27

10-4-78

250 HT IS

MASTER

MASTER

UCRL-50400

Volume 17, Part A, Rev. 1

**An Integrated System for Production of Neutronics
and Photonics Computational Constants
Volume 17, Part A, Rev. 1**

**PROGRAM LINEAR (VERSION 78-1): LINEARIZE DATA
IN THE EVALUATED NUCLEAR DATA FILE/
VERSION B (ENDF/B) FORMAT**

D. E. Cullen

July 4, 1978

Work performed under the auspices of the U.S. Department of
Energy by the UCLLL under contract number W-7405-ENG-48

**LAWRENCE
LIVERMORE
LABORATORY**
University of California - Livermore

LAWRENCE LIVERMORE LABORATORY

University of California Livermore, California 94550

UCRL-50400 Volume 17, Part A, Rev. 1

***PROGRAM LINEAR (VERSION 78-1): LINEARIZE DATA
IN THE EVALUATED NUCLEAR DATA FILE/
VERSION B (ENDF/B) FORMAT***

D. E. Cullen

MS. date: July 4, 1978

Foreword

The total cross section probability (TCP) system is designed to process evaluated neutron cross section data into a continuous energy or multigroup form. In the following figure, we have described the function of each of the computer codes in the TCP system by contrasting the form of the data before and after code processing.

Contents

Abstract	1
1978 Edition	1
Introduction	1
Interpolation Laws	2
Effect on Integral Quantities	5
Thinning	5
Limitations and Assumptions	6
Requirements	7
Input Cards	7
Example Output Listing	8
Error Messages	11
Operation at LL	11
Conversion to Other Installations	13
Program Listing	14
Acknowledgments	30
References	31

PROGRAM LINEAR (VERSION 78-1): LINEARIZE DATA IN THE EVALUATED NUCLEAR DATA FILE/VERSION B (ENDF/B) FORMAT

Abstract

We have written a computer code called LINEAR that converts evaluated cross sections in the ENDF/B format into a form that can be defined by applying linear-linear interpolation to energy and cross section parameters. The code also thins those sections already in that form. The main advantage of the code is that it eliminates the need for any subsequent code to consider anything but linear-linear data.

1978 Edition

This edition of the program is designed for operation at Lawrence Livermore Laboratory on a CDC-7600 computer, but it can easily be converted for use at other installations. This edition of the code includes improvements and corrections suggested by numerous users of earlier editions. In particular the code has been changed to ease the use of double precision on shorter word length computers (e.g., IBM-360) and to allow more efficient use of the scratch file.

Introduction

The ENDF/B format is used to store nuclear cross section data.¹ Within the ENDF/B format, each section of tabulated (File 3) data is represented by a table of energy vs cross section and any one of the five possible interpolation laws listed below:

1. Histogram - constant cross section between two energies
2. Linear in energy - linear in cross section
3. Log in energy - linear in cross section
4. Linear in energy - log in cross section
5. Log in energy - log in cross section

In this report we describe how the LINEAR program replaces each section of file 3 data with a new energy vs cross section table in which the data only obey the second interpolation law (the law imposes a linear-linear relation-

ship between any two tabulated points). This replacement process is called linearizing.¹ Figure 1 illustrates the importance of linearizing ENDF/B cross sections. The dashed curve in the figure represents cross-section data that have been reduced to linear-linear form; the solid curve represents data that have been plotted assuming linear-linear behavior.

Interpolation Laws

In the previous section we listed the five interpolation laws. The first law, histogram, is converted to linear-linear by substituting two points for each initial point. Interpolation law 2 is already linear-linear and need not

Fig. 1. Comparison of a $1/v$ cross section that has been reduced to linear-linear form (solid curve) with a cross section that has been plotted assuming linear-linear behavior (dashed curve) between 0.01 and 100 eV.

be considered further in the following discussion. For interpolation laws 3 through 5, the program starts from the tabulated values and uses an interval halving algorithm.² The interval is divided in half until the value at the middle of the interval can be approximated by linear-linear interpolation from the ends of the interval to within some acceptable accuracy. In halving the interval, the midpoint is defined in a linear sense $\left(\frac{1}{2}(E_K + E_{K+1})\right)$ for interpolation law 4 and in a log sense $(\sqrt{E_K E_{K+1}})$ for interpolation laws 3 and 5.

The convergence of the halving algorithm can be demonstrated by considering the difference between the exact interpolation law and linear-linear interpolation.* Using the same interval endpoints, we find that the interpolation laws are[†]:

$$\begin{aligned}
 2 \quad Y &= Y_K + (X - X_K)(Y_{K+1} - Y_K) / (X_{K+1} - X_K) = Y_K + A_K(X - X_K) \\
 3 \quad Y &= Y_K + \ln(X / X_K)(Y_{K+1} - Y_K) / \ln(X_{K+1} / X_K) = Y_K + B_K \ln(X / X_K) \\
 4 \quad Y &= Y_K \exp\{(X - X_K) \ln(Y_{K+1} / Y_K) / (X_{K+1} - X_K)\} = Y_K \exp[C_K(X - X_K)] \\
 5 \quad Y &= Y_K \exp\{\ln(X / X_K) \ln(Y_{K+1} / Y_K) / \ln(X_{K+1} / X_K)\} = Y_K \left(\frac{X}{X_K}\right)^{D_K} \quad (1)
 \end{aligned}$$

The differences are:

$$\begin{aligned}
 3 \quad A_K &= B_K \ln(X / X_K) \\
 4 \quad Y_K \left(1 - \exp\{C_K(X - X_K)\}\right) &+ A_K(X - X_K) \\
 5 \quad Y_K \left(1 - \left(\frac{X}{X_K}\right)^{D_K}\right) &+ A_K(X - X_K) \quad (2)
 \end{aligned}$$

We obtain the change in the difference by differentiating with respect to X

$$\begin{aligned}
 3 \quad A_K &= \frac{B_K}{X} \\
 4 \quad &= C_K Y_K \exp\{C_K(X - X_K)\} + A_K \\
 5 \quad &= D_K Y_K \left(\frac{X}{X_K}\right)^{D_K-1} + A_K \quad (3)
 \end{aligned}$$

The point of maximum difference is found by setting these expressions equal to zero:

*The procedure for checking convergence of the halving algorithm was suggested by Charles R. Weisbin, Oak Ridge National Laboratory.

†The numbers to the left of the equations correspond to the interpolation laws listed on p. 1.

$$3 \quad X = \frac{B_K}{A_K}$$

$$4 \quad X = X_K + \frac{1}{C_K} \ln \left(\frac{A_K}{C_K Y_K} \right)$$

$$5 \quad X = X_K \left(\frac{A_K}{D_K Y_K} \right)^{1/D_K-1} \quad (4)$$

Note that in each case there is only one point of maximum difference. The values of the difference at the point of maximum difference are:

$$3 \quad B_K - A_K X_K - B_K \ln \left(\frac{B_K}{A_K X_K} \right)$$

$$4 \quad Y_K \left(1 - \frac{A_K}{C_K Y_K} \right) + \frac{A_K}{C_K} \ln \left(\frac{A_K}{C_K Y_K} \right)$$

$$5 \quad Y_K \left[1 - \left(\frac{1}{X_K} \right)^{D_K} \left(\frac{A_K}{D_K Y_K} \right)^{D_K/D_K-1} \right] + A_K X_K \left(\frac{A_K}{D_K Y_K} \right)^{1/D_K-1} - 1 \quad (5)$$

By expanding and studying convergence as the length of the interval decreases (i.e., $E_K \rightarrow E_{K+1}$), it is possible to demonstrate that all of the above differences approach zero as fast as $(E_{K+1} - E_K)$, except in the following trivial cases:

$$(A) \quad Y_{K+1} = E_K, \quad Y_{K+1} \neq Y_K.$$

This is a cross section discontinuity which can be treated as a histogram (interpolation law 1).

$$(B) \quad Y_{K+1} = Y_K.$$

This is a constant cross section which is already linear-linear.

$$(C) \quad D_K = 0 \text{ or } 1.$$

This is already linear-linear; therefore, the interval halving algorithm will converge in all cases.

In principle, the above algorithm could have been used in program LINEAR instead of interval halving. (Both algorithms converge to roughly the same number of points.) However, comparison of the two algorithms in preliminary versions of this program indicated that interval halving is simpler and faster.

Note that although the halving algorithm works for all possible representations within the ENDF/B system, it is not a general method of reducing an arbitrary function $f(x)$ to linear-linear tabulated form. An example of a case that does not converge is:

$$f(x) = x^2 : x \in (0, 1).$$

Effect on Integral Quantities

If we start from any exactly known cross section $\sigma(E)$ (described by one or more of the five possible ENDF/B interpolation laws), we can construct an approximate cross section $\tilde{\sigma}(E)$, which can be linearly interpolated. In addition, we can guarantee that $\tilde{\sigma}(E)$ is everywhere within ϵ of $\sigma(E)$:

$$(1 - \epsilon) \sigma(E) \leq \tilde{\sigma}(E) \leq (1 + \epsilon) \sigma(E).$$

When we multiply by some weighting spectrum and integrate, we find:

$$(1 - \epsilon) \int_{E_1}^{E_2} \sigma(E) \phi(E) dE \leq \int_{E_1}^{E_2} \tilde{\sigma}(E) \phi(E) dE \leq (1 + \epsilon) \int_{E_1}^{E_2} \sigma(E) \phi(E) dE.$$

Similarly, by taking the reciprocal we can find,

$$\frac{1}{(1 + \epsilon)} \int_{E_1}^{E_2} \frac{\phi(E)}{\sigma(E)} dE \leq \int_{E_1}^{E_2} \frac{\phi(E)}{\tilde{\sigma}(E)} dE \leq \frac{1}{(1 - \epsilon)} \int_{E_1}^{E_2} \frac{\phi(E)}{\sigma(E)} dE.$$

Therefore, the integral quantities derived by using the cross section that can be linearly interpolated are always within ϵ of the exact integrals, and - because of cross cancellation in the differences - generally much closer than ϵ .

Thinning

The LINEAR program uses a multipoint, fractional-error thinning algorithm to minimize the size of each cross section table; the program eliminates those points that are not required. The algorithm starts from three successive energy points: 1, 2 and 3. Using linear-linear interpolation from end points 1 and 3, we can determine if point 2 can be approximated to within some acceptable accuracy. If it can, a fourth point, 4, is included. Using linear-linear interpolation from end points 1 and 4, we can determine if points 2 and 3 can be approximated to within some acceptable accuracy. In this manner the

algorithm includes an increasing number of points until one or more of the intermediate points cannot be accurately approximated by linear-linear interpolation. To illustrate what action is taken at this point, consider the case in which the thinning algorithm is using N points. Assume that points 2, 3, ... (N - 1) can be adequately approximated by linear-linear interpolation between points 1 and N. Under these circumstances, the (N + 1)th point will be added to the algorithm and we will then be able to determine if points 2, 3, ... N can be adequately approximated by linear-linear interpolation between points 1 and (N + 1). If we assume, in this case, that one or more points fail the test, then we have to find out which points can be thinned and which must be kept. Since, in the preceding test, points, 2, 3, ... (N - 1) could be adequately approximated by interpolation between points 1 and N, it is only necessary to keep points 1 and N, regardless of which point or points fail the test when (N + 1) points are used. The thinning algorithm then restarts with the next three points: N, (N + 1), and (N + 2).

The intermediate values are approximated by linear-linear interpolation from the ends of the interval:

$$Y_K^* = Y_1 + (X_K - X_1)(Y_N - Y_1) / (X_N - X_1), \quad (6)$$

where K = 2, 3, ... (N - 1). The error is then defined as the difference between the tabulated value Y_K and the value obtained by interpolating from the ends of interval Y_K^* .

$$ERR_K = |Y_K - Y_K^*|. \quad (7)$$

The error is considered to be acceptable if it does not exceed some maximum fraction of the tabulated value:

$$ERR_K \leq |\epsilon_{MAX} Y_K|. \quad (8)$$

Limitations and Assumptions

The LINEAR program only uses the ENDF/B BCD format (as opposed to binary).² It copies all sections, except file 3, as hollerith. Thus, the program may be used on data in the ENDF/B - I, II, III, IV or V format, since a section of file 3 data is identical in all ENDF/B versions. The program also nonselectively linearizes all sections of file 3 data on the original ENDF/B format tape (i.e., all sections between the tape label card and the tape end card).¹ This program does not require the evaluations to be in MAT¹ order; e.g., they may be in any order.

Because the program reads and linearizes the data a page at a time, there is no limit to the number of points in any section (i.e., each reaction may be described by any number of points). The program is written entirely in FORTRAN-IV and is designed to be easily implemented on any moderately sized computer.

Requirements

The I/O units are defined as follows:

<u>Filename*</u>	<u>Unit</u>	<u>Description</u>
INPUT	5	input card
OUTPUT	6	output report
ENDFIN	21	evaluated data to be linearized in the ENDF/B format
ENDFOUT	22	linearized evaluated data in the ENDF/B format
SCRATCH	23	scratch file

All read and write statements use variable I/O unit numbers. Thus, to convert to any other set of unit numbers, it is only necessary to redefine the unit numbers (see the program listing, p. 14).

When the program is run on the Livermore CDC-7600, it requires 14 000 words of small core memory and 6 000 words of large core memory (I/O buffers). It uses 0.6 min to linearize ENDF/B version IV tape 409 to 0.1% accuracy (28 970 points input and 21 994 points output).

Input Cards

The program requires a single input card to define the allowable fractional error:

<u>Card</u>	<u>Column</u>	<u>Format</u>	<u>Description</u>
1	1-11	E11.4	fractional convergence criteria

To linearize the data to within 0.1%, the following input card is required:

1.0000E-03

* The names in the Filename column only apply to the program used at Livermore. Disk files by these names are automatically assumed by the program.

Example Output Listing

The following listing is derived by linearizing ENDF/B, tape 409.

LINEARIZE ENDF/B CROSS SECTIONS LINPAR 78-11

THINNING CRITERIA-- 1.00000- 3

APR 148E1

ENDF/B-1, TAPE 409

29-AUG-74

409

Z	A	MAT	MT	KFLVIN	Q-VALUE	POINTS	
						EV	IN OUT
25055	1197	1	0.	+ 0 0.	+ 0	3390	2649
25055	1197	2	0.	+ 0 0.	+ 0	3202	2610
25055	1197	3	0.	+ 0 0.	+ 0	2823	154
25055	1197	4	0.	+ 0 -1.25800+ 5		101	83
25055	1197	16	0.	+ 0 -1.02250+ 7		26	301
25055	1197	17	0.	+ 0 -1.91680+ 7		3	3
25055	1197	22	0.	+ 0 -7.93060+ 6		11	179
25055	1197	28	0.	+ 0 -8.06330+ 6		14	88
25055	1197	51	0.	+ 0 -1.25800+ 5		75	53
25055	1197	52	0.	+ 0 -9.84000+ 5		58	44
25055	1197	53	0.	+ 0 -1.29200+ 6		47	27
25055	1197	54	0.	+ 0 -1.52800+ 6		37	
25055	1197	55	0.	+ 0 -1.88300+ 6		27	
25055	1197	91	0.	+ 0 -1.88300+ 6		38	35
25055	1197	102	0.	+ 0 7.27040+ 6		186	153
25055	1197	103	0.	+ 0 -1.80970+ 6		35	148
25055	1197	104	0.	+ 0 -5.83880+ 6		29	378
25055	1197	106	0.	+ 0 -1.23800+ 7		4	4
25055	1197	107	0.	+ 0 -6.21600+ 5		26	139
25055	1197	251	0.	+ 0 0.	+ 0	86	83
25055	1197	252	0.	+ 0 0.	+ 0	86	84
25055	1197	253	0.	+ 0 0.	+ 0	86	84

MAT TOTALS 10390 7367

92233	1260	1	0.	+ 0 0.	+ 0	3541	2669
92233	1260	2	0.	+ 0 0.	+ 0	764	126
92233	1260	4	0.	+ 0 -4.04000+ 4		56	47
92233	1260	16	0.	+ 0 -5.90400+ 6		20	14
92233	1260	17	0.	+ 0 -1.31930+ 7		5	4
92233	1260	18	0.	+ 0 1.90300+ 9		2809	2653
92233	1260	51	0.	+ 0 -4.04000+ 4		24	23
92233	1260	52	0.	+ 0 -9.20000+ 4		18	17
92233	1260	53	0.	+ 0 -3.12000+ 5		10	9
92233	1260	54	0.	+ 0 -3.40000+ 5		10	9
92233	1260	55	0.	+ 0 -3.99000+ 5		9	8
92233	1260	56	0.	+ 0 -4.16000+ 5		9	6
92233	1260	57	0.	+ 0 -4.61000+ 5		9	8
92233	1260	91	0.	+ 0 0.	+ 0	41	34
92233	1260	102	0.	+ 0 6.80000+ 6		2809	2517
92233	1260	251	0.	+ 0 0.	+ 0	43	36
92233	1260	252	0.	+ 0 0.	+ 0	43	37
92233	1260	253	0.	+ 0 0.	+ 0	43	37

MAT TOTALS 10263 8254

92238	1262	1	0.	+ 0 0.	+ 0	424	155
92238	1262	2	0.	+ 0 0.	+ 0	424	95
92238	1262	3	0.	+ 0 0.	+ 0	406	232
92238	1262	4	0.	+ 0 -4.50000+ 4		140	96

92238	1262	16	0.	+ 0	-6.04000+	6	33	30
92238	1262	17	0.	+ 0	-1.14600+	7	19	18
92238	1262	18	0.	+ 0	-1.94000+	8	93	77
92238	1262	19	0.	+ 0	-1.94000+	8	57	48
92238	1262	20	0.	+ 0	-1.94000+	8	23	19
92238	1262	21	0.	+ 0	-1.94000+	8	16	16
92238	1262	38	0.	+ 0	-1.94000+	8	3	3
92238	1262	51	0.	+ 0	-4.50000+	4	34	33
92238	1262	52	0.	+ 0	-1.46000+	5	21	18
92238	1262	53	0.	+ 0	-3.08000+	5	9	8
92238	1262	54	0.	+ 0	-6.80000+	5	18	18
92238	1262	55	0.	+ 0	-7.32000+	5	18	17
92238	1262	56	0.	+ 0	-8.27000+	5	15	14
92238	1262	57	0.	+ 0	-9.33000+	5	15	15
92238	1262	58	0.	+ 0	-9.67000+	5	14	13
92238	1262	59	0.	+ 0	-1.03000+	6	14	13
92238	1262	60	0.	+ 0	-1.04100+	6	14	13
92238	1262	61	0.	+ 0	-1.06000+	6	14	13
92238	1262	62	0.	+ 0	-1.12000+	6	14	13
92238	1262	63	0.	+ 0	-1.16000+	6	12	11
92238	1262	64	0.	+ 0	-1.22000+	6	12	12
92238	1262	65	0.	+ 0	-1.27000+	6	12	11
92238	1262	66	0.	+ 0	-1.30000+	6	11	10
92238	1262	67	0.	+ 0	-1.36100+	6	11	10
92238	1262	68	0.	+ 0	-1.40900+	6	11	9
92238	1262	69	0.	+ 0	-1.43700+	6	11	10
92238	1262	70	0.	+ 0	-1.47000+	6	12	12
92238	1262	71	0.	+ 0	-1.62500+	6	15	14
92238	1262	72	0.	+ 0	-1.87500+	6	12	12
92238	1262	73	0.	+ 0	-1.95000+	6	60	27
92238	1262	74	0.	+ 0	-2.95000+	6	52	22
92238	1262	75	0.	+ 0	-3.95000+	6	47	19
92238	1262	76	0.	+ 0	-4.95000+	6	45	17
92238	1262	91	0.	+ 0	-4.47000+	4	60	51
92238	1262	102	0.	+ 0	-4.80440+	6	263	283
92238	1262	251	0.	+ 0	0.	+ 0	28	27
92238	1262	252	0.	+ 0	0.	+ 0	28	26
92238	1262	253	0.	+ 0	0.	+ 0	28	25

MAT TOTALS				2568	1585
------------	--	--	--	------	------

93237	1263	1	0.	+ 0	0.	+ 0	323	193
93237	1263	2	0.	+ 0	0.	+ 0	316	57
93237	1263	4	0.	+ 0	-3.32000+	4	163	98
93237	1263	16	0.	+ 0	-6.76000+	6	77	55
93237	1263	17	0.	+ 0	-1.22900+	7	41	31
93237	1263	18	0.	+ 0	-2.00000+	8	316	228
93237	1263	51	0.	+ 0	-3.32000+	4	60	54
93237	1263	52	0.	+ 0	-5.96000+	4	51	44
93237	1263	53	0.	+ 0	-7.60000+	4	45	41
93237	1263	54	0.	+ 0	-1.03000+	5	39	35
93237	1263	55	0.	+ 0	-1.59000+	5	33	27
93237	1263	56	0.	+ 0	-2.24000+	5	30	25
93237	1263	57	0.	+ 0	-2.68000+	5	27	24
93237	1263	58	0.	+ 0	-3.05000+	5	25	23
93237	1263	59	0.	+ 0	-3.32000+	5	24	20
93237	1263	60	0.	+ 0	-3.69000+	5	21	19
93237	1263	61	0.	+ 0	-3.71000+	5	19	17
93237	1263	91	0.	+ 0	-3.32000+	5	128	70
93237	1263	102	0.	+ 0	-5.49000+	6	316	214

93237	1263	251	0.	+ 0	0.	+ 0	39	38
93237	1263	252	0.	+ 0	0.	+ 0	39	37
93237	1263	253	0.	+ 0	0.	+ 0	39	39

MAT TOTALS							2171	1392

22000	1286	1	0.	+ 0	0.	+ 0	1305	1043
22000	1286	2	0.	+ 0	0.	+ 0	913	898
22000	1286	3	0.	+ 0	0.	+ 0	444	397
22000	1286	4	0.	+ 0	-2.44000+	5	48	47
22000	1286	16	0.	+ 0	-1.16300+	7	6	6
22000	1286	51	0.	+ 0	-9.87000+	5	39	38
22000	1286	91	0.	+ 0	-2.44000+	5	17	16
22000	1286	102	0.	+ 0	8.14000+	6	395	390
22000	1286	103	0.	+ 0	-3.21000+	6	8	8
22000	1286	107	0.	+ 0	-2.03000+	4	6	5

MAT TOTALS							3181	2849

42000	1287	1	0.	+ 0	0.	+ 0	151	158
42000	1287	2	0.	+ 0	0.	+ 0	25	23
42000	1287	4	0.	+ 0	-9.19800+	5	19	16
42000	1287	16	0.	+ 0	-7.80000+	6	16	16
42000	1287	17	0.	+ 0	-1.48000+	7	7	7
42000	1287	91	0.	+ 0	0.	+ 0	19	16
42000	1287	102	0.	+ 0	7.25000+	6	121	272
42000	1287	251	0.	+ 0	0.	+ 0	13	13
42000	1287	252	0.	+ 0	0.	+ 0	13	13
42000	1287	253	0.	+ 0	0.	+ 0	13	13

MAT TOTALS							397	547

TAPE TOTALS							28970	21894

Error Messages

In the following error messages the underlined items are variables.

ALLOWABLE ERROR ----- 0.0-----

ALLOWANCE ERROR MUST BE POSITIVE EXECUTION TERMINATED

The input fractional convergence criteria must be greater than zero to allow convergence.

PAGING INDEX ERROR INDEX = 0 MAT/MF/MT = 1217 3 1

The paging index must be between 1 and N2 (the number of points in the section).¹ Normally, this message can only occur if there is a programming or computer error.

Operation at LLL

At Livermore a compiled, loaded, ready-to-execute version of the program is available from the photostore as file:

.193025:PROGRAMS:LINEAR

The program is small enough (14 000 words of small core memory and 6 000 words of large core memory) to run at standby during the daytime by executing file:

LINEAR / 1 S

The program will automatically use disk files as described on p. 6 and in Fig. 2, below.

Fig. 2. Linear operations.

Conversion to Other Installations

This edition of the code has been designed for operation at Lawrence Livermore Laboratory on a CDC-7600 computer. It uses special I/O routines to avoid the inefficiency of FORTRAN binary reads and writes. All points in the code that use these special routines are preceded and followed by comment cards of the form,

C**** LTSS*****

to indicate the Livermore time sharing system. For conversion to standard FORTRAN binary I/O, the required statements are included in the program as comment cards and can easily be activated. Alternatively, by recognizing all of these points one can easily implement his own direct disk access routines, which are equivalent to those used at Livermore. If FORTRAN binary I/O is used, subroutines

IONAME

FILEIO

WAIT

are not required. In addition, subroutines

SCROUT

SCRIN

may be replaced by a FORTRAN binary write or read, respectively (see SCROUT and SCRIN for details).

Program Listing

```

PROGRAM LINEAR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,          LIN00010
  I TAPE21,TAPE22)                                             LIN00020
C ***** LTSS *****                                         LIN00030
C   FORTRAN BINARY I/O REQUIRES AN ADDITIONAL UNIT (TAPE23)   LIN00040
C   I TAPE21,TAPE22,TAPE23)                                     LIN00050
C ***** LTSS *****                                         LIN00060
C                                                                 LIN00070
C   PROGRAM LINEAR                                             LIN00080
C   VERSION 74-1 (MAY 1974)                                     LIN00090
C   VERSION 75-1 (APRIL 1975)                                  LIN00100
C   VERSION 76-2 (OCTOBER 1976)                                LIN00110
C   VERSION 77-1 (JANUARY 1977) (REPORT UCRL-504C0, VOL. 17, PART A) LIN00120
C   VERSION 78-1 (JULY 1978)                                   LIN00130
C                                                                 LIN00140
C   WRITTEN BY DERMOTT E. CULLEN                                LIN00150
C       L-71                                                    LIN00160
C       UNIVERSITY OF CALIFORNIA                                LIN00170
C       LAWRENCE LIVERMORE LABORATORY                           LIN00180
C       P.O. BOX 808                                            LIN00190
C       LIVERMORE, CALIFORNIA                                  LIN00200
C   TELEPHONE 532-4061 (FTS)                                    LIN00210
C       415-422-4061 (COMMERCIAL)                               LIN00220
C                                                                 LIN00230
C   THIS PROGRAM IS DESIGNED TO CONVERT ENDF/B FILE 3 CROSS SECTIONS LIN00240
C   TO LINEAR-LINEAR INTERPOLABLE FORM. ANY SECTION THAT IS ALREADY LIN00250
C   LINEAR-LINEAR INTERPOLABLE WILL BE THINNED.                LIN00260
C                                                                 LIN00270
C   EACH ENDF/B SECTION (I.E. MT) IS REPRESENTED BY A TABLE OF ENERGY LIN00280
C   VS. CROSS SECTION AND ANY ONE OF FIVE POSSIBLE INTERPOLATION LIN00290
C   LAWS BETWEEN ANY TWO TABULATED POINTS. THIS PROGRAM WILL REPLACE LIN00300
C   EACH FILE 3 SECTION BY A NEW TABLE OF ENERGY VS. CROSS SECTION LIN00310
C   IN WHICH THE INTERPOLATION LAW IS ALWAYS LINEAR-LINEAR IN LIN00320
C   ENERGY-CROSS SECTION BETWEEN ANY TWO TABULATED POINTS    LIN00330
C                                                                 LIN00340
C   THE PROGRAM NON-SELECTIVELY LINEARIZES ALL SECTIONS OF FILE 3 LIN00350
C   DATA ON THE ORIGINAL ENDF/B FILE (I.E. ALL SECTIONS BETWEEN LIN00360
C   THE TAPE HEAD CARD AND THE TAPE END CARD).                 LIN00370
C                                                                 LIN00380
C   THIS PROGRAM ONLY USES THE ENDF/B BCD FORMAT AND COPIES ALL LIN00390
C   SECTIONS EXPECT FILE 3 AS HOLLERITH. AS SUCH IT MAY BE USED ON LIN00400
C   DATA IN THE ENDF/B-I, II, III OR IV FORMAT (SINCE A SECTION LIN00410
C   OF FILE 3 DATA IS IDENTICAL IN ALL ENDF/B VERSIONS).      LIN00420
C                                                                 LIN00430
C   THE PROGRAM READS AND LINEARIZES THE DATA A PAGE AT A TIME. LIN00440
C   AS SUCH THERE IS NO LIMIT TO THE NUMBER OF POINTS IN ANY LIN00450
C   SECTION.                                                    LIN00460
C                                                                 LIN00470
C   INPUT FILES                                                 LIN00480
C   -----                                                    LIN00490
C   UNIT   DESCRIPTION                                          LIN00500
C   ----   -                                                    LIN00510
C       5   INPUT CARDS                                         LIN00520
C       21  SOURCE FILE (ENDF/B DATA TO BE LINEARIZED)        LIN00530
C       23  SCRATCH FILE FOR LINEARIZED DATA.                 LIN00540
C                                                                 LIN00550
C   OUTPUT FILES                                                LIN00560
C   -----                                                    LIN00570
C   UNIT   DESCRIPTION                                          LIN00580
C   ----   -                                                    LIN00590
C       6   OUTPUT REPORT                                       LIN00600

```

```

C      22 RESULT FILE (LINEARIZED ENDF/B DATA)          LIN00610
C      23 SCRATCH FILE FOR LINEARIZED DATA              LIN00620
C                                                         LIN00630
C      INPUT CARDS                                         LIN00640
C      -----                                             LIN00650
C      CARD COLS. DESCRIPTION                             LIN00660
C      -----                                             LIN00670
C      1 1-11 FRACTIONAL CONVERGENCE CRITERIA             LIN00680
C              (MUST BE GREATER THAN ZERO)                LIN00690
C                                                         LIN00700
C      EXAMPLE INPUT                                       LIN00710
C      -----                                             LIN00720
C      TO LINEARIZE THE DATA TO 0.5 PER-CENT THE FOLLOWING INPUT LIN00730
C      CARD IS REQUIRED                                     LIN00740
C                                                         LIN00750
C      E.0000E-03                                          LIN00760
C                                                         LIN00770
C      ***** LTSS *****                               LIN00780
C                                                         LIN00790
C      ALL PROGRAM MODIFICATIONS FOR IMPLEMENTATION AT LIVERMORE ARE
C      ENCLOSED IN COMMENT CARDS THAT SAY LTSS (LIVERMORE TIME SHARING
C      SYSTEM). ALL OF THESE MODIFICATIONS ARE RELATED TO THE USE OF
C      DISK FILES AND I/O OPTIMIZATION. THE SUBROUTINES IONAME, FILEIO
C      WAIT ARE ONLY REQUIRED FOR OPERATION AT LIVERMORE.   LIN00840
C                                                         LIN00850
C      AS IMPLEMENTED AT LIVERMORE THIS PROGRAM WILL AUTOMATICALLY
C      USE THE FOLLOWING DISK FILE NAMES...                 LIN00860
C                                                         LIN00870
C      ENDFIN      = SOURCE DATA IN ENDF/B FORMAT        LIN00880
C      ENDFOUT     = RESULT DATA IN ENDF/B FORMAT        LIN00890
C      INPUT       = INPUT PARAMETERS                     LIN00900
C      OUTPUT      = OUTPUT REPORT                        LIN00910
C      SCRLINEAR   = SCRATCH FILE                         LIN00920
C                                                         LIN00930
C      THE SCRATCH FILE WILL BE WRITTEN AND READ USING THE RANDOM ACCESS
C      ROUTINE WOK AND ROK, WHICH AVOID THE INEFFICIENT FORTRAN BINARY
C      WRITE AND READ ROUTINES.                            LIN00940
C                                                         LIN00950
C      ***** LTSS *****                               LIN00960
C                                                         LIN00970
C      INTEGER OUTP,OTAPE,SCR,OUTZA,OUTTOT                LIN00980
C      COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR              LIN00990
C      COMMON/HEADER/C1H,C2H,L1H,L2H,N1H,N2H,MATH,MFH,MTH,NOSED
C      COMMON/COPCAT/CARD(17)                             LIN01000
C      COMMON/COUNTS/LASTZA,INZA,OUTZA,INTOT,OUTTOT       LIN01010
C      COMMON/GOOFY,ERROR                                  LIN01020
C      COMMON/NORMIT,XNORM(6),KSIGN(6),KEXP(6)            LIN01030
C      COMMON/ITCH/IOCH,IADR                               LIN01040
C      ***** LTSS *****                               LIN01050
C      CALL CHANGE(7H+LINEAR)                             LIN01060
C      ***** LTSS *****                               LIN01070
C                                                         LIN01080
C      DEFINE ALL I/O UNITS. TO CHANGE UNIT ASSIGNMENTS FOR ANY
C      OTHER INSTALLATION MERELY CHANGE THE FOLLOWING DEFINITIONS
C      TO SUIT YOURSELF, WHERE...                          LIN01090
C      INP          = INPUT CARD                           LIN01100
C      OUTP         = OUTPUT LISTING FILE                  LIN01110
C      ITAPE        = DATA IN ENDF/B FORMAT (TO BE LINEARIZED)
C      OTAPE        = DATA IN ENDF/B FORMAT (LINEARIZED) LIN01120
C      SCR          = SCRATCH FILE                         LIN01130
C                                                         LIN01140
C                                                         LIN01150
C                                                         LIN01160
C                                                         LIN01170
C                                                         LIN01180
C                                                         LIN01190
C                                                         LIN01200

```

C		LIN01210
C	IF YOU WISH TO ALLOW FOR VARIABLE UNIT ASSIGNMENTS MERELY	LIN01220
C	ACTIVATE THE SECOND INPUT FORM ALREADY LISTED AS A COMMENT	LIN01230
C	CARD BELOW (THE FORMAT IS ALREADY COMPATIBLE).	LIN01240
C		LIN01250
	INP=5	LIN01260
	OUTP=6	LIN01270
	ITAPE=21	LIN01280
	OTAPE=22	LIN01290
	SCR=23	LIN01300
C----	READ ALLOWABLE ERROR (E11.4)	LIN01310
	READ(INP,2000) ERROK	LIN01320
C----	VARIABLE UNIT NUMBER INPUT STATEMENT (E11.4,11X,3111)	LIN01330
C	READ(INP,2000) ERROK,ITAPE,OTAPE,SCR	LIN01340
	CALL NORMAL(ERROK,XNORM(1),KSIGN(1),KEXP(1))	LIN01350
C----	ABORT IF ALLOWABLE ERROR IS UNACCEPTABLE	LIN01360
	IF(ERROK.LE.0.0) GO TO 70	LIN01370
C*****	LTSS *****	LIN01380
C----	DEFINE ALL I/O UNITS AND DISK FILE NAMES.	LIN01390
	CALL IONAME	LIN01400
C*****	LTSS *****	LIN01410
C----	COPY TAPE LABEL RECORD TO LINEARIZED FILE.	LIN01420
	READ(OTAPE,5000) CARD,MATH	LIN01430
	WRITE(OTAPE,6000) CARD,MATH	LIN01440
C----	WRITE TITLE FOR OUTPUT REPORT.	LIN01450
	WRITE(OUTP,5900) XNORM(1),KSIGN(1),KEXP(1),CARD,MATH	LIN01460
C----	INITIALIZE TOTAL POINTS READ AND WRITTEN.	LIN01470
	INTOT=0	LIN01480
	OUTTOT=0	LIN01490
C----	INITIALIZE LAST ZA READ	LIN01500
	LASTZA=-100000	LIN01510
C----	INITIALIZE OUTPUT SEQUENCE NUMBER.	LIN01520
	NOSEQ=0	LIN01530
C----	READ AND WRITE NEXT CONTROL RECORD	LIN01540
	10 CALL CONT(1)	LIN01550
	CALL CONT(2)	LIN01560
C----	IS THIS A SEND,FEND,MENO OR TEND CARD.	LIN01570
	IF(MTH) 20,20,30	LIN01580
C----	YES. IF TEND CARD THIS IS END. IF NOT READ NEXT HEAD CARD.	LIN01590
	20 IF(MATH) 60,10,10	LIN01600
C----	SECTION HEAD CARD. IS IT FOR FILE 3 SECTION.	LIN01610
	30 IF(MFH-3) 40,50,40	LIN01620
C----	NO. COPY SECTION TO LINEARIZED FILE.	LIN01630
	40 CALL MTCOPY	LIN01640
	GO TO 10	LIN01650
C----	YES. LINEARIZE AND THIN SECTION.	LIN01660
	50 CALL MESH	LIN01670
C----	WRITE SECTION TO LINEARIZED FILE	LIN01680
	CALL COOUT	LIN01690
	GO TO 10	LIN01700
C----	END OF RUN.	LIN01710
	60 CONTINUE	LIN01720
C*****	LTSS *****	LIN01730
C	END FILE OTAPE	LIN01740
C	REWIND ITAPE	LIN01750
C	REWIND OTAPE	LIN01760
	CALL FILEIO(OTAPE,7HENDFOI'T,LL,3,1ERR)	LIN01770
	CALL DEVICE(7HDESTROY,9HSCLINEAR,1ERR,10CH)	LIN01780
C*****	LTSS *****	LIN01790
C----	WRITE POINT TOTAL FOR LAST MAT AND ADD TO TAPE TOTALS.	LIN01800

```

WRITE(OUTP,5920) INZA,OUTZA                                LIN01810
INTOT=INTOT+INZA                                           LIN01820
OUTTOT=OUTTOT+OUTZA                                        LIN01830
C-----WRITE POINT TOTALS FOR TAPE.                        LIN01840
WRITE(OUTP,5910) INTCT,OUTTOT                             LIN01850
CALL EXIT                                                  LIN01860
C-----ERROR IN INPUT. WRITE ERROR MESSAGE AND ABORT RUN. LIN01870
70 WRITE(OUTP,5930) XNORM(1),KXSIGN(1),KEXF(1)           LIN01880
CALL EXIT                                                  LIN01890
2000 FORMAT(11.4,11X,4111)                                LIN01900
5000 FORMAT(16A4,A2,14)                                    LIN01910
5900 FORMAT(33H1 LINEARIZE ENDF/B CROSS SECTIONS,         LIN01920
1 14H (LINEAR 78-1)/2X,54(1H-)/                          LIN01930
1 2X,19HTHINNING CRITERIA--F8.5,A1,12/2X,54(1H-)/        LIN01940
2 2X,10HTAPE LABEL/2X,16A4,A2,14/2X,54(1H-)/            LIN01950
3 6X,2HZA,5H MAT,5H MT,6X,6HKELVIN,5X,7HQ-VALUE,7H POINTS, LIN01960
4 7H POINTS/40X,2HEV,7H IN,7H OUT/2X,54(1H-))            LIN01970
5910 FORMAT(2X,54(1H-)/31X,11HTAPE TOTALS,217/2X,54(1H-)) LIN01980
5920 FORMAT(2X,54(1H-)/32X,10HMT TOTALS,217/2X,54(1H-))  LIN0,990
5930 FORMAT(33H1 LINEARIZE ENDF/B CROSS SECTIONS/2X,54(1H-)/ LIN02000
1 2X,20HALLOWABLE ERROR-----F8.5,A1,12/2X,54(1H-)/    LIN02010
2 2X,32HALLOWABLE ERROR MUST BE POSITIVE,                LIN02020
3 23H...EXECUTION TERMINATED)                             LIN02030
6000 FORMAT(16A4,A2,14,10H 0 0 0)                         LIN02040
END                                                         LIN02050
SUBROUTINE MESH                                             LIN02060
C                                                         LIN02070
C CONVERT DATA TO LINEAR-LINEAR FORM. IF LESS THAN 2011 POINTS LIN02080
C ARE GENERATED THEY WILL BE CORE RESIDENT. OTHERWISE THEY WILL LIN02090
C BE ON SCRATCH FILE (SCR)                                  LIN02100
C                                                         LIN02110
C INTEGER SCR                                               LIN02120
COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR                     LIN02130
COMMON/NBT/INT/NBT(20),INT(20)                             LIN02140
COMMON/PAGED/XP(1002),YP(1002),IXYLOW,IXYHI,LEFT         LIN02150
COMMON/HEADER/C1H,C2H,L1H,L2H,N1H,N2H,MATH,MFH,MTH,NOSEQ LIN02160
COMMON/LEADER/CIENDF,C2ENDF,L1ENDF,L2ENDF,N1ENDF,N2ENDF   LIN02170
COMMON/NEWTAB/XN(2010),YN(2010),NEW,N2NEW                LIN02180
COMMON/INDICE/K,KM1,KM2,LBC1,LBC2                         LIN02190
COMMON/GOOFY/ERROK                                         LIN02200
C-----DEFINE ARITHMETIC STATEMENT FUNCTION TO CALCULATE MINIMUM LIN02210
C-----ALLOWABLE ENERGY SPACING                          LIN02220
PLACEH(XD)=XD+0.00001*110.0**IFIX(ALOG10(XD)))            LIN02230
PLACEM(XD)=XD+0.000001*110.0**IFIX(ALOG10(XD)))           LIN02240
C-----DEFINE ARITHMETIC STATEMENT FUNCTION FOR LINEAR-LINEAR LIN02250
C-----INTERPOLATION.                                       LIN02260
TERP(X,X1,X2,Y1,Y2)=Y1+(X-X1)*(Y2-Y1)/(X2-X1)             LIN02270
C-----READ SECTION HEAD CARD AND INTERPOLATION LAW.      LIN02280
READ(1,TAPE,6000) CIENDF,C2ENDF,L1ENDF,L2ENDF,N1ENDF,N2ENDF, LIN02290
1 (NBT(1),INT(1)),1=1,N1ENDF)                              LIN02300
C-----INITIALIZE PAGING SYSTEM INDICES.                   LIN02310
IXYLOW=0                                                    LIN02320
IXYHI=0                                                     LIN02330
LEFT=N2ENDF                                                 LIN02340
C-----INITIALIZE TOTAL LINEARIZED POINT COUNT.           LIN02350
N2NEW=0                                                     LIN02360
C-----INITIALIZE POINT COUNT AND BACKWARD CONVERGENCE LIMITS LIN02370
K=1                                                         LIN02380
LBC1=1                                                      LIN02390
LBC2=2                                                      LIN02400

```

C-----DEFINE FIRST POINT IN TABLE.	LN02410
XN(1)=XPAGE(1)	LN02420
YN(1)=YPAGE(1)	LN02430
C-----SET UP LOOP OVER INTERPOLATION RANGES.	LN02440
NR2=1	LN02450
DO 150 NR=1,N1ENOF	LN02460
C-----DEFINE POINTS IN INTERPOLATION RANGE AND TYPE OF INTERPOLATION.	LN02470
NR1=NR2+1	LN02480
NR2=NB1(NR)	LN02490
INTYPE=INT(NR)	LN02500
C-----IF LINEAR-LINEAR JUST COPY POINTS INTO OUTPUT ARRAY.	LN02510
IF (INTYPE.NE.2) GO TO 20	LN02520
DO 10 NPT=NR1,NR2	LN02530
IF (K.GT.2002) CALL PAGOUT	LN02540
K=K+1	LN02550
XN(K)=XPAGE(NPT)	LN02560
YN(K)=YPAGE(NPT)	LN02570
GO TO 150	LN02580
C-----SET UP LOOP OVER POINTS.	LN02590
20 DO 140 NPT=NR1,NR2	LN02600
C-----DEFINE END OF INTERVAL.	LN02610
XEND=XPAGE(NPT)	LN02620
YEND=YPAGE(NPT)	LN02630
C-----IF CORE PAGES ARE FULL, THEN UNLOAD TO SCRATCH.	LN02640
30 IF (K.GT.2002) CALL PAGOUT	LN02650
C-----DEFINE INDICES FOR NEXT INTERVAL.	LN02660
KM2=K	LN02670
KM1=K+1	LN02680
K=K+2	LN02690
XN(K)=XEND	LN02700
YN(K)=YEND	LN02710
C-----DEFINE MINIMUM ENERGY SPACING	LN02720
IF (XN(KM2).GE.1.0) GO TO 40	LN02730
XSTOP=PLACEM(XN(KM2))	LN02740
GO TO 50	LN02750
40 XSTOP=PLACEM(XN(KM2))	LN02760
C-----DEFINE ENERGY AND CROSS SECTION AT MIDDLE OF INTERVAL.	LN02770
50 IF (XN(K).LE.XN(KM2)) GO TO 70	LN02780
60 GO TO (70,130,80,90,100),INTYPE	LN02790
C-----HISTOGRAM (AUTOMATIC CONVERGENCE)	LN02800
70 XN(KM1)=XN(KM2)	LN02810
YN(KM1)=YN(K)	LN02820
GO TO 130	LN02830
C-----LOG ENERGY, LINEAR CROSS SECTION	LN02840
80 XNKM1=SQRT(XN(KM2)*XN(K))	LN02850
YNKM1=0.5*(YN(KM2)+YN(K))	LN02860
GO TO 110	LN02870
C-----LINEAR ENERGY, LOG CROSS SECTION	LN02880
90 XNKM1=0.5*(XN(KM2)+XN(K))	LN02890
YNKM1=SQRT(YN(KM2)*YN(K))	LN02900
GO TO 110	LN02910
C-----LOG ENERGY, LOG CROSS SECTION	LN02920
100 XNKM1=SQRT(XN(KM2)*XN(K))	LN02930
YNKM1=SQRT(YN(KM2)*YN(K))	LN02940
C-----ASSUME CONVERGENCE IF POINTS CLOSER THAN ENDF ENERGY FORMAT	LN02950
C-----RESOLUTION.	LN02960
110 IF (XNKM1.LE.XSTOP) GO TO 120	LN02970
C-----DETERMINE IF THE EXACT VALUE CAN BE APPROXIMATED TO WITHIN THE	LN02980
C-----ALLOWABLE ERROR BY LINEAR-LINEAR INTERPOLATION.	LN02990
YAPROX=TERP(XNKM1,XN(KM2),XN(K),YN(KM2),YN(K))	LN03000

```

      IF (ABS(YNKM1-YAPROX).LE.ABS(ERRJK*YNKM1)) GO TO 120
C-----NO CONVERGENCE. SHORTEN INTERVAL.
      XN(K)=XNKM1
      YN(K)=YNKM1
      GO TO 60
C-----CONVERGENCE.
      120 XN(KM1)=XNKM1
      YN(KM1)=YNKM1
C-----IF NOT END OF CURRENT INTERVAL CONTINUE. OTHERWISE
C-----MOVE TO NEXT INTERVAL.
      130 IF (XN(K).LT.XEND) GO TO 30
      140 CONTINUE
      150 CONTINUE
C-----EACH OF SECTION. SET UP PAGES FOR OUTPUT.
      CALL PAGENO
      RETURN
      6000 FORMAT(2E11.4,4I11/(6I11))
      END
      SUBROUTINE COPOUT
C
C      COPY LINEARIZED SECTION TO OUTPUT FILE IN ENDF/B FORM. IF
C      LESS THAN 2011 POINTS IN SECTION THEY ARE CORE RESIDENT.
C      OTHERWISE THEY ARE READ FROM SCRATCH FILE (SCR) A PAGE AT
C      A TIME AND WRITTEN OUT.
C
      INTEGER OUTP,OTAPE,ONE,TWO,OUTZA,OUTTOT,SCR
      COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR
      COMMON/HEADER/C1H,C2H,L1H,L2H,N1H,N2H,MATH,MFH,MTN,N0SEQ
      COMMON/LEADER/C1,C2,L1,L2,N1,N2
      COMMON/NEWTAB/XN(2010),YN(2010),NEW,N2NEW
      COMMON/NORMIT/XNORM(6),KSIGN(6),KEXP(6)
      COMMON/COUNTS/LASTZA,INZA,OUTZA,INTOT,OUTTOT
      COMMON/ITCH/IOCH,IADR
      DATA ONE/1/
      DATA TWO/2/
      DATA TEMP/0.0/
      NOSEQ=NXTSEQ(NOSEQ)
      CALL NORMAL(C1,XNORM(1),KSIGN(1),KEXP(1))
      CALL NORMAL(C2,XNORM(2),KSIGN(2),KEXP(2))
      IF (ABS(L1).LE.0) L1=0
      IF (ABS(L2).LE.0) L2=0
      WRITE(OTAPE,6000) (XNORM(L),KSIGN(L),KEXP(L),L=1,2),
      1 L1,L2,ONE,N2NEW,MATH,MFH,MTN,NOSEQ
      NOSEQ=NXTSEQ(SEQ)
      WRITE(OTAPE,6010) N2NEW,TWO,MATH,MFH,MTN,NOSEQ
C-----WRITE OUTPUT REPORT.
      IZA=C1H
      IF (IZA.EQ.LASTZA) GO TO 20
      IF (LASTZA.LE.0) GO TO 10
      WRITE(OUTP,5910) INZA,OUTZA
      INTOT=INTOT+INZA
      OUTTOT=OUTTOT+OUTZA
      10 LASTZA=IZA
      INZA=0
      OUTZA=0
C-----DEFINE TEMPERATURE
      20 IF (L2.LE.0) GO TO 30
      TEMP=TEMP1
      GO TO 40
      30 TEMP=C1

```

```

      IF(MTH.EQ.1) TEMP=TEMP
40  CALL NORMAL(TEMP,XNORM(1),KSIGN(1),KEXP(1))
      CALL NORMAL(C2,XNORM(2),KSIGN(2),KEXP(2))
      WRITE(OUTP,5900) IZA,MATH,MTH,(XNORM(L),KSIGN(L),KEXP(L),L=1,2),
      I N2,N2NEW
C-----INCREMENT TOTAL POINTS READ AND WRITTEN.
      INZA=INZA+N2
      OUTZA=OUTZA+N2NEW
C--  GET UP LOOP OVER PAGES FROM SCRATCH.
      DO 110 NR1=1,N2NEW,2004
C---- POSITION UP TO TWO PAGES OF DATA IN CORE.
      NR2=NR1+2003
      IF(NR2.GT.N2NEW) NR2=N2NEW
      NN=NR2-NR1+1
      IF(N2NEW.LE.NEW) GO TO 50
      CALL SCRIN(XN,YN)
      IF(NN.GT.1002) CALL SCRIN(XN(1003),YN(1003))
50  DO 100 N3=1,NN,3
      N4=N3+2
      IF(N4.GT.NN) N4=NN
C-----CONVERT DATA TO NORMAL FORM
      N5=0
      DO 60 L=N3,N4
      N5=N5+1
      CALL NORMAL(XN(L),XNORM(N5),KSIGN(N5),KEXP(N5))
      N3=N5+1
60  CALL NORMAL(YN(L),XNORM(N5),KSIGN(N5),KEXP(N5))
C-----OUTPUT CARD.
      NOSEQ=NXTSEQ(NSEQ)
      NUSE=N4-N3+1
      GO TO (70,80,90),NUSE
70  WRITE(OTAPE,6020) (XNORM(L),KSIGN(L),KEXP(L),L=1,2),MATH,
      I MFH,MTH,NOSEQ
      GO TO 100
80  WRITE(OTAPE,6030) (XNORM(L),KSIGN(L),KEXP(L),L=1,4),MATH,
      I MFH,MTH,NOSEQ
      GO TO 100
90  WRITE(OTAPE,6040) (XNORM(L),KSIGN(L),KEXP(L),L=1,6),MATH,
      I MFH,MTH,NOSEQ
100 CONTINUE
110 CONTINUE
      RETURN
5900 FORMAT(2X,16,215,2(1X,F8.5,A1,12),217)
5910 FORMAT(2X,54(1H-)/32X,10HMAT TOTALS,217/2X,54(1H-);
6000 FORMAT(2(F8.5,A1,12),4111,14,12,13,15)
6010 FORMAT(2111,44X,14,12,13,15)
6020 FORMAT(2(F8.5,A1,12),44X,14,12,13,15)
6030 FORMAT(4(F8.5,A1,12),22X,14,12,13,15)
6040 FORMAT(6(F8.5,A1,12),14,12,13,15)
      END
      SUBROUTINE MTCOPY
C
C  COPY SECTION OF DATA FROM INPUT TO OUTPUT FILES IN THE
C  END/FB FORMAT. SECTION HEAD CARD HAS ALREADY BEEN READ AND
C  WRITTEN. ENTIRE SECTION IS SIMPLY COPIED AS HOLLERITH.
C
      INTEGER OTAPE
      COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR
      COMMON/HEADER/CIH,C2H,LIH,L2H,NIH,M2H,MATH,MFH,MTH,NOSEQ
      COMMON/COPCAT/CARD(17)

```



```

C-----COPY SECTION TO LINEARIZED FILE.
10 READ(1TAPE,5000) CARD,MATH,MFH,MTN
   NOSEQ=NXTSEQ(NONSEQ)
   IF(MTH) 30,30,20
20 WRITE(1TAPE,5000) CARD,MATH,MFH,MTN,NOSEQ
   GO TO 10
30 WRITE(1TAPE,5010) MATH,MFH,MTN,NOSEQ
   RETURN
5000 FORMAT(16A4,A2,14,12,13,15)
5010 FORMAT(66X,14,12,13,15)
   END
   SUBROUTINE CONT(IWAY)
C
C   READ OR WRITE ENDF/B CONTROL RECORD
C
   INTEGER OTAPE
   COMMON/UNITS/OUTP,INP,1TAPE,1TAPE,SCR
   COMMON/HEADER/C1H,C2H,L1H,L2H,N1H,N2H,MATH,MFH,MTN,NOSEQ
   COMMON/NORMIT/XNORM(6),KSIGN(6),KEXP(6)
C-----SELECT READ OR WRITE.
   GO TO (10,20),IWAY
C-----READ CONTROL RECORD.
10 READ(1TAPE,5000) C1H,C2H,L1H,L2H,N1H,N2H,MATH,MFH,MTN
   RETURN
C-----WRITE CONTROL RECORD.
20 NOSEQ=NXTSEQ(NONSEQ)
   IF(MTH) 30,30,80
30 IF(MATH) 40,50,50
40 NOSEQ=0
50 WRITE(1TAPE,6000) MATH,MFH,MTN,NOSEQ
   IF(MATH) 60,60,70
60 NOSEQ=0
70 RETURN
80 CALL NORMAL(C1H,XNORM(1),KSIGN(1),KEXP(1))
   CALL NORMAL(C2H,XNORM(2),KSIGN(2),KEXP(2))
   IF(ABS(L1H).EQ.0) L1H=0
   IF(ABS(L2H).EQ.0) L2H=0
   IF(ABS(N1H).EQ.0) N1H=0
   IF(ABS(N2H).EQ.0) N2H=0
   WRITE(1TAPE,6010) (XNORM(I),KSIGN(I),KEXP(I),I=1,2),L1H,L2H,
1 N1H,N2H,MATH,MFH,MTN,NOSEQ
   RETURN
5000 FORMAT(2E11.4,4I11,14,12,13,15)
6000 FORMAT(66X,14,12,13,15)
6010 FORMAT(F8.5,A1,12,F9.5,A1,12,4I11,14,12,13,15)
   END
   SUBROUTINE PAGOUT
C
C   THIN LINEARIZED DATA AND IF NECESSARY OUTPUT ONE PAGE TO
C   SCRATCH.
C
   INTEGER SCR
   COMMON/UNITS/OUTP,INP,1TAPE,1TAPE,SCR
   COMMON/NEWTAB/XN(2010),YN(2010),NEW,N2NEW
   COMMON/INDICE/K,KM1,KM2,LBC1,LBC2
   COMMON/GOOFY/ERROK
   COMMON/ITCH/IOCK,IADR
C-----SET UP LOOP OVER POINTS.
   LBC3=LBC2+1
   DO 40 N=LBC3,K

```

```

      MM1=M-1
C----- IS ENERGY OF TWO POINTS THE SAME.
      DXN=XN(M)-XN(LBC1)
      IF(DXN.GT.0.0) GO TO 10
C----- YES. KEEP POINT IF DIFFERENT CROSS SECTION. OTHERWISE ELIMINATE.
      IF(YN(M)-YN(LBC1)) 30,40,30
C----- DEFINE SLOPE BETWEEN ENDS OF INTERVAL.
      10 SLOPE=(YN(M)-YN(LBC1))/DXN
C----- DETERMINE IF ALL INTERMEDIATE POINTS CAN BE APPROXIMATED BE
C----- LINEAR-LINEAR INTERPOLATION FROM THE ENDS OF THE INTERVAL.
      DO 20 L=LBC2,MM1
      YAPROX=YN(LBC1)+(XN(L)-XN(LBC1))*SLOPE
      IF(ABS(YAPROX-YN(L)).GT.ABS(ERROK*YN(L))) GO TO 30
      20 CONTINUE
C----- CAN ELIMINATE ALL INTERMEDIATE POINTS.
      GO TO 40
C----- NO. NEED TO KEEP LAST PRECEDING POINT.
      30 LBC1=LBC1+1
      XN(LBC1)=XN(MM1)
      YN(LBC1)=YN(MM1)
      LBC2=M
      40 CONTINUE
C----- OUTPUT PAGE IF IT IS FULL. OTHERWISE THIN ALL INTERMEDIATE
C----- POINTS.
      IF(LBC1.GT.1002) GO TO 50
      LBC1=LBC1+1
      XN(LBC1)=XN(K)
      YN(LBC1)=YN(K)
      K=LBC1
      LBC2=LBC1+1
      RETURN
C----- REWIND SCRATCH BEFORE FIRST WRITE
      50 IF(N2NEW.GT.0) GO TO 60
C***** LTSS *****
C      REWIND SCR
      IADR=0
C***** LTSS *****
C----- OUTPUT ONE PAGE OF LINEARIZED DATA AND INCREASE POINT COUNT
      60 CALL SCROUT(XN,YN)
      N2NEW=N2NEW+1002
      LL=0
C----- COPY REMAINING POINTS TO BEGINNING OF 1ABLE
      DO 70 LLL=1003,K
      LL=LL+1
      XN(LL)=XN(LL+1)
      70 YN(LL)=YN(LL+1)
C----- DEFINE NEW IN CORE POINT INDICES.
      K=K-1002
      LBC1=LBC1-1002
      LBC2=LBC2-1002
      RETURN
      ENTRY PAGE=ND
C
C      END OF SECTION. THIN REMAINING LINEARIZED DATA AND IF
C      NECESSARY MOVE REMAINING DATA TO SCRATCH. IF SCRATCH IS
C      USED POSITION IT FOR READING.
C
C----- SET UP LOOP OVER POINTS.
      IF(LBC2.GE.K) GO TO 120
      LBC1=LBC2+1

```

```

LIN04810
LIN04820
LIN04830
LIN04840
LIN04850
LIN04860
LIN04870
LIN04880
LIN04890
LIN04900
LIN04910
LIN04920
LIN04930
LIN04940
LIN04950
LIN04960
LIN04970
LIN04980
LIN04990
LIN05000
LIN05010
LIN05020
LIN05030
LIN05040
LIN05050
LIN05060
LIN05070
LIN05080
LIN05090
LIN05100
LIN05110
LIN05120
LIN05130
LIN05140
LIN05150
LIN05160
LIN05170
LIN05180
LIN05190
LIN05200
LIN05210
LIN05220
LIN05230
LIN05240
LIN05250
LIN05260
LIN05270
LIN05280
LIN05290
LIN05300
LIN05310
LIN05320
LIN05330
LIN05340
LIN05350
LIN05360
LIN05370
LIN05390
LIN05390
LIN05400

```

```

DO 110 M=LBC3,K                                LIN05410
MM1=M-1                                          LIN05420
C-----DEFINE CHANGE IN ENERGY AND CHECK FOR TWO POINTS AT SAME LIN05430
C-----ENERGY.                                LIN05440
DXN=XN(M)-XN(LBC1)                             LIN05450
IF(DXN.LE.0.0) GO TO 90                        LIN05460
C-----DEFINE SLOPE BETWEEN ENDS OF INTERVAL. LIN05470
SLOPE=(YN(M)-YN(LBC1))/DXN                     LIN05480
C-----DETERMINE IF ALL INTERMEDIATE POINTS CAN BE APPROXIMATED BE LIN05490
C-----LINEAR-LINEAR INTERPOLATION FROM THE ENDS OF THE INTERVAL. LIN05500
DO 80 L=LBC2,MM1                               LIN05510
YAPROX=YN(LBC1)+(XN(L)-XN(LBC1))*SLOPE        LIN05520
IF(ABS(YAPROX-YN(L)).GT.ABS(ERROK*YN(L))) GO TO 100 LIN05530
80 CONTINUE                                     LIN05540
C-----CAN ELIMINATE ALL INTERMEDIATE POINTS. LIN05550
GO TO 110                                       LIN05560
C-----POINTS AT SAME ENERGY. KEEP BOTH IF DIFFRENT CROSS SECTION. LIN05570
C-----OTHERWISE ELIMINATE.                   LIN05580
90 IF(YN(M).EQ.YN(LBC1)) GO TO 110             LIN05590
C-----NO CONVERGENCE. NEED TO KEEP LAST PRECEDING POINT. LIN05600
100 LBC1=LBC1+1                                LIN05610
XN(LBC1)=XN(MM1)                               LIN05620
YN(LBC1)=YN(MM1)                               LIN05630
LBC2=M                                           LIN05640
110 CONTINUE                                    LIN05650
C-----THIN ALL INTERMEDIATE POINTS           LIN05660
120 LBC1=LBC1+1                                LIN05670
XN(LBC1)=XN(K)                                  LIN05680
YN(LBC1)=YN(K)                                  LIN05690
C-----IF DATA ALREADY ON SCRATCH COPY REMAINING DATA TO SCRATCH LIN05700
C-----IF LESS THAN TWO PAGES OF DATA SECTION WILL BE IN CORE; LIN05710
NEW=LBC1                                         LIN05720
IF(N2NEW.LE.0) GO TO 130                       LIN05730
CALL SCRUT(XN,YN)                              LIN05740
IF(NEW.GT.1002) CALL SCRUT(XN(1003),YN(1003)) LIN05750
C-----POSITION SCRATCH FOR READING           LIN05760
C***** LTSS *****                          LIN05770
C END FILE SCR                                  LIN05780
C REWIND SCR                                    LIN05790
IADR=0                                           LIN05800
C***** LTSS *****                          LIN05810
C-----DEFINE FINAL POINT TOTAL FOR SECTION LIN05820
130 N2NEW=N2NEW+NEW                             LIN05830
RETURN                                           LIN05840
END                                              LIN05850
SUBROUTINE NORMAL(X,XNORM,KSIGN,KEXP)           LIN05860
C                                              LIN05870
C CONVERT FLOATING POINT NUMBER TO NORMAL FORM FOR OUTPUT. LIN05880
C OUTPUT WILL BE IN THE FORM X.XXXXX+/-NN, WHICH GIVES LIN05890
C AN ADDITIONAL DIGIT OF ACCURACY WHEN COMPARED TO E11.4 LIN05900
C FORTRAN OUTPUT. IN ADDITION THE OUTPUT WILL BE COMPUTER LIN05910
C INDEPENDENT (I.E. SAME ON IBM, CDC, UNIVAC ETC.) LIN05920
C                                              LIN05930
C INTEGER PLUS                                  LIN05940
C-----DEFINE TWO POSSIBLE SIGNES FOR EXPONENT. LIN05950
DATA PLUS/1H+//                                LIN05960
DATA MINUS/1H-//                              LIN05970
C-----COMPUTE SIGNED EXPONENT FROM ABSOLUTE VALUE. LIN05980
XABS=ABS(X)                                     LIN05990
C-----IF NUMBER IS 0.0 DEFINE 0.00000+ 0     LIN06000

```

```

      IF(XABS) 50,50,10
C-----DEFINE EXPONENT TO NORMALIZE MANTISSA.
10 KEXP=ALOG10(XABS)
   SHIFT=10.0**KEXP
   XNORM=XABS/SHIFT
   IF(XNORM.GE.0.999995) GO TO 20
   KEXP=KEXP-1
   SHIFT=SHIFT/10.0
   GO TO 30
20 IF(XNORM.LT.0.999995) GO TO 30
   KEXP=KEXP+1
   SHIFT=10.0*SHIFT
C-----COMPUTE SIGNED MANTISSA.
30 XNORM=X/SHIFT
C-----DEFINE SIGN OF EXPONENT AND INSURE EXPONENT IS POSITIVE.
   IF(KEXP) 40,60,70
40 KSIGN=MINUS
   KEXP=-KEXP
   RETURN
C-----X IS 0.0. DEFINE AS 0.00000+ 0
50 XNORM=0.0
60 KEXP=0
70 KSIGN=PLUS
   RETURN
   END
   FUNCTION XPAGE(N)

C
C   LOGICAL PAGING SYSTEM FOR X ARRAY. NOTE THAT THIS ROUTINE
C   CAN ONLY BE USED TO ADVANCE FORWARD THROUGH AN ENDF/B
C   SECTION, SINCE IT READS THE DATA DIRECTLY FROM THE ENDF/B
C   DATA FILE A PAGE AT A TIME AND DOES NOT SAVE PAST PAGES.
C
C   INTEGER OUTP
C   COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR
C   COMMON/HEADER/C1H,C2H,L1H,L2H,N1H,N2H,MATH,MFH,MTH,NOSEQ
C   COMMON/LEADER/C1,C2,L1,L2,N1,N2
C   COMMON/PAGED/XP(1002),YP(1002),IXYLOW,IXYHI,LEFT
C-----INSURE THAT INDEX IS IN LEGAL RANGE.
   IF(N-IXYLOW) 20,20,10
10 IF(N-N2) 30,30,20
C-----INDEX ERROR.
20 WRITE(OUTP,6000) N,MATH,MFH,MTH
   XPAGE=0.0
   RETURN
C-----DETERMINE IF REQUIRED POINT IS IN CORE YET.
30 IF(N-IXYHI) 50,50,40
C-----NO. READ NEXT PAGE INTO CORE.
40 NN=1002
   IF(LEFT.LT.NN) NN=LEFT
   LEFT=LEFT-NN
   IXYLOW=IXYHI
   IXYHI=IXYHI+NN
   READ(ITAPE,5000) (XP(I),YP(I),I=1,NN)
   GO TO 30
C-----DEFINE INDEX RELATIVE TO BEGINNING OF IN CORE TABLE AND
C-----RETURN VALUE
50 INDEX=N-IXYLOW
   XPAGE=XP(INDEX)
   RETURN
5000 FORMAT(6E11.4)

```

```

6000 FORMAT(2BH PAGING INDEX ERROR...INDEX=,15,10HMAT/MF/MT=,15,13,14) LIN06618
END LIN06620
FUNCTION YPAGE(N) LIN06630
C LIN06640
C LOGICAL PAGING SYSTEM FOR Y ARRAY. NOTE THAT THIS ROUTINE LIN06640
C CAN ONLY BE USED TO ADVANCE FORWARD THROUGH AN ENDF/B LIN06660
C SECTION, SINCE IT READS THE DATA DIRECTLY FROM THE ENDF/B LIN06670
C DATA FILE A PAGE AT A TIME AND DOES NOT SAVE PAST PAGES. LIN06680
C LIN06690
INTEGER OUTP LIN06700
COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR LIN06710
COMMON/HEADER/C1H,C2H,L1H,L2H,N1H,N2H,HATH,MFH,MTH,NOSEQ LIN06720
COMMON/LEADER/C1,C2,L1,L2,N1,N2 LIN06730
COMMON/PAGED/XP(1002),YP(1002),IXYLOW,IXYHI,LEFT LIN06740
C----INSURE THAT INDEX IS IN LEGAL RANGE. LIN06750
IF(N-IXYLOW) 20,20,10 LIN06760
10 IF(N-N2) 30,30,20 LIN06770
C----INDEX ERROR. LIN06780
20 WRITE(OUTP,6000) N,MATH,MFH,MTH LIN06790
YPAGE=0.0 LIN06800
RETURN LIN06810
C----DETERMINE IF REQUIRED POINT IS IN CORE YET. LIN06820
30 IF(N-IXYHI) 50,50,40 LIN06830
C----NO. READ NEXT PAGE INTO CORE. LIN06840
40 NN=1002 LIN06850
IF(LEFT.LT.NN) NN=LEFT LIN06860
LEFT=LEFT-NN LIN06870
IXYLOW=IXYHI LIN06880
IXYHI=IXYHI+NN LIN06890
READ(ITAPE,5000) (XP(I),YP(I),I=1,NN) LIN06900
GO TO 30 LIN06910
C----DEFINE INDEX RELATIVE TO BEGINNING OF IN CORE TABLE AND LIN06920
C----RETURN VALUE LIN06930
50 INDEX=N-IXYLOW LIN06940
YPAGE=YP(INDEX) LIN06950
RETURN LIN06960
5000 FORMAT(6E11.4) LIN06970
6000 FORMAT(2BH PAGING INDEX ERROR...INDEX=,15,10HMAT/MF/MT=,15,13,14) LIN06980
END LIN06990
FUNCTION NXTSEQ(NOSEQ) LIN07000
C LIN07010
C DEFINE NEXT SEQUENCE NUMBER FOR ENDF/B OUTPUT. ALLOW FOR LIN07020
C MORE THAN 100000 CARDS PER EVALUATION BY RESETING NUMBER LIN07030
C TO 1 EVERY TIME 100000 IS REACHED LIN07040
C LIN07050
NN=NOSEQ+1 LIN07060
IF(NN.EQ.100000) NN=1 LIN07070
NXTSEQ=NN LIN07080
RETURN LIN07090
END LIN07100
SUBROUTINE SCROUT(X,Y) LIN07110
C LIN07120
C WRITE ONE PAGE OF ENERGY AND CROSS SECTION TO SCRATCH FILE. LIN07130
C LIN07140
INTEGER SCR LIN07150
COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR LIN07160
COMMON/ITCH/IOCW,IADR LIN07170
DIMENSION X(1002),Y(1002) LIN07180
C***** LTSS ***** LIN07190
C WRITE(SCR) X,Y LIN07200

```

CALL WDK(10CW,IADR,X,1002)	LN07210
CALL WAIT(10CW,IERR)	LN07220
IADR=IADR+1002	LN07230
CALL WDK(10CW,IADR,Y,1002)	LN07240
CALL WAIT(10CW,IERR)	LN07250
IADR=IADR+1002	LN07260
C***** LTSS *****	LN07270
RETURN	LN07280
END	LN07290
SUBROUTINE SCRIN(X,Y)	LN07300
C	LN07310
C READ ONE PAGE OF ENERGY AND CROSS SECTION FROM SCRATCH FILE.	LN07320
C	LN07330
IN EGER SCR	LN07340
COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR	LN07350
COMMON/ITCH/10CW,IADR	LN07360
DIMENSION X(1002),Y(1002)	LN07370
C***** LTSS *****	LN07380
C READ(SCR) X,Y	LN07390
CALL RDK(10CW,IADR,X,1002)	LN07400
CALL WAIT(10CW,IERR)	LN07410
IADR=IADR+1002	LN07420
CALL RDK(10CW,IADR,Y,1002)	LN07430
CALL WAIT(10CW,IERR)	LN07440
IADR=IADR+1002	LN07450
C***** LTSS *****	LN07460
RETURN	LN07470
END	LN07480
SUBROUTINE IOKAME	LN07490
C	LN07500
C DEFINE ALL I/O UNITS AND DISK FILE NAMES.	LN07510
C	LN07520
INTEGER OUTP,OTAPE	LN07530
COMMON/UNITS/OUTP,INP,ITAPE,OTAPE,SCR	LN07540
COMMON/ITCH/10CW,IADR	LN07550
C-----RE-DEFINE OUTPUT REPORT UNIT FOR TELETYPE WHILE I/O UNITS ARE	LN07560
C-----BEING DEFINED (ANY ERROR MESSAGES WILL THEN APPEAR ON LINE).	LN07570
OUTP=59	LN07580
C-----DEFINE ENDF/B FORMAT FILE TO BE LINEARIZED.	LN07590
CALL FILEIO(ITAPE,6,HENDFIN,LLL,1,IERR)	LN07600
IF(IERR.NE.0) CALL EXIT	LN07610
C-----DEFINE ENDF/B FORMAT FILE FOR LINEARIZED DATA.	LN07620
LLL=3*LLL	LN07630
IF(LLL.LT.100000) LLL=100000	LN07640
CALL FILEIO(OTAPE,7,HENDFOUT,LLL,2,IERR)	LN07650
IF(IERR.NE.0) CALL EXIT	LN07660
C-----DEFINE SCRATCH UNIT FOR LINEARIZED DATA.	LN07670
LLL=500000	LN07680
IADR=0	LN07690
ERRA CALL DEVICE(7HDESTROY,9HSCRLINEAR,IERR,10CW)	LN07700
CALL DEVICE(6HCREATE,9HSCRLINEAR,LLL,IERR,10CW)	LN07710
C- -IF ANY PROBLEM DELAY AND THEN TRY AGAIN.	LN07720
IF(IERR.LE.0) GO TO ERRB	LN07730
CALL FROST(2500B,0,5000000,ERRA)	LN07740
GO TO ERRA	LN07750
C-----RE-DEFINE OUTPUT REPORT UNIT FOR PRINTER.	LN07760
ERRB OUTP=6	LN07770
RETURN	LN07780
END	LN07790
SUBROUTINE FILEIO(UNIT,NAME,LENGTH,IWAY,IERR)	LN07800

```

C*                                     LINO7810
C* THIS ROUTINE IS DESIGNED TO HANDLE ALL DEVICE ASSIGNMENTS. LINO7820
C* FILE OPENING AND CLOSING AS WELL AS END OF FILES AND LAST LINO7830
C* BUFFER UNLOADING (EMPTY). LINO7840
C* LINO7850
C* THE TYPE OF OPERATION IS DEFINED BY IWAY AS FOLLOWS.... LINO7860
C* IWAY =1 OPEN FILE LINO7870
C*      =2 CREATE FILE LINO7880
C*      =3 END FILE, EMPTY BUFFER, TRUNCATE SIZE AND CLOSE FILE. LINO7890
C*      =4 END FILE, TRUNCATE SIZE AND CLOSE FILE. LINO7900
C*      =5 CLOSE FILE. LINO7910
C*      =6 RENAME FILES (DELETE UNIT, RENAME NAME TO UNIT LINO7920
C*          (AS BACKUP) RENAME LENGTH TO UNIT) LINO7930
C*      =7 OPEN TAPE. LINO7940
C*      =8 DESTROY FILE (USUALLY SCRATCH AT END OF EXECUTION) LINO7950
C* LINO7960
C* INTEGER TTY,UNIT,BLANK LINO7970
C* COMMON/UNIT5/TTY LINO7980
C* COMMON/Q10D789/ICBLK(8,16) LINO7990
C* DIMENSION IBETA(2),NBETA(2),MAXWAIT(2) LINO8000
C* DATA BLANK/IH / LINO8010
C-----DEFINE SUSPEND TIMES (IN MICROSECONDS) TO SUSPEND FOR 5 MINUTES LINO8020
C-----FOR AN OPEN OR 5 SECONDS FOR A CREATE. LINO8030
C* DATA NBETA/300000000,5000000/ LINO8040
C-----DEFINE SUSPEND COUNTS FOR A TOTAL OF 5 HOURS. LINO8050
C* DATA MAXWAIT/60,3600/ LINO8060
C-----INITIALIZE SUSPEND WAIT COUNT. LINO8070
C* NWAIT=0 LINO8080
C-----BRANCH TO PROPER OPERATION. LINO8090
C* GO TO (10,10,90,90,140,160,10,200),IWAY LINO8100
C-----DEFINE CORRESPONDENCE BETWEEN UNIT AND FILENAME (OPEN OR CREATE) LINO8110
C* 10 CALL ASSIGN(UNIT,NAME) LINO8120
C* CALL NOPAGE(UNIT) LINO8130
C-----BRANCH TO OPEN, CREATE OR TAPE. LINO8140
C* IF(IWAY-2) 20,40,30 LINO8150
C-----OPEN FILE. LINO8160
C* 20 CALL DEVICE(4HOPEN,NAME,LENGTH,IERR) LINO8170
C* IF(IERR.LE.0.OR.IERR.EQ.2) GO TO 150 LINO8180
C* IF(TTY.EQ.-59) RETURN LINO8190
C* GO TO 50 LINO8200
C-----OPEN TAPE LINO8210
C* 30 CALL DEVICE(4HTAPE,NAME,IERR) LINO8220
C-----CHECK FOR ERROR (ALLOW WRITE PROTECTED FILE,IERR=2). LINO8230
C* IF(IERR.LE.0.OR.IERR.EQ.2) GO TO 150 LINO8240
C-----ERROR DURING OPEN OR TAPE. BRANCH TO ERROR MESSAGE. LINO8250
C* GO TO 50 LINO8260
C-----CREATE FILE (FIRST DESTROY ANY FILE OF SAME NAME). LINO8270
C* 40 CALL DEVICE(7HDESTROY,NAME,IERR) LINO8280
C* CALL DEVICE(6HCREATE,NAME,LENGTH,IERR) LINO8290
C-----CHECK FOR ERROR DURING FILE CREATION. LINO8300
C* IF(IERR.LE.0) GO TO 150 LINO8310
C-----ERROR DURING OPE, TAPE OR CREATE. PRINT ERROR MESSAGE FIRST TIME LINO8320
C-----AND DETERMINE IF SUSPEND RE-TRY LOOP SHOULD BE ENTERED. LINO8330
C* 50 IF(NWAIT.GT.0) GO TO 60 LINO8340
C* IF(TTY.GT.0) WRITE(TTY,5900) NAME LINO8350
C* GO TO 70 LINO8360
C-----ALREADY IN SUSPEND LOOP. TEST IF MAXIMUM NUMBER OF SUSPENDS LINO8370
C-----HAS BEEN EXCEEDED. LINO8380
C* 60 IF(NWAIT.LE.MAXWAIT(IWAY)) GO TO 80 LINO8390
C* WRITE(TTY,5960) LINO8400

```

IF(TTY.NE.59) RETURN	LIN08410
NWAIT=0	LIN08420
GO TO 70	LIN08430
C-----NO SUSPEND LOOP FOR TAPE INITIALIZATION (OPERATORS WOULD	LIN08440
C-----GO BANANAS)	LIN08450
70 IF(IWAY.GT.2) RETURN	LIN08460
C-----SUSPEND UNTIL FILE OR DISK SPACE AVAILABLE.	LIN08470
IF(TTY.GT.0) WRITE(TTY,5940)	LIN08480
C-----INCREMENT SUSPEND COUNT, SUSPEND AND AFTER SUSPEND RE-TRY.	LIN08490
90 NWAIT=NWAIT+1	LIN08500
CALL FROST(2500B,0,NBETA(IWAY),ERRA)	LIN08510
ERRA GO TO (20,40).IWAY	LIN08520
RETURN	LIN08530
C-----WRITE END OF FILE.	LIN08540
90 END FILE UNIT	LIN08550
C-----EMPTY BUFFER IF REQUESTED.	LIN08560
IF(IWAY.EQ.3) CALL EMPTY(UNIT)	LIN08570
C-----TRUNCATE SIZE OF FILE. LOOK UP FILENAME IN ORDERLIB TABLE.	LIN08580
DO 100 I=1,1E	LIN08590
IF(ICBLK(I,1)-NAME) 100,110,100	LIN08600
100 CONTINUE	LIN08610
C-----FILENAME NOT FOUND. TRUNCATION NOT POSSIBLE.	LIN08620
IF(TTY.GT.0) WRITE(TTY,5930) NAME	LIN08630
GO TO 140	LIN08640
C-----RIGHT ADJUST FILENAME FOR FROST CALL.	LIN08650
110 INAME=NAME	LIN08660
DO 120 K=1,10	LIN08670
IF((INAME.INT.778).NE.0) GO TO 130	LIN08680
120 INAME=INAME.SHL.54	LIN08690
C-----FILENAME IS COMPLETELY ZERO	LIN08700
IF(TTY.GT.0) WRITE(TTY,5930) NAME	LIN08710
GO TO 140	LIN08720
C-----LOAD RIGHT ADJUSTED FILENAME AND FILE LENGTH FOR FROST CALL	LIN08730
130 IBETA(1)=INAME	LIN08740
IBETA(2)=(ICBLK(4,1).SHR.36).INT.37777777B	LIN08750
CALL FROST(1200B,0,IBETA,ERRB)	LIN08760
GO TO 140	LIN08770
C-----ERROR DURING FILE LENGTH TRUNCATION.	LIN08780
ERRB IF(TTY.GT.0) WRITE(TTY,5930) NAME	LIN08790
C-----CLOSE FILE (WRITE PROTECTED).	LIN08800
140 CALL DEVICE(6HDCLOSER,NAME,IERR)	LIN08810
C-----CHECK FOR ERROR DURING CLOSING.	LIN08820
IF(IERR.LE.0) GO TO 150	LIN08830
C-----CLOSING ERROR. WRITE ERROR MESSAGE.	LIN08840
IF(TTY.GT.0) WRITE(TTY,5910) NAME	LIN08850
RETURN	LIN08860
C-----SET ERROR FLAG TO STANDARD O. K. RETURN.	LIN08870
150 IERR=0	LIN08880
RETURN	LIN08890
C-----IS THERE A BACKUP FILENAME....	LIN08900
160 IF(UNIT.NE.BLANK) GO TO 170	LIN08910
C-----NO. DESTROY FILE AND DO RENAME WITHOUT BACKUP FILE.	LIN08920
CALL DEVICE(7HDESTROY,NAME,IERR)	LIN08930
GO TO 190	LIN08940
C-----YES. DESTROY BACKUP FILE.	LIN08950
170 CALL DEVICE(7HDESTROY,UNIT,IERR)	LIN08960
C-----IS THERE A SECOND FILENAME....	LIN08970
IF(NAME.NE.BLANK) GO TO 180	LIN08980
C-----NO. USE THIRD FILENAME AND BACKUP FILENAMES.	LIN08990
CALL CHN(LENGTH,UNIT,IERR)	LIN09000

IF(IERR.GT.0.AND.TTY.GT.0) WRITE(TTY,5920) LENGTH,UNIT	LN09010
RETURN	LN09020
C-----YES. CHANGE SECOND FILENAME TO BACKUP FILENAME.	LN09030
180 CALL CHN(NAME,UNIT,IERR)	LN09040
IF(IERR.GT.0.AND.TTY.GT.0) WRITE(TTY,5920) NAME,UNIT	LN09050
C-----IF NO THIRD FILENAME OPERATION WAS JUST SECOND FILENAME TO BACKUP.	LN09060
IF(LENGTH.EQ.BLANK) RETURN	LN09070
C-----CHANGE THIRD FILENAME TO SECOND FILENAME.	LN09080
190 CALL CHN(LENGTH,NAME,IERR)	LN09090
IF(IERR.GT.0.AND.TTY.GT.0) WRITE(TTY,5920) LENGTH,NAME	LN09100
RETURN	LN09110
C-----DESTROY FILE.	LN09120
200 CALL DEVICE(7HDESTROY,NAME,IERR)	LN09130
RETURN	LN09140
5900 FORMAT(1X,A10,2BH FILE INITIALIZATION ERROR)	LN09150
5910 FORMAT(1X,A10,19H FILE CLOSING ERROR)	LN09160
5920 FORMAT(1X,A10,4H TO ,A10,22H FILENAME CHANGE ERROR,16)	LN09170
5930 FORMAT(1X,A10,34H ERROR DURING FILE SIZE TRUNCATION)	LN09180
5940 FORMAT(12X,12HWILL SUSPEND)	LN09190
5950 FORMAT(1X)	LN09200
5960 FORMAT(12X,24HCANNOT PERFORM OPERATION)	LN09210
END	LN09220
SUBROUTINE WAIT(LCIOC,IERR)	LN09230
C	LN09240
C DELAY UNTIL ALL I/O IS COMPLETED ON UNIT DEFINED BY LCIOC	LN09250
C	LN09260
INTEGER TTY,TESTDK	LN09270
COMMON/UNITS/TTY	LN09280
10 NOC=TESTDK(K)	LN09290
IF(NOC) 20,40,30	LN09300
20 CALL DELAY(LCIOC)	LN09310
GO TO 10	LN09320
30 IERR=1	LN09330
IF(TTY.GT.0) WRITE(TTY,5900)	LN09340
RETURN	LN09350
40 IERR=0	LN09360
RETURN	LN09370
5900 FORMAT(29HWAIT....TROUBLE WITH DISK I/O)	LN09380
END	LN09390

Acknowledgments

Many people contributed useful suggestions for improving and correcting the 1977 version of program LINEAR; in particular the author would like to acknowledge the contributions of W. Rothenstein, the Technion, Haifa, Israel, L. B. Levitt, Brookhaven National Laboratory, and the staff of the Argonne Code Center.

References

1. D. Garber et al., *Data Format and Procedures for the Evaluated Nuclear Data File, ENDF*, Brookhaven National Laboratory, Upton, New York, Rept. BNL-NCS-50496 (ENDF-102), 1972).
2. D. E. Cullen et al., "Tabular Cross Section Generation and Utilization Techniques," in *Proc. Nuclear Cross Sections and Technology Conference* (National Bureau of Standards, Washington, D.C., March 1975), p. 419-421.