PNC TN9410 95-354

MCDAT

A Preprocessor to Generate Data for Monte Carlo Codes

December 1995

OARAI ENGINEERING CENTER

POWER REACTOR AND NUCLEAR FUEL DEVELOPMENT CORPORATION

PNC TN9410 95-354
December, 1995

ABSTRACT

J K Fletcher*

A preprocessor for the Monte Carlo Codes VIM and MVP is described in which geometric and compositional data are used to generate the complex and lengthy files of input information required to specify the models found in fast reactor studies. Further, in using Monte Carlo methods users must have knowledge of combinatorial geometry and the Boolean algebra techniques necessary to generate complicated structures from basic shapes. Most reactor physicists are not familiar with such procedures and even if they are, to set up or modify models tends to be tedious and error prone work.
As stated the preprocessor requires only geometric dimensions and material compositions to be input and then performs all the logical operations necessary to produce input files for VIM or MVP. As a consequence models can be easily modified to increase detail or introduce new regions for editing purposes. Also use has been made of free format data reading routines so that keywords can be utilized to clarify and provide flexibility in data preparation thus relieving users from carefully observing prescribed layouts.
Because only geometric and composition information is required by the preprocessor, data files become very compact compared to the actual Monte Carlo input. For example around 200 lines of preprocessor data are needed for the MASURCA assembly specification but a file of 3000 lines is generated as the VIM input. It should be stressed that the data for VIM and MVP differs only in the keyword which specifies the Monte Carlo code input to be created.

* Core Physics Research Section, Advanced Technology Division, O-arai Engineering Center, PNC ,Japan.

PNC TN9410 95-354
1 9 9 5 年 1 2 月

MCDAT
モンテカルロコード用入力データ作成プリプロセッサー

J. K. フレッチャー*

要 旨

高速炉の炉心解析に用いているモンテカルロコードVIMおよびMVPの入力データを生成するための前処理コードMCDATを開発した。モンテカルロ解析において、炉心の幾何学形状及び組成情報を詳細なモデルで入力するためには、非常に複雑でまた大量の入力データを作成する必要があり、また、コードが準備している基本形状ツールから実際の複雑な炉心形状用の入力を作成するためには、組合せ幾何学とブール代数の知識を必要とする。多くの炉心解析者はこのような手順には慣れておらず、また知識があったにしても、その入力作成作業には時間がかかり、誤りも起こりやすかった。

今回開発した前処理コードMCDATは、炉心の幾何学形状データと材料組成データのみを入力として、VIMコードとMVPコードの入力データ生成に必要な論理操作を全て自動的に実行するものである。このため、解析モデルの詳細化や、編集のための新しい領域の定義などを非常に簡単に行うことができる。また、MCDATの入力はフリーフォーマットであるため、解析者が入力フォーマットの誤りなどに神経を使う必要が無くなった。

MCDATコードに必要な入力情報は、幾何学形状と材料組成のみであるため、実際にモンテカルロコードに入力するのと比べると、そのデータ量は非常に少ない。例えば、MCDATに200行のデータを入力ことにより、VIMコードを用いて仏国のMASURCA臨界実験炉心の解析を行うために必要な3,000行の入力ファイルを自動的に生成できる。

なお、従来の炉心解析で使用されてきたVIMコード用と最近公開されたMVPコードへの入力データ作成をキーワードひとつで切り替えられるようにしたことも、今回のMCDATコードの特徴となっている。

* 大洗工学センター 基盤技術開発部 炉心技術開発室　国際特別研究員

CONTENTS

Chapter 1 INTRODUCTION

For many years computer power appeared insufficient to enable Monte Carlo codes to be used generally for reactor core physics studies even though for some calculations their capabilities were obviously superior to the standard deterministic approaches. Specific instances are:

a) the treatment of voids,

b) neutron streaming or anisotropic diffusion

and

c) modeling of heterogeneous subassemblies.

In recent times the appearance of work stations has made running time less important since charges are not made on calculation duration, but work using VIM[1] has demonstrated that input for a Monte Carlo code can be extremely lengthy and time consuming to prepare as well as difficult to free of errors. However because the structures being studied are generally regular geometrically e g an array of pins in hexagonal or rectangular subassemblies, the generation procedure seems an obvious candidate for implementation in a computer code.

Preliminary work showed this to be the case and a preprocessor has been developed for rectangular and hexagonal lattices using a set of data reading routines, described in Appendix 1, which do not require any formatting of input as well as allowing mixing of numeric and alphanumeric information. The rectangular option is directed toward investigations of MASURCA[2] experiments and contains options to accommodate the somewhat specialized structure that can arise with experimental zero power reactors.

In the following sections some details of the preprocessor are given and possible future developments outlined. Specification of input is given in Appendix 1 with examples in Appendices 4 and 5

Following extensive validation of the VIM option a capability to generate MVP[3] datafiles was incorporated in the preprocessor. MVP has a more sophisticated data specification available than VIM, for example a double lattice feature and this has been incorporated in the preprocessor as an option.

Chapter 2 DETAILS OF THE PREPROCESSOR

The coding is all in FORTRAN and has been tested and developed on the FACOM computer at OEC. A modular approach has been adopted so that after reading in the data, where necessary, different routines have been developed for hexagonal or rectangular geometry.

As with most Monte Carlo codes, VIM makes use of combinatorial geometry techniques to specify complicated structures. Thus for a hexagonal subassembly containing pins in a regular array the user must specify the location of each pin, its radius, the radius of the can and length of the pin in that particular axial region and this leads to a large amount of data for say 331 pins. As indicated in the previous chapter for MVP a double lattice may reduce input but considerable familiarity with the code is needed to use this approach with confidence. However the location of each pin is not really needed. As the array of pins is regular only the pitch is required to generate the position in the plane. This with the pin and subassembly dimensions is used by the preprocessor to prepare the VIM and MVP input. which includes the Boolean logic used to put primitive shapes together to form complicated structures Also many VIM specific details are evaluated automatically, including non-trivial parameters such as the number of code zones in a complex body, maximum number of exit zones and similar information[1]. MVP does not require many such parameters but those needed are similarly derived automatically. The only other information supplied is material compositions Thus the user is relieved of much tedious arithmetic, keying in of data and understanding of the VIM or MVP codes.

The preprocessor is held on the FACOM under the identifier POC0F16.VIM.VIMMD and consists of around 1800 lines of code including the free format input routines.

Chapter 3 PREPROCESSOR DATA INPUT

Data input is specified in Appendix 2 and as noted previously dealt with in the preprocessor by using free format reading routines described in Appendix 1. Relying on these enables users to lay out input in any convenient form and use comments to add clarity. Also much use is made of keywords and for example in composition data U235 may be used rather than an obscure library integer. Further, numerical data will not assume incorrect values because specified formats have not been observed.

The features described above can clearly be seen in the Examples of Appendices 4 and 5. The latter is a full specification of MASURCA in the ZONA2A ref configuration[2] and some idea of the labor saved by the preprocessor can be gained from the fact that the 200 lines or so of input generate around 3000 lines of VIM datafile.

Chapter 4 FUTURE DEVELOPMENTS

As indicated earlier the input to the preprocessor is dimensions and composition data nonspecific to VIM or MVP so that input for any other Monte Carlo code could be generated. Naturally other codes may have very different input requirements. However a preprocessor frees the user from understanding possibly rather complicated techniques and formats and provides the option of moving freely amongst codes for which data files can be generated.

Chapter 5 CONCLUSIONS

A small preprocessor code has been developed which enables users, with no specialized knowledge, to prepare the complex datafiles required by VIM or MVP from a comparatively small quantity of geometric and material composition data. Such a tool greatly eases use of Monte Carlo codes and provides the facility to generate new or modify existing models accurately and efficiently.

Acknowledgment

The development of the preprocessor was greatly aided by the unstinting help of Mr W Sato(NESI) particularly in modifying VIM to accept large problems and providing advice on use of the FACOM computer

REFERENCES

1. R N Blomquist et al   VIM. A Continuous Energy Monte Carlo Code at ANL.   ORNL/RSIC(1980)

2. P J Finck et al   CAPRA Physics and the CIRANO Experimental ProgramWorkshop on Advanced Fuel Cycles   PSI.(September 1995)

3. T Mori ,M Nakagawa   JAERI-Data/Code 94-007(1994)

APPENDIX 1

Using the Free Format Data Reading Routines

The routines are available under identifier POC0F16.PNC.CODE(DECIN) on the FACOM computer

Any line beginning with an asterisk (*) is regarded as a comment card and is ignored by the reading routines

FUNCTION INDEC(N)

Read an integer
N=0, terminate with a blank
N=1, terminate with any non-numeric character

FUNCTION FLDEC(N)

Read a floating point number in any FORTRAN format

N=0, terminate with a blank
N=1, terminate with any non-numeric character
(remember that E or D will be considered part of a floating point number)

SUBROUTINE WORDEC(WORD,N,M)

Read a word of up to N characters into WORD

M=0, terminate with any numeric or non-alphabetic character
M=1, terminate with a blank thus allowing alphanumeric words

SUBROUTINE JNDEX(J,WORD,'STR1*STR2*STR3* *STRn&',N)

Compare WORD with STR1 , STR2 etc, if a match for STRj is found J is set equal to j, For no match J=n+1 and the comparison is carried out for WORD being up to N alphanumeric characters long.

FUNCTION ITDEC(N)

Read one character of any type, N is currently not used and may have any value.

FUNCTION NEWDEC(N)

If next character to be read is numeric	NEWDEC>0
If next character is a letter	NEWDEC<0
If next character is special(e.g. ( ) * & etc)	NEWDEC=0

N=0 , + or - are assumed to be part of a number
N=1 ,+ or - treated as special

SUBROUTINE IARRAY(M,N)

Read into M an integer array in the general form

I1(I2*M1 I3(I4*M2 I5*M3) I6*M4 M5)

The asterisk means repeat e.g. . I2 of M1,
( ) means repeat the contents e. g. the whole of the contents of the brackets after I1 will be input I1 times

N is returned as the number read in on encountering a non-numeric character.

There can be up to 20 levels of brackets

SUBROUTINE FARRAY(A,N)

As IARRAY but for floating point numbers

APPENDIX 2

INPUT TO THE PREPROCESSOR

Code words put in data are given in **bold** type

TITLE

Two lines of up to 72 characters each describing the problem. For VIM the first 12 must be the random number generator seed in hexadecimal.

BATCHES n

There are to be n batches

HISTORIES m

Each batch will have m histories

GEOMETRY k

k=-1 rectangular lattice ,-2 hexagonal

TIME t

the problem will stop after t seconds

DELTA δ

δ will be used as specified in VIM[1] to ensure region boundaries are crossed rather than the default value 10^{-5}

MEDIA

M 1 **sub1** a_1 **sub2** a_2 . . .

M 2 **sub3** b_1 **sub4** b_2

.

,

where **subn** is taken from the list in Appendix 3 and a_i and b_i are atom/cc $x10^{24}$

NTYPES n

There are n different sub assembly structures.

Different input is now required depending on the geometry.

<u>Hexagonal geometry</u>

$r_1 \quad h_1 \quad a_1 \quad p_1 \quad cr_1 \quad fr_1 \quad s_1 \quad w_1 \quad \alpha_1^1 \quad \alpha_1^2 \ldots \alpha_1^{h_1}$

$r_2 \quad h_2 \quad a_2 \quad p_2 \quad cr_2 \quad fr_2 \quad s_2 \quad w_2 \quad \alpha_2^1 \quad \alpha_2^2 \ldots \alpha_2^{h_2}$

.

.

$r_n \quad h_n \quad a_n \quad p_n \quad cr_n \quad fr_n \quad s_n \quad w_n \quad \alpha_n^1 \quad \alpha_n^2 \ldots \alpha_n^{h_n}$

where $r_i, h_i, a_i, p_i, cr_i, fr_i, t_i$, and w_i are the number of rings of pins, number of axial regions, distance across flats ,pin pitch, can radius, fuel radius, half the inter-subassembly gap and wrapper thickness respectively for subassembly type i having axial lengths $\alpha_i^j, j = 1,2,..h_i$.

Then follows

$$
\begin{array}{lllllll}
i_1 & sm_1^{i_1} & wm_1^{i_1} & cm_1^{i_1} & fm_{11}^{i_1} & fm_{12}^{i_1} & \ldots fm_{1h_1}^{i_1} \\
 & sm_2^{i_1} & wm_2^{i_1} & cm_2^{i_1} & fm_{21}^{i_1} & fm_{22}^{i_1} & \ldots fm_{2h_1}^{i_1} \\
 & . & & & & & \\
 & . & & & & & \\
 & sm_{i_1}^{i_1} & wm_{i_1}^{i_1} & cm_{i_1}^{i_1} & fm_{i_1 1}^{i_1} & fm_{i_1 2}^{i_1} & \ldots fm_{i_1 h_1}^{i_1} \\
i_2 & sm_1^{i_2} & wm_1^{i_2} & cm_1^{i_2} & fm_{11}^{i_2} & fm_{12}^{i_2} & \ldots fm_{1h_2}^{i_2} \\
 & sm_2^{i_2} & wm_2^{i_2} & cm_2^{i_2} & fm_{21}^{i_2} & fm_{22}^{i_2} & \ldots fm_{2h_2}^{i_2} \\
 & . & & & & & \\
 & . & & & & & \\
 & sm_{i_2}^{i_2} & wm_{i_2}^{i_2} & cm_{i_2}^{i_2} & fm_{i_2 1}^{i_2} & fm_{i_2 2}^{i_2} & \ldots fm_{i_2 h_1}^{i_2} \\
 & . & & & & & \\
 & . & & & & & \\
i_n & sm_1^{i_n} & wm_1^{i_n} & cm_1^{i_n} & fm_{11}^{i_n} & fm_{12}^{i_n} & \ldots fm_{1h_1}^{i_n} \\
 & . & & & & & \\
 & . & & & & & \\
 & sm_{i_n}^{i_n} & wm_{i_n}^{i_n} & cm_{i_n}^{i_n} & fm_{i_n 1}^{i_n} & fm_{i_n 2}^{i_n} & \ldots fm_{i_n h_n}^{i_n}
\end{array}
$$

where i_j is the number of subassemblies having structure i and containing in set k, coolant material $sm_k^{i_j}$, wrapper $wm_k^{i_j}$, can $cm_k^{i_j}$ and pin contents $fm_{lk}^{i_j}$ in axial slice l.

An example is given in Appendix 4

Rectangular Geometry

Similar to the hexagonal case but with some extra options included to deal with configurations specific to MASURCA. First geometric details are specified as follows

$$l \quad g_1 \quad w_1 \quad d_1 \quad h_1 \quad np_1^1 \quad np_2^1 \quad ...np_{h_1}^1\ END \quad \alpha_1^1 \quad \alpha_2^1 \quad ...\alpha_{h_1}^1\ END$$

$$l \quad g_2 \quad w_2 \quad d_2 \quad h_2 \quad np_1^2 \quad np_2^2 \quad ...np_{h_2}^2\ END \quad \alpha_1^2 \quad \alpha_2^2 \quad ...\alpha_{h_2}^2\ END$$

.

.

.

$$l \quad g_n \quad w_n \quad d_n \quad h_n \quad np_1^n \quad np_2^n \quad ...np_{h_n}^n\ END \quad \alpha_1^n \quad \alpha_2^n \quad ...\alpha_{h_n}^n\ END$$

where l is the lattice pitch(the same for all types) withg_i ,w_i,h_i respectively half the gap between subassemblies , subassembly wall thickness and number of axial regions for type i having length α_j^i in section j and this slice containing an $np_j^i \times np_j^i$ array of pin locations. d_j is the pin diameter, assumed constant, for type j. If an np_j^i is negative the rod is taken to be a square bar filling the region i.e the gap is omitted as in the structure of breeder and Na/steel regions.

Next comes

$$i_1$$

indicating that there are i_1 different subassemblies based on type 1 structure. Then for $j = 1,2...i_1$, submit together the next two blocks of data,

$$nm_{j1}^1 \quad nm_{j2}^1 \quad ...nm_{jh_1}^1\ END$$

where the nm_{jk} parameters specify how many extra materials are in pins in axial slice k, so that the normal value will be zero. This feature is needed to deal with peripheral fuel. Follow immediately with

$$mg_{j1}^1 \quad mw_{j1}^1 \quad ms_{j1}^1 \quad mf_{j1}^1$$

$$mg_{j2}^1 \quad mw_{j2}^1 \quad ms_{j2}^1 \quad mf_{j2}^1$$

.

.

$$mg_{jh_1}^1 \quad mw_{jh_1}^1 \quad ms_{jh_1}^1 \quad mf_{jh_1}^1$$

where mg^1_{jk} mw^1_{jk} ms^1_{jk} and mf^1_{jk} are the gap, wrapper, interior and fuel materials in axial slice k for subtype j. If there are no pins in an axial section i.e. np^1_m is zero for some m, the fuel material is omitted. A negative sign before mg^1_{jk} will give edit output for axial region k.

For any nm^1_{jk} not equal to zero, input after mf^1_{jk} in the above

$$m^1_{jkl} \quad nl^1_{jkl} \quad lp^1_{jkl}(1) \quad lp^1_{jkl}(2) \quad \ldots lp^1_{jkl}(nl^1_{jkl})$$

for $l=1, nm^1_{jk}$, where m^1_{jkl} is the $l'th$ different pin material in axial region k of subtype j for subassemblies of the first type. Pin location, $lp^i_{jkl}(\ )$, numbers are shown in Figure 1

The above is repeated for the n subassembly structures with the superscript becoming 2 then 3 and so on up to n. Although apparently complicated to explain and specify the procedure is reasonably straightforward as can be seen from Appendix 5.

For both geometries

PLAN n_x n_y

where n_x is the number of subassemblies in the x direction and n_y that for y. Subassemblies are then specified in the hierarchy x then y using the condensing facility of asterisks and brackets as illustrated in the Appendices 4 and 5. For both VIM and MVP using the hexagonal geometry some care is needed as the y axis makes 120° with the x direction so that half a subassembly must be added with each row[1,3].

FRATES sub1 sub2

CRATES sub3 sub4

Fission and capture rates will be calculated for **subn** taken from the list of Appendix 2

ENERGIES e_1 e_2 $\ldots e_{NG}$

giving lower energy bounds for the VIM or MVP editing routines

VIM tells the preprocessor to create a VIM input file

DLAT double lattice for MVP (must precede **MVP** below)

MVP tells the preprocessor to create an MVP input file

APPENDIX 3

Isotopes Available in the Preprocessor

PU240	CR	HE
PU241	NI	LI6
U235	FE	LI7
U238	AL27	N
PU239	NA	AU
U234	O	ZR
TH232	C	MG
U236	MO	SM
PU238	MN	EU151
U233	BE	EU153
PU242	B9	HE3
NP237	B10	H2
AM241	TA	H3
AM243	CU	CA
PA233	H	
CM244	PB	
UO2	TI	
FIS	SI	

LIBRARIES AVAILABLE

VIM	JENDL 3.1
MVP	JENDL 3.1 and JENDL 3.2

APPENDIX 4

Sample Input for the Hexagonal Option

```
253577253677     SMALL FBR
SECOND TITLE CARD
BATCHES 200 HISTORIES 5000 GROUPS 27 TIME 7000 DELTA 1E-4
FRATES PU240 PU241 U238  CRATES U238 GEOMETRY -2
NTYPES 1 10 5
20. 0.9  0.85 0.8  .2 .4
60.96 20.32 60.96 20.32 60.96
     2
2 1 1 1 3 4 3 1  2 1 1 1 3 3 3 1
     EDIT   1   9     1 9     60.96 162.56 BOUNDARY 0 0 0
PLAN 9    9
5*2 4*0   6*2 3*0   2*2 3*1 2*2 2*0    2*2 4*1 2*2 0
2*2 5*1 2*2
0 2*2 4*1 2*2    2*0 2*2 3*1 2*2  3*0 6*2  4*0 5*2
MEDIA
M 1 CR    .01257 NI     .005749 FE     0.04474 0      .00025
   AL27   0.00005 MN     0.00101
M 2 NA    0.02226
M 3 U238 .0242   0      .0505
M 4 PU240 .0013PU241 .000226 U238 0.01835  PU239 .005335
   0      0.0505
M 5 0     0.000001 END
ENERGIES
6.053E6 3.5768E6 2.2313E6 1.3534E6 8.2085E5 4.9787E5 3.0197E5
1.8136E5 1.11095E5 6.7379E4 4.0868E4 3.788E4 1.5034E4 9.1188E3 5.5308E
3.3546E3
2.0347E3 1.2011E3 7.4852E2 4.5400E2 2.7536E2 1.6702E2 1.013E2 6.1442E1
3.7267E1 2.2603E1 5.0435
VIM
STOP
```

APPENDIX 5

Sample Input for the Rectangular Option

```
253577253677      MASURCA HETEROGENEOUS
SECOND TITLE CARD
BATCHES 200 HISTORIES 5000 GROUPS 27 TIME 7000 DELTA 1E-4
FRATES PU240 PU241 U238  CRATES U238
NTYPES 2
* THERE ARE 2 TYPES OF SUBASSEMBLIES
* TYPE 1
10.6  .05 .17  1.2466 7 0 8 0 8 0 8 0 END
60.96 20.32 1.169  60.391 1. 20.32  END
* TYPE 2
10.6  .05 .17 0  7 0 0 0 0 0 0 0 END
60.96 20.32 0.8 60.96 0.8 20.32  END
* TYPE 1 HAS 20 VARIANTS TO MODEL PERIPHERAL FUEL AND BREEDER
     30
* MAIN FUEL DR 1
 7*0 END  5 1 1   5 1 2 3  5 1 1  5 1 2 4  5 1 1  5 1 2 3  5 1 1
* BREEDER   DR 2
 7*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 3  5 1 1 5 1 2 3   5 1 1
* INSTRUMENT CHANNEL DR 3
     3*0  1 3*0  END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 1 25
5 1 1  5 1 2 3  5 1 1
* PERIPHERAL FUEL TAKEN ANTICLOCKWISE FROM BOTTOM
* NO 1  DR 4
     3*0 2 3*0  END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
3 8 1 2 3 4 5 6 7 8   6 10 9 10 11 12 13 14 15 16 20 24
5 1 1 5 1 2 3 5 1 1
*NO 2   DR 5
     3*0 2 3*0  END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
6 8 17 18 19 20 21 22 23 24
4 8 25 26 27 28 29 30 31 32     5 1 1  5 1 2 3  5 1 1
* NO 3  DR 6
     3*0 2 3*0  END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
3 4  3 4 7 8    6 4 2 6 20 24
5 1 1  5 1 2 3  5 1 1
* NO 4   DR 7
     3*0 2 3*0  END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
6 2 17 21    4  2 25 29
5 1 1  5 1 2 3  5 1 1
* NO 5   DR 8
     3*0 2 3*0  END 5 1 1  5 1 2 3  5 1 1 5 1 2 4
6 4  2 6 20 24  3 8 3 4 7 8 11 12 15 16
5 1 1 5 1 2 3 5 1 1
* NO 6 DR 9
     3*0 1 3*0  END 5 1 1 5 1 2 3   5 1 1  5 1 2 3
6 4 17 21 25 29
5 1 1  5 1 2 3  5 1 1
```

```
* NO 7 DR 10
     3*0 2 3*0  END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
4 8 1 5 9 13 17 21 25 29  6 8  2 6 10 14 18 22 26 30
5 1 1 5 1 2 3  5 1 1
*  NO 8 DR 11
    3*0 2 3*0 END 5 1 1 5 1 2 3 5 1 1 5 1 2 4
6 10 2 3 6 7 11 15 19 23 27 31   3 8 4 8 12 16 20 24 28 32
5 1 1  5 1 2 3  5 1 1
* NO 9  DR 12
    3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 10 3 7 11 15 19 23 26 27 30 31   3 8 4 8 12 16 20 24 28 32
5 1 1  5 1 2 3  5 1 1
* REPEAT 7  DR 10
* NO 10  DR 13
   3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 3
   4 2 1 5   6 2  9 13
5 1 1 5 1 2 3  5 1 1
* NO 11  DR 14
3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 4 12 16 26 30  3 8  19 20 23 24 27 28 31 32
5 1 1  5 1 2 3  5 1 1
* NO 12  DR 15
3*0  2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
4 2 1 5  6 4 2 6 9 13
5 1 1  5 1 2 3  5 1 1
* NO 13 DR 16
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 8 12 16 20 24 26 27 30 31   3 2 28 32
5 1 1  5 1 2 3  5 1 1
* NO 14  DR 17
3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 3
4 8 1 2 3 4 5 6 7 8  6 8 9 10 11 12 13 14 15 16
5 1 1  5 1 2 3  5 1 1
* NO 15  DR 18
3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 10 12 16 17 18 19 20 21 22 23 24  3 8 25 26 27 28 29 30 31 32
5 1 1  5 1 2 3  5 1 1
* NO 16  DR 19
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 10 9 13 17 18 19 20 21 22 23 24  3 8 25 26 27 28 29 30 31 32
5 1 1   5 1 2 3  5 1 1
* REPEAT 14 DR 17
*NO 17   DR 20
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 8 9 13 17 21 26 27 30 31  3 2 25 29
5 1 1  5 1 2 3  5 1 1
*NO 18  DR 21
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
4 2 4 8   6 4 3 7 12 16
5 1 1  5 1 2 3  5 1 1
* NO 19  DR 22
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 4 9 13 27 31 3 8 17 18 20 21 25 26 29 30
5 1 1  5 1 2 3  5 1 1
```

```
* NO 20    DR 23
3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 3
4 8 4 8 12 16 20 24 28 32  6 8 3 7 11 15 19 23 27 31
5 1 1  5 1 2 3  5 1 1
* NO 21  DR 24
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 10 2 6 10 14 18 22 26 27 30 31  3 8 1 5 9 13 17 21 25 29
5 1 1  5 1 2 3  5 1 1
*NO 22   DR 25
3*0 2 3*0 END 5 1 1  5 1 2 3   5 1 1  5 1 2 4
6 10 2 3 6 7 10 14 18 22 26 30 3 8 1 5 9 13 17 21 25 29
5 1 1  5 1 2 3  5 1 1
*REPEAT 20   DR 23
* NO 23  DR 26
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
6 2 20 24  4 2 28 32
5 1 1  5 1 2 3  5 1 1
* NO 24  DR 27
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 4 3 7 17 21   3 8 1 2 5 6 9 10 13 14
5 1 1  5 1 2 3  5 1 1
* NO 25  DR 28
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 3
6 4 20 24 27 31  4 2 28 32
5 1 1  5 1 2 3  5 1 1
* NO 26   DR 29
3*0 2 3*0 END  5 1 1  5 1 2 3  5 1 1  5 1 2 4
6 4 3 7 17 21 3 8 1 2 5 6 9 10 13 14
5 1 1  5 1 2 3 5 1 1
*REPEAT NO 2 DR 5
* NO 27  DR 30
3*0 2 3*0 END 5 1 1  5 1 2 3  5 1 1  5 1 2 4
3 8 1 2 3 4 5 6 7 8  6 10 9 10 11 12 13 14 15 16 17 21
5 1 1  5 1 2 3  5 1 1
*  REF   DR 31
2     7*0 END  5 1 1  5 1 1  5 1 1  5 1 1  5 1 1  5 1 1  5 1 1
* CONTROL ROD DR 32
7*0 END  5 1 5  5 1 5  5 1 5  5 1 5  5 1 5  5 1 5  5 1 5
     EDIT   4   18    4 18    60.96 162.56 BOUNDARY 0 0 0
```

```
PLAN 22 22
3(22*31)
7*31 8*2 7*31
6*31 10*2 6*31
5*31 12*2 5*31
4*31 5*2 5 30 4 5 5*2 4*31
3*31 4*2 28 29 4*1 6 7 4*2 3*31
3*31 3*2 26 27 6*1 8 9 3*2 3*31
3*31 3*2 23 8*1 10 3*2 3*31
3*31 3*2 25 4*1 32 3*1 11 3*2 3*31
3*31 3*2 24 8*1 12 3*2 3*31
3*31 3*2 23 8*1 10 3*2 3*31
3*31 4*2 22 6*1 14 13 3*2 3*31
3*31 4*2 21 20 4*1 16 15 4*2 3*31
4*31 5*2 17 19 18 17 5*2 4*31
5*31 12*2 5*31
6*31 10*2 6*31
7*31 8*2 7*31
3(22*31)
MEDIA
M 1 CR     .01257 NI      .005749 FE      0.04474 O       .00025
   MO      0.00005 MN      0.00101
M 2 NA     0.02226
M 3 U238 .0242   O       .0505
M 4 PU240 .0013PU241 .000226 U238 0.01835  PU239 .005335
   O       0.0505
M 5 O      0.000001
M 6 PU240 .000508   PU241 3.25E-5  U238 .0194 PU239 .005873
 O .0511       END
ENERGIES
6.053E6 3.5768E6 2.2313E6 1.3534E6 8.2085E5 4.9787E5 3.0197E5
1.8136E5 1.11095E5 6.7379E4 4.0868E4 3.788E4 1.5034E4 9.1188E3 5.5308E
3.3546E3
2.0347E3 1.2011E3 7.4852E2 4.5400E2 2.7536E2 1.6702E2 1.013E2 6.1442E1
3.7267E1 2.2603E1 5.0435
VIM
STOP
```

Figure 1 Pin Locations for Rectangular Geometry