

AD-A099 416

BOM CORP MONTEREY CA

F/G 9/2

HP9825 PROGRAMS USED WITH THE PMS DAS-32 TO ACCUMULATE AND ANAL--ETC(U)

APR 81 D E SPIEL

N00014-79-C-0088

UNCLASSIFIED

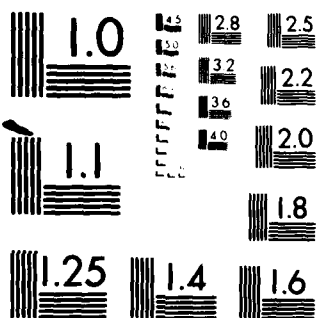
BDM/M-006-81

NL

1-1
A



END
DATE
FILMED
5 81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

LEVEL

12



P.O. BOX 2019 · 2600 GARDEN ROAD · MONTEREY, CALIFORNIA 93940 (408) 649-3880

AD A 099 416

AD A 099 416 COPY

DTIC
ELECTE
S MAY 28 1981 D
A

This document has been approved
for public release and sale; its
distribution is unlimited.

81 5 01 05



P. O. Box 2019
2600 Garden Rd.
Monterey, California 93940
Phone (408) 649-3880

1 April 1981

Report No. BDM/M-006-81

DTIC
ELECTE
MAY 28 1981
S D
A

HP9825 PROGRAMS USED WITH THE PMS DAS-32
TO ACCUMULATE AND ANALYZE AEROSOL DATA

D. E. Spiel
BDM/NPS
Monterey, California 93940

This document has been approved
for public release and sale; its
distribution is unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BDM/M-006-81 ✓	2. GOVT ACCESSION NO. AD-A099 426	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) HP9825 Programs used with the PMS DAS-32 to Accumulate and Analyze Aerosol Data	5. TYPE OF REPORT & PERIOD COVERED Technical Report, Nov-Dec '80	6. PERFORMING ORG. REPORT NUMBER BDM/M-006-81
7. AUTHOR(s) D. E. Spiel Project Engineer	8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0088 ✓	
9. PERFORMING ORGANIZATION NAME AND ADDRESS BDM Corporation - NPS Facility 2600 Garden Road Monterey, CA 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Order 315	
11. CONTROLLING OFFICE NAME AND ADDRESS Dean of Research, Code 012 Naval Postgraduate School Monterey, CA 93940	12. REPORT DATE 1 Apr 1981	13. NUMBER OF PAGES 37
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Mr. Robin Simpson, ONR Representative Stanford University, Room 165 Durond Aeronautics Building Stanford, CA 94305	15. SECURITY CLASS. (of this report) Unclassified	16. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of this Report) See Initial Distribution List, page 37		
18. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
19. SUPPLEMENTARY NOTES		
20. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
21. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report was prepared under Work Order No. 315 of Contract No. N00014-79-C-0088 in support of a U.S. Naval Postgraduate School (NPS) research project sponsored by the Office of Naval Research, Code 460. The work was done in support of the Environemtnal Physics Group at NPS under the direction of Professors G. E. Schacher and K. L. Davidson.		

DD FORM 1 JAN 78 1473
(Page 1)EDITION OF 1 NOV 66 IS OBSOLETE
S/N 0102-014-8001

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410 162

THE BDM CORPORATION

FOREWORD

This report was prepared under Work Order No. 315 of Contract No. N00014-C-0088 in support of a U.S. Naval Postgraduate School (NPS) research project sponsored by the Office of Naval Research, Code 460. The work was done in support of the Environmental Physics Group at NPS under the direction of Professors G. E. Schacher and K. L. Davidson.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
<i>File on file</i>	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
<i>A</i>	

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	iii
TABLE OF CONTENTS	v
ABSTRACT	vii
A. INTRODUCTION	1
B. PROGRAMS	2
1. Data Acquisition Program	2
2. Plot of $\log(dN/dr)$ vs. $\log(r)$	8
3. Plot $\Delta(\log(dN/dr))$ vs. $\log(r)$	11
4. Tabulation of $(\log(dN/dr))$ vs. r	14
5. Plot $\log(\Delta(dN/dr))$ vs. $\log(r)$	17
6. Tabulation of $\log(\Delta(dN/dr))$ vs. r	21
7. Integration of dN/dr over r from 0.2 to 7 microns	24
8. Integration of dN/dr over r from 1.5 to 7 microns	26
9. Plot Fractional Change in dN/dr	28
10. Plot Normalized Fractional Change in dN/dr	31
11. Tabulation of Aerosol Decay Constant	34
DISTRIBUTION LIST	35

ABSTRACT

HP9825 computer programs for the acquisition and analysis of aerosol data are listed. The programs were written to be used with a PMS system in wave tank experiments, but can be used in other applications.

THE BDM CORPORATION

A. INTRODUCTION

This report briefly describes, defines the matrices used in, and lists a series of programs for the HP9825 used to accumulate and analyze aerosol data taken in conjunction with Particle Measuring Systems, Inc., (PMS) aerosol instruments. The programs were written specifically for use in wave tank experiments, but have a wider application.

The PMS instruments consist of a model DAS-32 data acquisition system and models ASASP and CSASP aerosol probes which, between them, scan the aerosol spectrum from 0.1 to 15 micrometers radius.

In addition to a data acquisition program, programs which integrate dN/dr spectra over various radius intervals are given. The remainder of the analysis programs are used to examine differences in spectra taken at different times and to look at these differences from several perspectives.

The programs described are:

- (1) Data acquisition
- (2) Plots $\log(dN/dr)$ vs. $\log(r)$
- (3) Plots $\Delta(\log(dN/dr))$ vs. $\log(r)$
- (4) Tabulates $\Delta(\log(dN/dr))$ vs. r
- (5) Plots $\log(\Delta(dN/dr))$ vs. $\log(r)$
- (6) Tabulates $\log(\Delta(dN/dr))$ vs. r
- (7) Integrates dN/dr over r from 0.2, 0.3, 0.4, 0.6, 0.8 and 1 to 7 microns.
- (8) Integrates dN/dr over r from 1.5, 2, 2.5, 3, 3.5 and 4 to 7 microns.
- (9) Plots $(1/dN_1)(dN_2 - dN_1)$, the fractional change in dN/dr , vs. $\log(r)$.
- (10) Plots the fractional change in dN/dr normalized by its peak value.
- (11) Tabulates the aerosol density decay constant for various r .

THE BDM CORPORATION

B. PROGRAMS

1. Data Acquisition Program

This program provides the interfacing between the HP9825 computer and the PMS system. It accumulates data for a time period specified by the user and outputs it to a printer and a cassette tape. The data output to the printer is in both tabulated and plotted form. The tabulated data consists of raw counts vs. radius, dV/dr vs. radius and, at the option of the user, a table of the radius bin centers. The program calculates a polynomial fit, whose order is user specified (up to 10), of $\log(dV/dr)$ vs. $\log(r)$. This fit is plotted along with the individual data points (with some exclusions) by the HP9871A printer. A sample output is shown in Figure 1.

List of matrices:

<u>Name</u>	<u>Function</u>
A[*]	Accumulates sums used to calculate the polynomial fit.
B[*]	Denotes which printer symbols to be used for the six (6) radius ranges in the plot, except B[7] and B[8] used to extrapolate the fit.
C[*]	Cassette tape file bookkeeping, see below.
D	Current file number.
E[*]	Average radius within a bin. There are six ranges with 15 bins each.
F[*]	Accumulates sums used to calculate the polynomial fit.
G[*]	Coefficients of the polynomial fit.
H[*]	PMS analog housekeeping module outputs.
M[*]	Used in formulating extrapolations to polynomial curve.
N[*]	Radii range counter. Essentially keeps track of the total time the PMS was examining a given radii range.
O[*]	dN/dr and, later in the program, dV/dr .
P[*]	Sum of raw counts in each radius bin arranged in order of increasing radius.

THE BDM CORPORATION

R[*] Defines radii bin edges.
S[*] Sum of raw counts in each radius bin in the order specified by the DAS-32.
T[*] dN/dr
X[*] Defines which radius bins in each range are to be excluded from the polynomial fit calculations and the plot.
Y[*] Date and time, see below.

The components of C[*] are:

C[1] Number of files available on track 0 of the tape.
C[2] Number of files available on track 1 of the tape.
C[3] Tape number.

The components of Y[*] are:

Y[1] Year
Y[2] Month
Y[3] Day
Y[4] Hour
Y[5] Minute
Y[6] Second

Strings employed for outputting data, in split precision, to the tape are:

A\$ Contains T[*], H[*], Y[*], X[*] and T, the averaging time.

Special function keys are employed as follows:

<u>Key</u>	<u>Action</u>	<u>Purpose</u>
0	* "cont formats"	Restarts programs without redefining matrices or loading keys
4	* sfg4	Branches program so that other special function keys may be set and acted upon.
5	* sfg5	Change order of polynomial.
6	* sfg6	Change averaging period.
7	* sfg7	Suppress all printing and plotting.
8	* cfg7	Restart printing and plotting.
9	* sfg9	Print radius bin center values
10	* sfg10	Mark a data tape.

THE BDM CORPORATION

Program Listing:

```

0: "LAB-32 MACROBASIC, DIRECT PC 8 IS BASIC, TABULATE & PLOT 8v/3r":
1: dir AS[130],CT[36],C[1300];buf "0",35,3
2: dir Y[8],I[5,13],I[5],F[5,13],F[16],F[5,13],F[6],F[6,13],F[6]
3: "in F[6,13],I[5,13],F[16],F[10,13],F[10],K[6],C[3]
4: LK 1;35+I[1];111+2[2];43+I[3];6+L[4];42+I[5];64+2[6]
5: 715+1
6: sto 2,r:rev " ",1
7: "format":int 6,10x,c
8: int 1,f2.0
9: int 2,cl,E2,z
10: fmt 3,"date",f3,"/",f2,"/",f2," "time",f4,":",f2," at end of run"
11: int 4,/, "Averaging time = ",f4.1," minutes"
12: int 5,"Probe voltage A = ",f5.3," volts"
13: int 6,cl0.2,z
14: int 7,"Time =",f3," File #",f3," Event #"
15: int 8,"Polynomial of order ",f2,z
16: int 9,cl5.7,z
17: "reset":dis "Set any flags,continue.";sto
18: if fld10;sto "tanark"
19: if fld5;int "Order of polynomial?",C;irm 2
20: 7+C
21: C+1+I;run A[I],G[I],F[I,I]
22: if fld6;int "Averaging time?",I;int(1.5I)+Z
23: dis "Insert data time,continue.";sto
24: ldf 0,0,C[*]
25: dis "Printer on, set form, continue.";sto
26: wto 8,27,59,27,84,32,32,32,27,77,27,76,15,0,14
27: wto 1,27,79,15,18,0,32
28: if not fld6;27+I;int(1.5I)+3
29: 3+K[1]+K[2]+K[3];1+K[5]+K[6];15+K[4]
30: .0345+K[1,1];.0675+K[1,2];.0365+K[1,3];.064+K[1,4];.065+K[1,5]
31: .102+K[1,6];.1065+K[1,7];.111+K[1,8];.1165+K[1,9];.12+K[1,10]
32: for J=11 to 15:K[J],J-1+.035+K[1,J];next J
33: for J=1 to 15: (.23+.025(J-1))/2+K[2,J]
34: .2+.02(J-1)+K[3,J];.3+.03(J-1)+K[4,J];.25+.25(J-1)+K[5,J]
35: if fld5,11>1;.767*K[5,J]+.233+K[5,J]
36: 1+(J-1)+K[6,J];.767*K[5,J]+.233+K[5,J];next J
37: for I=1 to 6;for J=1 to 15:(K[1,J+1]+K[1,J])/2+K[1,J];next J;next I
38: "start":int(1.5I)+5;I+K;fld 0:for 1,2,3;int 2,10x,c;if fld7;I+1
39: int 1;fld 1;if I=1;sto 7;I+1
40: "date":on "0";dir 2," ",1234;K+1+K;fld 2
41: for "date"dir 1," ",on "A";int run("0")=1234
42: for "date"dir 1," ",on "A";int run("0")=1234
43: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
44: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
45: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
46: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
47: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
48: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
49: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
50: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
51: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
52: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
53: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
54: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
55: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
56: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
57: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
58: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
59: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
60: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
61: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
62: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
63: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
64: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
65: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
66: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
67: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
68: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
69: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
70: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
71: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
72: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
73: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
74: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
75: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
76: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
77: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
78: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
79: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
80: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
81: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
82: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
83: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
84: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
85: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
86: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
87: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
88: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
89: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
90: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
91: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
92: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
93: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
94: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
95: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
96: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
97: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
98: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
99: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234
100: dir 1," ",on "date"dir 1," ",on "A";int run("0")=1234

```

THE BDM CORPORATION

```

43: if J3: int (Y/100)→X[2J-1]; 100frc (Y/100)→Y[2J]; if Y[1]≠0; goto "Data"
49: if J5: goto "next"
50: (Y+J[4]→J[4]; goto "next"
51: "bin": if flg1; J+1→J; jmp 3
52: if X<2; J→J+2; jmp 2
53: J→J+3
54: J+6(S,1)→S(S,1)
55: "next": next I; 1+N(S)→N(S); next J; and 2; 1#→1; if flg2; goto "loop"
56: if X0; goto "Data"
57: C[1]+C[2]→.
58: if J17; goto "File"; 0, "Print successful."; jmp 2
59: goto "Accessing file"; 0, "Access no.,"; .
60: if X>0; call "Access"
61: for I=1 to 4; 100(I)→N(I); next I; 30.44(S)→N(2); 30.44(S)→N(6)
62: for I=1 to 4; for J=1 to 15; if S(I,J)=0; 1→S(I,J)
63: next J; next I; for J=1 to 15; if S(5,J)=0; 1→S(5,J)
64: next J
65: for J=1 to 15; S(1,J)/N(1)→T(1,J); S(2,J)/N(2)→T(2,J); S(3,J)/N(3)→T(2,J)
66: S(1,J)/N(4)→T(1,J); S(5,J)/N(5)→T(5,J); S(6,J)/N(6)→T(5,J)
67: S(1,J)+S(4,J); S(2,J)+S(3,J); S(3,J)+S(2,J); T(4,J)+S(1,J); S(5,J)+S(5,J)
68: S(5,J)+S(5,J)
69: T(1)/100→N(1); next J; S(15)/100→N(15)
70: for I=1 to 5
71: for J=1 to 15; (T(1,J)/N(1)+1)→T(1,J)+T(1,J)+T(1,J)
72: next J; next I
73: for I=1 to 3; for J=1 to 3; 0→T(I,J); next J; next I
74: T(X[2],1)+T(5,1); for J=1 to 15; 1→T(4,J); next J
75: goto "next"
76: "01": goto 1; 1→1
77: "02": rank(0 or 1("1"), 255)→r5; and(1-3, 15)+1000→r3; 4→r2
78: and(2 or 0("1"), 255)→r4; rank(r3, 15)+1000→r4; 4→r1; 1000→r1; rot
79: "next": and 4; for I=1 to 5; 0→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
80: for J=1 to 15; 1→J; if I=0 and 3J; 1→r5
81: if J<15; jmp 1
82: if I>1 and 15; goto "next"
83: if I=1; →T(1,J); goto "1"
84: if I=0 and 15 and 15→r7; goto (r7/r3)→r1; (r7/r3)→r7; 1→r5; goto "2"
85: 1.5→r7; goto "1"
86: if T(1,J)=0 or r5=0; goto "next"
87: T→T*(1,1)3*T(1,1)+T(1,1)
88: if r5=0; goto "extra"
89: if T(1,J)≠0 and r5≠0; goto "extra"
90: and(T(1,J)→r1; 100(T(1,J)→r1)
91: if I=1 and 3J; goto "2"
92: if I=0; goto "2"
93: goto "2"
94: "extra": r1→r0; r7/5(T(1,J)+r7; r5=0(T(1,J)+r4; goto "next"
95: "2": r0→X→r0; r1→r1; r2+1→r2; r3+X→r3; r4+0→r4; if r5=0; goto "2"
96: X→T(1,J)+T(1,1)
97: "1": (r3r2-r1r1)/(r0r2-r1r1)→T(2)
98: (r1r1-r2r1)/(r0r2-r1r1)+T(1,1)+T(2)*→.

```

THE BDM CORPORATION

```

90: "C":for K=0 to C;C+1-K+K;C+K[F,R]→F[F,R]
100: P←A[F]→A[F];P*V→P;C←C
101: if K=0;C←[C-1,R]→F[C-1,R]
102: C*3;next K
103: "aku":next C;next I
104: for I=1 to C+1
105: for K=1 to int((C+1)/2)
106: if I=1 and I+K≤C+2 and I-1>0;F[I,I]→F[I-1,I+1]→F[I+K,I-1]
107: if I+K≤C+1 and I-1+1>0;F[I,I+1]→F[I+K,I-1+1]
108: if I+K≤C and I-1>0;F[I,I+1]→F[I-1,I+K+1]
109: next K;next I;Iav →C;ret W→C;if ILS7;goto "alt"
110: "out":wrt "1.7",Z[3],
111: wrt "1.3",Z[3],X[3],Z[1],100Y[4]+Y[5],Z[5]
112: wrt "1.4",F;wrt "1.5",R[1];vds 7,10,13
113: vds 1,"C"/dr
114: vds 9,10,13
115: for J=1 to 15
116: for I=2 to 5 by 3
117: wrt "1.6",F[I,J]
118: next I;vds 1,10,13;next J;vds 9,10,13
119: if not flg0;irw 3
120: wrt Y,"radii";vds 1,10,13;for J=1 to 15;for I=1 to 5;wrt "1.6",Z[I,J]
121: next I;vds 1,10,13;next J;vds 1,10,13;irw 9
122: vds 1,17,55,int(-3.75),int(-245),5,15,"lat(radius)"
123: vds 1,17,55,int(-1.55),int(-129),7,32,"lat(Lx/Lr)"
124: "alt":I←I+1;ret 1,5,Z[5];if I-1>C[1];ccw 1:I←C[1]-2
125: ILS7;goto "radio print"
126: -1+3;-1+
127: vds 1,17,55,int(15/4),int(1.1),,
128: if C[6]≠0 and I=0;ret 1,"1",15,5,5;ret 1,"1";vds 1
129: -1+3;-1+
130: if (C[4]-0)<0.95;goto -1
131: vds 1,17,55,5,5,int(1.55),int(55)
132: if C[6]≠0 and I=0;ret 1,"1",15,5,5;goto -2
133: vds 1,"1"
134: if (C[4]-0)<0.1;goto -2
135: for I=0 to 1 by 2
136: for J=1 to 15
137: if I=1;R←F or F[I,J]←0;vds "1.8"
138: lat([I,J])←lat([I,J])
139: vds 1,17,55,int(15/4),lat([I,J]),int(1/4),int(-5)
140: vds 1,11
141: "1.9":next J;next I;irw 2
142: vds 1,17,55,int(15[7]/4),int(200[7]),int(30[8]/2),int(C[4]),"C"
143: "thr":vds 1;for I=1 to 5;for J=1 to 15;P←15[1]-15
144: lat([I,J])←S[1][J],C[4];next J;next I
145: →C;ret 1;C←15
146: ret ([I]+S[3][I]+30[4]+K);1+K;goto 138;goto "alt"
147: vds ([I])←S[3][I]+30[4]+K;1+K;goto 138;goto "thr"
148: vds ([I]-1)←S[3][I]+K;352+4K
149: vds ([I])←S[3][I]+K;353+K;264+4K;1+K→C

```

THE BDM CORPORATION

```

150: "NEXT":next I
151: fts (I)→A$[457,460]
152: -I→X;0→Y;goto "gulo"
153: "crv":0→Y;if X>.8;goto "belch"
154: for I=1 to C+1;YX+C[I]→Y;next I
155: Y→I[4];jmp 3
156: "belch":if X>1.1; if M[2]>0;I[6]→Y;jmp 2
157: M[4]+M[2]*(X-.3)→Y→I[6]
158: if Y<-4 or Y>5;jmp 4
159: Y+6/96→Y
160: with M,27,65,int(15X/4),int(240X),int(3Y/2),int(56Y)
161: wto M,"."
162: if (X+1/120→X)<1.5;goto "crv"
163: "gulo":wto M,12,13
164: of B,A$,93;trk 6;trf 0
165: "Flags":in 1104;goto "reset"
166: if flag=0: "oly order=","",0→P;end """,0;C+1→Y;wto M[P],I[3],I[6],I
167: if 116: for "version: flag=","",0→P;end """,0;int(1.57)→
168: for "rams continue to begin";out →to "start"
169: "tamern":for i=1 to 6;con;wait 300;next I
170: go "begin new character, continue.";con
171: for "Are you sure?Continue.";con
172: set "Are you sure?",010]
173: trk 6;trf 10;trf 30;trf 100→C[I+1]→C[I];I→0
174: for 0,0,0;trf 1;trf 100,150;trf 100;trf 0
175: if 1100;or 110;goto "reset"
176: ror
*5909

```

THE BDM CORPORATION

2. Plot of $\log(dN/dr)$ vs. $\log(r)$

This program retrieves the data stored on the cassette tape by the data acquisition program and produces a plot of $\log(dN/dr)$ vs. $\log(r)$. The plot consists of the individual data points, less those excluded, and a curve generated by a polynomial fit plus end point extrapolations. There is no tabulation of data.

The matrices used in this program serve the same function as in the data acquisition program.

The following is a listing of this program:

```
0: "BDM-32:PLDNG LOG(dN/dr) PLOT OVER IN FILES":
1: dim A[1:10],S[1:10]
2: dim F[1:10],G[1:10],F[1:10],G[1:10]
3: dim E[1:10],S[1:10],E[1:10]
4: dim P[1:10],A[1:10],F[1:10],G[1:10],S[1:10]
5: 35→C[1];111→C[2];43→C[3];0→C[4];42→C[5];64→C[6]
6: 715→I;dev "D",I
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,c1,f3.0,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0
11: fmt 4,"Averaging time = ",f3.0," minutes"
12: fmt 6,c10.2,z
13: fmt 7,"Iarc = ",f3.0," File #",f3.0," Event ="
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,c15.7,z
16: 7→C
17: C+1→I;for A[I],S[I],F[I],G[I]
18: dso "Insert data tape,continue";sto
19: ldf 0,0,C*I
20: crt "Track",I
21: dso "Printer on, set form,continue";sto
22: wth 1,27,50,27,84,32,32,32,32,27,77,27,76,15,0,14
23: wth 1,27,72,1,46,5,32
24: 1→X[1];X[2]→X[3];1→X[5];X[6]→X[4];15→X[4]
25: .0345→F[1,1];.0875→F[1,2];.0905→F[1,3];.094→F[1,4];.098→F[1,5]
26: .102→F[1,6];.1065→F[1,7];.111→F[1,8];.1155→F[1,9];.12→F[1,10]
27: for J=11 to 16:R[1,J-1]+.005→R[1,J];next J
28: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
29: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
30: if F[5,J]>1;.767→R[5,J]+.233→R[5,J]
31: 1+(J-1)→R[6,J];.767→R[6,J]+.233→R[6,J];next J
32: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→E[I,J];next J;next I
```


THE BDM CORPORATION

```

33: "cycle":ent "Begin with file",A;ent "End with file",B
34: for D=A to B;fmt 0,10x,z;cll 'read tape'
35: for I=1 to 6
36: for J=1 to 15;T[I,J]←O[I,J];next J;next I
37: for I=1 to 3;for J=1 to 3;O←I[I,J];next J;next I
38: O←O[5,1]←O[6,1];for J=1 to 15;O←O[4,J];next J
39: "mat":inn A,2;for I=1 to 5;O←r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
40: for J=1 to 16;I←3;if I=6 and J=1;I←r5
41: if J<16;inn 5
42: if I>1 and I<6;gto "next"
43: if I=1;←1.5×X;gto "1"
44: if I=6 and r9≠0 and r7≠0;log(r3/r9)←X;log(r7/r9)←P;I←r5;gto "3"
45: 1.5×X;gto "1"
46: if O[I,J]=0 and r5=0;gto "next"
47: if r9=0;gto "extrap"
48: if O[I,J]=0 and r5=1;gto "extrap"
49: log(O[I,J])←P;log(E[I,J])←X
50: if I=1 and J<9;gto "3"
51: if I=6;gto "2"
52: gto "3"
53: "extrap":r9←1+r9;r7←O[I,J]+r7;r8←E[I,J]+r8;gto "next"
54: "3":r0←X←r0;r1←X←r1;r2←1+r2;r3←P←r3;r4←P←r4;if r5=0;gto "2"
55: X←P[7];P←P[8];1.5×X
56: "1":(r3r2-r4r1)/(r0r2-r1r1)←P[2]
57: (r0r4-r3r1)/(r0r2-r1r1)←P[1];P[1]←X×P[2]←P
58: "2":for K=0 to C;O←1←X←P;G←P[P,K]←P[P,K]
59: P←A[P]←A[P];P←X←P;O←G
60: if K<C;G←[P-1,K]←P[P-1,P]
61: G←X←G;next K
62: "next":next J;next I
63: for I=1 to C+1
64: for K=1 to int((C+1)/2)
65: if I≠1 and I+K<C+2 and I-K>0;P[I,I]←P[I-K,I+K]←P[I+K,I-K]
66: if I+K=C+1 and I-K+1>0;P[I,I+1]←P[I+K,I-K+1]
67: if I+K=C and I-K>0;P[I,I+1]←P[I-K,I+K+1]
68: next K;next I;inv P←P;mat PA←C;if flq7;gto "olt"
69: "out":wrt "1.7",C[3],D
70: wrt "1.3",Y[2],Y[3],Y[4],Y[5]
71: wrt "1.4",P;wth M,10,13
72: gto "no dn/dr"
73: wth M,"dn/dr";wth M,10,13;for J=1 to 15;for I=1 to 6
74: wrt "1.6",P[I,J];next I;wth M,10,13;next J;wth M,10,13
75: "no dn/dr":
76: wth M,27,65,int(-1.85),int(-120),7,32,"log(dN/dr)"
77: wth M,27,65,1,56,7,32;wrt "1.8",C;5.01-1/6×Y
78: for I=1 to C+1;wth M,27,65,1,56,int(3Y/2),int(96Y)
79: wrt "1.9",J[I];Y←1/6×Y;next I
80: wth M,27,65,int(-3.75),int(-240),0,16,"log(radius)"

```

THE BDM CORPORATION

```

81: "plt":if flq7;gto "skip print"
82: -1+X;-1+Y
83: wtb M,27,65,int(15X/4),int(240X),0,0
84: if X#0 and Xmod1=0;wtb M,"I",10,8,8,8;wrt "1.1",X;gto +2
85: wtb M,"-"
86: if (X+.05*X)<2.05;gto -3
87: wtb M,27,65,0,0,int(1.5Y),int(96Y)
88: if Y#0 and Ymod1=0;wrt "1.2","-",Y;gto +2
89: wtb M,"I"
90: if (Y+.1*Y)<5.1;gto -3
91: for I=1 to 6
92: for J=1 to 15
93: if E[I,J]=0 or O[I,J]=0;gto "NEXT"
94: log(E[I,J])→X;log(O[I,J])→Y
95: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
96: wtb M,"I"
97: "NEXT":next J;next I;if r5=0;jno 2
98: wtb M,27,65,int(15X[7]/4),int(240B[7]),int(3B[8]/2),int(96Z[3]),"c"
99: gto "skip print"
100: "read tape":trk W;ldf D,AS,ES
101: for I=1 to 6;for J=1 to 15;J+15I-16→K
102: stf(AS[4K+1,4K+4])→F[I,J];next J;next I
103: 0→K;for I=1 to 16
104: stf(AS[361+4K,364+4K])→I[I];1+K→K;if I>5;gto "lex1"
105: stf(AS[361+4K,364+4K])→Y[I];1+K→K;if I>3;gto "next1"
106: itf(AS[361+4K,362+4K])→X[2I-1]
107: itf(AS[363+4K,364+4K])→X[2I];1+K→K
108: "next1":next I
109: stf(AS[457,460])→T
110: ret
111: "skip print":-1+X;0+Y
112: "crv":0+Y;if X>.3;gto "selch"
113: for I=1 to C+1;YX+C[I]→Y;next I
114: Y→I[4];inc 3
115: "selch":if Y>1.1;if I[2]>0;I[6]→Y;inc 2
116: I[4]+I[2]*(X-.3)→Y+I[6]
117: if Y<-4 or Y>5;inc 4
118: Y+6/96→Y
119: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
120: wtb M,"."
121: if (X+1/120*X)<1.5;gto "crv"
122: wtb M,12,13
123: next D
124: gto "cycle"
125: end
*554

```

THE BDM CORPORATION

3. Plot $\Delta(\log(dN/dr))$ vs. $\log(r)$.

This program retrieves data from the cassette tape for two specified files, generates a polynomial fit to the data for each file, in log space, and plots the difference between the fits.

The matrices used in this program serve the same function as in the data acquisition program except that certain of them have been increased by one dimension in order to accommodate two data sets. $K[*]$ are the coefficients of the polynomial fit to the second set of data.

A program listing follows:

```
0: "DMS-32: PLOTS DELTA LOG(DN/DR) FROM DATA ON FILE":
1: dim AS[2,460], BS[460], CS[26], D[2], E[6,15], FS[25], G[2]
2: dim Y[2,6], X[6], I[2,6,15], U[16], O[2,6,15]
3: dim S[3], S[6,15], D, C[3], A[10]
4: dim D[6,16], N[6,16], A[16], F[10,10], G[10], I[9], M[6]
5: 35→S[1]; 111→S[2]; 43→S[3]; 1→S[4]; 42→S[5]; 64→S[6]
6: 715→I; dev "4", A
7: Ent 0, 10x, z
8: Ent 1, F3.0
9: Ent 2, c1, F3.1, z
10: Ent 3, "Date", F3.0, "/", F2.0, "/", F2.0, " Time", F3.0, ":", F2.0, z
11: Ent 4, "Averaging time = ", F2.0, " and ", F2.0, " minutes"
12: Ent 6, c10.2, z
13: Ent 7, "Tape #", F3.0, " Files", F3.0, " and", F3.0, " Event ", c25
14: Ent 8, "Polynomial of order ", F2.0, z
15: Ent 9, c15.7, z
16: 7→C
17: C+1→I; rdm A[I], S[I], K[I], F[I, I]
18: also "Insert data tape, continue"; sto
19: Ent "Tape #", C[3]; Ent "Track", J
20: also "Printer on, set form, continue"; sto
21: wtp 1, 27, 69, 27, 64, 32, 32, 32, 32, 27, 77, 27, 76, 15, 0, 14
22: wtb 1, 27, 79, 4, 48, 6, 32
23: 3→X[1]→X[2]→X[3]; 1→X[5]→X[6]; 15→X[4]
24: .0845→R[1,1]; .0875→R[1,2]; .0905→R[1,3]; .034→R[1,4]; .078→R[1,5]
25: .192→R[1,6]; .1055→R[1,7]; .111→R[1,8]; .1155→R[1,9]; .12→R[1,10]
26: for J=11 to 16; R[1, J-1]+.005→R[1, J]; next J
27: for J=1 to 16; (.23+.025(J-1))/2→R[2, J]
28: .2+.02(J-1)→R[3, J]; .3+.06(J-1)→R[4, J]; .25+.25(J-1)→R[5, J]
29: if R[5, 1]>1; .767→R[5, J]+.233→R[5, J]
30: 1+(7-1)→R[6, J]; .767→R[6, J]+.233→R[6, J]; next J
31: for I=1 to 6; for J=1 to 15; (R[1, J+1]+R[1, J])/2→L[I, J]; next J; next I
32: "start"; Ent "Event", CS; Ent "Background file", J[1]; Ent "Event file", J[2]
```

THE BDM CORPORATION

```

33: trk R;for D=1 to 2;for J[D],BS,ES;BS+AS[D];call 'sort';next D
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;P[D,I,J]→O[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;O→C[D,I,J];next J;next I
37: O→C[D,5,1]→C[D,6,1];for J=1 to 15;O→C[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;C[2,I,J]→O[1,I,J]→S[I,J]
39: if S[I,J]<0;O→S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ina A,F;for I=1 to 6;O→r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
42: for J=1 to 16;1→3; if I=6 and J#1;1→r6
43: if J<15;inc 5
44: if I>1 and I<6;goto "next"
45: if I=1;-1.5→X;goto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)→X;log(r7/r9)→P;1→r5;goto "3"
47: 1.5→X;goto "1"
48: if O[D,I,J]=0 and r6=0;goto "next"
49: if r9#0;goto "extrao"
50: if O[D,I,J]=0 and r6=1;goto "extrao"
51: log(O[D,I,J])→P;log(E[I,J])→X
52: if I=1 and J<9;goto "3"
53: if I#6;goto "2"
54: goto "3"
55: "extrao":r9+1→r9;r7+O[D,I,J]→r7;r8+E[I,J]→r8;goto "next"
56: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;goto "2"
57: X→B[7];P→B[3];1.5→X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)→M[2]
59: (r0r4-r3r1)/(r0r2-r1r1)→M[1];M[1]+X→M[2]→P
60: "2":for K=0 to C;C+1→K→R;G+E[R,K]→E[R,P]
61: P+A[R]→A[R];P→X→P;EX→3
62: if K#C;G+E[P-1,R]→E[P-1,R]
63: C→X→G;next K
64: "next":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;E[I,I]→E[I-K,I+K]→E[I+K,I-K]
68: if I+K=C+1 and I-K+1>0;E[I,I+1]→E[I+K,I-K+1]
69: if I+K=C and I-K>0;E[I,I+1]→E[I-K,I+K+1]
70: next K;next I;inv P→P;if D=1;mat P→C;inc 2
71: mat P→K
72: next D;para K→G→3
73: "out":wrt "1.7",C[3],J[2],J[1],CS
74: Y[2,4]→Y[1,4]→2;Y[2,5]→Y[1,5]→60→Z→Z
75: wrt "4.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: wrt 5," GT=",F3.0," minutes":wrt "1.5",Z
77: wrt "1.4",B[2],B[1];wrt 1,10,13
78: rto "no in/cr"
79: wrt 1,"a/cr":wrt 1,10,13;for J=1 to 15;for I=1 to 6

```

THE BDM CORPORATION

```

80: wrt "M.5",P[I,J];next I;wtb M,10,13;next J;wtb M,10,13
81: "no dn/dr":
82: wtb M,27,65,-4,0,7,32,"delta[log(dN/dr)]"
83: wtb M,27,65,1,56,7,32;wrt "M.8",C;5.01-1/6+Y
84: for I=1 to C+1;wtb M,27,65,1,56,int(3Y/2),int(96Y)
85: wrt "M.9",G[I];Y-1/5+Y;next I
86: wtb M,27,65,-4,-210,-1,-16,"log(radius)"
87: "olt":if flq7;gto "skip print"
88: -1+X;-4+Y
89: wtb M,27,65,int(15X/4),int(240X),0,0
90: if X#0 and Xmod1=0;wto 4,"|",10,8,3,3;wrt "1.1",X;gto +2
91: wtb M,"-"
92: if (X+.05+X)<2.05;gto -3
93: wtb M,27,65,0,0,int(1.5Y),int(96Y)
94: if Y#0 and Ymod1=0;wrt "1.2","-",.5*Y;gto +2
95: wtb M,"|"
96: if (Y+.1+Y)<5.1;gto -3
97: gto "skip print"
98: "sort":for I=1 to 6;for J=1 to 15;J+15I-16+K
99: str (AS[D,4K+1,4K+4]) +I[D,I,J];next J;next I
100: 0+K;for I=1 to 16
101: str (AS[D,361+4K,364+4K]) +I[I];1+K+K; if I>5;gto "Next"
102: str (AS[D,361+4K,364+4K]) +Y[D,I];1+K+K; if I>3;gto "Next"
103: str (AS[D,361+4K,362+4K]) +X[2I-1]
104: str (AS[D,363+4K,364+4K]) +X[2I];1+K+K
105: "Next":next I
106: str (AS[D,457,460]) +Z[D]
107: ret
108: "skip print":-1+X;0+Y
109: "crv":0+Y
110: for I=1 to C+1;YX+G[I]+Y;next I
111: if Y<-4 or Y>5;imp 1
112: 2*(Y+G/96)+Y
113: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
114: wtb M,"-"
115: "help":if (X+1/120+X)<.85;gto "crv"
116: wtb M,12,13
117: gto "start"
118: end
*30256

```

THE BDM CORPORATION

4. Tabulation of $\Delta(\log(dN/dr))$ vs. r :

This program retrieves data from the cassette tape for two specified files and tabulates differences between them in two formats. In the first, a bin by bin difference between $\log(dN/dr)$ for the two files is tabulated vs. r . In the second, a tabulation of the differences between polynomial fits (as described in the preceding program) are given. The matrices serve the same functions as described above.

A program listing follows:

```

0: "DAS-32: INITIATES DELIA(LOC(J)/K) VS. F AND THE VALUES OF THE":
1: "POLYNOMIAL F1 TO THIS FUNCTION FOR VARIOUS PATH":
2: dim AS(2,40),BS(40),CS(35),J(2),S(5,15),DS(25),Z(2)
3: dim Y(2,3),X(5),A(2,5,15),H(16),I(2,5,15)
4: dim F(2),G(5,15),O(2,3),K(16),LS(25)
5: dim D(5,15),F(5,15),A(10),F(10,10),S(10),I(9),T(5)
6: " " "CS
7: 701->J;lev "1",1
8: fmt 6,10x,z
9: fmt 1,f3.0
10: fmt 2,c1,f4.0,z
11: fmt 3,"date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
12: fmt 5,c11.2,z
13: fmt 7,"lane #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 9,c15.7,z
15: 7->C
16: C+1->I;ret A(I),G(I),K(I),F(I,I)
17: dco "Insert data trace,continue";sto
18: ent "Face #",C(3);ent "Track",V
19: dco "Printer on, set form,continue";sto
20: wto M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
21: wto N,27,79,4,43,5,32
22: .0845+R(1,1);.0875+R(1,2);.0905+R(1,3);.094+R(1,4);.098+R(1,5)
23: .102+R(1,6);.1055+R(1,7);.111+R(1,8);.1155+R(1,9);.12+R(1,10)
24: for J=11 to 15;R(1,J-1)+.005+R(1,J);next J
25: for J=1 to 16;(.23+.025(J-1))/2+R(2,J)
26: .24+.02(J-1)+R(3,J);.2+.08(J-1)+R(4,J);.25+.25(J-1)+R(5,J)
27: if R(5,J)>.767+R(5,J)+.233+R(5,J)
28: 1+(J-1)+R(6,J);.757+R(6,J)+.233+R(6,J);next J
29: for I=1 to 5;for J=1 to 15;(R(1,I+1)+R(1,J))/2+R(1,I);next J;next I
30: "CHOMP";for V=1 to 2;wto v,10,10
31: ent "level",CS;CS=CS+1,24-len(CS))+CS
32: ent "DAS-32: D.9158",J(1);ent "D.9158",J(2)

```

THE BDM CORPORATION

```

33: tsk 1;for I=1 to 2;1# J[0],S,S;S#S#S[0];coll 'sort';next 2
34: for D=1 to 2;for I=1 to 5
35: for J=1 to 15;F[D,I,J]#D[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;D#D[D,I,J];next J;next I
37: D#D[D,5,1]#D[D,5,1];for J=1 to 15;D#D[D,4,J];next J;next D
38: for I=1 to 5;for J=1 to 15;if D[2,I,J]=0 or D[1,I,J]=0;-1000#3[I,J];tss 2
39: log(D[2,I,J])-log(D[1,I,J])#2[I,J]
40: next J;next I;for D=1 to 2
41: "nat":ina A,F;for I=1 to 5;D#r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
42: for J=1 to 15;1#3;if I=5 and J=1;1#r6
43: if J<15;tss 3
44: if I>1 and I<6;gto "next"
45: if I=1;-1.5#X;gto "1"
46: if I=6 and r0#0 and r7#0;log(r3/r0)+X;log(r7/r0)+X;1+r5;gto "3"
47: if D[D,I,J]=0 and r5#0;gto "next"
48: if r0#0;gto "extra"
49: if D[D,I,J]=0 and r5#1;gto "extra"
50: log(D[D,I,J])#P;log(1[I,J])#X
51: if I=1 and J<9;gto "3"
52: if I=6;gto "2"
53: gto "3"
54: "extra":r1#1+r5;r7#D[D,I,1]#r7;r8#F[I,J]#r8;gto "next"
55: "3":r0#X#r0;r1#X#r1;r2#r1+r2+r3+X#r3;r4#r4;if r5#0;gto "2"
56: X#X[7];X#X[8];1.5#X
57: "1":(r2r1-r4r1)/(r0r2-r1r1)#X[2]
58: (r0r4-r3r1)/(r0r2-r1r1)-F[1];F[1]+X#X[2]#P
59: "2":for X=0 to 5;D#r1-X#2;C#D[D,X]#D[D,X]
60: P#X[2]#X[0];P#X#P;C#C
61: if X#0;C#D[D-1,X]#C[D-1,X]
62: C#X#3;next X
63: "next":next J;next I
64: for I=1 to 3+1
65: for J=1 to int((C+1)/2)
66: if I=1 and I+K<C+2 and I-K>0;F[I,1]#F[I-K,I+K]#F[I+1,I-K]
67: if I+K<C+1 and I-K+1>0;F[I,I+1]#F[I+K,I-K+1]
68: if I+K<C and I-K>0;F[I,I+1]#F[I-K,I+K+1]
69: next J;next I;inv F#F;if D=1;acc F#C;tss 2
70: ret F#F
71: next D;acc F#G#3
72: "out":wrt "8.7",C[3],J[2],J[1],CS
73: Y[2,4]-Y[1,4]+Z;Y[2,5]-Y[1,5]+50*Z+Z
74: wrt "8.2",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
75: wrt 5," 81=",B.0," minutes";wrt "A.5",Z
76: wrt 1,10,13
77: wrt 1,"delta(log(S/dr))";wrt 1,10,13;for J=2 to 15;for I=1 to 5
78: if I=4;gto "OUTLAY"

```

THE BDM CORPORATION

```

90: if S[I,J]<999;wto M,"      ",42,"      ";jto 2
91: wrt "1.6",S[I,J]
92: "CALCULAY":next I;wto M,10,13;next J;wto M,10,13
93: qto "END"
94: "sort":for I=1 to 6;for J=1 to 15;D+15I-15+K
95: stf (AS[D,4K+1,4K+4])>F[D,I,J]:next J;next I
96: D+K;for I=1 to 15
97: stf (AS[D,361+4K,364+4K])>X[I];1+K+K;if I>5;qto "NEXT"
98: stf (AS[D,361+4K,364+4K])>Y[D,I];1+K+K;if I>3;qto "NEXT"
99: itf (AS[D,361+4K,362+4K])>X[2I-1]
90: itf (AS[D,363+4K,364+4K])>X[2I];1+K+K
91: "NEXT":next I
92: stf (AS[D,457,460])>Z[D]
93: ret
94: "ORV":for D=1 to 8;for J=1 to 5;.05(S(J-1)+D)-1.05+X;D+Y
95: for I=1 to C+1;YX+G[I]>Y;next I
96: Y>S[J,D]
97: next J;next D
98: wto M,13,"Fitted Curve";wto M,10,13;for J=1 to 6;for I=1 to 5
99: if S[I,J]=0;wto M,"      ",42,"      ";jto 2
100: wrt "1.6",S[I,J]
101: next I;wto M,10,13;next J
102: next V
103: wto M,13
104: qto "SIAPFI"
105: end
*6233

```


THE BDM CORPORATION

5. Plot $\log(\Delta(dN/dr))$ vs. $\log(r)$

This program retrieves data from two specified files on the cassette data tape and generates a polynomial fit, in log space, for each of the two files. It then plots, point by point, the logarithm of the difference between the antilogs of these two functions versus $\log(r)$. The matrices serve the same functions as described above.

A program listing follows:

```
0: "TAG-32: PLOTS LOG(DELTA(dN/dr)) FROM DATA ON FILE":
1: dim AS[2,460],BS[460],ES[36],J[2],G[6,15],CS[25],Z[2]
2: dim Y[2,6],X[6],F[2,6,15],I[15],O[2,6,15]
3: dim Z[3],B[6,15],D[3],K[10]
4: dim D[6,15],R[6,15],N[10],F[10,10],S[10],I[9],M[6]
5: 35*B[1];111*B[2];43*B[3];0*B[4];42*B[5];64*B[6]
6: 715*Y;dev "Y",4
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,e1,f4.0,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,e15.7,z
16: 7-C
17: C+1-I;rdm A[I],G[I],K[I],F[I,I]
18: dso "Insert data tape,continue";sto
19: ent "Tape #",C[3];ent "Track",I
20: dso "Printer on, set form,continue";sto
21: wtb 1,27,69,27,34,32,32,32,32,27,77,27,76,15,0,14
22: wtb 1,27,79,4,48,6,32
23: 3*X[1]+X[2]+X[3];1*X[5]+X[6];15*X[4]
24: .0845*R[1,1];.0875*R[1,2];.0905*R[1,3];.094*R[1,4];.096*R[1,5]
25: .102*R[1,6];.1035*R[1,7];.111*R[1,8];.1155*R[1,9];.12*R[1,10]
26: for J=11 to 15;R[1,J-11+.005*R[1,J];next J
27: for I=1 to 16;(.23+.025(J-1))/2*R[2,J]
28: .2+.02(J-1)+R[3,J];.3+.08(J-1)+R[4,J];.23+.25(J-1)+R[5,J]
29: if F[5,J]>1;.767*R[5,J]+.233*R[5,J]
30: 1+(J-1)*R[6,J];.767*R[6,J]+.233*R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2+E[I,J];next J;next I
32: "start";ent "Event",CS;ent "Background file",J[1];ent "Event file",J[2]
33: trk w;for D=1 to 2;ldf J[D],ES,ES;35*AS[D];c11 'sort';next D
```

THE BDM CORPORATION

```

34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;T[D,I,J]←O[D,I,J];next J;next I
36: for I=1 to 3;for J=1 to 3;O←O[D,I,J];next J;next I
37: O←O[D,5,1]←O[D,6,1];for J=1 to 15;O←O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;O[2,I,J]←O[1,I,J]÷3[I,J]
39: if S[I,J]<0;O←S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ina A,F;for I=1 to 6;O←r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
42: for J=1 to 15;1←3;if I=6 and J#1;1←r6
43: if J<15;jmp 5
44: if I>1 and I<6;goto "next"
45: if I=1;-1.5←X;goto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)←X;log(r7/r9)←P;1←r5;goto "3"
47: 1.5←X;goto "1"
48: if O[D,I,J]=0 and r6=0;goto "next"
49: if r9=0;goto "extrao"
50: if O[D,I,J]=0 and r6=1;goto "extrao"
51: log(O[D,I,J])←P;log(F[I,J])←X
52: if I=1 and J<9;goto "3"
53: if I#6;goto "2"
54: goto "3"
55: "extrao":r0←1+r9;r7←O[D,I,J]←r7;r8←F[I,J]←r8;goto "next"
56: "3":r0←X←r0;r1←X←r1;r2←1←r2;r3←PX←r3;r4←P←r4;if r5=0;goto "2"
57: X←B[7];P←B[8];1.5←X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)←M[2]
59: (r0r4-r3r1)/(r0r2-r1r1)←M[1];M[1]←X←M[2]←P
60: "2":for K=J to C;C←1-K←P;G←F[R,P]←F[P,P]
61: P←A[R]←A[P];P←X←P;G←G
62: if K#C;G←F[R-1,P]←F[P-1,P]
63: G←X←G;next K
64: "next":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;F[I,I]←F[I-K,I+K]←F[I+K,I-K]
68: if I+K=C+1 and I-K+1>0;F[I,I+1]←F[I+K,I-K+1]
69: if I+K=C and I-K>0;F[I,I+1]←F[I-K,I+K+1]
70: next K;next I;inv F←F;if D=1;mat FA←3;jmp 2
71: mat FA←K
72: next D
73: "out":wrt "1.7",C[3],J[2],J[1],CS
74: Y[2,4]←Y[1,4]÷2;Y[2,5]←Y[1,5]÷60×2÷2
75: wrt "1.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5," JT=",EJ.0," minutes";wrt "1.5",Z
77: wrt "1.4",Z[2],Z[1];wth 4,10,13
78: goto "no dn/3r"
79: wth M,"CM/dr";wth 4,10,13;for J=1 to 15;for I=1 to 6
80: wrt "1.6",T[I,J];next I;wth 4,10,13;next J;wth M,10,13
81: "no dn/3r":

```

THE BDM CORPORATION

```

82: wtb M,27,65,-4,0,7,32,"log(delta(dY/dr))"
83: wtb M,27,65,-4,-210,-1,-16,"log(radius)"
84: "plt":if flg7;gto "skip print"
85: -1+X;-4+Y
86: wtb M,27,65,int(15X/4),int(240X),0,0
87: if X#0 and Xmod=0;wtb M,"|",10,8,8,8;wrt "M.1",X;gto +2
88: wtb M,"-"
89: if (X+.05+X)<2.05;gto -3
90: wtb M,27,65,0,0,int(1.5Y),int(96Y)
91: if Y#0 and Ymod=0;wrt "M.2","-",Z;gto +2
92: wtb M,"|"
93: if (Y+.1+Y)<5.1;gto -3
94: gto "skip print"
95: "sort":for I=1 to 6;for J=1 to 15;J+15I-16+K
96: stf(AS[D,4K+1,4K+4])→F[D,I,J];next J;next I
97: 0→K;for I=1 to 15
98: stf(AS[D,361+4K,364+4K])→I[I];1+K→K;if I>5;gto "NoX"
99: stf(AS[D,351+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "NoX"
100: itf(AS[D,361+4K,362+4K])→X[2I-1]
101: itf(AS[D,363+4K,364+4K])→X[2I];1+K→K
102: "NoX":next I
103: stf(AS[D,457,460])→Z[D]
104: ret
105: "skip print":-1+X
106: "crv":0→Y→Z
107: for I=1 to C+1;YX+G[I]→Y;ZX+K[I]→Z;next I
108: tn^Z-tn^Y+Z;if Z<=0;gto "help"
109: log(Z)→Y
110: if Y<-4 or Y>5;jmp 4
111: Y+6/96→Y
112: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
113: wtb M,"."
114: "help":if (X+1/120+X)<.3;gto "crv"
115: wtb M,12,13
116: gto "start"
117: end
*24425

```

THE BDM CORPORATION

6. Tabulation of $\log (\Delta(dN/dr))$ vs. r

This program takes data from two specified files on the cassette tape and tabulates differences between them in two formats. In the first, the logarithm of the bin by bin differences between dN/dr is tabulated. In the second, a polynomial fit of $\log(dN/dr)$ vs. $\log(r)$ is generated for each of the two data sets and the logarithm of the differences of dN/dr , calculated from these two functions at specified radius values, is tabulated. The matrices employed are the same as in the previous programs.

A program listing follows:

```

0: "DAG-32: INPOLUTES LOG(DELTA((1+sqrt(r)) vs. r AND THE VALUES OF":
1: "THE POLYNOMIAL FIT TO THIS FUNCTION FOR VARIOUS FACIL":
2: DIM NS[2,480],NS[460],ES[36],J[2],S[6,15],CS[25],Z[2]
3: DIM X[2,6],X[6],F[2,5,15],R[15],C[2,5,15]
4: DIM E[6],E[5,15],O,C[3],A[10],DS[25]
5: DIM O[6,15],P[5,15],A[10],S[10,10],J[10],J[9],J[6]
6: " " "→FS
7: 35→L[1];11L→L[2];43→S[3];0→S[4];42→L[5];64→L[6]
8: 701→I;REV " ";I
9: ENT 0,LBX,Z
10: ENT 1,F3.0
11: ENT 2,C1,F4.0,Z
12: ENT 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
13: ENT 6,c11.2,z
14: ENT 7,"Face #",f3.0," Files",f3.0," and",f3.0," Event ",c25
15: ENT 8,"Polynomial of order ",f2.0,z
16: ENT 9,e15.7,z
17: 7→C
18: C+1→I;rdm A[I],S[I],K[I],F[I,I]
19: dso "Insert Data tape,continue";sto
20: ent "Time =",C[3];ent "Track",I
21: dso "Printer on, set form,continue";sto
22: wtb 4,27,59,27,84,32,32,32,32,27,77,27,75,15,0,14
23: wtb ",27,75,4,43,6,32
24: 3×X[1]+X[2]+X[3];1×X[5]+X[6];15×X[4]
25: .0845×P[1,1];.0875×P[1,2];.0405×P[1,3];.094×P[1,4];.005×P[1,5]
26: .102×P[1,6];.1055×P[1,7];.111×P[1,8];.1155×P[1,9];.12×P[1,10]
27: for J=1 to 15:P[1,J-1]+.005×P[1,J];next J
28: for J=1 to 16:(.12+.025(J-1))/2×P[2,J]
29: (.2+.02(J-1))×P[3,J];(.3+.03(J-1)+.1(4,J);.25+.25(J-1)×P[5,J]
30: if P[5,J]>.1;.757×P[5,J]+.223×P[5,J]
31: 1+(J-1)×P[15,J];.757×P[15,J]+.233×P[15,J];next J

```

THE BDM CORPORATION

```

32: for I=1 to 6; for J=1 to 15; (F[I,J+1]+F[I,J])/2+G[I,J]; next J; next I
33: "DIARI": for V=1 to 2; wtb M, 10, 10
34: ent "Event", CS; CS$DS[1, 24-len(CS)]+CS
35: ent "BACKGROUND FILE", J[1]; ent "EVENT FILE", J[2]
36: trk W; for D=1 to 2; ldf J[D], BS, ES; BS+AS[D]; call 'sort'; next D
37: for D=1 to 2; for I=1 to 6
38: for J=1 to 15; T[D,I,J]+O[D,I,J]; next J; next I
39: for I=1 to 3; for J=1 to 3; O+O[D,I,J]; next J; next I
40: O+O[D,5,1]+O[D,6,1]; for J=1 to 15; O+O[D,4,J]; next J; next D
41: for I=1 to 6; for J=1 to 15; O[2,I,J]-O[1,I,J]+S[I,J]
42: if S[I,J]<=0; O+3[I,J]; jmp 2
43: log(S[I,J])+3[I,J]
44: next J; next I; for D=1 to 2
45: "mat": ina A,F; for I=1 to 6; O+r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
46: for J=1 to 16; l+G; if I=5 and J=1; l+r6
47: if J<16; jmp 5
48: if I>1 and I<6; goto "next"
49: if I=1; -1.5+X; goto "1"
50: if I=6 and r0#0 and r7#0; log(r6/r7)+X; log(r7/r9)+r; l+r5; goto "3"
51: 1.5+X; goto "1"
52: if O[D,I,J]=0 and r6=0; goto "next"
53: if r0#0; goto "extra"
54: if O[D,I,J]=0 and r6=1; goto "extra"
55: log(O[D,I,J])+r; log(r6/r7)+X
56: if I=1 and J<3; goto "3"
57: if I#6; goto "2"
58: goto "3"
59: "extra": r6+l+r; r7+O[D,I,J]+r7; r6+F[I,J]+r6; goto "next"
60: "3": r0+X+r0; r1+r+r1; r2+l+r; r3+X+r3; r4+r+r4; if r5#0; goto "2"
61: X+3[7]; -F[8]; 1.5+X
62: "1": (r3r2-r4r1)/(r0r2-r1r1)+1[2]
63: (r0r4-r3r1)/(r0r2-r1r1)+4[1]; 3[1]+X*M[2]+F
64: "2": for K=0 to C; C+1-K+X; G+F[R,R]+F[R,F]
65: P+A[R]+A[F]; P*X+P; GX+G
66: if K#C; G+F[R-1,R]+F[F-1,R]
67: G*X+G; next K
68: "next": next J; next I
69: for I=1 to C+1
70: for K=1 to int((C+1)/2)
71: if I#1 and I+K<C+2 and I-K>0; F[I,I]+F[I-K,I+K]+F[I+K,I-K]
72: if I+K=C+1 and I-K+1>0; F[I,I+1]+F[I+K,I-K+1]
73: if I+K=C and I-K>0; F[I,I+1]+F[I-K,I+K+1]
74: next K; next I; inv F+F; if D=1; mat FA+G; jmp 2
75: mat FA+K
76: next D
77: "out": wrt "v.7", C[3], J[2], J[1], CS
78: Y[2,4]-Y[1,4]+Z; Y[2,5]-Y[1,5]+60*Z+Z
79: wrt "v.3", Y[2,2], Y[2,3], Y[2,1], Y[2,4], Y[2,5]

```

THE BDM CORPORATION

```

80: fmt 5," dT=",f3.0," minutes";wrt "M.5",Z
81: wtb M,10,13
82: wtb M,"log(delta(dN/dr))";wtb M,10,13;for J=2 to 15;for I=1 to 6
83: if I=4;gto "CALWAY"
84: if S(I,J)=0;wtb M,"      ",42,"      ";jmp 2
85: wrt "1.6",3[I,J]
86: "CALWAY":next I;wtb M,10,13;next J;wtb M,10,13
87: gto "CPV"
88: "sort":for I=1 to 6;for J=1 to 15;J+15I-15+K
89: sti (AS[D,4K+1,4K+4]) +P(D,I,J);next J;next I
90: 0+K;for I=1 to 15
91: sti (AS[D,351+4K,354+4K]) +H(I);1+K+K;if I>5;gto "NextI"
92: sti (AS[D,351+4K,354+4K]) +Y[D,I];1+K+K;if I>3;gto "NextI"
93: ite (AS[D,351+K,352+4K]) +K[2I-1]
94: ite (AS[D,353+4K,354+4K]) +K[2I];1+K+K
95: "NextI":next I
96: sti (AS[D,457,460]) +Z[I]
97: ret
98: "CPV":for D=1 to 3;for J=1 to 5;.05(D(J-1)+3)-1.05+K;0+Y+Z
99: for I=1 to 6;1K+3[I]+Y;1K+K[I]+Z;next I
100: tn^2-tn^1+1;if Z<=0;0+3[I,D];jmp 2
101: log(3)+3[I,D]
102: next D;next 0
103: wtb M,13,"Fitted Curve";wtb M,10,13;for J=1 to 6;for I=1 to 5
104: if S(I,J)=0;wtb M,"      ",42,"      ";jmp 2
105: wrt "1.6",S(I,J)
106: next I;wtb M,10,13;next J
107: next v
108: wtb M,13
109: gto "SCATT"
110: end
*22120

```

7. Integration of dN/dr over r from 0.2 to 7 microns

This program integrates dN/dr over r from 0.2, 0.3, 0.4, 0.6, 0.8 and 1 to 7 microns for any specified file on the data tape. The integration employs the polynomial fit and the trapezoidal rule. The matrices serve the same function as in previous programs.

A program listing follows:

```

0: "INTEGRATES OVER dN/dr 0.2 TO 7 MICRONS GIVING TOTAL":
1: "N IN VARIOUS RADII INTERVALS. FROM DATA ON FILE":
2: dim A$(460),ES(36),U(6),B$(100)
3: dim Y(6),X(6),F(6,15),H(16),O(6,15)
4: dim B(8),E(6,15),D,C(3)
5: dim R(6,16),A(10),F(10,10),M(6),G(10)
6: 715*4;dev "M",4
7: 7*0
8: C+1+I;rdm A(I),B(I),F(I,I)
9: dso "Insert data tape,continue";sto
10: ent "Tape #",C(3);ent "Track #",4
11: dso "Printer on, set form,continue";sto
12: wtb M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
13: wtb M,27,79,4,48,6,32
14: fnt 0,"TAPE #",E2.0," INTEGRATION OVER RADII."
15: wrt "4.0",C(3);fnt 0
16: wrt "1.0","SUM IS FROM INDICATED RADIUS TO 7 MICRONS. UNITS ARE CM-3"
17: " Date Time File Sum(.2) Sum(.3) Sum(.4) "PS
18: PS8"Sum(.6) Sum(.8) Sum(1)"PS;wto B,ES,10,13
19: 0+X(1)+X(2)+X(3)+X(5)+X(6)+15*X(4)
20: .0845+R(1,1);.0875+R(1,2);.0905+R(1,3);.094+R(1,4);.098+R(1,5)
21: .102+R(1,6);.1065+R(1,7);.111+R(1,8);.1155+R(1,9);.12+R(1,10)
22: for J=11 to 16;R(1,J-1)+.005+R(1,J);next J
23: for J=1 to 16;(.23+.025(J-1))/2+R(2,J)
24: .2+.02(J-1)+R(3,J);.3+.08(J-1)+R(4,J);.25+.25(J-1)+R(5,J)
25: if R(5,J)>1;.767+R(5,J)+.233+R(5,J)
26: 1+(J-1)+R(6,J);.767+R(6,J)+.233+R(6,J);next J
27: for I=1 to 6;for J=1 to 15;(R(I,J+1)+R(I,J))/2+E(I,J);next J;next I
28: "start":ent "Begin with file",E;ent "End with file",F
29: for D=E to F;trk W;ldf D,AS,ES;cll 'sort'
30: for I=1 to 6
31: for J=1 to 15;T(I,J)+O(I,J);next J;next I
32: for I=1 to 3;for J=1 to 3;0+O(I,J);next J;next I
33: 0+O(5,1)+O(6,1);for J=1 to 15;0+O(4,J);next J
34: "mat":ind A,F;for I=1 to 6;0+r0+r1+r2+r3+r4+r5+r7+r8+r9
35: for J=1 to 16;1+G;if I=6 and J#1;1+r6

```

THE BDM CORPORATION

```

36: if J<16; jmp 5
37: if I>1 and I<6; goto "nExt"
38: if I=1; -1.5*X; goto "1"
39: if I=6 and r9#0 and r7#0; log(r8/r9)*X; log(r7/r9)*P; 1-r5; goto "3"
40: 1.5*X; goto "1"
41: if O[I,J]=0 and r6=0; goto "nExt"
42: if r9#0; goto "extrao"
43: if O[I,J]=0 and r6=1; goto "extrao"
44: log(O[I,J])>F; log(F[I,J])>X
45: if I=1 and J<9; goto "3"
46: if I#6; goto "2"
47: goto "3"
48: "extrao": r9+1-r9; r7+O[I,J]*r7; r3+O[I,J]*r3; goto "nExt"
49: "3": r0+X*X+r0; r1+X+r1; r2+1-r2; r3+P*X+r3; r4+P+r4; if r5=0; goto "2"
50: X+3[7]; P+3[8]; 1.5*X
51: "1": (r3r2-r4r1)/(r0r2-r1r1)*4[2]
52: (r0r4-r3r1)/(r0r2-r1r1)*4[1]; 4[1]+X*4[2]>P
53: "2": for K=1 to C; C+1-K>R; G+F[P,R]>F[P,R]
54: P+A[R]>A[R]; P*X>P; 3X>G
55: if K#C; G+F[R-1,R]>F[R-1,R]
56: G*X>G; next K
57: "nExt": next J; next I
58: for I=1 to C+1
59: for K=1 to int((C+1)/2)
60: if I#1 and I+K<C+2 and I-K>0; F[I,I]>F[I-K,I+K]>F[I+K,I-K]
61: if I+K=C+1 and I-K+1>0; F[I,I+1]>F[I+K,I-K+1]
62: if I+K=C and I-K>0; F[I,I+1]>F[I-K,I+K+1]
63: next K; next I; inv F>F; mat F>G
64: goto "zisisit"
65: "sort": for I=1 to 6; for J=1 to 15; J+15I-16>K
66: stf(AS[4K+1,4K+4])>F[I,J]; next J; next I
67: 0>K; for I=1 to 16
68: stf(AS[361+4K,364+4K])>I[I]; 1+K>K; if D>5; goto "JeXT"
69: stf(AS[361+K,364+4K])>I[I]; 1+K>K; if I>3; goto "JeXT"
70: itf(AS[361+4K,362+4K])>X[2I-1]
71: itf(AS[363+4K,364+4K])>X[2I]; 1+K>K
72: "JeXT": next I
73: stf(AS[457,460])>F
74: ret
75: "zisisit":
76: ina J; for I=.2 to 6.99 by .01; 0>Y>Z; log(R)>A; log(R+.01)>E
77: for K=1 to C+1; Y+G[K]>Y; ZB+G[K]>Z; next K
78: trf Y>Y; trf Z>Z; .005*(Y+Z)>A; A+U[1]>U[1]
79: if R>.29; A+U[2]>U[2]; if R>.39; A+U[3]>U[3]; if R>.59; A+U[4]>U[4]
80: if R>.79; A+U[5]>U[5]; if R>.99; A+U[6]>U[6]
81: next R; fnt 0, f2.0, "/", f2.0, "/", f2.0, f4.0, ":", f2.0, z
82: fnt 1, f5.0, f11.2, f11.2, f11.2, f11.2, f11.2, f10.2
83: wrt " ", Y[2], Y[3], Y[4], Y[5]
84: wrt " ", 0, U[1], U[2], U[3], U[4], U[5], U[6]
85: next D; goto "start"
86: end

```

*20070

THE BDM CORPORATION

8. Integration of dN/dr over r from 1.5 to 7 microns

This program is identical to the previous program except that it provides integrations of dN/dr over the intervals of r from 1.5, 2, 2.5, 3, 3.5 and 4 to 7 microns.

A program listing follows:

```

0: "DAS-32: INTEGRATES FROM 1.5,2,2.5,3,3.5,4 TO 7 MICRONS":
1: dim AS(460),CS(36),U(6),BS(100)
2: dim Y(6),X(6),T(5,15),H(16),C(5,15)
3: dim S(8),Z(5,15),D,C(3)
4: dim P(6,16),A(10),F(10,10),M(6),S(10)
5: 715*1;dev "A",1
6: 7*0
7: C+1+I;rdm AH),C(I),F(I,I)
8: dim "Insert data tape,continue";sto
9: ent "Tape #",C(3);ent "Track #",I
10: dim "Printer on, set form,continue";sto
11: wtb 1,27,59,27,84,32,32,32,32,27,77,27,76,15,0,14
12: wtb 1,27,79,4,48,6,32
13: fnt 0,"TAPE #",F2.0," INTEGRATION OVER FACIL."
14: wrt "1.0",C(3);fnt 0
15: wrt "1.0","SUM IS FROM INDICATED RADIUS TO 7 MICRONS.UNITS ARE CM-3"
16: " Date Time File Sum(1.5) Sum(2) Sum(2.5) "→BS
17: BS→"Sum(3) Sum(3.5) Sum(4)"→BS;wtb 1,BS,10,13
18: 3→X(1)→X(2)→X(3);1→X(5)→X(6);15→X(4)
19: .0345→R(1,1);.0875→R(1,2);.0905→R(1,3);.094→R(1,4);.093→R(1,5)
20: .102→R(1,6);.1055→R(1,7);.111→R(1,8);.1155→R(1,9);.12→R(1,10)
21: for J=11 to 16;R(1,J-1)+.005→R(1,J);next J
22: for J=1 to 15;(.23+.025(J-1))/2→R(2,J)
23: .2+.02(J-1)→R(3,J);.3+.03(J-1)→R(4,J);.25+.25(J-1)→R(5,J)
24: if R(5,J)>1;.767→R(5,J)+.233→R(5,J)
25: 1+(J-1)→R(6,J);.767→R(6,J)+.233→R(6,J);next J
26: for I=1 to 6;for J=1 to 15;(R(I,J+1)+R(I,J))/2→S(I,J);next J;next I
27: "start":ent "Begin with file",3;ent "End with file",8
28: for D=8 to 8;trk 4;ldf D,AS,ES;cll 'sort'
29: for I=1 to 5
30: for J=1 to 15;T(I,J)→O(I,J);next J;next I
31: for I=1 to 3;for J=1 to 3;O→O(I,J);next J;next I
32: O→O(5,1)→O(6,1);for J=1 to 15;O→O(4,J);next J
33: "mat":dim A,F;for I=1 to 5;O→r0→r1→r2→r3→r4→r5→r6→r7→r8→r9
34: for J=1 to 15;1→G;if I=5 and J#1;1→r6
35: if J<15;jmp 5
36: if I>1 and I<6;goto "next"
37: if I=1;-1.5→X;ato "1"

```

THE BDM CORPORATION

```

38: if I=5 and r9#0 and r7#0;log(r8/r9)*X;log(r7/r9)*F;1+r5;goto "3"
39: 1.5*X;goto "1"
40: if O[I,J]=0 and r6=0;goto "nEXt"
41: if r9#0;goto "extrao"
42: if O[I,J]=0 and r6=1;goto "extrao"
43: log(O[I,J])*F;log(F[I,J])*X
44: if I=1 and J<9;goto "3"
45: if I#6;goto "2"
46: goto "3"
47: "extrao":r9+1+r9;r7+O[I,J]*r7;r8+O[I,J]*r8;goto "nEXt"
48: "3":r0+X+r0;r1+X+r1;r2+1+r2;r3+FX+r3;r4+1+r4;if r5=0;goto "2"
49: X=E[7];P=B[8];1.5*X
50: "1":(r3r2-r4r1)/(r0r2-r1r1)+1[2]
51: (r0r4-r3r1)/(r0r2-r1r1)+M[1];M[1]+X*M[2]*F
52: "2":for K=0 to C;C+1-K+R;G+F[F,R]*F[R,R]
53: P+A[F]*A[F];P*X+P;GX*G
54: if K#C;G+F[R-1,R]*F[R-1,R]
55: G*X+G;next K
56: "nEXt":next J;next I
57: for I=1 to C+1
58: for K=1 to int((C+1)/2)
59: if I#1 and I+K<C+2 and I-K>0;F[I,I]+F[I-K,I+K]*F[I+K,I-K]
60: if I+K=C+1 and I-K+1>0;F[I,I+1]+F[I+K,I-K+1]
61: if I+K=C and I-K>0;F[I,I+1]+F[I-K,I+K+1]
62: next K;next I;inv F*F;mat FA*G
63: goto "zisisit"
64: "cort":for I=1 to 5;for J=1 to 15;J+15I-15*X
65: stf(AS(4K+1,4K+4))*F[I,J];next J;next I
66: 0*X;for I=1 to 15
67: stf(AS(361+4K,364+4K))*H[I];1+K*X;if D5;goto "leXI"
68: stf(AS(361+4K,364+4K))*Y[I];1+K*X;if I>3;goto "leXI"
69: itf(AS(361+4K,362+4K))*X(2I-1)
70: itf(AS(363+4K,364+4K))*X(2I);1+K*X
71: "leXI":next I
72: stf(AS(457,460))*F
73: ret
74: "zisisit":
75: ind J;for R=1.5 to 6.93 by .02;0+Y+Z;log(Z)*A;log(F+.02)*B
76: for K=1 to C+1;YA+G[K]*Y;ZB+G[K]*Z;next K
77: tn^Y*X;tn^Z*Z;.01*(Y+Z)*A;A+U[1]*U[1]
78: if P>1.03;A+U[2]*U[2];if R>2.48;A+U[3]*U[3];if R>2.93;A+U[4]*U[4]
79: if R>3.48;A+U[5]*U[5];if P>3.99;A+U[6]*U[6]
80: next R;fmt 0,f2.0,"/",f2.0,"/",f2.0,f4.0,"",f2.0,z
81: fmt 1,f3.0,f12.4,f11.4,f11.4,f11.4,f11.4,f11.4
82: wrt " ",Y[2],Y[3],Y[1],Y[4],Y[5]
83: wrt "5.1",U[1],U[2],U[3],U[4],U[5],U[6]
84: next I;goto "start"
85: end
*14097

```

THE BDM CORPORATION

9. Plot Fractional Change in dN/dr

This program plots $(1/dN_1)(dN_2 - dN_1)$ as a function of $\log(r)$ from the data in any two specified files. It does so by utilizing polynomial fits, in log space, to the two data sets. The matrices employed have the same correspondence as in previous programs.

A program listing follows:

```
0: "EAS-32: PLOTS FRACTIONAL CHANGES IN  $dN/dr$ ... (1/N) (DELTA (dN/dr))":
1: dim AS[2,450],DS[450],FS[36],J[2],S[6,15],CS[25],Z[2]
2: dim Y[2,6],X[6],T[2,6,15],H[15],O[2,6,15]
3: dim C[8],B[6,15],D,C[3],K[10]
4: dim D[6,15],R[6,16],A[10],F[10,10],G[10],I[9],M[6]
5: 35*B[1];111*B[2];43*B[3];0*B[4];42*B[5];64*B[6]
6: 715*4;dev "1",M
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,c1,t5.1,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,e10.2,z
13: fmt 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: fmt 8,"Polynomial of order ",f2.0,z
15: fmt 9,e15.7,z
16: 7*C
17: C+1*I;rdm A[I],G[I],K[I],F[I,I]
18: dsp "Insert data tape,continue";sto
19: ent "Tape #",C[3];ent "Track",J
20: dsp "Printer on, set form,continue";sto
21: wtb M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
22: wtb M,27,79,4,43,6,32
23: 3*X[1]+X[2]+X[3];1*X[5]+X[6];15*X[4]
24: .0845*R[1,1];.0875*R[1,2];.0905*R[1,3];.094*R[1,4];.098*R[1,5]
25: .102*R[1,6];.1065*R[1,7];.111*R[1,8];.1155*R[1,9];.12*R[1,10]
26: for J=11 to 16;R[1,J-1]+.005*R[1,J];next J
27: for J=1 to 16;(.23+.025(J-1))/2*R[2,J]
28: .2+.02(J-1)*R[3,J];.3+.08(J-1)*R[4,J];.25+.25(J-1)*R[5,J]
29: if R[5,J]>1;.767*R[5,J]+.233*R[5,J]
30: 1+(J-1)*R[6,J];.767*R[6,J]+.233*R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2+E[I,J];next J;next I
32: "start":ent "Event",CS;ent "background file",J[1];ent "Event file",J[2]
33: urf J;for D=1 to 2;if J[D],CS,CS;CS+AS[D];call 'sort';next D
34: for D=1 to 2;for I=1 to 6
35: for J=1 to 15;F[D,I,J]+O[D,I,J];next J;next I
```

THE BDM CORPORATION

```

36: for I=1 to 3;for J=1 to 3;O=O[D,I,J];next J;next I
37: O=O[D,5,1]+O[D,5,1];for J=1 to 15;O=O[D,4,J];next J;next D
38: for I=1 to 6;for J=1 to 15;O[2,I,J]=O[1,I,J]+S[I,J]
39: if S[I,J]<0;O=S[I,J]
40: next J;next I;for D=1 to 2
41: "mat":ino A,F;for I=1 to 6;O=r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
42: for J=1 to 16;1+G;if I=6 and J#1;1+r6
43: if J<16;jmp 5
44: if I>1 and I<6;goto "next"
45: if I=1;-1.5+X;goto "1"
46: if I=6 and r9#0 and r7#0;log(r8/r9)+X;log(r7/r9)+P;1+r5;goto "3"
47: 1.5+X;goto "1"
48: if O[D,I,J]=0 and r6=0;goto "next"
49: if r9#0;goto "extrap"
50: if O[D,I,J]=0 and r6=1;goto "extrap"
51: log(O[D,I,J])+P;log(F[I,J])+X
52: if I=1 and J<9;goto "3"
53: if I#6;goto "2"
54: goto "3"
55: "extrap":r9+1+r9;r7+O[D,I,J]+r7;r8+E[I,J]+r8;goto "next"
56: "3":r0+X+X+r0;r1+X+r1;r2+1+r2;r3+2X+r3;r4+P+r4;if r5=0;goto "2"
57: X+6[7];P+6[8];1.5+X
58: "1":(r3r2-r4r1)/(r0r2-r1r1)+H[2]
59: (r0r4-r3r1)/(r0r2-r1r1)+H[1];H[1]+X*E[2]+P
60: "2":for K=0 to C;2+1-K+P;G+E[R,F]+F[R,R]
61: P+A[P]+M[P];P*X+P;2X+G
62: if K#C;3+F[R-1,R]+F[R-1,R]
63: O*X+1;next K
64: "next":next J;next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I#1 and I+K<C+2 and I-K>0;F[I,I]+F[I-K,I+K]+F[I+K,I-K]
68: if I+K<C+1 and I-K+1>0;F[I,I+1]+F[I+K,I-K+1]
69: if I+K<C and I-K>0;F[I,I+1]+F[I-K,I+K+1]
70: next K;next I;inv F+P;if D=1;mat FA=G;jmp 2
71: mat FA=K
72: next D
73: "out":wrt "M.7",C[3],J[2],J[1],CS
74: Y[2,4]-Y[1,4]+Z;Y[2,5]-Y[1,5]+60*Z+Z
75: wrt "M.3",Y[2,2],Y[2,3],Y[2,1],Y[2,4],Y[2,5]
76: fmt 5," dT=",E3.0," minutes";wrt "M.5",Z
77: wrt "M.4",Z[2],Z[1];wtb M,10,13
78: wtb M,27,65,-4,0,7,32,"(1/N)(DELTA(dN/dR))"
79: wtb M,27,65,-4,-210,-1,-16,"log(radius)"
80: "plt":if flg7;goto "skip print"

```

THE BDM CORPORATION

```

81: -1→X;-4→Y
82: wtb M,27,65,int(15X/4),int(240X),0,0
83: if X#0 and Xmod1=0;wtb M,"|",10,8,2,8;wrt "4.1",X;gto +2
84: wtb M,"-"
85: if (X+.05→X)<2.05;gto -3
86: wtb M,27,65,0,0,int(1.5Y),int(96Y)
87: if Y#0 and Ymod1=0;wrt "4.2","-",2*Y;gto +2
88: wtb M,"|"
89: if (Y+.1→Y)<5.1;gto -3
90: gto "skin print"
91: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
92: stf (AS[D,4K+1,4K+4])→T[D,I,J];next J;next I
93: 0→K;for I=1 to 15
94: stf (AS[D,361+4K,364+4K])→I[I];1+K→K;if I>5;gto "lexI"
95: stf (AS[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;gto "lexI"
96: itf (AS[D,361+4K,362+4K])→X[2I-1]
97: itf (AS[D,363+4K,364+4K])→X[2I];1+K→K
98: "lexI":next I
99: stf (AS[D,457,460])→Z[D]
100: ret
101: "skin print":-1→X
102: "crv":0→Y;0→Z
103: for I=1 to C+1;YX+G[I]→Y;ZX+K[I]→Z;next I
104: tn^2/tn^2-1→Y
105: if Y<-8 or Y>10;jmp 4
106: .5*(Y+6/96)→Y
107: wtb M,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
108: wtb M,"."
109: "help":in (X+1/120→X)<.85;gto "crv"
110: wto M,12,13
111: gto "start"
112: end
*17194

```

10. Plot Normalized Fractional Change in dN/dr

This program is identical to that just preceding except that the plot is normalized by the peak value of the fractional change in dN/dr . This is accomplished by determining the maximum value of the fractional change in dN/dr in the radius interval of interest and plotting the fractional change function divided by this maximum value. All such curves, then, have a maximum of one. The matrices are the same as in the previous program.

A program listing follows:

```

0: "DAS-32: PLOTS (1/N) (dN/dr) NORMALIZED BY ITS PEAK VALUE":
1: dim AS[2,460],BS[460],CS[35],J[2],S[0,15],CS[25],I[2]
2: dim Y[2,6],X[6],P[2,6,15],R[15],Z[2,6,15]
3: dim B[3],C[0,15],D,C[3],K[10]
4: dim D[6,16],P[5,15],A[10],F[10,10],S[10],I[9],E[6]
5: 35→B[0];111→B[2];43→B[3];0→B[4];42→B[5];64→B[6]
6: 715→I;3cv " ",4
7: Ent 0,10x,z
8: Ent 1,13.0
9: Ent 2,e1,55.1,z
10: Ent 3,"Date",f3.0,"/",f2.0,"/",f2.0," Time",f3.0,":",f2.0,z
11: Ent 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: Ent 5,e10.2,z
13: Ent 7,"Tape #",f3.0," Files",f3.0," and",f3.0," Event ",c25
14: Ent 8,"Polynomial of order ",f2.0,z
15: Ent 9,e15.7,z
16: 7→C
17: C+1→I;rdm A[I],G[I],K[I],F[I,I]
18: dso "Insert data tape,continue";sto
19: Ent "Tape #",C[3];Ent "Track",w
20: dso "Printer on, set form,continue";sto
21: wtb M,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
22: wtb N,27,79,4,48,6,32
23: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4]
24: .0845→R[1,1];.0875→R[1,2];.0905→R[1,3];.094→R[1,4];.098→R[1,5]
25: .102→R[1,6];.1055→R[1,7];.111→P[1,3];.1155→R[1,9];.12→R[1,10]
26: for J=11 to 16;R[1,J-1]→.005→R[1,J];next J
27: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
28: .2+.02(J-1)→R[3,J];.3+.08(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
29: if R[5,J]>1;.767→R[5,J];.233→R[5,J]
30: 1+(J-1)→P[6,J];.757→R[6,J];.233→R[6,J];next J
31: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2→Z[I,J];next J;next I
32: "start":Ent "Event",CS;Ent "Background file",J[1];Ent "Event file",J[2]

```

THE BDM CORPORATION

```

33: trk W; for D=1 to 2; if J[D], BS, ES; BS+AS[D]; call 'sort'; next D
34: for D=1 to 2; for I=1 to 6
35: for J=1 to 15; T[D,I,J]←O[D,I,J]; next J; next I
36: for I=1 to 3; for J=1 to 3; O←O[D,I,J]; next J; next I
37: O←O[D,3,1]+O[D,6,1]; for J=1 to 15; O←O[D,4,J]; next J; next D
38: for I=1 to 6; for J=1 to 15; O[2,I,J]←O[1,I,J]+S[I,J]
39: if S[I,J]<0; O←S[I,J]
40: next J; next I; for D=1 to 2
41: "mat":ina A,F; for I=1 to 6; O←r0+r1+r2+r3+r4+r5+r6+r7+r8+r9
42: for J=1 to 15; l←3; if I=6 and J≠1; l←r6
43: if J<15; inc 5
44: if I>1 and I<6; goto "next"
45: if I=1; -1.5←X; goto "1"
46: if I=6 and r9≠0 and r7≠0; log(r8/r9)←X; log(r7/r9)←P; l←r5; goto "3"
47: 1.5←X; goto "1"
48: if O[D,I,J]=0 and r6=0; goto "next"
49: if r9≠0; goto "extrao"
50: if O[D,I,J]=0 and r6=1; goto "extrap"
51: log(O[D,I,J])←P; log(E[I,J])←X
52: if I=1 and J<9; goto "3"
53: if I≠6; goto "2"
54: goto "3"
55: "extrap": r9←l+r9; r7←O[D,I,J]+r7; r8←E[I,J]+r8; goto "next"
56: "3": r0←XX+r0; r1←X+r1; r2←l+r2; r3←PX+r3; r4←P+r4; if r5=0; goto "2"
57: X←3[7]; P←P[8]; 1.5←X
58: "1": (r3r2-r4r1)/(r0r2-r1r1)←X[2]
59: (r0r4-r3r1)/(r0r2-r1r1)←X[1]; X[1]+X*[2]←P
60: "2": for K=0 to C; C←l-X←P; C←F[P,P]+F[D,P]
61: P←A[P]+A[P]; P←X←P; C←C
62: if K≠C; C←F[P-1,C]+F[P-1,P]
63: C←X←3; next K
64: "next": next J; next I
65: for I=1 to C+1
66: for K=1 to int((C+1)/2)
67: if I≠1 and I+K≤C+2 and I-K>0; F[I,I]+F[I-K,I+K]+F[I+K,I-K]
68: if I+K≤C+1 and I-K+1>0; F[I,I+1]+F[I+K,I-K+1]
69: if I+K≤C and I-K>0; F[I,I+1]+F[I-K,I+K+1]
70: next K; next I; inv F←F; if D=1; mat FA←G; jmp 2
71: mat F←K
72: next D
73: "out": wrt "M.7", C[3], J[2], J[1], CS
74: Y[2,4]←Y[1,4]+Z; Y[2,5]←Y[1,5]+60*Z←Z
75: wrt "M.3", Y[2,2], Y[2,3], Y[2,1], Y[2,4], Y[2,5]
76: int 5, " Jt=", f3.0, " minutes"; wrt "1.5", Z
77: wrt "1.1", Z[2], Z[1]; wrt "1,10,13
78: wrt "1,27,35,-4,0,7,32,"<(1/10)(1/10)(1/10)(1/10)>"
79: wrt "1,27,35,-4,-210,-1,-16,"log(radius)"

```

THE BDM CORPORATION

```

30: "rlt":if fl47; goto "skip print"
31: -1→X; -3→Y
32: wtb 1,27,65,int(15X/4),int(240X),0,0
33: if X≠0 and Kmod1=0; wtb 0,"|",10,3,3,3; wrt "1",X; goto +2
34: wtb 1,"-"
35: if (X+.05→X)<2.05; goto -3
36: wtb 1,27,65,0,0,int(1.5Y),int(96Y)
37: if Y≠0 and Kmod1=0; wrt "1.2","-",.2*Y; goto +2
38: wtb 1,"|"
39: if (Y+.1→Y)<5.1; goto -3
40: goto "skip print"
41: "sort":for I=1 to 6;for J=1 to 15;J+15I-16→K
42: stf(AS[D,4K+1,4K+4])→P[D,I,J];next J;next I
43: 0→K;for I=1 to 16
44: stf(AS[D,361+4K,364+4K])→I[I];1+K→K;if I>5;goto "next"
45: stf(AS[D,361+4K,364+4K])→Y[D,I];1+K→K;if I>3;goto "next"
46: itf(AS[D,361+4K,362+4K])→X[2I-1]
47: itf(AS[D,363+4K,364+4K])→X[2I];1+K→K
48: "next":next I
49: stf(AS[D,457,460])→Z[D]
50: ret
51: "skip print":0→A;for L=1 to 2;-1→X
52: "crv":0→Y;0→Z
53: for I=1 to C+1;YX+G[I]→Y;3X+K[I]→Z;next I
54: tn^2/tn^Y-1→Y;if B=2;jmp 3
55: if Y>A;Y→A
56: goto "help"
57: Y/A→Y;if Y<-1 or Y>1.1;jmp 4
58: 5*Y→Y
59: wtb 1,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
60: wtb 1,"."
61: "hole":if (X+1/120→X)<.35;goto "crv"
62: next 0
63: wtb 1,12,13
64: goto "start"
65: end
*3764

```


THE BDM CORPORATION

11. Tabulation of Aerosol Decay Constant

This program calculates the decay constant for aerosols as a function of radius. It assumes an exponential decay and requires the input from three data files, the third being the background. The time elapsed between the two post-event data sets must be specified. The calculations proceeds from polynomial fits in log space and is given by:

$$\tau = (t_2 - t_1) (\ln(dN_2/dr - dN_b/dr) - \ln(dN_1/dr - dN_b/dr))$$

Where t_i is the time of the i^{th} file and the subscript b denotes background.

The matrices employed are the same as in foregoing programs except the dimension of some arrays has been increased by one to accommodate inclusion of the background data set.

A program listing follows:

```
0: "DAS-32: TABULATES THE AEROSOL DECAY CONSTANT, TAU, FOR VARIOUS RADII":
1: dim AS[3,460],DS[460],ES[36],J[4],CS[25],Z[3];sfq 14
2: dim Y[3,6],X[6],H[16],O[3,6,15]
3: dim E[6,15],D,C[3],K[10],S[6,15]
4: dim R[6,16],A[10],F[10,10],G[10],I[10],H[6],DS[25]
5: " " "DS
6: 701-M;dev "M",M
7: fmt 0,10x,z
8: fmt 1,f3.0
9: fmt 2,cl,f4.0,z
10: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0,z
11: fmt 4,"Averaging time = ",f2.0," and ",f2.0," minutes"
12: fmt 6,cl0.2,z
13: fmt 7,"Name ",f2.0," Skand ",f3.0," Files ",f3.0," & ",f3.0," dt=",f5.2
14: fmt 9,cl5.7,z
15: 7-C
16: C+1-I;rem A[I],S[I],K[I],I[I],F[I,I]
17: den "Insert data tape,continue";sto
18: ent "Ine #",C[3];ent "Track",J
19: dep "Printer on, set form,continue";sto
20: wtp 1,27,59,27,84,32,32,32,32,27,77,27,76,15,0,14
21: wtb 1,27,79,4,43,6,32
22: 3-X[1]-X[2]-X[3];1-X[5]-X[6];15-X[4]
```

THE BDM CORPORATION

```

23: .0845*R[1,1];.0875*R[1,2];.0905*R[1,3];.094*R[1,4];.098*R[1,5]
24: .102*R[1,6];.1065*R[1,7];.111*R[1,8];.1155*R[1,9];.12*R[1,10]
25: for J=11 to 16;R[1,J-1]+.005*R[1,J];next J
26: for J=1 to 15;(.23+.025(J-1))/2*R[2,J]
27: .2+.02(J-1)*R[3,J];.3+.08(J-1)*R[4,J];.25+.25(J-1)*R[5,J]
28: if R[5,J]>1;.767*R[5,J]+.233*R[5,J]
29: 1+(J-1)*R[6,J];.767*R[6,J]+.233*R[6,J];next J
30: for I=1 to 6;for J=1 to 15;(R[I,J+1]+R[I,J])/2+E[I,J];next J;next I
31: "START":for V=1 to 2;with M,10,10
32: "start":ent "Event",CS;CS=CS(1,24-len(CS))+CS
33: ent "BACKGROUND FILE?",J[1];if J[1]=J[4];sf=1
34: J[1]=J[4]
35: ent "BEGIN FILE",J[2];ent "ENDING FILE",J[3];ent "TIME DIF?",F
36: trk 0;for D=1 to 3;if fil;cf=1;imp 2
37: Lsf J[C],.3,CS=BS+AS[D];cll "sort"
38: next D;for I=1 to 6;for J=1 to 15
39: O[2,I,J]-O[1,I,J]+O[2,I,J];O[3,I,J]-O[1,I,J]+O[3,I,J]
40: if O[3,I,J]=0;-10000*S[I,J];qto "BELFAST"
41: O[2,I,J]/O[3,I,J]+O[2,I,J]*S[I,J]
42: if S[I,J]<0;-10000*S[I,J];imp 2
43: T/ln(S[I,J])+S[I,J]
44: "BELFAST":next J;next I;for I=2 to 3;O+C[3,1,I]+O[3,2,I];next I
45: for I=2 to 3;-10000*S[1,I]+S[2,I];next I
46: for D=2 to 2
47: "mat":ina A,F;for I=1 to 6;if I=4;qto "nEXtI"
48: for J=2 to 15;I=J
49: if O[3,I,J]=0;qto "nEXtJ"
50: O[D,I,J]+P;log(E[I,J])*X
51: "2":for K=0 to C;C+1-K*R;C+F[P,R]+F[R,R]
52: P+A[P]+A[F];P*X+P;C=J
53: if K=C;C=F[P-1,R]+F[P-1,R]
54: G*X=C;next K
55: "nEXtJ":next J
56: "nEXtI":next I
57: for I=1 to C+1
58: for K=1 to int((C+1)/2)
59: if I=1 and I+K<C+2 and I-K>0;F[I,I]+F[I-K,I+K]+F[I+K,I-K]
60: if I+K=C+1 and I-K+1>0;F[I,I+1]+F[I+K,I-K+1]
61: if I+K<C and I-K>0;F[I,I+1]+F[I-K,I+K+1]
62: next K;next I;inv F=F;in T=1;mat F=C;imp 3
63: if C=2;mat 1A+K;imp 2
64: mat FA+I
65: next I
66: "out":prt ".7",C[3],J[1],J[2],J[3],F
67: prt ".2",Y[1,2],Y[1,3],Y[1,1]
68: prt "Event",c25;prt " ",CS;web 1,10,13
69: with 1,"Tau (minutes)";with 4,10,13;for J=2 to 15;for I=1 to 6

```

THE BDM CORPORATION

```

70: if I=4;goto "GALWAY"
71: if S[I,J]<-9999;wth M,"      ",42,"      ";imo 3
72: if S[I,J]>9999;wth M,"      ",42,"      ";jmp 2
73: wrt "M.6",S[I,J]
74: "GALWAY":next I;wth M,10,13;next J;wth M,10,13
75: goto "crv"
76: wth M,27,65,0,0,int(1.5Y),int(96Y)
77: if Y#0 and Ymod1=0;wrt "M.2","-",2Y;goto +2
78: wth M,"|"
79: if (Y+.1+Y)<5.1;goto -3
80: goto "skip print"
81: "sort":for I=1 to 6;for J=1 to 15;J+15I-16+K
82: stf(AS[D,4K+1,4K+4])>C[D,I,J];next J;next I
83: C>K;for I=1 to 15
84: stf(AS[D,361+4K,364+4K])>X[I];1+K>K;if I>5;goto "nextI"
85: stf(AS[D,361+4K,364+4K])>Y[D,I];1+K>K;if I>3;goto "nextI"
86: itf(AS[D,361+4K,362+4K])>X[2I-1]
87: itf(AS[D,363+4K,364+4K])>X[2I];1+K>K
88: "nextI":next I
89: stf(AS[D,457,460])>Z[D]
90: ret
91: "skip print":-1+X
92: "crv":for I=1 to 8;for J=1 to 5;.05(3(J-1)+1)-1.05+X;C+Y
93: for I=1 to CH;YX=X[I]+Y;next I
94: Y+S[J,C]+Z;if Z<=0;-10000>S[J,C];imo 2
95: 1/ln(Z)>S[J,C]
96: next J;next I
97: wth M,13,"Fitted Curve";wth M,10,13;for J=1 to 8;for I=1 to 5
98: if S[I,J]<-9999;wth M,"      ",42,"      ";jmp 2
99: wrt "M.6",S[I,J]
100: next I;wth M,10,13;next J
101: next V
102: wth M,13
103: goto "SIASI"
104: end
*31767

```

THE BDM CORPORATION

DISTRIBUTION LIST

	<u>DODAAD Code</u>	<u>No. of Copies</u>
1. Mr. Robin Simpson, ONR Rep. Stanford University, Room 165 Durond Aeronautics Building Stanford, California 94305		1
2. Dean of Research, Code 012 Naval Postgraduate School Monterey, California 93940	N62271	3
3. Defense Documentation Center Cameron Station, Building 5 Alexandria, Virginia 22314	S47031	12
4. Library, Code 0212 Naval Postgraduate School Monterey, California 93940		
5. Director, Naval Research Lab. Attn: Code 2627 Washington, D.C. 20375	N00173	6
6. Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91106	N62879	1
7. Associate Professor K. Davidson Naval Postgraduate School Code 63Ds Monterey, California 93940		1
8. Professor Gordon Schacher Naval Postgraduate School Code 615q Monterey, California 93940		1

DAI
FILM