OAK RIDGE NATIONAL LABORATORY LIBRARY
3 4456 0003704 7

ORNL/TM-9767

ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

The Integrated TIGER Series of Electron/Photon Monte Carlo Transport Codes: A User's Guide for Use on IBM Mainframes

Bernadette L. Kirk

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A04; Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-9767

Engineering Physics and Mathematics Division

THE INTEGRATED TIGER SERIES OF ELECTRON/PHOTON MONTE CARLO TRANSPORT CODES: A USER'S GUIDE FOR USE ON IBM MAINFRAMES

Bernadette L. Kirk

Radiation Shielding Information Center

Research partially supported by
Defense Nuclear Agency
under Interagency Agreement No. 40-46-64

Date published: December 1985

NOTICE: This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract No. DE-AC05-84OR21400

OAK RIDGE NATIONAL LABORATORY LIBRARY
3 4456 0003704 7

CONTENTS

ABSTRACT 1

INTRODUCTION 1

CONTENTS OF CCC-467 CODE PACKAGE 2

INSTALLATION AND EXECUTION OF ITS 4

I. Preparatory steps in running the 'ITS' code 4

II. Running the 'TIGER' Code 9

III. Running the 'TIGERP' Code 12

IV. Running the 'CYLTRAN' Code 15

V. Running the 'CYLTRANP' Code 18

VI. Running the 'CYLTRANM' Code 21

VII. Running the 'ACCEPT' Code 24

VIII. Running the 'ACCEPTP' Code 28

IX. Running the 'ACCEPTM' Code 31

REFERENCES 35

APPENDIX A: JSTIME 37

APPENDIX B: EXPLANATION OF DATA SETS USED IN THE JCL 39

APPENDIX C: IBM SPECIFIC PROBLEMS 41

THE INTEGRATED TIGER SERIES OF ELECTRON/PHOTON MONTE CARLO TRANSPORT CODES: A USER'S GUIDE FOR USE ON IBM MAINFRAMES

Bernadette L. Kirk

Abstract

The ITS (Integrated Tiger Series) Monte Carlo code package developed at Sandia National Laboratories and distributed as CCC-467/ITS by the Radiation Shielding Information Center (RSIC) at Oak Ridge National Laboratory (ORNL) consists of eight codes—the standard codes, TIGER, CYLTRAN, ACCEPT; the P-codes, TIGERP, CYLTRANP, ACCEPTP; and the M-codes ACCEPTM, CYLTRANM. The codes have been adapted to run on the IBM 3081, VAX 11/780, CDC-7600, and Cray 1 with the use of the update emulator UPEML (see reference 1). This manual should serve as a guide to a user running the codes on IBM computers having 370 architecture. The cases listed were tested on the IBM 3033, under the MVS operating system using the VS Fortran Level 1.3.1 compiler.

INTRODUCTION

Intergrated Tiger Series (ITS) is a user-friendly Monte Carlo code system dealing with electron/photon radiation transport problems. J. A. Halbleib and T. A. Melhorn of Sandia National Laboratories (1) developed the ITS code system. Depending on the type of calculation needed, the user has the choice of running eight codes.

TIGER is strictly a one-dimensional code and is the simplest code to use since it involves very basic transport problems.

CYLTRAN is a three-dimensional code, applicable in cases where the radiation source is a cylindrical beam. CYLTRAN requires the material geometry to be cylincrically symmetric.

ACCEPT is a general three-dimensional transport code useful for multi-material geometries that can be described in combinatorial forms. In the combinatorial geometry method, problem zones are built from primitive bodies.

TIGER, CYLTRAN, and ACCEPT were designed for primary source energies from a few terms of MeV to 1.0 and 10.0 keV for electrons and photons, respectively. In these codes, fluorescence and Auger processes are only allowed for the k-shell of the highest atomic number in a given material.

The P-codes in the ITS series are TIGERP, CYLTRANP, and ACCEPTP. These P-codes describe ionization and relaxation of all K, L1, L2, L3, M (average) and N (average) shells having binding energies greater than 1.0 keV for elements with atomic number Z=1 to Z=100.

In the M-codes, ACCEPTM and CYLTRANM, coupled electron/photon collision transport is combined with transport in macroscopic electric and magnetic fields of arbitrary spatial dependence. In applications where transport is macroscopic electromagnetic fields cannot be neglected, ACCEPTM and CYLTRANM prove valuable codes.

More detailed explanations of the differences in the eight ITS codes can be found in the report by Halbleib and Melhorn (2).

In the next sections are sample JCL (job control language) listings pertaining to the proper installation and running of the ITS code package. Every effort was made to list step-by-step instructions, as IBM JCL is complicated to follow. In order to use this guide the user is required to be familiar with IBM JCL.

It is necessary to have direct access devices to store the cross-section data. The disk packs used for the testing are the IBM 3330s. Note that for this disk type one track is equivalent to 13030 bytes, so the track allocation mentioned in the following JCL listings must be interpreted as such.

It should be noted that the JCL statements listed on the following pages are not foolproof. System dependencies must be accounted for. This manual should serve as a guide and not a bible in running ITS on IBM mainframes.

The user must be familiar with the ITS and UPEML reports (1,2). The steps in this manual should complement the instructions in those reports. The user may choose to combine several job steps into one.

CONTENTS OF CCC-467 CODE PACKAGE

(RSIC TAPE LIST)

The following is a description of the files contained in the CCC-467 code package. The numbers indicate the position of the files on the tape.

File Number and Description	NUMBER OF RECORDS WITH RECFM = FB, LRECL=136, BLKSIZE=4760
1. ITS (to be processed by UPEML)	19651
2. XGEN (to be processed by UPEML)	4448
3. XDATA (cross section data)	10039
4. UPEML UPDATE EMULATOR PROGRAM (FORTRAN 77 Source)	2563
5. Input data to XGEN to generate cross sections for the standard Monte Carlo codes (TIGER , ACCEPT , CYLTRAN)	19
6. Output from running XGEN to generate cross sections for the standard codes	273
7. Input data to XGEN to generate cross sections for the P-codes Monte Carlo codes (TIGERP, CYLTRANP, ACCEPTP)	20
8. Output from running XGEN to generate cross sections for the P-codes.	279
9. Input for TIGER	59
10. Output from TIGER	514
11. Input for TIGERP	62
12. Output from TIGERP	511
13. Input for TIGERP (to generate pseudo-pulse-height distribution)	70
14. Output from TIGERP (generated pseudo-pulse height distribution)	526
15. Input for CYLTRAN	60
16. Output from CYLTRAN	570
17. Input for CYLTRANP	63
18. Output from CYLTRANP	567
19. Input for CYLTRANM (small gap problem)	69
20. Output from CYLTRANM (small gap problem)	573
21. Input for CYLTRANM (large gap problem)	75
22. Output from CYLTRANM (large gap problem)	574
23. Input for ACCEPT	81
24. Output from ACCEPT	515
25. Input for ACCEPT (to generate pseudo-pulse-height distribution)	90
26. Output from ACCEPT (generated pseudo-pulse-height distribution)	525
27. Input for ACCEPTP	84
28. Output from ACCEPTP	512
29. Input for ACCEPTM	89
30. Output from ACCEPTM	517
TOTAL	43998

INSTALLATION AND EXECUTION OF ITS

I. Preparatory steps in running the 'ITS' codes.

A. Copy files 1-30 (ITS, XGEN, XDATA and UPEML) to direct access. Suggested disk data set characteristics are listed. The number of tracks is a liberal estimate.

File	Tracks	RECFM	LRECL	BLKSIZE
ITS	220	FB	80	6400
XGEN	50	FB	80	6400
UPEML	40	FB	80	6400
XDATA	120	FB	136	6120
files 5-30	10 or less	FB	136	6120

NOTE: 1 track = 13030 bytes (IBM 3330 disk).

The following are common IBM direct access storage disks (reference 3).

Disk type	Number of bytes per track	Number of tracks per cylinder
3330	13030	19
2305	14660	24
3340	8368	12
3350	19060	30
3375	35616	12
3380	47476	15

In general, to approximate the number of tracks needed per data set, use the formula

$$\frac{\text{Number of records} * \text{Record length}}{\text{Number of bytes per track}}$$

For example, file 30 has 517 records written at LRECL=136. If your disk drive is a 3330 type, then (517 * 136)/13030 = 5.4. Since tracks are integral numbers, this amounts to six tracks. You should allow for system dependencies and allocate 10 tracks.

Depending on the directives given, the following table lists the number of lines generated by UPEML for each of the codes. This information is obtained from the correction runs (see section A of each of Sections II-IX in the next few pages).

* Code	Number of lines	*
TIGER	17080	
TIGERP	18810	
CYLTRAN	18349	
CYLTRANP	20115	
CYLTRANM	22832	
ACCEPT	20291	
ACCEPTP	22075	
ACCEPTM	24896	

B. Create a load module for the update processor UPEML.

```
//jobcard ; allocate about 10 seconds for this job.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSIN DD DSN=UPEML,DISP=SHR
/*
//LKED.SYSLMOD DD DSN=UPELOAD(UPEML),DISP=(NEW,CATLG),
// UNIT=      , VOL=SER=       , SPACE=(TRK,(100,50,10),RLSE),
// DCB=(RECFM=U,LRECL=80,BLKSIZE=3200)
/*
//
```

C. Execute UPEML to create a new program library for ITS.

```
//jobcard; allocate 30 seconds for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//*COMMENT THE NEWPL LIBRARY FOR 'ITS' IS 'ITS1'.
//GO.FT15F001 DD DSN=ITS1,DISP=(NEW,CATLG),UNIT=      ,VOL=SER=       ,
// SPACE=(1024,5000),DCB=(RECFM=F,BLKSIZE=1024)
//FT17F001 DD DSN=ITS,DISP=SHR
//FT16F001 DD DSN=ITSLST,DISP=(NEW,CATLG),
// SPACE=(TRK,(200,100),RLSE),VOL=SER=       ,UNIT=      ,
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
I,N,L
/*
//
```

NOTE: It is useful to have the data set ITSLST listed. It has 19659 lines of data and is very helpful in making correction runs.

D. Execute UPEML to create a new program library for XGEN.

```
//jobcard; allocate 10 seconds for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//GO.FT15F001 DD DSN=XGEN,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(50,50),RLSE),
// DCB=(RECFM=F,BLKSIZE=1024)
//FT17F001 DD DSN=XGEN,DISP=SHR
//FT16F001 DD DSN=XGENLST,DISP=(NEW,CATLG),UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(100,50),RLSE),DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
I,N,L
/*
```

E. Steps E.1 - E.4 are needed to generate cross sections for input to the standard codes -TIGER, ACCEPT, CYLTRAN; and the M-codes, ACCEPTM and CYLTRANM.

E.1. Execute UPEML to create a compile ready file for XGEN. This is known as a correction run.

```
//jobcard; allocate 15 seconds for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL    DD DSN=XGEN,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//FT16F001 DD DUMMY
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE XGEN.
//FT20F001 DD DSN=CXGEN,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P
*DEFINE,IBM
/*
//
```

E.2. Truncate the 90-character XGEN file to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY      EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
```

```
//SYSUT2    DD DSN=FXGEN,DISP=(NEW,CATLG),
// UNIT=       ,VOL=SER=        ,
// SPACE=(TRK,(200,100),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//SYSUT1    DD DSN=CXGEN,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN     DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

E.3. Create a load module for XGEN.

```
//jobcard; allocate 15 seconds for this job.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FXGEN,DISP=SHR
/*
//LKED.SYSLMOD DD DSN=XGENLOAD(XGEN),DISP=(NEW,CATLG),
//  SPACE=(TRK,(500,100,5),RLSE),DCB=(RECFM=U,LRECL=80,BLKSIZE=3200),
//  UNIT=       ,VOL=SER=
/*
//
```

E.4. Execute the program XGEN to create cross sections for the standard codes and the M-codes.

```
//jobcard; allocate 1 minute for this job.
//S1 EXEC PGM=XGEN,REGION=2048K
//STEPLIB DD DSN=XGENLOAD(XGEN),DISP=SHR
//*COMMENT CROSS SECTION FILE FOR INPUT TO 'ITS' IS WRITTEN TO UNIT 11.
//GO.FT11F001 DD DSN=XSECS1,DISP=(NEW,CATLG,DELETE),
// UNIT=       ,VOL=SER=        ,SPACE=(TRK,(10,5),RLSE),
// DCB=(RECFM=FB,LRECL=120,BLKSIZE=4800)
//FT07F001 DD DSN=&&TEMP,UNIT=SYSDA,SPACE=(3520,(200,100)),
// DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),DISP=(NEW,PASS)
//*COMMENT XDATA IS READ FROM UNIT 9.
//FT09F001 DD DSN=XDATA,DISP=SHR
//FT06F001 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD DSN=ISTAND,DISP=SHR
/*
//
```

F. Steps F.1 - F.4 are needed to generate cross sections for input to the P-codes :TIGERP, ACCEPTP, CYLTRANP.

F.1. Execute UPEML to create a compile-ready file for XGEN. This is known as a correction run.

```
//jobcard; allocate 15 seconds for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL    DD DSN=XGEN,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//FT16F001 DD DUMMY
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE XGEN.
//FT20F001 DD DSN=CXGEN2,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P
*DEFINE,IBM
*DEFINE,PCODES
/*
//
```

F.2. Truncate the 90-character XGEN file to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FXGEN2,DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(200,100),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//SYSUT1   DD DSN=CXGEN2,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

F.3. Create a load module for XGEN.

```
//jobcard; allocate 15 seconds for this job.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FXGEN2,DISP=SHR
/*
//*COMMENT Data set XGENLOAD was created in step E.3.
//LKED.SYSLMOD DD DSN=XGENLOAD(XGEN2),DISP=OLD
/*
//
```

F.4. Execute the program XGEN to create cross sections for the P-codes.

```
//jobcard; allocate 1 minute for this job.
//S1 EXEC PGM=XGEN,REGION=2048K
//STEPLIB DD DSN=XGENLOAD(XGEN),DISP=SHR
//*COMMENT CROSS SECTION FILE FOR INPUT TO ITS IS WRITTEN TO UNIT 11.
//GO.FT11F001 DD DSN=XSECS2,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(10,5),RLSE),
// DCB=(RECFM=FB,LRECL=120,BLKSIZE=4800)
//FT07F001 DD DSN=&&TEMP,UNIT=SYSDA,SPACE=(3520,(200,100)),
// DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),DISP=(NEW,PASS)
//*COMMENT XDATA IS READ FROM UNIT 9.
//FT09F001 DD DSN=XDATA,DISP=SHR
//FT06F001 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD DSN=IPCODES,DISP=SHR
/*
//
```

II. Running the 'TIGER' code.

A. Execute UPEML to create a compile-ready file for the TIGER code. This is a correction run. The input to FT05F001 is from the first few lines of file 9, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL     DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
```

```
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=TIGLST,UNIT=      ,VOL=SER=       ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE TIGER.
//FT20F001 DD DSN=CTIGER,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,TIGER
*IDENT,TEMQ
*D,PARAMS.5
      PARAMETER ( INMT=2,     NELEM=20,    INMAX=64,   NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  2,   IJMAX  =  2,
     $             IKPMAX =  2,   IJPMAX =  2,
     $             INZON  = 2)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,   IJFMP1 = IJFMXP+1,
     $             INLF   = 4,    INLFP  = 4)
*D,PARAMS.52
      PARAMETER (NESC  = 2,   ILZMAX = 4,    ILZMX1 = ILZMAX+1,
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
/*
//
```

B. Truncate the 90-character TIGER file to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FTIGER,DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(200,100),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//SYSUT1   DD DSN=CTIGER,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for TIGER.

```
//jobcard; allocate 1 minute for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
  Insert JSTIME source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FTIGER,DISP=SHR
/*
//LKED.SYSLIB DD
//            DD
//            DD
//            DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=TIGLOAD(TIGER),DISP=(NEW,CATLG),
//  SPACE=(TRK,(500,100,5),RLSE),DCB=(RECFM=U,LRECL=80,BLKSIZE=3200),
//  UNIT=      ,VOL=SER=
/*
//
```

D. Execute TIGER on test case (file 9 of RSIC tape list).

```
//jobcard; allocate 50 minutes for this job.
//S1 EXEC PGM=TIGER,REGION=2048K
//STEPLIB DD DSN=TIGLOAD(TIGER),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS1,DISP=SHR
//FT12F001 DD DUMMY
```

```
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
*  MAT  NZONE  THIK   ECUT  PTCZ
 GEOMETRY 2
  1 2 0.007
  2 2 0.05
************************* SOURCE *********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
* DEFAULT DIRECTION
 DIRECTION 0.0
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 2
 PHOTON-ESCAPE
   NBINE 2
   NBINT 2
 ELECTRON-FLUX 1 4
   NBINE 2
 PHOTON-FLUX 1 4
   NBINE 2
************************* OTHER OPTIONS ************************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
/*
//
```

III. Running the 'TIGERP' code.

A. Execute UPEML to create a compile ready file for the TIGER code. This is a correction run. The input to FT05F001 is from the first few lines of file 11, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL     DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
```

```
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=TIGPLST,UNIT=      ,VOL=SER=       ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE TIGERP.
//FT20F001 DD DSN=CTIGERP,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,TIGER
*DEFINE,PCODES
*IDENT,TEMQ
*D,PARAMS.5
      PARAMETER ( INMT=2,     NELEM=20,    INMAX=64,   NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  2,   IJMAX  =  2,
     $             IKPMAX =  2,   IJPMAX =  2,
     $             INZON  = 2)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,   IJFMP1 = IJFMXP+1,
     $             INLF   = 4,    INLFP  = 4)
*D,PARAMS.52
      PARAMETER (NESC  = 2,   ILZMAX = 4,    ILZMX1 = ILZMAX+1,
*D,PARAMS.82
      PARAMETER(INEM=1, INEM1=INEM, INSH=6, INZZ=2,
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
*DELETE,BETAM.7
      DOUBLE PRECISION FUNCTION BETAM(E)
*DELETE,PEC.7
      DOUBLE PRECISION FUNCTION PEC(EXL,EPH,BETAE,Z,EK)
/*
//
```

B. Truncate the 90-character TIGERP file to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY      EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2    DD DSN=FTIGERP,UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(200,100),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
// DISP=(NEW,CATLG)
//SYSUT1    DD DSN=CTIGERP,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN     DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for TIGERP.

```
//jobcard; allocate 1 minute for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
 Insert JSTIME source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FTIGERP,DISP=SHR
/*
//LKED.SYSLIB DD
//            DD
//            DD
//            DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=TIGLOAD(TIGERP),DISP=OLD
/*
//
```

D. Execute TIGERP on test case (file 11 of RSIC tape list).

```
//jobcard; allocate 70 minutes for this job.
//S1 EXEC PGM=TIGERP,REGION=2048K
//STEPLIB DD DSN=TIGLOAD(TIGERP),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS2,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
```

```
************************* GEOMETRY ********************************
*  MAT  NZONE  THIK   ECUT  PTCZ
 GEOMETRY 2
  1 2 0.007
  2 2 0.05
************************* SOURCE *********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
* DEFAULT DIRECTION
 DIRECTION 0.0
************************* OUTPUT OPTIONS *************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 2
 PHOTON-ESCAPE
   NBINE 2
   NBINT 2
 ELECTRON-FLUX 1 4
   NBINE 2
 PHOTON-FLUX 1 4
   NBINE 2
************************* OTHER OPTIONS **************************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
/*
//
```

IV. Running the 'CYLTRAN' code.

A. Execute UPEML to create a compile ready file for CYLTRAN. This is a correction run. The input to FT05F001 is from the first few lines of file 15, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=UPELOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL    DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=CYLLST,UNIT=      ,VOL=SER=      ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE CYLTRAN.
//FT20F001 DD DSN=CCYLTRAN,DISP=(NEW,CATLG,DELETE),
```

```
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,CYLTRAN
*IDENT,TEMQ
*D,PARAMS.5
      PARAMETER ( INMT=2,      NELEM=20,     INMAX=64,    NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  4,    IJMAX  =  2,
     $             IKPMAX =  4,    IJPMAX =  2,
     $             INZON  = 9)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,    IJFMP1 = IJFMXP+1,
     $             INLF   = 8,     INLFP  = 8)
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
/*
//
```

B. Truncate the 90-character CYLTRAN code to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FCYLTRAN,UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
// DISP=(NEW,CATLG)
//SYSUT1   DD DSN=CCYLTRAN,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for CYLTRAN.

```
//jobcard; allocate 1 minute for this job.
// EXEC ASMFCL
//ASM.SYSIN *
 Insert the JSTIME source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FCYLTRAN,DISP=SHR
/*
//LKED.SYSLIB DD
//             DD
//             DD
//             DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=CYLLOAD(CYLTRAN),DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(100,50,5),RLSE),
// DCB=(RECFM=U,LRECL=80,BLKSIZE=3200)
/*
//
```

D. Execute CYLTRAN on test case (file 15 of RSIC tape list).

```
//jobcard; allocate 75 minutes for this job.
//S1 EXEC PGM=CYLTRAN,REGION=2048K
//STEPLIB DD DSN=CYLLOAD(CYLTRAN),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS1,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
*  ZL   ZR   RI   RO   MAT   IZR   NZ   NR   ECUT PTCZ
 GEOMETRY 3
  -0.007      0.0         0.0         10.0          1   1   2   2
  0.1         0.15        0.0         10.0          2   1   2   2
  0.0         0.1         0.0         10.0
************************* SOURCE ********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
 POSITION 0.0 0.0 -0.007
   RADIUS 2.50
* DEFAULT DIRECTION
 DIRECTION 0.0 0.0
```

```
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 4
 PHOTON-ESCAPE
   NBINE 2
   NBINT 4
 ELECTRON-FLUX 1 8
   NBINE 2
 PHOTON-FLUX 1 8
   NBINE 2
************************* OTHER OPTIONS ************************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
/*
//
```

V. Running the 'CYLTRANP' code.

A. Execute UPEML to create a compile ready file for CYLTRANP. This is a correction run. The input to FT05F001 is from the first few lines of file 17, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=ITSLOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL     DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=CYLPLST,UNIT=      ,VOL=SER=       ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE CYLTRANP.
//FT20F001 DD DSN=CCYLTRAP,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
```

```
*DEFINE,CYLTRAN
*DEFINE,PCODES
*IDENT,TEMQ
*D,PARAMS.5
      PARAMETER ( INMT=2,      NELEM=20,     INMAX=64,    NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  4,   IJMAX  =  2,
     $             IKPMAX =  4,   IJPMAX =  2,
     $             INZON  = 9)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,   IJFMP1 = IJFMXP+1,
     $             INLF   = 8,    INLFP  = 8)
*DELETE,PARAMS.82
      PARAMETER(INEM=1, INEM1=INEM, INSH=6, INZZ=2,
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
*DELETE,BETAM.7
      DOUBLE PRECISION FUNCTION BETAM(E)
*DELETE,PEC.7
      DOUBLE PRECISION FUNCTION PEC(EXL,EPH,BETAE,Z,EK)
/*
//
```

B. Truncate the 90-character CYLTRANP code to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FCYLTRAP,UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
// DISP=(NEW,CATLG)
//SYSUT1   DD DSN=CCYLTRAP,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for CYLTRANP.

```
//jobcard; allocate 1 minute for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
Insert JSTIME source here
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FCYLTRAP,DISP=SHR
/*
//LKED.SYSLIB DD
//             DD
//             DD
//             DD DSN=IMSL,DISP=SHR
//*COMMENT DATA SET CYLLOAD WAS ALLOCATED PREVIOUSLY.
//LKED.SYSLMOD DD DSN=CYLLOAD(CYLTRANP),DISP=OLD
/*
//
```

D. Execute CYLTRANP on test case (file 17 of RSIC tape list).

```
//jobcard; allocate 105 minutes for this job.
//S1 EXEC PGM=CYLTRANP,REGION=2048K
//STEPLIB DD DSN=CYLLOAD(CYLTRANP),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS2,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
*  ZL   ZR   RI   RO   MAT   IZR   NZ   NR   ECUT PTCZ
 GEOMETRY 3
  -0.007      0.0         0.0         10.0          1   1   2   2
  0.1         0.15        0.0         10.0          2   1   2   2
  0.0         0.1         0.0         10.0
************************* SOURCE *********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
 POSITION 0.0 0.0 -0.007
   RADIUS 2.50
* DEFAULT DIRECTION
 DIRECTION 0.0 0.0
```

```
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 4
 PHOTON-ESCAPE
   NBINE 2
   NBINT 4
 ELECTRON-FLUX 1 8
   NBINE 2
 PHOTON-FLUX 1 8
   NBINE 2
************************* OTHER OPTIONS **********************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
/*
//
```

VI. Running the 'CYLTRANM' code.

A. Execute UPEML to create a compile ready file for CYLTRANM. This is a correction run. The input to FT05F001 is from the first few lines of file 19, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=ITSLOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL     DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=CYLMLST,UNIT=      ,VOL=SER=       ,
//  SPACE=(TRK,(10,5),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE CYLTRANM.
//FT20F001 DD DSN=CCYLTRAM,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,CYLTRAN
*DEFINE,MCODES
*IDENT,TEMQ
```

```
*D,BFLD.42
*I,56
      EX = 0.0D0
      EY = 0.0D0
      EZCON = -1.0D0
*I,BFLD.57
      EZ = 0.0
      BX = 0.0
      BY = 0.0
      BZ = 1.0
      RETURN
*D,PARAMS.5
      PARAMETER ( INMT=2,      NELEM=20,     INMAX=64,    NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  4,    IJMAX  =  2,
     $             IKPMAX =  4,    IJPMAX =  2,
     $             INZON  = 9)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,    IJFMP1 = IJFMXP+1,
     $             INLF   = 8,     INLFP  = 8)
*DELETE,PARAMS.82
      PARAMETER(INEM=1, INEM1=INEM, INSH=6, INZZ=2,
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
/*
//
```

B. Truncate the 90-character CYLTRANM code to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FCYLTRAM,UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
// DISP=(NEW,CATLG)
//SYSUT1   DD DSN=CCYLTRAM,DISP=SHR,
```

```
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN     DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for CYLTRANM.

```
//jobcard; allocate 5 minutes for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
 Insert JSTIME source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FCYLTRAM,DISP=SHR
/*
//LKED.SYSLIB DD
//             DD
//             DD
//             DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=CYLLOAD1(CYLTRANM),DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=      ,SPACE=(TRK,(100,50,5),RLSE),
// DCB=(RECFM=U,LRECL=80,BLKSIZE=3200)
//
```

D. Execute CYLTRANM on test case (file 19 of RSIC tape list).

```
//jobcard; allocate 65   minutes for this job.
//S1 EXEC PGM=CYLTRANM,REGION=2048K
//STEPLIB DD DSN=CYLLOAD1(CYLTRANM),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS1,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
*  ZL   ZR   RI   RO   MAT   IZR   NZ   NR   IFLD   ECUT PTCZ
 GEOMETRY 3
  -0.007      0.0         0.0         10.0   1   1   2   2
  0.1         0.15        0.0         10.0   2   1   2   2
  0.0         0.1         0.0         10.0   0   0   0   0   1
```

```
************************* SOURCE ********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
 POSITION 0.0 0.0 -0.007
   RADIUS 2.50
* DEFAULT DIRECTION
 DIRECTION 0.0 0.0
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 4
 PHOTON-ESCAPE
   NBINE 2
   NBINT 4
 ELECTRON-FLUX 1 8
   NBINE 2
 PHOTON-FLUX 1 8
   NBINE 2
************************* OTHER OPTIONS *********************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
*PLOTS
/*
//
```

VII. Running the 'ACCEPT' code.

A. Execute UPEML to create a compile ready file for ACCEPT. This is a correction run. The input to FT05F001 is from the first few lines of file 23, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL    DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=ACCLST,UNIT=      ,VOL=SER=       ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE ACCEPT.
//FT20F001 DD DSN=CACCEPT,DISP=SHR
```

```
//FT20F001 DD DSN=CACCEPT,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,ACCEPT
*IDENT,TEMQ
*D,PARAMS.5
      PARAMETER ( INMT=2,     NELEM=20,    INMAX=64,   NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  4,   IJMAX  =  2,
     $             IKPMAX =  4,   IJPMAX =  2,
     $             INZON  = 4)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,   IJFMP1 = IJFMXP+1,
     $             INLF   = 2,    INLFP  = 2)
*DELETE,PARAMS.71
     $  IFPD = 43, INUMR = 4, ITMA = 88, NAZ = 5,
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
*/COMMENT THE NEXT LINE IS INSERTED IN SUBROUTINE OPREAD
*/        SO THAT THE FORTRAN UNIT NUMBER IS DEFINED.
*/         THIS IS AN IBM MVS PROBLEM.
*INSERT,OPREAD.48
       IF (NIN   .GT. 100) NIN = NUNIT
/*
//
```

B. Truncate the 90-character ACCEPT code to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FACCEPT,UNIT=      ,VOL=SER=       ,
// SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
// DISP=(NEW,CATLG)
//SYSUT1   DD DSN=CACCEPT,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for ACCEPT.

```
//jobcard; allocate 5 minutes for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
 Insert JSTIME source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FACCEPT,DISP=SHR
/*
//LKED.SYSLIB DD
//            DD
//            DD
//            DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=ACCLOAD(ACCEPT),DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100,5),RLSE),
// DCB=(RECFM=U,LRECL=80,BLKSIZE=3200)
/*
//
```

D. Execute ACCEPT on test case (file 23 of RSIC tape list).

```
//jobcard; allocate 75 minutes for this job.
//S1 EXEC PGM=ACCEPT,REGION=2048K
//STEPLIB DD DSN=ACCLOAD(ACCEPT),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS1,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
```

```
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
 GEOMETRY
   RCC       0.00       0.00       0.00       0.00       0.00       -0.007
             10.0
   RCC       0.00       0.00       0.00       0.00       0.00       0.10
             10.0
   RCC       0.00       0.00       0.10       0.00       0.00       0.05
             10.0
   SPH       0.00       0.00       0.00       12.0
   END
 Z1    +1
 Z2    +3
 Z3    +2
 Z4    +4    -1    -2    -3
   END
* MAT  ECUT  PTCZ
  1
  2
  0
  0
************************* SOURCE ********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
 POSITION 0.0 0.0 -0.007
   RADIUS 2.50
* DEFAULT DIRECTION
 DIRECTION 0.0 0.0
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 4
 PHOTON-ESCAPE
   NBINE 2
   NBINT 4
 ELECTRON-FLUX 1 2
   NBINE 2
 PHOTON-FLUX 1 2
   NBINE 2
************************* OTHER OPTIONS ************************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
* PLOTS 3
*  -15.0   15.0   -15.0   15.0   180.   90.
*  -15.0   15.0   -15.0   15.0   0.0    0.0
*  -15.0   15.0   -15.0   15.0   180.   30.
/*
//
```

VIII. Running the 'ACCEPTP' code.

A. Execute UPEML to create a compile ready file for ACCEPTP. This is a correction run. The input to FT05F001 is from the first few lines of file 27, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=ITSLOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL    DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=ACCPLST,UNIT=      ,VOL=SER=       ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE ACCEPTP.
//FT20F001 DD DSN=CACCEPTP,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=       ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,ACCEPT
*DEFINE,PCODES
*IDENT,TEMQ
*D,PARAMS.5
      PARAMETER ( INMT=2,     NELEM=20,    INMAX=64,   NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  4,   IJMAX  =  2,
     $             IKPMAX =  4,   IJPMAX =  2,
     $             INZON  = 4)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,   IJFMP1 = IJFMXP+1,
     $             INLF   = 2,    INLFP  = 2)
*DELETE,PARAMS.71
     $  IFPD = 43, INUMR = 4, ITMA = 88, NAZ = 5,
```

```
*DELETE,PARAMS.82
      PARAMETER(INEM=1, INEM1=INEM, INSH=6, INZZ=2,
*INSERT,ITS.117
      CALL ERRSET(208,256,1,1,1)
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
*DELETE,BETAM.7
      DOUBLE PRECISION FUNCTION BETAM(E)
*DELETE,PEC.7
      DOUBLE PRECISION FUNCTION PEC(EXL,EPH,BETAE,Z,EK)
/*
//
```

B. Truncate the 90-character ACCEPTP code to one of 80-character length.

```
//jobcard; allocate 5 seconds for this job.
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD DSN=FACCEPTP,UNIT=      ,VOL=SER=      ,
// SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
// DISP=(NEW,CATLG)
//SYSUT1   DD DSN=CACCEPTP,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for ACCEPTP.

```
//jobcard; allocate 5 minutes for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
 Insert JSTIME source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//      DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FACCEPTP,DISP=SHR
/*
```

```
//LKED.SYSLIB DD
//            DD
//            DD
//            DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=ACCLOAD(ACCEPTP),DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=      ,SPACE=(TRK,(200,100,5),RLSE),
// DCB=(RECFM=U,LRECL=80,BLKSIZE=3200)
/*
//
```

D. Execute ACCEPTP on test case (file 27 of RSIC tape list).

```
//jobcard; allocate 110 minutes for this job.
//S1 EXEC PGM=ACCEPTP,REGION=2048K
//STEPLIB DD DSN=ACCLOAD(ACCEPTP),DISP=SHR
//FT07F001 DD DUMMY
//FT11F001 DD DSN=XSECS2,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
 GEOMETRY
   RCC      0.00      0.00      0.00      0.00      0.00      -0.007
            10.0
   RCC      0.00      0.00      0.00      0.00      0.00      0.10
            10.0
   RCC      0.00      0.00      0.10      0.00      0.00      0.05
            10.0
   SPH      0.00      0.00      0.00      12.0
   END
 Z1   +1
 Z2   +3
 Z3   +2
 Z4   +4   -1   -2   -3
   END
* MAT  ECUT  PTCZ
  1
  2
  0
  0
************************* SOURCE ********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
 POSITION 0.0 0.0 -0.007
   RADIUS 2.50
* DEFAULT DIRECTION
 DIRECTION 0.0 0.0
```

```
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 4
 PHOTON-ESCAPE
   NBINE 2
   NBINT 4
 ELECTRON-FLUX 1 2
   NBINE 2
 PHOTON-FLUX 1 2
   NBINE 2
************************* OTHER OPTIONS ************************
 HISTORIES 10000
 BATCHES 10
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.
* PLOTS 3
*  -15.0   15.0   -15.0   15.0   180.   90.
*  -15.0   15.0   -15.0   15.0   0.0    0.0
*  -15.0   15.0   -15.0   15.0   180.   30.
/*
//
```

IX. Running the 'ACCEPTM' code.

A. Execute UPEML to create a compile ready file for ACCEPTM. This is a correction run. The input to FT05F001 is from the first few lines of file 29, ended by 'eor,' plus some other directives as described in the introduction.

```
//jobcard; allocate 2 minutes for this job.
//S1 EXEC PGM=UPEML,REGION=600K
//STEPLIB DD DSN=ITSLOAD(UPEML),DISP=SHR
//*COMMENT DDNAME OLDPL (UNIT 13)
//GO.OLDPL    DD DSN=ITS1,DISP=SHR
//*COMMENT UNIT 15 DSN IS A TEMPORARY FILE WHICH GETS DELETED.
//GO.FT15F001 DD DSN=&&TEMP1,DISP=(NEW,PASS,DELETE),
// UNIT=SYSDA,SPACE=(1024,5000),
// DCB=(RECFM=F,BLKSIZE=1024)
//*COMMENT UNIT 16 IS THE LIST FILE OF CORRECTIONS.
//FT16F001 DD DSN=ACCMLST,UNIT=      ,VOL=SER=      ,
//  SPACE=(TRK,(200,100),RLSE),
//  DCB=(RECFM=FB,LRECL=133,BLKSIZE=3059),DISP=(NEW,CATLG)
//FT17F001 DD DUMMY
//*COMMENT UNIT 20 DSN IS THE COMPILE FILE ACCEPTM.
//FT20F001 DD DSN=CACCEPTM,DISP=(NEW,CATLG,DELETE),
// UNIT=      ,VOL=SER=      ,SPACE=(TRK,(200,100),RLSE),
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
F,P,L
```

```
*DEFINE,IBM
*DEFINE,DOUBLE
*DEFINE,ACCEPT
*DEFINE,MCODES
*IDENT,TEMQ
*/COMMENT THE NEXT DELETE AND INSERT ARE IBM VS FORTRAN REQUIREMENTS.
*D,BFLD.42
*I,56
      EX = 0.0D0
      EY = 0.0D0
      EZCON = -1.0D0
*I,BFLD.57
      EZ = 0.0
      BX = 0.0
      BY = 0.0
      BZ = 1.0
      RETURN
*D,PARAMS.5
      PARAMETER ( INMT=2,     NELEM=20,    INMAX=64,   NSURV=2775,
*DELETE,PARAMS.12,PARAMS.15
      PARAMETER ( IMTOP1=IMTOP, INMAX1=INMAX+1, INMTP1=INMT+1,
     $            INEEL1=INEEL, INGAS1=INGAS+1,  INLAN1=INLAN+1,
     $            INEPS1=INEPS, NSURV1=NSURV+1,
     $            INRNG1=INRANG, INTNG1=INTANG)
*D,PARAMS.24,PARAMS.26
      PARAMETER (  IKMAX  =  4,   IJMAX  =  2,
     $             IKPMAX =  4,   IJPMAX =  2,
     $             INZON  = 4)
*D,PARAMS.33
      PARAMETER ( IJSMAX = 1, IJFMAX =2,
*D,PARAMS.35,PARAMS.36
     $             IJFMXP = 2,   IJFMP1 = IJFMXP+1,
     $             INLF   = 2,    INLFP  = 2)
*D,PARAMS.71,PARAMS.72
     $  IFPD = 33, INUMR = 4, ITMA = 88, NAZ = 5,
     $  IJTY =  4, IARB  = 1, NVIEWS = 3, NCZONE = 20)
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT (D25.0)
 88   FORMAT (' THE RESTART RANDOM NUMBER IS ', D25.0)
*DELETE,INPUT.1044,INPUT.1106
*DELETE,INPUT.1121,INPUT.1148
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
*/COMMENT THE NEXT LINE IS INSERTED SO THAT THE FORTRAN
*/        UNIT NUMBER IS DEFINED. THIS IS AN IBM MVS PROBLEM.
*INSERT,OPREAD.48
       IF (NIN   .GT. 100) NIN = NUNIT
/*
//
```

B. Truncate the 90-character ACCEPTM code to one of 80-character length.

```
//BLKTRUN JOB (13498,IO2),'KIRK    6025',TIME=(,5),
//COPY     EXEC PGM=IEBGENER,REGION=300K
//SYSPRINT DD SYSOUT=A
//*SYSUT2   DD DSN=FACCEPTM,DISP=OLD,
//*DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//SYSUT2   DD DSN=FACCEPTM,DISP=OLD
//*SYSUT2   DD DSN=FACCEPTM,UNIT=      ,VOL=SER=      ,
//*SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
//*DISP=(NEW,CATLG)
//SYSUT1   DD DSN=CACCEPTM,DISP=SHR,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD *
      GENERATE MAXFLDS=1
      RECORD FIELD=(80)
/*
//
```

C. Create an executable load module for ACCEPTM.

```
//jobcard; allocate 5 minutes for this job.
// EXEC ASMFCL
//ASM.SYSIN DD *
 Insert JSTIME  source here.
// EXEC FORTVCL,PARM.FORT='XREF',REGION.FORT=2048K
//FORT.SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(50,5),RLSE),
//     DCB=BLKSIZE=3200
//FORT.SYSIN DD DSN=FACCEPTM,DISP=SHR
/*
//LKED.SYSLIB DD
//            DD
//            DD
//            DD DSN=IMSL,DISP=SHR
//LKED.SYSLMOD DD DSN=ACCLOAD1(ACCEPTM),DISP=(NEW,CATLG),
// UNIT=      ,VOL=SER=      ,SPACE=(TRK,(200,100,5),RLSE),
// DCB=(RECFM=U,LRECL=80,BLKSIZE=3200)
/*
//
```

D. Execute ACCEPTM on test case (file 29 of RSIC tape list).

```
//jobcard; allocate 65 minutes for this job.
//S1 EXEC PGM=ACCEPTM,REGION=2048K
//STEPLIB DD DSN=ACCLOAD1(ACCEPTM),DISP=SHR
//FT07F001 DD DUMMY
```

```
//FT11F001 DD DSN=XSECS1,DISP=SHR
//FT12F001 DD DUMMY
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3520),
// SPACE=(3520,(200,100))
//FT05F001 DD *
 ECHO 1
 TITLE
 ...1.0 MEV TA/AL TEST PROBLEM
************************* GEOMETRY ********************************
 GEOMETRY
   RCC      0.00      0.00      0.00      0.00      0.00      -0.007
            10.0
   RCC      0.00      0.00      0.00      0.00      0.00      0.10
            10.0
   RCC      0.00      0.00      0.10      0.00      0.00      0.05
            10.0
   SPH      0.00      0.00      0.00      12.0
   END
 Z1   +1
 Z2   +3
 Z3   +2
 Z4   +4   -1   -2   -3
   END
* MAT  ECUT  PTCZ
  1
  2
  0
  0
************************* SOURCE ********************************
 ELECTRONS
 ENERGY 1.0
 CUTOFFS 0.05 0.001
 POSITION 0.0 0.0 -0.007
   RADIUS 2.50
* DEFAULT DIRECTION
 DIRECTION 0.0 0.0
************************* OUTPUT OPTIONS ************************
 ELECTRON-ESCAPE
   NBINE 2
   NBINT 4
 PHOTON-ESCAPE
   NBINE 2
   NBINT 4
 ELECTRON-FLUX 1 2
   NBINE 2
 PHOTON-FLUX 1 2
   NBINE 2
************************* OTHER OPTIONS *********************
 HISTORIES 10000
 BATCHES 10
```

```
* ... X-RAY PRODUCTION SCALING
 SCALE-BREMS 500.  * PLOTS 3 *  -15.0   15.0   -15.0   15.0   180.   90.
*  -15.0   15.0   -15.0   15.0   0.0    0.0 *  -15.0   15.0   -15.0   15.0
180.   30.  /* //
```

REFERENCES

[1] T. A. Melhorn and M. F. Young, *UPEML, A Machine-Portable CDC Update Emulator,* SAND 84-1896, Sandia National Laboratories, Albuquerque, New Mexico, December 1984.

[2] J. A. Halbleib and T. A. Melhorn, *ITS: The Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes,* Sand 84-0573, Sandia National Laboratories, Albuquerque, New Mexico, November 1984.

[3] Marilyn Bohl, *Introduction to IBM Direct Access Storage Devices,* Science Research Associates, Inc., 1981, Chicago.

APPENDIX A: JSTIME

```
JSTM     TITLE  'JOB STEP TIME REPORTER.'
*  FORTRAN FUNCTION-TYPE PROGRAM TO RETURN REMAINING JOB-STEP TIME
*    TO THE CALLING PROGRAM.  (TIME IS PSEUDO-TIME IN HUNDREDTH-SECONDS )
*    USAGE:    JOBTIM =JSTIME(JTIME)  JTIME & JOBTIM BOTH GET TIME.
*       OR:    CALL JSTIME(JOBTIM)   PUTS THE PSEUDO TIME INTO JOBTIM
JSTIME   START 0
         USING *,15
         SAVE  (14,12),,*
         LR    4,13
         CNOP  0,4
         BAL   13,NDSAVE
SAVE     DC    18F'0'
NDSAVE   ST    13,8(4)
         ST    4,4(13)
         USING SAVE,13
         DROP  15
         LR    2,1
         TIME  BIN
         LR    1,2
         LA    2,6            ALLOW SEARCH OF 6 TCB'S TO FIND SYS TASK.
         L     3,16           GET CVT POINTER.
         L     4,88(0,3)      POINT TO PSEUDO-CLOCK.
         L     3,0(3)         POINTER TO TCB POINTERS.
         L     3,4(3)         ACTIVE TCB POINTER.
GETOTCB  L     3,132(3)       OTCB POINTER.
         CLI   34(3),X'FF'      HIGHEST DISPATCHING PRIORITY?
         BE    HAVEINIT
         BCT   2,GETOTCB      BUT TRY ONLY 6 TCB'S IN THE SEARCH.
         SR    3,3          RETURN TIME=0 IF TCB NOT FOUND.
         B     ZERO
HAVEINIT L     3,120(3)       TQE FOR INITIATOR.
         LTR   3,3            IS THERE A TIMER ELEMENT?
         BZ    ZERO           IF NOT, GIVE UP]
         L     3,12(3)        TIME OF EXPIRATION IN TQE.
         S     3,0(0,4)       GR-4 STILL POINTING AT PSEUDO-CLOCK.
         L     4,80           GET TIMER FROM LOWER CORE.
         SRA   4,1            CHANGE TO 26 MICRO-SEC. INTERVALS.
         AR    3,4             AND ADD IT TO THE EXPIRATION TIME.
TIMECALC SR    2,2
         D     2,=F'384'        CONVERT IT TO HUNDREDTH-SECONDS.
ZERO     EQU   *
         L     1,0(1)         GET POINTER TO DESTINATION IN CALLER.
         ST    3,0(1)         SHIP TIME THERE.
         LR    0,3            AND PUT IT IN GR-0 FOR FUNCTION TYPE CALL.
         L     13,SAVE+4
         SR    15,15
         L     14,12(13)
         MVI   12(13),X'FF'
         LM    2,4,28(13)
         BR    14
         END
```

APPENDIX B : EXPLANATION OF DATA SETS USED IN JCL

ACCLST	- listing of correction run for ACCEPT.
ACCMLST	- listing of correction run for ACCEPTM.
ACCPLST	- listing of correction run for ACCEPTP.
ACCLOAD(ACCEPT)	- executable load module for ACCEPT.
ACCLOAD(ACCEPTP)	- executable load module for ACCEPTP.
ACCLOAD1(ACCEPTM)	- executable load module for ACCEPTM.
CACCEPT	- compile ready ACCEPT file (LRECL=90).
CACCEPTP	- compile ready ACCEPTP file (LRECL=90).
CACCEPTM	- compile ready ACCEPTM file (LRECL=90).
CYLLOAD(CYLTRANP)	- executable load module for CYLTRANP.
CYLLOAD(CYLTRAN)	- executable load module for CYLTRAN.
CYLLOAD1(CYLTRANM)	- executable load module for CYLTRANM.
CYLLST	- listing of correction run for CYLTRAN.
CYLMLST	- listing of correction run for CYLTRANM.
CYLPLST	- listing of correction run for CYLTRANP.
CCYLTRAN	- compile ready CYLTRAN file (LRECL=90).
CCYLTRAM	- compile ready CYLTRANM file (LRECL=90).
CCYLTRAP	- compile ready CYLTRANP file (LRECL=90).
CTIGER	- compile ready TIGER file (LRECL=90).
CTIGERP	- compile ready TIGERP file (LRECL=90).
CXGEN	- compile ready XGEN file (LRECL=90) ;standard codes.
CXGEN2	- compile ready XGEN file (LRECL=90) ; P-codes.
FACCEPT	- compile ready ACCEPT file (LRECL=80).
FACCEPTM	- compile ready ACCEPTM file (LRECL=80).
FACCEPTP	- compile ready ACCEPTP file (LRECL=80).
FCYLTRAM	- compile ready CYLTRANM file (LRECL=80).
FCYLTRAN	- compile ready CYLTRAN file (LRECL=80).
FCYLTRAP	- compile ready CYLTRANP file (LRECL=80).
FTIGER	- compile ready TIGER file (LRECL=80).
FTIGERP	- compile ready TIGERP file (LRECL=80).
FXGEN	- compile ready XGEN file (LRECL=80); standard codes.
FXGEN2	- compile ready XGEN file (LRECL=80); P-codes.
IMSL	- system library of IMSL mathematical routines, in double precision. GGUBFS is from this library.
IPCODES	- data file input to XGEN2 used to generate cross sections for the P-codes (file 7)
ISTAND	- data file input to XGEN used to generate cross sections for the standard codes (file 5).
ITS	- as copied from tape (file 1).
ITS1	- new program library for ITS.
ITSLST	- listing of ITS (after being processed by UPEML).
TIGLOAD(TIGER)	- executable load module for TIGER.
TIGLOAD(TIGERP)	- executable load module for TIGERP.
TIGLST	- listing of correction run for TIGER.
TIGPLST	- listing of correction run for TIGERP.
UPELOAD(UPEML)	- executable load module for UPEML.
UPEML	- as copied from tape (file 4).
XDATA	- cross sections as copied (file 3).
XGEN	- as copied from tape (file 2).

XGENLOAD(XGEN)	- executable load module for XGEN (standard codes).
XGENLOAD(XGEN2)	- executable load module for XGEN (P-codes).
XGENLST	- listing of XGEN (after being processed by UPEML).
XSECS1	- generated cross sections for input to standard codes and M-codes.
XSECS2	- generated cross sections for input to P-codes.

APPENDIX C: IBM SPECIFIC PROBLEMS

For every IBM correction run, it is necessary to have the directives:

```
'*DEFINE,IBM'
'*DEFINE,DOUBLE'.
```

The following lists IBM problems and possible solutions.

1. 'IF-THEN-ELSE' structure

The IBM VS Fortran 77 (level 1.3.1) that was used for testing the ITS codes gave a severe error with more than 25 'IF - THEN - ELSE' in the code. It was therefore necessary to delete some of the keywords (pages 14-33, ref. 2) in Subroutine INPUT. These keywords are not necessary in running the sample problems. The user should note which keywords are needed for other problems. The keywords are deleted in the correction runs by the statements :

```
'*DELETE,INPUT.1044,INPUT.1106' and
'*DELETE,INPUT.1121,INPUT.1148'.
```

2. 'MAX' function in a PARAMETER statement.

The statement '*DELETE,PARAMS.12,PARAMS.15' was also necessary and the proper statements were inserted, because the IBM compiler does not recognize the 'MAX' function in a PARAMETER statement. In addition to this, for the P-codes, the directive '*DELETE,PARAMS.82' and the corresponding correct form was included in the correction run.

3. Defining a function subroutine with 'REAL*8'.

Statements of the form 'REAL*8 FUNCTION(....)' are unacceptable to the IBM VS Fortran compiler. For the P-codes, the specific change is thru the statements:

```
*DELETE,PEC.7
      DOUBLE PRECISION FUNCTION PEC(EXL,EPH,BETAE,Z,EK)
*DELETE,BETAM.7
      DOUBLE PRECISION FUNCTION BETAM(E)
```

4. Variables in a COMMON area.

For the M-codes, it was necessary to include the following directives that affect 'SUBROUTINE BFLD'.

```
'*DELETE,BFLD.42'
'*INSERT,56'
        EX = 0.0D0
        EY = 0.0D0
        EZCON = -1.0D0
```

This is because the variables EX, EY and EZCON appear in the named common area 'FILD'. The above delete directive eliminates the data initialization for these variables. The VS FORTRAN compiler objects to variables appearing in common areas and initialized by DATA statements.

5. FORMAT statements.

Another directive added to the correction runs involved 'FORMAT' statements. The form 'D25' is unacceptable to the VS FORTRAN compiler. The following are the corrections:

```
*DELETE,INPUT.356,INPUT.357
 4913 FORMAT(D25.0)
 88   FORMAT(' THE RESTART RANDOM NUMBER IS ',D25.0)
```

6. Unit numbers.

If operating under MVS, one should note that variables not initialized will have undefined values. For some reason, both the codes ACCEPT and ACCEPTM had trouble resolving the FORTRAN read unit number 5 in subroutine OPREAD. The subroutine tests the variable NIN for a value of 0 or less. Since the VS FORTRAN compiler recognizes unit numbers 0-99, the following is included in the correction runs for the ACCEPT and ACCEPTM codes:

```
*INSERT,OPREAD.48
       IF (NIN .GT. 99) NIN=NUNIT
```

7. Underflows

Both the ACCEPTP and CYLTRANP codes produced underflows in running the sample problems. The underflows occurred in SUBROUTINE KXRAY, particularly in the statement

```
      CWCF = CWCF/RWT(ND,KLMN,MT)
```

whose execution does not affect output. This is a known bug. A temporary solution for an IBM run is to include the following statements:

```
*INSERT,ITS.117
      CALL ERRSET(208,256,1,1,1)
```

8. System dependent timing subroutine.

Because the subroutine that returns the CPU remaining time is system dependent, a subroutine JSTIME (Appendix A-IBM assembly) is included. This is reflected in the directives :

```
*DELETE,TIMER.42
      CALL JSTIME(ICPU)
      ICPU = ICPU / 100
```

Even with the above corrections, the user should be aware of several Level 4 error warnings when compiling the codes. These are just minor errors.

DISTRIBUTION

1-2. Central Research Library
3. ORNL-Y-12 Technical Library
Document Reference Section
4-5. Laboratory Records Department
6. Laboratory Records ORNL-RC
7. ORNL Patent Office
8-9. L. S. Abbott
10. R. G. Alsmiller, Jr.
11. D. E. Bartine
12. D. G. Cacuci
13. R. H. Cooper, Jr.
14. T. A. Gabriel
15. B. L. Kirk
16. F. C. Maienschein
17. F. R. Mynatt
18. R. W. Peelle
19. M. W. Rosenthal
20. R. T. Santoro
21. R. C. Ward
22. C. R. Weisbin
23. A. Zucker
24. P. W. Dickson, Jr. (Consultant)
25. G. H. Golub (Consultant)
26. R. M. Haralick (Consultant)
27. D. Steiner (Consultant)
28-47. Radiation Shielding
Information Center

EXTERNAL DISTRIBUTION

48. D. A. Auton, Defense Nuclear Agency, 6801 Telegraph Road, Alexandria, VA 22301.

49. S. E. Berk, Division of Development and Technology, Office of Fusion Energy, ER-533, U.S. Department Energy, Washington, DC 20545.

50. J. A. Halbleib, Simulation Theory Division, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185.

51. P. B. Hemmig, Safety and Physics Div., Office of Advanced Reactor Programs, NE-53, U.S. Department Energy, Washington, DC 20545.

52. Robert Howerton, Lawrence Livermore National Laboratory, L-71, P.O. Box 808, Livermore, CA 94550.

53. T. A. Melhorn, Pulsed Power Theory Division, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185.

54. J. H. Renken, Div. 2320, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185.

55. Yo Taik Song, Division of Safety, Environment and Emergency Actions (DP-226), DOE,

Washington, DC 20545.

56. W. L. Thompson, Office of the Director, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545.

57. E. J. Wahlquist, Office of Space Reactor Projects, (WE-54), U.S. Department Energy, Washington, DC 20545.

58. Office of Assistant Manager for Energy Research and Development, DOE-ORO, Oak Ridge, TN 37831.

59-61. Technical Information Center, P.O. Box 62, Oak Ridge, TN 37830.