outer side of the array (3 in this case)

To define the contents (part number) of each hexagonal prism in the order shown below

The half width across the hexagonal cross section of a subsidiary hexagon (d) is related to the half width of the hexagonal array (D) by the following equation:

$$D = d(3N-1)/(2 \cos 30)$$



2.3 ALBEDO MATERIALS

The detailed treatment of albedo processes requires extensive collision processing and tracking within the materials comprising the geometry beyond a specified boundary. The consequent processing of these events adversely affects the efficiency of the code and so an alternative approximate treatment has been incorporated. Note that due to the approximate nature of the albedo treatment it should be used with care and is probably best suited to survey calculations where trends rather than absolute results are of interest.

The new albedo materials option provides a mechanism which models the process in which a particle crosses a material boundary and re-emerges with changed energy and direction by replacing the tracking process with a set of probabilities, angular distributions and simplifying assumptions. In addition, instead of explicitly entering the data in the input, a library has been created which can be read by MONK. This library contains a range of materials with the data for each of the materials stored in 16 energy groups. The list of materials available in the library currently comprises:

WATER
CONCRETE
JABROC

The above library will be extended to include additional materials in the future. If the required material is not available in the library but the required albedo data are available then the albedo materials input unit can be used to supply these data and make use of them during tracking.

Note that MONK7 includes a very simple albedo approximation where the albedo treatment is summarised by a single probability. The new process in MONK8 is clearly a more sophisticated treatment and in general will produce more accurate results than the MONK7 treatment. However, care should be used when employing either treatment in calculations.

Examples demonstrating the use of the Albedo Data input unit are listed in Appendix 3. These examples include details of using the library and entering the data explicitly.

2.4 SIMPLE ROTATION

A facility for the rotation of bodies by altering the orientation of their defining vectors is currently available within FG. The simple rotation option augments this with a means of specifying rotations about the X, Y, or Z axis. The rotation is specified after the normal body data by giving a keyword followed by an angle. A positive angle means a clockwise rotation when looking down the axis towards the origin. An example input is given in Appendix 4

2.5 K-EFFECTIVE HALT

This new option in the CONTROL DATA unit allows the user to halt the MONK case if the value of k-effective is outside a pre-selected range. The option is controlled by three keywords with which the user can set one or both of the lower and upper limits (KLOWER, KUPPER)

of the acceptable range of k-effective, and the stage (KSTAGE) at which the checking begins (but not during the settling stages). The default first stage for this checking is stage 3.

The choice of lower and upper limit needs to take into account the possible movement in the value of k-effective due to the Monte Carlo process. The choice of stage at which the checking is started will depend on the number of settling stages and number of generations in each superhistory.

A suggested use for k-effective halt is within looped scoping calculations where the looping facility enables many separate cases to be analysed in a single run. The option to halt as soon as possible those cases which are giving values of k-effective outside the acceptable range can be useful to ensure that computing time is concentrated on the more appropriate cases.

Examples demonstrating the use of this option are listed in Appendix 5.

2.6 QUICK DICE

The routines which read and process the DICE nuclear data have been rewritten to handle the data more efficiently with the effect of reducing the run time required to process the data. This reduction in run time is case dependent, but has been shown to be as much as a factor of 100 in one case where a large number of JEF nuclides were used.

2.7 SPLITTING AND ROULETTE

The splitting and roulette option gives the user the capability to encourage the samples to concentrate on selected sections of the model by overlaying an importance map which gives the relative importance of sections of the model. This is particularly useful if the region of interest is in such a position that few samples would reach it. By overlaying a suitable importance map more samples can be encouraged to reach the required region thus giving a more accurate score. At present this option is only available for reactor physics calculations.

The splitting and roulette method requires two sets of input: the first specifies the geometrical mesh and energy group structure in which the importance map will be used and must therefore cover the extent of the model in space and energy; the second comprises the importance map itself containing the importance of each mesh element in space and energy. The ratio between the importance in adjacent meshes is used during the tracking process either to increase the number of samples but correspondingly reducing the weight of each (splitting when the weight increases), or to use a random number to determine if the sample should be removed (roulette when the weight decreases).

An example of a case which uses splitting and roulette is given in Appendix 6.

2.8 EMBEDDED FILES

This option allows sections of the input data to be stored in files that are embedded within the main input file. The basic form is illustrated by the following segment of data:

```
....  
BEGIN MATERIAL GEOMETRY  
< geomdata  
END  
....
```

The character '<' signifies a request to take the input from an embedded file. The character string ('geomdata' in the example) is a named channel to which the embedded file is assigned in the 'datsets' file used by MCANO programs.

Ideally, the reference to an embedded file would be directly to its name in the user's file system; unfortunately, the syntax of file names varies with machine and operating system. The chosen method of referring to a channel name maintains the portability between systems. For the above example on a UNIX platform, the datsets file would contain a line of the form:

```
geomdata /disc/dir/myfile
```

The following rules apply:

- The channel name may be chosen freely by the user provided that the standard names used in MCANO programs are avoided – do not use: input, output, dice, impmap etc. (see the MONK User Guide [2])

- Channel names are not case sensitive.

- Current system limitations restrict the channel name to nine characters.

- No data may be included on the line after an embedding instruction but an optional, in-line comment (beginning with !) is allowed.

- In this basic mode, all lines of the embedded file are read before returning to the main file to continue reading at the line following the embedding instruction.

- An embedded file may include pointers to further embedded files to a depth of nine levels below the basic input file.

An extension of this basic mode allows a section of an embedded file to be read. To force a return to the containing file, the embedded file may contain the line:

```
< UP
```

The keyword UP is not case sensitive. To force a return to the main input file from any depth of embedding, the embedded file may contain the line:

```
< INPUT
```

The keyword INPUT is not case sensitive.

To enter an embedded file at a specific point, labels may be added. The label in the embedded file is placed on a line beginning < UP and may be any character string (no embedded blanks) chosen by the user, e.g.

< UP Entry1

(The logic here is to make a single, distinctive line into a point of return and a point for potential re-entry.)

The reference to the embedded file takes the example form:

< geomdata Label Entry1

where geomdata is the name of the channel on which the embedded file is mounted, and Label is a case insensitive keyword. Entry1 is a character string that matches the marker placed in the embedded file. The match must be exact - including the case of any letters.

The following rules apply:

Input is read from the embedded file beginning at the line after the label.

Input is read from the embedded file until exhausted or a return marker (< UP) is encountered.

The character string used as a label must not contain more than ten characters.

The sections of a file delimited by marker labels may be accessed in any order, may be accessed repeatedly or not accessed at all from the file in which it is embedded.

The transfers to and from embedded files may be made at any points in the input.

Return markers (< UP lines), with or without labels, are ignored in the main input file.

A file containing such labels may be executed normally in one calculation and referenced from another calculation as an embedded file to 'borrow' some of its contents.

Example uses are:

Several cases are to be executed using input files containing many sections of data that are common. These may be stored in a file with suitable '< Up' labels at the start of each section. The 'master' file may direct the input to the appropriate sections of the common file and be confined to containing only the input that is to be varied.

A file of material specifications, each prefixed by a distinctive '< Up' label, may be prepared. An input file may include references to these specifications. A similar mechanism could be used for sets of energy group boundaries or subsets of a geometry model.

A block of data, extending over several lines, is to be repeated many times. Duplicating the block would make the input file long; subsequent edit activity on the data in the block would be tedious. As an alternative, it may be placed in a separate file which is referenced by a sequence of duplicated embedding instructions.

Error conditions are caused by the following:

- Reference to a named channel that is not defined in the 'datasets' file.
- Reference to a marker that cannot be found in the included file.
- Nesting included files to a depth greater than nine.
- Using more than ten characters for a marker string.

Restrictions, identified in the documentation of the host program, are as follows:

- The transfers between files are controlled from within the RCARD free format reading routines. Any sections of a host program that have their own 'private' reading mechanisms will fail if the required data is not in the main input file.
- If the host program includes a preliminary scan to identify particular keywords or to produce a listing of the input file then, depending on the mechanism used, the scan might not descend into embedded files.

An example demonstrating the use of this facility is given in Appendix 7.

2.9 DATSETS AND INTFACE FILES

The datasets and intface files used by MONK8 specify the location of the files required during a calculation. The datasets file is used to specify the input, output, nuclear data, burnup and intface file locations, using either relative or absolute pathnames. The intface file is used for reactor physics calculations only, and identifies the files associated with each interface.

An example datasets file for a reactor physics burnup case is given below:

```
WIMS      /answers/qaset/datasets/w86e69v1.dat
INPUT     input/case01.dat
OUTPUT    output/case01.out
BURN      case01.burn
INTFACE   input/case01.intface
```

The MONK8 code uses predefined channel names within the datasets file to identify the file from which the corresponding data must be read (additional predefined channel names are listed in the User Guide [2]). In the above example the datasets file is for use with a reactor physics case which must use the WIMS format nuclear data libraries. The above example therefore assigns the nuclear data library to the WIMS named channel with an absolute pathname to w86e69v1.dat. The INPUT and OUTPUT channels use relative pathnames to subdirectories containing the input and output files respectively. The BURN channel is only required for a burnup case and is used to store temporarily the input for the burnup calculation. The INTFACE channel is used to identify the location of the file containing the list of interface file names (see description below).

An example datasets file for a point energy criticality case is given below:

```
DICE      /answers/qaset/datasets/dicedat1c.dat
THERM    /answers/qaset/datasets/therm1a.dat
INPUT    input/case02.dat
OUTPUT   output/case02.out
```

In this example the nuclear data is read from the named channel DICE to which is assigned the appropriate nuclear data library. The INPUT and OUTPUT channels are as before. Note that neither the INTFACE nor the BURN channels are required for criticality cases and will be ignored if used. The additional channel, THERM, is used to assign the nuclear data library containing the thermal data needed for hydrogen bound in either water or polythene.

All reactor physics calculations must have an INTFACE channel specified in the datasets file. The intface file placed on the INTFACE channel contains a list of names, one per interface, which form the root of the name assigned to each of the five files of that interface. An example intface file (for a calculation requiring up to six interfaces) is given below:

```
1      'int1'
2      'int2'
3      'int3'
4      'int4'
5      '/machine/dir1/dir2/b0_ints'
6      'b1_ints'
```

The format is thus a number (mandatory but not used - see below) followed by a file or pathname in single quotes. The file specified on interface 5 (/machine/dir1/dir2/b0_ints) shows that absolute pathnames can be included in the intface file if required. The remainder of the files specified in the intface file will be expected to be in the directory from which MONK8 was launched. In this example the use of interface 1 would result in five files being generated thus:

```
int11.data      on channel 31
int12.data      on channel 32
int13.data      on channel 33
int14.data      on channel 34
int15.data      on channel 35
```

and the use of interface 5 would access the files:

```
/machine/dir1/dir2/b0_ints1.data      on channel 51
/machine/dir1/dir2/b0_ints2.data      on channel 52
/machine/dir1/dir2/b0_ints3.data      on channel 53
/machine/dir1/dir2/b0_ints4.data      on channel 54
/machine/dir1/dir2/b0_ints5.data      on channel 55
```

Therefore the name specified in the intface file has 'n.data' appended to it, where n runs from 1 to 5 for each of the files required for the interface. Similarly the channel number used is

incremented as shown. In this release of MONK8 the channel number assigned to the first file of the interface is given by

$$\text{channel_number} = ((\text{interface} - 1) * 5) + 31$$

In MONK8 the number at the beginning of each line of the interface file is ignored and the channel numbers are calculated as above. This format differs from that of MONK5W where the number represents the channel number for the first of the five channels required for each interface file. Thus a suitable interface file for a MONK5W case would be:

```
11      'int1'  
16      'int2'  
21      'int3'  
26      'int4'  
31      '/machine/dir1/dir2/b0_ints'  
36      'b1_ints'
```

3 Summary

This report has introduced the main new features of MONK8A, which provide a range of additional options to assist both the criticality and the reactor physics assessor:

- the inclusion of a reactor physics capability into MONK8 which gives the assessor access to leading-edge modelling capabilities and nuclear data libraries which are constantly improving
- new geometry bodies to enable complex models to be constructed more easily
- the option selectively to halt a case based on the value of k-effective thus saving time if running many consecutive cases, particularly for scoping calculations
- a more sophisticated albedo treatment to improve accuracy and efficiency in scoping calculations
- enhanced file handling capabilities allowing sections of the input data to be stored in files separate from the main input file thus allowing the same data to be referenced from several input files
- the capability for reactor physics assessors to use their existing material, geometry and hole data with only a few changes necessary simplifying the move to the new code
- a splitting and roulette mechanism capable of using both space and energy weighting values to encourage samples to score in selected regions thus improving the efficiency of the code

A major programme of pre-release testing has been completed which has not only considered the new features of the code but also a large set of older models to ensure that existing features have not been disturbed. This test programme has also included significant contributions from the user community with beta code versions. Further details on the new features are contained in the user guide issued to accompany MONK8A. Additional advice and information is available via the usual ANSWERS customer support channels.

4 References

- [1] The ANSWERS Software Service
An Introduction to the New Features of MONK7B
ANSWERS/MONK(97)5
- [2] The ANSWERS Software Service
MONK User Guide for Version 8
ANSWERS/MONK(98)6

Appendices

CONTENTS

Appendix 1	Reactor Physics Examples
Appendix 2	Hemisphere and Hexagonal Prism Examples
Appendix 3	Albedo Material Example
Appendix 4	Simple Rotation Example
Appendix 5	K-effective Halt Examples
Appendix 6	Splitting and Roulette Example
Appendix 7	Embedded File Example
Appendix 8	Conversion of MONK5W Input to MONK8A

Appendix 1

Reactor Physics Examples

This appendix contains four examples highlighting some of the reactor physics features of MONK8. The examples comprise a burnup case, and adjoint tracking case, a point estimator case and a fixed source case. Each case comprises an input listing with relevant lines highlighted in bold and marked with a note number of the form !(%n), where n is the note number. At the end of each listing are the detailed descriptions corresponding to each of the note numbers in the listing.

A1.1 BURNUP EXAMPLE

The following listing is an example input for a depletion case.

```

NCYCLE 1 8 ! (%1)

BEGIN CONTROL DATA
  STAGES - 10 125 250
  SUPERHIST 1 1.0
END

BEGIN MATERIAL DATA
  WIMS
  BURN ! (%2)

  READ FROM INTERFACE 1 ! (%3)
  WRITE TO INTERFACE 2 ! (%3)

  PONE
  NMAT 3

  GOTO 1 2-100 ! (%4)
  * 2.24 w/o U235/U was MAT 1
  MATERIAL 1 - 1 300 2235 5.1E-4 8238 2.2E-2 6016 4.5020E(25)
  * Water was MAT 3
  MATERIAL 2 - 1 300 3001 3.2E-2 6016 1.6E-2 ! (%5)
  * Zirconium was MAT 2
  MATERIAL 3 - 1 300 91 4.3E-2 ! (%5)
  LABEL 1 ! (%4)

  GOTO 2 1 ! (%6)
  MATERIAL 1 - 1 300 1=1 ! (%7)
  MATERIAL 2 - 1 300 3=2
  MATERIAL 3 - 1 300 3=3
  LABEL 2

  *
  MAPPING 1 3 2 1 3 1 ! (%8)
  END

BEGIN BURNUP DATA ! (%9)
  READ 2
  WRITE 1
  RATING 1 25.0
  QUAL 1 1 ! (%10)
  STEPS 1 1
  QUAL 1 2 - 8 ! (%10)
  STEPS 4 40.0
  END

  * Switch off FG print for cycles > 1
  QUAL 1 2-8 ! (%11)

```

```

NO PRINT
BEGIN MATERIAL GEOMETRY
  NEST 2
  * region 1
  1 ZCYL H1 0.715 9.0 0.0
    VOLUME 5 1 6.27467 2 2.13825 4 3.39292 !(%12)
           5 1.51770 6 1.13097
  * region 2
  2 CUBOID 3 1.04575 1.04575 9.0 -1.04575 -1.04575 0.0 !(%13)
    VOLUME 1 3 24.9148
  * total reflection at outer boundary
  ALBEDO 1 1 1 1 1 1
END

BEGIN HOLE DATA
  GLOBE
  * rings
  5
  * can
  0.715 2
  0.66 5
  * fuel
  0.618 1
  0.4 4
  0.2 6
  * otherwise can
  2
END

BEGIN SOURCE GEOMETRY
  ZONEMAT
  ZONE 1 PART 1 /
END

BEGIN ENERGY DATA
  SUBGROUP 69
END

BEGIN ACTION TALLIES
  NO ACTION TALLIES !(%14)
END

FINISH !(%15)

```

The corresponding datasets file for this case would need to include the channel name **BURN** to which is attached a file used by the burnup process and **INTERFACE** which identifies the file describing the interface file names (this could be the original **MONK5W** intface if required).

The explanations for each of the notes are:

- 1 The **NCYCLE** keyword is followed by the first and last cycle numbers respectively and determines how many cycles the burnup case will run for. If a suitable set of burnup interface files exist then the first number after **NCYCLE** can be greater than 1 so the material number densities will be picked up from the interface files.
- 2 The **BURN** keyword is required to instruct the code both to perform a burnup calculation and to write interface files suitable for the burnup cycling mechanism.

- 3 These keywords instruct the code to read material compositions from the interface files: this will automatically happen for cycles greater than 1. The BURNUP DATA input unit must therefore write its data to this same interface. Similarly interface 2 must be read by the BURNUP DATA input unit (see note 8). Note that no data will be expected on interface 1 for cycle 1 in a subgroup case where material compositions are specified.
- 4 The GOTO keyword is used to skip over input which is not needed during one or more cycles. The keyword must begin in column 1, be in upper case, and has the format GOTO label_number cycle_selection. The value of label_number must correspond to a line 'LABEL label_number'; the value of cycle_selection can be a single number, an explicit list of numbers, or a list specified using '-' to define a range of cycles as in this example which is instructing the code to skip the lines between GOTO 1 ... and LABEL 1 for cycles 2 to 100 inclusive (the end of range value can be larger than the NCYCLE end of cycles value). The example is using this construct to ensure that the initial material compositions are read during cycle 1 only after which they are ignored.
- 5 The highlighted value in each of the lines is used to control how the material is treated in a burnup calculation. The fuel material must have a value of 1, other burnable materials must have a value of 2, and all other materials must be given a value of 3. The use of 1 and 2 instructs the code to calculate the material cross-section when a collision occurs, thus taking account of the current subgroup and number density for each nuclide comprising the material.
- 6 This example of the use of GOTO is to skip around the '=' material instructions for cycle 1 only, after which these lines will be read. Thus the 'GOTO 2 1' line instructs the code to jump to LABEL 2 for cycle 1 only.
- 7 These lines are the format in which the material data must be entered for cycles greater than 1 in a burnup case. After cycle 1 the compositions are read from the interface files, but it is necessary to specify the remaining material data.
- 8 During a burnup case the same initial material would burn up at different rates in different regions of the model. The MAPPING keyword allows the geometry to be assembled using different material numbers in different regions and then mapping these back to a small number of initial materials. After the first cycle the materials will have burned differently causing their composition to differ. Cycles after 1 will therefore use the geometry material numbers and save their data to the interface files accordingly. In this example geometry materials 1, 4 and 6 will initially have the composition of defined material 1; geometry materials 2 and 5 will have the composition of defined material 3; and geometry material 3 will have the composition of defined material 2.
- 9 The BURNUP DATA input unit defines the input and output interfaces to be used, the power specification, the number of substeps, and the length of each substep in days. Note that the input and output interfaces must match those used by the MATERIAL DATA input unit to ensure that the output interface containing the burnup data matches the input interface read by the material data input unit. In the example the data are read from interface 2 and written to interface 1 (see note 3).

- 10 In many cases each cycle will have different RATING and STEPS data: the QUAL instruction can be used to control which data are read for each cycle. The format of the QUAL instruction is 'QUAL number_of_lines cycle_selection' where number_of_lines defines the number of lines after the QUAL line which are controlled by it, and cycle_selection is as defined in note 4. In this example the RATING is fixed, but 'QUAL 1 1' is used to ensure that the following 1 line is only read for cycle 1, while the 'QUAL 1 2 -8' is used to ensure that the line following it is used for cycles 2 to 8 inclusive.
- 11 This set of input is not mandatory for a burnup case, but it is advisable to use it to switch off the printout of the FG analysis for cycles greater than 2. This printout can be very large and for cases with many cycles its use can save considerable file space. The instruction NO PRINT prior to a BEGIN... statement suppresses all the normal printout from that unit. By qualifying the NO PRINT with a QUAL instruction it is possible to limit the printout to cycle 1 only as in this example.
- 12 In a burnup case it is necessary to know the volumes of the materials comprising each region so that the correct reaction rate per unit volume can be calculated for each material. This is a particular requirement when the region contains a HOLE material. Thus for each region it is necessary to calculate the volume of all the fissile and burnable materials comprising it. If the region is used several times because the PART it is in is referenced more than once, then the material volumes will automatically be adjusted to compensate. This is unlike MONK5W where the volumes entered by the user had to be manually adjusted to compensate for PART reuse.
- 13 The volumes of regions which contain only a single material can be calculated by the code automatically in most NEST and CLUSTER cases . However it is possible to enter a hand calculated volume and indeed it is mandatory for the general part where the code cannot easily determine the appropriate volumes. An additional use for this method is to enter a very large value for the volume of regions which contain a burnable material which the user does not wish to burn up: the large value causes a negligible reaction rate per unit volume to be used in the burn calculations.
- 14 The action tally output can use significant file space and it is thus recommended that NO ACTION TALLIES be specified to suppress all the output. Alternatively a QUAL instruction could be used to specify printout for certain cycles only.
- 15 The FINISH instruction defines the end of the current NCYCLE block. If the current cycle is not the last cycle then the code will search back until it reads an NCYCLE keyword after which it will increment the cycle and continue processing the input for that cycle.


```
ADJOINT                                ! (%2)
MULTI GROUP 3
BIAS                                    ! (%3)
1. 0E- 04   1. 0E- 03   1. 0E- 05
END

BEGIN ACTION TALLIES
NONORM
END
```

Notes:

- 1 The multigroup case must read the material data from a set of interface files written by WIMS.
- 2 The ADJOINT keyword is entered in the ENERGY DATA unit and instructs the code to do an adjoint calculation.
- 3 The BIAS keyword and following data are used by the code to adjust the cross-sections of the materials used in the case. The number of values entered must equal the number of energy groups in the problem (3 in this case) and are used to improve the rate of convergence of the calculation. In a homogeneous case with no spatial dependence the BIAS values are the forward fluxes. In an a typical inhomogeneous case where the flux is spatially dependent the BIAS values will be some average of the fluxes in each region.


```

BEGIN SOURCE GEOMETRY
  ZONEMAT
  REGION 2 IN TYPE 11 /
END

```

```

BEGIN ENERGY DATA
  WIMS
  SUBGROUP 3
END

```

```

BEGIN ACTION TALLIES
  NONORM
END

```

```

BEGIN POINT ESTIMATOR
  FLUX                                     ! (%2)
  STAGES 200                               ! (%3)
  TARGETS 9                                ! (%4)
  COORDS 5.0 5.0 5.0                       ! (%5)
  TRACKS 25000                              ! (%6)
  DIRECTIONS 1                              ! (%7)
  COORDS 15.0 5.0 5.0                       ! (%8)
  TRACKS 25000
  DIRECTIONS 1
  COORDS 25.0 5.0 5.0
  TRACKS 25000
  DIRECTIONS 1
  COORDS 5.0 15.0 5.0
  TRACKS 25000
  DIRECTIONS 1
  COORDS 15.0 15.0 5.0
  TRACKS 25000
  DIRECTIONS 1
  COORDS 25.0 15.0 5.0
  TRACKS 25000
  DIRECTIONS 1
  COORDS 5.0 5.0 5.0
  TRACKS 25000
  DIRECTIONS 1
  COORDS 15.0 25.0 5.0
  TRACKS 25000
  DIRECTIONS 1
  COORDS 25.0 25.0 5.0
  TRACKS 25000
  DIRECTIONS 1
END

```

Notes:

- 1 The volumes of materials comprising each region must be entered for a point estimator calculation. The volumes are summed to give a region volume which is then factored into the calculation.
- 2 These keywords both identify the start of the point estimator input unit and instruct the code to perform a point estimator calculation.
- 3 Two scores can be calculated by the point estimator method: FLUX and FISSION. Of these two scores only one can be calculated during a run.

- 4 STAGES is followed by a list of one or more stage numbers identifying those stages at which the point estimator calculation is to be done. Specifying a value for STAGE which is greater than the final stage of the calculation will result in a warning message and the STAGE value being set to the final stage of the calculation.
- 5 Having selected the scored item and those stages at which it is to be determined it is necessary to specify the number target points at which to tally the score. The keyword TARGETS is followed by the number of target points to be processed.
- 6 For each of the target points three sets of data are required: the point co-ordinates, the number of tracks to be traced (see note 7) and the direction of those tracks (see note 8) . The co-ordinates are specified in the global frame NOT the PART/BODY frame.
- 7 TRACKS is followed by the number of tracks to be started from the current target point. The total number of tracks started is the product of this number and the number of directions (see note 8).
- 8 DIRECTIONS specifies the number of starting directions and is redundant in MONK8. It is included for back compatibility with MONK5W and has a default value of 1 if it is not input.

A1.4 FIXED SOURCE EXAMPLE

The following listing is an example input for a fixed source case. The example shows the items which must be included, each of which will be explained in the text following the example. The explanations are numbered thus !(%n) where n is note number.

BEGIN CONTROL DATA

```
* fixed source keywords entered in control data unit
  FIXED SOURCE                               !(%1)
  STAGES - 1 20 600
* use SUPERHIST to set the number of generations tracked
  SUPERHIST 7 1.0                             !(%2)
END
```

BEGIN MATERIAL GEOMETRY

```
NEST 4
* region 1
  1 ZCYL BH 1 50.0 50.0 0.0
  2 ZCYL 1 100.0 100.0 -50.0
  3 ZCYL 3 150.0 150.0 -100.0
  4 ZCYL 3 200.0 200.0 -150.0
* total reflection at outer boundary
  ALBEDO 1 1 1
END
```

BEGIN HOLE DATA

```
HOLES 1
  GLOBE 2 50.0 1 25.0 1 1
END
```

BEGIN MATERIAL DATA

```
WIMS
  NMAT 3
* 2.24 w/o U235/U
  MATERIAL 1 - 1 300 1 2235 5.1E-6 8238 2.2E-4 6016 4.5020E-2
* Zirconium
  MATERIAL 2 - 1 300 2 91 4.3E-2
* Water
  MATERIAL 3 - 1 300 3 3001 3.2E-2 6016 1.6E-2
END
```

BEGIN SOURCE GEOMETRY

```
* 8 sources in an XYZ geometry
* the format of the data is:
* POINT probability spectrum X Y Z
* LINE probability spectrum Xstart Ystart Zstart Xend Yend Zend
* LAMINA probability spectrum Xstart Ystart Zstart Xend Yend Zend
* VOLUME probability spectrum Xmin Ymin Zmin Xmax Ymax Zmax

* note:
* - the probabilities will be summed and normalised to unity
* - the spectrum 0 is the default fission spectrum
* - a lamina must have one co-ordinate the same
* - a line must have two co-ordinates the same
  MULTIPLE XYZ 8                               !(%3)
  POINT 1.0 1 2.0 3.0 4.0                       !(%4)
  LINE 1.0 2 2.0 3.0 4.0 2.0 3.0 14.0
  LINE 1.0 3 2.0 3.0 4.0 12.0 3.0 4.0
  LINE 1.0 4 2.0 3.0 4.0 2.0 13.0 4.0
```

```
LAMINA 1.0 5 2.0 3.0 4.0 12.0 13.0 4.0
LAMINA 1.0 6 2.0 3.0 4.0 2.0 13.0 14.0
LAMINA 1.0 7 2.0 3.0 4.0 12.0 3.0 14.0
VOLUME 1.0 0 2.0 3.0 4.0 12.0 13.0 14.0
```

```
* Define number of energy groups in library
ENERGY GROUPS 69 ! (%5)
```

```
* Enter the spectra for each source
* Note that the maximum spectrum number entered above was 7,
* thus 7 spectra required
* Order of data is high energy first to low energy last
```

```
SPECTRUM 1 10*1.0 1000000.0 58*1.0 ! (%6)
SPECTRUM 2 11*1.0 1000000.0 57*1.0
SPECTRUM 3 12*1.0 1000000.0 56*1.0
SPECTRUM 4 13*1.0 1000000.0 55*1.0
SPECTRUM 5 14*1.0 1000000.0 54*1.0
SPECTRUM 6 15*1.0 1000000.0 53*1.0
SPECTRUM 7 16*1.0 1000000.0 52*1.0
```

```
END
```

```
BEGIN ENERGY DATA
```

```
WIMS
```

```
SUBGROUP 69
```

```
END
```

```
BEGIN ACTION TALLIES
```

```
NO ACTION TALLIES
```

```
END
```

The explanations for each of the notes are:

- 1 The FIXED SOURCE keywords instruct the code to do a fixed source calculation.
- 2 The SUPERHIST keyword is used to set the number of generations tracked. When the specified number of generations have been tracked the sample is terminated and an estimate of any further contribution the sample may have made is tallied. The number of generations is case dependent, but in general a decreasing the value gives more approximate results due to the increasing error on the tally estimate at particle termination. Increasing the number of generations tracked will increase the accuracy of the result but at the expense of increasing the run time.
- 3 The MULTIPLE keyword allows the user to specify more than one source body. In this case eight sources are specified in XYZ geometry (to ensure the correct lamina surface is treated if used).
- 4 POINT is one of the four source types allowed and is used as an example of how to specify the sources. The keyword is followed by a relative probability of sampling from this source, a spectrum identifier, and a list of co-ordinates specifying the limits of the source. The sampling probabilities of all the sources are summed and normalised to unity within the code. The spectrum identifier is either 0 for a fission spectrum or greater than 0 for a user defined spectrum (see note 6).

- 5 **The number of ENERGY GROUPS in the problem must be specified here if this input unit precedes the ENERGY DATA input unit and SPECTRA have been specified. This requirement is to ensure that the correct number of items are read for each spectrum.**

- 6 **Each user defined spectrum identifier must have the corresponding spectrum specified. In this line the SPECTRUM keyword is followed by a number identifying which spectrum it refers to and then a list of spectrum data in the order high to low energy. The spectra do not need to be entered in numerical order, but there must be the correct number of spectra specified.**

Appendix 2

Hemisphere and Hexagonal Prism Examples

This appendix contains three examples describing the use of the new FG bodies available in MONK8: the hemisphere body, the hexagonal prism body and the hexarray part for tessellating hexagonal prisms.

A2.1 HEMISPHERE EXAMPLE

The example below comprises an enriched uranium sphere in a void. The sphere in this example has been constructed using two hemispheres with their flat surfaces coincident: these are highlighted in bold.

```
BEGIN MATERIAL DATA
MONK 1 3 NUCNAMES
CONC      U234 0.00049  U235  0.045  U238  0.0025
END
```

```
BEGIN MATERIAL GEOMETRY
PART 1 CLUSTER
+ZHEMI  M1  0 0 0  8.741
-ZHEMI  M1  0 0 0  8.741
BOX MO  - 10 - 10 - 10 20 20 20
END
```

```
BEGIN CONTROL DATA
STAGES - 1 50 1000  STDV 0.0015
END
```

```
BEGIN SOURCE GEOMETRY
ZONEMAT
ZONE 3 PART 1 /
END
```

A2.2 HEXAGONAL PRISM EXAMPLE

The example below comprises an FG general part containing an outer box within which are two nested, rotated hexagonal prisms. The hexagonal prisms have been rotated through 90 degrees using the simple rotation option described in section 2.4. Both hexagonal prisms are initially oriented with their axes parallel to the z-axis using the ZHEX body: XHEX and YHEX bodies are similarly available. The inner hexagonal body contains a triangle hole to demonstrate that these new bodies can contain holes as well as other FG bodies.

BEGIN MATERIAL DATA

```
MONK 5 6 NUCNAMES
ATOM 9.2      U238 0.97621  U235 0.02379  0  2.0
WGT  2.692   AL27  1.0
WGT 0.9982   0 0.888089   J2H/H20 0.111881
ATOM 7.86    FE  1.0
WGT 0.9982   0 0.888089   J2H/H20 0.111881
END
```

BEGIN MATERIAL GEOMETRY

```
PART 1
* The format is:
* ZHEX material origin half-width height simple rotation
  ZHEX  1      0 0 0      8      20      ZROT 90
  ZHEX  2      0 0 0      9      20      ZROT 90
BOX  3  -10 -10 0 20 20 20
ZONES
H1  +1
MO  +2 +3 -1
M5  +3 -2
END
```

BEGIN HOLE DATA

```
TRIANGLE
3.0 1.0 1.2 WRAP 3 8 8
1 2 3 0 0
END
```

BEGIN CONTROL DATA

```
STAGES -1 20 1000
END
```

BEGIN SOURCE GEOMETRY

```
ZONEMAT ZONE 1 PART 1 / MATERIAL 1
END
```

A2.3 HEXAGONAL ARRAY EXAMPLE

The example listed below demonstrates the use of both the ZHEX body and its corresponding ZHEXARRAY tessellating part. The format of the ZHEXARRAY input is described in section 2.2 of the main text. The XHEXARRAY and YHEXARRAY parts are also available for use with the XHEX and YHEX bodies respectively.

```

BEGIN MATERIAL GEOMETRY
PART 1
  ZHEXARRAY M1 7. 21688 20. 0 2                ! (%1)
  2 3 4 5 4 3 2                                ! (%2)
PART 2
  ZHEXARRAY M1 2. 5      20. 0 3                ! (%3)
  6 7 6 7 6 7 6 7 6 7 6 7 6 7 6 7 6          ! (%4)
PART 3
  ZHEXARRAY M1 2. 5      20. 0 4                ! (%5)
  9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9      ! (%5)
  8 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9
PART 4
  ZHEXARRAY M1 2. 5      20. 0 1                ! (%6)
  10
PART 5
  ZHEXARRAY M1 2. 5      20. 0 2                ! (%7)
  11 12 11 12 11 12 11
!
! PARTs 6 to 12 each contains a ZHEX body with a height of 20cm and
! varying half-widths
PART 6
  NEST ZHEX M2 0. 0. 0. 0. 54127 20. 0
PART 7 SAME 6
PART 8
  NEST ZHEX M2 0. 0. 0. 0. 39365 20. 0
PART 9 SAME 8
PART 10
  NEST ZHEX M2 0. 0. 0. 2. 16506 20. 0
PART 11
  NEST ZHEX M2 0. 0. 0. 0. 86603 20. 0
PART 12 SAME 11
END

```

- 1 This line defines a HEXARRAY part with its axis parallel to the z-axis. The part is 20cm high with a half-width of 7.21688cm. The interstitial material is material 1 and there are 2 hexagons along each face, giving a total of 7 hexagons within the part.
- 2 This line identifies the 7 subsidiary parts required to complete the part definition.
- 3 This line again defines a HEXARRAY part with its axis parallel to the z-axis. The part is 20cm high with a half-width of 2.5cm. The interstitial material is material 1 and there are 3 hexagons along each face, giving a total of 19 hexagons within the part.
- 4 This line identifies the 19 subsidiary parts required to complete the part definition.
- 5 The data for this part defines a ZHEXARRAY with 4 hexagons along each face, giving a total of 37 hexagons within the part: the subsidiary parts for each hexagon are listed on the following lines.

- 6 This input demonstrates the use of the part to contain a single subsidiary part.
- 7 Another example of the ZHEXARRAY part with 7 subsidiary parts.

Appendix 3

Albedo Material Example

The Albedo Data input unit provides the user with the capability to model more accurately and efficiently the albedo processes than was available in MONK7. The albedo data required to specify the behaviour of the material can be entered explicitly, however the recommended procedure is to use the library of albedo data whenever possible.

The material number specified in the Albedo Data input unit corresponds to a material in the geometry data preceded by a negative sign. For example albedo material 1 would be referenced in the geometry as material -1. Note that albedo materials cannot be used in HOLE geometries.

Two examples are listed below. The first example shows how to use the library to access the albedo data. The second example shows how to enter the albedo data explicitly.

A3.1 Using the library

This example shows how to use the library to define albedo material 1 as jabroc. Within the material geometry data this albedo material will be referenced as material -1. Note that the library data is stored in a 16 energy group scheme.

```
BEGIN ALBEDO DATA
LIBRARY
16
1
MATERIAL 1 JABROC
END
```

A3.2 Explicitly entering the data

These data comprise an energy group scheme followed by a set of probabilities in that group scheme specifying the behaviour of each material. These probabilities comprise incident and emergent angular parameters followed by an upper diagonal matrix of incident to emergent energy group probabilities.

```
BEGIN ALBEDO DATA
GROUPS 16
1.00e+1 7.50e+0 6.50e+0 5.50e+0 4.50e+0 3.50e+0 2.50e+0 1.50e+0 5.00e-1 5.00e-2
5.00e-3 5.00e-4 5.00e-5 5.00e-6 5.50e-7 7.07e-8 0.00e+0
WATER
0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.00
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4 0.4 0.4 0.4 0.4 0.4 0.0
0.018 0.021 0.013 0.010 0.011 0.019 0.028 0.038 0.031 0.014 0.011 0.010 0.009 0.008 0.022
0.037
0.005 0.036 0.018 0.013 0.020 0.023 0.037 0.035 0.016 0.012 0.010 0.009 0.009 0.025 0.040
0.006 0.044 0.016 0.021 0.024 0.039 0.038 0.018 0.013 0.011 0.011 0.010 0.027 0.044
0.010 0.044 0.023 0.026 0.044 0.044 0.019 0.016 0.013 0.012 0.011 0.030 0.049
0.020 0.069 0.031 0.054 0.052 0.023 0.018 0.015 0.014 0.013 0.034 0.053
0.042 0.041 0.058 0.060 0.029 0.022 0.019 0.016 0.015 0.038 0.060
0.031 0.077 0.076 0.036 0.027 0.024 0.021 0.018 0.045 0.072
```

0.093 0.109 0.056 0.042 0.034 0.028 0.025 0.056 0.084
0.124 0.100 0.072 0.056 0.044 0.036 0.069 0.101
0.091 0.129 0.096 0.070 0.052 0.090 0.123
0.091 0.129 0.100 0.072 0.112 0.144
0.093 0.134 0.100 0.144 0.176
0.092 0.138 0.195 0.219
0.106 0.280 0.280
0.352 0.384
0.824

CONCRETE

0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.00
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4 0.4 0.4 0.4 0.4 0.4 0.0
0.000 0.097 0.079 0.072 0.062 0.070 0.060 0.047 0.032 0.017 0.013 0.012 0.010 0.008 0.005
0.010
0.000 0.136 0.104 0.082 0.086 0.070 0.053 0.035 0.019 0.014 0.012 0.011 0.008 0.005 0.010
0.000 0.168 0.114 0.108 0.082 0.061 0.039 0.021 0.016 0.013 0.012 0.009 0.005 0.011
0.000 0.194 0.149 0.103 0.073 0.045 0.024 0.018 0.015 0.013 0.010 0.006 0.012
0.000 0.265 0.147 0.095 0.055 0.028 0.021 0.017 0.016 0.011 0.006 0.013
0.000 0.265 0.144 0.078 0.038 0.028 0.022 0.018 0.014 0.008 0.016
0.054 0.279 0.121 0.054 0.036 0.028 0.022 0.017 0.009 0.018
0.254 0.230 0.078 0.046 0.033 0.026 0.018 0.010 0.018
0.278 0.184 0.082 0.052 0.038 0.026 0.015 0.043
0.182 0.227 0.106 0.065 0.041 0.022 0.058
0.188 0.237 0.108 0.060 0.030 0.072
0.179 0.238 0.101 0.046 0.099
0.155 0.228 0.084 0.148
0.133 0.200 0.246
0.069 0.503
0.744
END

Appendix 4

Simple Rotation Example

Prior to the addition of the simple rotation option the FG input required rotations to be specified by altering the alignment vectors in at least two of the three possible dimensions. In those cases where the rotation is about the X, Y or Z axis the new option can be used to simplify the input. Consider the simple case below:

```

BEGIN MATERIAL GEOMETRY
PART 1
BOX 1  0.0 0.0 0.0 100.0 100.0 100.0
BOX 2  20.0 20.0 10.0 20.0 20.0 80.0
      VX  COS -10.0 SIN -10.0 0.0
      VZ  0.0 0.0 1.0
BOX 3  60.0 60.0 10.0 20.0 20.0 80.0
BOX 4  20.0 60.0 90.0 20.0 20.0 80.0
      VX  COS -10.0 SIN -10.0 0.0
      VZ  0.0 0.0 1.0
ZONES
/world/  M 0 +1 -2 -3 -4
/box1 /  M 1 +2
/box2 /  M 2 +3
/box3 /  M 3 +4
END

```

can now be written:

```

BEGIN MATERIAL GEOMETRY
PART 1
BOX 1  0.0 0.0 0.0 100.0 100.0 100.0
BOX 2  20.0 20.0 10.0 20.0 20.0 80.0
      ZROT 10.0
BOX 3  60.0 60.0 10.0 20.0 20.0 80.0
BOX 4  20.0 60.0 90.0 20.0 20.0 80.0
      ZROT 10.0
ZONES
/world/  M 0 +1 -2 -3 -4
/box1 /  M 1 +2
/box2 /  M 2 +3
/box3 /  M 3 +4
END

```


Appendix 5

K-Effective Halt Examples

The k-effective halt option allows the user to stop the case should the value of k-effective be outside the specified limits during and after a specified stage. The default stage at which the case can first be halted is 3, and either or both of the upper and lower limits can be set. When one or both limits are set the code will begin comparing the specified values against k-effective at the first non-settling stage (stage 1), but it will not be able halt the case until the specified stage has been reached. Also, it is necessary for three consecutive values of k-effective to fall outside the specified limits before the case can be halted: thus there is no benefit in setting KSTAGE less than 3. Note that no checking is allowed during the settling stages.

The keywords for controlling this option are entered in the CONTROL DATA input unit. The keywords are:

KLOWER lower_limit - specifies the lower limit, a real value
KUPPER upper_limit - specifies the upper limit, a real value
KSTAGE start_stage - specifies the stage at which checking begins, an integer value

The following examples demonstrate the use of this option:

A5.1 Stop if k-effective becomes greater than 1.5, start checking at stage 3 (the default)

```
BEGIN CONTROL DATA
STAGES - 10 20 1000
KUPPER 1.5
END
```

A5.2 Stop if k-effective becomes greater than 1.5, start checking at stage 5

```
BEGIN CONTROL DATA
STAGES - 10 20 1000
KUPPER 1.5
KSTAGE 5
END
```

A5.3 Stop if k-effective becomes less than 0.5, start checking at stage 3 (the default)

```
BEGIN CONTROL DATA
STAGES - 10 20 1000
KLOWER 0.5
END
```

A5.4 Stop if k-effective becomes less than 0.5, start checking at stage 4

```
BEGIN CONTROL DATA
STAGES - 10 20 1000
KLOWER 0.5
KSTAGE 4
END
```

A5.5 Stop if k-effective becomes greater than 1.5 or less than 0.5, start checking at stage 3 (the default)

```
BEGIN CONTROL DATA
STAGES - 10 20 1000
KLOWER 0.5
KUPPER 1.5
END
```

A5.6 Stop if k-effective becomes greater than 1.5 or less than 0.5, start checking at stage 7

```
BEGIN CONTROL DATA
STAGES - 10 20 1000
KLOWER 0.5
KUPPER 1.5
KSTAGE 7
END
```

Appendix 6

Splitting and Roulette

Example

The listing below contains a model with a small cylinder at its centre in which the results are to be scored. Due to the small size of this cylinder there would normally be few scoring events recorded during a typical run. The splitting and roulette feature allows the user to encourage samples towards the small cylinder, increasing the number of events scored.

The listing of the input data is given below with relevant input highlighted in bold and a note number with the format !(%n) attached:

```

BEGIN MATERIAL DATA
  WIMS
  READ FROM WIMS INTERFACE 3
END

BEGIN MATERIAL GEOMETRY
  NEST 5
  1 ZCYL 1 0.65 32.5 20.5 !(%1)
  2 CUBOID 2 1.32 1.32 33.0 -1.32 -1.32 20.0
  3 ZCYL 3 15.18 34.4996 18.0004
  4 ZCYL 4 25.30 34.4998 18.0002
  5 CUBOID 5 35.64 35.64 35.0 -35.64 -35.64 18.0
  NEST 4
  1 CUBOID T1 35.64 35.64 17.00000 -35.64 -35.64 0
  2 CUBOID 6 270.18 271.48 376.4 -270.18 -271.48 -134.6
  3 CUBOID 7 270.5 271.8 376.4 -270.5 -271.8 -136.6
  4 CUBOID 8 400 401.3 470.4 -400 -401.3 -136.6
  ALBEDO 0 0 0 0 0 0
END

BEGIN SOURCE GEOMETRY
  ZONEMAT
  ZONE 1 PART 2 /
  MATERIAL 1
END

BEGIN ENERGY DATA
  WIMS MULTIGROUP 69
  SPLITTING REDUCED MULTIGROUP 1 69 !(%2)
END

BEGIN SPLITTING GEOMETRY !(%3)
  R 4 0.0 1.32 5.06 10.12 300.0
  THETA 1 0.0 360.0 DEGREES !(%4)
  Z 7 -200.0 18.0005 19.0 20.0 33.0 34.0 34.4995 200.0
END

BEGIN IMPORTANCE MAP !(%5)
  DIMENSIONS 4 1 7 1 1 !(%6)
  COMPONENT R THETA Z ENERGY !(%7)
  READ FROM INPUT !(%8)
  1.0 1.0 1.0 1.0 !(%9)
  4.0 4.0 4.0 1.0
  32.0 32.0 4.0 1.0
  256.0 32.0 4.0 1.0
  32.0 32.0 4.0 1.0
  4.0 4.0 4.0 1.0
  1.0 1.0 1.0 1.0
END
BEGIN CONTROL DATA
  STAGES -10 250 250
END

```

Notes:

- 1 This is the small cylinder at the centre of the model in which the scoring is required.
- 2 The energy group scheme which will be used by the splitting and roulette processing must be specified. In this example the keywords '**SPLITTING REDUCED MULTIGROUP 1 69**' instruct the code that for **SPLITTING** we will be using a **REDUCED MULTIGROUP** group scheme with 1 energy group boundary with the top nuclear data library group of **69** (i.e. energy groups 1 to 69 inclusive will be mapped onto a single splitting energy group).
- 3 The boundaries of the mesh overlaying the model in which the splitting and roulette are to be processed are specified in the **BEGIN SPLITTING GEOMETRY...END** input unit. The data comprises mesh boundaries for each of the relevant axes, X, R, Y, THETA, or Z. The data specified after the axis has been input comprises an integer describing the number of intervals (NI) followed by a set of real numbers defining the boundaries (NI+1 values). One or more set of data can be dummied out if there is no variation in the importance along that axis. Note that the mesh must cover the entire model.
- 4 The THETA dimension data can be specified as either RADIANS or DEGREES.
- 5 The data comprising the importance map are entered in the **BEGIN IMPORTANCE MAP...END** input unit and comprises a set of values for each dimension of the problem specifying the relative importance of each mesh.
- 6 To ensure that the correct importance map is being used the number of dimensions for the importance map must be specified. These dimensions are X (or R), Y (or THETA), Z, ENERGY and TIME respectively. In this case the dimensions are 4 (R), 1 (THETA), 7 (Z), 1 (ENERGY), 1 (TIME - not yet supported).
- 7 This line informs the code that the R, THETA, Z and ENERGY components have been combined into a single set of data. An alternative input option is available where a set of data for each dimension is entered and the code combines them into a single set of data. This latter option is useful for large sets of data: in this example it is both easier and clearer to input the combined values.
- 8 This line is used to specify that the data are to be read from the following lines in the input file. Alternative options include reading from a separate file which is identified by a channel number (a suggested value is 29) in the input file and a corresponding entry in the datasets file matching the channel number to the data file.
- 9 The set of values comprising the importance map are then given, with those meshes into which we wish to encourage the samples to travel given the largest values. The values used will depend on the case being run, but a suggested initial ratio between meshes tracking towards the region of interest (shown in bold below) is 2, but if the case

needs a more aggressive importance map then that should be investigated. In this example the initial importance map was:

1.0	1.0	1.0	1.0
2.0	2.0	2.0	1.0
4.0	4.0	2.0	1.0
8.0	4.0	2.0	1.0
4.0	4.0	2.0	1.0
2.0	2.0	2.0	1.0
1.0	1.0	1.0	1.0

A short run was used to test the adequacy of this map. Unfortunately there was no increase in the number of samples reaching the region of interest. The ratio was therefore increased to 4:

1.0	1.0	1.0	1.0
4.0	4.0	4.0	1.0
16.0	16.0	4.0	1.0
64.0	16.0	4.0	1.0
16.0	16.0	4.0	1.0
4.0	4.0	4.0	1.0
1.0	1.0	1.0	1.0

This map resulted in more samples reaching the region of interest, but not a sufficient number to give reasonable statistics in the available run time. The decision was made to use a more peaked map around the region of interest:

1.0	1.0	1.0	1.0
4.0	4.0	4.0	1.0
32.0	32.0	4.0	1.0
256.0	32.0	4.0	1.0
32.0	32.0	4.0	1.0
4.0	4.0	4.0	1.0
1.0	1.0	1.0	1.0

The above map used in a short test run resulted in a significant number of samples scoring in the region of interest without significant loss of performance due to tracking the sample fragments from the splitting events. This map was thus the one chosen for the case.

Appendix 7

Embedded File Example

The following input (test1.inc) comprises both the material geometry and the source geometry. Here the '< up *label_name*' instructions (labelled 'matgeom' and 'source') are used to specify separate sections of the embedded file which are to be read at the appropriate point in the main input (test1.monk - see below).

```
*start of test1.inc
< up  matgeom
BEGIN MATERIAL GEOMETRY
PART 1 NEST
SPHERE M1 0 0 0 5.0
END

< up  source
BEGIN SOURCE GEOMETRY
ZONEMAT
ZONE 1 PART 1
END
*end of test1.inc
```

In the datasets file a channel must be assigned to the above embedded file with which it can be referenced in the main input file. In this case the channel is called embed.

```
*start of datasets
input  test1.monk
embed  test1.inc
output test1.out
*end of datasets
```

Finally, the main input file (test1.monk) includes the two references to each of the sections in the embedded file. This uses the '< *channel_name label label_name*' mechanism, where *channel_name* is 'embed' and *label_name* is first 'matgeom' and then 'source'

```
*start of test1.monk
BEGIN MATERIAL DATA
MONK 1 3  NUCNAMES
* material 1 - uranium metal (93.71% U235)
CONC  U234 0.00049  U235  0.045  U238  0.0025
END

< include label matgeom

BEGIN CONTROL DATA
STAGES -1 10 1000
KUPPER 1.05 KLOWER 0.90
END

< include label source
*end of test1.monk
```


Appendix 8

Conversion of MONK5W Input to MONK8A

A8.1 INTRODUCTION

The new code MONK8A comprises the criticality functions of MONK7B and the reactor physics capability of MONK5W. However the user image of MONK8A is based upon that of MONK7A, MONK7B and MCBEND (the 'MCANO' user image), all of which differ from the MONK5W user image. It is thus necessary to identify these changes and provide guidance to the new MONK8A user who is more familiar with MONK5W.

The New Features document, of which this is Appendix 8, details the new capabilities of the code with particular reference to the reactor physics options in a series of examples contained within the earlier Appendices. In addition there is a report [1] which describes the conversion of MONK6 inputs for use in MONK7A (and thus MONK8A). This latter document will prove useful in addressing the conversion of both the geometry and the hole data inputs.

Two important changes made to the code since the conversion document [1] was written must be identified. In section 3.1.1 when describing the WIMS Multigroup data the document states that the subgroup method is not available, this is no longer the case for a reactor physics calculation using MONK8A. Also in section 3.4.2 when describing region based tallies the document states that bodies containing subsidiary PARTs receive a zero score, this has been changed to ensure that scores are also tallied in any parent PARTs.

The additional input units in MONK8A derived from MONK5W which were not described by the conversion document [1] are:

- WIMS material data input
- Energy data for selecting the subgroup method
- Burnup data input
- Point Estimator data input

Two new keywords, DSVP and SAUR, have also been added to MONK8 to simplify the process of converting a MONK5W input for use by MONK8: these are described in section A8.6.

Each of the following sections describes those changes required, referring the user to the appropriate section of the above Appendices where necessary.

A8.2 WIMS MATERIAL DATA INPUT

Appendix 1 describes the material input unit when used for both WIMS subgroup and WIMS multigroup cases.

Example A1.1 highlights the changes to the MONK5W material input when used in MONK8A for a subgroup case detailing the material compositions. In addition this example shows how to use the QUAL keywords to select the material input format used with second and subsequent cycles of a burnup case. This example also demonstrates the keywords for selecting the READ and WRITE interfaces.

Example A1.2 shows the input for reading the material composition data from a WIMS interface.

A8.3 ENERGY DATA INPUT

Appendix 1 describes the energy data input unit when used for both WIMS subgroup and WIMS multigroup cases.

Example A1.1 highlights the energy data input for selecting a subgroup case using 69 energy groups.

Example A1.2 shows the energy data input for selecting an adjoint multigroup case in three groups with a bias function applied.

Note that the choice of SUBGROUP or WIMS MULTIGROUP must match the material data. Similarly the number of energy groups entered must match those in either the nuclear data library or the interface files depending on the method chosen.

A8.4 BURNUP DATA INPUT

The changes required to convert the MONK5W burnup data into a MONK8A burnup input unit are described in Appendix 1 section A1.1. This section comprises an example burnup input with additional notes describing important lines within the input which differ from MONK5W. The changes comprise small changes to the burnup data, principally the use of a BEGIN...END input unit, the new READ and WRITE interface assignment keywords, and some changes to the material input data (see above).

Attention must be drawn to the specification of material volumes for burnable materials in a model. In MONK5W if a PART (a Fractal Geometry entity equivalent to a MONK5W TYPE) in which volumes had been specified was referenced several times then the user would have to ensure that the volumes entered had been scaled by the number of uses of the PART. In MONK8A the code automatically scales the volumes by the number of uses of each PART, thus the volumes entered for MONK5W must be divided by the number of uses of the PART before they can be used in a MONK8A calculation.

A8.5 POINT ESTIMATOR INPUT

Example A1.3 in Appendix 1 describes the changes in input format required to use the MONK8A Point Estimator option. The input is now contained within a BEGIN...END input unit, but otherwise is similar to the MONK5W format.

Attention is drawn to the warning in section A8.4 regarding the use of material volumes in those PARTs which are used more than once within a model.

A8.6 NEW KEYWORDS

Two new keywords have been added to simplify the conversion of MONK5W inputs for use by MONK8: DSVP for suppressing the automatic summation of region and material volumes if PARTs are reused; SAUR to allow the material lists in the HOLE DATA to remain in the input.

DSVP has been included primarily for use in those burnup cases where the MONK5W input has already multiplied both the region and material volumes by the number of uses of each PART. MONK8 automatically scales these volumes by the number of uses of each PART which, with the MONK5W volume estimates, would result in incorrect volumes being used. The DSVP keyword suppresses the automatic volume scaling in MONK8 thus allowing the correct MONK5W volumes to be used. The keyword DSVP is placed directly after the BEGIN MATERIAL GEOMETRY keywords and will apply to all the volumes specified within the FG data.

SAUR has been included to allow MONK5W format HOLE DATA to be read by MONK8. The difference between the two formats was the need in MONK5W to specify a material list for each hole. The SAUR keyword instructs MONK8 to skip over the material list for each of the holes entered. Note that if SAUR is used then every hole must have a material list specified. The SAUR keyword is placed immediately after the BEGIN HOLE DATA keywords and will apply to all the holes.

A8.7 REFERENCES

- [1] The ANSWERS Software Service
An Introduction to MONK7A for MONK6 Users
ANSWERS/MONK(93)2