

AD-A144 898

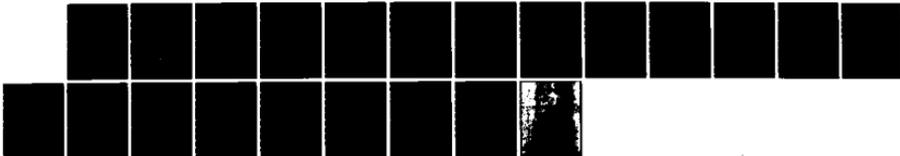
SANDYL RADIATION TRANSPORT CODE AT THE NAVAL RESEARCH
LABORATORY(U) NAVAL RESEARCH LAB WASHINGTON DC
J B LANGWORTHY 25 JUL 84 NRL-MR-5381 SBI-AD-E888 598

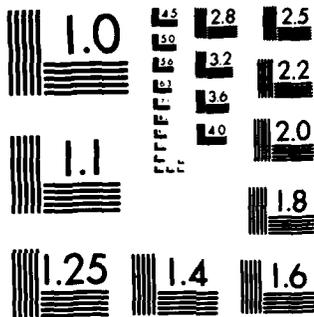
1/1

UNCLASSIFIED

F/G 18/8

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

2

AT-EDDDE??

NRL Memorandum Report 5381

SANDYL Radiation Transport Code at the Naval Research Laboratory

J. B. LANGWORTHY

*Radiation Survivability Branch
Condensed Matter and Radiation Sciences Division*

July 25, 1984

AD-A144 890

DTIC FILE COPY



NAVAL RESEARCH LABORATORY
Washington, D.C.

AUG 21 1984
A

Approved for public release: distribution unlimited.

84 08 21 042

A31-117

REPORT DOCUMENTATION PAGE				
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION / AVAILABILITY OF REPORT		
2b DECLASSIFICATION / DOWNGRADING SCHEDULE		Approved for public release; distribution unlimited.		
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 5381		5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory	6b OFFICE SYMBOL (if applicable) Code 6610	7a. NAME OF MONITORING ORGANIZATION		
6c ADDRESS (City, State, and ZIP Code) Washington, DC 20375		7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Office of Naval Research	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State, and ZIP Code) Arlington, VA 22217		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO. 61153N	PROJECT NO. RR012-01-40	WORK UNIT ACCESSION NO.
11 TITLE (Include Security Classification) SANDYL Radiation Transport Code at the Naval Research Laboratory				
12 PERSONAL AUTHOR(S) J. B. Langworthy				
13a. TYPE OF REPORT Interim	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) July 25, 1984	15. PAGE COUNT 21	
16 SUPPLEMENTARY NOTATION				
17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	Radiation transport code Radiation damage	
			Electron transport code	
19 ABSTRACT (Continue on reverse if necessary and identify by block number)				
SANDYL is a Monte Carlo photon and electron transport code used mainly to calculate dose in any material. This report is a history of the translation and modification of the local version, which is being run on the Texas Instrument Advanced Scientific Computer, and the report details how this version differs from the original versions. Together with original documentation, it will serve as a user's manual. Use of codes like SANDYL or of equivalent more expensive, experimental information is necessary in radiation-vulnerability studies and hardness design work on Navy satellites. The special utility of SANDYL lies in its ability to handle rather general geometries while retaining a close connection with physical mechanism.				
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL J. B. Langworthy		22b TELEPHONE (Include Area Code) (202) 767-3938	22c. OFFICE SYMBOL Code 6611	

CONTENTS

I. Introduction	1
II. Translation and Modifications	2
III. User Information	10
References	17

FIGURES

1. Comment section of routine TMAPQ	3
2. Job activity file for a normal run	11
3. A portion of a DEBUG\$\$ file	15

TABLES

I. Sample problem CP times	8
II. Changes to input cards	13



A-1

SANDYL Radiation Transport Code
at the
Naval Research Laboratory

I. INTRODUCTION

SANDYL¹ is a Monte Carlo transport code with a two dimensional (and restricted three dimensional) cylindrical geometry which may be used to calculate dose deposited in any material by x- or gamma-radiation or by energetic electrons. It has been available at the Naval Research Laboratory for general use for nearly six years and probably has been run several hundred times on productive calculations here. The original source, obtained in the spring of 1975, consisted of more than 13,000 lines. The local version is substantially shorter mainly because common block duplication is avoided. It consists of 50 routines in addition to mathematical and utility routines from the local library. This report will detail the translation and modification of SANDYL for use on the Texas Instrument Advanced Scientific Computer (ASC) here. Also, information facilitating use of the local version will be offered.

The special utility of SANDYL lies in problems combining a need for moderately complex geometry with the possibility of eliciting contributory physical mechanisms. Examples of recent work at NRL using SANDYL include studies of:

- a. satellite dosimeters,
- b. backscattering from a beam of electrons in air,
- c. gamma ray buildup from embedded emitters in a layered structure,
- d. dose deposition from an electron beam of specified profile,
- e. effects of a gold "solder" blob in the center of a CMOS dosimeter,
- f. an absorption spectrometer for use with Casino radiation facility.

The translation of a code, preparing one working on one computer to work on a different computer, proceeds generally in four phases: (a) character set (e.g., ASCII) translation, (b) language (e.g., some dialect of FORTRAN) translation, (c) verification (e.g., by comparing to accepted results), and (d) refinements (e.g., taking advantage of local computer capabilities). What follows conforms to this outline but much of it is in the last phase or beyond. Thus some of the modifications made are not really for translation purposes. One modification amounts to a change to the original implementation. Somewhere in the second phase the entire source was set up under the local Source Management System (SMS), a source line editor similar to the more common UPDATE editor, which provides an audit trail documenting changes. After setting up the source all subsequent changes are entered by means of modsets, of which there are now fourteen. The next section will describe each phase of translation and each modset in

Manuscript approved May 8, 1984.

detail. The last section offers warnings and advice to the user and exhibits the Job Specification Language (JSL) for certain types of runs. It will be assumed that the user has a copy of Reference 1 and knows relevant ASC procedures.

II. TRANSLATION AND MODIFICATIONS

The following background on SANDYL, especially on aspects that remain unchanged, may be helpful. In any Monte Carlo calculation the following phases occur: (a) input and problem setup, (b) random walk, and (c) scoring and output. In order to save memory the code has an overlay organization with one node. Thus there is a section of code residing in memory at all times, the root segment or segment 1. The remaining code is in four segments, only one of which is in memory at any one time. Segments 2 through 5 correspond to Monte Carlo phase as follows: (a) segment 2: card input and photon data input from tape, (b) segment 3: random walk and scoring module for photons only, (c) segment 4: electron data input from tape, and (d) segment 5: random walk and scoring module for photons and electrons. Segment 1 consists of the run supervisor, TMAPQ, mathematical and utility routines used by more than one other segment, and a large blank common which stores all input constants, material data, and parameters for communicating between overlays. Part of the blank common area is a 30,000 word array, storing material constants using a scheme of computed relative addressing designed to utilize minimum memory for each problem. This allows the user the convenience of tailoring memory use to needs at the expense of a new compile and link. Routines heading segments 2 through 5 are POTL, POT3, DATPAC and POT5, respectively. Each overlay has its own common blocks and in particular the random walk modules have common blocks for dynamic variables and scoring registers. As noted in reference 1, portions of this code are from an early version of ETRAN.² In particular DATPAC, the electron data tape, and the electron random walk portion of POT5 have been adopted with little change. Thus the DATPAC segment and its associated tape are the only parts affected by the SANDYL upgrade incorporating the low energy electron elastic scattering correction.³ This latest version of SANDYL, available since about 1976, will be referred to below as SANDYL II. Certain recent improvements⁴ in electron data, mainly in bremsstrahlung data, have yet to be added. Without being completely exhaustive, data input on cards may be described as problem title, zone parameters, random walk parameters, source geometry option and parameters, source spectrum options and the spectrum, if any, geometry, continuation option, output options, scoring options, variance reduction and physics options, assignment of material for each zone, and elemental composition of each material.

SANDYL is a product of Sandia Laboratories (Figure 1 exhibits a standard disclaimer, the part enclosed in asterisks) and the source listings received from them are for the CDC 6600. Phase 1 and part of phase 2 of the translation was accomplished by the staff of NRL's Research Computer Center (RCC). Their work on phase 2 mainly involved obvious changes required by the known differences in input and output between the 6600 and the ASC. Because of differences in linkers, some changes were made in routine names. Thus under 6600, the output routines for segments 3 and 5 receive the same name, EDIT, even though the two are not identical. Typically the routine called in segment 5 has had a "2" added under ASC, e.g., EDIT2. Where under

```

SUBROUTINE TMAPQ
C *****
C * ISSUED BY SANDIA LABORATORIES,
C * A PRIME CONTRACTOR TO THE
C * UNITED STATES ATOMIC ENERGY COMMISSION
C ***** NOTICE *****
C * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE
C * UNITED STATES GOVERNMENT. NEITHER THE UNITED STATES NOR THE
C * UNITED STATES ATOMIC ENERGY COMMISSION, NOR ANY OF THEIR
C * EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTOR, OR THEIR
C * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY
C * LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS
C * OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT OR PROCESS
C * DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE
C * PRIVATELY OWNED RIGHTS.
C *****
C * THE BASIC REFERENCE DOCUMENT FOR THIS CODE IS SCL-OR-720109,
C * NOVEMBER, 1972.
C * THIS CODE IS SUPPLIED FOR OFFICIAL GOVERNMENT USE ONLY. NO
C * FURTHER DISSEMINATION IS PERMITTED WITHOUT SPECIFIC PERMISSION
C * FROM SANDIA LABORATORIES, LIVERMORE, CALIFORNIA 94550.
C *****
C CODE HIST AT NRL BEFORE SMS: 77:2 TRANSLATED TO ASC FTM - LUN CHANGES: FCRJ01
C IN=5, OUT=6, PHOT TAPE=8, ELEC TAPE=9, 7LD 7=17 - 7LD CHAIN IN TMAPQ FCRJ01
C 77:7 INPUT MOSTLY VERIFIED - PNTL, GEN2, PPOTLY, LIMIT, SCATT, MULT, SING FCRJ01
C HAD OVERFLOWS OR DIVIDE CHECKS BYPASSED OR PRECISION LOSS FIXED. FCRJ01
C HOROLG, EDIT, EDIT2 TIMING WAS REWRITTEN, FIRST STEP TOWARD DJMP-RESTART FCRJ01
C SMS PROTOCOL: APPEND COMMENT DESCRIBING #DSET PURPOSE FCRJ01
C 77:10 FCRJ01: ADJUST SOURCE TO PASS FTMX0522, CORR EDIT, ECIT2 DUMP FCRJ01
C XCR002: EXEC CORR TO GIVE NORMAL TERM & FIX TIMING USING TLFTS XCRJ02
C DRJ003: DUMP AFTER INPUT & SAMPLE, RESTART NEXT RUN FROM THERE DRJ003
C 78:1 ADC004: ANG DISTRIBS: CUT DEADWOOD & COMMONS, DOUBLED MODE SCATT ADCJ04
C 78:2 XTC005: EXEC TIME CORR IN ELECT, CHG CLASS XTCJ05
C 78:3 HSR006: HAST STEP ENERGY SELECT & BNOPY DIST CK CHG IN ELTRAN HSRJ06
C UPC007: UNDERFLOW & PRINTOUT CORR IN EDIT2 78:5 UPCJ07
C COR008: CALLS TO RND ROUTINES REDUCED 78:5 CORJ08
C UFC009: UNDERFLOW CORR IN SING & MULT 78:5 & 82:4 UFCJ09
C RCC010: RUN CONTINUATION CORRECTIONS JBL 82:5 RCCJ10
C CPA011: DOUBLE PRECISION ACCUMULATORS & COMMON CLEAN UP 82:9 CPAJ11
C LEU012 LOW ENERGY UPDATE FROM SANDYL II, NEW FT09, PJTL ERR 83:4 LEUJ12

```

Figure 1. Comment section of routine TMAPQ

6600 overlay supervision was performed by CHAIN, under ASC overlay is transparent. Thus under ASC, CHAIN was made an entry point of TMAPQ, to allow special handling of nonstandard sequencing of overlays. Also under ASC a degree of buffering of input and output is transparent, so all calls to buffering routines have been eliminated, and it turns out this confers quite comparable input/output (I/O) efficiency. Under 6600 the timing routine HOROLOG is a library routine not furnished in the source listing. Since its exact procedure was not known, it was at first eliminated and then later HOROLG was added for local needs. Although SANDYL has its own protocol for error reporting and termination, it was found useful in a few instances to terminate by calling ERROR which does a traceback of subroutine calls and prints any predetermined areas in memory. While it is difficult to completely differentiate the contributions of the RCC staff and the author in phase 2, generally when routines were missing RCC simply provided dummies, leaving to the author the restoration of omitted functions. Changes extending capabilities were due to the author. Thus early in the translation process local debugging aids were invoked and two of these have remained in the local version since they are useful even during productive runs. They both have initializing calls in TMAPQ. A call to R\$STOP sets up an abnormal termination trap which flushes output buffers, provides a traceback, and any special user anticipated dumps. Also a call to R\$PCHA initializes the local patch utility which allows patching overlays. Use of these utilities in productive runs will be mentioned in the next section. While changes made by RCC were quite conservative, those made by the author were also restrained. Only two changes have been made which affect Monte Carlo function and these the author believes to be correction of defects. (See discussion of modset HSR006 below.) Some changes have been made for ease of use. Thus for long Monte Carlo runs one desires a dump-restart capability. This capability can often avoid loss of the time investment when a run terminates prematurely. Writing key common blocks out at propitious times allows one to begin a new run at that point later. The decision early on to retain the overlay structure (only because it presented no problem -- even without overlays the ASC memory would be only two-thirds filled) already facilitated a primitive dump - restart. Thus it is quite natural to copy all necessary input blocks at the completion of input phase and to again copy all necessary blocks at the termination of the random walk phase. As an example of restraint, i.e., requiring changes to adhere to translation and ease of use guidelines, the following will be noted. In both original SANDYL and in SANDYL II, there exist sections of dead code, i.e., sections unreachable under any known option. Generally these sections were not disturbed because if there was at one time reason for their presence, a section could become useful again when some inactive function was activated. It needs to be mentioned also that not all original functions have been faithfully translated or verified. Thus the photon-only random walk module has never been run here. No doubt it could be activated, because parallel changes to those made in segment 5 have usually been applied, but without investigation no guess will be offered as to how easy this might be. One impediment is that a sample problem is not available to verify this segment.

After RCC completed phase 1 and began work toward a link, their corrected source was turned over to the author late in 1976. By February 1977 a link with overlays had been completed. From then until August when the sample problem was completed for the first time, the debugging procedure was largely the following. An execution of the sample problem was made to a fatal error point, perhaps a divide check or a floating point overflow, and the error was debugged, often requiring analysis of a post mortem dump. As progress was made, the error point moved later into the run until in August a normal termination occurred. Also at various points in this process, comparison was made with the sample problem detailed input printout, available by setting ICH = -1 in the card input. Thus, the comparison of the sample problem source spectrum was determined to be inadequate, so after first trying several short repairs, the relevant section in POTL was converted to double precision, which improved the comparison substantially. The photon and card input, segment 2, was completed in April 1977 and progress halted due to a problem with the electron data tape. This was resolved by A. Namenson and the input phase was completed first in July. Before attempting execution of the random walk module it was necessary to link a random number routine. In previous Monte Carlo work on the ASC, a random number generator for 32 bit machines, RANDUL, was adapted from the SHARE library and thoroughly tested.⁵ This routine was renamed RNFL for use with SANDYL. Also the SANDYL provision of an option to select among a few predetermined beginning random numbers was replaced by a provision to input a beginning random number and to print out the current random number at the end of each sample. Relatively little work was required to debug the random walk segment. While this was a significant point, there had been mild indications of problems which had been bypassed during the march of progress. In the electron data input it had been noted in passing that certain Legendre coefficients used to produce composite angular distributions showed strong deviations when compared to the 6600 sample problem run. This was ignored at the time because consequent data showed much smaller deviations. Also in the single complete execution it had been noted that the ratio of the random walk central processor (CP) time to input CP time was more than an order of magnitude larger for the ASC than for the 6600. Because there was a plethora of potential reasons for this, effort turned in another direction.

At this point the sample problem had been run and the comparison of results was reasonable, physically. The output of a Monte Carlo code has statistical variation which can be reduced by increasing the number of histories. SANDYL is set up to run histories in samples (a physically complete batch), so that the sample to sample variation, actually standard deviation, can be used as a measure of statistical error. The sample problem only called for 3 samples, so there was only one tally bin in the output with less than 10% error which could provide a reasonably critical comparison. Thus by "reasonable, physically" one means the comparison was smaller than the standard deviation. It can be seen that this is actually a weak test. Because the respective library random number generators depend on the machine word length, identical random number sequences of sufficient length to run the sample problem could not be obtained, so identical

comparison was not an attainable objective. While it would have been possible at this point then to declare verification complete, the weakly negative indications mentioned above, particularly the second, were interpreted as requiring further checking.

During the fall of 1977, the foregoing sequence was recapitulated by setting the source up on SMS. The input source program listing (SPL) included those compile and execute corrections mentioned above as well as those provided by RCC and was modified to provide the deck structure expected by SMS. To avoid duplication, common blocks were entered once only, as decks to be called by routine decks. This way of listing source gives a count of 9663 lines, not including three routines (BRANG, BRAX1 and SCREEN) which are never called. SMS is basically a line editor and changes are made by entering modsets consisting of edit instructions and lines to be inserted or to replace lines in the SPL or earlier modsets. Modset names, a brief description and the date the modset was first applied are given in Fig. 1. For the rest of this section changes will be described as effects of a modset. Thus the first modset, FCRO01, consists of FORTRAN corrections, i.e., corrections responding to the optimizing compiler diagnostics. Previously only the fast compiler was used and the optimizing compiler found some errors not caught by the fast compiler. Also there is a correction to the dump arrangement in EDIT2. The second modset, XCRO02, was somewhat inaccurately termed "execute corrections," because rather than responding to execute diagnostics, it is mostly a revision of the internal timing to utilize the ASC library function TLFT\$ which returns the CP seconds left on the current job phase, originally determined by JSL statements. As noted in the comments of Figure 1, a timing method had already been provided in HOROLG. This provided output of the input time, each sample time, and a total run time. Use of TLFT\$ allowed the use of remaining job time as a key to invoke dump-restart, the object of the next modset. It might be thought that more resources were devoted to the dump-restart provision than were warranted. However, during the timing runs which would start soon, even the simplest part of this provision would amply justify itself. Having a dump of the input avoided rerunning the input phase during every random walk phase debug run. For the sample problem the input phase was taking 70s (seconds) and costing some \$19. The comment on overflows and divide checks in Fig. 1 is so brief as to be inaccurate. Only underflows were bypassed. However the ASC allows initializing the entire load area of memory with either zeroes or "indefinite forms," equivalent to a floating point infinity. Use of the latter was made as a diagnostic, causing an overflow trap whenever the code assumed a zero initialization. It was necessary to examine every one of these cases and make a definite internal resolution to avoid the possibility of occasionally invoking different initialization than was used on the 6600. The usual resolution was to provide zeroing of the relevant array.

The third modset, DRS003, completed the dump restart facility. Where the last modset provided for a new input parameter, ITMN, an estimate of the sample CP time in seconds, and provided in HOROLG for comparison of ITMN with the remaining job time after each sample, provision was now made to terminate the run normally if there were too little time to complete a sample. Taking a dump of appropriate common blocks was made part of normal termination. The estimation of run time for SANDYL can only be based on

experience and often there is no relevant comparison run. Thus before this the usual case for an ASC run was that the JSL required a specification of maximum run time in order to determine job class which affected turn around time. Also the run input requires a request for a certain number of samples, each executing a fixed number of histories. Thus if the JSL time estimate was short, the system terminated the job abruptly and the entire run to that point was of little or no value, lacking final answers and a statistical summary. This modset then provided for the input of a dump, using as key the previously disused parameter N2 on the first input card. Reading of this first card was removed from POTL and put into TMAPQ. Then if N2 is set to 4, TMAPQ reads a second card and goes directly to POT5 to fill common arrays from the previous dump on logical unit 25 or 26. It was necessary to provide two logical units because the input dump is shorter than the random walk dump. The second card allows the user to change NSORS and IB. Setting NSORS = 0 has been used as a signal to terminate after input and give an input dump. Also it is used by HOROLG to cause termination when there is less than ITMN seconds left. Thus in either case it must be reset. Also if one merely wants more histories for better statistics it is most convenient just to extend NSORS. The second parameter, IB is set to 1 if accumulators are to be initialized and to 2 if not, i.e., if the restart is intended to continue from the same point. Provision was also made to print the final random number. These provisions can be even more time saving when it is noted that a number of input constants are not used during input phase. These constants may therefore be patched for change or correction at random walk time. These provisions were verified by comparing the final sample of the sample problem dumped after two samples and restarted, with that of an uninterrupted run. The two were identical.

As mentioned above there were discrepancies in the comparison of the cross-section printouts for the sample problem the seriousness of which was difficult to evaluate. These problems occurred in the electron data input segment in routines combining various constituent angular distributions into a composite for each material. The method being used was convolution using Legendre coefficients. While some of the constituent coefficients were grossly in error, a first moment of the final distribution, COSAV, the average scattering cosine, was much less discrepant. The latter was used as main criterion in the attempt to improve the comparison. After several attempts to get improvement by less comprehensive means, all calculations of routine SCATT, routines called by SCATT, and contributory calculations above SCATT, including some in DATPAC, RANGE, QPOL2 and routines calling QPOL2 were made double precision. This was the main purpose of modset ADC004 and it did achieve significant improvement in COSAV without, however, removing all discrepancies in the Legendre coefficients. Since this involved manipulation of several common blocks, use of common blocks was improved. When no variable of a common block was used in a routine, the block was removed from that routine. In some cases when only a few variables from a block were used, the block was removed and the variables were passed as arguments. Also in this segment are a number of sections of dead code, remnants of disused options. Some of these had also been removed in SANDYL II, so when such sections were encountered they were removed without trying to be comprehensive.

During these manipulations, timings for the complete sample problem had been under observation. While the switch to double precision had increased input phase time by 9s, the prior timing was deemed to be a more representative comparison and is shown in Table I. Neither machine was purportedly including I/O in its CP time count but this sort of claim raises more questions than it answers. The ratios differ by a factor of 33. While this can lead to a detailed discussion of relative machine I/O bound and

Table I. Sample Problem CP Times
(Seconds)

	<u>6600</u>	<u>ASC</u>
Input	109	70
Random Walk	13	280*

*later changed, see text

compute bound efficiencies, such a large factor seemed most likely due to a translation error. An attempt to track down this problem was begun by timing various random walk routines using the ASC utility Z\$TIMR. Because comparison with 6600 in this detail was not available, it was necessary to surmise physical function. This was aided by use of the JDEBUG = 1 printout. After familiarization appeared the code was spending too much time in some very low energy and thus non-contributory processes. Finally a variable name was found to be misspelled in ELECT. When this was fixed the 280 of Table I became 17. This event was accepted as completing the verification phase of translation.

The changes made in modset HSR006 amount to changes in Monte Carlo selection procedure for electrons in routine ELTRAN. However it seems likely they give no detectable change in results and no change in results was found in comparison tests. When a sampling of the straggling distribution failed by producing a negative energy, the previous procedure was to set the energy to zero. This was changed to give the same energy as before sampling on the theory that the old way was biased. Clearly only very low energy electrons are affected at all. Also in the test whether an electron had energy to reach a boundary, made only in the IBTST = 0 option, the energy index was fixed to update each transport subcycle. This is also likely to have little result because if the electron had insufficient energy to reach a boundary in the first step, it probably always would.

The purpose of modset UPC007 is to modify EDIT2 to improve printout and to avoid underflow, a condition resulting in a trap on the ASC unless it is disabled. Because underflow may also be an indication of poor coding, each case needs to be examined and it is preferable not to disable this trap, either globally or throughout an entire routine. The printout improvement provides two numbers of possible diagnostic value, the total energy deposited in all zones and the net electron deposit in all zones.

Prior to the changes in modset COR008, the random number generator RNFL was modified by providing the following entry points: EXPRNF, SFLRAF, AZIRN, GTISO, GTIS1, DIRCS, DIRC1, FISRN and APSRN. These each generate random numbers sampled from a specific distribution needed by SANDYL. The modset then deleted in line coding accomplishing these selections and replaced it by a call to the relevant entry point. This is a significant improvement in efficiency for two reasons. The first is that a single call furnishes one or two random numbers and replaces a cycle of calls to RNFL which all rejection techniques require. This is valuable because any subroutine call has a significant overhead of nonproductive CP time. Secondly, in the case of DIRCS, which provides particle direction cosines following a scattering and is much used by SANDYL, a cascade of calls is reduced to one call. The effect of these changes was to reduce the sample problem random walk execution time from the 17s noted above to 14s. At about this time the ASC optimizing compiler was used on the random walk module routines to optimize scalar code only, which in turn reduced random walk time to 9s. While the random number generator could be further improved by writing it in machine language, the above effort has nearly removed the incentive. For the sample problem, not a particularly heavy user of random numbers, the random walk phase only spends about one percent of its time in the random number generator.

Underflow corrections are the object of modset UFC009 and are made by locally disabling the underflow trap. In two cases it was determined the underflow could be allowed with no consequence. In a third case this could not be determined and an error print followed by a call to ERROR was provided.

The original SANDYL had a provision for run continuations when the source spectrum was monoenergetic and provided for a change in the source energy. The object of modset RCC010 was to activate this facility.

In translating from a 64 bit to a 32 bit machine, one expects difficulties related to the reduced precision. In Monte Carlo problems one would not try for precision better than a few percent in the final results without the use of special variance reduction techniques. Thus outside of the two cases above where double precision was used, production run precision loss was not encountered until it was necessary in one problem to run as many histories as possible to improve statistics. This case encountered precision loss in tally bins, evidenced by a continued deterioration in a particular result as more histories were run. The effect is well known and easily demonstrated on 32 bit machines. Adding a floating point number whose fraction does not terminate in less than 6 hexadecimal figures, 10^5 times gives only one or two correct figures and adding it 10^6 times results in complete loss of precision. Later 32 bit machines have improved on this somewhat but have not eliminated the effect. The cure is to make the tally bins double precision and this is the purpose of DPA011. However deciding that this would be sufficient required an extensive survey of nearly all common block variables. As a result of this survey an extensive clean up of common block structure was also made. Thus when it was discovered that a common block variable was not being passed for later use, the variable was demoted to local. Also a couple of routines in the random walk module had common blocks eliminated and replaced by passed

arguments. In order to compensate for the additional memory requirement for double precision, a number of array dimensions were reduced. In particular the number of edit zones allowed was reduced from 30 to 20 and the product of the number of time steps and the number of inside zones was reduced from 110 to 55. Checks were provided in POTL to stop if such limits would be exceeded. Actually this effort overcompensated slightly, reducing memory demand about 5K words. Finally certain internal tally bins were printed out for diagnostic purposes. These were labelled: source, boundary death, production less absorption, energy death, and net sum. These modifications were verified by comparing a sample problem run with that for the previous modset and the two were identical. The double precision modifications were then verified by rerunning a case where precision loss had been noted and it was not present.

At about this same time a problem of the SMS editor was discovered. It didn't give a unique line sequence number in certain cases involving changes to call decks. As a result it was necessary to separate call deck modifications into their own modsets. This gave rise to modsets CDM004 and CDM011, the effects of which have already been described, in the modset texts for the same serial numbers, 4 and 11.

The intention of the latest modset at this writing, LEU012, was to incorporate the low energy update of SANDYL II. Since the changes in this update were not extensive, it was obvious that making only the changes was much less work than a complete new translation would have been. This update consists of adding two new routines GOUDS and GINTP and two common blocks in the DATPAC segment and adding data to and changing the format of the electron data tape. Where the 6600 used mass storage for intermediate steps, the ASC was set up to use logical unit 4, ordinarily on disc. The copy of SANDYL II provided was in UPDATE format with binary files. While source files were recoverable using a nearby 6600, the electron data tape was not. Thus the additional data was obtained from the authors of this correction³ by an indirect route. Verification consisted in comparisons with the detailed cross-sections for the sample problem from SANDYL II. For gold at low energy where the elastic scattering correction has its greatest impact, the correction changed 4 out of 5 printed figures in COSAV and the comparison showed all 5 figures identical at this point. This update resulted in an extension of memory demand by about 1K and in an extension of input phase time of about 7s so that total input time without detailed cross section printout is now 64s. Also included in LEU012 was a correction to POTL concerning multiple sphere or cylinder (ICOOD = 9 or 10) sources.

III. User Information

For convenience a copy of the job activity file for a normal run is shown in Figure 2. The first part is job deck and macro expansion cards. The latter are marked by asterisks and may be ignored for present purposes. Those numbered 1 through 12 are the job deck as submitted. This run was for the sample problem, which, though undemanding of computer resources, is adequate as an example. Prior to this job the four files under the SN node, at nodes PDT, E2T, SZE and LMD had been uploaded to disk. Had this not been done, the file assignments for FT08F001, FT09F001, D1 and SYS.LMOD would have needed an additional disk space of 30, 10, 1 and 5 bands, respectively,

```

14:21:14 021A 0001 VERSION # 29 WAS ASSIGNED TO FILE SYS.WCRN.
0000      1 / JCR LANGWORTHVSPC.66175203,LANGJ1,OPT=(C,D),CAT=1
0004 JCR      **** / JCR. LANGWORTHVSPC.66175203,LANGJ1,CAT=1,OPT
0005      2 / LIMIT RAND=20,MIN=7
0006      3 / PD SN,USERCAT/066/R20/LANGJ1/SND
0007      4 / ASG FT05F001,SN/001,USE=SHR
0008      5 / ASG FT06F001,SN/E20,USE=SHR
0009      6 / ASG 01,SN/S2E,VERS=4,USE=SHR
000A      7 / ASG SYS.LMND,SN/LND,USE=SHR
000B      8 / FD FT25F001,RAND=1/5/1
000C      9 / FD FT26F001,RAND=1/6/1
0000     10 / FXOT DATA=01,CPTIME=9000,OPT=(T),ADDLEN=12K
0000 FXOT     **** / CHECKPT FMSYS #
0000 FXOT     **** / RENAME 01,FT05F001
000E FXOT     **** / FD FT06F001
000F FXOT     **** / YOT SYS.LMND,ADDLEN=12K,OPT=(T),CPTIME=9000
0010 FXOT     **** / RENAME FT05F001,01
0000 FXOT     **** / OPTFMSYS FT06F001,*,EXTD=SYS.LMND
0011 OPTFMSYS **** / FMSYS FT06F001,EXTD=SYS.LMND
0012      11 / CATV SN/IDP,ACNN=FT25F001
0013      12 /END E9J
14:21:17 0000 0002 JSC NORMAL TERMINATION ISIX0579
14:21:17 0492 0002 S9JSLT TERMINATED NORMALLY. TERM = 0.
14:21:37 021A 0007 VERSION # 0 WAS ASSIGNED TO FILE FT05F001.
14:21:37 021A 0008 VERSION # 0 WAS ASSIGNED TO FILE FT06F001.
14:21:37 021A 0009 VERSION # 4 WAS ASSIGNED TO FILE 01.
14:21:39 021A 000A VERSION # 0 WAS ASSIGNED TO FILE SYS.LMND.
14:21:41 00FD 000F FILE PATCHES DOES NOT EXIST.
14:21:42 00FD 000F FILE 01YSEG02 DOES NOT EXIST.
14:23:39 00FD 000F FILE 01YSEG04 DOES NOT EXIST.
14:27:15 00FD 000F FILE 01YSEG05 DOES NOT EXIST.
14:27:34 0000 000F STAB
14:27:34 0492 000F SYS.LMND TERMINATED NORMALLY. TERM = 0.
14:27:42 035F 0012 JCR LANGWORTHVSPC IN SC58 WAS CATALOGED A RESIDENT FILE.
14:27:42 0219 0012 FILE FT25F001 WAS CATALOGED AS VERSION # 0.
14:27:43 0338 FFFF JCR TERMINATED NORMALLY

```

Figure 2. Job activity file for a normal run

thus increasing the BAND specification on the LIMIT card. Also this run utilized 31 pages of memory (or 124K) which includes 8.2K of ADDMEM, some 16 bands of disk, and 275 pseudo-seconds, a combined measure of CP and I/O time. Nearly all of the I/O contribution to pseudo-seconds occurs during input phase. Note that extra ADDMEM was allowed because the demand varies with the number of logical units actually opened. Where the 11th card saves the dump from the input phase, FT25F001, the ordinary practice in a production run would be to save also the random walk dump, FT26F001. The second part of the job activity file is execute-time comments by the system. The four comment lines before the stop line may not be familiar. They arise because the patch utility was initialized and no patch files were submitted. Note that from these file names one can track each segment's execution.

Table II documents all changes to the card input file and only includes changes. It is noted that INRAN should be odd as this gives the longest cycle length (2^{30}). Users interested in the deficiencies of 32 bit congruential multiplication random number generators should consult Reference 5. The most notable deficiency is that groups of three random numbers are not very random. However SANDYL's demand for random numbers would seldom result in the application of a sequence of groups of three numbers to similar uses. Random comments will now be offered on unchanged card input in the same sequence as listed in Ref. 1.

- a. Any user invoking essential 3D geometry by use of X or Y planes is responsible for his own scoring. The code does not compute zone volumes for such cases so it cannot get dose internally. On the other hand use of cylindrical geometry allows setting a value for UNUT based on the source geometry to give dose in any desired units.
- b. No use of JM or JQ has been made locally and they may not be implemented.
- c. Ref. 1 calls different outputs "edits." Not all edit options have been explored and some may not be implemented.
- d. The quantities RTWi have a peculiar status, somewhere between physical and non-physical, i.e., weights. The author therefore does not know how to set these quantities to "no-weight" values and he doubts that the default settings are such. In any problem where shell ionizations were important, the user might find it necessary to compare with a better documented code such as ETRAN, even though SANDYL purports to have more detailed Auger cascade physics.
- e. One of the most difficult aspects of setting up input is the geometry. The author's experience has induced the following rule, not mentioned in Ref. 1. Any zone must be locally non-concave in every radial half-plane. In applying this rule, a spherical shell is allowed because even though one boundary is concave, there is no point that is an inside corner, i.e., a "locally concave"

Table II Changes to Input Cards

<u>Card</u>	<u>Columns (Format)</u>	<u>Variable</u>	<u>Description</u>
1	2(12)	N2	If not blank, input phase is by passed, control is passed to segment N2+1 after reading card 1A. Dump input must be on FT25 or FT26 (same unit it was written on). N2=4 is the only nonblank option now open.
(Use only if N2 not zero.)			
1A	1-4(14)	NSORS	After dump is read in, old NSORS is changed to this number.
	5-8(14)	IB	After dump is read in, old IB is changed to this number: 1: initialize to begin new random walk, 2: continue random walk from dump
2	57-60(14)	ITMN	Estimated time per sample in seconds. If remaining job time is less than ITMN at the end of any sample, the run terminates with a dump.
	63-64(12)	IM	(Same as before, position changed.)
3	1-2(12)	ISPEC	(Same as before, format changed.)
	3-10(78)	INRAN	Initial random number (odd hex).
	11-12(12)	ISENT	(Same as before, format changed.)

"point." Even though it's an abuse of language, this rule conveys a definite prohibition. While lots of wrong ways of doing the geometry will cause internal error messages, there are some that don't, and the resulting error could be the size of a statistical error and thus nearly undetectable. Thus in the geometry there is no substitute for being right.

- f. Because it is affected by machine precision, the setting of FUDGE, a boundary offset distance, needs to be mentioned. While observing the minimum mentioned in Ref. 1, it cannot be less than 10^{-7} times the maximum geometry dimension and preferably not less than 10^{-4} of this. When this value is too small, the code can't decide the zone assignment and stops with error numbers 260 or 280. Since the original default of 10^{-7} cm has not been changed, FUDGE must nearly always be set.
- g. Some variance reduction settings are available, one of which is PTCZ. In Monte Carlo, "no free lunch" becomes "reducing variance in one zone usually causes variance increase in some other zone." Thus a conservative policy is to start with PTCZ = 0 (no weight), gradually turning on weights while making sure the results are understandable.
- h. Concerning FNS, the source spectrum, it is usually possible to use the "total energy deposited" printout to check whether FNS has the correct format as required by ISPEC. If it does and if zone arrangement is such that a sum over zones gives total energy, then this printout should be spectrum average energy, in some ISPEC cases identically and in others within statistical error. -

As mentioned above SANDYL anticipates some common errors by providing standard error returns identified by error numbers. A partial list of error numbers with a brief identification occurs at the end of EDIT2. Errors not on this list must be traced back to the originating routine and this may be necessary anyway. These errors are primarily for debugging the problem input. Not all of them are fatal errors, some are just information or are not serious if only a few occur. Another class of potential errors has been mentioned, those calling ERROR, necessarily fatal. The occurrence of such an error should be reported to the author for a complete analysis. The author believes that none will occur. There is a third class of errors related to use of a machine of reduced precision. Occasionally a run will be stopped by an arithmetic exception. Any exception except underflow should be considered a bug and the cause eliminated. Underflow should be treated the same way even though it is often resolved by locally disabling the trap and even though there is often a simple work-around. Tracing such problems helps to avoid future annoying stops. The work-around for those with immediate production goals is simply to change the INRAN and resubmit the job. It may seem strange that bugs of this nature still exist. However, the high precision of the original machine allowed original coders and users to ignore poor coding practices, avoidance of which would have required a strict coding standard unusual even today. So far as our local version is concerned, our extensive experience has encountered a number of these bugs which were each eliminated, but there is no practical way to comb the entire source for such problems.

The various options available under dump-restart should by now be obvious to the user. Use of the patch utility in conjunction extends the usefulness of this facility greatly. There are a number of run constants which the input phase merely passes on to the random walk, making no essential use of them. A few of these have internal representation changed by the input phase, but most are passed on just as received. These variables may be patched at random walk execution time, either to correct an error or to change the run. As an example consider the exhibit in Fig. 3, part of the DEBUG\$\$ file created by the patch utility in response to the submission of a patch file. The patch file consisted of the cards on the second and fourth printed lines of Fig. 3, beginning with the patch commands I and L for "insert" and "last," and was submitted in a start/stop file inserted in the job deck before the FXQT card. The start JSL card reads:

```
/ START ACNM=OLYSEG05,FORG=DS
```

When the patch utility encounters an OLYSEG05 file, it executes the patch commands at the overlay time for segment 5. The insert command results in a two word patch at the absolute location given by the first argument. These two words are machine instructions to jump to the insert table. The insert command also places machine code in the insert table consisting of the two over-patched instructions, the remaining insert card arguments, and a return jump to the location following the patch. The first argument is determined by use of the link listing and an object listing for the target routine, in this case POT5. This patch may be inserted at any location of POT5 following the dump read-in that must be executed once only, barring jumps or return locations. The insert code of Fig. 3 consists of two pairs each having the same action. The arithmetic register 0 is loaded with the right half of the instruction and is then stored to the object location. In this case the result is that JDEBUG is set to 1 and NSAMP is set to 10. Offsets and base registers giving these locations are best determined by example from the object code near the insert location. Of course one must choose an arithmetic register that is not in use.

```
* THIS EXECUTION IS USING PIPES - 0 - 1
* I 1380.54000001.2400703B.5400000A.240070B1
*      CODE PLACED IN INSERT TABLE AT LOCATION      0001965A
* L
* THIS EXECUTION IS USING PIPES - 0 - 1
```

Figure 3. A portion of a DEBUG\$\$ file

The SANDYL transport code lies in a spectrum of codes in terms of the complexity of geometric detail to which it applies and in terms of the degree to which the basic algorithms are based on applicable and documented physics. At one end of this spectrum one might mention ETRAN,² the physics of which is very well documented but the geometry of which applies only to layers of a single material. At the other end, BRANDE⁶ could be mentioned using a straight ahead approximation with little more than a blessing of physics pedigree but using sector analysis applicable to a geometry of any degree of complexity, limited only by the effort one wishes to make. To be safe one establishes the applicability of BRANDE to each problem spectrum by calibration, i.e., comparison with a more standard code. SANDYL lies midway in this spectrum and in fact we have made several efforts to compare it with ZTRAN, a close relative of ETRAN able to handle slabs of different materials. Such efforts have not had the broad approach required for a critical comparison but have been aimed at answering the needs of specific productive efforts and therefore will only be mentioned here informally. Using a common fission beta spectrum to get dose in an aluminum slab, we find SANDYL on the low side of ZTRAN, about 15 percent near 1 g cm^{-2} and probably getting worse near 2 g cm^{-2} . The latter is close to where bremsstrahlung begins to dominate. However because there is prospect of an improvement⁴ in the bremsstrahlung contribution (increasing it) in the near future, incentive for critically investigating this error is somewhat vitiated.

References

1. H.M. Colbert, "SANDYL, A Computer Program for Calculating Combined Photon-Electron Transportation in Complex Systems," Sandia Laboratory SLL-74-12, May 1974.
2. Radiation Shielding Information Center (Oak Ridge National Laboratory) code package CCC-107 and M.J. Berger and S.M. Seltzer, National Bureau of Standards Report 9837 (1968).
3. L.G. Haggmark, C.J. MacCallum and M.E. Riley, Trans. Am. Nucl. Soc. 19, 471 (1974).
4. S.M. Seltzer and M.J. Berger, "Status of Electron Transport Cross Sections," National Bureau of Standards NBSIR 82-2572, Sept. 1982. PB 83-112128
5. B.T. Carruthers and G.H. Herling, "Tests of Random Number Generators for the ASC," Naval Research Laboratory Memorandum Report 3509, May 1977. (AD/A041-732)
6. A. Produced by J.F. Janni of Air Force Weapons Laboratory. For details and further references see J.B. Langworthy, "A Demonstration of Satellite Shielding," Naval Research Laboratory Memorandum Report 4275, July 1980 and J.B. Langworthy, "Satellite Shielding Using Sector B. Analysis," Naval Research Laboratory Memorandum Report 4577, August 1981. (6.A.) AD/A087 752 (6.B.) AD A102 236

END

FILMED