DISCRETE - A program for the automatic analysis of multicomponent exponential decay data

Stephen W. Provencher
sp@S-provencher.COM
http://S-provencher.COM

CONTENTS

A. INTRODUCTION

DISCRETE is a FORTRAN IV program for the automatic analysis of data being represented by:

$$y_k = \sum_{j=0}^{N_\lambda} \alpha_j \exp(-\lambda_j t_k), \quad k = 1,2,\ldots,N, \quad N_\lambda \leq 9,$$

where provision can be made for an unknown baseline component α_0 with $\lambda_0 = 0$. It is completely automatic in that only the raw data (i.e., the y_k and t_k) are input; no potentially biased initial guesses at the α_j, λ_j or N_λ are needed or even allowed.

The approach is the same as in I[†] and II; i.e., the use of transforms to obtain good starting estimates for nonlinear least squares analyses of the data. Improvements include: (1) making direct use of the knowledge that the data is being represented by a discrete sum (rather than an integral over a continuous distribution) of exponentials. (The picturesque but less accurate analysis of overlapping spectral peaks is avoided.) and (2) making much use of the special properties of exponentials for the rapid evaluation of their sums, products, and derivatives. This allows, if necessary, a very thorough and reliable (but without these shortcuts much too time-consuming) grid search in parameter space for least squares solutions. However, for referencing purposes, I and II can be referenced.

This description is meant to be complete as far as using the program and interpreting the results are concerned.

DISCRETE was written with the following order of priorities:

(1) accuracy, resolving power, and reliability

(2) flexibility, wide range of applicability, and ease of use

(3) computation speed

(4) economy of core storage and FORTRAN statements

Priorities (1) and (2) were considered very high and it is by far the most powerful and reliable method I know of for analyzing exponentials. Priorities (1)-(3) were constantly in direct conflict with (4), and hence the program has about 2500 card images.

[†] I ≡ S. W. Provencher, Biophys. J. 16, 27 (1976).

II ≡ S. W. Provencher, J. Chem. Phys. 64, 2772 (1976).

eq. I12 ≡ Equation 12 of I; ref. II3 ≡ reference 3 of II, etc.

There are two versions of DISCRETE. Version 1A (nearly all in single precision) can analyze for up to $N_\lambda = 5$ and must be run on a computer with at least 7 significant figures in its single precision floating point FORTRAN arithmetic. However, this is the bare minimum allowable; on an IBM 370/158 (with $\sim$ 7 significant figures) 4 out of 10 test analyses had difficulties or failed because of loss of accuracy in matrix inversions. Although these were 10 especially difficult cases, Version 1B, which contains double precision parts where needed and can handle up to $N_\lambda = 9$, is strongly recommended for computers with 7 significant figures (and is absolutely necessary for those with less).

About 100 tests on a UNIVAC 1108 (with $\sim 8\frac{1}{2}$ sig. figs.) were less conclusive. Only 3 had difficulties, and these involved closely spaced λ_j with $N_\lambda \geq 4$. In this borderline area of 8 or 9 sig. figs., the choice of versions will depend on economic considerations of core storage and computer time, your data, and how essential thoroughness and reliability are to you. One way to help evaluate your situation would be to initially use Version 1B for the first series of your data sets, take the most difficult cases (i.e., the ones with closely spaced λ_j and large N_λ) and rerun these with Version 1A and compare. Another strategy would be to always use Version 1A and rerun any "suspicious" cases (e.g., where the chosen N_λ seems too small) with Version 1B. However, this tends to defeat the objectivity with which N_λ is automatically determined, since your personal bias would dictate which solutions were "suspicious" and which were not.

Except for this borderline area of 8 or 9 sig. figs., the choice of versions should be clear. With 10 or more sig. figs., you should always be able to use Version 1A (except of course if you feel your data warrants looking for $N_\lambda > 5$).

The computation times and core storage requirements will vary greatly with the size of the problem and the options chosen. However, for a typical case of $N \simeq 200$ and $N_\lambda \leq 4$, a UNIVAC 1108 (with a storage cycle time of 0.75 μsec) requires about 40 sec. CPU time and about 26 K UNIVAC words ($\sim$ 117 K IBM bytes) for Version 1A and about 70 sec. CPU and 40 K words ($\sim$ 180 K bytes) for Version 1B. Overlays (see section F.1) can reduce the core requirements to about 20 K (90 K bytes) and 32 K (144 K bytes) words, respectively.

Versions 1A and 1B have the same input and output structure and, except where noted, everything that follows applies to both versions.

B. COPYING DISCRETE FROM THE TAPES

Instructions for copying DISCRETE from each of the three types of tapes used to distribute it are given below. All three types are standard 9 track, 800 bits per inch, odd parity, unlabeled magnetic tapes.

B.1. EBCDIC tape

This type of tape (as well as the ASCII type in section B.2) contains Version 1A, Version 1B, and test input data in 3 successive files, which are written as (FORTRAN) card images with 80 bytes per record and 800 bytes (10 records) per block. In the first two files the main program is first, followed by the 15 subprograms: BLOCK DATA, DATAIO, WEIGHT, FANLYZ, YANLYZ, LSTSQR, EVAR, VARF, ETHEOR, PIVOT, PIVOT1, ANLERR, FISHER, RESIDU, and PLPRES.

The following control statements were used to read the tape onto a disk of an IBM 370/158:

```
// (JOB card - 40K, 3 sec. CPU)
/*SETUP      TAPE 1E?? WITHOUT RING      (?? depends on your reel no.)
//COPY1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=2400,DSN=SINGL,VOL=SER=T01,LABEL=(1,NL),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DEN=2),DISP=(OLD,PASS)
//SYSUT2 DD DSN=SINGL,UNIT=DISK,VOL=SER=DISK53,SPACE=(CYL,(3,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(NEW,KEEP)
//COPY2 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=2400,DSN=DOUBL,VOL=SER=T01,LABEL=(2,NL),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DEN=2),DISP=(OLD,PASS)
//SYSUT2 DD DSN=DOUBL,UNIT=DISK,VOL=SER=DISK53,SPACE=(CYL,(3,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(NEW,KEEP)
//COPY3 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=2400,DSN=TDATA,VOL=SER=T01,LABEL=(3,NL),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DEN=2),DISP=(OLD,DELETE)
//SYSUT2 DD DSN=TDATA,UNIT=DISK,VOL=SER=DISK53,SPACE=(CYL,(3,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(NEW,KEEP)
//
```

Versions 1A, 1B, and the test input data are then in disk files named SINGL, DOUBL, and TDATA, respectively. They can then be compiled, printed or punched. Most other IBM 360/370 computer centers should be able to use the above statements, perhaps with a

few obvious minor modifications due to installation-dependent conventions. Most non-IBM computer centers should be able to use the above information to read this tape.

B.2. ASCII tape

This tape is the same as the EBCDIC tape described above, except that it was written in (standard 8-bit-per-character) ASCII code rather than EBCDIC.

B.3. UNIVAC tape

This tape is intended specifically for UNIVAC 1100 series computers. It has two files, containing Version 1A and 1B, respectively. Each file also contains the test data, as well as some other useful runstreams as described below. The following run should catalog two files, read the two tape files into them and print a listing of Version 1A and of the test data:

```
@RUN ...  (~60 sec. total time, 12K, 60 pages)
@ASG,T JH T.,U9,2A63R
@REWIND T.
@DELETE,C FILE1A
@DELETE,C FILE1B
@ASG,UP FILE1A,F4C
@ASG,UP FILE1B,F4C
@COPIN T.,FILE1A.
@COPIN T.,FILE1B.
@FREE T
@PRT,S FILE1A.MAIN1A
@PRT,S FILE1A.BLOCKDATA1A
@PRT,S FILE1A.DATAI01A
@PRT,S FILE1A.WEIGHT1A
@PRT,S FILE1A.FANLYZ1A
@PRT,S FILE1A.YANLYZ1A
@PRT,S FILE1A.LSTSQR1A
@PRT,S FILE1A.EVAR1A
@PRT,S FILE1A.VARF1A
@PRT,S FILE1A.ETHECR1A
@PRT,S FILE1A.PIVOT1A
@PRT,S FILE1A.PIVOT11A
@PRT,S FILE1A.ANLERR1A
@PRT,S FILE1A.FISHCR1A
@PRT,S FILE1A.RESIDU1A
@PRT,S FILE1A.PLPRES1A
@PRT,S FILE1A.TESTDATA

@FIN
```

or better, simply replace all this by:

```
@ADD FILE1A.PRINT
```

There may be a few minor installation-dependent changes necessary in the above statements (e.g., more detail in the ASG,TJH statement, or something other than F4O in the ASG,UP statements). The element FILE1A.PRINT contains the control statements to print out Version 1A and the test data. Similarly, the elements .PUNCH, .COMPILE, and .MAP contain the control statements to punch a source deck and the test data, compile, and map (collect an absolute executable element called .MAIN). Thus you could compile, map, and run the test data with

```
@RUN ...
@ADD FILE1A.COMPILE
@ADD FILE1A.MAP
@XQT FILE1A.MAIN
@ADD FILE1A.TESTDATA
@FIN
```

After the absolute element has been produced, later runs need only:

```
@RUN ...
@XQT FILE1A.MAIN
 [data]
@FIN
```

Version 1B can be used by simply exchanging the characters 1A and 1B everywhere above (except in the COPIN statements).

C. ADAPTING DISCRETE TO YOUR COMPUTER

C.1. Compilers tested

Every attempt has been made to adhere to the rules of ANSI FORTRAN IV in order to have full compatability with a wide variety of FORTRAN compilers. DISCRETE has been run mainly with UNIVAC FORTRAN V which is very poor as far as diagnostics go. However 10 test runs were also made with Version 1A and 10 with Version 1B on an IBM 370/158 with each of the compilers: WATFIV, FORTRAN G and FORTRAN H. It is intended that, except for the possible specification changes mentioned in section C.2, no changes whatsoever will be necessary. However, because of the limited experience with other computers at this early stage of distribution, and because of the wide range of computers intended to be used, this cannot be guaranteed.

C.2. Possible necessary changes

There are some possible necessary changes to DATA and DIMENSION statements in the main and BLOCK DATA subprograms that you may have to make (once and for all) to specify the maximum size of your problem and some characteristics of your computer. They are described in self-explanatory comment statements and are clearly set off from the rest of the program by card images with a C in column 1 and stars in columns 2-72.† These parts of Version 1A are shown below (see section D.1 for definitions of the variable names). Version 1B is the same except for numerical values and no NSIGFG. After these changes have been made, you should be ready to compile and run the program with the test data (see section D.3).

---

† Comments set off by card images with a C in column 1 and minus signs in columns 2-72 are found throughout DISCRETE, but are mainly intended for my information only. No attempt has been made to make the program self-explanatory, since it is intended to be widely applicable with no modifications.

```
C                                                                       MAI01900
C*************************************************************************MAI02000
C*************************************************************************MAI02100
C                                                                       MAI02200
C  THE FOLLOWING INSTRUCTIONS DESCRIBE ALL POSSIBLE NECESSARY CHANGES   MAI02300
C  THAT YOU MAY HAVE TO MAKE IN THE MAIN PROGRAM. (SEE ALSO THE         MAI02400
C  BLOCK DATA SUBPROGRAM.)                                              MAI02500
C                                                                       MAI02600
C*************************************************************************MAI02700
C  IF YOU CHANGE NMAX IN THE BLOCK DATA SUBPROGRAM, YOU MUST READJUST   MAI02800
C  THE DIMENSIONS IN THE FOLLOWING STATEMENT TO NMAX -                  MAI02900
C                                                                       MAI03000
      DIMENSION T(333), Y(333), SUMTW(333), YLYFIT(333)                 MAI03100
C                                                                       MAI03200
C*************************************************************************MAI03300
C  ORDINARILY E SHOULD BE DIMENSIONED E(NMAX,6) IN THE FOLLOWING        MAI03400
C  STATEMENT.  HOWEVER, IF YOU WILL ALWAYS INPUT REGINT=.TRUE., AND     MAI03500
C  IF NMAX IS GREATER THAN 333, YOU CAN SAVE STORAGE BY DIMENSIONING    MAI03600
C  IT AS E(1,6) -                                                       MAI03700
C                                                                       MAI03800
      DIMENSION E(333,6)                                                MAI03900
C                                                                       MAI04000
C*************************************************************************MAI04100
C  ORDINARILY GSE SHOULD BE DIMENSIONED GSE(500,4).  HOWEVER, IF YOU    MAI04200
C  WILL ALWAYS INPUT REGINT=.FALSE., AND IF NMAX IS LESS THAN 333,      MAI04300
C  YOU CAN SAVE A LITTLE STORAGE BY DIMENSIONING IT AS GSE(1,4) IN      MAI04400
C  THE FOLLOWING STATEMENT -                                            MAI04500
C                                                                       MAI04600
      DIMENSION GSE(500,4)                                              MAI04700
C                                                                       MAI04800
C*************************************************************************MAI04900
C  IF YOU CHANGE NINTMX IN THE BLOCK DATA SUBPROGRAM, YOU MUST          MAI05000
C  READJUST THE DIMENSIONS IN THE FOLLOWING STATEMENT TO NINTMX -       MAI05100
C                                                                       MAI05200
      DIMENSION TSTART(10), NT(10), DELTAT(10)                          MAI05300
C                                                                       MAI05400
C  THIS IS THE END OF ALL POSSIBLE CHANGES YOU MAY HAVE TO MAKE IN      MAI05500
C  THE MAIN PROGRAM.                                                    MAI05600
C                                                                       MAI05700
C*************************************************************************MAI05800
C*************************************************************************MAI05900
C                                                                       MAI06000
```

C.3. Underflow and WATFIV

In several parts of DISCRETE, arithmetic underflows are expected to occur and be replaced by zero, which is the normal action of most compilers. However the version of WATFIV used stopped execution at underflows, and the following statement had to be inserted after card MAI06200:

```
CALL TRAPS (0,0,99999,0,0)
```

The FORTRAN G and H compilers printed out diagnostics the first few times only and then continued without further diagnostics. If you have one of the few compilers that stops execution at underflows or prints an unlimited number of diagnostics, you will also almost certainly have an option or subroutine call like the one above to prevent this.

The version of WATFIV used also required all DOUBLE PRECISION functions to be explicitly typed (even built-in ones like DBLE).

```
C                                                                       BLKD1700
C*****************************************************************BLKD1800
C*****************************************************************BLKD1900
C                                                                       BLKD2000
C  THE FOLLOWING INSTRUCTIONS DESCRIBE ALL POSSIBLE NECESSARY CHANGES   BLKD2100
C  THAT YOU MAY HAVE TO MAKE IN THE BLOCKDATA SUBPROGRAM. (SEE THE      BLKD2200
C  MAIN PROGRAM ALSO.)                                                  BLKD2300
C                                                                       BLKD2400
C*****************************************************************BLKD2500
C  ALL STATEMENTS FOR READING CARDS AND FOR PRINTING OUTPUT ARE OF      BLKD2600
C  THE FORM READ (IREAD,.... AND WRITE(IWRITE,..... THUS, IREAD AND     BLKD2700
C  IWRITE MAY HAVE TO BE CHANGED TO THE VALUES APPROPRIATE FOR YOUR     BLKD2800
C  COMPUTER CENTER IN THE FOLLOWING STATEMENT -                         BLKD2900
C                                                                       BLKD3000
      DATA IREAD/5/,IWRITE/6/                                           BLKD3100
C                                                                       BLKD3200
C*****************************************************************BLKD3300
C  IN THE FOLLOWING STATEMENT, NSIGFG MUST BE SET TO THE NUMBER OF      BLKD3400
C  SIGNIFICANT FIGURES CARRIED IN THE SINGLE PRECISION FLOATING POINT   BLKD3500
C  ARITHMETIC OF YOUR COMPUTER (E.G., 7 FOR IBM 360, 8 FOR              BLKD3600
C  UNIVAC 1100, E(C.) -                                                 BLKD3700
C                                                                       BLKD3800
      DATA NSIGFG/8/                                                    BLKD3900
C                                                                       BLKD4000
C*****************************************************************BLKD4100
C  IN THE FOLLOWING STATEMENT, LINEPG MUST BE SET TO THE NUMBER OF      BLKD4200
C  LINES PER PAGE AVAILABLE FOR PRINTED OUTPUT ON YOUR PRINTER -        BLKD4300
C                                                                       BLKD4400
      DATA LINEPG/62/                                                   BLKD4500
C                                                                       BLKD4600
C*****************************************************************BLKD4700
C  NMAX MUST BE GREATER THAN OR EQUAL TO THE MAXIMUM NUMBER OF DATA     BLKD4800
C  POINTS YOU WILL EVER USE. CORE STORAGE CAN BE SAVED BY KEEPING       BLKD4900
C  NMAX AS LITTLE ABOVE THIS MAXIMUM AS POSSIBLE.  IF YOU CHANGE NMAX   BLKD5000
C  IN THE FOLLOWING STATEMENT, YOU MUST ALSO RESET THE DIMENSIONS AS    BLKD5100
C  DIRECTED IN THE MAIN PROGRAM -                                       BLKD5200
C                                                                       BLKD5300
      DATA NMAX/333/                                                    BLKD5400
C                                                                       BLKD5500
C*****************************************************************BLKD5600
C  NINTMX MUST BE GREATER THAN OR EQUAL TO THE MAXIMUM VALUE OF NINT    BLKD5700
C  YOU WILL EVER USE WITH NEGINT=.TRUE.. IF YOU CHANGE                  BLKD5800
C  NINTMX IN THE FOLLOWING STATEMENT, YOU MUST ALSO RESET THE           BLKD5900
C  DIMENSIONS AS DIRECTED IN THE MAIN PROGRAM -                         BLKD6000
C                                                                       BLKD6100
      DATA NINTMX /10/                                                  BLKD6200
C                                                                       BLKD6300
C  THIS IS THE END OF ALL POSSIBLE NECESSARY CHANGES YOU MAY HAVE TO    BLKD6400
C  MAKE IN THE BLOCK DATA SUBPROGRAM                                    BLKD6500
C                                                                       BLKD6600
C*****************************************************************BLKD6700
C*****************************************************************BLKD6800
C                                                                       BLKD6900
```

For Version 1A this involved only adding DBLE, DABS, and DEXP to the DOUBLE PRECISION statement in subprogram ETHEOR. This is not even legal in most compilers and since WATFIV is too slow to be used routinely, the long list of DOUBLE PRECISION functions for Version 1B will be furnished on request rather than here.

D. INPUT DATA

In this section the input variables are defined and discussed (in approximate order of input), the input data deck is described and an example given.

D.1. Input variable names

LABEL = up to 80 Hollerith characters that will appear as a heading at the top of pages throughout the output.

LAST = T (for .TRUE.) if the present data set is the last one in the input deck.
= F (for .FALSE.) if another data set(s) is to follow. As many data sets as desired may be analyzed in one run.

REGINT = T if the data points are in "regular intervals" of t; i.e., if the y_k and the t_k are read in in NINT groups such that for each group L there are NT(L) t_k running in equal increments between TSTART(L) and TEND(L). For example, if the t_k are

1,2, 4,6,8,10, 20,30,40,50, 100,200,300,400,500

then you could input REGINT=T, NINT=4,

L	TSTART	TEND	NT
1	1.	2.	2
2	4.	10.	4
3	20.	50.	4
4	100.	500.	5

For this artifically small set there is no advantage to this, but when the total number of data points N $\gtrsim$ 10•NINT, this input option can not only save input effort (by not requiring the t_k), but much more importantly core storage (see card images MAI03400-MAI03900 in section C.2) and a great deal of computer time, since long sums of exponentials can be evaluated by simple formulas for geometric sums. For large N, where this is most important, the data is often automatically sampled in such sets of regular intervals. Test data set 1 in section D.3 provides another example of this input option.

The following conditions must always be satisfied:

NINTMX $\geq$ NINT $\geq$ 1 (see card images BLKO57OO-BLKO62OO in section C.2)

NT(L) $\geq$ 2

$$\text{NMAX} \geq \sum_{L=1}^{\text{NINT}} \text{NT(L)}$$ (see card images BLKO48OO-BLKO54OO in section C.2)

TSTART(L) < TEND(L); however, there are no restrictions on any relations between TSTART(L) and TSTART(K), TSTART(L) and TEND(K), or TEND(L) and TEND(K), K≠L. For example, TSTART(L+1) < TEND(L) is allowed.

NOBASE = T if it is known a priori that there is *no* baseline component α_0.

NONNEG = T if it is known a priori that $\alpha_j \geq 0$ for j >0; e.g., if the α_j correspond to concentrations or populations. If there is a baseline component, α_0 is *not* so constrained.

PRY = T if the input values of y_k, t_k, and (if least squares weights w_k are read in) $w_k^{1/2}$ are to be printed out (as discussed in section E.3).

PRPREL = T if results of each iteration of the preliminary analyses (as discussed in section E.5) are to be printed out.

PRFINL = T if the results of each iteration of the final analyses of the raw data (as discussed in section E.4) are to be printed out.

PLOTRS = T if the residuals of the fit of the final solution to the raw data are to be plotted by the printer (as discussed in section E.2).

REPEAT = T if the final one- or two-page summary of the results (as discussed in section E.1) is to be printed a second time. This second summary can be detached from the full list and kept in a more compact binder with summaries of other runs.

NLAMMX = the maximum value of N_λ that will be searched for. The following condition must be satisfied:

$1 \leq$ NLAMMX ≤ 5 (Version 1A)

$1 \leq$ NLAMMX ≤ 9 (Version 1B)

To preserve objectivity and check for systematic experimental errors or unexpected components, it is always a good idea to choose NLAMMX at least 1 larger than the maximum N_λ you expect. On the other hand, since most of the computer time is usually spent looking for components that are not here, it is a great waste to put in a value of NLAMMX that is too large for the data to determine (e.g., NLAMMX = 4 for 40 points with equally spaced t_k and 5% experimental error).

IWT = 1 for the normal (unweighted) case of unit least squares weights w_k; i.e., when $\sigma(y_k)$, the expected error in y_k, is independent of k.

$$= 2 \text{ for } w_k = \left[\left| \sum_{j=0}^{N_\lambda} \alpha_j \exp(-\lambda_j t_k) \right| + \text{ERRFIT} \right]^{-1},$$

i.e., $\sigma(y_k) \propto [y(t_k)]^{1/2}$, where ERRFIT is described below and $y(t_k)$ is the exact value without random experimental error. This weighting is appropriate (to a good approximation) for many counting and correlation experiments.

$$= -2 \text{ for } w_k = \left[\left| \sum_{j=1}^{N_\lambda} \alpha_j \exp(-\lambda_j t_k) \right| + \text{ERRFIT} \right]^{-1},$$

i.e., $\sigma(y_k) \propto [y(t_k) - \alpha_0]^{1/2}$

$$= 3 \text{ for } w_k = \left[\left| \sum_{j=0}^{N_\lambda} \alpha_j \exp(-\lambda_j t_k) \right| + \text{ERRFIT} \right]^{-2},$$

i.e., $\sigma(y_k) \propto y(t_k)$

$$\text{IWT} = -3 \text{ for } w_k = \left[\left| \sum_{j=1}^{N_\lambda} \alpha_j \exp(-\lambda_j t_k) \right| + \text{ERRFIT} \right]^{-2} ,$$

i.e., $\sigma(y_k) \propto y(t_k) - \alpha_0$

= 4 if you are going to input your own w_k.

Thus, when IWT = ± 2 or ± 3, the w_k are calculated from a smooth least squares fit to the y_k, rather than directly from y_k, which would lead to erratic and biased weights. However, even the smooth curve could lead to a disastrously large w_k if it happened to come very close to the x axis near a t_k. ERRFIT is the standard deviation of the fit to the 10 y_k in the interval where the least squares curve is closest to the x axis, and eliminates this danger.

MTRY = the maximum number of tries that will be made to find a solution for a single value of N_λ during the preliminary analyses of the transforms. If the first try fails, a grid search is performed in (λ_j-) parameter space and up to MTRY-1 tries are started from points on this grid. You must input:

$1 \leq \text{MTRY} \leq 45$

However, these limits are extreme values; e.g., for MTRY = 1, no grid search is performed at all. DISCRETE was tested on the 24 data sets used in I and II, as well as 15 more difficult cases that could only be successfully analyzed by DISCRETE. Of these 39 unusually difficult cases, only 3 needed MTRY between 11 and 13; all others worked with MTRY = 5 and nearly all with MTRY = 3. During stages when DISCRETE is searching for more components than there actually are, the computer time is often almost directly proportional to MTRY, since MTRY unsuccessful searches are often performed. Thus if the size of the problem makes computer time a major consideration, MTRY = 5 should be adequate. If reliability and thoroughness are of utmost importance an MTRY ∿15 can be used. An MTRY = 45 would be practically always a great waste of computer time.

N = total number of data points. You must have

$2 \cdot \text{NLAMMX} + 2 < N \leq \text{NMAX}$ (see card images BLK04800-BLK05400 in section C.2).

$\underline{T}$ = array of t_k values. They are only needed when REGINT = F, and in this case they need not be in any special monotonic order and need not have any regular spacing (duplicate values are even permitted).

$\underline{Y}$ = array of data points y_k measured at t_k.

$\underline{W}$ = array of least squares weights w_k.

There is no need to normalize any of the input data like TSTART, TEND, T, Y, or W. However, the number 1.E+20 has been used throughout the program to represent an infinitely large number and hence nothing should get larger than 10^{20}. Since some of the input quantities are squared and summed over N, it is best to choose units so that all input values are between about 1.E-9 and 1.E+6 in absolute value.

D.2. Input data deck

Below are listed the data cards in order of input, the FORTRAN FORMAT in parentheses, the input variable list, and finally in parentheses when the cards are necessary.

```
Card 1   (80A1)    (LABEL(J),J=1,80)    (always needed)
Card 2   (9L1,3I3)   LAST, REGINT, NOBASE, NONNEG, PRY, PRPREL,
                     PRFINL, PLOTRS, REPEAT, NLAMMX, IWT, MTRY
                     (always needed)
Card 3A  (I5)    N    (only if REGINT=F)
Card set 4A  (5E15.6)    (T(J),J=1,N)    (only if REGINT=F)
Card 3B  (I5)    NINT    (only if REGINT=T)
Card set 4B  (2E15.6, I5)    (TSTART(J),TEND(J),NT(J),J=1,NINT)
             (only if REGINT=T)
Card set 5   (5E15.6)    (Y(J),J=1,N)    (always needed)
Card set 6   (5E15.6)    (W(J),J=1,N)    (only if IWT=4)
Card 1 ...
  :          } only if the preceeding LAST=F
Card set 6 ...
  :
  :
```

Card 1 - Card set 6 comprise one data set. As many data sets as desired may be run together; you must input LAST=F in all but the last data set.

D.3. Test input data

The card images of the test data provided on the tapes are shown below.

```
 TEST DATA SET 1 - WITH PROPER UNIT WEIGHTING AND NORMALLY COMPREHENSIVE OUTPUT (Card 1)
FTFFTFTTF  4  1  5 (Card 2)
    2 (Card 3B)
    .000000          .450000E02    10 } (card set 4B)
    .750000E02       .945000E03    30 )
    .317951E01       .272385E01       .238005E01       .217513E01       .196817E01 \
    .183133E01       .171227E01       .166732E01       .160041E01       .152702E01  |
    .128123E01       .103639E01       .911274E00       .793438E00       .704310E00  |
    .649749E00       .595533E00       .499157E00       .461291E00       .351978E00  } (Card set 5)
    .381508E00       .328100E00       .273046E00       .263287E00       .266722E00  |
    .248385E00       .177709E00       .133162E00       .155815E00       .817959E-1  |
    .103332E00       .161067E00       .768593E-1       .119229E00       .639275E-1  |
    .587735E-2       .199340E-1       .295760E-1       .000000          .292465E-1 /
 TEST DATA SET 2 - WITH INCORRECT WEIGHTING AND MOST COMPREHENSIVE OUTPUT (Card 1)
TFFFTTTTT  4  -3  5 (Card 2)
   40 (Card 3A)
    .000000          .500000E01       .100000E02       .150000E02       .200000E02 \
    .250000E02       .300000E02       .350000E02       .400000E02       .450000E02  |
    .750000E02       .105000E03       .135000E03       .165000E03       .195000E03  |
    .225000E03       .255000E03       .285000E03       .315000E03       .345000E03  } (Card set 4A)
    .375000E03       .405000E03       .435000E03       .465000E03       .495000E03  |
    .525000E03       .555000E03       .585000E03       .615000E03       .645000E03  |
    .675000E03       .705000E03       .735000E03       .765000E03       .795000E03  |
    .825000E03       .855000E03       .885000E03       .915000E03       .945000E03 /
    .317951E01       .272385E01       .238005E01       .217513E01       .196817E01 \
    .183133E01       .171227E01       .166732E01       .160041E01       .152702E01  |
    .128123E01       .103639E01       .911274E00       .793438E00       .704310E00  |
    .649749E00       .595533E00       .499157E00       .461291E00       .351978E00  } (Card set 5)
    .381508E00       .328100E00       .273046E00       .263287E00       .266722E00  |
    .248385E00       .177709E00       .133162E00       .155815E00       .817959E-1  |
    .103332E00       .161067E00       .768593E-1       .119229E00       .639275E-1  |
    .587735E-2       .199340E-1       .295760E-1       .000000          .292465E-1 /
```

The raw data for two data sets is the same. It was simulated using the α_j and λ_j parameters of Example II of I and II for $j = 1,2,3$. However, noise with a constant rms error = $0.01[y(0)-\alpha_0] = 0.03395$ was added instead of the 5% rms error of Example II. The baseline error was $\alpha_0 = -0.207$. Hence data set 1 has the appropriate weighting (i.e., unweighted) with IWT = 1, while set 2 has an incorrect IWT = -3. In set 2, REGINT=F in order to test the corresponding part of DISCRETE and illustrate the input format for this case; however, this is actually a great waste of computer time. The CPU time for this test data with the two sets was 75 sec. on a UNIVAC 1108 using Version 1B.

You should run this test data first and compare your results with the output listing accompanying the tape. You should not expect perfect agreement down to the last figure, especially for quantities like N_ϕ, error estimates, etc., whose accuracy depends on the accuracy of matrix inversions. This is especially true for nearly singular solutions with a large N_ϕ (e.g., the third best

solution on p. 27 of the output).

D.4. Modifying input format

All the input is read in subprogram DATAIO. If your raw data always has a special or unusual format or weighting, or if it needs some preprocessing, you may wish to make changes in DATAIO.

E. INTERPRETING THE OUTPUT

In this section we discuss in detail the interpretation of the output. Examples are drawn from the output from the test data (page numbers refer to the output list sent with the tape). The output is discussed in decreasing order of importance rather than in order of printout.

E.1. Final summary of results

This is the most important output and is always printed out. For test data set 1 it is on p. 7, and for test data set 2 it is on pp. 27-28 and, because REPEAT = T, again on pp. 29-30. The first line has the contents of LABEL. Then each of the five best solutions are summarized, starting with the one chosen best, then the second best and so on down to the fifth best (if there are that many). The terms in the printout are defined below in terms of quantities already defined above or in I or II.

ALPHA,LAMBDA = α_j^Y, λ_j^Y of I (i.e., the α_j, λ_j of the final solution).

STD ERR = $\sigma(\alpha_j^Y)$ and $\sigma(\lambda_j^Y)$ of I (i.e., the estimated standard error of the respective α_j and λ_j).

These are obtained in the usual way (see ref. II3) from the estimated variance-covariance matrix. Thus, ignoring the non-linearity, the 95% confidence intervals are approximately $\pm 2\sigma(\alpha_j^Y)$ and $\pm 2\sigma(\lambda_j^Y)$.

If you are interested in a more detailed error analysis, you can always use the α_j, λ_j and the raw data as input for a straightforward, but time consuming, exploration of the contour lines of the variance surface to get even more accurate confidence regions. Furthermore, since Beale's N_ϕ is also printed out you could correct approximately for nonlinearity [see ref. II14 and I. Guttman and D. A. Meeter, Technometrics 7, 623 (1975)]. However STD ERR is probably adequate for most situations, especially with N_ϕ as a guide. An $N_\phi \gtrsim 1$ indicates that the problem is so nonlinear that even corrected confidence regions would be unreliable, and an $N_\phi \lesssim .1$ indicates that the corrections would be reliable but also small.

PERCENT = $100\sigma(\alpha_j^Y)/|\alpha_j|$ and similarly for λ_j.

STARTING LAMBDA = the λ_j that was obtained from the preliminary least squares solution of the transforms, and was then used as a starting value for the final least squares fit to the raw data.

The main purpose and advantage of this method is the ability to get good unbiased starting values.

Next to STARTING LAMBDA is printed (FROM FIT TO TRANSFORMS - ν TRIES), where ν is the number of preliminary analyses that had to be attempted before a solution was found. If no solution was found after MTRY tries, ν = MTRY, the set of λ_j from the best fit to the transforms is used, and (NO EXACT FIT TO THE TRANSFORMS FOUND) is printed next to each starting λ_j (e.g., see the third best solution p. 27).

PNG(K/J) = $P_{NG}(K|J)$ of eq. II19a.

Since this quantity is the key to deciding N_λ and the "best" solution, the short discussion of II will be expanded upon. $P_{NG}(K|J)$ is the approximate probability that the solution with $N_\lambda = K$ is worse (or less likely) than the solution with $N_\lambda = J$. It is defined using Beale's nonlinearity parameter N_ϕ, which he intended to be used to modify Fisher's F distribution to approximately correct for moderate nonlinearity in estimating confidence regions. The extension to P_{NG} is somewhat arbitrary: When comparing two solutions, P_{NG} is defined in such a way that the burden is placed on the solution with the larger N_λ to pass an F test that has been corrected approximately for the added uncertainty due to the non-linearity at this solution. For example, direct application of the usual linear hypothesis test without correction, often failed because too large an N_λ was chosen (see also ref. II1). This solution had a slightly better fit often because a small very short-lived false component made an improvement to the fit to the first few data points and then decayed away. As one would expect, a few data points cannot determine an α and a λ with very much certainty, and N_ϕ measures the large nonlinearity at such a solution. Just as with rigorous linear hypothesis tests, when P_{NG} rejects such a solution it does not imply that the short-lived component does not exist; it only means that there is not enough information in the data to decide. Only further experiments (e.g., with the t range extended

toward smaller t) could decide.

Although P_{NG} might be better called a "guide" or a "criterion" rather than a full-fledged "probability", hundreds of numerical experiments have shown it to be very effective. Of course, just as with rigorous hypothesis tests, the usual significance levels should be considered. To stress this, whenever a second best solution has a $P_{NG} < 0.95$, messages are printed in the summaries of the best and second best solutions reminding you to LOOK AT SECOND BEST SOLUTION ALSO since it is still A SIGNIFICANT POSSIBILITY. For example, $P_{NG}(3|2) = 0.581$ on p. 27, means that one cannot say with any reasonable confidence whether $N_\lambda = 2$ or 3. The incorrect weighting has so distorted the solution that for $N_\lambda = 3$ there is much more uncertainty, as reflected by $N_\phi = 0.623$, than in the correctly weighted solution on p. 7 with $N_\phi = 0.027$. The third best solution on p. 27 is disastrously nonlinear with huge percent errors, $N_\phi = 142.$, and $P_{NG}(4|2) = 1.000$; however the UNCORRECTED PNG (i.e., from the standard F test with $N_\phi = 0.$) WOULD BE an inconclusive 0.725 (as printed out at the far right of that line).

NPHI = N_ϕ of II (see above discussion of PNG)

ITERATIONS IN FIT = the number of iterations needed for the final least squares fit to the raw data to converge to the solution.

STD.DEV. OF FIT = σ_{yy} of eq. I27 (except that if NOBASE=T, replace $n-2N_\lambda-1$ by $n-2N_\lambda$).

SIGNAL/NOISE RATIO OF FIT = $|y_{ext}|/\sigma_{yy}$, where y_{ext} is the y_k with the maximum absolute value. This quantity is only useful if the rms errors of the y_k are independent of k, i.e., IWT = 1; otherwise it is not printed.

Since the rms noise added to the test data was $0.01[y(0)-\alpha_0] \approx 0.01y_{ext}$, we would expect a S/N $\approx$ 100 on p. 7, which is true for the first two solutions.

LAMBDA HELD BETWEEN λ_{min} AND λ_{max}, where λ_{min} and λ_{max} are the limits defined in eqs. II17 and II9b (except that q = 0.02 if NOBASE = T, and t_1, t_5 and t_n represent the t_k but reordered if

necessary in monotonically increasing order with k).

During all the least squares analyses the λ_j are constrained to be within these limits. This prevents the λ_j from entering physically unfeasible (or, at least for the range of the t_k, physically indeterminable) regions, and from perhaps causing arithmetic overflows. It also gives the solution a chance to rebound back into a feasible region after a bad iteration step.

PROB. RESIDUALS UNCORRELATED = $1-A(\tau_K|N-K-2)$, K = 1,2, ... 5, as defined in eqs. II2O.

$$\text{PUNC} = \frac{1}{2.283333} \sum_{K=1}^{5} \frac{[1-A(\tau_K|N-K-2)]}{K}$$

This is slightly different from the P_{unc} of eq. II2Oa. It weights the average to better reflect the fact that the autocorrelations with the shortest lags are the most powerful detectors of nonrandomness in the residuals caused by poor fits to the data. This is especially true for small N, as can be seen in the third best solution on p. 7. Even a plot of the residuals would show them to be strongly correlated, but, because N is so small, a lag of 4 or 5 spans too much of the data, and the correlation is not seen for K = 4 or 5. Thus for $N \lesssim 100$, even the above PUNC is too weakly weighted and one should look more closely at K = 1,2, and 3. Actually the whole test is quite weak for small N,† and there are in general large fluctuations in the estimated probabilities versus K. Except for small N, where you should mentally place more weight on K = 1 and 2 (especially if they are both $\lesssim$ 0.005, as in this example), you should only consider a PUNC $\lesssim$ 0.050 a significant indication that there are systematic errors in the data or that the N_λ is too small. A PUNC $\gtrsim$ 0.050 is inconclusive. For MTRY $\gtrsim$ 5, DISCRETE is quite thorough in finding all possible solutions and P_{NG} is a much more powerful test for deciding N_λ than PUNC. Thus the main function of PUNC is to detect systematic experimental errors in the data or non-exponential components.

The autocorrelation sums are evaluated in the order that the t_k and y_k were read in; i.e., there is no reordering with respect

† The least squares fit yields n residuals with only $n-2N_\lambda-1$ degrees of freedom, and this introduces an extra correlation into the residuals. This can cause a significant false reduction in PUNC when $(n-2N_\lambda-1)/n$ is small.

to t_k. Thus, if for some reason the t_k are not monotonically increasing (or if REGINT=F, perhaps monotonically decreasing) with k, PUNC loses its meaning.

E.2. Plot of residuals

If PLPRES = T, the weighted residuals d_k (see eq. II18b) for the best solution are plotted on the printer (see pp. 6 and 26). The abscissa value is the subscript k; i.e., just as for PUNC, there is no reordering with respect to t_k.

The plots are useful for two reasons:

(1) You can quickly spot a gross error in the input value of a y_k; the abscissa value gives you the subscript of the data point.

(2) Although PUNC is usually much more powerful at detecting systematic errors, p. 26 illustrates one case where PUNC is quite helpless and the plots are useful. When the wrong type of weighting is used, there can be a steady increase or decrease in the magnitude of the residuals, although they remain well scattered about the value 0. This is clearly seen on p. 26, but PUNC is actually unusually large on p. 27. Even the nonrandomness of the first 18 residuals is not detected by PUNC because their small magnitude makes their contribution to the autocorrelation sums small.

E.3. Summary of input

The first page of output always starts with LABEL and the input parameters (see pp. 1 and 8). If PRY = T, the t_k,y_k, and (if IWT=4) $w_k^{1/2}$ are printed under the headings T, Y, and SQRT(WT).

E.4. Final least squares analyses of the raw data

If PRFINL = T, the results of each iteration of the final least squares fits to the raw data are printed (see pp. 2-5, 22-25). The terms appearing in this output are defined below.

ITR = the number of the iteration

VARIANCE = v of eq. II18a.

An asterisk next to the value means that the variance has increased (e.g., see p. 5). This can occur if the normal equations matrix is nearly singular, or sometimes at the end of the analysis when the variance is already very close to the minimum.

DAMPING Q = the usual damping factor used to modify the length of a Gauss step in nonlinear least squares (e.g., q on p. 155 of ref. II3).

A value of Q consistently near 1 indicates the problem is nearly linear, and a Q $\lesssim$ 0.01 indicates a highly nonlinear situation (of course, N_ϕ is a much more accurate measure of nonlinearity). The value of Q giving the minimum v is first estimated by fitting a quadratic polynomial usually to $v_{Q=0}$, $(dv/dQ)_{Q=0}$, $v_{Q=1}$. If v for this Q turns out to be larger than any of the v's used for the quadratic fit, the Q with the lowest v is used and an asterisk is printed next to it (as in ITR 5, p. 2).

BASELINE = α_0 (only present if NOBASE=F).

ALPHA,LAMBDA = current values of α_j,λ_j.

If a λ_j was temporarily held out of the regression, either because it would violate a constraint or lead to a nearly singular normal equations matrix, an asterisk is printed next to it (as in ITR 6 and 24, p. 5). When an α_j is held, it is obvious, since it is set to zero (as in ITR 23, p. 5).

CORRELATION COEFFICIENTS Unless a nearly singular normal equations matrix occurs (as on p. 5), a table is printed of the usual (see ref. II3) estimates of the coefficients of correlation between parameter pairs (as on pp. 2-4).

The remaining quantities, like NPHI, STANDARD DEVIATION, etc., have the same meanings and numerical values as in the final summary (see section E.1).

This output is not essential, and some of it (like the asterisks next to Q) was of interest mainly for development and testing. However it does give more complete pictures of the final least squares analyses. Furthermore, a final least squares solution that is for some reason rejected does not appear in the final summary.

The only way to know why it was rejected is to have PRFINL = T; all rejections are indicated by a diagnostic message (as on p. 5) or in the final iteration by an asterisk next to a λ_j or an α_j=0.

Another reason to have PRFINL = T is that it is conceivable that the computation of N_ϕ for a nearly singular "solution" (more likely a search stalled by the near singularity rather than a real minimum) can cause an arithmetic overflow to kill the run. An overflow has never occurred anywhere in DISCRETE so far, but values as large as $N_\phi \approx 10^{22}$ have occurred (twice in about 500 times). Since such a large N_ϕ obviously means an unacceptable solution and since N_ϕ is only calculated at this final stage, you could simply repeat the run with NLAMMX = N_λ - 1, where N_λ is value for which N_ϕ overflowed. (The small probability that a still higher N_λ might have given a good solution multiplied by the probability of the overflow occurring is very small.) Without PRFINL = T, you would not know the N_λ for which the overflow occurred.

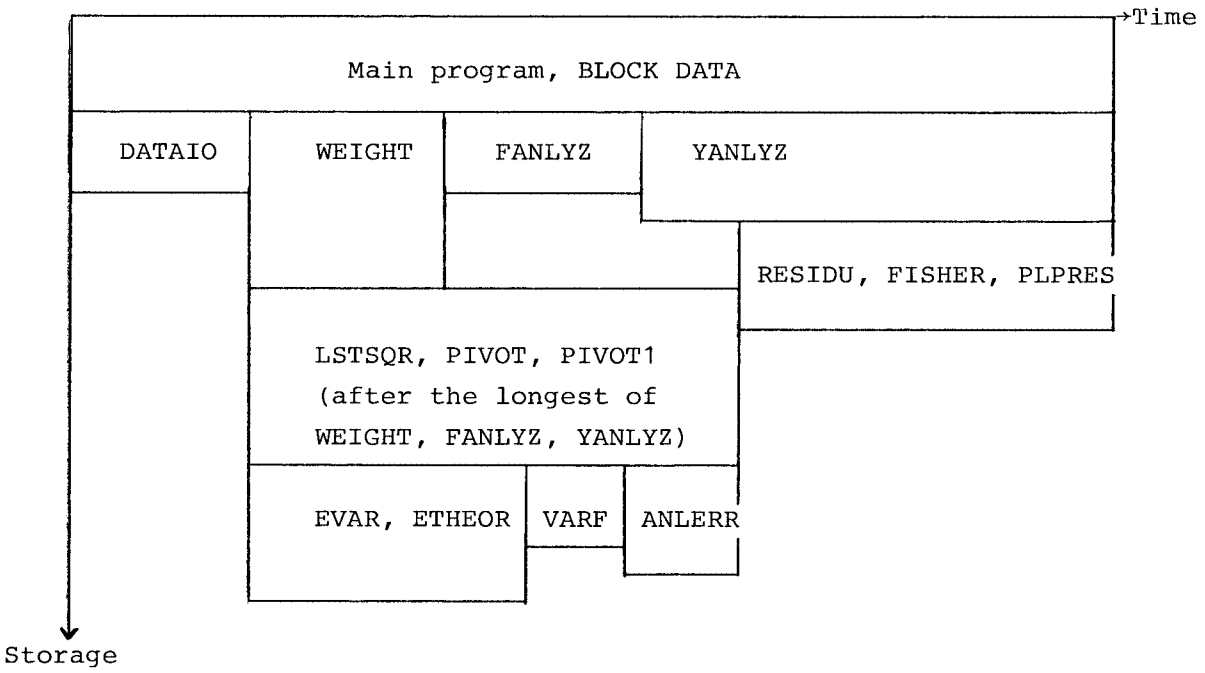
E.5. Preliminary analyses

If PRPREL = T, results are printed for each iteration in the preliminary analyses to determine the weights (if IWT = $\pm$ 2 or $\pm$ 3) and for each iteration in the preliminary analyses of the transforms. The output format is the same as discussed in section E.4. If weights must be calculated or if one or more of the searches requires all MTRY tries, there can be quite a bit of output (see pp. 9-21). This was mainly of interest during development and testing and is probably not of interest for routine data analysis.

F. MISCELLANEOUS CONSIDERATIONS

F.1. Overlays

As illustrated in section A considerable core storage can be saved with the use of overlays (different parts of the program that are never used simultaneously sharing the same storage area). The element .MAP on the UNIVAC tape does this. For other users, the following diagram (not to scale) shows what subroutines can share storage area:



G. ACKNOWLEDGEMENTS

I am very grateful to Gerhard Kessling for complete advice and guidance in making the tapes and in testing the program on the IBM 370/158 at the Aerodynamische Versuchsanstalt (AVA), Göttingen, and to the staff of the AVA for efficient and helpful service. The UNIVAC tapes and tests were made on the UNIVAC 1108 at the Gesellschaft für wissenschaftliche Datenverarbeitung m.b.H., Göttingen.

DISCRETE Update 1

A. A POSSIBLE MODIFICATION TO DISCRETE

The last paragraph on E.1(3) and the first on E.1(4) of the user's manual explain that λ is held between λ_{min} and λ_{max}. In very rare cases it can occur that these limits are too strict. In particular when there is a component with a very large amplitude and $\lambda > \lambda_{max}$, λ can be held at λ_{max}, the solution rejected and not even appear in the final summary of results (see E.1.). This solution can be seen, however, in the final least squares analyses of the raw data (see E.4) if PRFINL=T. The following procedure is therefore suggested:

(1) Always first run DISCRETE in its original form with PRFINL=T.

(2) If the final summary of results does not list all NLAMMX solutions, check through the final least squares analyses of the raw data to see if any of the solutions missing from the final summary have a VARIANCE significantly less than that of the BEST SOLUTION in the final summary. If so, then check to see if this missing solution has a LAMBDA equal to λ_{min} or λ_{max}. [Information on where to find the numerical values of λ_{min} and λ_{max} is given at the bottom of E.1(3).]

(3) Only if all of these conditions have occurred is it necessary to run a version of DISCRETE modified as outlined below.

$\lambda_{min} \propto$ TNLAMN(J), J=1 if NOBASE=F
J=2 if NOBASE=T

$\lambda_{max} \propto$ T1LAMX

[More details are given on E.1(3).] The current values are set in subroutine WEIGHT on card WT003200 in Version 1A and on card WT003700 in Version 1B in a DATA statement:

```
TNLAMN/.2, .02/,  T1LAMX/2.08/
```

You can therefore compile another version with TNLAMN set smaller (by up to a factor of ~ 10) or T1LAMX set larger (by up to a factor of ~ 3). For reasons given in the first paragraph of E.1(4), it is not recommended to routinely run DISCRETE with these modified limits; it is only recommended in those rare cases that, after running the normal version, all the conditions in (2) are satisfied.

B. SOME MINOR CORRECTIONS TO THE USER'S GUIDE (20. May 1976)

D.1(4), lines 8 and 10: change y axis to x axis
" , next to the last line: change 2·NLAMMX+1 to 2·NLAMMX+2
E.1(1), lines 21 and 22: change $\pm$ 3σ to $\pm$ 2σ

C. MODIFICATION FOR SPECIAL WEIGHTING OF PHOTON CORRELATION SPECTROSCOPY (PCS) DATA

In PCS the measured second-order correlation function is often represented as:

$$z(\tau_j) = B \{1 + \Delta_1 + [g(\tau_j) + \Delta_2]^2\} , \; j = 1, \ldots, N, \qquad (1)$$

where B is measured, Δ_1 and Δ_2 are small unknowns that arise from dust, stray laser light, etc., and $g(\tau)$ is proportional to the absolute value of the first-order correlation function.[1,2] If Δ_1 is ignored, DISCRETE can be used to fit $g(\tau)$ to a sum of exponentials by rearranging eq. (1) to:

$$\eta(\tau_j) = [B^{-1} z(\tau_j) - 1]^{1/2} ,$$

where

$$\eta(\tau_j) \equiv g(\tau_j) + \Delta_2$$

The following input data should then be used:

$Y(J) = [B^{-1} z(\tau_j) - 1]^{1/2}$
NOBASE = F
NONNEG = T
IWT = 2

With this input, only one change in subroutine WEIGHT is necessary. Change card WTO16600 in Version 1A or card WTO17000 in Version 1B from:

```
IF (IWT .EQ. 2) SQRTW(K)=SQRT(SQRTW(K))
```

to:

```
IF (IWT .EQ. 2) SQRTW(K)=1./SQRT(1.+SQRTW(K)**2)
```

Notes:

(1) If there is a j such that $B^{-1} z(\tau_j) < 1$, then you must discard this data point and all later ones.

(2) If you want to ignore Δ_2 (not recommended), then input NOBASE=T.

(3) The above change in WEIGHT assumes that the variance in $z(\tau)$ satisfies [3]

$VAR(z) \propto z$

It is then easy to show, using an approximation like eq. (13) of ref. 4, that

$VAR(\eta) \propto 1+\eta^{-2}$

and that the above modification is appropriate. [SQRTW is the square root of the least squares weight.]

(4) If Δ_1 cannot be ignored, then DISCRETE can be run without modification with input:

Y(J) = $z(\tau_j)$

NOBASE = F

NONNEG = T

IWT = 2 [assuming $VAR(z) \propto z$]

However, if there is more than one exponential in $g(\tau)$, the extra exponentials due to the squaring in eq.(1) can make the separation of the exponentials difficult. The best policy is to do both types of analysis and compare.

D. NEW PROGRAMS TO BECOME AVAILABLE

Hopefully this year, as promised, [2,4] a general program will be available for analyzing data represented by an integral equation of the first kind:

$$y(\tau) = \int K(\lambda,\tau) f(\lambda) d\lambda \qquad (2)$$

or data represented by a (possibly unstable) system of linear algebraic equations. In eq. (2), $y(\tau)$ is the data, $K(\lambda,\tau)$ is known, and $f(\lambda)$ is to be estimated. The program is completely objective and automatic, and linear equality and inequality (e.g., nonnegativity) constraints on $f(\lambda)$ can be imposed.

Perhaps by next year, a general program is planned for analyzing data represented by

$$y(\tau) = \sum_{j=1}^{N_\lambda} \alpha_j f(\lambda_j,\tau) + \sum_{k=1}^{N_g} \beta_k g_k(\tau),$$

where N_g and the functional forms of $f(\lambda,\tau)$ and of the $g_k(\tau)$ are known and $\alpha_j,\lambda_j,\beta_k$, and N_λ are to be estimated. Therefore exponentials, but also convoluted exponentials, and other functions will be able to be analyzed.

If you will have interest in either of these programs, let me know:

S. W. Provencher
MPI f. biophys. Chemie
Postfach 2841
D-3400 Göttingen
West-Germany

R E F E R E N C E S

[1] H. Z. Cummins and P. N. Pusey, in Photon Correlation Spectroscopy and Velocimetry, H. Z. Cummins and E. R. Pike, eds. (Plenum, 1977), p. 164.

[2] S. W. Provencher, J. Hendrix, L. De Maeyer, and N. Paulussen, J. Chem. Phys. 69, 4273 (1978).

[3] E. Jakeman, E. R. Pike, and S. Swain, J. Phys. A: Gen. Phys. 4, 517 (1971).

[4] S. W. Provencher, Makromol. Chem. 180, 201 (1979).

DISCRETE Update 2

A. A BETTER REFERENCE AND NEW PROGRAMS

The enclosed reprint [1] contains a more definitive description of the methods used by DISCRETE and is a better literature reference than I or II on p. A(1).

There are two new programs that were discussed in [1] and Update 1. The program for analyzing data represented by an integral equation of the first kind [Eq (2) in Section D of Update 1] is now available. In addition to photon correlation spectroscopy and chemical kinetics (see references in [1]), it has also been applied to the estimation of globular protein secondary structure from circular dichroic spectra [2] and to the analysis of equatorial fiber diffraction data [3] and can be easily modified to handle many other problems. The second new program is for the analysis of sums of one-parameter functions (e.g., convoluted exponentials, as well as pure exponentials; see the last equation in Section D of Update 1). It should be available in 1981.

B. ALWAYS LOOK BACK FOR SOLUTIONS MISSING FROM THE FINAL SUMMARY

As already discussed in Update 1, there may be important possible solutions that are not printed at the end of the output in the final summary of results (see Section E.1). If this final summary does not list all NLAMMX solutions, then you should always check back through the final least squares analyses of the raw data (and, to be even safer, through the initial analyses of the raw data and then the transforms) to see if there are any solutions with a VARIANCE significantly less than that of the "BEST SOLUTION" in the final summary. Any such solutions are often important possibilities. They may have been rejected from the final summary of results because

(1) a λ is held at an upper or lower bound, i.e., $\lambda=\lambda_{min}$ or $\lambda=\lambda_{max}$; this is discussed in Section A of Update 1; or
(2) the normal-equations matrix is nearly singular, e.g., two λ's may be nearly equal.

Usually these conditions are a result of overfitting with too many exponentials. In these cases the VARIANCE is not significantly better than that of the BEST SOLUTION and this missing solution should be rejected anyway. A typical example is on page 5 of the output from the run with the test data, where (2) occurs with $\lambda_3 \simeq \lambda_4$.

Often (1) can occur with $\lambda=\lambda_{max}$ if your data is really a sum of convoluted exponentials and you have included data from the early part of the curve, where the convolution is still affecting the data. Several extra exponentials may be required to fit this distorted early part of the data, and the entire solution is usually distorted. Your only possibility with DISCRETE is to throw away more of the early data. However, it would be much better to analyze all the data with the correct model of convoluted exponentials using the program to become available in 1981.

If you are measuring the experimental curve by hand and you tilt the curve with respect to the horizontal baseline, then the tilted baseline can result in (1) with a $\lambda=\lambda_{min}$.

There are, however, cases where condition (1) or (2) occurs, but where there is a significant decrease in the VARIANCE and where the solution is meaningful. For example, as mentioned in Update 1, there may really be a very large λ with a large α and (1) can occur.

It also can happen that condition (2) occurs because there really are two closely spaced λ's. If both of the corresponding α's are of the same sign, then it is usually not possible to decide whether there are really two λ's or only one, because the VARIANCE has not decreased significantly. In this case, the solution with the two closely spaced λ's should be rejected in favor of the simpler one with only one λ. However, when the two α's are of

opposite sign, then it is often obvious that the two closely-spaced λ's are necessary despite condition (2). To see this, consider the following theoretical curve:

$$y(t) = \alpha_1 \exp[-(\lambda-\delta_\lambda)t] + \alpha_2 \exp[-(\lambda+\delta_\lambda)t], \quad (1)$$

$$\delta_\lambda << \lambda; \quad \alpha_1 > 0; \quad \alpha_2 < 0 \quad (2)$$

$$y(t) = \alpha_1 \exp(-\lambda t)[\exp(\delta_\lambda t) - \exp(-\delta_\lambda t)] + (\alpha_1+\alpha_2)\exp[-(\lambda+\delta_\lambda)t] \quad (3)$$

$$y(t) = 2(\alpha_1\delta_\lambda)te^{-\lambda t} + \text{terms of order } [\alpha_1(\delta_\lambda t)^3 e^{-\lambda t}] + (\alpha_1+\alpha_2)\exp[-(\lambda+\delta_\lambda)t] \quad (4)$$

The function $te^{-\lambda t}$ in Eq (4) has a peak at $t=\lambda^{-1}$, which is characteristic of curves with two closely-spaced λ's and with α's of opposite sign. They can by no means be well fit by a single exponential; the two closely-spaced λ's are necessary to fit the peak. DISCRETE is quite effective at finding these solutions, but the two closely-spaced λ's usually cause condition (2) because of the nearly singular normal-equations matrix. Therefore these solutions do not appear in the final summary of results and you must look for them back in the final least squares analyses of the raw data.

Note from Eq (4) that when $\delta_\lambda << \lambda$ and $\delta_\lambda t << 1$, that neither the individual α's nor δ_λ can be accurately determined. Their only significant effects on y(t) are as the product $\alpha_1\delta_\lambda$ in the first term on the right and as the sum $(\alpha_1+\alpha_2)$ in the last term on the right. Thus there can be a whole series of solutions, all fitting the data, with very large α's and closely-spaced λ's such that $\alpha_1\delta_\lambda$ and $(\alpha_1+\alpha_2)$ are the required constants. However, λ and $(\alpha_1+\alpha_2)$ are often quite accurately determined, and it can be useful to know that there are two closely-spaced λ's near λ with total amplitude $(\alpha_1+\alpha_2)$. Unfortunately, no plots of these solutions are made; this will be possible in the 1981 program. Your only guide is to see if the VARIANCE is significantly better than that of the BEST SOLUTION, especially if the plot, PUNC, etc. of the BEST SOLUTION indicate an inadequate fit.

C. CONSIDER ALTERNATIVE SOLUTIONS IN THE FINAL SUMMARY

The term NPHI was (somewhat arbitrarily) introduced in the selection criterion for the BEST SOLUTION to discriminate against solutions with nearly singular normal-equations matrices (see discussion of PNG and NPHI in Section E.1). This is fine if the resultant BEST SOLUTION adequately fits the data. (This you can judge from the plot of the residuals, PUNC, STD. DEV. OF FIT, etc., as discussed in Section E.1.) However, we have seen above that solutions with nearly singular normal-equations matrices are sometimes necessary to fit the data and that they may be meaningful, despite great uncertainties in the absolute values of the λ's or α's. Therefore it was suggested above that you look back in earlier parts of the output for solutions missing in the final summary of results. Similarly, if the adequacy of the fit with the BEST SOLUTION seems questionable, then you should also check to see if any of the other solutions in the final summary of results lead to a significantly better fit. To help you judge this, the PNG that would have been obtained by ignoring NPHI and performing a standard F-test [4] is printed after "UNCORRECTED PNG WOULD BE". For example, by this criterion the (correct) SECOND BEST SOLUTION on page 27 of the test-run output would have been preferred and the THIRD BEST SOLUTION considered a significant possibility. (In this case, however, comparisons of PUNC and STD. DEV. OF FIT do not seem to indicate that the fit of the two-component BEST SOLUTION is significantly worse than the three-component SECOND BEST, and you should therefore prefer the simpler two-component solution.)

Of course, you should never use these criteria blindly. They are based on assumptions (like independent normally-distributed noise and the complete lack of systematic measurement or modeling errors) that are practically never completely correct. Therefore, your considerations should also be based upon all the prior information that you have from other experiments and theory.

REFERENCES

[1] S.W. Provencher and R.H. Vogel, Math. Biosci. 50, 251 (1980).

[2] S.W. Provencher and J. Glöckner, Biochemistry, 20, 33 (1981).

[3] C. Nave, R.S. Brown, A.G. Fowler, J.E. Ladner, D.A. Marvin, S.W. Provencher, A. Tsugita, J. Armstrong, and R.N, Perham, J. Mol. Biol. 149, 675 (1981)

[4] T. Soderstrom, Internat. J. Control 26, 1 (1977).

DISCRETE Update 3

A. Address Correction Request

If necessary, please correct the address label used for this update, and send it to

Stephen W. Provencher
Max-Planck-Institut für biophysikalische Chemie
Postfach 2841
D-3400 Göttingen
Federal Republic of Germany.

B. Corrections to DISCRETE

All the conditions corrected in this update occur only rarely and would lead to aborts, not misleading results.

Please make the following replacements in **Version 1B**:

```
  200 IF (N.GT.2*NLAMMX+3 .AND. N.LE.NMAX) GO TO 205                    DAT10900
 5205 FORMAT (///4H N =,I5,37H IS NOT BETWEEN 2*NLAMMX+3 AND NMAX =,I5) DAT11000

  110 PLMTRY(J)=DMAX1(DBLE(LAMNMX(1,1)),                                EVR03500
     1                DMIN1(DBLE(LAMNMX(2,1)),PLAM(J)+Q*DELTAP(J)))     EVR03501

      PLMTRY(J)=DMAX1(DBLE(RBLOKB(1)),                                  VRF03800
     1                DMIN1(DBLE(RBLOKB(2)),PLAM(J)+Q*DELTAP(J)))       VRF03801
```

or make the following replacements in **Version 1A**:

```
  200 IF (N.GT.2*NLAMMX+3 .AND. N.LE.NMAX) GO TO 205                    DAT10700
 5205 FORMAT (///4H N =,I5,37H IS NOT BETWEEN 2*NLAMMX+3 AND NMAX =,I5) DAT10800

  110 PLMTRY(J)=AMAX1(LAMNMX(1,1),                                      EVR03000
     1                AMIN1(LAMNMX(2,1),PLAM(J)+Q*DELTAP(J)))           EVR03001

      PLMTRY(J)=AMAX1(RBLOKB(60),AMIN1(RBLOKB(61),PLAM(J)+Q*DELTAP(J))) VRF03400
```

Replace the lines with the same labels as those in columns 73–80 above. Note that the number of lines is increased by 2 in Version 1B and by 1 in Version 1A, because of continuation lines. For completeness, you should also change Version 1A or 1B to Version 2A or 2B, and April 1976 or March 1976 to December 1990 throughout the programs.

The replacements in DATAIO simply increase the minimum number of data points by 2 to avoid problems in the analysis of correlations in the residuals in case you are using an extremely small number of data points.

The replacements in EVAR and VARF prevent round-off error in large steps in the least-squares analyses from taking λ parameters out of bounds and causing an illegal subscript reference. This can occur if the range of decades in your T values exceeds the

number of significant figures in your REAL arithmetic, typically 7 decades. Data with such wide T ranges are becoming more common.

C. Inputting Unfavorable T Arrays Can Cause Arithmetic Overflows

DISCRETE is quite robust to strange T arrays, but inputting identical T values when they are among the smallest 5 can cause excessively wide allowed ranges of λ, and inputting 5 identical minimum values can cause an overflow.

A far more common problem occurs when the beginning of a data set is discarded, say, with $T = nr, (n+1)r, (n+2)r, \ldots$, where $n > 1$, i.e., a data set with equal spacing, r, in T, but with the first $(n-1)$ points missing. In this case, it is best to input a modified T array,

$$T' = T + \Delta. \tag{1}$$

A component $\alpha \exp(-\lambda T)$ in T will become

$$\alpha \exp[-\lambda(T' - \Delta)] = \alpha' \exp(-\lambda T'), \qquad \alpha' \equiv \alpha \exp(\lambda \Delta). \tag{2}$$

Thus, inputting the modified array T' results in the same λ values, and you can easily compute your original amplitudes as

$$\alpha = \alpha' \exp(-\lambda \Delta). \tag{3}$$

In the above example, you should use $\Delta = -(n-1)r$ so that $T' = r, 2r, 3r, \ldots$. With a spacing in T or T' of r, it is feasible to measure as fast a process as about $\lambda = 1/r$. Substituting this Δ and λ into Eq. (3), we have $\alpha = \alpha' \exp(n-1)$. If the first 100 points are missing, then α will be e^{100} times α', and overflow can occur when the original T array is used rather than T'.

In general, then, there is danger of overflow if your smallest T is at least several times greater than the smallest spacing between the T values. You should then use a Δ so that the smallest T' is approximately equal to this smallest spacing. DISCRETE actually computes this Δ and outputs it as DELTA with the transforms when you input PRPREL=T. However, it's usually clear what a reasonable value of Δ, if any, is needed. With REGINT=T, the only change to your input data is to add Δ to the original TSTART and TEND values.

D. Version 2 and Microcomputers

The changes in Section B result in Version 2 of DISCRETE. In addition, the magnetic tape format described in Section B of the users manual should be corrected from 800 bpi to 1600 bpi. Only ASCII tapes are now available.

Version 2B of DISCRETE is also available on 3.5-inch double-density (720 k-byte) and 5.25-inch high-density (1.2 M-byte) MS-DOS diskettes. In this case the test data are appended to the source code, so that there is only one file.

The test output from Version 1B is still sent out. The only significant differences are the version number and the date.

Most Fortran 77 compilers accept the Fortran 66 of DISCRETE without change. In rare cases, it may be necessary to declare Hollerith strings that I stored in INTEGER arrays explicitly as CHARACTER, but such changes are usually clearly indicated by the compilation diagnostics.

Extremely poor quality Fortran compilers are widespread among microcomputers, and errors in these caused aborts until a few years ago. However, these problems seem to have been slowly corrected, and DISCRETE is running without problem on more than 100 microcomputers. Because of limitations in many microcomputer operating systems, it is often necessary to break the code into a few segments for compilation and then to link the object files together.

E. Truncating Convoluted Data

The data truncation problem discussed in Section C of this update is often a result of discarding the beginning of the data because they are convoluted with a system response function. This is bad because information is lost and because most people tend to do this by eye and retain data that are still significantly distorted by the convolution. This can result in artifacts with large λ values. A package, SPLMOD [1,2], is available for analyzing multicomponent convoluted exponentials, but it does not seem to be fully appreciated that the response function must be very accurately determined, either directly or by reference measurements [2], regardless of the method used. Otherwise, the same type of large-λ artifacts can occur.

F. Always Look Back for Solutions Missing from the Final Summary

The main problem still seems to be that users only look at the final summary on the last page, and therefore miss important alternative solutions that did not meet the stringent standards for entry into the final summary. This is discussed in detail in Sections B and C of Update 2.

References

[1] S.W. Provencher & R.H. Vogel, "Regularization Techniques for Inverse Problems in Molecular Biology," in *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, P. Deuflhard and E. Hairer, eds. (Birkhäuser, Boston, 1983), pp. 304–319.

[2] R.W. Wijnaendts van Resandt, R.H. Vogel & S.W. Provencher, "Double beam fluorescence lifetime spectrometer with subnanosecond resolution: Application to aqueous tryptophan," Rev. Sci. Instr. **53**, 1392–1397 (1982).