ANL-7356

ANL-7356

MASTER

Argonne National Laboratory

The Codes NURF, COMBO, and TWOSORCE
for Processing Foil Counting Data

by

George S. Stanford

DISCLAIMER

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

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona	The University of Iowa	Northwestern University
Carnegie Institute of Technology	Kansas State University	University of Notre Dame
Case Institute of Technology	The University of Kansas	The Ohio State University
The University of Chicago	Loyola University	Purdue University
University of Cincinnati	Marquette University	Saint Louis University
Illinois Institute of Technology	Michigan State University	Washington University
University of Illinois	The University of Michigan	Wayne State University
Indiana University	University of Minnesota	The University of Wisconsin
Iowa State University	University of Missouri	

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America

Available from

Clearinghouse for Federal Scientific and Technical Information

National Bureau of Standards, U. S. Department of Commerce

Springfield, Virginia 22151

Price: Printed Copy \$3.00; Microfiche \$0.65

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

The Codes NURF, COMBO, and TWOSORCE
for Processing Foil Counting Data

by

George S. Stanford

Reactor Physics Division

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

August 1967

THIS PAGE
WAS THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	7
PART ONE. GENERAL DISCUSSION OF THE <u>NURF</u> CODE	
I. INTRODUCTION.	8
II. FEATURES.	9
III. MATHEMATICAL METHOD.	11
A. Fission-product Fit	11
B. Decay-constant Fit by Least Squares	12
C. Processing with Fixed Decay Constant	12
D. Deadtime Corrections	13
E. Background Corrections	13
F. Weight Corrections.	14
G. Accidental-coincidence Corrections.	15
H. Average Foil Activations	16
I. Error Analysis	16
IV. PERFORMANCE	18
V. DISCUSSION	19
PART TWO. <u>NURF</u> USERS' MANUAL	
VI. INPUT CARDS.	20
A. Composition of the Input Deck.	20
B. Control Deck.	20
C. Weight-card Deck.	20
D. Counting-data Deck.	22
VII. MISCELLANEOUS INFORMATION FOR <u>NURF</u> USERS.	23
A. Collation.	23
B. Simultaneous Irradiation	24
C. Grouping Foils for Counting	24
D. Input Time Information	24
E. Background.	25
F. Accidental Coincidence Corrections.	26
G. Intercalibration "Weight" Cards from COMBO.	26
H. Weighted-average Cards for Use in COMBO	26
I. Preset Count versus Preset Time	27
J. Counting Rates and Deadtime	28

TABLE OF CONTENTS

	<u>Page</u>
VIII. DESCRIPTIVE COMMENTS ON THE <u>NURF</u> PROGRAM AND SUBROUTINES	28
A. List of Subprograms	28
B. Program NURF	29
C. Subroutine CONIN.	29
D. Subroutine WTCARDS	29
E. Subroutine DCARDS	29
F. Subroutine REVERSER	29
G. Subroutine ELMER.	30
H. Subroutine NIMFO	30
I. Subroutine MINFAC	30
J. Subroutine ITER 7	30
K. Subroutines LSQPOL and MATINV.	30
L. Function EXDIF(X)	30
M. Subroutine NEWPAGE	33
N. Subroutine DATOUT	33
O. Subroutine GRAFFIT.	33
PART THREE: <u>COMBO</u> : A CODE FOR NORMALIZING AND COMBINING LARGE AMOUNTS OF FOIL- COUNTING DATA	
IX. INTRODUCTION.	35
X. OPERATION OF THE PROGRAM	36
XI. INTERCALIBRATED SETS OF FOILS	38
XII. CARD PREPARATION FOR <u>COMBO</u>	38
PART FOUR. THE <u>TWOSORCE</u> PROGRAM FOR DETERMINING COUNTING SYSTEM DEADTIME	
XIII. INTRODUCTION.	40
XIV. PRINCIPLE OF THE DOUBLE-SOURCE METHOD	41
XV. OPERATION OF PROGRAM <u>TWOSORCE</u>	42
XVI. DISCUSSION OF DEADTIMES	43
XVII. CARD PREPARATION FOR PROGRAM <u>TWOSORCE</u>	44

TABLE OF CONTENTS

	<u>Page</u>
APPENDIX A. ITER 7, A Directed-search Subroutine for Finding Minima	45
Abstract	45
1. One-dimensional Applications	45
2. Multidimensional Applications	48
3. Discussion	51
APPENDIX B. FORTRAN Listing for the NURF Code	52
1. NURF	52
2. CONIN	55
3. WTCARDS	58
4. DCARDS	60
5. REVERSER	63
6. ELMER	64
7. NIMFO	64
8. MINFAC	65
9. EXDIF	65
10. NEWPAGE	65
11. DATOUT	66
12. GRAFFIT	68
APPENDIX C. FORTRAN Listing for COMBO	69
1. COMBO	69
2. NORMGRUP	74
APPENDIX D. FORTRAN Listing for TWOSORCE	77
APPENDIX E. Tables	81
ACKNOWLEDGMENTS	87
REFERENCES	87

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Flow Diagram for NURF	10
2.	Flow Diagram for Subroutine WTCARDS.	31
3.	Flow Diagram for Subroutine DCARDS	32
4.	An Example of the Output Produced by Subroutine GRAFFIT .	34
A-1.	FORTRAN Listing for ITER 7.	47
A-2.	Example of a Three-dimensional Application of ITER 7.	50

LIST OF TABLES

<u>No.</u>	<u>Title</u>	<u>Page</u>
I.	Description of NURF Control-deck Cards	81
II.	Abridged Control-deck Description, for Routine Problems (NURF)	83
III.	Description of Cards for Weight-card Deck	84
IV.	Examples of Weight-card Decks.	84
V.	Description of Counting-data Cards	85
VI.	Data Deck for COMBO	85
VII.	Data Deck for Program TWOSORCE.	86

The Codes NURF, COMBO, and TWOSORCE
for Processing Foil Counting Data

by

George S. Stanford

ABSTRACT

For the routine processing of foil counting data, the codes NURF and COMBO have been written in FORTRAN for the CDC-3600, providing a variety of input and processing options. A novel feature of the NURF code is its method for treating fission-product counting data. In making decay corrections for a set of fission foils, the traditional method has been to count simultaneously a reference foil that was irradiated along with the others. Although that method gives satisfactory results, it has several drawbacks in practice. The NURF code permits one to dispense with the reference foil, provided that each foil of the set is counted several times. It converts the counting rates and elapsed times (after irradiation) to logarithms, and synthesizes a decay curve by means of a least-squares polynomial fit. This step is iteratively alternated with one in which the set of counts of each foil is multiplied by the factor that causes that set best to fit the previously found curve. The results using this method agree, within counting statistics, with those of the more cumbersome reference-foil method.

The COMBO code is used in conjunction with NURF to perform further normalizing and combining tasks. The program is divided into two parts. The first part provides weighted averages when the same group of foils has been counted on more than one scaler, in more than one system, or at several different times. The second part is for use in internormalizing the data from an irradiated set of foils that has been separated into two or more groups, for counting on different channels, in different systems, or at different times. Each group must include one or more foils that are common to some other group.

For preparing intercalibrated sets of foils, the program can be requested to punch out weight cards for use as input to NURF.

When counting rates are greater than approximately 4,000 per second, it has been desirable to measure the deadtime for each experiment. The TWOSORCE code facilitates the routine determination of the counting-channel resolving times, using the paired-source technique.

An appendix to this report describes a FORTRAN directed-search subroutine for determining the minimum of a nondifferentiable function.

PART ONE

GENERAL DISCUSSION OF THE NURF CODE

I. INTRODUCTION

In nuclear reactors it is important to know the fission-rate distribution within the fuel zone. Since the most direct and convenient way to measure this distribution is by irradiating foils, critical experiments tend to generate large numbers of activated fission foils. Each set of such foils must be counted after being activated, with correction made in some manner for the decay that occurs during counting.

Traditionally, the fission-product decay correction has been made by means of a reference foil which is irradiated along with the samples whose relative activations are to be determined. Since the reference foil decays at the same rate as the other foils, the relative fission rates are obtained by dividing each observed count rate by the simultaneously obtained count rate of the reference foil. The method gives satisfactory results, but it has several drawbacks: The counting system must contain at least two counting channels, at least one of which is preempted by the reference foil; care must be taken that all counting channels have the same bias settings; additional error is introduced by the counting statistics of the reference foil; and the reference foil is an extra experimental complication.

If each foil of a set is counted more than once--preferably at least three or four times--the need for a reference foil can be eliminated by determining the decay curve from the successive counts of the individual foils. The NURF (No Uranium Reference Foil) code (RP-385X) has been written to accomplish this. The language used is CDC-3600 FORTRAN.

Instead of the fission-product treatment just mentioned, two other methods of data processing are available: The user can elect to have the code do a similar least-squares fit to determine decay parameters and relative activities for a set of single-half-life foils, or to have his data used in the standard calculation of saturated activities, using the input half-life of the nuclide being counted.

To let the program be of general utility, most of the features and options of the foil-data code RP-202 (Ref. 1) have been incorporated into NURF. Corrections for background and deadtime are made, and a $2\tau_1\tau_2$ accidental-coincidence correction is available. The output provides a graphical display, averaging of the results for each foil, and statistical error analysis.

Reference 2 contains a preliminary description of the NURF code.

II. FEATURES

In the assembling of the NURF program, an attempt was made to achieve a high degree of flexibility, so that the data could be processed with minimum handling. As already mentioned, three major processing options are provided. In this section, other features of the code will be discussed in outline, with more detail in Chapter III and Part Two.

The automatic counting system used in the ANL Reactor Physics Laboratory punches the counting data directly on IBM cards. Each set of foils is cycled repetitively, typically with a blank background planchet at the end of each cycle. The background counts are automatically identified as such.

The input cards for NURF consist of three groups: a five-card control deck, a set of weight cards, and the counting-data cards. The control deck contains such input constants as: the approximate half-life, for a decay-constant least-squares fit, or the accurate half-life if a fixed decay constant is to be used; coefficients for use in the series-expansion correction for deadtime; duration of irradiation; time elapsed between the end of irradiation and the start of counting; counter efficiencies and deadtimes; number of coefficients to use in the least-squares fit (maximum of ten); an indication of the type of weight correction desired; the cadmium ratio (if relevant to the weight correction); an indication as to whether the stack of foils was counted in the same order each time (as opposed to reverse order on successive cycles); and convergence criteria for the least-squares fit and for the iterative determination of the A_f (Eq. 3).

The weight-card deck contains the identifiers for the individual foils, the corresponding weights (or intercalibration factors), and an indication of the foils that were cadmium-covered (for use in weight corrections for gold and indium, where epicadmium self-shielding is taken into account).

Each card of the counting-data deck contains the clock time, the duration of the count, the number of pulses recorded by each of up to six scalers, and an indication for each scaler as to whether the recorded count is to be treated as background. Dimensioning is such that the program can accommodate up to 200 foils and up to 200 counts per foil, subject to the restriction that only the first 400 data points for each scaler, exclusive of background, will be processed. The data deck does not need to terminate at the end of a counting cycle.

As indicated in Fig. 1, the data are processed one scaler at a time. The corrected relative activity is printed out for each count, grouped according to foil. Then, as in RP-202, for each foil the results are averaged in two ways: weighted according to the counting statistics, and unweighted.

Printed out along with these averages are the statistical error in the weighted average, the standard deviation in the unweighted average, and the ratio of those two errors--this last being useful as an indicator of possible trouble. Finally, the corrected counting rates are presented graphically, grouped foil by foil--again for the main purpose of helping to isolate any bad counts.

If desired, the weighted average specific activities will be punched out on IBM cards, for subsequent processing by the normalizing and combining program COMBO (see Section VII-H and Chapter XII).

III. MATHEMATICAL METHOD

A. Fission-product Fit

The input data for NURF consist chiefly of the raw counts, the count duration, the elapsed time since irradiation, and background and deadtime information. The count rates, corrected for background and deadtime, are computed. When a fission-product fit is to be made, both the rates and the elapsed times are converted to natural logarithms, producing for each foil a decay curve that is roughly linear. The linear approximation has the form

$$\ln R = A + B \cdot \ln T, \quad (1)$$

where $B \approx -1.2$ if R is in counts per second, and T , in seconds, is the time from the midpoint of the irradiation.^{9,10}

A_f , an average value of A , is found for each foil, with B set at -1.2 . Then a least-squares polynomial fit is made to the function

$$r_{if} - A_f = a + Bt + Ct^2 + Dt^3 + \dots, \quad (2)$$

where i is the count number; f is the foil number; $r = \ln R$; and $t = \ln T - \ln T_1$, T_1 being the time of the first count of the first foil. Using the resulting values of B , C , D , ... (the value of a being both small and, at this point, irrelevant), the code applies a directed-search subroutine (ITER 7--see Appendix A) to the set of counts for each foil, to obtain that new value of A_f which produces a least-squares fit to

$$r_{if} - A_f = Bt + Ct^2 + Dt^3 + \dots, \quad (3)$$

where B , C , D , ... are kept fixed.

With these new values of the A_f , the least-squares fit to Eq. 2 is repeated, followed by calculation of a new set of A_f , and so on, the final

calculation being a fit to Eq. 2. Two to four of these iterative cycles usually produce a fit that is adequately close to the asymptotic one, the criterion for "adequately close" being a fractional change of less than 0.004 in the weighted sum of the squares of the residuals.

The corrected relative counting rates are now given by the ratio of the observed rate to the fitted curve, i.e., by $\exp(\alpha_i)$, where

$$\alpha_i \equiv r_i - (a + Bt + Ct^2 + Dt^3 \dots). \quad (4)$$

In all the averaging and fitting procedures, each data point is weighted according to the inverse first power of the statistical precision, rather than to the inverse second power. The basis for this is the assumption that counting statistics are not the only source of random errors.

B. Decay-constant Fit by Least Squares

The above procedure can be used equally well to discover a decay constant λ for a single-half-life isotope. The only differences are: (a) The times are not converted to logarithms; (b) an approximate value of $-\lambda$ is used for B in Eq. 1; and (c) only a two-coefficient (linear) fit will normally be made: the second coefficient will be $-\lambda$ as calculated by the code.

To date, sets of foils of gold, indium, manganese, and copper have been processed in this way, and comparisons with the results of fixed-decay-constant processing have been entirely satisfactory, as regards relative activation within sets. However, since the calculated values of λ are subject to influence by resolving time uncertainties, drift in the electronics, and perhaps other variables, these decay constants should not be used except for comparing foils that are both (a) of the same material; and (b) irradiated, counted, and processed together. For other cases, the accepted values of the decay constants (see, for example, Ref. 3) are to be used, with fixed-decay-constant processing (see C below).

For the record, manganese activation data have been successfully processed by NURF with the times converted to logarithms, and likewise, fission-product counts have been handled satisfactorily without the times converted to logarithms (four coefficients in each case). However, times near zero (i.e., less than an hour after the termination of the irradiation) were not present.

C. Processing with Fixed Decay Constant

When the fixed-decay-constant option is used, the least-squares portion of the program is bypassed, and the saturated activity A_{sat} is calculated according to

$$A_{\text{sat}} = \lambda \frac{N}{E} \left[e^{-\lambda t_s} (1 - e^{-\lambda t_i}) (e^{\lambda t_c} - 1) \right]^{-1}, \quad (5)$$

where N is the number of counts, corrected for background and deadtime, recorded during the counting interval t_c ; E is the efficiency of the counter; t_i is the length of the irradiation; t_s is the time between the end of the irradiation and the end of the counting interval; and λ is the decay constant of the sample.

Equation 5 contains no approximations. The neutron flux ϕ during the irradiation is assumed constant, so that $A_{\text{sat}} = V \Sigma_a \phi$, where V is the volume of the foil and Σ_a is the macroscopic neutron-absorption cross section. The saturated activation (number of activated atoms per unit weight at saturation) is equal to A_{sat}/λ ; it can be obtained from the code by using, in place of the counter efficiency E , the quantity λE .

D. Deadtime Corrections

The actual observed counting rate R_a is corrected for counter resolving-time losses by means of the formula

$$R_c = R_a \left[1 + R_a \tau + B(R_a \tau)^2 + C(R_a \tau)^3 + D(R_a \tau)^4 \right], \quad (6)$$

where τ is the resolving time, and R_c is not yet corrected for background. The correct choice for the coefficients B , C , and D depends on the counting system.

For a "nonparalyzable" counter, where there is a fixed resolving time τ for each event counted, the exact expression is $R_c = R_a / (1 - R_a \tau)$, and the coefficients of the series are all unity. For a "paralyzable" counter, which is dead for a time τ after each pulse, whether it is counted or not, $R_c = R_a e^{R_c \tau}$; in this case, the coefficients B , C , and D are $3/2$, $8/3$, and $125/24$, respectively.⁴

E. Background Corrections

In the course of counting a set of samples, background counts are typically taken periodically. Following the method of RP-202 (Ref. 1), NURF permits the user to select either of the following options:

1. An identification code (IDC) number of 2 on the background data card causes the subsequent background rate R_B for the corresponding scaler to be determined solely by that card; i.e.,

$$R_B = \frac{C_B}{t_B}, \quad (7)$$

where C_B is the number of background counts recorded in the counting interval t_B .

2. An IDC number of 1 causes the background rate (except for the first background card) to be recalculated according to the formula

$$R_B = KR_B^{\circ} + \frac{(1-K) C_B}{t_B}, \quad (8)$$

where R_B° is the previous background rate, now being replaced by R_B , and $K(0 < K \leq 1)$ is a weighting factor specified by the user. It can be seen that if $K = 1$, all background counts after the first are ignored; if $K \approx 0$, R_B° is ignored. Thus any desired compromise can be made between the assumption that the background is constant and the assumption that the background is systematically changing with time.

The background rate is subtracted from the counting rate after correction for deadtime.

F. Weight Corrections

The NURF code provides two weight-correction options: (1) the usual method, where the activation is merely divided by the weight of the foil to obtain the specific activation; and (2) a special procedure suitable for foils such as gold or indium with a very large epicadmium resonance.

The latter procedure is based on the assumption that weight variations within a set of foils are due to differences in thickness rather than area, so that self-shielding of neutrons near the resonance energy must be considered. For bare foils (i.e., those irradiated without cadmium covers), the thermal and epithermal components of the activation must be treated separately; this can be done if the cadmium ratio CR is approximately known.

If A is the specific epicadmium activation for a foil of thickness t , a function $H(T)$ will be defined by

$$H(T) = \frac{t}{A} \frac{dA}{dt}. \quad (9)$$

Let the nominal weight for a set of foils be w_0 , corresponding to a thickness t_0 , and let the actual weight of an individual foil be w . Then the fractional deviation f from the nominal weight will be defined by

$$f = 2 \frac{w - w_0}{w + w_0} = 2 \frac{t - t_0}{t + t_0} \quad (10)$$

It can be shown⁵ that

$$A_0 \approx A \frac{2 + fH(t_0)}{2 - fH(t_0)}, \quad (11)$$

where A_0 is the specific epicadmium activation, corrected to thickness t_0 . Equation 11 is used by NURF in performing weight corrections for cadmium-covered foils, when the gold/indium option has been selected. The formula used for bare foils is

$$A_{0b} = A_b \left\{ 1 - \frac{2fH(t_0)}{[2 + fh(t_0)] CR} \right\}^{-1} \quad (12)$$

An implicit assumption in Eq. 12 is that the thermal self-shielding and flux depression are negligible, for weight-correction purposes, even though they might still be important considerations in extrapolating to zero-thickness specific activations. As a rule of thumb, the approximations involved in the above weight-correction procedure start to lose their validity for $f > 10\%$.

Some values of $H(t_0)$ for gold foils are given in the following tabulation:

t_0 , mils	0.05	0.1	0.2	0.5	1.0	5.0	20
$H(t_0)$ (gold)	0.10	0.17	0.24	0.36	0.42	0.48	0.50

For indium, the value of $H(t_0)$ can be expected to be approximately the same as for gold of the same $t_0 \Sigma_a$, where Σ_a is the macroscopic neutron-absorption cross section at the peak of the resonance.

G. Accidental-coincidence Corrections

If one of the counting channels is used to record coincidences between events in two of the other channels, NURF can correct for accidental coincidences according to the formula

$$T_c = \frac{R_C T_A T_B}{R_A R_B} - 2\pi T_A T_B - (B_C - 2\pi B_A B_B), \quad (13)$$

where

T_c is the true coincidence rate due to events occurring in the sample;
 R_C , R_A , and R_B are the observed counting rates in Channels C (the coincidence channel), A, and B, respectively;

B_C , B_A , and B_B are the observed background counting rates;

T_A and T_B are obtained by correcting R_A and R_B for deadtime losses;

and

τ is the resolving time of the coincidence circuit.

H. Average Foil Activations

From the set of counting data for each foil, both a weighted and an unweighted average specific activation, A_w and A_u , respectively, are calculated. To compute the former, the code weights each determination of the specific activation A_j according to the inverse square of the statistical standard deviation, s_j . If E_j is the relative error (as defined below in Eq. 19), then the weighting factor, w_j , is given by

$$w_j = \frac{1}{s_j^2} = (E_j A_j)^{-2}. \quad (14)$$

If N is the number of times a given foil was counted, then

$$A_w = \frac{\sum w_j A_j}{\sum w_j}, \quad (15)$$

and

$$A_u = \frac{1}{N} \sum A_j, \quad (16)$$

where all summations are from $j = 1$ to $j = N$.

I. Error Analysis

1. Error in Least-squares Coefficients

The relative errors E_i in the coefficients C_i for the least-squares fit are derived from the diagonal elements M_{ii} of the error matrix calculated by the LSQPOL subroutine. The formula used is⁶

$$E_i = \frac{1}{|C_i|} \sqrt{\frac{M_{ii} S}{N - N_c}}, \quad (17)$$

where S is the weighted sum of squares of the residuals, N is the number of data points, and N_c is the number of coefficients.

2. Statistical Error in Counting Rate

The statistical standard deviation σ_1 in the counting rate (where n is corrected for deadtime losses but not for background) is calculated according to⁷

$$\sigma_1 = \sqrt{\frac{n(e^{n\tau} - 2n\tau)}{t_c(1 - n\tau)^2}}, \quad (18)$$

where τ is the deadtime and t_c is the counting interval. This formula is accurate for the case of the paralyzable counter (see Section III-D) and is satisfactory for other cases, provided $n\tau$ is not very large.

If R is the counting rate corrected for background ($R = n - R_B$, where R_B is the background rate), then the relative statistical standard deviation E_R in R is given by

$$E_R = \frac{1}{R} \sqrt{\sigma_1^2 + \sigma_B^2}, \quad (19)$$

where σ_B is the standard deviation in the background rate. E_R also is the relative error in the quantities that are derived from R , such as the saturated activation and the specific activity.

3. Statistical Error in Background Rate

For those calculations of the background rate R_B where the previously stored value of R_B is not included (see Section II-E), $\sigma_B^2 = C_B/t_B^2$.

When R_B is calculated according to Eq. 8, then

$$\sigma_B^2 = K\sigma_{0B}^2 + \frac{(1 - K) C_B}{t_B^2}, \quad (20)$$

where σ_{0B} is the preceding value of σ_B .

4. Error in the Weighted Average Activation

The weighted average activation for a foil, A_w , is calculated according to Eq. 15. The statistical error s_w in A_w is given by

$$s_w = \frac{1}{\sum w_j} \sqrt{\sum (w_j s_j)^2} = (\sum w_j)^{-1/2}, \quad (21)$$

and the relative statistical error E_w in A_w is given by

$$E_w = \frac{s_w}{A_w} = \frac{1}{A_w \sqrt{\sum w_j}} \quad (22)$$

The summations are from $j = 1$ to $j = N$, where N is the number of times the foil was counted.

5. Error in the Unweighted Average Activation

Equation 16 gives the unweighted arithmetic mean A_u of the specific activation determinations for each foil. NURF calculates the relative standard deviation E_u in this average from the scatter of the data according to

$$E_u = \frac{1}{A_u} \sqrt{\frac{\sum (A_j - A_u)^2}{N(N-1)}} \quad (23)$$

Included in the printed output from NURF is the ratio E_u/E_w for each foil, and also the average of these ratios. This information gives the user an indication as to whether random nonstatistical errors are affecting his results. At this time, a rigorous chi-squared test has not been incorporated into the code.

IV. PERFORMANCE

The performance of NURF's least-squares procedure has been entirely satisfactory. Comparisons of the results of processing the same fission-product data by NURF, using the least-squares process, and by RP-202, using a reference foil, have failed to reveal any problem areas. The relative activations of a set of foils as determined by NURF's fitting procedure have invariably agreed with the RP-202 reference-foil results to an extent consistent with statistical errors, most of which have been less than 0.2%. For single-half-life foils, similar agreement has been found between the relative activations as determined with least-squares fitting and with fixed-half-life processing.

Regarding the standard deviation in repeated counts of the same foil, where single-half-life foils are concerned, the results for the two methods of processing have been comparable and usually consistent with the statistical uncertainties. For the fission foils, however, the standard deviations in the least-squares results have averaged perhaps 20% less than for the reference-foil method, probably because in the latter case the counting statistics for the reference foil are a contributing factor. At the same time,

the standard deviations for the fission foils have been slightly greater than would be expected on the basis of the counting statistics. This is partly, at least, because the number of coefficients used (four) has sometimes been less than optimum when counting has gone on for a long time. We have found, nevertheless, that the calculated relative activities are not very sensitive to the degree of the polynomial used for the least-squares fit.

As a matter of interest, four-coefficient fits to fission-product counting data have yielded values near -1.25, -0.5, and +0.25 for the second, third, and fourth coefficients, respectively. However, there has been considerable scatter in these numbers.

The extra computer time due to the iteration procedure is not large. In a test run, on the CDC-3600, a problem with an average of 100 data points for each of six scalers was processed with a six-coefficient fit in a total time of 1 min 6 sec.

V. DISCUSSION

The principal novel feature of the NURF general-purpose foil data code is its provision for eliminating the reference foil in determining the relative activations of a set of fission foils. In removing the need for the reference foil, the method provides increased statistical accuracy, frees counting equipment for more useful work, and makes the counting bias uncritical--provided that it is high enough (≥ 500 keV, if gamma counting, for uranium) that there is no significant interference from the products of radiative capture--gamma rays, beta rays, or bremsstrahlung.

Additionally, and this applies to both fission and single-half-life foils, an empirically determined decay rate provides first-order cancellation of the effects of any long-term gain drifts in the counting system. Certain background, deadtime, and contamination problems can sometimes be masked by a fit of the third degree or higher, but even these will usually show up if the foils in the set include more than a narrow range of activities.

The chief disadvantage of the least-squares method is that it is not well suited for absolute counting. For this reason, NURF provides an option permitting the conventional use of a fixed decay constant.

PART TWO

NURF USERS' MANUAL

VI. INPUT CARDS

A. Composition of the Input Deck

The set of input cards for a NURF problem on the CDC-3600 consists of the following, where it is assumed that the program is on a library tape, to be mounted on tape unit No. 35:

1. Sequence card
2. Job card
3. Accounting card
4. $\frac{7}{9}$ +MOUNT,Lxxxxx
where the xxxxx field contains the library number for the current tape
5. $\frac{7}{9}$ EQUIP,35=SV,(,)
where the parentheses contain the tape identifier and edition number
6. $\frac{7}{9}$ LOAD,35
7. Run card
8. Input data, consisting of: Control deck
Weight deck
Counting data deck
Blank card
9. Further input-data decks, as in 8, as desired
10. $\frac{7}{8}$ END OF FILE.

B. Control Deck

The input constants that determine the way the problem is to be run are contained in the five-card control deck. Table I* gives a complete listing of the parameters on those cards; Table II contains an abridged description, to facilitate the preparation of control decks for routine problems.

*For the convenience of the user, all tables in this report are included in Appendix E.

The cards of the control deck are read by the CONIN subroutine. That subroutine has been written so as to make it unnecessary, for the most part, to enter the values of parameters that tend to be the same for most problems. Those cases are indicated in Tables I and II, where the standard values corresponding to blank (or zero) entries are given.

One could conceive of a problem (South counting room, fission fit, etc.) that could be run successfully with a control deck consisting of five completely blank cards. However, to avoid confusion in organizing the cards, it is suggested that blank title cards not be used as a general practice, and that cards 2 through 5 contain their card number in column 1.

C. Weight-card Deck

The control deck must be followed by a weight-card deck, which is composed of the types of cards described in Table III. The weight-card is assembled according to the following considerations: A set of the WEIGHT cards, or substitute, is required for each scaler for which the counting data are to be analyzed, that is, for each scaler, whose number I is less than IMAX, that is not deleted by a nonzero entry for NS(I). If the scaler is deleted by setting IDC(I) to 4, a set of WEIGHT cards, or substitute, is required. A completely blank card can be used as a WEIGHT card (but not as one of the other types in the weight-card deck).*

As a substitute for a set of actual WEIGHT cards, either a REPEAT card or a NONE card can be used. The REPEAT card causes the weight data for the preceding scaler to be repeated; it can follow a set of WEIGHT cards, a NONE card, or another REPEAT card. It must not be the first card in the weight deck.

When a set of WEIGHT cards is not needed for the experiment, a NONE card may be used, with the number of foils (excluding background) punched, with a decimal point, in the foil-weight field. Each foil is then assigned a weight of 1.0 by the code, and the foil identifiers (NAME) are left blank. A NONE card may be followed by one or more REPEAT cards, if the same number of foils was counted in the succeeding scalers.

If each of two successive scalers has its own set of actual WEIGHT cards, these two sets must be separated by a NEWSET card. A NEWSET card must not be used if there is an intervening NONE or REPEAT card, nor may it be used as the first card of a weight-card deck.

*A minor drawback in using a blank WEIGHT card is the risk that, if a fatal error should occur before that card, it could cause the program to interpret the following card as the title card for a new problem, and thus lead to an abnormal termination, preventing the processing of any subsequent problems.

Every weight-card deck must be terminated by a LAST+1 card.

Four examples of possible weight-card decks are given in Table IV.

The maximum number of foils, for any scaler, is 200.

D. Counting-data Deck

The counting-data cards are produced by the automatic counting equipment. Frequently the counting-data deck can be removed from the card punch and placed behind the weight-card deck for processing, without further handling. At other times, various manipulations are desired.

The cards are described in Table V, where the items are self-explanatory except for the IDC (identification code) numbers. There is an IDC column for each scaler, and the significance of the possible entries is as follows:

IDC(I) = 0, blank, 3, 5, 7, or 9: Scaler I contains valid, nonbackground counting data.

IDC(I) = 1: Scaler I contains a background count. If this is the first background card for Scaler I, the background rate will be calculated according to Eq. 7. If this is not the first background card for Scaler I, the background rate will be calculated according to Eq. 8, making use of the background weighting factor BKW.

IDC(I) = 2: Scaler I contains a background count, and the background rate will be calculated according to Eq. 7.

IDC(I) = 4: Data for Scaler I will not be processed for this or for subsequent cards, until cancelled by IDC(I) = 8.

IDC(I) = 6: The datum for Scaler I on this card will be deleted. (See Section VII-A.)

IDC(I) = 8: The data for Scaler I are reinstated, if previously deleted by IDC(I) = 4, starting with the following card. Thus the succeeding data can start with a background count.

The counting equipment provides the option of automatically punching IDC(I) = 1 or 2 for the last planchet in the stack. If the foils are counted without restacking ("continuous cycle"), the planchet at each end of the stack will only be counted once for every two cycles, making it convenient to use the end planchets for background, with IDC(I) = 1 or 2 automatically punched out.

The data deck need not contain an integral number of traverses through the stack of foils. If the cards are collated (see below), and there is a partial traverse, the cards for the foils that were counted in the partial traverse must precede the others. If the foils were counted without restacking, the data for a partial even-numbered traverse will not be processed if the cards are uncollated; such data can be salvaged by collating the cards.

Background cards may be inserted in the counting-data deck wherever desired. $IDC(I) = 6$ can be used to avoid adding spurious data to the counting channels to which new background information is not being added.

If the counting-data deck is uncollated, the cards for a partial traverse must always come at the end. However, the cards do not have to be in chronological order, so that if the partial traverse was counted first, the cards for that traverse can be transferred to the end of the deck. It must be kept in mind that the code determines the foil associated with each card by the position of that card in the counting-data deck.

There can be an arbitrarily large number of cards in the counting-data deck, if the cards are uncollated, but only the first 400 nonbackground, nondeleted data points for each scaler will be processed. If the cards are collated, the 400 number must not be exceeded.

The counting-data deck may be terminated by a blank card, which may be followed by the control deck for another set of data or by a $\frac{7}{8}$ END OF FILE card. A blank card does not need to be used before the $\frac{7}{8}$ END OF FILE card.

VII. MISCELLANEOUS INFORMATION FOR NURF USERS

A. Collation

If desired, the counting-data cards may be collated--that is, the cards may be grouped foil by foil, rather than in the original counting sequence. This procedure is convenient when the background rates are different for different foils (see E-3 below). It can also be used to salvage data from a partial, even-numbered traverse, when the foils were counted without restacking (see Section VI-D above). The cards for the foils that were counted the extra time must precede the others. The cards need not be arranged in the order in which they were counted.

If two or more sets of foils are counted simultaneously (i.e., using more than one sample changer), and the data cards are to be collated, there should, of course, be the same number of planchets in each stack. To avoid

problems with a partial traverse, the empty planchets in one stack, corresponding to foils in the other, should come at the end of the traverse. In processing such a deck of collated cards, use of $IDC(I) = 4$ can avoid the necessity for using dummy WEIGHT cards for the empty planchets.

It might occasionally be desired to delete a single wild data point. One way to do this without affecting the code's collation of the data is as follows: (1) Delete the offending point by using $IDC(I) = 6$ (see Section VI-C) for the scaler in question. (2) Since that point will be ignored by the code, a substitute is required. Obtain the substitute point by duplicating one of the other cards for the same foil, deleting the data for all the other scalers by means of $IDC(I) = 6$. (3) Insert the new card in the data deck, next to the one with the bad point.

B. Simultaneous Irradiation

For successful decay correction of fission-product counting data, all foils of a set must have been irradiated simultaneously. (This would also be true for decay correcting by means of a reference foil.) It is often desirable--and sometimes necessary, if any of the foils have been irradiated previously--to determine the counting rates for the individual foils before the irradiation; these cards can be punched with 2's in the IDC columns, for use as the background cards for the postirradiation counting data. It will then be necessary to collate the data cards. See Section E-3.

C. Grouping Foils for Counting

When the least-squares fitting process is used, either for fission products or for single half-life, all the foils of a set should be counted together--or, at the least, with a fair amount of overlap in time. A problem might arise with a set of foils having a large range of activities, where it would be desired to count the weak foils early and the more active ones later. One solution would be to count them all together, and then collate the cards and remove the early counts for the hot foils and the later low counts for the cool foils, leaving the same number of cards for each foil, and being careful that there is a time overlap of at least several cards. However, probably a better solution would be to count the foils in two groups, with some common foils, and use the COMBO program (Part Three) to combine the results. COMBO is also useful when there are too many foils to count conveniently in one set.

D. Input Time Information

When only relative activation rates are desired, it is usually unnecessary to input the irradiation time and the time elapsed between the end of irradiation and the start of counting. If the counter clock is not reset at

the start of counting, and if it contains a spuriously large reading, at least an approximate entry for TCLOCK (columns 56-65 on control card No. 4) is advisable.

The counter clock should be reset before counting begins, if it is likely to run over and start again at zero during the counting. However, if this does happen the data can be saved by manually punching, on each data card, the missing significant digit, or by processing the data cards in two groups, combining the results with COMBO.

E. Background

1. Synthesized Background Rates

When background cards are punched by hand, on the basis of synthesized background rates, it should be kept in mind that the background statistics enter into the error analysis and the weighted averaging. For example, the entry of five background counts in 10 sec is not equivalent to entering 50 background counts for 100 sec, since the standard deviation σ_B in the background rate (see Section III-I-3) will be lower in the second case than in the first. The time and number of counts entered on the background card should be commensurate with the estimated accuracy with which the background rate is known.

2. Background Multiplier

Occasionally an experiment requires that the background rates as determined from the background counting cards be multiplied by some factor before being subtracted from the counting data. This can occur when early counts for a short-lived activity are to be corrected for a background due to a long-lived activity (e.g., lutetium) on the basis of a later count of the same foils. The need has also arisen in cases where foils used for background correction have not had the same weight as the foils whose data are to be corrected.

This background adjustment can be made automatically by entering the desired multiplier in the BKF field, which is on the second card of the control deck, columns 5 through 12.

3. Different Background for Different Foils

In some experiments, the background rates for all the foils are not the same: the background rates for fission foils, for example, are frequently individually determined before irradiation. In such a case, collation of the counting-data cards permits the use of just one background card for each different background rate, placed in the deck just before the

counting data for the corresponding foils. These background cards should have 2's in the IDC columns.

F. Accidental Coincidence Corrections

NURF can be requested to make corrections for accidental coincidences (see Section III-G), by entering 1 in column 40 of control card No. 4. The resolving time of the coincidence circuit is entered in the appropriate DT(I) field of control card No. 5. The numbers of the coincidence channel C and the counting channels A and B are entered on the fifth card. Up to three sets of A, B, and C can be entered. Since the same scaler can appear more than once in these lists (the only restrictions being $A_j \neq B_j$ and $C_1 \neq C_2 \neq C_3$), there is considerable flexibility in the possible coincidence configurations. For example, three coincidence circuits with differing resolving times could be fed by the same two channels A and B. Another possibility would be a triple-coincidence arrangement, by having $C_1 = A_2$, with another coincidence channel, C_2 , recording coincidences between C_1 and B_2 . In a third possible experiment, C_1 could record the coincidences between A_1 and B_1 , while a similar coincidence circuit C_2 detected coincidences between A_1 and a delayed signal from B_1 . C_2 could thus provide a direct measure of the accidental coincidence rate. With the correct background-rate inputs, accidental-coincidence correcting by NURF should, on the average, cancel the observed counting rate in C_2 .

G. Intercalibration "Weight" Cards from COMBO

For some sets of foils, particularly if composed of an alloy or mixture, the relative specific activations are more accurately determined with respect to the activities observed in a separate intercalibration irradiation, than with respect to the actual weight of the foil. In such a case, the counting data from the intercalibration experiment can be processed by NURF, using WEIGHT cards containing the foil identifiers and weights of 1.0, and using the punched-card option [ITCH(6)]. The output cards can then be used as input for the COMBO code, which will perform its normalizing and combining function and produce a set of WEIGHT cards with the correct format for use in the NURF weight deck. (See Chapter XI.)

H. Weighted-average Cards for Use in COMBO

If the user enters 1 for ITCH(6) (column 28 on card No. 4 of the control deck), output cards will be produced in a format suitable for input for the COMBO program. This deck of output cards will begin with two title cards, and end with a NEXT card and a LAST card. One of the latter two is to be discarded, depending on whether or not a deck of cards for another group of foils follows.

If there is only one group of foils for the COMBO problem, meaning that COMBO is being used only to combine the data from several scalers, the NEXT card must be discarded, and the deck may otherwise be used exactly as produced by NURF--including both title cards, the first of which corresponds to the Problem Title card for COMBO, and the second to the Group Title card. If there is more than one group of foils, it will probably be desired that a new Problem Title card be substituted for the one produced by NURF.

If the deck is not for the first group of foils for the COMBO problem, one of the title cards is to be discarded as well as either the NEXT or LAST card.

If, in the scaler-combining process performed by COMBO, all the scalers used are not to be weighted equally, a new Scaler-weight card must be made. This is the third card in the deck produced by NURF; it is described in Table VI.

I. Preset Count versus Preset Time

In counting a set of foils, one should use the preset count option, whenever possible, to terminate the count. (A reasonable preset time must, of course, also be used as a backup, to prevent background and very weak foils from counting indefinitely.) This approach has the following advantages:

1. The counting time is distributed more efficiently among the different foils, since the more active foils are counted rapidly and more time is spent acquiring good statistics for the weaker foils.
2. As the foils decay, the statistics obtained tend to remain constant. This is desirable because in the averaging procedure all counts get approximately the same weight, making the result less sensitive to a possible bad count early in the counting, and also because the comparison of the statistical errors with the observed scatter (see Section III-I-5) is more meaningful.
3. One further advantage applies particularly to fission-product counting. The decay rate of a fission-product mixture diminishes as time proceeds, so that using the preset count capability tends, to some extent, to keep the counting time commensurate with the rate of decay. However, even for fixed-half-life foils, the preset-count termination causes the early cycles to proceed rapidly, leading to a more efficient use of the available counting time.

J. Counting Rates and Deadtime

Depending on the accuracy desired, it becomes increasingly important to pay attention to the deadtime assigned to each scaler as the counting rate rises above 4,000 or 5,000 counts/sec. Quantitatively, if R is the counting rate and $\Delta\tau$ is the uncertainty in the deadtime, then the resulting fractional uncertainty in the corrected counting-rate determination is $R\Delta\tau$.

With the counting equipment currently in use at this Laboratory, the actual deadtime is a function of many variables. (This is undoubtedly true of most counting arrangements.) If it must be known accurately (i.e., with an uncertainty $\approx \pm 2 \mu\text{sec}$), it must be measured using the same isotopes and the same bias settings, preferably when the counting is done.

A computer code (TWSORCE) is available for the routine determination of the deadtime by means of the paired-source technique. The program is discussed in Part Four.

Differential counting, or integral counting with a high bias setting, can be misleading with regard to estimating deadtime errors, since unrecorded pulses can sometimes contribute significantly to the deadtime. When in doubt, measure the deadtime. See Chapter XVI for reference to after-the-fact deadtime determination.

VIII. DESCRIPTIVE COMMENTS ON THE NURF PROGRAM AND SUBROUTINES

A. List of Subprograms

The complete NURF program (RP-385X) consists of the following 15 subprograms, listed approximately in the order used:

1. NURF	6. ELMER	11. MATINV
2. CONIN	7. NIMFO	12. EXDIF
3. WTCARDS	8. MINFAC	13. NEWPAGE
4. DCARDS	9. ITER 7	14. DATOUT
5. REVERSER	10. LSQPOL	15. GRAFFIT

The functions performed by each of these subprograms are described briefly in the following paragraphs.

B. Program NURF

The NURF main program causes the data to be treated scaler by scaler. Some of the data processing is done here, but most is done by the other subprograms, which are called as needed. The input cards are read by the subroutines CONIN, WTCARDS, and DCARDS. Then REVERSER is called, if the foils were counted without restacking, followed by the requested decay-correcting procedure: either by fixed decay constant or by least-squares fit. For the latter, the times (in the case of a fission-product fit) and counting rates are converted to logarithms, and the appropriate subroutines are used to obtain the least-squares coefficients. The errors in the coefficients are found, and finally the output subroutines (DATOUT and GRAFFIT) are called.

The overall flow of the program is diagrammed in Fig. 1 (p. 10).

C. Subroutine CONIN

CONIN reads the control deck, sets the various processing parameters accordingly, and prints a table of the resulting input constants.

D. Subroutine WTCARDS

The weight-card deck is read by WTCARDS, obtaining, in essence, (a) the number of foils counted by each scaler, and (b) their weights and identifying names, if any. Figure 2 is a flow chart for this subroutine.

E. Subroutine DCARDS

DCARDS reads the counting-data cards. It calculates the counting rates, makes corrections for background and deadtime, and determines the statistical error in each counting rate. It also makes corrections for accidental coincidences and lists the input counting data and background rates. Figure 3 is a flow chart for DCARDS.

F. Subroutine REVERSER

Subroutine REVERSER is used when the counting-data cards have not been collated and the foils have been counted without restacking ("continuous cycle"). The sequence numbers for the even-numbered traverses are reversed, so that each foil then has the same sequence number for all traverses. If some of the foils were not counted during the last even-numbered traverse, the data for that partial traverse are not processed. Such data can be saved by collating the cards (see Section VI-C and Chapter VII).

G. Subroutine ELMER

For the averaging and collating procedures followed by the program, the sequence numbers for the counting data for each foil must be determined. This is done by ELMER, which determines the sequence numbers LMIN and LMAX, and the interval LDEL, for use as the parameters of a DO statement. ELMER is called each time the data for an individual foil are to be extracted from the full data stack for separate handling.

H. Subroutine NIMFO

NIMFO performs the inverse of the ELMER procedure: given a sequence number, NIMFO determines the number of the corresponding foil.

I. Subroutine MINFAC

Using the directed-search subroutine ITER 7, MINFAC determines for each foil the multiplier A_f (see Eq. 3) that makes the data for that foil fit the least-squares curve found in the main program, in the sense that the sum of squares of the residuals is minimized. Every time a least-squares curve is found by the main program (Eq. 2), then, MINFAC is called for each foil, to determine an adjusted set of the foil multipliers A_f (called "multipliers" because the r_{if} of Eqs. 2 and 3 are logarithms).

J. Subroutine ITER 7

ITER 7, which is called by MINFAC, is a subroutine for finding, by a directed-search procedure, the minimum of a nondifferentiable function. It is described in detail in Appendix A.

K. Subroutines LSQPOL and MATINV

The coefficients for the least-squares polynomial fit (Eq. 2) are found by LSQPOL, which is a standard Argonne subroutine (ANL-E206). This routine in turn requires the matrix-inversion subroutine MATINV (ANL-F402). Both codes were written by Burton S. Garbow.

L. Function EXDIF(X)

In the evaluation of the quantity $e^x - 1$ by means of the library function EXPF(X), significant-figure limitations can be encountered if $|x|$ is very small. To avoid this type error in fixed-half-life decay corrections, the function subprogram EXDIF(X) is used in place of EXPF(X)-1.0. Written by Charles E. Cohn of Argonne National Laboratory, this routine uses a series expansion for $e^x - 1$ when $|x| \leq 0.2$.

Fig. 3. Flow Diagram for Subroutine DCARDS

M. Subroutine NEWPAGE

NEWPAGE is a small subroutine for starting a new page. The page is headed by the TITLE from the first card of the control deck, the date of processing, and the page number.

N. Subroutine DATOUT

When a least-squares option has been selected, DATOUT calculates the relative activities by means of Eq. 4, using the least-squares coefficients found in the main program. This subroutine does the weight correcting and the normalizing, and calculates the average activations for each foil and the errors therein. It prints the output data, and, if requested, punches the weighted-average cards for use in the COMBO program.

O. Subroutine GRAFFIT

Using a method adapted from work by Charles E. Cohn, the GRAFFIT subroutine prints a graphical representation of the normalized activities, with the points grouped foil by foil. The points are plotted on a 0 to 100 scale, providing a resolution of 1% of full scale. One decade of scale expansion is provided for points that would fall between 1 and 10% of full scale; such points are indicated by the symbol X instead of the symbol *. For points that would lie at less than 1% of the unexpanded full scale, two decades of scale expansion are used; these points are plotted with the symbol O. An example of the output is given in Fig. 4.

In the determination of the value of the point for plotting, rounding off to the nearest integer on the 0 to 100 scale is done by adding 0.5 to the normalized activation, and then permitting the computer to truncate.

LUTETIUM FOILS, IRRADIATED 1/31/67, 3.7-HR ACTIVITY.

Fig. 4. An Example of the Output Produced by Subroutine GRAFFIT

PART THREE

COMBO: A CODE FOR NORMALIZING AND COMBINING LARGE AMOUNTS OF FOIL COUNTING DATA

IX: INTRODUCTION

When detailed activation traverses in experimental reactor cores are made, large sets of foils--50 to 100 or even more--are often irradiated. All the foils must be counted before the activity has decayed, and it is frequently necessary or desirable to use two or more counting systems simultaneously, or to divide the set of foils into groups for counting at different times. It is then necessary to normalize the results in some manner, and this can be done by having, in each group, one or more foils which also have been counted in another group. An appropriate normalizing factor is then the average ratio of the counting rates for the common foils.

In the ANL Reactor Physics Laboratory, activated foils are usually gamma-counted, in an automatic foil-changing system having a dual crystal arrangement with one detector above the foil and the other below. Two complete systems are in operation, each having two automatic changers feeding a bank of amplifiers and single-channel analyzers. The foils in the group being counted are repeatedly cycled, and the data are punched out on IBM cards. Corrections for decay, deadtime, background, and reactor running time can be made by means of the foil data-processing code NURF. The maximum number of foils that can conveniently be counted in one group is about 40. The counting facilities are discussed in Ref. 1, although various additions and improvements have been made since that report was written.

Recent practice has been to feed the output from each detector to a linear amplifier, and thence in parallel to a set of two or three single-channel analyzers, each with its own scaler. Each detector provides an independent count of each foil, and the redundancy of having more than one analyzer-scaler for each detector provides additional insurance against some of the possible malfunctions that might otherwise cause the loss of large amounts of data. In fact, more than once or twice erratic behavior has developed somewhere in a counting channel, a situation that tends not to be discovered until the data are analyzed. The redundancy in collected data has made it easier both to diagnose the problem and to identify and reject bad data.

Thus, counting data are frequently on hand for several groups of foils, with counts from up to six scalers for each foil. Even when the output from NURF (which provides for each scaler the weight-corrected average of the corrected successive counts for each foil) is at hand, the

complete job of normalizing and averaging the results for all the scalers, and then determining and applying the normalizing factors for combining the data from the different counting groups, is a formidable task without further computer assistance. Program COMBO was developed for these reasons.

Relative, not absolute, activation rates are being considered here-- for instance, the neutron flux in one part of the reactor core is desired relative to the flux somewhere else. These measurements are placed on an absolute basis, where required, with a different type of calibration experiment.

X: OPERATION OF THE PROGRAM

COMBO (RP-367X) is a program written in FORTRAN for the CDC-3600 computer. The principal input consists of the weighted-average specific saturated activation (i.e., activation per unit weight) for each foil and each scaler, as punched out on IBM cards by NURF or by RP-202.

Processing proceeds in two stages. First, for each group of foils the data for all the scalers used are normalized and combined, the result being stored for subsequent use and also printed out. In the second stage, the program searches for the common foils, determines the factors for normalizing one group to the next, combines everything into one large group, and prints the result.

For the first stage of processing, input information is required giving for each group the desired weighting factor to be used in averaging the results for the scalers used. In determining these factors, the experimenter takes into account any feelings he may have regarding the relative reliability of the different channels, plus some additional considerations. For instance, with the counting equipment used in the Reactor Physics laboratory, although the counting rate in each gamma detector is sensitive to small vertical displacements of the foil, the average of the results from the two detectors, weighed inversely according to their relative efficiencies, is insensitive to such displacements. Thus, it has been usual, in using COMBO, to weight the scalers so that the bottom and top detectors get equal emphasis. Secondly, sometimes the same set of foils has been counted at different times or on different systems, and it is desired to combine all the data collected. This can be accomplished by assigning a scaler number and suitable weighting factor to each set of data.

The operation of Program COMBO will now be discussed in greater detail.

Let N and I be the number of foils and the number of scalers, respectively, and let A_{ni} be the activity of the n th foil as determined by the

ith scaler. Using the first scaler as an arbitrary reference, the program calculates for each scaler after the first (i.e., for $i \geq 2$) an average ratio R_i , where

$$R_i = \frac{1}{N} \sum_{n=1}^N \frac{A_{n1}}{A_{ni}}. \quad (24)$$

The standard deviation σ_i in the individual ratios is then determined, and a new value for R_i is found, this time rejecting any ratios that differ from the average by more than $j\sigma_i$, where j is 3 unless otherwise specified in the input. The process is repeated until σ_i is unchanged, and a printed message identifies the foils for which the ratios were finally rejected. This feature has been found useful in pinpointing bad counts that might otherwise have been missed.

The average activation A_n for each foil is then calculated by

$$A_n = \frac{\sum_{i=1}^I w_i R_i A_{ni}}{\sum_{i=1}^I w_i}, \quad (25)$$

where R_1 is identically unity and the w_i are the scaler weights assigned by the experimenter. The results are printed out for each group, and processing proceeds to the second stage, where the groups are combined.

The second-stage processing is done by a subroutine called NORMGRUP. It starts with the first two groups of foils. The foils common to both groups are located, and the normalizing ratio is determined according to

$$R_k = \frac{1}{N} \sum_{n=1}^N \frac{a_{n1}}{a_{nk}}, \quad (26)$$

where N is now the number of common foils, and the subscripts 1 and k refer, respectively, to the first and second groups. A message is printed out which identifies the common foils, their ratios, and the standard deviation in the ratios. The activities of the foils in group k are multiplied by R_k and combined with the foils from the first group into a new group 1. The process is then repeated with $k = 3$, then 4, and so on. (The code is dimensioned to handle up to 20 groups, with up to 100 foils in any group, of which 50 or less may be common to other groups.)

When two groups of foils are merged, the activity assigned to the common foils is computed according to

$$A_n = \frac{u_n A_1 + R_k A_k}{u_n + 1}, \quad (27)$$

where u_n is an integer equal to the number of times the n th foil has appeared as a common foil. This ensures that each count of a common foil has the same weight in determining its final relative activation.

The final printout includes a column containing the foil names (as assigned originally in the input to NURF or RP-202) and two columns labeled "Normalized Activity." Of this latter pair of columns, the first contains the activities as multiples of the activity of the weakest foil, and the second contains the activities as fractions of the activity of the strongest foil.

XI. INTERCALIBRATED SETS OF FOILS

In work with foils made from certain alloys, notably Dy-Al and U-Al, the concentration of the material of interest is so variable that ordinary weight corrections cannot be made to the activations. In these cases, sets of foils must be intercalibrated by some process such as simultaneous irradiation on a rotating wheel. The COMBO program can be used to facilitate the assembly of such sets of intercalibrated foils. In particular, a feature has been provided that makes it possible to request in the input that the program punch out a set of intercalibrated WEIGHT cards in a format compatible with the input requirements of RP-202 and NURF. (See Section VII-G.)

XII. CARD PREPARATION FOR COMBO

To run COMBO on the CDC-3600, the input card deck is made up in the following sequence:

1. Sequence card
2. Job card
3. Accounting card
4. Binary deck for Program COMBO
5. Binary deck for Subroutine NORMGRUP

6. Binary deck for Subroutine NEWPAGE
7. Run card
8. Data deck, in the sequence as given in Table VI, starting with a problem title card. Several problems can be included in the same data deck.
9. $\begin{smallmatrix} 7 \\ 8 \end{smallmatrix}$ END OF FILE.

PART FOUR
THE TWOSORCE PROGRAM FOR DETERMINING
COUNTING-SYSTEM DEADTIME

XIII. INTRODUCTION

If the relative activation of a set of irradiated foils covers a wide range, the accurate determination of this relative activation frequently depends upon, among other things, an accurate knowledge of the resolving-time losses in the counting equipment. To take a quantitative example, suppose that background and counting-time limitations make it desirable that the weakest foil gives a counting rate of at least 100 per second, and suppose that the strongest foil is a factor of 100 more active. A 2- μ sec uncertainty in the deadtime (regardless of the magnitude of the deadtime) will then result in a 2% uncertainty in the activation ratio under the optimum counting-rate conditions--conditions that cannot be maintained unless the sample has a relatively long half-life. It is not unusual for the resolving-loss errors to amount to 3% or more, unless special precautions are taken.

The actual mechanism by which resolving-time losses occur varies from one counting system to another.⁹ In the particular case that each pulse counted causes a fixed deadtime τ , with no extension of that interval if an unresolved pulse occurs, the true counting rate N (excluding background) is given exactly by

$$N = \frac{n}{1 - n\tau} - b, \quad (28)$$

where n is the observed counting rate and b is the background rate. In another possible case, each resolved or unresolved pulse has associated with it a deadtime τ which starts when the pulse occurs. In this case the true counting rate is given by

$$N = ne^{(N+b)\tau} - b, \quad (29)$$

which cannot be solved exactly for N . The above two types of counter are sometimes referred to as "nonparalyzable" and "paralyzable," respectively.

There are also other possibilities. In general, N can be expressed by the series

$$N + b = n[1 + n\tau + B(n\tau)^2 + C(n\tau)^3 + D(n\tau)^4 + \dots], \quad (30)$$

where the values of the coefficients B , C , D , ... depend on the type of mechanism operating. For the series corresponding to Eq. 28 all coefficients are unity, whereas for Eq. 29 the values for B , C , and D are $3/2$, $8/3$, and $125/24$,⁴ respectively.

For a given counting system it is possible, at least in principle, to determine reasonable values for the first few coefficients, either from a knowledge of how the system works, or by actual measurement. Usually not more than the first few coefficients are important, since there is an upper limit to the tolerable counting rate (typically $\sim 10^4$ counts/sec), determined by system stability. The counting data obtained in the automatic counting facility are usually processed by means of the computer code NURF, a code that permits the user to specify the values of τ , B , C , and D to be used in making the deadtime correction. Higher coefficients are neglected.

Until recently our practice has been to determine a value of τ with a double-pulse generator. However, in working with data from foils that were counted over several half-lives, we observed that the input deadtimes apparently were often in error. Program TWOSORCE processes the data for double-source deadtime measurements. It gives the deadtime according to an exact solution of Eq. 28, and it also gives a value of τ for input values of B , C , and D selected by the experimenter, the latter result being determined by an iterative solution of Eq. 30. The operation of the code will be described in more detail in Chapter XV below.

The effective deadtime for the equipment used, even for a given counting channel, has been a variable quantity. In particular, it depends to such an extent on the discriminator level that for very accurate work it must be measured each time a set of foils is counted.

XIV. PRINCIPLE OF THE DOUBLE-SOURCE METHOD

The double-source technique, widely used for the determination of counting-system deadtime, is based on the fact that the observed counting rate, when two sources are counted together, is less than the sum of their individual counting rates, the difference being due to increased deadtime losses.

Consider first the case where Eq. 28 applies; i.e., the deadtime is nonextendable. Let the subscripts 1, 2, and 12 refer, respectively, to source 1, source 2, and both together. Then from the three measurements of a paired-source experiment we get three equations like Eq. 28. Remembering

that $N_{12} = N_1 + N_2$, we can solve the equations to give τ in terms of the observed counting rates:*

$$\tau = \frac{1}{n_{12}} - \sqrt{\left(\frac{1}{n_1} - \frac{1}{n_{12}}\right)\left(\frac{1}{n_2} - \frac{1}{n_{12}}\right) + \frac{b}{n_1 n_2 n_{12}}}. \quad (31)$$

We have assumed that $bn_{12}\tau$ and similar terms are negligible with respect to b . This is reasonable since the background correction itself is normally small.

Turning now to the series-expansion formulation, which includes Eqs. 28 and 29 as special cases, we get from the three measurements of a paired-source determination three equations like Eq. 30 (and we also have $N_1 + N_2 = N_{12}$). From these equations, we can eliminate N_1 , N_2 , and N_{12} , to get

$$\begin{aligned} (n_{12} - n_1 - n_2 + b) + (n_{12}^2 - n_1^2 - n_2^2) \cdot \tau + (n_{12}^3 - n_1^3 - n_2^3) \cdot B\tau^2 \\ + (n_{12}^4 - n_1^4 - n_2^4) \cdot C\tau^3 + (n_{12}^5 - n_1^5 - n_2^5) \cdot D\tau^4 = 0, \end{aligned} \quad (32)$$

where only the first four coefficients have been kept. Given values for B , C , and D , Eq. 32 can be solved for τ by iteration.

By terminating Eq. 30 after the first few terms, we make the implicit assumption that the higher terms are negligibly small--a condition that is easily violated for large counting rates or large deadtimes. Thus, for accurate use of the series representation, the product $n\tau$ must be kept small enough so that the higher terms are unimportant.

XV. OPERATION OF PROGRAM TWOSORCE

The input for Program TWOSORCE consists of a title card; a card with the decay constant of the material counted; a card with additional control constants, including values for the coefficients B , C , and D ; and the deck of counting-data cards, as produced by the automatic foil-counting equipment. The first of the counting-data cards is a background card,

*The following formula is sometimes used in place of Eq. 31:

$$\tau = \frac{\frac{n_1 + n_2 - n_{12}}{2} \cdot \frac{n_1 + n_2 - n_{12}}{2}}{n_{12} - n_1 - n_2} \approx \frac{n_1 + n_2 - n_{12}}{2n_1 n_2}, \quad (31a)$$

where the second expression comes from letting $n_{12} = n_1 + n_2$ in the denominator. Equation 31a can be obtained from Eq. 30 by dropping the quadratic and higher terms. As $n\tau$ increases to moderately high values, the first denominator in Eq. 31a can actually pass through zero and become negative.

followed by three cards containing, respectively, the count of foil 1, foil 2, and both together. This last set of three cards can be repeated up to 49 times. The program will average the results.

Using the times and durations of the counts (punched automatically on the counting-data cards) and the input decay constant, the program calculates the individual counting rates that would have been observed when the combined count was taken, and then calculates τ using Eq. 31. With this value of τ as a starting point, an iterative solution of Eq. 32 is found, using the input values of B, C, and D. Both values of τ are printed out. When all the counting data have been processed, the average τ is computed for each of the two methods, along with the standard deviation in each average and the rms deviation in the individual values of τ .

The above calculations are made for each scaler used, and then the program passes on to the next problem, if there is one.

XVI. DISCUSSION OF DEADTIMES

With Program TWOSORCE, two of the more active foils can be used to make an expeditious before-the-fact determination of the deadtime prior to the start of counting, at the actual gain and voltage settings to be used. However, at present there is no confidence that the deadtimes are known, in any given case, to much better than $\pm 1 \mu\text{sec}$. The reason for this uncertainty lies not so much in the reproducibility of a given experiment, as in unpredictable and apparently inconsistent changes in measured deadtime when parameters such as counting rate or bias level are changed.

Part of the solution could lie in determining the optimum values of the higher coefficients on a routine basis, but such a determination is time-consuming: the optimum values have varied widely from one counting channel to another. In general, the coefficients probably cannot be expected to match one of the clear-cut cases like Eqs. 28 or 29, especially for those systems where pile-up rejection circuitry rejects pulses that normally would be counted, but which are superimposed on the tail of other pulses which were too small to be counted at the bias setting used. In this latter case, even the introduction of an artificially fixed deadtime following each counted pulse cannot be expected to lead to an effective deadtime that is independent of the bias setting.

As pointed out above, the TWOSORCE program permits a before-the-fact measurement of the deadtime of the counting system, which should be made whenever the deadtime correction is likely to be appreciable. If, however, it is discovered only later that there happened to be a deadtime problem, a computer code such as DEADTIME¹¹ (not described in this report) can sometimes be used for an after-the-fact determination of the

deadtime, by means of the decaying-source technique. In NURF, ITCH(6) = 2 causes cards for DEADTIME to be produced.

XVII. CARD PREPARATION FOR PROGRAM TWOSORCE

To run TWOSORCE on the CDC-3600, the input card deck is made up in the following sequence:

1. Sequence card
2. Job card
3. Accounting card
4. Binary deck for Program TWOSORCE
5. Binary deck for Subroutine NEWPAGE
6. Run card
7. Data deck, in the sequence as given in Table VII, starting with a title card, and ending with a blank card and a $\frac{7}{8}$ END OF FILE card.

APPENDIX A

ITER 7, A Directed-search Subroutine for Finding Minima

ABSTRACT

ITER 7 is a FORTRAN subroutine for finding, by a directed-search procedure, the value of x corresponding to the minimum of the function $y = f(x)$, where y is computed numerically. The function $f(x)$ need not be differentiable or even continuous, but there are certain limitations related to continuity and to the proximity of maxima. A convergence test is provided that is independent of the slope of the function. The subroutine itself requires less than a millisecond for an iteration (on the CDC-3600), to which must be added the time used in computing the value of f . There are no dimensioned variables.

Application to the multidimensional function $f(x, y, z, \dots)$ is easily accomplished, with an initially coarse search mesh which is refined as the search proceeds. In tests with simple one-, two-, and three-dimensional problems, the numbers of times that f had to be calculated were typically in the ratio 1:6:20, for comparable convergence criteria. In routine one-dimensional use as part of the NURF foil data code, about 12 cycles provide satisfactory convergence.

1. One-dimensional Applications

The FORTRAN subroutine ITER 7 has been written for use in finding the value of x that corresponds to a minimum of the function $y = f(x)$, where y is computed numerically. Thus, $f(x)$ may be nondifferentiable, but there must be a nonzero separation between extremum points. In the NURF application (see Section III-A), y is the sum of the squares of a set of residuals, which can be minimized by appropriate selection of x .

To discover the minimum, ITER 7 is called repetitively. An initial increment dx must be specified; subsequent increments are determined by ITER 7. The calling program uses x_0 , an initial value of x , to calculate an initial value of y , and then calls ITER 7, which returns an incremented value of x to the calling program for recalculation of y , and so on. Let c be the number selected for determining when convergence has occurred (see below), and let n_m be the upper limit for the number of cycles through the loop. Then ITER 7 is used in the following manner:

```

X = x0
DX = dx
NMAX = nm
CONCRI = c
NCYCL = 0

100 [calculation of y = f(X)]
140 CALL ITER 7 (NCYCL, Y, X, DX, CONCRI, NMAX, ISTOP, DDX)
150 IF (NCYCL.EQ. NMAX .OR. ISTOP.EQ. 1) 200, 100
200 [Continue]

```

The parameter ISTOP is the convergence indicator: it is initially set to zero by ITER 7, and becomes 1 when the convergence criterion has been met. The only other parameters whose values are changed within the subroutine are NCYCL, X, and DDX. The value of DX, the initial x-increment, is not changed within ITER 7, new values of the increment being stored in the location DDX. DDX is included in the parameter list because, in multidimensional applications (see below), having it available in the calling program is useful; for one-dimensional problems it is not normally needed.

Because y is calculated in the calling program, rather than (as in some similar codes) by a separate subroutine called by the iterating subroutine, certain loading complications are avoided in computers such as the DDP-24, where the order of loading subroutines is important.

The FORTRAN listing (CDC-3600) for ITER 7 is given in Fig. A-1. In operation, the subroutine changes X by increments DX until the minimum has been passed. (The initial direction of march is reversed if the second value of Y is greater than the first.) When a value of Y greater than (or equal to) the preceding one indicates that a minimum has been passed, the sign of DDX is reversed, and its magnitude is made half as great. Since the second such step in the reverse direction would repeat a calculation already made, that point is jumped over, the corresponding value of Y having been retained. This procedure eliminates most recalculations, but not all, since still earlier points are occasionally repeated.

The criterion used by ITER 7 in determining convergence takes advantage of the fact that each time the increment is divided by two, the new value of x cannot be away from the minimum by more than three times the new increment. Thus the user inputs for CONCRI a suitable fraction of his

selected initial dx. (The convergence test occurs in statement 20 of the FORTRAN listing, Fig. A-1.) This method for determining convergence avoids problems that can arise with a slope-sensitive criterion in the following situations:

1. A slope that changes very gradually near the minimum. In that case, the significant-figure limitation of the computer can be exceeded because successive values of y become approximately equal.
2. A slope that is discontinuous at the minimum.
3. The case where there is some jitter in successive calculations of y; this can happen, for instance, if that calculation also involves an iteration procedure, as in a multidimensional application.

```

SUBROUTINE ITER 7 (NCYCL, A, B,DB, CONCRI, NMAX, ISTOP, DELB)

C   PRODUCES INCREMENTED VALUES OF B IN SEARCH FOR A MINIMUM VALUE OF A.
C   INITIAL INCREMENT IS DB, AND CONCRI IS THE CONVERGENCE CRITERION (SEE
C   STATEMENT 20). CALLING PROGRAM MUST INITIALLY SET NCYCL TO ZERO, AND IT
C   MUST TEST FOR (NCYCL .GE. NMAX) AND FOR (ISTOP .EQ. 1), AND LEAVE LGCP WHEN
C   EITHER OCCURS. ISTOP IS INITIALLY SET TO ZERO BY ITER 7, AND BECOMES 1
C   WHEN THE CONVERGENCE CRITERION HAS BEEN MET.

9010 FORMAT (/* MESSAGE FROM ITER 7... CALLING PROGRAM HAS FAILED TO*
2       * NOTE THAT CONVERGENCE CRITERION HAS BEEN MET. (*
3       * E10.3 *.) * / * EXECUTION TERMINATED. *)
9020 FORMAT (/* MESSAGE FROM ITER 7... CALLING PROGRAM HAS FAILED TO*
2       * NOTE THAT MAX. NO. OF CYCLES (* I6 *) HAS BEEN*
3       * EXCEEDED.* / * EXECUTION TERMINATED. *)

C
  IF (NCYCL .LT. 0) NCYCL = 0
  IF (NMAX .GT. 300) NMAX = 300
  IF (NCYCL .GT. (NMAX+2)) GO TO 210
  NCYCL = NCYCL + 1
  IF (NCYCL .GT. 1) GO TO 10
  DELB = DB      $ ISTOP = K = 0      $ Q2=A      $ GO TO 100
10 IF (ISTOP.EQ. 1) GO TO 200
  IF (K .EQ. 1) GO TO 15
  Q2 = A      $ GO TO 20
15 Q1 = A      $ Q2 = QK
20 IF ( ABSF(DELB/DB).GT. CONCRI ) GO TO 30
25 ISTOP = 1      $ RETURN
30 IF (Q2 .EQ. 0.0) GO TO 35
  IF (ABSF ((Q1 - Q2)/Q2) .LT. 1. E-10) GO TO 50
35 IF (Q2 .GE. Q1) GO TO 40
C NOTE THAT IF K=1, DELB GETS ADDED TWICE (ONCE BY STATEMENT 100).
  IF (K.EQ.0) GO TO 100 $ B = B+DELB      $ Q1=QK      $ GO TO 100
40 IF (NCYCL .GT. 2) GO TO 50
  DELB = - DELB
  B = B + 2. * DELB      $ RETURN
50 IF (K .EQ. 0) GO TO 60 $ QK=Q1      $ DELB = DELB/2      $ GO TO 100
60 DELB = -DELB/2      $ K = 1      $ QK = Q1
  B = B + DELB      $ RETURN
100 B = B + DELB      $ IF (K.EQ.0) Q1 = Q2      $ K = 0
  RETURN

200 PRINT 9010, CONCRI      $:STOP
210 PRINT 9020, NMAX      $ STOP      $ END

```

Fig. A-1. FORTRAN Listing for ITER 7

If, however, a different type of convergence test is desired, it can be inserted in the calling program between statements 140 and 150 in the above example. CONCRI must be made sufficiently small, and the tests performed by statement 150 should never be bypassed, except on leaving the loop.

To test the one-dimensional performance, we used the parabola $y = x^2$ along with the input quantities, $x_0 = 5$, $dx = 1.5462$, and $c = 1.27 \times 10^{-4}$. Thirty-two cycles were required for convergence, the final values being $x = 1.024 \times 10^{-4}$, $y = 1.048 \times 10^{-8}$.

The CDC-3600 computer time required was checked with the same problem, with $c = 10^{-15}$. The number of cycles was 115, and the total time spent in the loop was 0.107 sec, for an average of 0.93 msec per cycle. (The final value of x was 1.4×10^{-15} .)

The approximate number of cycles, N , required for convergence can be estimated from

$$N \approx 2.3n + b,$$

where n is given by $2^{-n} = c$, and b is the number of initial increments dx that must be traversed before the minimum is reached for the first time.

The memory space required by ITER 7 is minimal, since the subroutine does not use any dimensioned variables.

ITER 7 has been in routine use as part of the NURF foil data-processing code. In that application, satisfactory convergence is obtained in about 12 cycles.

2. Multidimensional Applications

Although the above discussion explicitly considers only one-dimensional problems, ITER 7 can also be used to find minima in two or more dimensions. For a two-dimensional problem, for instance, one can use two such subroutines, called ITER 7 and (say) ITER 8. To find the minimum of $z = f(x, y)$, one would use ITER 7 to find the minimum of $z = f(x, y_1)$, where y_1 is kept fixed, followed by a single call of ITER 8 to produce an incremented y for recalculation of $f(x, y_1)$. ITER 7 is again used to find the minimum in the x direction, followed by another call of ITER 8, and so on. One possible method is as follows, where the subscripts 7 and 8 refer to the x and y iterations, respectively. Two separate ITER subroutines are required here, because ITER 8 stores some quantities during operation of ITER 7.


```

      Y = y0
      DY = dy
      DDY = dy
      NMAX8 = nm8
      CONCRI8 = c8
      NCYCL8 = 0
40  F = f
      X = x0
      NMAX7 = nm7
      CONCRI7 = c7
50  NCYCL7 = 0
51  DX = f * DDY

100 [Calculation of Z = f(x,y)]
      CALL ITER 7 (NCYCL7, Z, X, DX, CONCRI7, NMAX7, ISTOP7, DDX)
      IF (NCYCL7 .EQ. NMAX7 .OR. ISTOP7 .EQ. 1) 200, 100
200  CALL ITER 8 (NCYCL8, Z, Y, DY, CONCRI8, NMAX8, ISTOP8, DDY)
      IF (NCYCL8 .EQ. NMAX8 .OR. ISTOP8 .EQ. 1) 300, 50
300 [Continue]

```

Statements 40 and 51, in the above example, permit starting with a coarse mesh which becomes finer as the search proceeds. The x-convergence number c_7 should be made large--like 0.7. (It must be < 1 . Any c_7 between $1/2$ and 1 will cause the inner loop to be left after a single reversal of DDX.) That way ITER 7 will only be called a few times on each pass, but X will converge along with Y, since the initial increment DX for each new round is tied to DDY by statement 51. The factor F would be set equal to unity if the convergence requirements for X and Y were roughly the same. Ultimate convergence is determined by c_8 .

As a test of this procedure, it was used to find the minimum (which occurs at the point $[0,0,0]$) of the paraboloid $z = x^2 + xy + y^2$, using the following input quantities:

```

x0 = y0 = 5
c7 = 0.7
c8 = 10-4
dy = 1.5462
f = 1.

```

Convergence was reached after ITER 8 was called 38 times. ITER 7 was called an average of 4.7 times per pass, it having been necessary to

calculate z 180 times. (Note that z does not need to be recalculated just before the y iteration, since x is not changed by ITER 7 on the final call.) The total computer time spent in the iteration loop, which included 38 lines of printout, was 496 msec. The final values were

$$\begin{aligned}x &= 2.9 \times 10^{-7}, \\y &= 1.4 \times 10^{-4}, \\z &= 2.0 \times 10^{-8}.\end{aligned}$$

Figure A-2 contains a test program (PROGRAM ITR3D) which applies ITER 7 to the three-dimensional problem of finding the minimum of the four-space surface $t = x^2 + y^2 + z^2 + x^2y^2$. A 30 to 50% time saving occurred when CONCRI7 and CONCRI8 were between 0.5 and 1, as compared with the range 0.25 to 0.5. With CONCRI7 and CONCRI8 in the former range, and starting out at $x = y = z = 5$, six values of $DZ (= DX = DY)$ were tried, ranging from 1.5 to 2.1. (CONCRI9 was kept at 10^{-4} .) The number of times t was calculated was variable, ranging from 483 to 1021, with an average of 710. The final magnitudes of X , Y , and Z were all below 1.6×10^{-4} , the final values of T all being less than 8.4×10^{-8} .

```

PROGRAM ITR3D
C
C          TO CHECK 3-DIMENSIONAL PERFORMANCE OF ITER 7
8010 FORMAT (4E10,3)
9110 FORMAT (6H DZ = F8.4, 6X9HCONCRI7 = E12.5, 6X9HCONCRI8 = E12.5,
X      6X 9HCONCRI9 = E12.5 // )
9120 FORMAT (* NCYCL7 NCYCL8 NCYCL9* 8X1HX, 13X1HY, 13X1HZ, 13X1HT,
X      13X2HDX, 12X2HDY / )
9140 FORMAT (//* ELAPSED TIME FOR ITER7,8,9 WAS **3PF8.4* SECONDS.*)
9150 FORMAT ( 3(I5,3X), 6E14.4 )
9160 FORMAT (//* TOTAL NUMBER OF ITER 7 CALLS WAS * 18.1H.)
9170 FORMAT ( 1H1)
C
20 READ 8010, CONCRI7, CONCRI8, CONCRI9, DZ $ IF (EOF,60) 400,30
30 X = 5 $ Y = 5 $ Z = 5
NMAX7 = 20 $ NMAX8 = 20 $ NMAX9 = 50
DDZ = DZ $ NCYCL9 = 0
PRINT 9110, DZ, CONCRI7, CONCRI8, CONCRI9
PRINT 9120 $ NSUM7 = 0
C
T0 = TIMELEFT(T0)
50 NCYCL8 = 0 $ DY = DDZ $ DDY = DY
100 NCYCL7 = 0 $ DX = DDY
150 T = X**2 + Y**2 + Z**2 + (X*Y)**2 $ NSUM7=NSUM7+1
CALL ITER7 (NCYCL7, T, X, DX, CONCRI7, NMAX7, ISTOP7, DDY)
IF (NCYCL7.EQ.NMAX7 .OR. ISTOP7.EQ.1) 250,150
250 CALL ITER8 (NCYCL8, T, Y, DY, CONCRI8, NMAX8, ISTOP8, DDY)
IF (NCYCL8.EQ.NMAX8 .OR. ISTOP8.EQ.1) 300,100
300 CALL ITER9 (NCYCL9, T, Z, DZ, CONCRI9, NMAX9, ISTOP9, DDZ)
PRINT 9150, NCYCL7, NCYCL8, NCYCL9, X, Y, Z, T, DX, DY
IF (NCYCL9.EQ.NMAX9 .OR. ISTOP9.EQ.1) 350,50
350 T1 = TIMELEFT (T1) $ TELAPSED = T0 - T1
PRINT 9140, TELAPSED $ PRINT 9160, NSUM7
C
PRINT 9170 $ GO TO 20
400 STOP $ END

```

Fig. A-2. Example of a Three-dimensional Application of ITER 7

3. Discussion

ITER 7 appears to provide a reasonably efficient means for finding the minimum (or maximum, by inverting the function; or zero, by using the absolute value of the function) of a nondifferentiable function of one or more variables. If the function has more than one extremum point, the subroutine can still be used, provided that there is no maximum between the initial approximation and the desired minimum, and that the initial independent-variable increments are not large enough to cause one of the maxima to be passed over in the first one or two steps. A code employing ITER 7 to find successive extrema (or zeroes) could easily be written.

Some additional limitations in the use of ITER 7 should be noted, for completeness:

1. The search procedure can be disrupted if there is, between the starting point and the minimum being sought, a region where y is constant.
2. The search process might also be disrupted if a δ -function singularity coincides with a search point; other such singularities will not be detected.
3. If the minimum should occur at a cusp or pole in $f(x)$, or at or very near a discontinuity, the search procedure will still find, with the specified precision, the value of x where the minimum occurs; however, the value of y at that value of x might differ appreciably from the actual minimum value.

As is typical of such procedures, the time required for satisfactory convergence depends strongly on the quality of the initial guess. In the examples given above, starting out closer to the origin, with correspondingly smaller initial increments, could have produced equally good convergence with many fewer iterations. As already mentioned, in routine use as part of the NURF code, about 12 iterations are needed.

APPENDIX B

FORTRAN Listing for the NURF Code

This appendix contains FORTRAN listings for the various NURF subprograms, except for three: ITER 7 is listed in Appendix A, and the standard ANL subroutines LSQPOL and MATINV are not listed in this report.

PROGRAM NURF

RP 385 X

C WITH LEAST-SQUARES FITTING OPTION, DETERMINES DECAY CONSTANTS OR FISSION-
C PRODUCT DECAY, AND CALCULATES RELATIVE SPECIFIC ACTIVITIES.
C WITH FIXED-DECAY-CONSTANT OPTION, GIVES SATURATED SPECIFIC ACTIVITIES.

```

9009 FORMAT (      * OUTPUT DATA FOR SCALER* I2,1H.  29X I4 * CCUNTS*
2      * PROCESSED.* I6 * COMPLETE TRAVERSES.* I6
3      * FOILS. * / 75X I5* FOILS IN INCCMPLETE TRAVERSE.* /)
9010 FORMAT(      * OUTPUT DATA FOR SCALER* I2,1H. I6 * OVER-ALL CYCLE*
2      *S FOR LSQ FIT.      SUM OF RESID.SQUARED =* E11.4
3      *..      DELS = * E9.2 /  54X I4 * COUNTS PROCESSED.*
4      * I6 * COMPLETE TRAVERSES. * I6 * FOILS. * )
9020 FORMAT(      * COEFF.NO. COEFFICIENT ERROR,PERCENT JERCO *
X      30X I5 * FOILS IN INCCMPLETE TRAVERSE. * )
9029 FORMAT( 4X I3, 6X E11.4,2PF13.4, OP, 5X I2 /
2      4X I3,6X E11.4,2PF13.4,OP,5X I2,10X* CONVERGENCE CRITERION*
3      * NOT MET.* / ( 4X I3, 6X E11.4, 2PF13.4, OP, 5X I2))
9030 FORMAT( 4X I3, 6X E11.4,2PF13.4, OP, 5X I2)
9031 FORMAT ( 2X)
9040 FORMAT ( / 1X * END OF CALCULATIONS FOR * 9A8)
9610 FORMAT ( // * FIRST TEN VALUES OF TT(K)... * / 10E12.4 )
9618 FORMAT(/* K,T(15,K),FA(K)...FOR LINEAR APPROX.* / (5(I5,2E11.3)))
9619 FORMAT(/* SUM,NFOIL,NCYCL(NFOIL),F(NFOIL)...FOR EACH LSQ CYCLE*)
9620 FORMAT ( I3, E11.3, 6(I5,I4,E11.3) /(14X, 6(I5,I4,E11.3)))

```

```

COMMON/A/BKW,      CMULT,      COEFF(10),  CONCR(6),  CRATE(400),  COM1
2      DT(6),      DTB,DTIC,DTD,EFC(6),  ENTAU(6,400),F (201),  COM2
3      FWT(6,201), IMAX, IS,  ISCALR(6),  ISMAX,  ITCH(10),  COM3
4      KGUNTS(201),KROOM,      LAMBDA,      LDEL,      LMAX, LMIN,  COM4
5      NAME(6,201), NCARD(6,400),NCARDS(6),  NCCUNTS,  NCYCL(201),  COM5
6      NEXTRA,      NFOIL,      NFOILS(6),  NREM,      RATE(6,400),  COM6
7      RELSIG(6,400),T (6,400),  TCORR,      WGT(400),  NTRAVS  COM7
COMMON /DC/ JSTOP, LL(6),BKF
COMMON/DATWT/ JWT(6,201), CDR, HR, FWZ
COMMON /PAGEBLK/ NPAGE, ITITLE
COMMON/GRAF/ ACNORM(400)
COMMON/DECA/ TIRRAD, TSCRAM, CTIM(6,400),  ACT(400)
COMMON/ACC/ INCA(6), INCB(6), MAC(6)

TYPE REAL LAMEDA
DIMENSION ERCO(10), ERMAT(10,10), FA(400), ITITLE(9), JERCO(10), 385X
X      REZIC(400),      TT(400), WACT(201) 385X

```

```

5 NPAGE = JSTOP = 0
C READ INPUT CONSTANTS
10 CALL CONIN      $ ITCH2 = ITCH(2)
C IF THERE WAS FATAL ERROR IN CONIN, PROCEED TO NEXT PROBLEM.
IF (JSTOP .EQ. 1) GO TO 5

C READ WEIGHT-CARD DECK
CALL NEWPAGE
CALL WTCARDS
C IF THERE WAS FATAL ERROR IN WTCARDS, PROCEED TO NEXT PROBLEM.
IF (JSTOP .EQ. 1) GO TO 5

```

```

C READ COUNTING-DATA CARDS
  CALL NEWPAGE
  CALL CCARDS
C IF THERE WAS FATAL ERROR IN DATA CARDS, PROCEED TO NEXT PROBLEM.
  IF (JSTOP .EQ. 1 .OR. JSTOP .EQ. 3) GO TO 10
C START MAJOR DO-LOOP, TREATING DATA ONE SCALER AT A TIME.
  DO 400 IS = 1, ISMAX $ IF (NCARDS(IS) .EQ. 0) GO TO 400
                        IF (NCARDS(IS) .GT. 400) NCARDS(IS) = 400
C FIND NO. OF COUNTS PER FOIL
  NCCOUNTS = NCARDS(IS) / NFOILS(IS)
  NREM = NCARDS(IS) - (NCCOUNTS * NFOILS(IS))
  NF = NFOILS(IS) $ NR = NREM + 1
C IF FOILS WERE COUNTED REVERSIBLY, A PARTIAL EVEN-NUMBERED TRAVERSE WILL BE
C DROPPED BY SUBROUTINE REVERSER. THUS KOUNTS(NFOIL) MUST NOT INCLUDE SUCH A
C PARTIAL TRAVERSE.
  IF (ITCH(1) .NE. 2) GO TO 50
  IF NCCOUNTS IS EVEN, GO TO 50
  IF ((NCCOUNTS - (NCCOUNTS/2)*2) .EQ. 0) GO TO 50
  DO 30 NFOIL = 1, NF
30 KOUNTS(NFOIL) = NCCOUNTS $ GO TO 60
50 DO 52 NFOIL = 1, NREM
52 KOUNTS(NFOIL) = NCCOUNTS + 1
  DO 54 NFOIL = NR, NF
54 KOUNTS(NFOIL) = NCCOUNTS
60 CALL NEWPAGE
C IF FOILS WERE COUNTED REVERSIBLY, CALL REVERSER.
  IF (ITCH(1) .NE. 2) GO TO 65
  CALL REVERSER
65 NCDS = NCARDS(IS) $ NFLS = NFOILS(IS)
  IF (ITCH(9) .EQ. 0) GO TO 85
C DECAY CORRECTIONS FOR FIXED HALF-LIFE.
  DO 80 K = 1, NCDS
    TA = -LAMBDA * (TSCRAM + T(IS, K))
C TSCRAM IS THE TIME IN SECONDS FROM SCRAM TO CLOCK RESET, T(IS, K)
C IS THE TIME ELAPSED ON DATA CARD
    TB = -LAMBDA * (TSCRAM + T(IS, K) + TIRRAD)
C TIRRAD IS THE IRRADIATION TIME IN SECONDS AND SO TB INVOLVES THE TIME
C FROM START OF IRRADIATION TO THE END OF THE COUNTING INTERVAL
C CTIM(IS, K) IS THE COUNTING INTERVAL OFF THE DATA CARD
    TG = LAMBDA * CTIM(IS, K)
C TJ IS THE TOTAL DENOMINATOR
C EXCIF(X) IS VALUE OF (EXP(X)-1). SEE C.E.COHN, DDP-1J MEMO NO. 27.
    TJ = -EXP(TA) * EXCIF(TB-TA) * EXCIF(TG) * EFC(ISCALR(IS))
80 ACT(K) = (RATE(IS, K) / TJ) * CTIM(IS, K) * LAMBDA
  PRINT 9009, ISCALR(IS), NCARDS(IS), NCCOUNTS, NFOILS(IS), NREM
  GO TO 320
C CONVERT RATES AND TIMES TO LOGARITHMS, FOR FISSION-PRODUCT FIT.
C CONVERT RATES ONLY, FOR DECAY-CONSTANT FIT.
85 BZ = -1.2 $ IF (LAMBDA .NE. 0.0) BZ = -LAMBDA
  DO 100 K = 1, NCDS
    T(IS, K) = T(IS, K) + TCORR
    IF (RATE(IS, K) .LE. (10.**(-20))) RATE(IS, K) = 10.**(-20)
    RATE(IS, K) = LOGF(RATE(IS, K))
    IF (LAMBDA .NE. 0.0) GO TO 90
    IF (T(IS, K) .LE. (10.**(-20))) T(IS, K) = 10.**(-20)
    T(IS, K) = LOGF(T(IS, K))
C CONVERT TIME ZERO TO TIME WHEN FIRST COUNT OF FIRST FOIL FINISHED.

```

```

90 IF (K .EQ. 1) T1 = T(IS,1)                $ T(IS,K)=T(IS,K)-T1
100 FA(K) = RATE(IS,K) - BZ * T(IS,K)
    IF (ITCH(7) .NE. 3) GO TO 110

C      DIAGNOSTIC PRINTOUT
      PRINT 9618,(K,T(IS,K), FA(K), K=1,NCDS)          $ PRINT 9619

C  FIND INITIAL AVERAGE FOIL FACTORS FOR EACH FOIL.
110 DO 140 NFOIL =1,NFLS
    CALL ELMER
    F(NFOIL) = 0.                $ TOTAL = 0.
    DO 120 L=LMIN,LMAX,LOEL
C  WEIGHT F(NFOIL) IN INVERSE PROPORTION TO THE RELATIVE STANDARD DEVIATION.
    F(NFOIL) = F(NFOIL) + FA(L)/RELSIG(IS,L)
120 TOTAL = TGTAL + 1.0/RELSIG(IS,L)
140 F(NFOIL) = F(NFOIL)/TOTAL

C  CALCULATE (LOGARITHMIC) COUNTING RATES, ALTERED BY FOIL FACTOR.
    DO 160 K=1,NCDS                $ TT(K) = T(IS,K)
    CALL NIMFO(K)                  $ CRATE(K) = RATE(IS,K)-F(NFOIL)
160 WGT(K) = 1.0 / RELSIG(IS,K)

C  DO LEAST-SQUARES FIT TO ALL POINTS FOR CURRENT SCALER.
    MCYCL = 0
180 CALL LSQPOL (TT, CRATE, WGT, REZID, NCARDS(IS), SUM, 1, ERMAT,
X      COEFF, ITCH(2), 400, 10)
    MCYCL = MCYCL + 1
    IF (MCYCL .GT. 1) GO TO 200      $ S2B = SUM          $ GO TO 220
200 S2A = S2B      $ S2B = SUM      $ DELS = ABSF((S2B - S2A) / S2B)
    IF (DELS .LE. CONCR(2)) GO TO 240
    IF (MCYCL .GE. ITCH(5)) GO TO 240
220 CALL MINFAC          $ IF (ITCH(7) .NE. 3) GO TO 180

C      DIAGNOSTIC PRINTOUT
      PRINT9620,MCYCL,SUM,(INFOIL,MCYCL(NFOIL),F(NFOIL), NFOIL=1,NFLS)
      GO TO 180

C
240 PRINT 9010, ISCALR(IS), MCYCL, SUM, DELS, NCARDS(IS), NCOUNTS,
X      NFOILS(IS)

C  FIND ERROR IN LEAST-SQUARES COEFFICIENTS. IF ANY ERRORS ARE NEGATIVE, ILL-
C  CONDITIONING IS INDICATED. THESE ARE FLAGGED WITH JERCO. ERRORS CALCULATED
C  ACCORDING TO BURTON GARBOW (PRIVATE COMMUNICATION).
255 DO 280 MM=1,ITCH2      $ JERCO(MM)=0
    ERCO(MM) = ((ERMAT(MM,MM)*SUM) / (NCARDS(IS) - ITCH(2)))
    IF (ERCO(MM) .GE. 0.0) GO TO 260
    ERCO(MM) = -ERCO(MM)      $ JERCO (MM) = 1
260 ERCO(MM)= ABSF(SQRT(ERCO(MM)) / COEFF(MM))
280 CONTINUE          $ PRINT 9020,NREM
    IF (DELS .LE. CONCR(2)) GO TO 300
    PRINT 9029, ( MM, COEFF(MM), ERCO(MM), JERCO(MM), MM=1,ITCH2)
    GO TO 310
300 PRINT 9030, ( MM, COEFF(MM), ERCO(MM), JERCO(MM), MM=1,ITCH2)
310 PRINT 9031

320 CALL CATOUT $ CALL NEWPAGE      $ CALL GRAFFIT
C  DIAGNOSTIC PRINTOUT OF FIRST TEN TIME VALUES...
    IF (ITCH(7) .EQ. 2) PRINT 9610, (T(K), K=1,10)
C  END OF MAJOR DO LOOP. PROCEED TO NEXT SCALER, OR TO NEW PROBLEM.
400 CONTINUE          $ PRINT 9040, ITITLE
    GO TO 10          $ END

```

```

SUBROUTINE CONIN
C      READS AND PRINTS THE INPUT CONSTANTS

8010 FORMAT ( 9A8)
8020 FORMAT(4X,E8.0,I1,1X,A4,2X,6E10.5)
8030 FORMAT (4X, 6I1,E9.0, A1, 6E10.5)
8040 FORMAT (10X, 10I3, 1E5.0, 2E10.5, 3E5.0)
8050 FORMAT (10X, 6E10.5, 9I1)
9010 FORMAT (2X      A5 * COUNTING ROOM -- FISSION-PRODUCT LEAST*
X*-SQUARES FIT* / )
9011 FORMAT (2X      A5 * COUNTING ROOM -- DECAY-CONSTANT LEAST*
X*-SQUARES FIT * / )
9012 FORMAT (2X      A5 * COUNTING ROOM -- FIXED DECAY CONSTANT * / )
9020 FORMAT (      * LISTING OF INPUT CONSTANTS      *
B      /      * HALF-LIFE      * F11.3,6X,A1
C      /      * DECAY CONSTANT      * E16.4 * /SEC*
D      /      * SCALER MULTIPLIER      * F10.2
E      /      * DEAD-TIME COEFFICIENTS      * F12.4
F      /      *      * F12.4
G      /      *      * F12.4
H      /      * NO. OF HIGHEST SCALER      * I7
I      /      * BACKGROUND ADJUSTMENT FACTOR      * F11.3
J      /      * BACKGROUND MULTIPLIER      * F12.4
K      /      * IRRADIATION TIME      * F10.2,7X *MIN*
L      /      * CLOCK CORRECTION FOR SCRAM TIME* F10.2,7X *MIN*
M      /      * CLOCK READING      * F10.2,7X *SEC*
N      //      * SCALER EFFICIENCIES * 9X12, E16.4
G      /(      *      * 9X12, E16.4      )
9030 FORMAT ( 6P/ * SCALER DEAD TIMES*12X12,F11.3,6X*MICROSECONDS*/
X      (      *      *12X12,F11.3,5X*      ..      * )
9040 FORMAT (      * ITCH NUMBERS      *
B      /      * 1. COLLATION      * 14
C      /      * 2. NO OF COEFFS IN LSQ FIT      * 14
D      /      * 3. DIVISOR TO GET DELF      * 14
E      /      * 4. MAX CYCLES FOR FOIL FACTOR      * 14
F      /      * 5. MAX CYCLES FOR LSQ FIT      * 14
G      /      * 6. PUNCHED-CARD OPTIONS      * 14
H      /      * 7. DIAGNOSTIC PRINTOUT      * 14
I      /      * 8. GOLD-INDIUM WGT CORRECTIONS      * 14
J      /      * 9. DECAY, SINGLE-HALF-LIFE FOIL      * 14
K      /      * 10. ACCIDENTAL COINCIDENCE CORR.      * 14 )
9045 FORMAT ( /      * CONVERGENCE CRITERIA      *
2      /      * 1. FOR FOIL FACTOR      * E15.3
3      /      * 2. FOR LSQ FIT      * E15.3 )
9047 FORMAT ( /      * GOLD-INDIUM WGT-CORR. CONSTANTS      *
2      /      * CADMIUM RATIO      *F11.3
3      /      * NOMINAL FOIL WEIGHT      *F14.6
4      /      * HR      *F12.4 )
9048 FORMAT ( 55X 44H*****NOTE**** CDR NOT SPECIFIED... THE VALUE F7.3
X      * HAS ARBITRARILY BEEN USED. * )
9049 FORMAT ( 55X 12H*****NOTE**** * NOMINAL FOIL WEIGHT NOT SPECIFIED.*
X      / 55X 12X* ORDINARY WEIGHT CORRECTIONS HAVE BEEN MADE.*)
9050 FORMAT (//      * DATA CARDS ARE ASSUMED TO BE UNCOLLATED, COUNTED*
X      * NON-REVERSIBLY.*)
9051 FORMAT (//      * DATA CARDS ARE ASSUMED TO BE COLLATED. * )
9052 FORMAT (//      * DATA CARDS ARE ASSUMED TO BE UNCOLLATED, COUNTED*
X      * REVERSIBLY.*)
9055 FORMAT (//      * FIXED DECAY CONSTANTS USED, TO CALCULATE*
X      * SATURATED ACTIVITIES.*)
9056 FORMAT (// * ACCIDENTAL COINCIDENCE CORRECTIONS REQUESTED*
2      /* CHANNEL A SCALER* I4,216
3      /* CHANNEL B SCALER* I4,216
4      /* COINCIDENCE CHANNEL SCALER* I4,216 )
9060 FORMAT ( / 60X, 15(1H*) / 60X, 3(1H*)* THE END *3H***
X      / 60X, 5 ( 3H*** ) )
9600 FORMAT (* HALF-LIFE IMPROPERLY SPECIFIED (*A1*). EXECUTION*)

```

```

X      * TERMINATED.*
9610 FORMAT ( 6H ***** * ERROR IN SPECIFYING COINCIDENCE SCALERS. *
X      * ACCIDENTAL COINCIDENCE CORRECTIONS NOT MADE.* 6H ***** )

```

```

COMMON/A/BKW,      CMULT,      COEFF(1C),  CONCR(6),  CRATE(400),  COM1
2      DT(6),      DTB,DTC,DTD,EFC(6),  ENTAU(6,400),F (201),  COM2
3      FWT(6,201), IMAX, IS,  ISCALR(6),  ISMAX,  ITCH(10),  COM3
4      KOUNTS(201),KROOM,      LAMBDA,      LDEL,      LMAX, LMIN,  COM4
5      NAME(6,201), NCARD(6,400),NCARDS(6), NCCOUNTS,  NCYCL(201),  COM5
6      NEXTRA,      NFOIL,      NFOILS(6),  NREM,      RATE(6,400),  COM6
7      RELSIG(6,400),T (6,400),  TCORR,      WGT(400),  NTRAVS  COM7

COMMON/DATWT/ JNT(6,201), CDR, HR, FWZ
COMMON /PAGEBLK/ NPAGE, ITITLE(9)
COMMON/DECA/ TIRRAD, TSCRAM, CTIM(6,400), ACT(400)
COMMON/ACC/ INCA(6), INCB(6), MAC(6)
COMMON /CC/ JSTOP, LL(6),BKF
DIMENSION MYRTLE(9) , NS(6)

```

```
TYPE REAL LAMBDA
```

```
NPAGE = 0
```

```
IF (JSTOP .GE. 2) GO TO 5      $ JSTOP = 0
```

```

C READ FIRST CARD
  READ 8010,ITITLE
  IF (EOF,60) 5,1C
  5 PRINT 9060      $ STOP
  10 CALL NEWPAGE

```

```

C READ SECOND CARD
  READ 8020, BKF, IMAX, JROOM, BKW, DTB, DTC, DTD, (CONCR(K),K=1,2)
  IF (BKF.EQ.0.0) BKF = 1.0
  IF (CONCR(1) .EQ. 0.0) CONCR(1) = 10. ** (-2) * 4.
  IF (CONCR(2) .EQ. 0.0) CONCR(2) = 10. ** (-3) * 4.
  CMULT = 1.
  IF (JROOM .EQ. 4HNOLD) CMULT = 2.0
  KROOM = 5HSOUTH
  IF (JROOM .EQ. 4HNOLD) KROOM = 5HN OLD
  IF (JROOM .EQ. 4HN ) KROOM = 5HNORTH
  IF (IMAX .NE. 0) GO TO 20
  IF (KROOM .EQ. 5HSOUTH) IMAX = 4
  IF (KROOM .EQ. 5HN OLD) IMAX = 6
  IF (KROOM .EQ. 5HNORTH) IMAX = 6      $ GO TO 60
20 IF (KROOM .EQ. 5HSOUTH) GO TO 40
  IF (IMAX .GT. 6) IMAX = 6      $ GO TO 60
40 IF (IMAX .GT. 4) IMAX = 4
60 IF (BKW .LE. 0.0) BKW = 0.5
  IF (BKW .GT. 1.0) BKW = 1.0
  IF (DTB .NE. 0.0) GO TO 80
  DTB = 1.5      $ DTC = 8.0/3.C      $ DTD = 125./24.

```

```

C READ THIRD CARD
80 READ 8030, (NS(I), I = 1,6), TTHALF, MULT, (EFC(I),I=1,IMAX)
  IF (TTHALF .EQ. 0.0) GO TO 160      $ THALF = 0.0
  IF (MULT .EQ. 1HS) THALF = TTHALF
  IF (MULT .EQ. 1HM) THALF = TTHALF * 60.
  IF (MULT .EQ. 1HH) THALF = TTHALF * 3600.
  IF (MULT .EQ. 1HD) THALF = TTHALF * 3600. * 24.
  IF (MULT .EQ. 1HY) THALF = TTHALF * 3600. * 24. * 365.2422
  IF (MULT .EQ. 1H ) 100,120
100 THALF = 0.0      $ MULT = 1HL
  LAMBDA = TTHALF      $ GO TO 160
120 IF (THALF .NE. 0.0) GO TO 140
  PRINT 9600, MULT      $ GO TO 300
140 LAMBDA = 0.69315 / THALF
160 IF (TTHALF .EQ. 0.0) LAMBDA = 0.0
C NON-ZERO NS(I) DELETES PROCESSING FOR SCALER(I) BY MAKING EFC(I) EQUAL ZERO.

```



```

      DO 162 I=1,IMAX          $ IF(EFC(I) .EQ. 0.0) EFC(I) = 1.0
      IF (NS(I) .NE. 0) EFC(I) = 0.0
162 CONTINUE

```

```

C  READ FOURTH CARD
      ICDR = 0
      READ 8040, (ITCH(J),J=1,10),TTIRRAD,TTSCRAM, TCLOCK, CDR, FWZ, HR
      IF (HR .EQ. 0.) HR = 0.43          $ IF (CDR .NE. 0.) GO TO 165
      ICDR = 1          $ CDR = 2.

C  TTSCRAM IS TIME IN MINUTES FROM END OF IRRADIATION TO INSTANT WHEN INITIAL
C  CLOCK READING (TCLOCK) OCCURRED.
C  TSCRAM IS TIME IN SECONDS FROM END OF IRRADIATION TO INSTANT WHEN CLOCK
C  READING WAS ZERO.
C  IF TTIRRAD WAS NOT SPECIFIED, USE 8.88 MINUTES.
165 IF (TTIRRAD .EQ. 0.0) TTIRRAD = 8.88
C  IF TTSCRAM WAS NOT SPECIFIED, USE 88.88 MINUTES.
      IF (TTSCRAM .EQ. 0.0) TTSCRAM = 88.8
      TIRRAD=TTIRRAD * 60.    $ TSCRAM = TTSCRAM * 60. - TCLOCK
      TCORR = TSCRAM + 0.5 * TIRRAD
      IF (ITCH(2) .EQ. 1) ITCH(2) = 2

C  IF NO. OF COEFFICIENTS IS NOT SPECIFIED, USE 2 FOR DECAY-CONSTANT FIT,
C  4 FOR FISSION-PRODUCT FIT.
      IF (TTHALF .NE. 0.0) GO TO 170
      IF((ITCH(2) .LE. 0) .OR. (ITCH(2) .GT.10)) ITCH(2) = 4
      GO TO 175
170 IF((ITCH(2) .LE. 0) .OR. (ITCH(2) .GT.10)) ITCH(2) = 2
175 IF (ITCH(3) .LE. 0) ITCH(3) = 1000
      IF (ITCH(4) .LE. 0) ITCH(4) = 50
      IF (ITCH(5) .LE. 0) ITCH(5) = 10

C  READ FIFTH CARD
      READ 8050, (DT(I),I=1,6), INCA(1), INCB(1), MAC(1), INCA(2),
      X INCB(2), MAC(2), INCA(3), INCB(3), MAC(3)
      DO 180 I = 1,IMAX
180 DT(I) = DT(I) * 10.**(-6)

C  PRINT CONTENTS OF CONSTANTS DECK
      IF (LAMBDA .NE. 0.0) GO TO 200
      PRINT 9010, KROOM          $ GO TO 220
200 IF (ITCH(9) .EQ. 0) PRINT 9011, KROOM
      IF (ITCH(9) .EQ. 1) PRINT 9012, KROOM
220 PRINT 9020, TTHALF, MULT, LAMBDA, CMULT, DTB, DTC, DTC,
      X IMAX, BKW,BKF,TTIRRAD, TTSCRAM, TCLOCK,
      X (I, EFC(I), I=1,IMAX)
      PRINT 9030, (I,DT(I), I=1,IMAX)
      PRINT 9040, (ITCH(J), J=1, 10 )
      IF (ITCH(8) .NE. 1) GO TO 230    $ PRINT 9047, CDR, FWZ, HR
      IF (ICDR .EQ. 1) PRINT 9048, CDR
      IF (FWZ .NE. 0.) GO TO 230    $ ITCH(8)=0          $ PRINT 9049
230 IF (LAMBDA .EQ. 0.0) ITCH(9) = 0
      IF (ITCH(9) .EQ. 0) PRINT 9045, (CONCR(K), K=1,2 )
      IF (ITCH(10) .EQ. 0) GO TO 250
      PRINT 9056, (INCA(J),J=1,3),(INCB(J),J=1,3),(MAC(J),J=1,3)
C  THE TEST FOR INCA(J), ETC. .GT. IMAX OCCURS IN SUBROUTINE DCARDS.
      IF ((INCA(1).LE.0) .OR. (INCB(1).LE.0) .OR. (MAC(1).LE.0))240,250
240 ITCH(10) = 0          $ PRINT 9610
250 IF (ITCH(9).NE. 0) PRINT 5055
      IF (ITCH(1).EQ. 0) PRINT 5050
      IF (ITCH(1).EQ. 1) PRINT 5051
      IF (ITCH(1).EQ. 2) PRINT 5052
      RETURN

C  IF FATAL ERROR OCCURRED, PASS ON TO NEXT PROBLEM. (COMES HERE FROM 120+1)
300 DO 310 J = 1,2
310 READ 8010, MYRTLE
320 READ 8010, MYRTLE          $ IF (EOF,60) 5,330
330 DO 340 J = 1,4          $ IF (MYRTLE(J) .NE. 8H          ) GO TO 320
340 CONTINUE          $ JSTOP = 1          $ RETURN          $ END

```

SUBROUTINE WTCARDS

READS WEIGHT-CARD DECK

C
C

```

8010 FORMAT ( A6, A4, E10.5, 1CX, A2, 6A8)
8600 FORMAT ( A6, 9A8, A2)
8610 FORMAT ( 4A8 )
9010 FORMAT ( * FOIL WEIGHT INPUT DATA * /
X      /5X * NO. FOIL=7X *WEIGHT* 13X7HCOMMENT )
9020 FORMAT ( * SCALER* I2 )
9030 FORMAT ( 17X, A6 )
9031 FORMAT ( 17X, A6, 9A8, A2)
9040 FORMAT ( 17X, A6, 30X, I4 * FOILS, NO WEIGHT CARDS. * )
9050 FORMAT (/ 12X, I3, 2X, A6, 3X, A4, F13.5, * TOO MANY WT CARDS. EXEC*
X      *UTION STOPPED. * )
9060 FORMAT (12X I3, 2XA6, 3X, A4, F13.5, 10X, A2, 4X6A8)
9065 FORMAT ( 50X 12H****NOTE**** * NOMINAL FOIL WEIGHT IS TOO*
2      * DIFFERENT FROM FOIL WEIGHT.*
3      / 62X * ORDINARY WEIGHT CORRECTIONS HAVE BEEN USED.*)
9070 FORMAT ( 1H1 )
9600 FORMAT ( * WEIGHT-CARD DECK DID NOT TERMINATE CORRECTLY. EXECUTI*
2      *ON STOPPED. *
3      // * CARD IMAGE * A6, 9A8, A2 )
9610 FORMAT ( * WEIGHT-CARD DECK MUST NOT START WITH REPEAT CARD. EXE*
X      *CUTION TERMINATED. * )
9620 FORMAT ( * NEWSET CARD MUST NOT BE USED EXCEPT AFTER A FOIL WEIG*
X      *HT CARD. EXECUTION TERMINATED. * )
9630 FORMAT(17X*TOO MANY FOILS CALLED FOR ON NONE CARD (MAX. 200). *
X      * EXECUTION ASSUMES ONLY ONE FOIL FOR THIS SCALER. * )
9631 FORMAT(17X*NUMBER OF FOILS NOT SPECIFIED ON NONE CARD. ONE FOIL*
X      * IS ASSUMED FOR THIS SCALER. * )
9632 FORMAT ( 30X * BAD WEIGHT CARD OR IMPROPER TERMINATION OF WEIGHT*
X      * DECK. EXECUTION TERMINATED.* /30X * CARD CONTAINED*)

COMMON/A/BKW, CMULT, COEFF(1C), CONCR(6), CRATE(400), COM1
2 DT(6), DTB, DTC, DTD, EFC(6), ENTAU(6,400), F (201), COM2
3 FWT(6,201), IMAX, IS, ISCALR(6), ISMAX, ITCH(10), COM3
4 KOUNTS(201), KROOM, LAMBDA, LDEL, LMAX, LMIN, COM4
5 NAME(6,201), NCARD(6,400), NCARDS(6), NCOUNTS, NCYCL(201), COM5
6 NEXTRA, NFOIL, NFOILS(6), NREM, RATE(6,400), COM6
7 RELSIG(6,400), T (6,400), T CORR, WGT(400), NTRAVS COM7
COMMON/CATWT/ JWT(6,201), CDR, HR, FWZ
COMMON/DC/ JSTOP, LL(6), BKF
COMMON /PAGEBLK/ NPAGE, ITITLE(9)

DIMENSION MESSAGE(6), MUMSI(10)

PRINT 9010
I = IS = 0 $ MIMSI = 6HWEIGHT $ NSET = 2
10 I = I + 1 $ IF (I .LE. 6 ) GO TO 20
12 READ 8600, MIMSI, MUMSI
PRINT 9031, MIMSI, MUMSI
IF (MIMSI.EQ.6HLAST+1) GO TO 160
IS = IS+1 $ IF (IS.LE.6) GO TO 12
PRINT 9600, MIMSI, MUMSI $ GO TO 200

20 IF (EFC(I) .EQ. 0.0) GO TO 10
PRINT 9020, I $ IS = IS + 1 $ ISCALR(IS) = I
IF (MIMSI.NE.6HREPEAT) GO TO 40
25 NFOILS(IS) = NFOILS(IS-1) $ PRINT 9030, MIMSI
N = NFOILS(IS)
DO 30 NF = 1, N
JWT(IS, NF) = JWT(IS-1, NF)
NAME (IS, NF ) = NAME (IS-1, NF )
30 FWT (IS, NF ) = FWT (IS-1, NF)

```

```

MIMSI = 6HWEIGHT          $ GO TO 10
40 IF (MIMSI .NE. 6HNONE ) GO TO 60
    NFOILS(IS) = XINTF (FWT(IS-1,NFOIL) + .0001)
45    PRINT 9040, MIMSI, NFOILS(IS)
    N = NFOILS(IS)      $ IF (N .LT. 201) GO TO 47      $ PRINT 9630
    NFOILS(IS) = 1      $N = 1                          $ GO TO 48
47    IF (N .GT. 0) GO TO 48                          $ PRINT 9631
    NFOILS(IS) = 1      $ N = 1
48    DO 50 NF=1,N      $ NAME(IS,NF)= 4H
50    FWT (IS,NF ) = 1.0      $ MIMSI = 6HWEIGHT $ GO TO 10
60 NFOIL = 0
70 NFOIL = NFOIL + 1
    READ 8010, MIMSI, NAME(IS,NFOIL), FWT(IS,NFOIL),JWT(IS,NFOIL),
X    MESSAGE
    IF (MIMSI .EQ. 6H ) MIMSI = 6HWEIGHT
    IF (MIMSI .EQ. 6HNEWSET .OR. MIMSI .EQ. 6HLAST+1) PRINT 9030,MIMSI
    IF (MIMSI .NE. 6HREPEAT) GO TO 75
        NSET = 2
        IF((1.EQ.1).AND.(NFOIL.EQ.1))72,73
72    PRINT 9610      $ GO TO 200
73    IF (NREP .EQ. 2) GO TO 25 $ NREP=2 $ GO TO 80
75 NREP=1      $ IF (MIMSI.EQ.6HNONE ) NREP=2
    IF (MIMSI .NE. 6HNEWSET) GO TO 76
        IF (NSET .EQ. 1) GO TO 80      $ PRINT 9620
        GO TO 200
76 IF (MIMSI .EQ. 6HNONE ) NSET = 2
    IF (MIMSI .EQ. 6HWEIGHT) NSET = 1      $ GO TO 90
80    NFOILS(IS) = NFOIL-1      $ GO TO 10
90 IF ( MIMSI .NE. 6HNONE ) GO TO 100
    IF (NFOIL .GT. 1 ) GO TO 80
    NFOILS(IS) = XINTF (FWT(IS,NFOIL) + 0.0001)      $ GO TO 45
100 IF ( MIMSI .EQ. 6HLAST+1) GO TO 135
    IF (NFOIL .LT. 201) GO TO 120
    PRINT 9050, NFOIL, MIMSI, NAME(IS,NFOIL), FWT(IS,NFOIL)
    GO TO 200
120 IF (MIMSI .EQ. 6HWEIGHT) GO TO 125      $ PRINT 9632
    PRINT 9060, NFOIL, MIMSI, NAME(IS,NFOIL), FWT(IS,NFOIL),
X    JWT(IS,NFOIL), MESSAGE      $ GO TO 200
125 IF (FWT(IS,NFOIL) .NE. 0.0) GO TO 130      $ FWT(IS,NFOIL) = 1.0
    MESSAGE(1) = 8H *** WE      $ MESSAGE(2) = 8HIGHT NOT
    MESSAGE(3) = 8H SPECIFI      $ MESSAGE(4) = 8HED ***
130 PRINT 9060, NFOIL, MIMSI, NAME(IS,NFOIL), FWT(IS,NFOIL),
X    JWT(IS,NFOIL), MESSAGE
    IF (ITCH(8) .NE. 1) GO TO 70      $ WTEST = FWZ / FWT(IS,NFOIL)
    IF ((WTEST .LT. 2.) .AND. (WTEST .GT.0.5)) GO TO 70
    ITCH(8) = 0      $ PRINT 9065      $ GO TO 70
135 IF (NSET .EQ. 1) GO TO 140      $ ISMAX = IS-1      $ RETURN
140 NFOILS(IS) = NFOIL - 1
160 ISMAX = IS
C  DIAGNOSTIC PRINTOUT
    IF (ITCH(7).NE.1) GO TO 180      $ PRINT 907C
    DO 500 IS=1,ISMAX      $ PRINT 9020, ISCALR(IS)
    N = NFOILS(IS)
    DO 500 NF=1,N
500 PRINT 9060, NF ,6HWEIGHT, NAME(IS,NF), FWT(IS,NF)
180 RETURN

C
C  COMES HERE IF FATAL ERROR IS DETECTED. FIND END OF PROBLEM.
200 READ 8610, (MUMSI(I), I=1,4)      $ IF (EOF,60) 210, 220
210 STOP
220 DO 230 I = 1,4      $ IF (MUMSI(I) .NE. 8H ) GO TO 200
230 CONTINUE      $ JSTOP = 1      $ RETURN      $ END

```

C SUBROUTINE CCARCS
READS COUNTING DATA

```

8005 FORMAT ( A5, F8.1, F6.1, 6F8.C, 1X6I1)
8010 FORMAT ( A5, F6.0, F5.0, 6F8.C, 2X6I1)
8011 FORMAT ( A5, F9.1, 1XF9.1, 1X 4F9.C, 1X4I1)
9010 FORMAT ( * LISTING OF INPUT COUNTING DATA * )
9020 FORMAT ( * CARD CLOCK INTERVAL* 6(3X6HSCALER I2) NORTH
X * IDC NOS. *) NORTH
9021 FORMAT ( * CARD CLOCK INTERVAL* 4(3X6HSCALER I2) NORTH
X * IDC NOS. *) SOUTH
9030 FORMAT ( 1X I4, 1X A5, 2F12.1, 6F11.0, 4X 6I1) SOUTH
9031 FORMAT ( 1X I4, 1X A5, 2F12.1, 4F11.0, 5X 4I1) SOUTH
9035 FORMAT (11H BKG RATES 27X 6F11.3)
9040 FORMAT ( * END OF INPUT DATA FOR * 9A8 )
9050 FORMAT ( * COUNT LIMIT OF 400 EXCEEDED FOR SCALER* I2 *. LAST*
X I4 * COUNTS HAVE BEEN IGNORED. *)
9051 FORMAT ( * COUNT LIMIT OF 400 EXCEEDED FOR SCALER* I2, 1H( I4
2 * EXTRA COUNTS). SINCE CARDS ARE COLLATED, DATA HAVE*
3 / * NOT BEEN PROCESSED. * )
9060 FORMAT (1H1, 123X, *PAGE* I4)
9070 FORMAT (* THERE ARE*I3* SCALERS CALLED FOR BY WEIGHT-CARD DECK,*
2 * BUT*I3* ACCORDING TO INPUT FOR NS(I) AND IMAX. */
3 * DATA HAVE BEEN PROCESSED USING THE SMALLER NUMBER, BUT*
4 * YOU MIGHT WANT TO RE-RUN THE PROBLEM. * )
9080 FORMAT ( 6H **** *COINC. SCALER NUMBER GREATER THAN IMAX. ACC.*
X * COINC. CORRECTIONS HAVE NOT BEEN MADE.* 5H **** )

```

```

COMMON/A/BKW, CMULT, COEFF(1C), CONCR(6), CRATE(400), COM1
2 DT(6), DTB, DTC, DTD, EFC(6), ENTAU(6,400), F(201), COM2
3 FWT(6,201), IMAX, IS, ISCALR(6), ISMAX, ITCH(10), COM3
4 KCOUNTS(201), KROOM, LAMBDA, LDEL, LMAX, LMIN, COM4
5 NAME(6,201), NCARD(6,400), NCARDS(6), NCOUNTS, NCYCL(201), COM5
6 NEXTRA, NFOIL, NFOILS(6), NREM, RATE(6,400), COM6
7 RELSIG(6,400), T(6,400), TCORR, WGT(400), NTRAVS COM7
COMMON /DC/ JSTOP, LL(6), BKF
COMMON /PAGEBLK/ NPAGE, ITITLE(9)
COMMON/DECA/ TIRRAD, TSCRAM, CTIM(6,400), ACT(400)
COMMON/ACC/ INCA(6), INCB(6), MAC(6)

```

```

DIMENSION BKG(6), DATA(6), IDC(6), KLINK(6), KLANK(6),
2 KLUNK(6), RATE1(6), RATEA(6), ISC(6), ISA(6),
3 ISB(6), NOC(6), SIG2BK(6), LBK(6), BK(6)

```

```

PRINT 9010
JSTOP = L = L1 = 0 $ IBK = 0
DO 20 IS= 1,6 $ BK(IS) = 0.0
BKG(IS) = 0.0 $ KLINK(IS) = C $ KLANK(IS) = 0
20 KLUNK(IS) = LL(IS) = LBK(IS) = 0

```

```

C COMPARE ISMAX FROM WTCARDS WITH INPUT FOR NS(I) AND IMAX.
IS=0 $ DO 30 I=1,IMAX $ IF(EFC(I).NE.0.) IS=IS+1
30 CONTINUE $ IF(IS.EQ.ISMAX) GO TO 32
PRINT 9070, ISMAX, IS $ ISMAX = XMINCF(IS, ISMAX)
32 IF (KROOM .EQ. 5HN OLD) PRINT 9020, (I, I=1,6)
IF (KROOM .EQ. 5HNORTH) PRINT 9020, (I, I=1,6)
IF (KROOM .EQ. 5HSOUTH) PRINT 9021, (I, I=1,4)
IF (ITCH(10) .EQ. 0) GO TO 40

DO 635 J = 1,3 $ IF ((INCA(J) .GT. IMAX) .CR.

```

```

      X      (INCB(J) .GT. IMAX) .OR. (MAC(J) .GT. IMAX))633,635
633 ITCH(10) = C      $ PRINT 9080      $ GO TO 40
635 CONTINUE
C STORE SCALER NUMBERS FOR ACCIDENTAL COINCIDENCE CORRECTIONS
  IS = 0
  DO 640 J = 1,3
  DO 640 I = 1, IMAX      $ IF(EFC(I).EQ.C) GO TO 640      $ IS = IS+1
  IF (I .EQ. INCA(J)) ISA(J) = IS      $ IF(I .EQ. INCB(J)) ISB(J) = IS
  IF (I .EQ. MAC(J)) ISC(J) = IS
640 CONTINUE

C   CARDS ARE COUNTED BY L, LINES BY L1.
40 L = L+1
41 L1 = L1 + 1
C SEE WHETHER TO START ANOTHER PAGE
  DO 45 J = 1,100      $ JP = 55 + 60*(J-1)
  IF(L1-JP) 47,46,45
45 CONTINUE
46 NPAGE = NPAGE + 1      $ PRINT 9060, NPAGE
47 CONTINUE
  IF (IBK .EQ. 0) GO TO 48      $ IBK = 0
  PRINT 9035, (BK(I), I = 1,IMAX)      $ GO TO 41
48 DO 49 IS = 1,ISMAX
49 NOC(IS) = 0
  IF (KROOM .EQ. 5HNORTH) READ 8005,KARD,CLOCK,CTIME,(DATA(I),I=1,6)
      X      ,      (IDC(I),I=1,6)
  IF (KROOM .EQ. 5HN OLD) READ 8010,KARD,CLOCK,CTIME,(DATA(I),I=1,6)
      X      ,      (IDC(I),I=1,6)
  IF (KROOM .EQ. 5HSOUTH) READ 8011,KARD,CLOCK,CTIME,(DATA(I),I=1,4)
      X      ,      (IDC(I),I=1,4)
C BLANK CARD OR END-OF-FILE TERMINATES COUNTING DATA DECK.
  IF (EOF,60) 318, 50
50 IF (CTIME .EQ. 0.0) GO TO 320
  I = IS = 0
60 I = I+1      $ IF (I .LE. IMAX) GO TO 100
  IF (KROOM .EQ. 5HSOUTH ) GO TO 80
  PRINT 9030,L,KARD,CLOCK, CTIME,(DATA(I),I=1,6),(IDC(I),I=1,6)      NORTH
  IF (ITCH(10) .EQ. 0) 40, 302
80 PRINT 9031,L,KARD,CLOCK, CTIME,(DATA(I),I=1,4),(IDC(I),I=1,4)      SOUTH
  IF (ITCH(10) .EQ. 0) 40, 302
100 IF (EFC(I) .EQ. 0.0) GO TO 60
  IS = IS+1
  IF (IDC(I) .EQ. 0) GO TO 140
  IF (IDC(I) .NE. 8) GO TO 120      $ KLANK(IS) = 0      $ GO TO 300
120 IF ((IDC(I).EQ.3).OR.(IDC(I).EQ.5).OR.(IDC(I).EQ.7).OR.(IDC(I).EQ.
      X      9))140,200
140 IF (KLANK(IS) .NE. 0) GO TO 300
160 LL(IS)=L-KLUNK(IS)      $ IF (LL(IS) .LE. 400) GO TO 180
      KLINK(IS) = KLINK(IS) + 1      $ GO TO 300
180 RATE(I) = CMULT * (DATA(I)/CTIME)
  ENTAUM = RATE(I) * DT(I)
C CALCULATE COUNTING RATE, CORRECTED FOR BACKGROUND AND DEAD TIME.
  RATE1(I)=RATE(I)*(1+ENTAUM + DTB* ENTAUM**2 + DTC*ENTAUM
      X      **3 + DTD*ENTAUM**4)
  RATE(IS,LL(IS)) = RATE1(I) - BKG(IS) * BKF
C CALCULATE (APPROX.) FRACTIONAL COUNTING LOSSES.
  ENTAU (IS,LL(IS)) =(RATE1(I)) * DT(I)

C SKIP STATEMENT 187 IF THERE WILL BE A NEGATIVE ARGUMENT FOR THE SQRTF.
C NOTE THAT RELSIG BECOMES 88.8 IF ENTAU IS TOO BIG, AND 99.9 IF BKG IS TOO BIG

```

```

      IF (ENTAU(IS,LL(IS)) .LT. 1.0) GO TO 185
      RELSIG(IS,LL(IS)) = 88.8      $ GO TO 194
185    IF (RATE(IS,LL(IS)).GT. 0.0) GO TO 187
      RELSIG(IS,LL(IS)) = 99.9      $ GO TO 194
C    CALCULATE SQUARED STATISTICAL ERROR IN RATE1 ACCORDING TO FORMULA
C    BY C.E.COHN, RPD-EPM 36, FOR THE CASE OF THE PARALYZABLE COUNTER.
187    SIG2 = RATE1(I) * (EXPF(ENTAU(IS,LL(IS))) - 2.*ENTAU(IS,LL(IS)))
      X / (CTIME * (1. - (ENTAU(IS,LL(IS))) ** 2))
      RELSIG(IS,LL(IS)) = SQRTF (SIG2 + SIG2BK(IS)*BKF**2)
      X RATE(IS,LL(IS))
194    NCARD(IS,LL(IS)) = KARD
      T(IS,LL(IS)) = CLCK      $ CTIM(IS,LL(IS)) = CTIME
      GO TO 60

C    CONTINUED FROM STATEMENT 120.
200    IF (IDC(I) .EQ. 2) GO TO 240      $ IF (IDC(I) .EQ. 4) GO TO 280
      IF (IDC(I) .EQ. 1) GO TO 260      $ IF (IDC(I) .EQ. 6) GO TO 300
C    SIG2BK(IS) IS SQUARE OF ABSOLUTE ERROR IN BACKGROUND RATE.
240    BKG(IS) = CMULT*DATA(I)/CTIME      $ SIG2BK(IS) = BKG(IS)/CTIME
      LBK(IS) = LBK(IS)+1      $ GO TO 270
C    LBK(IS) COUNTS THE BACKGROUND CARDS.
C    IF IDC(I) = 1 FOR FIRST BACKGROUND COUNT, TREAT COUNT AS THOUGH IDC(I) = 2.
260    IF (LBK(IS) .EQ. C) GO TO 240      $ LBK(IS) = LBK(IS)+1
      BKG(IS) = BKW*BKG(IS) + (1.-BKW) * DATA(I)*CMULT/CTIME
      SIG2BK(IS) = BKW*SIG2BK(IS) + (1.-BKW)*DATA(I)*CMULT/(CTIME**2)
C    STORE BACKGROUND RATE FOR PRINTING (FOLLOWING STATEMENT 40).
270    BK(I) = BKG(IS)*BKF      $ IBK = 1      $ GO TO 300
280    KLANK(IS) = 1
C    NOC(IS) KEEPS TRACK OF BKG AND DELETED DATA, FOR COINC. CORRECTIONS (302+2).
300    KLUNK(IS) = KLUNK(IS)+1      $ NOC(IS)=1      $ GO TO 60

C    ACCIDENTAL COINCIDENCE CORRECTIONS (COMES HERE FROM NEAR STATEMENT 80).
302    DO 310 J = 1,3
C    BYPASS ACCIDENTAL-COINCIDENCE CORRECTIONS FOR A BACKGROUND CARD.
      NCC = NOC(ISC(J)) + NOC(ISA(J)) + NOC(ISB(J))
      IF (NCC .NE. 0) GO TO 308
      TBK = BK (MAC(J))-2.0*DT(MAC(J))*BK (INCA(J))*BK (INCB(J))
      RACC = 2.0 * DT(MAC(J)) * RATE1(INCA(J)) * RATE1(INCB(J))
      TCOIN = RATEA(MAC(J)) * (RATE1(INCA(J))/RATEA(INCA(J)))
      X * (RATE1(INCB(J))/RATEA(INCB(J)))
      RT = TCOIN - RACC - TBK
      RATE(ISC(J),LL(ISC(J))) = RT
308    IF (MAC(2) .EQ. 0) GO TO 40
      IF((J .EQ. 2) .AND. (MAC(3) .EQ. 0)) GO TO 40
310    CONTINUE      $ GO TO 40

C    PROGRAM COMES HERE FROM 50, WHEN BLANK CARD OR END OF FILE IS ENCOUNTERED.
318    JSTOP = 2
320    PRINT 9040, ITITLE
      DO 360 IS = 1,ISMAX      $ NCARDS(IS) = LL(IS)
      IF (KLINK(IS) .EQ. 0) GO TO 360      $ IF (ITCH(1) .EQ. 1) GO TO 340
      PRINT 9050, ISCALR(IS), KLINK(IS)      $ GO TO 360
340    PRINT 9051, ISCALR(IS), KLINK(IS)
C    JSTOP = 2 OR 3 MEANS END OF FILE. 1 OR 3 MEANS TOO MANY COLLATED CARDS.
      JSTOP = JSTOP + 1      $ GO TO 380
360    CONTINUE
380    RETURN      $ END

```


SUBROUTINE ELMER
C CALCULATES SEQUENCE NUMBERS LMIN, LMAX, AND LDEL FOR FOIL, FOR DC LCCPS.

COMMON/A/BKW,	CMULT,	COEFF(1C),	CONCR(6),	CRATE(4C0),	COM1
2 DT(6),	DTB,DTC,DTD,EFC(6),	ENTAU(6,4C0),F (201),			COM2
3 FWT(6,201),	IMAX, IS,	ISCALR(6),	ISMAX,	ITCH(1C),	COM3
4 KOUNTS(201),	KROOM,	LAMBDA,	LDEL,	LMAX, LMIN,	COM4
5 NAME(6,201),	NCARD(6,4C0),	NCARDS(6),	NCCOUNTS,	NCYCL(2C1),	COM5
6 NEXTRA,	NFOIL,	NFOILS(6),	NREM,	RATE(6,4C0),	COM6
7 RELSIG(6,4C0),	T (6,4C0),	TCORR,	WGT(4C0),	NTRAVS	COM7

IF (ITCH(1) .EQ. 1) GO TO 100

C FOR UNCOLLATED CARDS...
LDEL = NFOILS(1S) \$ LMIN = NFOIL
LMAX = LMIN + (NCCOUNTS * NFOILS(1S))
IF (LMAX .GT. NCARDS(1S)) LMAX = LMAX - NFOILS(1S)
RETURN

C FOR COLLATED CARDS...
100 LDEL = 1 \$ IF(NREM .NE. 0) GO TO 12C
LMIN = (NFOIL-1)*NCCOUNTS+1 \$ LMAX = LMIN+NCCOUNTS-1
RETURN
120 IF (NREM .GE. NFOIL) GO TO 14C
LMIN = (NCCOUNTS+1) * NREM + (NFOIL - NREM - 1)*NCCOUNTS + 1
LMAX = LMIN + NCCOUNTS - 1 \$ RETURN
140 LMIN = (NFOIL-1) * (NCCOUNTS+1) + 1
LMAX = LMIN + NCCOUNTS \$ RETURN \$ END

SUBROUTINE NIMFO (LL)
C DETERMINES FOIL NUMBER (NFOIL) FOR COUNT NO. LL.

COMMON/A/BKW,	CMULT,	COEFF(1C),	CONCR(6),	CRATE(4C0),	COM1
2 DT(6),	DTB,DTC,DTD,EFC(6),	ENTAU(6,4C0),F (201),			COM2
3 FWT(6,201),	IMAX, IS,	ISCALR(6),	ISMAX,	ITCH(1C),	COM3
4 KOUNTS(201),	KROOM,	LAMBDA,	LDEL,	LMAX, LMIN,	COM4
5 NAME(6,201),	NCARD(6,4C0),	NCARDS(6),	NCCOUNTS,	NCYCL(2C1),	COM5
6 NEXTRA,	NFOIL,	NFOILS(6),	NREM,	RATE(6,4C0),	COM6
7 RELSIG(6,4C0),	T (6,4C0),	TCORR,	WGT(4C0),	NTRAVS	COM7

IF (ITCH(1) .EQ. 1) GO TO 100

C FOR UNCOLLATED CARDS...
NFOIL=LL-((LL-1)/NFOILS(1S))*NFOILS(1S) \$ RETURN

C FOR COLLATED CARDS...
100 IF (NREM .GT. 0) GO TO 120
NFOIL = (LL-1)/NCCOUNTS + 1 \$ RETURN
120 IF (LL .GT. (NREM * (NCCOUNTS+1))) GO TO 140
NFOIL = 1 + (LL-1) / (NCCOUNTS+1) \$ RETURN
140 NFOIL = ((LL-1-NREM * (NCCOUNTS+1)) / NCCOUNTS) + NREM+1
RETURN \$ END

SUBROUTINE MINFAC
 C USING LEAST-SQUARES FIT FOUND IN MAIN PROGRAM, MINFAC DETERMINES FOR EACH
 C FOIL THE FOIL-FACTOR WHICH MINIMIZES THE SUM OF SQUARES OF RESIDUALS.

```

COMMON/A/EKW,      CMULT,      COEFF(10),  CONCNR(6),  CRATE(400),  COM1
2      DT(6),      DTB,DTC,DTD,EFC(6),  ENTAU(6,400),F (201),  COM2
3      FWT(6,201), IMAX, IS,   ISCALR(6),  ISMAX,      ITCH(10),  COM3
4      KOUNTS(201),KROOM,      LAMBDA,      LOEL,      LMAX, LMIN,  COM4
5      NAME(6,201), NCARD(6,400),NCARDS(6), NCOUNTS,    NCYCL(201),  COM5
6      NEXTRA,      NFOIL,      NFOILS(6),  NREM,      RATE(6,400),  COM6
7      RELSIG(6,400),T (6,400),  TCORR,      WGT(400),  NTRAVS  COM7

      DIMENSION RESID(400)

      NMAX = ITCH(4)          $ FITCH3 = FLOATF(ITCH(3))
      NFLS = NFOILS(15)      $ ITCH2 = ITCH(2)
      DO 200 NFOIL = 1,NFLS
      CALL ELMER
      F(NFOIL) = F(NFOIL) + COEFF(1)  $ NCYCL(NFOIL) = 0
100  SUMSQ = 0
      DO 140 L=LMIN,LMAX,LDEL  $ FIT = 0
      DO 120 M=2,ITCH2
120      FIT = FIT + COEFF(M) * (T(IS,L)**(M-1))
      CRATE(L) = RATE(IS,L) - F(NFOIL)
      RESID(L) = CRATE(L) - FIT
140      SUMSQ = SUMSQ + (RESID(L)**2) * WGT(L)
      DELF = F(NFOIL) / FITCH3
      CALL ITER 7 (NCYCL(NFOIL), SUMSQ, F(NFOIL), DELF, CONCNR(1),
      X      NMAX, ISTOP, DDDD)
      IF ((NCYCL(NFOIL).GE.NMAX).OR.(ISTOP.EQ.1)) 200,100
200  CONTINUE  $ RETURN  $ END
  
```

```

FUNCTION EXCIF(XA)
C  ACCURATE EVALUATOR OF EXP(X) - 1.0
C  CHARLES E. COHN, ARGONNE NATIONAL LABCRATORY
X=XA      $ XB = ABSF(X)
IF(XB.LE.16.)GOTO2
1 IF(X)3,3,4
3 EXCIF=1.
GOTO100
4 EXCIF=EXPF(X)
GOTO100
2 IF(XB.LE.(.2)) GO TO 6
5 EXCIF=EXPF(X)-1.
GOTO100
6 EXCIF=X/(1.-X*(.5-(X/12.) *. (1. - (X**2)/60.)))
100 RETURN  $ END
  
```

```

SUBROUTINE NEWPAGE
C
TYPE INTEGER TITLE
DIMENSION TITLE ( 9)
COMMON /PAGEBLK/ NPAGE, TITLE
NPAGE = NPAGE + 1
DATE = DATE1(DUMMY)
PRINT 1100, TITLE, DATE, NPAGE
1100 FORMAT (1H1// 2CX 9A8, 6X 10HPROCESSED A8, 6X 4HPAGE 14 / )
RETURN
END
  
```

C SUBROUTINE CATOUT
PRINTS PROCESSED FOIL DATA

```

9010 FORMAT (      *      FOIL  CYCLES  NO.  SEC.  CARD  COR.ACT*
2      *IVITY  SP.ACTIVITY  REL.DEV.,PC  NORM.ACTIVITY*
3      *  FRAC.DEAD TIME *)
9015 FORMAT ( 1X I4, 2X A4)
9020 FORMAT ( 1X I4, 2X A4, 3X I4)
9030 FORMAT ( 18X, 2I7, 3XA5, 2E15.5, 2PF12.4,OP, E17.5, 2X F12.4)
9045 FORMAT ( 1H1, 123X, *PAGE* I4 )
9050 FORMAT ( * SCALER* I2 *. AVERAGE ACTIVITIES, IN DISINTEGRATIONS/*
X      *UNIT WEIGHT/SEC., WITH ALL CORRECTIONS APPLIED. * / )
9060 FORMAT ( 12X * FOIL  WTD.AVERAGE  REL.STAT.DEV.,PC  MEAN  *
X      * REL.STD.DEV.,PC  STD./STAT.  NORM.WTD.AV. * / )
9070 FORMAT ( 3X I4, 6X A4, E14.5, 1X2PF14.5,OP,E17.5,1X2PF12.5,OP,
X      4X F12.3, 5X 3PF12.4, OP )
9075 FORMAT ( 87X, 5(1H-) / 80X F12.3, 4H AV. )
C FORMATS FOR COMBO CARDS, 9080-9084
9080 FORMAT ( 9A8, / 1X9A8, / 5HSCALR WTS (I 11, 9H 1.0 ) )
9081 FORMAT (      * SCALER* I2 * WTD-AVERAGE CARDS FOR FOILNORM. (USE*
X      * THIS CARD.) * )
9082 FORMAT ( A4, 5X E15.7)
9083 FORMAT ( 4HSCAL )
9084 FORMAT ( 4HNEXT / 4HLAST )
9090 FORMAT ( / * WEIGHTED-AVERAGE CARDS HAVE BEEN PUNCHED. * )
C FORMATS FOR CARDS FOR PROGRAM DEADTIME.
9104 FORMAT ( / * SPECIFIC-ACTIVITY CARDS HAVE BEEN PUNCHED. * )
9105 FORMAT ( 9A8, 8HDEADTIME )
9110 FORMAT ( 1X,E14.7, 39XI3, 1XI1, 4XA5, A4)

```

```

COMMON/A/BKW,      CMULT,      COEFF(10),  CONCR(6),  CRATE(400),  COM1
2      DT(6),      DTB,DTC,DTD,EFC(6),  ENTAU(6,400),F (201),  COM2
3      FWT(6,201), IMAX, IS,  ISCALR(6),  ISMAX,  ITCH(10),  COM3
4      KOUNTS(201),KROOM,      LAMBDA,      LDEL,      LMAX, LMIN,  COM4
5      NAME(6,201), NCARD(6,400),NCARDS(6),  NCOUNTS,  NCYCL(201),  COM5
6      NEXTRA,      NFOIL,      NFOILS(6),  NREP,      RATE(6,400),  COM6
7      RELSIG(6,400),T (6,400),  TCORR,      WGT(400),  NTRAVS  COM7
COMMON/CATWT/ JWT(6,201), CDR, HR, FWZ
COMMON /PAGEBLK/ NPAGE, ITITLE
COMMON/GRAF/  ACNORM(400)
COMMON/DECA/  TIRRAD, TSCRAM, CTIM(6,400), ACT(400)
DIMENSION
X      ERWACT(201),      ITITLE(5),  SPACT(400), WACT(201)

```

```

PRINT 9010      $ ITCH2 = ITCH(2)
IF ((ITCH(6) .GE. 2).AND. (IS .EQ. 1)) PUNCH 9105, ITITLE
K=0      $ NFLS = NFOILS(IS)      $ NCDS = NCARDS(IS)
DO 100 NFOIL=1,NFLS
CALL ELMER
TOTAL = AVACT(NFOIL) = WACT(NFOIL) = C
DO 80 L=LMIN,LMAX,LDEL      $ K=K+1
IF (ITCH(9) .NE. 0) GO TO 50
FIT = COEFF(1)
DO 46 M=2,ITCH2
46      FIT = FIT + COEFF(M)*(T(IS,L)**(M-1))
C      CONVERT ACT BACK FROM LOGARITHM.
ACT(L) = EXPF(RATE(IS,L) - FIT)
50      SPACT(K) = ACT(L)/FWT(IS,NFOIL)
IF (ITCH(8) .NE. 1) GO TO 70

FF = ((FWT(IS,NFOIL) - FWZ) / (FWT(IS,NFOIL) + FWZ)) * 2.
IF (JWT(IS,NFOIL) .NE. 2HCD) GO TO 60
C      WEIGHT CORRECTION FOR CADMIUM-COVERED GOLD OR INDIUM
SPACT(K) = SPACT(K) * (2.+HR*FF)/(2.-HR*FF)      $ GO TO 70
C      WEIGHT CORRECTION FOR BARE GOLD OR INDIUM
60      SPACT(K) = SPACT(K) / (1.-2.*FF*HR/ (CDR*(2.+HR*FF)))

70      AVACT(NFOIL) = AVACT(NFOIL) + SPACT(K)
WT = 1.0 / (SPACT(K) * RELSIG(IS,L)) ** 2

```

```

      WACT(NFOIL) = WACT(NFOIL) + WT*SPACT(K)
80      TOTAL = TOTAL + WT

      AVACT(NFOIL) = AVACT(NFOIL)/KOUNTS(NFOIL)
      WACT(NFOIL) = WACT(NFOIL)/TOTAL
100     ERWACT(NFOIL) = 1.0 / (WACT(NFOIL) * SQRTF(TOTAL))

C     NORMALIZE TO MAX. ACT.
      ACTMAX=0.
      DO 120 K=1,NCDS
120     ACTMAX = MAX1F(ACTMAX, SPACT(K))
      DO 140 K=1,NCDS
140     ACNORM(K) = SPACT(K)/ACTMAX

C     PRINT OUTPUT.  CALCULATE STD. DEV. IN AVERAGE ACTIVITIES.
      K = 0
      DO 200 NFOIL = 1,NFLS
      CALL ELMER
C     SEE WHETHER TO START ANOTHER PAGE...
      DO 151 J = 1,100          $ JP = 51-ITCH(2) + 58*(J-1)
      IF ((K+NFOIL) - JP) 153,152,151
151     CONTINUE
152     NPAGE = NPAGE+1 $ PRINT 9045,NPAGE $ PRINT 9010
153     CONTINUE
      IF (ITCH(9) .NE. 0) 154, 155
154     PRINT 9015, NFOIL, NAME(IS,NFOIL) $ GO TO 156
155     PRINT 9020, NFOIL, NAME(IS,NFOIL), NCYCL(NFOIL)
156     ERACT(NFOIL) = 0.0
      DO 180 L=LMIN,LMAX,LDEL          $ K=K+1
C     SEE WHETHER TO START ANOTHER PAGE...
      DO 161 J = 1,100          $ JP= 51-ITCH(2) + 58 *(J-1)
      IF ( K+NFOIL - JP) 163,162,161
161     CONTINUE
162     NPAGE = NPAGE+1 $ PRINT 9045,NPAGE $ PRINT 9010
163     CONTINUE
      PRINT 9030, K, L, NCARD(IS,L), ACT(L), SPACT(K), RELSIG(IS,L),
      X      ACNORM(K), ENTAU(IS,L)
C     PUNCH SP. ACTIVITY CARDS, FOR PROGRAM DEACTIME, IF ITCH(6) IS 2 OR 3.
      IF (ITCH(6) .GE. 2) PUNCH 9110,SPACT(K),K,ISCALR(IS),NCARD(IS,L),
      X      NAME(IS,NFOIL)
180     ERACT(NFOIL) = ERACT(NFOIL) + (SPACT(K)-AVACT(NFOIL))*2
      IF (KOUNTS(NFOIL) .GT. 1) GO TO 190
      ERACT(NFOIL) = 0.0          $ GO TO 200
190     ERACT(NFOIL) = (SQRTF ( ERACT(NFOIL) / ( KOUNTS(NFOIL)*( KOUNTS
      X      (NFOIL)-1)))) / AVACT(NFOIL)
200     CONTINUE          $ IF (ITCH(6) .GE. 2) PRINT 9104
C     END OF FOIL DO LOOP.
      CALL NEWPAGE

C     PRINT FOIL AVERAGES.
      PRINT 9050, ISCALR(IS)          $ PRINT 9060
C     NORMALIZE TO HOTTEST FOIL.
      HOT1 = 0.0          $ DO 210 NFOIL = 1,NFLS
210     HOT1 = MAX1F (HOT1, WACT(NFOIL) )          $ AVRAT = 0
      DO 220 NFOIL=1,NFLS
      RATIO = ERACT(NFOIL) / ERWACT(NFOIL)          $ AVRAT = AVRAT+RATIO
      AVNORM = WACT(NFOIL) / HOT1
220     PRINT 9070, NFOIL, NAME(IS,NFOIL), WACT(NFOIL), ERWACT(NFOIL),
      X      AVACT(NFOIL), ERACT(NFOIL) , RATIO , AVNORM
      AVRAT = AVRAT/NFLS          $ PRINT 9075, AVRAT
      IF (( ITCH(6).NE.1) .AND. (ITCH(6).NE.3)) RETURN

C     PUNCH CARDS FOR COMBO, IF ITCH(6) IS 1 OR 3.
      IF (IS .EQ. 1) PUNCH 9080, ITITLE, ITITLE, (ISCALR(II),II=1,ISMAX)
      PUNCH 9081, ISCALR(IS)
      DO 240 NFOIL = 1,NFLS
240     PUNCH 9082, NAME(IS,NFOIL), WACT(NFOIL)
      IF (IS .LT. ISMAX) PUNCH 9083
      IF (IS .EQ. ISMAX) PUNCH 9084
      PRINT 9090          $ RETURN          $ END

```

SUBROUTINE GRAFFIT

```

9010 FORMAT ( * SCALAR* I2, 24X, 1H0, 10(1S,1H0))
9020 FORMAT (33X, 1H0, 10(1S,1H0))
9030 FORMAT (1X *FOIL NO. SEQ. CARD N.ACT. *
      X      2H.., 20(5H .) 1H.)
9031 FORMAT (32X, 2H.., 20(5H .) 1H.)
9040 FORMAT (1X,A4, 27X, 1H., 101X, 1H.)
9045 FORMAT (1H1, 126X, *PAGE* I4 )
9050 FORMAT (5X,2X13,1X, 15, 2XA5, 2PF8.3,CP,2H .,101A1, 1H.)
9051 FORMAT (5X,2X13,1X, 15, 2XA5, 2PF8.3,CP,2H .,A1, 100X,1H.)
9052 FORMAT (5X,2X13,1X, 15, 2XA5, 2PF8.3,CP,2H .,100X, A1,1H.)

COMMON/A/BKW, CMULT, COEFF(1C), CONCR(6), CRATE(4C0), COM1
2 DT(6), DTB,DTC,DTD,EFC(6), ENTAU(6,4C0),F (201), COM2
3 FWT(6,2C1), IMAX, IS, ISCALR(6), ISMAX, ITCH(10), COM3
4 KOUNTS(201),KROOM, LAMEDA, LDEL, LMAX, LMIN, COM4
5 NAME(6,201), NCARD(6,4C0),NCARDS(6), NCCOUNTS, NCYCL(2C1), COM5
6 NEXTRA, NFOIL, NFOILS(6), NREM, RATE(6,4C0), COM6
7 RELSIG(6,4C0),T (6,4CC), T CORR, WGT(4C0), NTRAVS COM7
COMMON /PAGEBLK/ NPAGE, ITITLE(9)
COMMON/GRAF/ ACNCRM(40C)

TYPE INTEGER CHAR
DATA (BLANK = 1H )

PRINT 9010, ISCALR(IS), (I, I=1,10)
PRINT 9020
K = 0 $ NFLS = NFOILS(IS)
DO 20 NFOIL = 1,NFLS
CALL ELMER
CHECK WHETHER TO START ANOTHER PAGE
DO 11 J = 1,10C
JP = 55 + 58*(J-1)
IF (( K + NFOIL ) - JP ) 211,111,11
11 CONTINUE
111 PRINT 9031 $ NPAGE = NPAGE + 1
PRINT 9045, NPAGE $ PRINT 9010, ISCALR(IS), (I, I=1,10)
PRINT 9031
211 CONTINUE
PRINT 9040, NAME(IS,NFOIL)
DO 20 L=LMIN,LMAX,LDEL
K = K+1
CHECK WHETHER TO START ANOTHER PAGE
DO 12 J = 1,10C
JP = 55 + 58*(J-1)
IF (( K + NFOIL ) - JP ) 212,112,12
12 CONTINUE
112 PRINT 9031 $ NPAGE = NPAGE + 1
PRINT 9045, NPAGE $ PRINT 9010, ISCALR(IS), (I, I=1,10)
PRINT 9031
212 CONTINUE

C FOR LOW VALUES OF ACNORM, MULT. BY 1C OR 10C, AND USE X OR G, RESPECTIVELY
CCNCRM = ACNORM(K) $ CHAR = 1H*
IF (CCNCRM.GT.0.C95) GO TO 13 $ CCNCRM = 1C.*CCNCRM
CHAR = 1HX
IF (CCNCRM.GT.0.C95) GO TO 13 $ CCNCRM = 1C.*CCNCRM
CHAR = 1HG

13 JACK = XINTF( 10C. * CCNCRM + C.4999999)
IF (JACK.GT.100) JACK = 1C0
IF (JACK.LT. 0 ) JACK = C
JACK2 = 100 - JACK
IF (JACK.GT.0) GO TO 14
PRINT 9051, K, L, NCARD(IS,L), ACNORM(K),CHAR $ GO TO 20
14 IF (JACK.EQ.100) GO TO 15
PRINT 9050, K, L, NCARD(IS,L), ACNORM(K), (BLANK, I=1,JACK),
X CHAR, (BLANK,I=1,JACK2)
GO TO 20
15 PRINT 9052, K, L, NCARD(IS,L), ACNORM(K),CHAR
20 CONTINUE
PRINT 9031 $ PRINT 9020, (I, I=1,10)
RETURN
END

```

APPENDIX C

FORTRAN Listing for COMBO

This appendix contains the FORTRAN listings for the COMBO main program and the subroutine NORMGRUP. The NEWPAGE subroutine, which is also used by NURF, is listed in Appendix B.

PROGRAM COMBO

RP367X

```

COMBINES DATA WHEN THE SAME FOILS HAVE BEEN COUNTED ON DIFFERENT SYSTEMS (E.G.
C   DIFFERENT SCALERS OR AT DIFFERENT TIMES), PRODUCING A WEIGHTED AVERAGE
C   FOR EACH FOIL. SUBROUTINE NORMGRUP PROVIDES INTERNORMALIZATION OF
C   FOILS COUNTED IN TWO OR MORE GROUPS, WHEN EACH GROUP CONTAINS SOME
C   FOILS WHICH WERE COUNTED WITH SOME OTHER GROUP.
C   INPUT DATA CONSISTS CHIEFLY OF WEIGHTED-AVERAGE CARDS AS PUNCHED BY
C   NURF. CARD ONE CONTAINS THE TITLE AND ITCH OPTIONS. CARD TWO CON-
C   TAINS FACTORS DETERMINING THE RELATIVE IMPORTANCE TO BE ATTACHED TO THE
C   DATA FROM EACH SCALER USED.
C   IF ITCH2 IS POSITIVE, OUTPUT WEIGHT CARDS WILL BE PRODUCED IN THE
C   FORMAT COMPATIBLE WITH RP-202 AND NURF.

C....MAXIMUM NUMBERS -- FOILS IN ANY ONE GROUP .....100
C                      GROUPS OF FOILS.....20
C                      COMMON FOILS IN ANY ONE GROUP.....50
C                      SCALERS .....6

9001 FORMAT ( 9A8, 5X 3I1)
9002 FORMAT (1H010X48HPUNCHED WEIGHT CARDS HAVE BEEN REQUESTED (ITCH2))
9003 FORMAT (1H010X48HINTERMEDIATE PRINTOUT HAS BEEN REQUESTED (ITCH1))
9004 FORMAT (1H0 10X *RATIOS DIFFERING FROM MEAN BY *I1* SIGMA (ITCH3)
XOR MORE HAVE * /
X11X *BEEN REJECTED IN CALCULATING SCALER NORMALIZING FACTORS.*)
9005 FORMAT (2X)
9006 FORMAT (1H0 10X*FOIL NORMALIZATION CALCULATIONS (PROGRAM COMBO)*/)
9044 FORMAT (10A8)
9050 FORMAT (8X 6(1XI1,F8.C))
1051 FORMAT (1H02X *DO 52 LOOP...JSCALR=*I1*, KSCALR=*I1*, IJ(JSCALR)=*
X11*, IK(KSCALR)=*I1*, SCALWT(KSCALR)=*F10.4*, SCALW (JSCALR)=*F10.
X4*.)
9055 FORMAT (8X 12)
9056 FORMAT (1H0 2X *SCALER *I1* ON SCALWT CARD MATCHED TO SCALER *I1
X * IN DATA DECK (GROUP * I2 *). EXECUTION TERMINATED. * )
9080 FORMAT (A4,5X E15.7)

```

```

9201 FORMAT(1X*CARDS FOR SCALERS *I1* AND *I1*, GROUP *I2*, DO NOT MATC
X. EXECUTION TERMINATED.*)
9221 FORMAT (1H0 2X*NUMBER OF WTAVE DECKS INCONSISTENT WITH SCALWT CARD
X, FOR GROUP *I2 * . EXECUTION TERMINATED. * )
9225 FORMAT (1H01X5HFOIL A4* IN SCALER *I1* (GROUP *I2*) MATCHED TO FOI
XL *A4* IN SCALER *I1*. EXECUTION TERMINATED.*)
9270 FORMAT (1H0 1X *INPUT DATA FOR FOIL GROUP NO.*I2*..... 10A8)
9271 FORMAT (1H010X *WEIGHTING FACTORS - SCALER* I3, F10.3, /
X
( 31X 6HSCALER I3, F10.3 ))
9272 FORMAT (1H0 7X 8HSEQUENCE 4X 9HFOIL NAME 7X6HSCALERI2,10X12,4(13X
X I2))
9273 FORMAT ( 5X I3, 10X A4, 5X 6(E15.6))
9280 FORMAT (1H0/ 1X *OUTPUT DATA FOR FOIL GROUP NUMBER * I2 *.....
X 10A8)
9335 FORMAT (1H0 / 10X *ALL RATIOS FOR (SCALER *I1*/SCALER * I1*) ARE W
XITHIN * I1 * SIGMA OF THE AVERAGE RATIO, WHICH * /
X 13X *IS* F9.4* PLUS OR MINUS*
X F9.4 * PERCENT (STD. DEV.). * / 13X *RMS DEVIATION (SIGMA
X) IN INDIVIDUAL RATIOS IS * F9.4 * PERCENT.*)
9370 FORMAT ( /10X *THE RATIOS MARKED R DIFFERING FROM THE AVERAGE
X RATIO * F 9.4 * BY MORE THAN * I1 * SIGMA HAVE * / 12X
X *BEEN FOUND FOR (SCALER *I1 */SCALER * I1 *). * / 12X
X *THE PROGRAM CYCLED * I2 * TIMES THRU THE DATA (LIMIT = 20), AN
XD THE CHANGE IN SIGMA ON THE LAST* / 12X
X *CYCLE WAS *E10.3 * PERCENT OF THE AVERAGE RATIO. * /
X12X *RELATIVE STD. DEV. IN AV. RATIO, AFTER REJECTING THESE RATI
XOS, IS * F9.4 * PERCENT. * /
X12X *RELATIVE STD. DEV. IN REMAINING INDIVIDUAL RATIOS IS * F9.4
X* PERCENT.* / 12X *THERE ARE * I3 * UNREJECTED RATIOS.* )
9372 FORMAT(/16X 8HSEQUENCE 3X 9HFOIL NAME 5X 5HRATIO 5X12HPERCENT DEV
X. 5X 10HDEV./SIGMA / )
9373 FORMAT (19X I3, 8X A4, 3X E10.3, 3X F10.4, 8X F7.2)
9374 FORMAT ( 1H+ 35X 1HR )
9450 FORMAT (1H0 10X *GROUP * I2 *..... 10A8)
9460 FORMAT (1H010X*WEIGHTED AVERAGE ACTIVATION FOR THE FOILS IN GROUP*
X I2, *, USING DATA FROM *I1* SCALERS.* / )
9461 FORMAT (16X 8HSEQUENCE, 4X 9HFOIL NAME, 4X *WEIGHTED AVERAGE* / )
9465 FORMAT (18X I3, 10X A4, 6X E15.6)
9466 FORMAT (1H0 20X*SCALER WT. FACTORS TOTAL * F10.4 )
9470 FORMAT (1H0* TOO MANY GROUPS OF FOILS. ONLY 20 PROCESSED.*)

```

```

TYPE INTEGER TITLE, GTITLE
COMMON /PAGEBLK/ NPAGE, TITLE
COMMON /A/ NAME, WTDV, NGROUPS, NFOILS, ITCH1, ITCH2
DIMENSION NAMB (100,6), WTAVE (100,6), NFOILS (20), I (6),
X TITLE (9), NAME (500,20), WTDV(500,20), SCALW (6), IS(6),
X SIGMA(20), RATIO(100), NOBAD (100), SCALWT(6), DEV(100),
X IJ (6), IK(6), DEVSIG(100), GTITLE(10),RELDEV(100)

```

```

READ 9001, TITLE, ITCH1, ITCH2, ITCH3
2 NPAGE = 0
CALL NEWPAGE
PRINT 9006
PRINT 9005
IF (ITCH2.GT.0) PRINT 9002
11 IF (ITCH1.GT.0) PRINT 9003
IF (ITCH3.EQ.0) ITCH3 = 3.
PRINT 9004, ITCH3
PRINT 9005
29 NGROUP = 0

```

```

30 NGROUP = NGROUP + 1 $ IF (NGROUP .GT. 20) GO TO 474
   NFOILSA = 0
   DO 40 J = 1,6
40 SCALWT(J) = 0
   DO 41 NO = 1,100 $ DO 41 J = 1,6
41 NAMB(NO,J) = 4H
   READ 9044, GTITLE
   READ 9050, (IJ(JSCALR), SCALW (JSCALR), JSCALR = 1,6)
   KSCALR = 0
   DO 52 JSCALR = 1,6
   IF(SCALW (JSCALR).EQ.0) GO TO 49
   KSCALR = KSCALR + 1
   IK(KSCALR) = IJ(JSCALR)
   SCALWT(KSCALR) = SCALW (JSCALR)
C LAST 6 STATEMENTS HAVE CORRELATED SCALER NO. (IJ) AND SCALER SEQUENCE (NSCALR)
C DIAGNOSTIC OUTPUT.
49 IF (ITCH1.GT.1) 51,52
51 PRINT 1051, JSCALR, KSCALR, IJ(JSCALR), IK(KSCALR), SCALWT(KSCALR)
   X, SCALW (JSCALR)
C
52 CONTINUE
   NSCALRS = KSCALR
   NSCALR = 0
54 NSCALR = NSCALR + 1
55 READ 9055, I(NSCALR)
   IF (I(NSCALR).NE.IK(NSCALR)) 56,79
56 PRINT 9056, IK(NSCALR), I(NSCALR), NGROUP
   STGP
79 NO = 0
80 READ 9080, NAMA, WTAVA
   IF (NAMA.EQ.4HSCAL) GO TO 200
   IF (NAMA.EQ.4HNEXT) GO TO 200
   IF (NAMA.EQ.4HLAST) GO TO 200
   NO = NO + 1
   NAMB(NO,NSCALR) = NAMA
   WTAVE(NO,NSCALR) = WTAVA
   GO TO 80
C....REACH 200 AFTER DATA FROM EACH SCALER ARE READ.
200 IF (NFOILSA.EQ.C) GO TO 210
CHECK WHETHER NO. OF CARDS IN DECK IS SAME AS FOR PRECEDING SCALER.
   IF (NFOILSA.NE.NO) 201,220
201 PRINT 9201, I(NSCALR), I(NSCALR - 1), NGROUP
   STGP
210 NFOILSA = NO
COMPLETES INPUT FOR CURRENT SCALER. ON TO NEXT ONE.
C
220 IF (NAMA.EQ.4HSCAL) GO TO 54
   NFOILS(NGROUP) = NFOILSA
   IF (NSCALRS.NE. NSCALR) 221,222
221 PRINT 9221, NGROUP
   STGP
222 DO 260 NO = 1,NFOILSA
   DO 240 NSCALR = 2,NSCALRS
   IF (NAMB(NO,NSCALR).EQ.NAMB(NO,NSCALR-1)) GO TO 240
   NSM1 = NSCALR - 1
   PRINT 9225, NAMB(NO, NSM1 ), NSM1 ,NGROUP,NAMB(NO,NSCALR),NSCA
   X LR
   STGP
240 CONTINUE
260 NAME(NO,NGROUP) = NAMB(NO, 1)

```

```

C  FOIL NAMES FOR CURRENT GROUP ALL MATCH, AND HAVE BEEN STORED.
C
C  PRINTOUT OF INPUT DATA.
    PRINT 9270, NGROUP, GTITLE
    PRINT 9271, (I(NSCALR), SCALWT(NSCALR), NSCALR = 1, NSCALRS)
    PRINT 9272, (I(NSCALR), NSCALR = 1, NSCALRS)
    PRINT 9005
    DO 273 NO = 1, NFOILSA
273  PRINT 9273, NO, NAME(NO, NGROUP), (WTAVE(NO, NSCALR), NSCALR = 1,
    X      NSCALRS)
C
C  GET WEIGHTED AVERAGES FOR GROUP NGROUP.
    CALL NEWPAGE
    PRINT 9280, NGROUP, GTITLE
    DO 390 NSCALR = 2, NSCALRS
    IF (NSCALR .EQ. 2) GO TO 288
C.....IF NO. OF FOILS .GT. 15, AND IF NSCALR IS EVEN, CALL NEW PAGE.
    IF (NFOILSA .GT. 15) GO TO 285
280  IF (( 2*(NSCALR/2) - NSCALR) .NE. 0) GO TO 288
285  CALL NEWPAGE
288  NCYCL = 0
    DO 290 NBAD = 1, 100
290  NOBAD(NBAD) = 0
291  AVRAT = 0
    NGOOD = 0      $ NBAD = 1      $ NCYCL = NCYCL + 1
    DO 300 NO = 1, NFOILSA
    RATIO (NO) = WTAVE (NO, 1) / WTAVE (NO, NSCALR)
    IF (NOBAD(NBAD).EQ.NO) 292, 294
292  NBAD = NBAD + 1
    GO TO 300
294  NGOOD = NGOOD + 1
    AVRAT = AVRAT + RATIO (NO)
300  CONTINUE
    AVRAT = AVRAT / NGOOD
C
C  CALCULATION OF STANDARD DEVIATION (SIGMA) IN AV. RATIO (AVRAT)
    NBAD = 1
    SUMDEV2 = 0
    DO 320 NO = 1, NFOILSA
    DEV(NO) = RATIO(NO) - AVRAT
    RELDEV(NO) = (DEV(NO) / AVRAT)*100.0
    IF (NOBAD(NBAD) .EQ. NO) 310, 312
310  NBAD = NBAD + 1      $ GO TO 320
312  SUMDEV2 = SUMDEV2 + DEV(NO)**2
320  CONTINUE
C.....SOGMA IS STANDARD DEVIATION IN INDIVIDUAL RATIOS.
    SOGMA = SQRTF ( SUMDEV2 / NGOOD )
    SOGREL = 100. * SOGMA / AVRAT
    IF (NGOOD .EQ. 1 ) 321, 322
321  SIGMA (NCYCL) = 0      $ GO TO 325
322  SIGMA (NCYCL) = SQRTF((SUMDEV2) / (NGOOD * (NGOOD - 1)))
C  SIGREL IS PERCENT STANDARD DEVIATION IN AVRAT.
325  SIGREL = (SIGMA (NCYCL) / AVRAT) * 100.0
C
C  LOCK FOR BAD RATIOS
    DO 327 NBAD = 1, 100
327  NOBAD(NBAD) = 0      $ NBAD = 0
    DO 330 NO = 1, NFOILSA
    IF (ABSF( DEV(NO)) .GT. ( SOGMA * ITCH3)) 328, 330
328  NBAD = NBAD + 1

```



```

      NOBAD(NBAD) = NO
330 CONTINUE
      IF(NBAD .GT. 0) GO TO 350
      PRINT 9335, I(1), I(NSCALR), ITCH3, AVRAT, SIGREL, SOGREL
C.....(SIGMA IN PRINTOUT MEANS THE VARIABLE *SOGMA*.)
      GO TO 375
350 IF (NCYCL.EQ.1) GO TO 291
C
CALCULATE RELATIVE CHANGE (DELSIG) IN SIGMA SINCE PRECEDING CYCLE. IF CHANGE IS
C LESS THAN 10**(-10) PERCENT OF AVRAT, MOVE ON TO NEXT SCALER.
      DELSIG = 100*( SIGMA(NCYCL-1) - SIGMA(NCYCL)) / AVRAT
      IF (ABS(DELSIG) .LT. (10.C**(-10))) GO TO 370
360 IF (NCYCL .LT. 20) GO TO 291
370 PRINT 9370, AVRAT, ITCH3, I(1), I(NSCALR), NCYCL, DELSIG, SIGREL,
      XSOGREL, NGOOD
375 PRINT 9372
      JJ = 1
      DO 373 NO = 1, NFGILSA
      DEVSIG(NO) = DEV(NO) / SOGMA
      IF (NO .NE. NOBAD (JJ)) GO TO 373
      PRINT 9374
      JJ = JJ + 1
373 PRINT 9373, NO, NAME(NO,NGROUP), RATIO(NO), RELDEV(NO), DEVSIG(NO)
COMPLETES CALCULATIONS FOR SCALER(I).
C
380 DO 385 NO = 1, NFOILSA
385 WTAVE(NO, NSCALR) = WTAVE(NO, NSCALR) * AVRAT
390 CONTINUE
400 DO 440 NO = 1, NFOILSA
      WTDV (NO, NGROUP) = 0
      SUMWT = 0
      DO 420 NSCALR = 1, NSCALRS
      WTDV (NO, NGROUP) = WTDV (NO, NGROUP) + WTAVE(NO, NSCALR)*SCALWT(N
      X SCALR)
420 SUMWT = SUMWT + SCALWT(NSCALR)
440 WTDV (NO, NGROUP) = WTDV (NO, NGROUP) / SUMWT
      CALL NEWPAGE
      PRINT 9450, NGROUP, GTITLE
      PRINT 9460, NGROUP, NSCALRS
      PRINT 9461
      PRINT 9465, (NO, NAME(NO, NGROUP), WTDV (NO, NGROUP), NO=1, NFOILSA)
      PRINT 9466, SUMWT
      IF (NAMA.EQ.4HNEXT) 470, 475
470 CALL NEWPAGE
      GO TO 30
474 NGROUP = NGROUP - 1 $ PRINT 9470
475 NGROUPS = NGROUP
      CALL NORMGRUP
C
CALL IN NEW PROBLEM. *END-OF-FILE* CARD ENDS CALCULATIONS.
      READ 9001, TITLE, ITCH1, ITCH2, ITCH3
      IF (EOF, 60) 499, 2
499 CONTINUE
      END

```

SUBROUTINE NORMGRP

C.....FOR USE WHEN SEVERAL GROUPS OF FOILS HAVE BEEN COUNTED SEPARATELY,
 C AND MUST BE INTERNORMALIZED BY MEANS OF FOILS COMMON TO MORE THAN
 C ONE GROUP.
 C.....NOTE THAT THE GROUPS MUST BE ORDERED SO THAT EACH GROUP AFTER THE
 C FIRST CONTAINS SOME FOILS WHICH WERE ALSO COUNTED IN A PRECEDING GROUP.

C....MAXIMUM NUMBERS -- FOILS IN ANY ONE GROUP100
 C GROUPS OF FOILS.....20
 C COMMON FOILS IN ANY ONE GROUP.....50

C....INTERMEDIATE RESULTS ARE PRINTED IF ITCH1 IS POSITIVE.
 C....OUTPUT WEIGHT CARDS ARE PUNCHED IF ITCH2 IS POSITIVE.

```

1120 FORMAT (1H0 *NO COMMON FOILS FOUND IN GROUP * I3 *....EXECUTION T
      XERMINATED.*)
1180 FORMAT(1H0//5X*LISTING OF FOILS COMMON TO GROUP *I2* AND PRECEDING
      X GROUPS * / /
      X 4 ( 2X4HNAME,5X5H RATIO, 4X 8HDEV.FROM 6X)/
      X 4 (18X              12HMEAN,PERCENT 4X)/)
1200 FORMAT (4(2XA4,E12.3,F5.3, 7X ))
1220 FORMAT(1H035X* AVERAGE RATIO FOR MULTIPLYING GROUP*I3,4X*IS*E12.3,
      X1H./ 36X * RELATIVE STANDARD DEVIATION IN THIS RATIO IS* F8.3* PER
      XCENT.*)
1422 FORMAT (1H0,5X *LISTING OF NORMALIZED ACTIVITIES THRU GROUP *I2//
      X 10X 8HSEQUENCE,4X9HFOIL NAME,4X *NORMALIZED ACTIVITY SO FAR * / )
1423 FORMAT (11X I4, 9X A4, 2X E23.6)
1430 FORMAT (1H0,5X *COMPLETE LIST OF NORMALIZED ACTIVITIES FOR* I3
      X * GROUPS OF FOILS *)
1431 FORMAT (1H0 10X 8HSEQUENCE,4X9HFOIL NAME,11X*NORMALIZED ACTIVITY*
      X/)
1432 FORMAT ( 12X I4, 9X A4, 9X F10.4, 3X E14.5)
1446 FORMAT (1H02X *CARDS HAVE BEEN PUNCHED FOR THE ABOVE.*)
1447 FORMAT (1CA8)
1448 FORMAT (*INTERCAL.FOR * 8A8,A3)
1450 FORMAT (6HWEIGHT, A4, F10.4)
1460 FORMAT (1H0// * END OF CALCULATIONS FOR *1CA8)

```

```

      TYPE INTEGER TITLE
      COMMON /PAGEBLK/ NPAGE, TITLE
      COMMON /A/ NAME,WTDAV,NGROUPS,NFOILS,ITCH1,          ITCH2
      DIMENSION NFOILS(20), FWT(100,20), NAME(500,20),
      X INDEXA (50), NAMC(50),
      XCOMRAT(50), TITLE( 9),FWT2(500),NAME2(500),WTDAV2(500),INDEXB(50),
      X DEV(50),RELDEV(50)

```

C....INITIALIZE FWT(NO) TO UNITY.... FWT(NO) IS TO BE WTG. FACTOR FOR AVERAG-
 C ING COMMON FOILS.

```

      DO 20 NO = 1,100      $ DO 20 NP = 1,20

```

```

      20 FWT(NO,NP) = 1.0

```

```

      NUMGROUP = 1

```

```

      IF (NGROUPS.EQ.1) GO TO 425

```

```

      IF (ITCH1) 25,25,30

```

```

      25 CALL NEWPAGE

```

```

      30 DO 400 NGROUP = 2,NGROUPS

```

```

      DO 35 KOM = 1,50

```

```

      INDEXA (KOM) = 0

```

```

      35 INDEXB (KOM) = 0

```

```

      IF(ITCH1) 40,40,420

```

```

      40 KOM = 0

```

C.... KOM IS SEQUENCE NUMBER FOR ASSIGNMENT TO COMMON FOILS.

```

      NF1 = NFOILS(1)

```

```

      DO 100 NO = 1,NF1

```

C....SEARCH FOR FOILS COMMON TO GROUP 1 AND GROUP NGROUP.

```

      NFNGRP = NFOILS(NGROUP)

```

```

      DO 60 MO = 1,NFNGRP

```

```

      IF (NAME(NO,1) - NAME(MO,NGROUP))60,8C,60

```

```

      60 CONTINUE

```

```

GO TO 100
C....IF COMMON FOIL IS FOUND, STEP KOM AND KEEP TRACK OF COMMON SEQUENCE NUMBERS
      80 KOM = KOM + 1
C....INDEXA AND INDEXB STORE SEQUENCE NUMBERS FOR COMMON FOILS.
      INDEXA(KOM) = NO
      INDEXB(KOM) = MO
C....COMRAT IS THE RATIO OF THE WDAV FROM THE TWO GROUPS.
      COMRAT (KOM) = WTCV(NO,1) / WDAV(MO,NGROUP)
      100 CONTINUE

C....HAVE ACQUIRED THE RATIOS FOR ALL THE COMMON FOILS...NOW GET THE AV.RATIO.
      SUMRAT = 0
      KOMS = KOM
      IF(KOMS)120,120,140
      120 PRINT 1120, NGROUP
      RETURN
      140 GO 160 KOM = 1,KOMS
      160 SUMRAT = SUMRAT + COMRAT(KOM)
      AVRAT = SUMRAT / KOMS

C....CALCULATION OF STANDARD DEV. (SIGMA) IN AVRAT.
      SUMDEV2 = 0
      GO 162 KOM = 1,KOMS
      DEV (KOM) = (COMRAT(KOM) - AVRAT)
      SUMDEV2 = SUMDEV2 + DEV (KOM)**2
      162 RELDEV(KOM) = (DEV(KOM) / AVRAT)*100.0
      IF (KOMS.EQ.1)163,164
      163 SIGMA = 0 $ GO TO 165
      164 SIGMA = SQRTF((SUMDEV2) / (KOMS * (KOMS - 1)))
C....SIGREL IS PERCENT STANDARD DEVIATION IN MEAN (AVRAT).
      165 SIGREL = (SIGMA / AVRAT)*100.0

C... PRINTOUT OF COMMON FOILS AND RATIOS.
      180 PRINT 1180, NGROUP
      GO 190 KOM = 1,KOMS
      190 NAMC(KOM) = NAME(INDEXA(KOM))
      200 PRINT 1200, (NAMC(KOM), COMRAT(KOM),RELDEV(KOM),KOM=1,KOMS
      X)
      220 PRINT 1220, NGROUP, AVRAT, SIGREL

C....REPLACE ORIGINAL WDAV IN GROUP NGROUP WITH NORMALIZED VALUES.
      NFNGRP = NFOILS(NGROUP)
      DO 240 MO = 1,NFNGRP
      240 WTCV(MO,NGROUP) = WDAV(MO,NGROUP) * AVRAT
C....HAVE NOW NORMALIZED GROUP(NGROUP) TO GROUP 1... NEXT COMBINE INTO 1 GROUP.
C PUT COMMON FOILS AT FRONT OF NEW GROUP... LO IS SEQUENCE NO. FOR NEW GROUP.
      LO = 0
      DO 260 KOM = 1,KOMS
      LO = LO + 1
      INCA= INDEXA(KOM)
      INCB = INDEXB(KOM)
      NAME2 (LO) = NAME(INCA,1)
      WDAV2 (LO) = (WDAV(INCA,1)* FWT(INCA,1) + WDAV(INCB,NGROUP))
      X / (FWT(INCA,1) + 1)
C....STEP FWT TO BE READY FOR NEXT TIME...FWT PREVENTS OVERWEIGHTING COMMON
C FOILS IN THE LATER GROUPS.
      FWT(INCA,1)=FWT(INCA,1)+1.0
      FWT2(LO) = FWT(INCA,1)
      260 CONTINUE

C....NEW GROUP HAS BEEN STARTED WITH THE COMMON FOILS...
C NOW ADD REMAINING FOILS FROM GROUP 1 (SKIPPING OVER COMMON FOILS).
      KOM = 1
      DO 320 NO = 1,NFOILS(1)
      280 IF (NO - INDEXA(KOM))290,300,290
      290 LO = LO + 1
      NAME2 (LO) = NAME(NO,1)
      FWT2 (LO) = FWT (NO,1)
      WDAV2(LO) = WDAV(NO,1)

```

```

      GO TO 320
300 KOM = KOM + 1
320 CONTINUE
C....NEXT ADD NON-COMMON FOILS FROM GROUP 2.
      NFNGRP = NFOILS(NGROUP)
      DO360 MO = 1,NFNGRP
      DO340 KOM = 1,KOMS
      IF (MO - INDEXB(KOM))340,360,340
340 CONTINUE
      LG = LG + 1
      NAME2(LG) = NAME(MO,NGROUP)
      WTCV2(LG) = WTCV(MO,NGROUP)
C....NUMGROUP STORES NGROUP FOR USE IN SUBSEQUENT DIAGNOSTIC OUTPUT (IF ITCH1=1)
      NUMGRUP = NGROUP
360 CONTINUE

C....MAKE SOME SUBSTITUTIONS BEFORE GOING TO NEXT GROUP, SO THAT NEW GROUP
C      BECCMES NEW GROUP 1.
      NFOILS(1) = LG
      DO 380 LG = 1,NFOILS(1)
      NAME(LG,1) = NAME2(LG)
      WTCV(LG,1) = WTCV2(LG)
      FWT(LG,1)=FWT2(LG)
380 CONTINUE
C....COMPLETES DO LOOP FOR GROUP OF FOILS.

C      NOW PROCEED TO NORMALIZE NEXT GROUP TO THE NEW GROUP 1.
400 CONTINUE
      IF(ITCH1) 425,425,419
419 ITCH1 = 0
C....INTERMEDIATE OUTPUT. PROGRAM GETS HERE FROM STATEMENT 30 + 1, IF ITCH1=1.
420 CALL NEWPAGE
422 PRINT 1422, NUMGROUP
      NF1 = NFOILS(1)
      DO 423 NO = 1,NF1
423 PRINT 1423, NO, NAME(NO,1), WTCV(NO,1)
      IF(ITCH1) 425,425,40
C....(PROGRAM IS NOW THRU WITH ITCH1, WHICH WAS SET TO ZERO ON INPUT OR BY 419.)

C....NORMALIZE ALL WTCV TO (A) STRONGEST FCIL (B) WEAKEST FCIL
425 REF = WTCV(1,1)
      NF1 = NFOILS(1)
      DO 428 NO = 2,NF1
428 REF = MAX1F (REF, WTCV(NO,1))
      DO 429 NO = 1, NF1
429 WTCV2(NO) = WTCV(NO,1) / REF
      REF = WTCV(1,1)
      DO 426 NO = 2,NF1
426 REF = MIN1F (REF, WTCV(NO,1))
      DO 427 NO = 1,NF1
427 WTCV(NO,1)= WTCV(NO,1) / REF

C....PRINT,AND PUNCH (IF CALLED FOR),FINAL RESULTS.
      CALL NEWPAGE
430 PRINT 1430, NGROUPS
431 PRINT 1431
      NF1 = NFOILS(1)
      DO 432NO=1,NF1
432 PRINT 1432, NO, NAME(NO,1), WTCV(NO,1), WTCV2(NO)

C....PUNCH IF ITCH2 IS POSITIVE.
445 IF(ITCH2) 460,460,446
446 PRINT 1446
447 PUNCH 1447, TITLE
448 PUNCH 1448, TITLE
450 PUNCH 1450, (NAME(NO,1), WTCV(NO,1), NO = 1,NFOILS(1))
460 PRINT 1460, TITLE
      RETURN
      END

```

APPENDIX D

FORTRAN Listing for TWOSORCE

```

PROGRAM TWOSORCE
C      CALCULATES DEAD TIME, USING PAIRED-SOURCE METHOD

C      IF ITCH1 IS NON-ZERO, THE PROGRAM CALCULATES BOTH FOR THE EXACT NON-PARA-
C      LYZABLE CASE, AND ALSO USES THE INPUT COEFFICIENTS (CARD 3) FOR THE FOURTH-
C      ORDER SERIES CASE.
C      IF ITCH1 IS NON-ZERO AND THE INPUT COEFFICIENT B IS OMITTED, THE PROGRAM
C      SETS B, C, AND D TO 3/2, 8/3, AND 125/24, RESPECTIVELY.
C      IF ITCH1 IS 3, DIAGNOSTIC PRINTOUT IS PROVIDED.
C      IF ITCH1 IS 4, ALL OF THE ABOVE IS DONE, PLUS A SERIES CALCULATION USING
C      1, 1, AND 1 FOR THE COEFFICIENTS.
C      IF ITCH1 IS 2, ALL OF THE ABOVE IS DONE EXCEPT THE DIAGNOSTIC PRINTOUT.
C
      TYPE INTEGER  TITLE, CROOM
      TYPE REAL  LAMBCA
      DIMENSION CLOCK1(50), COUNT1(50,6),  CTIME (50), CTIME1(50),
      X  TAU(50),CLOCK2(50), COUNT2(50,6),  BKG ( 6), CTIME2(50),
      X  NGT (6), CLOCK3(50), COUNT3(50,6),  TITLE (10), CTIME3(50)
      COMMON /PAGEBLK/ NPAGE, TITLE
C
      8000 FORMAT (10A8)
      8001 FORMAT (30X E20.6, / A5, 4X11,4X611, 9X11, 3F10.0)
      8002 FORMAT (16X, F8.1,1X, 4( 1X F8.0) )
      8003 FORMAT ( 6X, F8.1,2X, F8.1,1X, 4( 1X F8.0) )
      8004 FORMAT (13X, F6.1, 6(F8.0))
      8005 FORMAT (5X, F8.1, F6.1, 6(F8.0))
      9007 FORMAT (1X *DOUBLE SOURCE CALCULATIONS* )
      9008 FORMAT (1H0 5X *BACKGROUND... TIME AND SCALER DATA* 7 ( 3X F8.1 ))
      9009 FORMAT (5X E14.6, 10X I3, 15X A5, 16X I2, 20X , 6I1 )
      9010 FORMAT (1H0 5X *DECAY CONSTANT NO. OF DETERMINATIONS COUNTING RO
      XGM NO. OF HIGHEST SCALER  UNUSED SCALERS(1) *)
      9011 FORMAT (1H0 1X 6HSCALER 1X I1, 1H, 2XA5 * COUNTING ROOM * )
      9012 FORMAT (1H0 10X,8X6HCLOCK3, 9X6HCOUNT1, 4X6HRAE A, 9X6HCOUNT2,
      X 4X6HRAE B, 9X6HCOUNT3, 4X6HRAE C, 13X SHDEAD TIME / )
      9013 FORMAT (10X, 6XF9.1, 3(7XF8.0, 2XF8.1), 6XE11.4, 2XF8.3)
      9014 FORMAT (1H0/ 3X*AVERAGE DEAD TIME FOR SCALER *I1*  =*F9.4* MICRO
      XSECONDS* / 3X*STANDARD DEVIATION IN AVERAGE  =*F9.4 * MICROSE
      XCONDS* / 3X*RMS DEV. IN INDIVIDUAL DEAD TIMES =*F9.4 * MICROSE
      XCONDS * )
      9015 FORMAT ( 98X *DEAD TIME USING INPUT COEFFICIENTS*
      X / 4X6HCLOCK3, 5X6HCOUNT1, 4X6HRAE A, 5X6HCOUNT2,
      X 4X6HRAE B, 5X6HCOUNT3, 4X6HRAE C, 10X9HDEAD TIME,
      X 6X6HCYCLES, 3X9HLAST.TTAU, 6X3HTAU, 3X 4HNTAU / )
      9016 FORMAT (1X F9.1, 3(3XF8.0,1XF9.1), 3XE11.4,1XF8.3, 4XI2, 3XE11.4,
      X 1X 6PF8.3,OPF7.3)
      9017 FORMAT(/1H*AVERAGE USING THE INPUT COEFFICIENTS* 3(F9.4,1H,)
      X * FOR B, C, AND D *)
      9018 FORMAT(101XI2, E14.4, 6PF9.3, OPF7.3)
      9019 FORMAT (1X *OUTPUT WITH B = C = D = 1*)
      9020 FORMAT (1H0, 5E15.4, / )

      NPAGE = 0
      READ 8000, TITLE
      2 READ 8001, LAMBCA, CROOM, NOS, (NOT(I),I=1,6), ITCH1, BIN,CIN,DIN
      J = 0
      IF (BIN .EQ. 0.0) CIN = 8./3.
      IF (BIN .EQ. 0.0) DIN = 125. / 24.
      IF (BIN .EQ. 0.0) BIN = 3./2.
      IF(CROOM .EQ. 5HNORTH) GO TO 10
      IF(NOS .EQ. 0) NOS = 4 $ D = 1.0

```

SOUTH

```

      READ 8002, BTIME, (BKG(I), I=1, NOS)
3  J = J+1
      READ 8003, CLOCK1(J), CTIME1(J), (COUNT1(J,I), I=1, NOS)
      IF (CTIME1(J) .EQ. 0.0) GO TO 2C
      READ 8003, CLOCK2(J), CTIME2(J), (COUNT2(J,I), I=1, NOS)
      READ 8003, CLOCK3(J), CTIME3(J), (COUNT3(J,I), I=1, NOS)
      GO TO 3
10 IF (NOS .EQ. 0) NOS = 6 $ D = 1.0
      READ 8004, BTIME, (BKG(I), I=1, NOS)
11 J = J+1
12 READ 8005, CLOCK1(J), CTIME1(J), (COUNT1(J,I), I=1, NOS)
      IF (CTIME1(J) .EQ. 0.0) GO TO 2C
      READ 8005, CLOCK2(J), CTIME2(J), (COUNT2(J,I), I=1, NOS)
      IF (CTIME2(J) .EQ. 0.0) GO TO 32C
      READ 8005, CLOCK3(J), CTIME3(J), (COUNT3(J,I), I=1, NOS)
      IF (CTIME3(J) .EQ. 0.0) GO TO 32C
      GO TO 11
C...JTOT IS TOTAL NO. OF DOUBLE-SOURCE DETERMINATIONS.
20 JTOT = J - 1
      CALL NEWPAGE
      PRINT 9007
      IF (CROOM .EQ. 5H ) CROOM = 5HSOUTH
      PRINT 9010
      PRINT 9009, LAMBDA, JTOT, CROOM, NOS, (NOT(I), I=1, 6)
      PRINT 9008, BTIME, (BKG(I), I = 1, NOS)
C
      KOUNTER = 0
25 NSCALR = 0
      DO 300 I = 1, NOS
          KOUNT = 0
          IF (NOT(I) .NE. C) GO TO 30C
          NSCALR = NSCALR + 1
          IF (NSCALR .GT. 1) CALL NEWPAGE
          RATEBK = (BKG(I) / BTIME) * D
          PRINT 9011, I , CROOM
          IF (ITCH1 .GE. 1) GO TO 30
          PRINT 9012 $ GO TO 40
30 PRINT 9015
40 AVTIME = 0.0
      AVTIM2 = 0.0
      DO 250 J = 1, JTOT
          COUNTA = D * COUNT1(J,I) * EXPF(-LAMBDA * (CLOCK3(J) - CLOCK1(J)))
          COUNTB = D * COUNT2(J,I) * EXPF(-LAMBDA * (CLOCK3(J) - CLOCK2(J)))
          RATEA = COUNTA / CTIME1(J)
          RATEB = COUNTB / CTIME2(J)
          RATEC = (COUNT3(J,I) / CTIME3(J)) * D
          RATECK = 1.0 / RATEC
          DTIME(J) = RATEC - SQRTF ((( 1.0 / RATEA) - RATECK) *
X          ((1.0 / RATEB) - RATECK) + RATEBK / (RATEA * RATEB *
X          RATEC ))
          AVTIME = AVTIME + DTIME(J)
          UDTIME = DTIME(J) * 10.0**6
          IF (ITCH1 .GE. 1) GO TO 11C
          PRINT 9013, CLOCK3(J), COUNT1(J,I), RATEA, COUNT2(J,I), RATEB,
X          COUNT3(J,I), RATEC, DTIME(J), UDTIME
      GO TO 250
C
CALCULATE DEACTIME USING INPUT COEFFICIENTS
110 TTAU = CTIME(J)
      JTRIES = 0
      DIF1 = RATEC - RATEA - RATEB + RATEBK
      DIF2 = (RATEC**2 - RATEA**2 - RATEB**2)
      DIF3 = (RATEC**3 - RATEA**3 - RATEB**3) * BIN
      DIF4 = (RATEC**4 - RATEA**4 - RATEB**4) * CIN
      DIF5 = (RATEC**5 - RATEA**5 - RATEB**5) * DIN
      IF (KOUNT .GT. 16) GO TO 120

```

```

      IF (ITCH1 .LT. 3) GO TO 120
      CF2 = DIF2 * TTAU
      CF3 = DIF3 * TTAU**2
      CF4 = DIF4 * TTAU**3
      CF5 = DIF5 * TTAU**4
      PRINT 9020, DIF1, DIF2, DIF3, DIF4, DIF5
120  CONTINUE
      FRAC = RATEA * TTAU
      IF (FRAC .LT. .10) GO TO 1120
      IF (FRAC .LT. .20) GO TO 1130
      IF ((FRAC .LT. .3) .OR. (DIN .LT. .2)) GO TO 1140
      TAU4 = - (DIF1 + DIF2 * TTAU + DIF3 * TTAU**2 + DIF4 * TTAU**3
X      ) / DIF5
      IF (TAU4 .GE. 0.) GO TO 121
      TAU(J) = 3. * 10. **(-6) $ GO TO 122
121  TAU(J) = SQRTF ( SQRTF (TAU4))
      GO TO 122
1120 TAU(J) = - (DIF1 + DIF3 * TTAU**2 + DIF4 * TTAU**3
X      + DIF5 * TTAU**4) / DIF2
      GO TO 122
1130 IF (BIN .LT. .1) GO TO 1120
      TAU2 = - (DIF1 + DIF2 * TTAU + DIF4 * TTAU**3
X      + DIF5 * TTAU**4) / DIF3
      IF (TAU2 .GE. 0.) GO TO 1131
      TAU(J) = 3. * 10. **(-6) $ GO TO 122
1131 TAU(J) = SQRTF (TAU2)
      GO TO 122
1140 IF (CIN .LT. .2) GO TO 1120
      TAU3 = - (DIF1 + DIF2 * TTAU + DIF3 * TTAU**2
X      + DIF5 * TTAU**4) / DIF4
      IF (TAU3 .GE. 0.) GO TO 1141
      TAU(J) = 3. * 10. **(-6) $ GO TO 122
1141 TAU(J) = CUBERTF (TAU3)
122  IF (ABSF (TAU(J) - TTAU) .LT. (10.**(-9))) GO TO 130
      IF (ITCH1 .LT. 3) GO TO 125
      IF (JTRIES .GT. 5) GO TO 125
      KOUNT = KOUNT + 1
      IF (KOUNT .GT. 15) GO TO 125
      PRINT 9018, JTRIES, TTAU, TAU(J), FRAC
125  JTRIES = JTRIES + 1
      IF (JTRIES .GT. 40) GO TO 130
      TTAU = (TAU(J) + 2. * TTAU) / 3.0
      GO TO 120
130  AVTIM2 = AVTIM2 + (TTAU + TAU(J)) / 2.0
      PRINT 9016, CLOCK3(J), COUNT1(J,I), RATEA, COUNT2(J,I), RATEB,
X      COUNT3(J,I), RATEC, DTIME(J), UDTIME, JTRIES, TTAU,
X      TAU(J), FRAC
250  CONTINUE
      AVTIME = AVTIME / JTOT
C
CALCULATE ERROR IN AVTIME (AVERAGE DEAD TIME).
      SUMDEV2 = 0
      DO 260 J = 1, JTOT
260  SUMDEV2 = SUMDEV2 + (DTIME(J) - AVTIME)**2
      IF (JTOT .LT. 2) GO TO 265
      STDEV = SQRTF (SUMDEV2 / (JTOT * (JTOT - 1)))
      SIGMA = SQRTF (SUMDEV2 / JTOT)
      GO TO 270
265  STDEV = 0 $ SIGMA = 0
270  USIGMA = SIGMA * (10.0**6) $ UVTIME = AVTIME * (10.0**6)
      USTDEV = STDEV * (10.0**6)
      PRINT 9014, I, UVTIME, USTDEV, USIGMA
      IF (ITCH1 .EQ. 0) GO TO 300
C
CALCULATE ERROR IN AVTIM2
      PRINT 9017, BIN, CIN, DIN

```

```

SUMDEV2 = 0
AVTIM2 = AVTIM2 / JTOT
DO 460 J = 1, JTOT
460 SUMDEV2 = SUMDEV2 + (TAU(J) - AVTIM2) ** 2
   IF(JTOT .LT. 2) GO TO 465
   STDEV = SQRTE (SUMDEV2 / (JTOT * (JTOT - 1)))
   SIGMA = SQRTE (SUMDEV2 / JTOT )
   GO TO 470
465 STDEV = 0   $ SIGMA = 0
470 USIGMA = SIGMA * (10.0**6)   $ UVTIME = AVTIM2 * (10.0**6)
   USTDEV = STDEV * (10.0**6)
   PRINT 9014,I,UVTIME,USTDEV, USIGMA
300 CONTINUE
   KOUNTER = KOUNTER + 1
   IF ((ITCH1 .NE. 2) .AND. (ITCH1 .NE. 4)) GO TO 320
   IF (KOUNTER .GT. 1) GO TO 320
   BIN=1.   $ CIN=1.   $ DIN=1.
   CALL NEWPAGE   $ PRINT 9019   $ GO TO 25.
320 READ 8000, TITLE
   IF (ECF,60) 350, 2
350 CONTINUE
   END

```


APPENDIX E

Tables

TABLE I. Description of NURF Control-deck Cards

Card No.	Columns	Format	Name of Parameter	Permissible Entries ^a	Use
1	1-72	9A8	ITITLE	[Any]	Title of problem.
2	5-12	E8.0	BKF		Background multiplier. Blank ^a = 1.0.
	13	I1	IMAX	0-6	Number of highest scaler used. Blank ^a = 6 for North Counting Room. Blank ^a = 4 for South Counting Room.
	15-18	A4	JRØØM	[Any]	Counting room indicator. NØLD: North Counting Room, old equipment. N(Col. 15): North Counting Room, 1967 equipment. [Anything else]: South Counting Room.
	21-30	E10.5	BKW	0.0-1.0	Background weighting factor K (Eq. 8); used when IDC(1) = 1 (see Section VI-D). Blank ^a = 0.5.
	31-40	E10.5	DTB		Coefficient B for deadtime corrections. If blank, ^a DTB = 1.5, DTC = 8.0/3.0, and DTD = 125.0/24.0.
	41-50	E10.5	DTC		Coefficient C for deadtime corrections. If DTB is blank, ^a DTC = 8.0/3.0.
	51-60	E10.5	DTD		Coefficient D for deadtime corrections. If DTB is blank, ^a DTD = 125.0/24.0.
	61-70	E10.5	CØNCR(1)		Convergence criterion for ITER 7. Blank ^a = 0.04.
	71-80	E10.5	CØNCR(2)		Convergence criterion for least-squares fit. Blank ^a = 0.004.
3	5-10	6I1	NS(1)	0-9	Nonzero ^a entry deletes processing for Scaler 1. (On p. 1 of the output, deleted scalers are indicated by zero efficiency.) Scaler: 1 2 3 4 5 6 Column: 5 6 7 8 9 10
	11-19	E9.0	TTHALF		Half-life or decay constant, in units indicated in Col. 20. For least-squares fit, this need only be approximate. If blank, ^a fission-product least-squares fit will be made.
	20	A1	MULT	{ S,M,H,D, Y, blank }	Units of half-life (sec, min, hr, day, yr). If blank, MULT = L, and TTHALF is assumed to be the decay constant λ , in reciprocal seconds.
	21-80	6E10.5	EFC(1)		Efficiency of Scaler 1. Blank ^a = 1.00. Scaler: 1 2 3 4 5 6 Columns: 21-30 ^a 31-40 ^a 41-50 ^a 51-60 ^a 61-70 ^a 71-80 ^a
4	11-13	I3	ITCH(1)	0,1,2	Collation indicator. If blank: ^a Data cards uncollated; foils restacked between counting cycles. If 1: Data cards collated. If 2: Data cards uncollated; foils counted without restacking.
	14-16	I3	ITCH(2)	0,2-10	No. of coefficients for least-squares fit. Blank ^a = 2 for decay-constant fit. Blank ^a = 4 for fission-product fit.
	17-19	I3	ITCH(3)	0-999	Divisor for determining the value of the DELF input for ITER 7. Blank ^a = 1,000.
	20-22	I3	ITCH(4)	0-300	Maximum number of ITER 7 cycles. Blank ^a = 50.
	23-25	I3	ITCH(5)	0-999	Maximum number of least-squares iterative cycles. Blank ^a = 10.
	26-28	I3	ITCH(6)	0-3	If ITCH(6) = 1 or 3, output cards are punched for use in the CØMBØ program. 2 or 3 punches cards for Program DEADTIME. ¹¹
	29-31	I3	ITCH(7)	0-3	If ITCH(7) = 1, 2, or 3, some diagnostic printout is produced.
	32-34	I3	ITCH(8)	0,1	If ITCH(8) = 1, the special gold-indium weight corrections are made.
	35-37	I3	ITCH(9)	0,1	If ITCH(9) = 1, the least-squares fitting process is replaced by a fixed-decay-constant calculation.
	38-40	I3	ITCH(10)	0,1	If ITCH(10) = 1, the accidental-coincidence corrections are made to the data for the coincidence scalers.
	41-45	E5.0	TTIRRAD		Irradiation time, in <u>minutes</u> . Blank ^a = 8.88.

TABLE I (Contd.)

Card No.	Columns	Format	Name of Parameter	Permissible Entries ^a	Use
4 (Contd.)	46-55	E10.5	TTSCRAM		Time in <u>minutes</u> from end of irradiation to when the clock reading TCLØCK occurred. Blank ^a = 88.88.
	56-65	E10.5	TCLØCK		Counter clock reading, in <u>seconds</u> , at some known clock time. Blank ^a = 0.0.
	66-70	E5.0	CDR		Cadmium ratio, for use if ITCH(8) = 1. Blank ^a = 2.0.
	71-75	E5.0	FWZ		Nominal foil weight, for use if ITCH(8) = 1.
	76-80	E5.0	HR		Value of H-function, for use if ITCH(8) = 1. Blank ^a = 0.43.
5	11-70	6E10.5	DT(1)		Resolving time for Scaler 1, in <u>microseconds</u> . Scaler: 1 2 3 4 5 6 Columns: 11-20 ^a 21-30 ^a 31-40 ^a 41-50 ^a 51-60 ^a 61-70 ^a
	71	11	INCA(1)	0-6	Scaler number for Channel A of coincidence experiment. Should not be blank ^a if ITCH(10) = 1.
	72	11	INCB(1)	0-6	Scaler number for Channel B of coincidence experiment. Should not be blank ^a if ITCH(10) = 1.
	73	11	MAC(1)	0-6	Scaler number for coincidence channel of coincidence experiment. Should not be blank ^a if ITCH(10) = 1.
	74	11	INCA(2)	0-6	Scaler number for Channel A of second coincidence equipment. May be equal to INCA(1), INCB(1), or MAC(1).
	75	11	INCB(2)	0-6	Scaler number for Channel B of second coincidence experiment. May be equal to INCA(1), INCB(1), or MAC(1).
	76	11	MAC(2)	0-6	Scaler number for coincidence channel of second coincidence experiment. May be equal to INCA(1) or INCB(1).
	77	11	INCA(3)	0-6	Scaler numbers for third coincidence experiment.
	78	11	INCB(3)	0-6	
	79	11	MAC(3)	0-6	

^aFor I, E, and F formats, zero and blank are equivalent.

TABLE II. Abridged Control-deck Description, for Routine Problems (NURF)

Card No.	Columns	Name of Parameter	Entry	Description
1	1-72	ITITLE		Title of problem.
2	13	IMAX	Blank	Number of highest scaler used. 6 for North Counting Room, 4 for South.
	15	JRØØM	N Blank S	North Counting Room (1967 equipment). South Counting Room.
3	5-10	NS(I)		Digit other than zero (or blank) deletes processing for corresponding scaler. For Scaler No. > IMAX (Col. 13, Card 2), no entry is needed. Scaler: 1 2 3 4 5 6 Column: 5 6 7 8 9 10
	11-19 ^a	TTHALF ^b		Numerical value of half-life. ^c For least-squares fit, this need only be approximate. If Column 20 is blank, this entry is interpreted as the decay constant, in reciprocal seconds. If Columns 11-19 are blank, a fission-product least-squares fit will be made.
	20	MULT ^b	S M H D Y Blank	Seconds Minutes Hours Days Years (See Columns 11-19.)
	21-80 ^a	EFC(I) ^b		Scaler efficiencies. Blank = 1.0. Normally left blank. Scaler: 1 2 3 4 5 6 Columns: 21-30 ^a 31-40 ^a 41-50 ^a 51-60 ^a 61-70 ^a 71-80 ^a
4	13	ITCH(1)	Blank 1 2	Cards uncollated; foils restacked between cycles. Cards collated. Cards uncollated; foils counted without restacking.
	28	ITCH(6)	1 Blank	Weighted-average punched cards will be produced, for COMBO. No punched cards.
	34	ITCH(8) ^b	1 Blank	Gold-indium type of weight corrections. Ordinary weight corrections.
	37	ITCH(9) ^b	1 Blank	Fixed-decay-constant calculation. Decay corrections by least-squares fit.
	40	ITCH(10) ^b	1 Blank	Accidental-coincidence corrections. No accidental-coincidence corrections.
	41-45 ^a	TTIRRAD ^b	Blank	Irradiation time, in minutes. Not needed for least-squares fit. 8.88 min. Except for absolute counting, this arbitrary number is usually satisfactory, regardless of actual irradiation time.
	46-55 ^a	TTSCRAM ^b	Blank	Time, in minutes, from end of irradiation to the instant when the clock reading (Columns 56-65) occurred. Not needed for least-squares decay-constant fit. 88.88 min. (Usually satisfactory--see Columns 41-45.)
	56-65 ^a	TCLØCK	Blank	Counter clock reading, in seconds, at some known clock time--usually at or near the start of counting. 0.0 sec. (Clock is frequently reset at start of counting.)
	66-70 ^a	CDR ^b	Blank	Approximate cadmium ratio.
	71-75 ^a	FWZ ^b	Blank	2.0. Nominal foil weight.
			Blank	0.0. } Needed only for gold-indium weight corrections (Column 34).
5	11-70 ^a	DT(I)		Scaler resolving time, in microseconds. Scaler: 1 2 3 4 5 6 Columns: 11-20 ^a 21-30 ^a 31-40 ^a 41-50 ^a 51-60 ^a 61-70 ^a
	71, 72, 73	b		Scaler numbers for accidental coincidence corrections: Channel A, Channel B, and coincidence channel, respectively. Used if there is 1 in Column 40, Card 4.
	74, 75, 76	b		Same for second set of coincidence scalers.
	77, 78, 79	b		Same for third set of coincidence scalers.

^aDecimal point should be included if field is not blank.^bUsually left blank for fission-product fit.^cColumns 11-19 of Card 3 must be blank (or zero) in order to obtain a fission-product least-squares decay fit.

TABLE III. Description of Cards for Weight-card Deck

Card Type	Columns	Format	Name of Variable	Entry	Use
WEIGHT	1-6	A6	MIMS1	WEIGHT Blank	Indicates that card contains the weight of a foil.
	7-10	A4	NAME	[Any]	
	11-20	E10.5	FWT		Weight or intercalibration factor of foil.
	21-30	10X		Blank	Foil weight is set to 1.0.
	31-32	A2	JWT	[Any]	(These columns are not read.)
				CD	Indicates cadmium-covered foil; used for gold-indium type of weight corrections.
				Blank	(Or anything other than CD.) Indicates bare foil, if gold-indium weight corrections are requested.
	33-80	6A8	MESSAGE	[Any]	Contents of this field will be printed out, but are otherwise ignored by program.
NONE	1-6	A6	MIMS1	NONE	(Columns 1-4 contain NONE, with Columns 5 and 6 blank.) Indicates no WEIGHT cards for the corresponding scaler.
	11-20	E10.5	FWT		Number of foils for the corresponding scaler. A decimal point should be used. Each foil will be assigned a weight of one.
				Blank	One foil is assumed.
NEWSET	1-6	A6	MIMS1	NEWSET	Terminates set of WEIGHT cards, when another set of WEIGHT cards follows for the next scaler. This card must not be used except between two sets of WEIGHT cards.
REPEAT	1-6	A6	MIMS1	REPEAT	Applies the latest preceding NONE card or set of WEIGHT cards to the next scaler. May follow another REPEAT card.
LAST+1	1-6	A6	MIMS1	LAST+1	Terminates weight-card deck; counting data follow.

TABLE IV. Examples of Weight-card Decks

Scaler	Six Scalers Used			Four Scalers Used; South Counting Room	Scalers 2 and 4 Deleted; IMAX = 5
1	WEIGHT set	NONE	33.	WEIGHT set NEWSET	WEIGHT set
2	REPEAT	NONE	32.	WEIGHT set	NEWSET
3	REPEAT	REPEAT		REPEAT	WEIGHT set
4	WEIGHT set	NONE	61.	WEIGHT set LAST+1	
5	REPEAT	REPEAT			NONE 23. LAST+1
6	NONE 27. LAST+1	WEIGHT set LAST+1			

TABLE V. Description of Counting-data Cards

South Counting Room		North Counting Room				Name of Parameter	Use
Columns	Format	1967 Equipment		Previous Equipment			
Columns	Format	Columns	Format	Columns	Format		
1-5	A5	1-5	A5	1-5	A5	KARD	Usually contains card number, but any alphanumeric entry is permitted.
6-14	F9.1	6-13	F8.1	6-11	F6.0	CLØCK	Clock time (in seconds) at end of counting interval.
16-24	F9.1	14-19	F6.1	12-16	F5.0	CTIME	Length of counting interval (in seconds). Blank or zero signals end of data deck.
26-34	F9.0	20-27	F8.0	17-24	F8.0	DATA(1)	Number of counts registered by Scaler 1 during the counting interval.
36-44	F9.0	28-35	F8.0	25-32	F8.0	DATA(2)	Counting datum for Scaler 2.
46-54	F9.0	36-43	F8.0	33-40	F8.0	DATA(3)	Same for Scaler 3.
56-64	F9.0	44-51	F8.0	41-48	F8.0	DATA(4)	Same for Scaler 4.
-	-	52-59	F8.0	49-56	F8.0	DATA(5)	Same for Scaler 5.
-	-	60-67	F8.0	57-64	F8.0	DATA(6)	Same for Scaler 6.
66	I1	69	I1	67	I1	IDC(1)	Identification code number for Scaler 1.
67	I1	70	I1	68	I1	IDC(2)	Same for Scaler 2.
68	I1	71	I1	69	I1	IDC(3)	Same for Scaler 3.
69	I1	72	I1	70	I1	IDC(4)	Same for Scaler 4.
-	-	73	I1	71	I1	IDC(5)	Same for Scaler 5.
-	-	74	I1	72	I1	IDC(6)	Same for Scaler 6.
70-80	-	75-80	-	73-80	-	-	Ignored by program; may contain anything.

TABLE VI. Data Deck for COMBO

Card Type No.	Card Type	Columns	Format	Parameter Name	Entry	Use		
1	Problem-title card	1-72	9A8	TITLE	[Any]	Problem title.		
		78	I1	ITCH 1	0	Minimum output.		
					1	Produces intermediate output from NORMGRUP.		
					≥ 2	As above, plus diagnostic output from FOILNORM.		
		79	I1	ITCH 2	0	No punched-card output.		
					≥ 1	Punched weight cards will be produced.		
		80	I1	ITCH 3	0	Program sets ITCH 3 to 3.		
				> 0	ITCH 3 is the multiplier for SIGMA, used in rejecting deviant ratios from the averaging routine in normalizing the scaler data.			
2	Group-title card	1-80	10A8	GTITLE	[Any]	Group title, probably as punched out by RP-202 or NURF.		
3	Scaler-weight card	1-9	9X	-	[Any]	Can have message, such as "SCALR WT."		
		10, 20 30, 40	I1	IJ(JSCALR)	-	Scaler number.		
		11-18 21-28 31-38 41-48 51-58 61-68	F8.0	SCALW(JSCALR)	-	Weighting factor to go with preceding scaler number. Scalers do not have to be in order, but order <u>must</u> correspond to order of weighted-average cards.		
		4	Scaler card	1-8	8X	-	-	Can contain the word "SCALER."
		9-10		I2	I(NSCALR)	-	Contains in Column 10 the scaler number for the succeeding weighted-average cards. This card is punched out by RP-202 or NURF.	
		5	Weighted-average card deck	1-4	A4	NAMA	(Foil Identifier)	These cards are weighted-average cards, for the scaler specified on the preceding scaler card (No. 4), as punched out by NURF or RP-202. Scaler order must correspond to order on scaler-weight card (No.3).
				10-24	E15.7	WTAVA	-	
6	Terminator	1-4	A4	NAMA	SCAL	This card is read under the preceding format. The message SCAL informs the program that another scaler deck follows, starting with scaler card (No. 4).		
					NEXT	This message starts the processing of a new group of foils. Following this card must come card of Type No. 2, etc.		
					LAST	This message terminates the last weighted-average deck of the last group of foils. Next there may come (a) the data for a new problem, starting with a problem title card (No. 1), or (b) an "End of File" card to terminate the calculations.		

TABLE VII. Data Deck for Program TWOSORCE

Card Type No.	Type	Columns	Format	Parameter Name	Entry	Comments
1	Title card	1-72	9A8	TITLE	[Any]	
2	Decay constant	1-30	30X	-	-	Can contain name of element; e.g., "GOLD DECAY CONSTANT." Not read.
		31-50	E20.6	LAMBDA	-	Decay constant in sec^{-1} .
3	Input parameters	1-5	A5	CROOM	NORTH	Code uses appropriate data card formats for North Counting Room.
					SOUTH or blank	Code uses appropriate data card formats for South Counting Room.
		10	I1	NOS	>0	Number of highest scaler used.
					0 or blank	Code sets NOS to 6 for North Counting Room, or 4 for South.
		15-20	6I1	NOT(I)	-	NOT(I) should be nonzero for any unused scaler if $I \leq \text{NOS}$.
		30	I1	ITCH 1	0	Program calculates exact solution of Eq. 9.
					1	Program calculates exact solution of Eq. 9, and also an iterative solution of Eq. 10, using the input coefficients BIN, CIN, DIN.
					2	As for ITCH 1 = 1, plus an iterative solution of Eq. 10 using 1, 1, and 1 for B, C, and D.
		31-40	F10.0	BIN	≠0.0	BIN is input value of B for use in iterative solution of Eq. 10.
					0.0 or blank	Program sets BIN to 3/2, CIN to 8/3, and DIN to 125/24.
		41-60	2F10.0	CIN, DIN	-	Input values of C and D for use in iterative solution of Eq. 10. Note that input values are overruled if BIN = 0.0.
4	Background card	-	-	-	-	First counting-data card is interpreted as a background card. (The IDC numbers are not read.)
5	Counting-data card for source A	-	-	-	-	
6	Counting-data card for source B	-	-	-	-	Cards 5, 6, and 7 can be repeated up to 49 times; the results of all the determinations will be averaged.
7	Counting-data card for source (A + B)	-	-	-	-	
8	Terminator	-	-	-	-	Blank card terminates data and initiates processing and averaging.
	Next problem, if any	-	-	-	-	New problem starts with title card (Type No. 1).
9	⁷ / ₈ END OF FILE	-	-	-	-	Terminates calculations (follows blank card).

ACKNOWLEDGMENTS

W. R. Robinson collaborated in several aspects of the programming, and C. E. Cohn made some valuable comments. Many of the features of NURF were adapted from the RP-202 code, which was written by K. E. Plumlee, and expanded by J. M. Christenson and W. R. Robinson.

REFERENCES

1. K. E. Plumlee and M. T. Wiggins, Automatic Foil Activity Counting Facility and Data-reduction Program, ANL-6628 (Oct 1962).
2. G. S. Stanford, Techniques for Processing Large Quantities of Foil Counting Data, Proc. Conf. on Rad. Meas. in Nucl. Power at Berkeley Nuclear Laboratories, Gloucestershire (Sept 1966), pp. 376-382.
3. R. Beauge, Sections Efficaces Pour Les Detecteurs De Neutrons Par Activation-Recommandées Par Le Groupe De Dosimetrie D'Euratom; Et Coefficients Pour Les Calculs De Flux Et De Spectres A Partir Des Mesures D'Activation, NP-13722 (1963), p. 102.
4. K. E. Plumlee, Theory and Methods of Corrections for Resolution Losses of Counting Equipment, Trans. Am. Nucl. Soc. 7, No. 2, 341 (1964).
5. G. S. Stanford, Foil Data Thickness Corrections as Affected by Self-shielding and Flux Depression, Trans. Am. Nucl. Soc. 5(1), 33-34 (1962).
6. B. S. Garbow, private communication.
7. C. E. Cohn, Effect of Dead Time on Counting Errors, Nuclear Instr. Meth. 41, 338 (1966).
8. D. Halliday, Introductory Nuclear Physics, John Wiley & Sons, Inc., New York (1950), p. 208 ff.
9. K. Way and E. P. Wigner, Radiation from Fission Products, Phys. Rev. 70, 115 (1946).
10. F. H. Clark, Decay of Fission Product Gammas, NDA-27-39 (1954).
11. G. S. Stanford, Computer Techniques for the Investigation of Counting System Deadtime, Reactor Physics Division Annual Report, July 1, 1964 to June 30, 1965, ANL-7110, pp. 247-251 (Dec 1965).